piped_DWDS_microbial_water_quality_analyses_v2

November 2, 2020

0.1 Version & History

This worksheet was created by Lauren C Kennedy for Effect of disinfectant residual, pH, and temperature on microbial abundance in piped drinking water distribution systems

0.2 1. Import libraries and data

[1]:
```
#load stuff
library(ggplot2)
library(reshape2)
library(scales)
library(tidyr)
library(extrafont)
library(ggpubr)
library(Hmisc)
library(stargazer)#output tables
library(DescTools) #gmean and gstd
library(dplyr)
library(lme4)
library(effects) #plot predicted values for models
library(broom.mixed)#to extract info from models
library(MuMIn) #model selection for glmms
library(goft)#goodness of fit test for gamma distribution
library(GGally) #check for multicolinearity
library(merTools)#glmms plot variation in fixed and random effects


options(jupyter.plot_mimetypes = c("text/plain", "image/png" )) #to make␣
 ↪conversion to pdf work properly

library(viridis)#great color options designed to be visible to people with␣
 ↪colorblindness

colors <- c("#89C5DA", "#DA5724", "#74D944", "#CE50CA", "#3F4921", "#C0717C",␣
 ↪"#CBD588", "#5F7FC7",
            "#673770", "#D3D93E", "#38333E", "#508578", "#D7C1B1", "#689030",␣
 ↪"#AD6F3B", "#CD9BCD",
```

```
            "#D14285", "#6DDE88", "#652926", "#7FDCC0", "#C84248", "#8569D5",␣
 ↪"#5E738F", "#D1A33D",
            "#8A7C64", "#599861")

blue_s<- c("#29453E","#7DD1BC","#508577","#579183", "#406B60")
pink_s<- c("#E97CFC","#AE5CBD","#683770","#733D7D","#502A57")
green_s<- c("#35593A", "#87E694","#5A9963","#62A66B","#4B8052")
brown_s<- c("#F2645C", "#B34944","#662A27","#732F2C","#4D1F1D")
grey_s<- c("#4A4235", "#D6C09A","#8A7C63","#96876C","#706551")

#sizing in mm based on https://www.elsevier.com/authors/author-schemas/
 ↪artwork-and-media-instructions/artwork-sizing
min_w=30
single.col_w= 90
half.col_w= 140
max_w= 190
max_h=240
```

```
Attaching package: 'tidyr'


The following object is masked from 'package:reshape2':

    smiths


Registering fonts with R

Loading required package: lattice

Loading required package: survival

Loading required package: Formula


Attaching package: 'Hmisc'


The following objects are masked from 'package:base':

    format.pval, units


Please cite as:
```

```
 Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary
Statistics Tables.

 R package version 5.2.2. https://CRAN.R-project.org/package=stargazer


Attaching package: ‘DescTools’


The following objects are masked from ‘package:Hmisc’:

    %nin%, Label, Mean, Quantile


Attaching package: ‘dplyr’


The following objects are masked from ‘package:Hmisc’:

    src, summarize


The following objects are masked from ‘package:stats’:

    filter, lag


The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union


Loading required package: Matrix


Attaching package: ‘Matrix’


The following objects are masked from ‘package:tidyr’:

    expand, pack, unpack


Registered S3 methods overwritten by 'lme4':
  method                          from
```

```
  cooks.distance.influence.merMod car
  influence.merMod                car
  dfbeta.influence.merMod         car
  dfbetas.influence.merMod        car

Loading required package: carData

Use the command
    lattice::trellis.par.set(effectsTheme())
  to customize lattice options for effects plots.
See ?effectsTheme for details.

Registered S3 method overwritten by 'broom.mixed':
  method      from
  tidy.gamlss broom

Loading required package: fitdistrplus

Loading required package: MASS


Attaching package: ‘MASS’


The following object is masked from ‘package:dplyr’:

    select


Loading required package: npsurv

Loading required package: lsei

Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2

Loading required package: arm


arm (Version 1.11-1, built: 2020-4-27)


Working directory is /Users/owner/Desktop/Berkeley_Work/General_Science/Papers/D
WDS_Survey/Final_submission/2020_0802_github
```

```
Attaching package: 'arm'


The following object is masked from 'package:scales':

    rescale


Attaching package: 'merTools'


The following object is masked from 'package:DescTools':

    ICC


Loading required package: viridisLite


Attaching package: 'viridis'


The following object is masked from 'package:scales':

    viridis_pal
```

[2]:
```
set.seed(30)
sessionInfo()
```

```
R version 3.6.2 (2019-12-12)
Platform: x86_64-apple-darwin15.6.0 (64-bit)
Running under: macOS Catalina 10.15.6

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] viridis_0.5.1       viridisLite_0.3.0   merTools_0.5.0
```

```
 [4] arm_1.11-1          GGally_2.0.0        goft_1.3.4
 [7] fitdistrplus_1.0-14 npsurv_0.4-0        lsei_1.2-0
[10] MASS_7.3-51.5       MuMIn_1.43.15       broom.mixed_0.2.4
[13] effects_4.1-4       carData_3.0-3       lme4_1.1-23
[16] Matrix_1.2-18       dplyr_1.0.2         DescTools_0.99.32
[19] stargazer_5.2.2     Hmisc_4.3-1         Formula_1.2-3
[22] survival_3.1-8      lattice_0.20-40     ggpubr_0.4.0
[25] extrafont_0.17      tidyr_1.1.2         scales_1.1.0
[28] reshape2_1.4.3      ggplot2_3.3.2

loaded via a namespace (and not attached):
 [1] minqa_1.2.4         colorspace_1.4-1    ggsignif_0.6.0
 [4] ellipsis_0.3.0      rio_0.5.16          IRdisplay_0.7.0
 [7] htmlTable_1.13.3    base64enc_0.1-3     rstudioapi_0.11
[10] mvtnorm_1.0-12      codetools_0.2-16    splines_3.6.2
[13] knitr_1.30          IRkernel_1.1        jsonlite_1.6.1
[16] nloptr_1.2.2.1      broom_0.7.2         Rttf2pt1_1.3.8
[19] cluster_2.1.0       png_0.1-7           shiny_1.4.0
[22] compiler_3.6.2      backports_1.1.5     fastmap_1.0.1
[25] survey_4.0          later_1.0.0         acepack_1.4.1
[28] htmltools_0.4.0     tools_3.6.2         coda_0.19-3
[31] gtable_0.3.0        glue_1.4.0          Rcpp_1.0.4.6
[34] cellranger_1.1.0    vctrs_0.3.4         nlme_3.1-145
[37] extrafontdb_1.0     iterators_1.0.12    xfun_0.18
[40] stringr_1.4.0       openxlsx_4.1.4      mime_0.9
[43] lifecycle_0.2.0     statmod_1.4.34      rstatix_0.6.0
[46] promises_1.1.0      hms_0.5.3           expm_0.999-4
[49] TMB_1.7.16          RColorBrewer_1.1-2  curl_4.3
[52] gridExtra_2.3       rpart_4.1-15        reshape_0.8.8
[55] latticeExtra_0.6-29 stringi_1.4.6       foreach_1.4.8
[58] blme_1.0-4          checkmate_2.0.0     boot_1.3-25
[61] zip_2.0.4           repr_1.1.0          rlang_0.4.8
[64] pkgconfig_2.0.3     evaluate_0.14       purrr_0.3.3
[67] htmlwidgets_1.5.1   tidyselect_1.1.0    plyr_1.8.6
[70] magrittr_1.5        R6_2.4.1            generics_0.0.2
[73] pbdZMQ_0.3-3        DBI_1.1.0           pillar_1.4.3
[76] haven_2.2.0         foreign_0.8-75      withr_2.1.2
[79] abind_1.4-5         nnet_7.3-12         tibble_3.0.4
[82] crayon_1.3.4        car_3.0-6           uuid_0.1-2
[85] jpeg_0.1-8.1        grid_3.6.2          readxl_1.3.1
[88] data.table_1.12.8   forcats_0.4.0       digest_0.6.25
[91] xtable_1.8-4        httpuv_1.5.2        stats4_3.6.2
[94] munsell_0.5.0       mitools_2.4
```

[3]:
```
#following the recommendation from this article https://www.r-bloggers.com/
 ↪how-to-cite-packages/
#here are all the packages used
```

```
library(knitr)
write_bib(x = .packages(), file="/Users/owner/Desktop/R_packages.bib", tweak =␣
↪TRUE, width = NULL,
    prefix = getOption("knitr.bib.prefix", "R-"))
```

```
tweaking Hmisc
```

[4]:
```
give.n <- function(x){
  return(c(y = -.2, label = length(x)))
  # experiment with the multiplier to find the perfect position
}
```

0.2.1 import csv files

[5]:
```
#import worksheet
#where is the data?
path="/your/path/to/data/"

#where should figures and tables go?
path_fig="/your/output/path/for/figures"
path_tab="/your/output/path/for/tables"

SGPI_all<-read.csv(file=paste(path,"SGPI_all.csv",sep=""))
SG_all<-read.csv(file=paste(path,"SG_all.csv",sep=""))
SG_all$stain<-"SG"
SGPI_all$stain<-"SGPI"
DWDS_F_dat<-read.csv(file=paste(path,"DWDS_F.csv",sep=""))

SG_all<- dplyr::select(SG_all,-"X")
SGPI_all<- dplyr::select(SGPI_all,-"X")
```

0.2.2 convert date columns

[6]:
```
#set date columns
SG_all$sample_date <- as.Date(SG_all$sample_date, "%m/%d/%y")
SGPI_all$sample_date <- as.Date(SGPI_all$sample_date, "%m/%d/%y")
```

0.2.3 set quantification limits

[7]:
```
#quantification Limits
HPC_lim<-0.9999
TCC_lim<-12
ICC_lim<-22
ATPt_lim<-0.0001
ATPi_lim<- 0.0000183
```

```
cl2_lim<-0.02
```

0.2.4 set types of data

[8]:
```
SG_all$intra_ATP_gmean_nM <- as.numeric(SG_all$intra_ATP_gmean_nM)
SG_all$extra_ATP_gmean_nM <- as.numeric(SG_all$extra_ATP_gmean_nM)
SG_all$total_ATP_gmean_nM <- as.numeric(SG_all$total_ATP_gmean_nM)
SG_all$intra_ATP_gmean_nM <- SG_all$intra_ATP_gmean_nM
SG_all$extra_ATP_gmean_nM <- SG_all$extra_ATP_gmean_nM
SG_all$total_ATP_gmean_nM <- SG_all$total_ATP_gmean_nM
SG_all$sampling_year <- as.factor(SG_all$sampling_year)


SGPI_all$intra_ATP_gmean_nM <- as.numeric(SGPI_all$intra_ATP_gmean_nM)
SGPI_all$extra_ATP_gmean_nM <- as.numeric(SGPI_all$extra_ATP_gmean_nM)
SGPI_all$total_ATP_gmean_nM <- as.numeric(SGPI_all$total_ATP_gmean_nM)
SGPI_all$sampling_year <- as.factor(SGPI_all$sampling_year)

unique(SG_all$site)
unique(SG_all$disinfectant)
SG_all$disinfectant<-gsub("chloramine", "Chloramine", SG_all$disinfectant)
SGPI_all$disinfectant<-gsub("chloramine", "Chloramine", SGPI_all$disinfectant)
unique(SG_all$disinfectant)
```

1. other_site_in_DWDS_F 2. site_15 3. site_10 4. site_24 5. site_ut 6. other_site_in_DWDS_D 7. other_site_in_DWDS_E 8. other_site_in_DWDS_C 9. other_site_in_DWDS_B 10. other_site_in_DWDS_A

Levels: 1. 'other_site_in_DWDS_A' 2. 'other_site_in_DWDS_B' 3. 'other_site_in_DWDS_C' 4. 'other_site_in_DWDS_D' 5. 'other_site_in_DWDS_E' 6. 'other_site_in_DWDS_F' 7. 'site_10' 8. 'site_15' 9. 'site_24' 10. 'site_ut'

1. Chloramine 2. Chlorine 3. chloramine

Levels: 1. 'chloramine' 2. 'Chloramine' 3. 'Chlorine'

1. 'Chloramine' 2. 'Chlorine'

0.3 2. Prep

0.3.1 Make long format data frames

[9]:
```
#make dfs in long format that need two parameters plotted
# ATP
ATP_long_avg= reshape2::melt(dplyr::select(SG_all,-core_id),␣
↪id=c("std_note","flow_cytometer", "sample_date", "location_code",␣
↪"broad_location", "disinfectant", "wtp", "total_Cl2_mg.L","free_Cl2_mg.
↪L","site","sampling_year","water_age_h"), measure=c("intra_ATP_gmean_nM",␣
↪"total_ATP_gmean_nM","extra_ATP_gmean_nM"))
```

```
ATP_long_std= reshape2::melt(dplyr::select(SG_all,-core_id),␣
 ↪id=c("std_note","flow_cytometer", "sample_date", "location_code",␣
 ↪"broad_location", "disinfectant", "wtp", "total_Cl2_mg.L","free_Cl2_mg.
 ↪L","site","sampling_year","water_age_h"), measure=c("intra_ATP_gstd_nM",␣
 ↪"total_ATP_gstd_nM","extra_ATP_gstd_nM"))
names(ATP_long_std)[names(ATP_long_std)=="value"] <- "ATP_stdev"
names(ATP_long_std)[names(ATP_long_std)=="variable"] <- "ATP_stdev_type"
ATP_long= full_join(ATP_long_avg, ATP_long_std)
ATP_long= subset(ATP_long,␣
 ↪substring(ATP_long$variable,1,5)==substring(ATP_long$ATP_stdev_type,1,5))


#SG + SGPI together
fcm_all_long=full_join(dplyr::select(SG_all,-core_id),dplyr::
 ↪select(SGPI_all,-core_id))
```

```
Joining, by = c("std_note", "flow_cytometer", "sample_date", "location_code",
"broad_location", "disinfectant", "wtp", "total_Cl2_mg.L", "free_Cl2_mg.L",
"site", "sampling_year", "water_age_h")

Joining, by = c("sample_date", "broad_location", "sampling_year",
"location_code", "site", "disinfectant", "wtp", "flow_cytometer",
"regrowth_key", "std_note", "HPC_label", "avg_cells_per_mL_gmean",
"avg_cells_per_mL_gstd", "ICC_to_TCC", "ICC_to_TCC_stdev", "intra_ATP_gmean_nM",
"intra_ATP_gstd_nM", "total_ATP_gmean_nM", "total_ATP_gstd_nM",
"extra_ATP_gmean_nM", "extra_ATP_gstd_nM", "HPC_gmean_MPN_per_100mL",
"HPC_gstd_MPN_per_100mL", "HPC_label.1", "pH", "temp_C", "total_Cl2_mg.L",
"free_Cl2_mg.L", "water_age_h", "summer_water_age_h", "stain")
```

[10]:
```
head(ATP_long)
dim(ATP_long)
```

A data.frame: 6 × 16

	std_note <fct>	flow_cytometer <fct>	sample_date <date>	location_code <fct>	broad_location <fct>	disinfe <chr>
1	NA	Canto	2018-02-09	site_9	DWDS_F	Chlora
4	NA	Canto	2018-02-09	site_1	DWDS_F	Chlora
7	NA	Canto	2018-02-09	site_7	DWDS_F	Chlora
10	NA	Canto	2018-02-09	site_15	DWDS_F	Chlora
13	NA	Canto	2018-02-09	site_10	DWDS_F	Chlora
16	NA	Canto	2018-02-09	site_16	DWDS_F	Chlora

1. 504 2. 16

[11]:
```
#mwq_all
```

```
FCM<-dcast(as.data.frame(fcm_all_long),␣
 ↪sampling_year+water_age_h+std_note+HPC_label+ wtp+ pH+ free_Cl2_mg.L+␣
 ↪total_Cl2_mg.L+ temp_C+ disinfectant+ sample_date+  broad_location+␣
 ↪location_code+ HPC_gmean_MPN_per_100mL ~ stain,  value.var=␣
 ↪c("avg_cells_per_mL_gmean"))
ATP<-dcast(as.data.frame(ATP_long), std_note+ sample_date+  broad_location+␣
 ↪location_code ~ variable,  value.var= c("value"))
mwq_all<-left_join(FCM,ATP)

FCM_s<-dcast(as.data.frame(fcm_all_long),␣
 ↪sampling_year+water_age_h+std_note+HPC_label+ wtp+ pH+ free_Cl2_mg.L+␣
 ↪total_Cl2_mg.L+ temp_C+ disinfectant+ sample_date+  broad_location+␣
 ↪location_code+  HPC_gstd_MPN_per_100mL ~ stain,  value.var=␣
 ↪c("avg_cells_per_mL_gstd"))
ATP_s<-dcast(as.data.frame(ATP_long), std_note+ sample_date+  broad_location+␣
 ↪location_code ~ ATP_stdev_type,  value.var= c("ATP_stdev"))
mwq_all_s<-left_join(FCM_s,ATP_s)
colnames(mwq_all_s)[colnames(mwq_all_s)== "SG"] <- "SG_gstd"
colnames(mwq_all_s)[colnames(mwq_all_s)== "SGPI"] <- "SGPI_gstd"

mwq_all<-left_join(mwq_all,mwq_all_s)


#define mwq_quant to be used for modeling purposes -- samples bdl are assigned␣
 ↪to be at the detection limit
#this value was chossen because it is a conservative representation of the BDL␣
 ↪values
#ADL values were not included (HPC only 5 samples) because they cannot be␣
 ↪conservatively represented

length(mwq_all$pH)#starting size
sum(!is.na(mwq_all$SGPI))#starting size
mwq_quant<-mwq_all
mwq_quant[(!is.na(mwq_quant$intra_ATP_gmean_nM))&(mwq_quant$intra_ATP_gmean_nM␣
 ↪<= ATPi_lim),"intra_ATP_gmean_nM"]<- ATPi_lim
mwq_quant[!is.na(mwq_quant$total_ATP_gmean_nM)&(mwq_quant$total_ATP_gmean_nM <=␣
 ↪ATPt_lim),"total_ATP_gmean_nM"]<- ATPt_lim
mwq_quant[!is.na(mwq_quant$SGPI)&(mwq_quant$SGPI <= ICC_lim),"SGPI"]<-ICC_lim
mwq_quant[!is.na(mwq_quant$SG)&(mwq_quant$SG <= TCC_lim),"SG"]<-TCC_lim
mwq_quant[!is.na(mwq_quant$free_Cl2_mg.L)&(mwq_quant$free_Cl2_mg.L <=␣
 ↪cl2_lim),"free_Cl2_mg.L"]<-cl2_lim
mwq_quant[!is.na(mwq_quant$total_Cl2_mg.L)&(mwq_quant$total_Cl2_mg.L <=␣
 ↪cl2_lim),"total_Cl2_mg.L"]<-cl2_lim
mwq_quant[!is.na(mwq_quant$HPC_label)&(mwq_quant$HPC_label ==␣
 ↪"BDL"),"HPC_gmean_MPN_per_100mL"]<-HPC_lim
```

```
length(mwq_quant$pH)#ending size
sum(!is.na(mwq_quant$SGPI))#starting size

#make copies to use if needed
mwq_all_c<- mwq_all
mwq_quant_c<- mwq_quant

## make dataframes from mwq_quant for individual assays to calculate ADL an BDL␣
 ↪percentages
mwq_quant_ATPi<-mwq_all[((mwq_all$intra_ATP_gmean_nM > ATPi_lim) | is.
 ↪na(mwq_all$intra_ATP_gmean_nM)),]
mwq_quant_ATPt<-mwq_all[((mwq_all$total_ATP_gmean_nM > ATPt_lim) | is.
 ↪na(mwq_all$total_ATP_gmean_nM)),]
mwq_quant_SGPI<-mwq_all[((mwq_all$SGPI > ICC_lim) | is.na(mwq_all$SGPI)),]
mwq_quant_SG<-mwq_all[((mwq_all$SG > TCC_lim) | is.na(mwq_all$SG)),]
mwq_quant_HPC<-mwq_all[((mwq_all$HPC_label != "BDL")| is.
 ↪na(mwq_all$HPC_label)),]
mwq_quant_HPC<-mwq_quant_HPC[((mwq_quant_HPC$HPC_label != "ADL")| is.
 ↪na(mwq_quant_HPC$HPC_label)),]

dim(mwq_all)
dim(mwq_quant)
```

```
Joining, by = c("std_note", "sample_date", "broad_location", "location_code")

Joining, by = c("std_note", "sample_date", "broad_location", "location_code")

Joining, by = c("sampling_year", "water_age_h", "std_note", "HPC_label", "wtp",
"pH", "free_Cl2_mg.L", "total_Cl2_mg.L", "temp_C", "disinfectant",
"sample_date", "broad_location", "location_code")
```

168

166

168

166

1. 168 2. 25

1. 168 2. 25

[12]:
```
a<-colnames(fcm_all_long)
b<-colnames(FCM)
# setdiff(a,b)
```

[13]:
```
sum(!is.na(mwq_quant_HPC$HPC_gmean_MPN_per_100mL))
sum(!is.na(mwq_quant_HPC$HPC_gstd_MPN_per_100mL))
```

```
sum(!is.na(mwq_all$HPC_gmean_MPN_per_100mL))
sum(!is.na(mwq_all_c$HPC_gmean_MPN_per_100mL))
```

83

83

102

102

[14]:
```
length(fcm_all_long$sample_date)
length(FCM$sample_date)
length(mwq_all$sample_date)
length(SGPI_all$sample_date)
```

336

168

168

168

0.4 3. Modeling

I am following the steps in this book: https://highstat.com/index.php/beginner-s-guide-to-glm-and-glmm

0.4.1 A. Data Exploration (Zuur et al. 2010)

Subset out NAs

[15]:
```
raw_dat<-mwq_quant
raw_dat<-subset(raw_dat,wtp=="No")
raw_dat$disinfectant<-droplevels(as.factor(raw_dat$disinfectant))
raw_dat$broad_location<-droplevels(as.factor(raw_dat$broad_location))
raw_dat$location_code<-droplevels(as.factor(raw_dat$location_code))
raw_dat<-subset(raw_dat,select=c( SGPI,intra_ATP_gmean_nM,␣
 ↪HPC_gmean_MPN_per_100mL, pH, temp_C, total_Cl2_mg.L,free_Cl2_mg.L,␣
 ↪broad_location, location_code,disinfectant  ))
dim(raw_dat)

raw_ICC<-subset(raw_dat,select=c( SGPI, pH, temp_C, total_Cl2_mg.L,free_Cl2_mg.
 ↪L, broad_location, location_code,disinfectant  ))
raw_ICC<-raw_ICC[which(complete.cases(raw_ICC)),]

raw_ICC[,c("pH", "temp_C", "total_Cl2_mg.L","free_Cl2_mg.
 ↪L")]<-scale(raw_ICC[,c("pH", "temp_C", "total_Cl2_mg.L","free_Cl2_mg.L")],␣
 ↪center = TRUE, scale = TRUE)
dim(raw_ICC)
```

```r
raw_ATPi<-subset(raw_dat,select=c( intra_ATP_gmean_nM, pH, temp_C, total_Cl2_mg.
 ↪L,free_Cl2_mg.L, broad_location, location_code,disinfectant  ))
raw_ATPi<-raw_ATPi[which(complete.cases(raw_ATPi)),]

raw_ATPi[,c("pH", "temp_C", "total_Cl2_mg.L","free_Cl2_mg.
 ↪L")]<-scale(raw_ATPi[,c("pH", "temp_C", "total_Cl2_mg.L","free_Cl2_mg.L")],␣
 ↪center = TRUE, scale = TRUE)
dim(raw_ATPi)

raw_HPC<-subset(raw_dat,select=c( HPC_gmean_MPN_per_100mL, pH, temp_C,␣
 ↪total_Cl2_mg.L,free_Cl2_mg.L, broad_location, location_code,disinfectant  ))
raw_HPC<-raw_HPC[which(complete.cases(raw_HPC)),]

raw_HPC[,c("pH", "temp_C", "total_Cl2_mg.L","free_Cl2_mg.
 ↪L")]<-scale(raw_HPC[,c("pH", "temp_C", "total_Cl2_mg.L","free_Cl2_mg.L")],␣
 ↪center = TRUE, scale = TRUE)
dim(raw_HPC)

CAM_ICC<-subset(raw_ICC, raw_ICC$disinfectant=="Chloramine")
CAM_ATPi<-subset(raw_ATPi, raw_ATPi$disinfectant=="Chloramine")
CAM_HPC<-subset(raw_HPC, raw_HPC$disinfectant=="Chloramine")


#p should be >0.05
# gamma_test(raw_dat$HPC_gmean_MPN_per_100mL) #yes -- thus cannot rule out gamma
# gamma_test(raw_dat$intra_ATP_gmean_nM) #yes -- thus cannot rule out gamma

gamma_test(raw_dat$SGPI) #yes -- thus cannot rule out gamma
lnorm_test(raw_dat$SGPI)#no
shapiro.test(raw_dat$SGPI) # NO

gamma_test(CAM_ICC$SGPI) #yes -- thus cannot rule out gamma
lnorm_test(CAM_ICC$SGPI)#no
shapiro.test(CAM_ICC$SGPI) #NO
```

1. 168 2. 10

1. 103 2. 8

1. 102 2. 8

1. 88 2. 8

```
Test of fit for the Gamma distribution

data:  raw_dat$SGPI
V = 1.4929, p-value = 0.2911
```

```
Test for the lognormal distribution based on a transformation to
normality

data:  raw_dat$SGPI
p-value = 0.0124


Shapiro-Wilk normality test

data:  raw_dat$SGPI
W = 0.58834, p-value < 2.2e-16


Test of fit for the Gamma distribution

data:  CAM_ICC$SGPI
V = 2.5479, p-value = 0.0716


Test for the lognormal distribution based on a transformation to
normality

data:  CAM_ICC$SGPI
p-value = 0.001115


Shapiro-Wilk normality test

data:  CAM_ICC$SGPI
W = 0.54411, p-value = 2.266e-14
```

Add sampling_frequency column to each

[16]:
```
raw_ICC<- left_join(raw_ICC, summarise(group_by(raw_ICC,location_code),count␣
 ↪=n()))
raw_ICC$count<-as.factor(raw_ICC$count)

raw_ATPi<- left_join(raw_ATPi, summarise(group_by(raw_ATPi,location_code),count␣
 ↪=n()))
raw_ATPi$count<-as.factor(raw_ATPi$count)

raw_HPC<- left_join(raw_HPC, summarise(group_by(raw_HPC,location_code),count␣
 ↪=n()))
raw_HPC$count<-as.factor(raw_HPC$count)
```

```
raw_dat<- left_join(raw_dat, summarise(group_by(raw_dat,location_code),count␣
 ↪=n()))
raw_dat$count<-as.factor(raw_dat$count)
```

```
`summarise()` ungrouping output (override with `.groups` argument)

Joining, by = "location_code"

`summarise()` ungrouping output (override with `.groups` argument)

Joining, by = "location_code"

`summarise()` ungrouping output (override with `.groups` argument)

Joining, by = "location_code"

`summarise()` ungrouping output (override with `.groups` argument)

Joining, by = "location_code"
```

Are there outliers?

[17]:
```
colnames(raw_dat)
head(raw_dat)
```

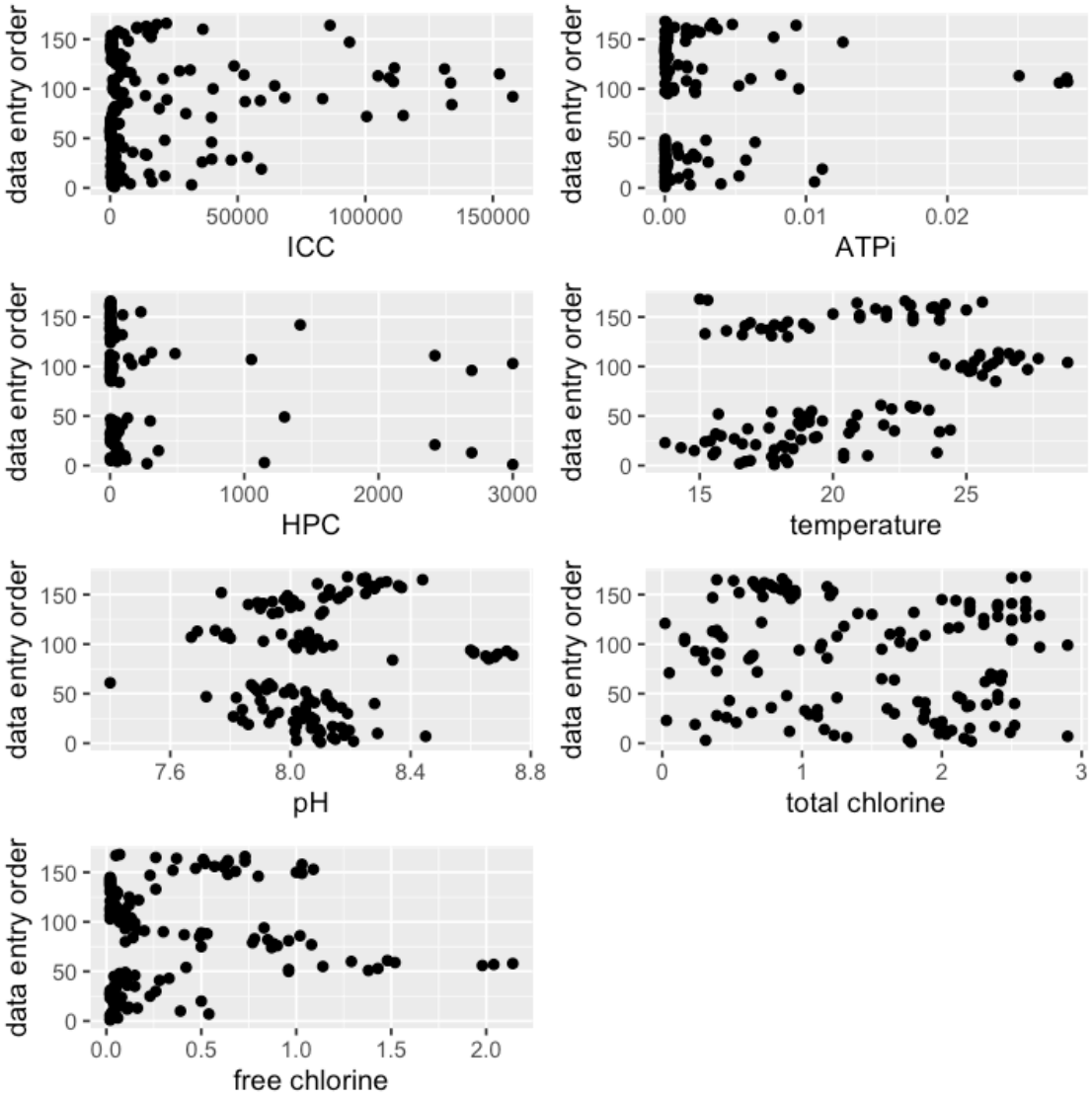
1. 'SGPI' 2. 'intra_ATP_gmean_nM' 3. 'HPC_gmean_MPN_per_100mL' 4. 'pH' 5. 'temp_C' 6. 'total_Cl2_mg.L' 7. 'free_Cl2_mg.L' 8. 'broad_location' 9. 'location_code' 10. 'disinfectant' 11. 'count'

A data.frame: 6 × 11

	SGPI <dbl>	intra_ATP_gmean_nM <dbl>	HPC_gmean_MPN_per_100mL <dbl>	pH <dbl>	te <dbl>
1	1721.4856	0.000034100	2999.90000	8.10	1
2	862.8820	0.000018300	275.70305	8.21	1
3	31952.1376	0.001801706	1149.55878	8.02	1
4	7889.9774	0.003975449	53.93774	8.15	1
5	614.4766	0.000018300	4.64758	8.09	1
6	16469.3871	0.010599329	117.89665	8.14	1

[18]:
```
options(warn=-1)# turn off warnings for plots + font extra


a<-raw_dat
a$rw<-seq.int(nrow(a))
raw_dat_num<-raw_dat[,c("SGPI", "intra_ATP_gmean_nM","HPC_gmean_MPN_per_100mL",␣
 ↪"temp_C","pH","total_Cl2_mg.L","free_Cl2_mg.L")]
```

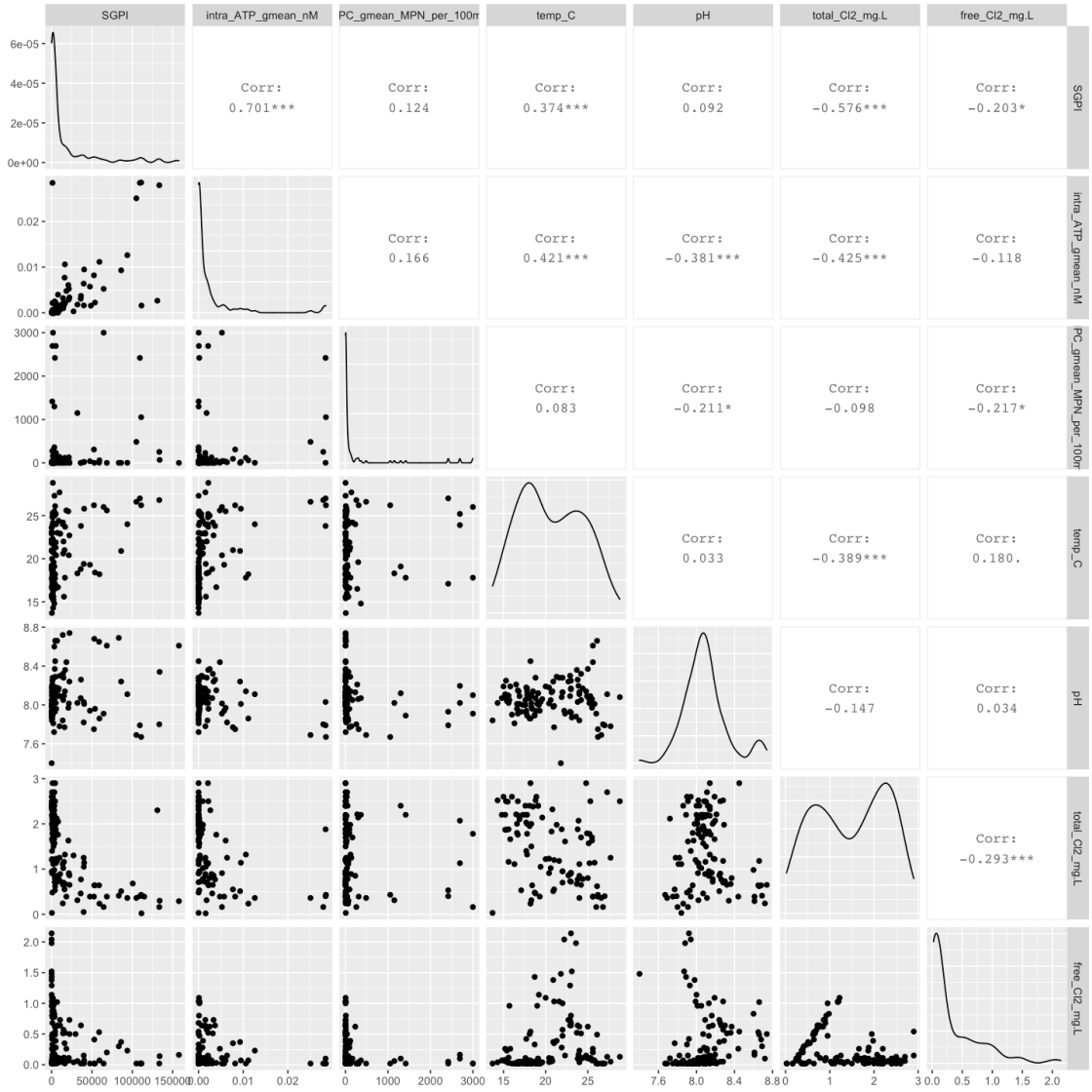
```
ylabel<-ylab("data entry order")
plt<-geom_point()
options(repr.plot.width = 6, repr.plot.height = 6) #for plotting size in jupyter

aa<-ggplot(a, aes(x = SGPI, y = rw)) +
    plt+
    xlab("ICC") +
    ylabel
b<-ggplot(a, aes(x = intra_ATP_gmean_nM, y = rw)) +
    plt+
    xlab("ATPi") +
    ylabel
c<-ggplot(a, aes(x = HPC_gmean_MPN_per_100mL, y = rw)) +
    plt+
    xlab("HPC") +
    ylabel
d<-ggplot(a, aes(x = temp_C, y = rw)) +
    plt+
    xlab("temperature") +
    ylabel
e<-ggplot(a, aes(x = pH, y = rw)) +
    geom_point()+
    xlab("pH") +
    ylabel
f<-ggplot(a, aes(x = total_Cl2_mg.L, y = rw)) +
    plt+
    xlab("total chlorine") +
   ylabel
g<-ggplot(a, aes(x = free_Cl2_mg.L, y = rw)) +
    plt+
    xlab("free chlorine") +
    ylabel

ggarrange(aa,  b,c,d,e,f,g,
          ncol = 2, nrow = 4 , legend="none")
```



Possibly some outliers at the high end of ATPi and HPC

Is there colinearity in continuous variables?

[19]:
```
options(repr.plot.width = 12, repr.plot.height = 12) #for plotting size in␣
↪jupyter
ggpairs(raw_dat_num)
```

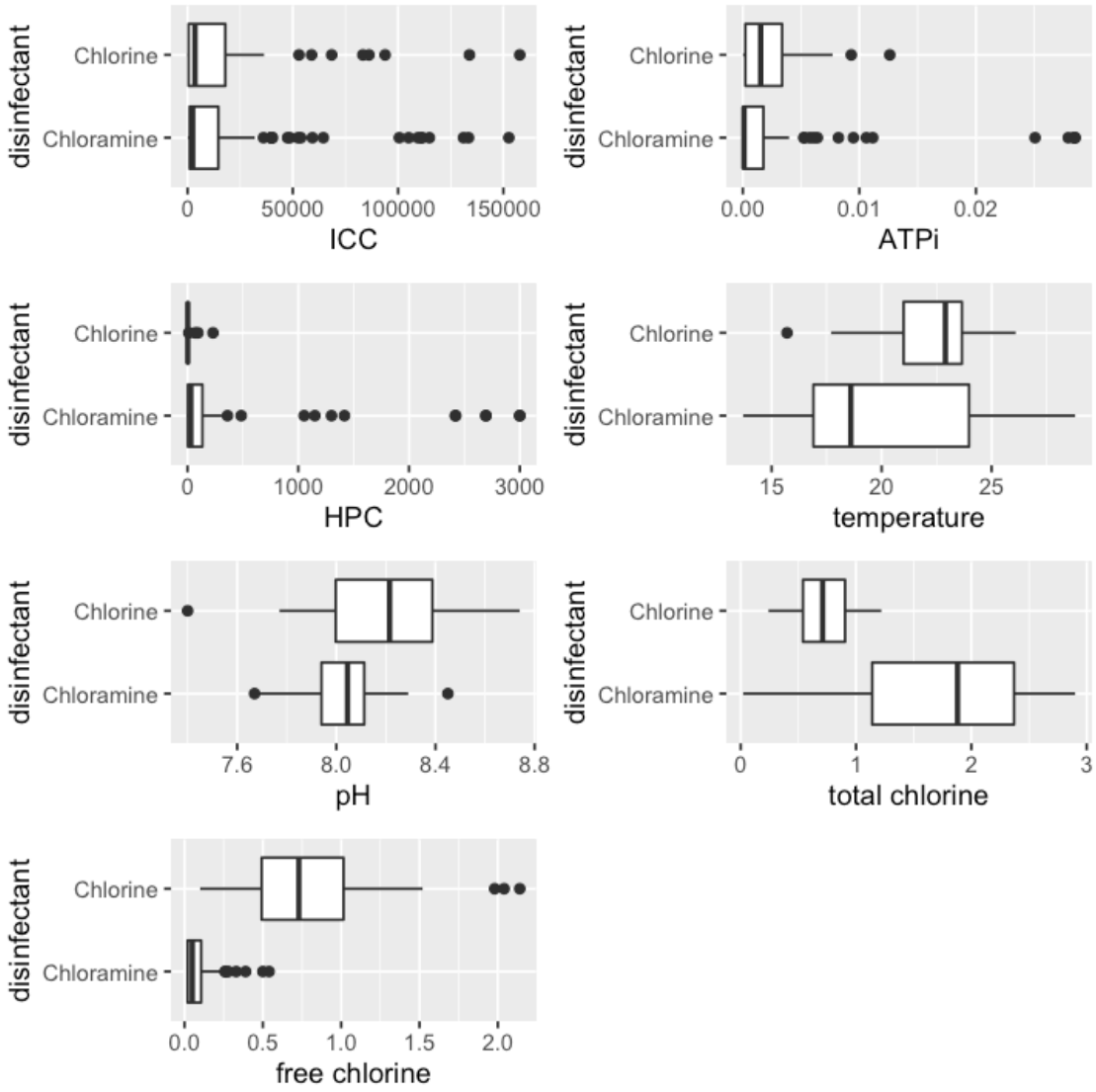


yes, pH and free chlorine covary

Is there covariation in categorical variable (disinfectant used)?

[20]:
```
a<-raw_dat
a<-a%>% rename("rw" = disinfectant)
ylabel<-ylab("disinfectant")
plt<-geom_boxplot()
options(repr.plot.width = 6, repr.plot.height = 6) #for plotting size in jupyter

aa<-ggplot(a, aes(x = SGPI, y = rw)) +
    plt+
    xlab("ICC") +
    ylabel
```

```
b<-ggplot(a, aes(x = intra_ATP_gmean_nM, y = rw)) +
    plt+
    xlab("ATPi") +
    ylabel
c<-ggplot(a, aes(x = HPC_gmean_MPN_per_100mL, y = rw)) +
    plt+
    xlab("HPC") +
    ylabel
d<-ggplot(a, aes(x = temp_C, y = rw)) +
    plt+
    xlab("temperature") +
    ylabel
e<-ggplot(a, aes(x = pH, y = rw)) +
    plt+
    xlab("pH") +
    ylabel
f<-ggplot(a, aes(x = total_Cl2_mg.L, y = rw)) +
    plt+
    xlab("total chlorine") +
   ylabel
g<-ggplot(a, aes(x = free_Cl2_mg.L, y = rw)) +
    plt+
    xlab("free chlorine") +
    ylabel


ggarrange(aa,  b,c,d,e,f,g,
          ncol = 2, nrow = 4 , legend="none")
```

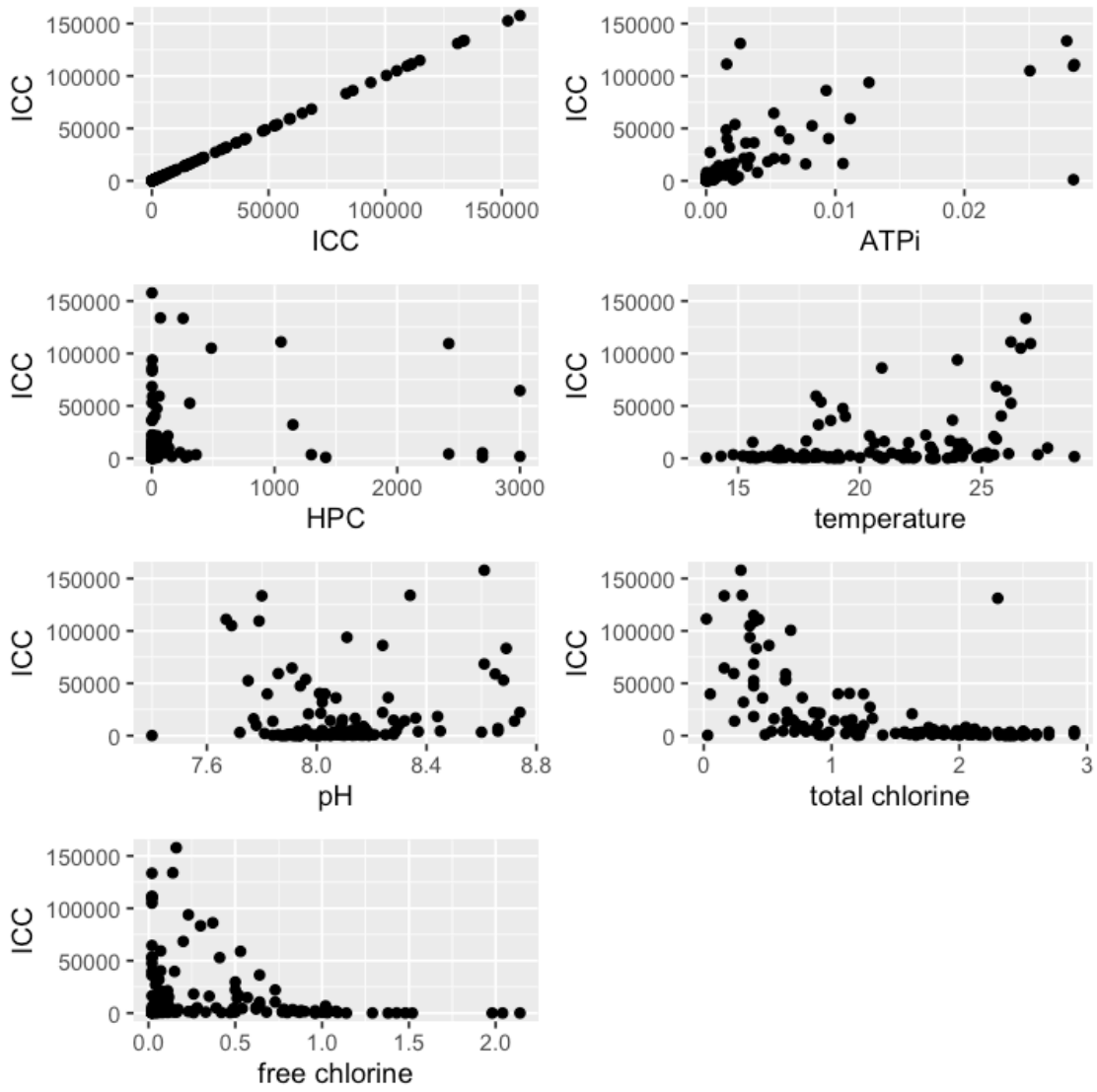


all covariates depend on the disinfectant type used in the DWDSs

What is the relationship between each fixed variable and each microbial abundance measure?

[21]:
```
a<-raw_dat
a$rw<-a$SGPI
raw_dat_num<-raw_dat[,c("SGPI", "intra_ATP_gmean_nM","HPC_gmean_MPN_per_100mL",␣
↪"temp_C","pH","total_Cl2_mg.L","free_Cl2_mg.L")]


ylabel<-ylab("ICC")
plt<-geom_point()
options(repr.plot.width = 6, repr.plot.height = 6) #for plotting size in jupyter
```

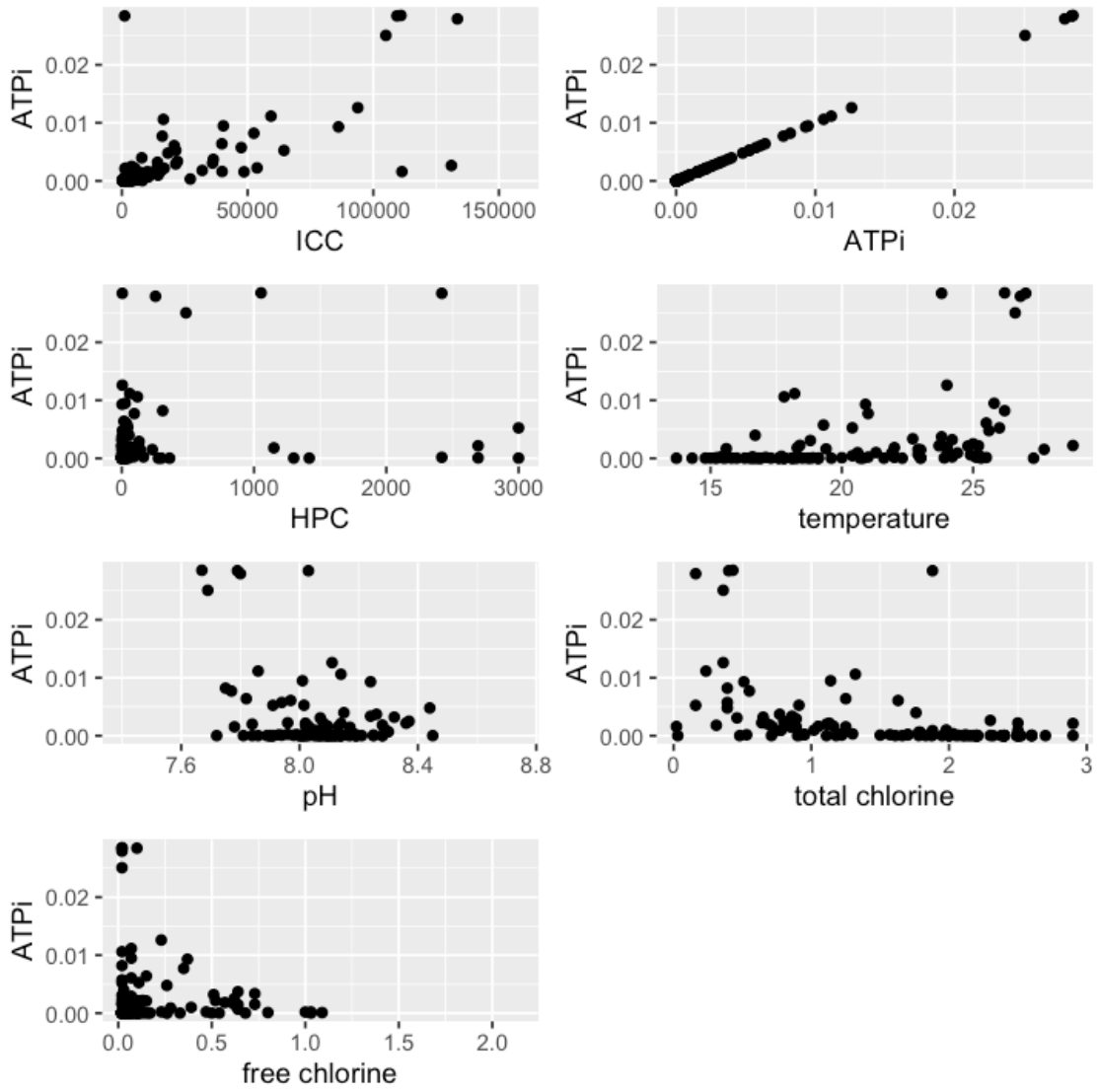
```
aa<-ggplot(a, aes(x = SGPI, y = rw)) +
    plt+
    xlab("ICC") +
    ylabel
b<-ggplot(a, aes(x = intra_ATP_gmean_nM, y = rw)) +
    plt+
    xlab("ATPi") +
    ylabel
c<-ggplot(a, aes(x = HPC_gmean_MPN_per_100mL, y = rw)) +
    plt+
    xlab("HPC") +
    ylabel
d<-ggplot(a, aes(x = temp_C, y = rw)) +
    plt+
    xlab("temperature") +
    ylabel
e<-ggplot(a, aes(x = pH, y = rw)) +
    geom_point()+
    xlab("pH") +
    ylabel
f<-ggplot(a, aes(x = total_Cl2_mg.L, y = rw)) +
    plt+
    xlab("total chlorine") +
   ylabel
g<-ggplot(a, aes(x = free_Cl2_mg.L, y = rw)) +
    plt+
    xlab("free chlorine") +
    ylabel

ggarrange(aa,  b,c,d,e,f,g,
          ncol = 2, nrow = 4 , legend="none")
```



Only a linear realitonship between SGPI and ATPi

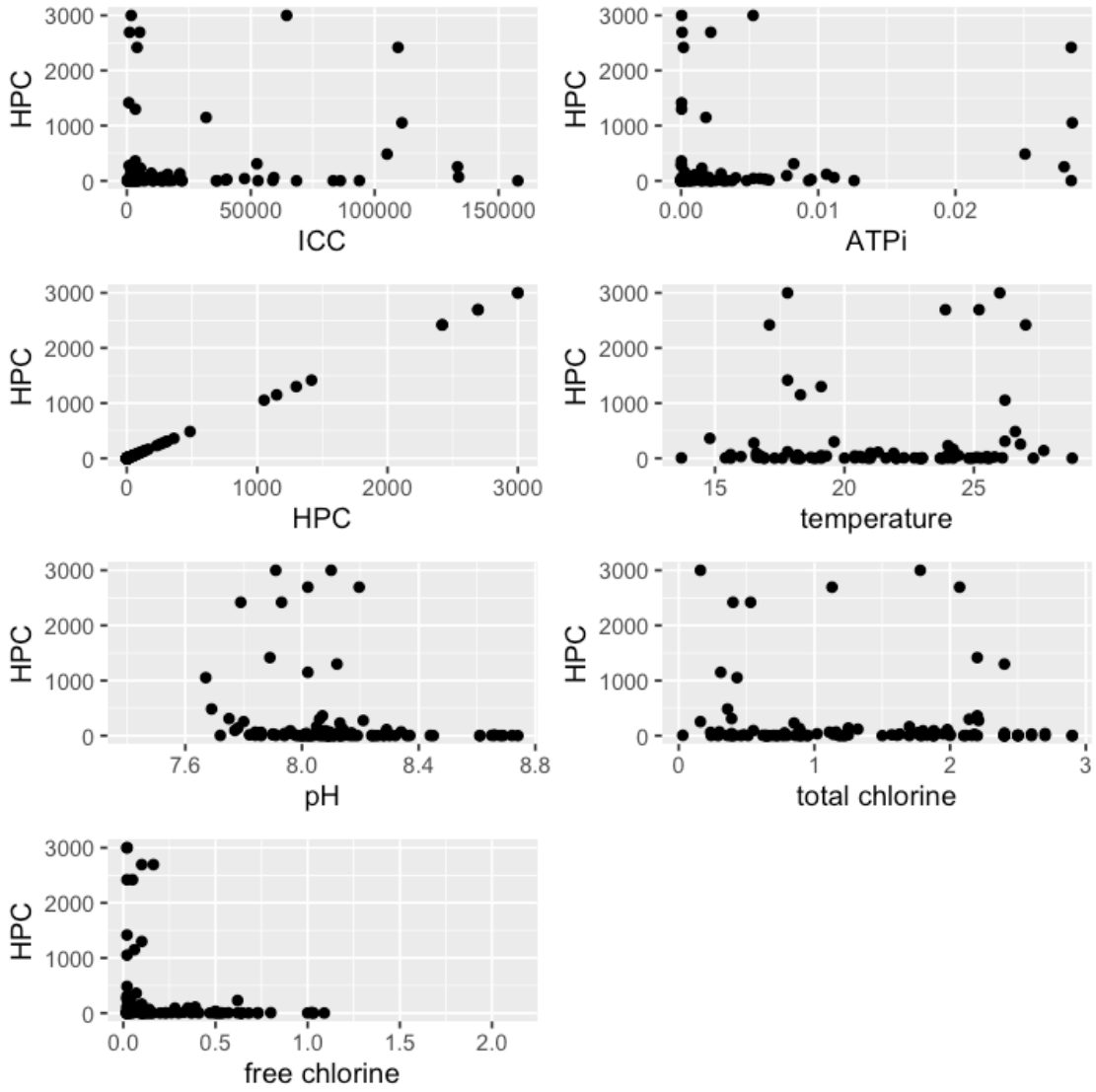
[22]:
```
a<-raw_dat
a$rw<-a$intra_ATP_gmean_nM
raw_dat_num<-raw_dat[,c("SGPI", "intra_ATP_gmean_nM","HPC_gmean_MPN_per_100mL",␣
↪"temp_C","pH","total_Cl2_mg.L","free_Cl2_mg.L")]


ylabel<-ylab("ATPi")
plt<-geom_point()
options(repr.plot.width = 6, repr.plot.height = 6) #for plotting size in jupyter

aa<-ggplot(a, aes(x = SGPI, y = rw)) +
```

```
    plt+
    xlab("ICC") +
    ylabel
b<-ggplot(a, aes(x = intra_ATP_gmean_nM, y = rw)) +
    plt+
    xlab("ATPi") +
    ylabel
c<-ggplot(a, aes(x = HPC_gmean_MPN_per_100mL, y = rw)) +
    plt+
    xlab("HPC") +
    ylabel
d<-ggplot(a, aes(x = temp_C, y = rw)) +
    plt+
    xlab("temperature") +
    ylabel
e<-ggplot(a, aes(x = pH, y = rw)) +
    geom_point()+
    xlab("pH") +
    ylabel
f<-ggplot(a, aes(x = total_Cl2_mg.L, y = rw)) +
    plt+
    xlab("total chlorine") +
   ylabel
g<-ggplot(a, aes(x = free_Cl2_mg.L, y = rw)) +
    plt+
    xlab("free chlorine") +
    ylabel

ggarrange(aa,  b,c,d,e,f,g,
          ncol = 2, nrow = 4 , legend="none")
```



[23]:
```
a<-raw_dat
a$rw<-a$HPC_gmean_MPN_per_100mL
raw_dat_num<-raw_dat[,c("SGPI", "intra_ATP_gmean_nM","HPC_gmean_MPN_per_100mL",␣
↪"temp_C","pH","total_Cl2_mg.L","free_Cl2_mg.L")]


ylabel<-ylab("HPC")
plt<-geom_point()
options(repr.plot.width = 6, repr.plot.height = 6) #for plotting size in jupyter

aa<-ggplot(a, aes(x = SGPI, y = rw)) +
    plt+
    xlab("ICC") +
```

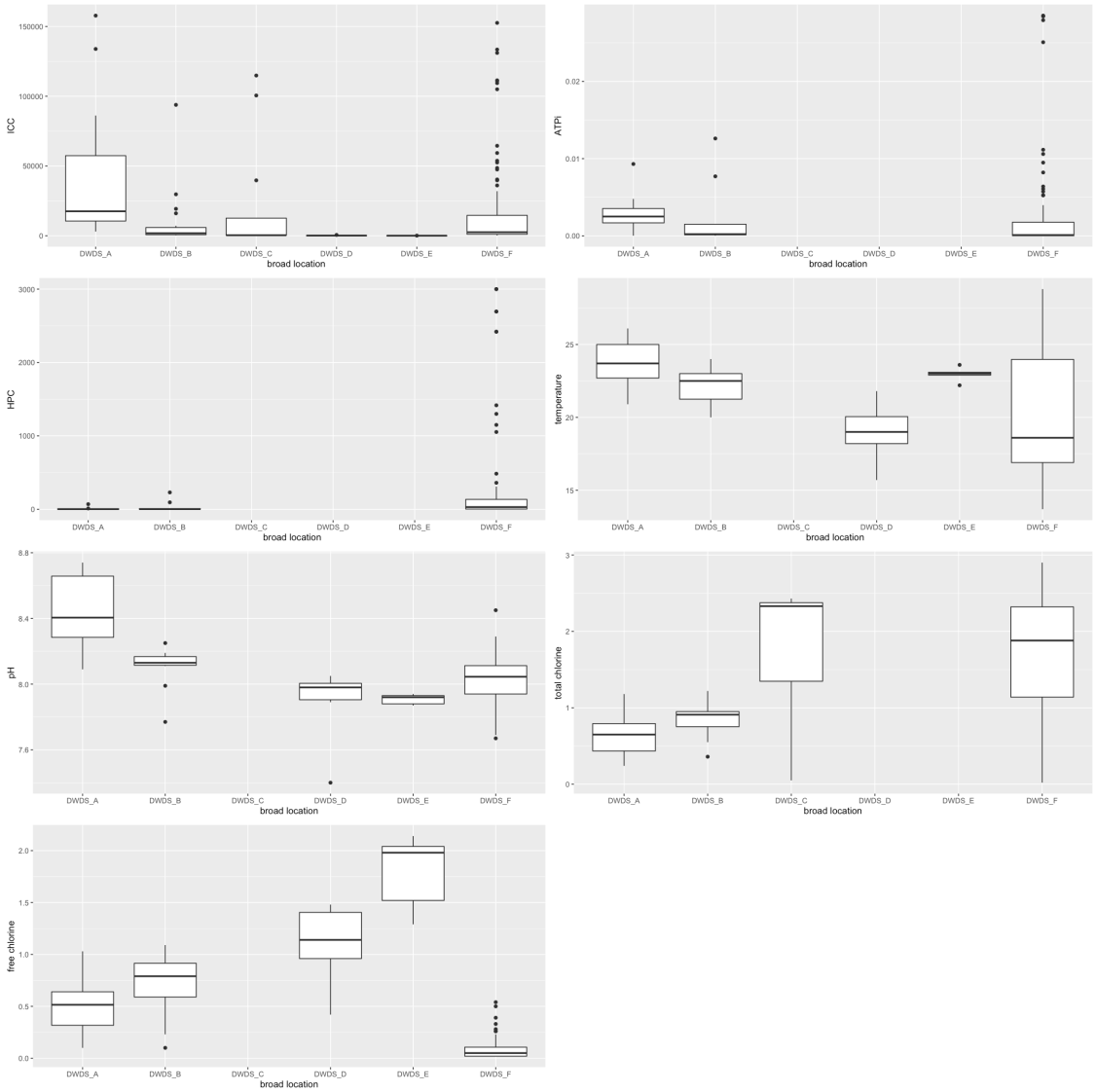
```
    ylabel
b<-ggplot(a, aes(x = intra_ATP_gmean_nM, y = rw)) +
    plt+
    xlab("ATPi") +
    ylabel
c<-ggplot(a, aes(x = HPC_gmean_MPN_per_100mL, y = rw)) +
    plt+
    xlab("HPC") +
    ylabel
d<-ggplot(a, aes(x = temp_C, y = rw)) +
    plt+
    xlab("temperature") +
    ylabel
e<-ggplot(a, aes(x = pH, y = rw)) +
    geom_point()+
    xlab("pH") +
    ylabel
f<-ggplot(a, aes(x = total_Cl2_mg.L, y = rw)) +
    plt+
    xlab("total chlorine") +
   ylabel
g<-ggplot(a, aes(x = free_Cl2_mg.L, y = rw)) +
    plt+
    xlab("free chlorine") +
    ylabel

ggarrange(aa,  b,c,d,e,f,g,
          ncol = 2, nrow = 4 , legend="none")
```



Are there clear dependency structures in the data?

[24]:
```
a<-raw_dat
a$rw<-a$broad_location
raw_dat_num<-raw_dat[,c("SGPI", "intra_ATP_gmean_nM","HPC_gmean_MPN_per_100mL",␣
↪"temp_C","pH","total_Cl2_mg.L","free_Cl2_mg.L")]


xlabel<-xlab("broad location")
wrap<-theme()
plt<-geom_boxplot()
options(repr.plot.width = 18, repr.plot.height = 18) #for plotting size in␣
↪jupyter
```

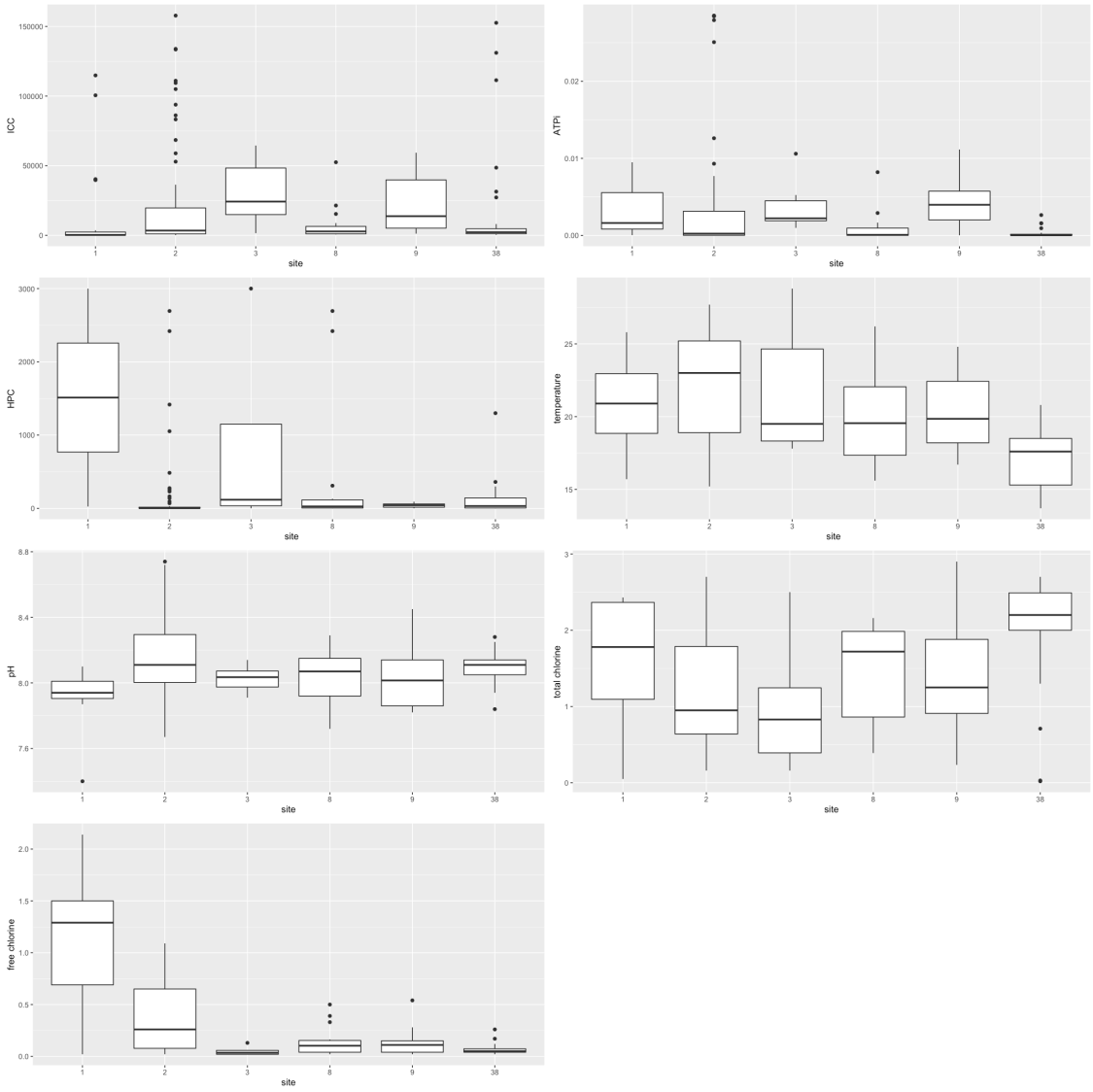
```
aa<-ggplot(a, aes(y = SGPI, x = rw)) +
    plt+
    ylab("ICC") +
    xlabel +
    wrap
b<-ggplot(a, aes(y = intra_ATP_gmean_nM, x = rw)) +
    plt+
    ylab("ATPi") +
    xlabel +
    wrap
c<-ggplot(a, aes(y = HPC_gmean_MPN_per_100mL, x = rw)) +
    plt+
    ylab("HPC") +
    xlabel +
    wrap
d<-ggplot(a, aes(y = temp_C, x = rw)) +
    plt+
    ylab("temperature") +
    xlabel +
    wrap
e<-ggplot(a, aes(y = pH, x = rw)) +
    plt+
    ylab("pH") +
    xlabel +
    wrap
f<-ggplot(a, aes(y = total_Cl2_mg.L, x = rw)) +
    plt+
    ylab("total chlorine") +
    xlabel +
    wrap
g<-ggplot(a, aes(y = free_Cl2_mg.L, x = rw)) +
    plt+
    ylab("free chlorine") +
    xlabel +
    wrap

ggarrange(aa,  b,c,d,e,f,g,
          ncol = 2, nrow = 4 , legend="none")
```



variation by broad locaion for most variables

[25]:
```
a<-raw_dat
a$rw<-a$count
raw_dat_num<-raw_dat[,c("SGPI", "intra_ATP_gmean_nM","HPC_gmean_MPN_per_100mL",␣
↪"temp_C","pH","total_Cl2_mg.L","free_Cl2_mg.L")]


xlabel<-xlab("site")
wrap<-theme()
plt<-geom_boxplot()
options(repr.plot.width = 18, repr.plot.height = 18) #for plotting size in␣
↪jupyter
```

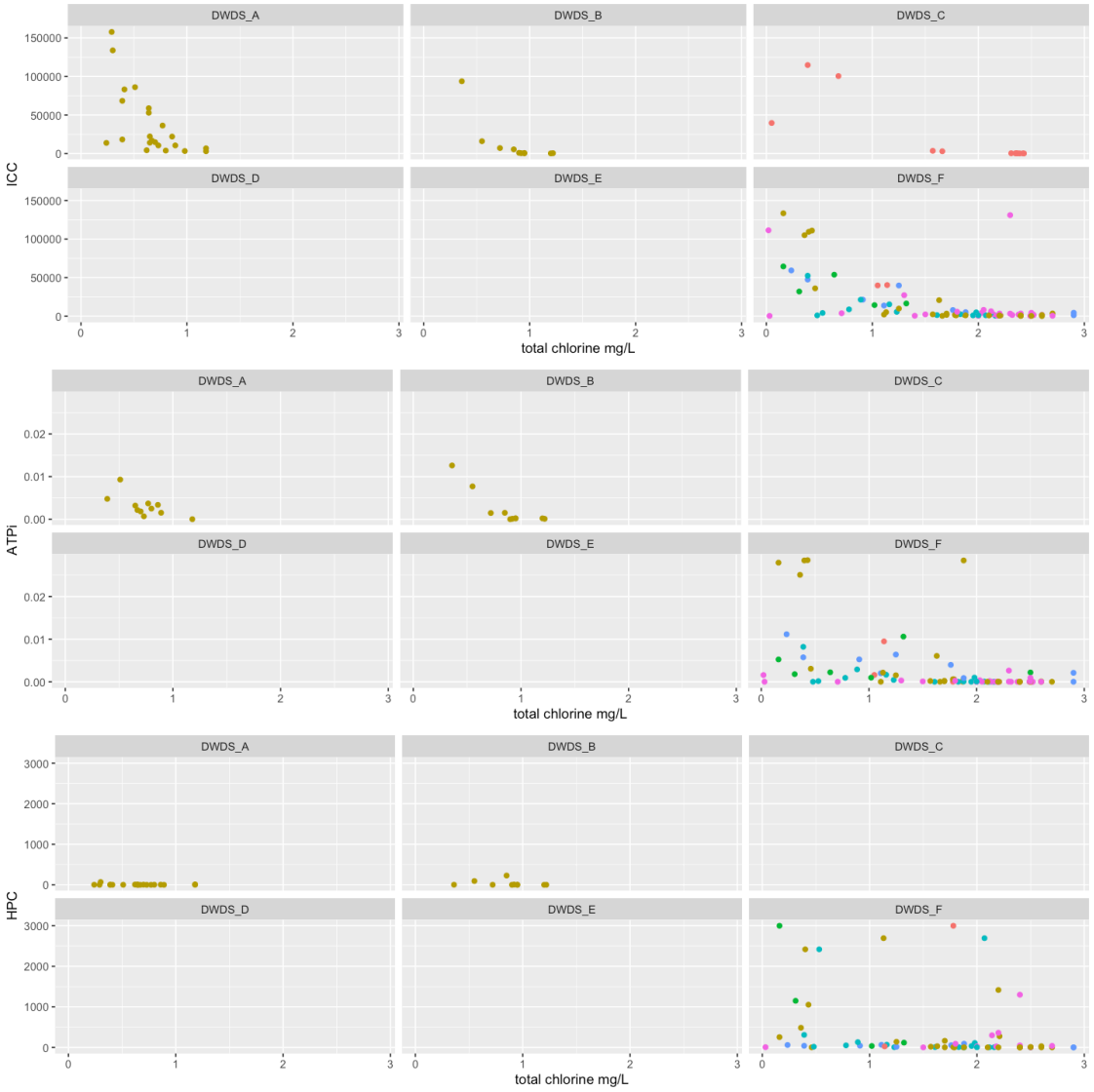
```
aa<-ggplot(a, aes(y = SGPI, x = rw)) +
    plt+
    ylab("ICC") +
    xlabel +
    wrap
b<-ggplot(a, aes(y = intra_ATP_gmean_nM, x = rw)) +
    plt+
    ylab("ATPi") +
    xlabel +
    wrap
c<-ggplot(a, aes(y = HPC_gmean_MPN_per_100mL, x = rw)) +
    plt+
    ylab("HPC") +
    xlabel +
    wrap
d<-ggplot(a, aes(y = temp_C, x = rw)) +
    plt+
    ylab("temperature") +
    xlabel +
    wrap
e<-ggplot(a, aes(y = pH, x = rw)) +
    plt+
    ylab("pH") +
    xlabel +
    wrap
f<-ggplot(a, aes(y = total_Cl2_mg.L, x = rw)) +
    plt+
    ylab("total chlorine") +
    xlabel +
    wrap
g<-ggplot(a, aes(y = free_Cl2_mg.L, x = rw)) +
    plt+
    ylab("free chlorine") +
    xlabel +
    wrap

ggarrange(aa,  b,c,d,e,f,g,
          ncol = 2, nrow = 4 , legend="none")
```



variation by site in most variables

[26]:
```
a<-raw_dat
a$rw<-a$total_Cl2_mg.L
raw_dat_num<-raw_dat[,c("SGPI", "intra_ATP_gmean_nM","HPC_gmean_MPN_per_100mL",␣
↪"temp_C","pH","total_Cl2_mg.L","free_Cl2_mg.L")]


xlabel<-xlab("total chlorine mg/L")
wrap<-facet_wrap(~broad_location)
plt<-geom_point()
options(repr.plot.width = 12, repr.plot.height = 12) #for plotting size in␣
↪jupyter
```

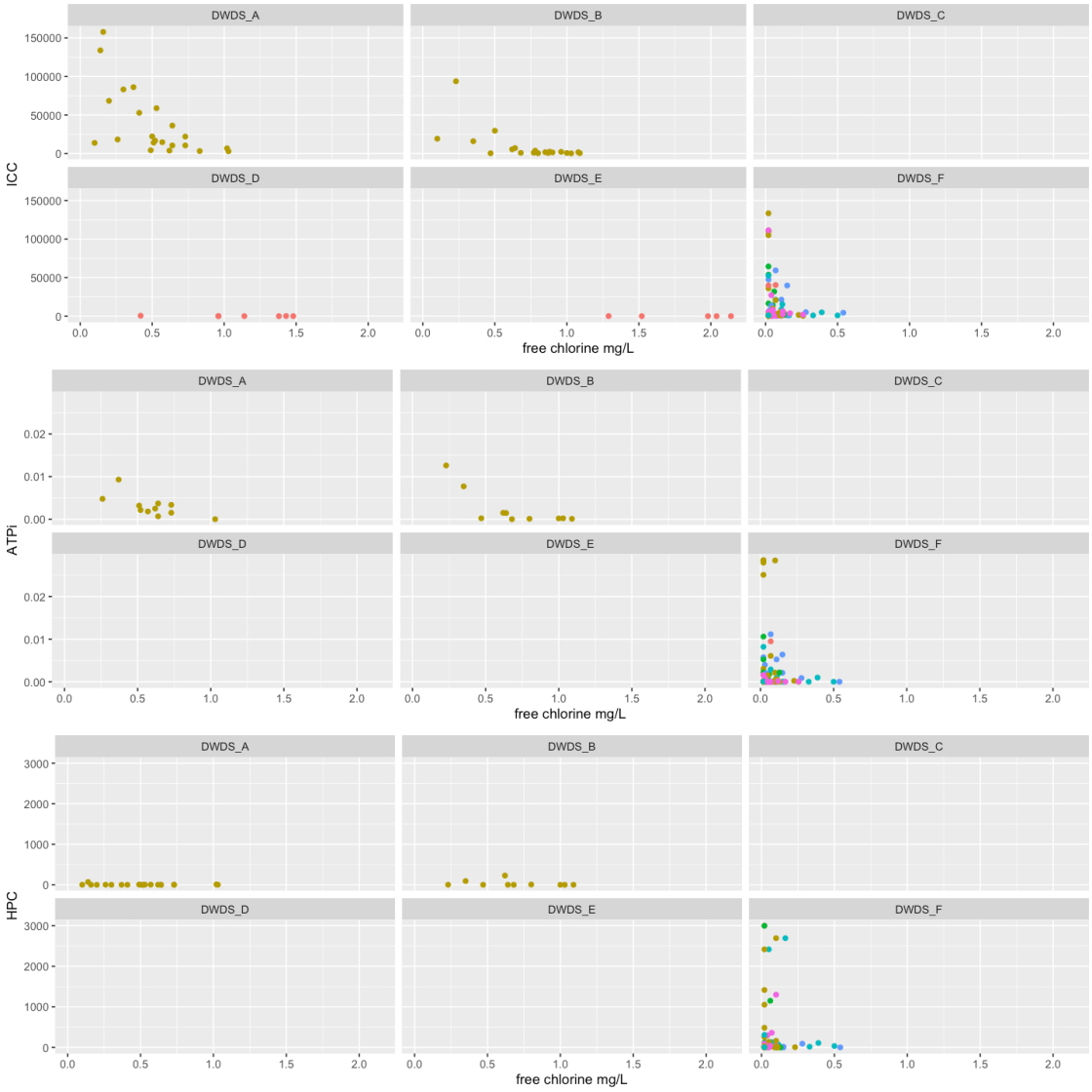
```
aa<-ggplot(a, aes(y = SGPI, x = rw, color= count)) +
    plt+
    ylab("ICC") +
    xlabel +
    wrap
b<-ggplot(a, aes(y = intra_ATP_gmean_nM, x = rw, color= count)) +
    plt+
    ylab("ATPi") +
    xlabel +
    wrap
c<-ggplot(a, aes(y = HPC_gmean_MPN_per_100mL, x = rw, color= count)) +
    plt+
    ylab("HPC") +
    xlabel +
    wrap


ggarrange(aa,  b,c,
          ncol = 1, nrow = 3, legend="none")
```



possibly exponential relationship between ICC & total chlorine and ATPi & total chlorine, looks similar by site, but chlorinated systems (A & B) tend to have lower total chlorine

[27]:
```
a<-raw_dat
a$rw<-a$free_Cl2_mg.L
raw_dat_num<-raw_dat[,c("SGPI", "intra_ATP_gmean_nM","HPC_gmean_MPN_per_100mL",␣
 ↪"temp_C","pH","total_Cl2_mg.L","free_Cl2_mg.L")]


xlabel<-xlab("free chlorine mg/L")
wrap<-facet_wrap(~broad_location)
plt<-geom_point()
```

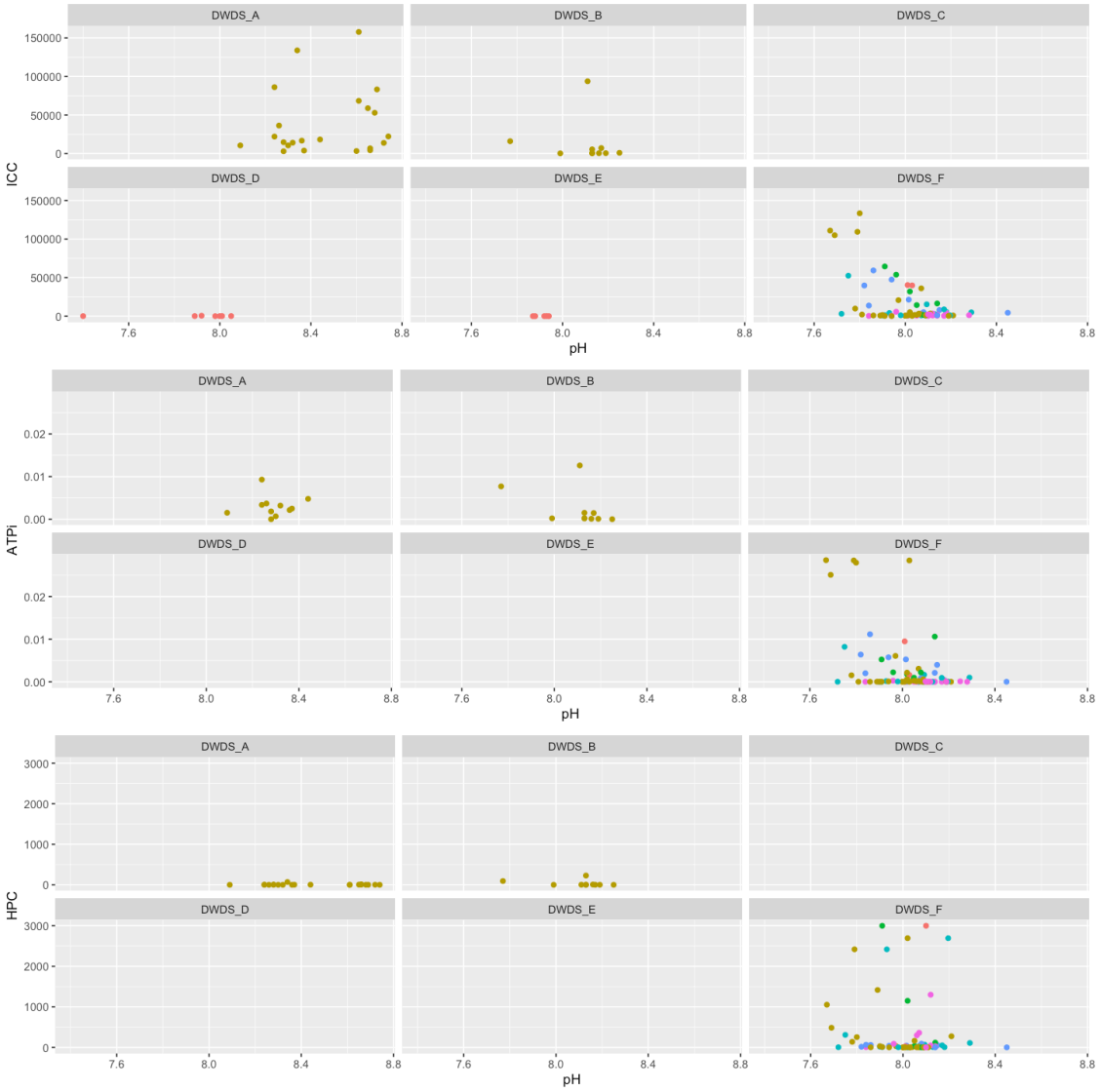
```
options(repr.plot.width = 12, repr.plot.height = 12) #for plotting size in␣
 ↪jupyter

aa<-ggplot(a, aes(y = SGPI, x = rw, color= count)) +
    plt+
    ylab("ICC") +
    xlabel +
    wrap
b<-ggplot(a, aes(y = intra_ATP_gmean_nM, x = rw, color= count)) +
    plt+
    ylab("ATPi") +
    xlabel +
    wrap
c<-ggplot(a, aes(y = HPC_gmean_MPN_per_100mL, x = rw, color= count)) +
    plt+
    ylab("HPC") +
    xlabel +
    wrap


ggarrange(aa,  b,c,
          ncol = 1, nrow = 3, legend="none")
```



possibly exponential relationship between ICC & total chlorine and ATPi & total chlorine, looks similar by site, but free chlorinated systems have higher free chlorine concentrations

[28]:
```
a<-raw_dat
a$rw<-a$pH
raw_dat_num<-raw_dat[,c("SGPI", "intra_ATP_gmean_nM","HPC_gmean_MPN_per_100mL",␣
↪"temp_C","pH","total_Cl2_mg.L","free_Cl2_mg.L")]


xlabel<-xlab("pH")
wrap<-facet_wrap(~broad_location)
plt<-geom_point()
```

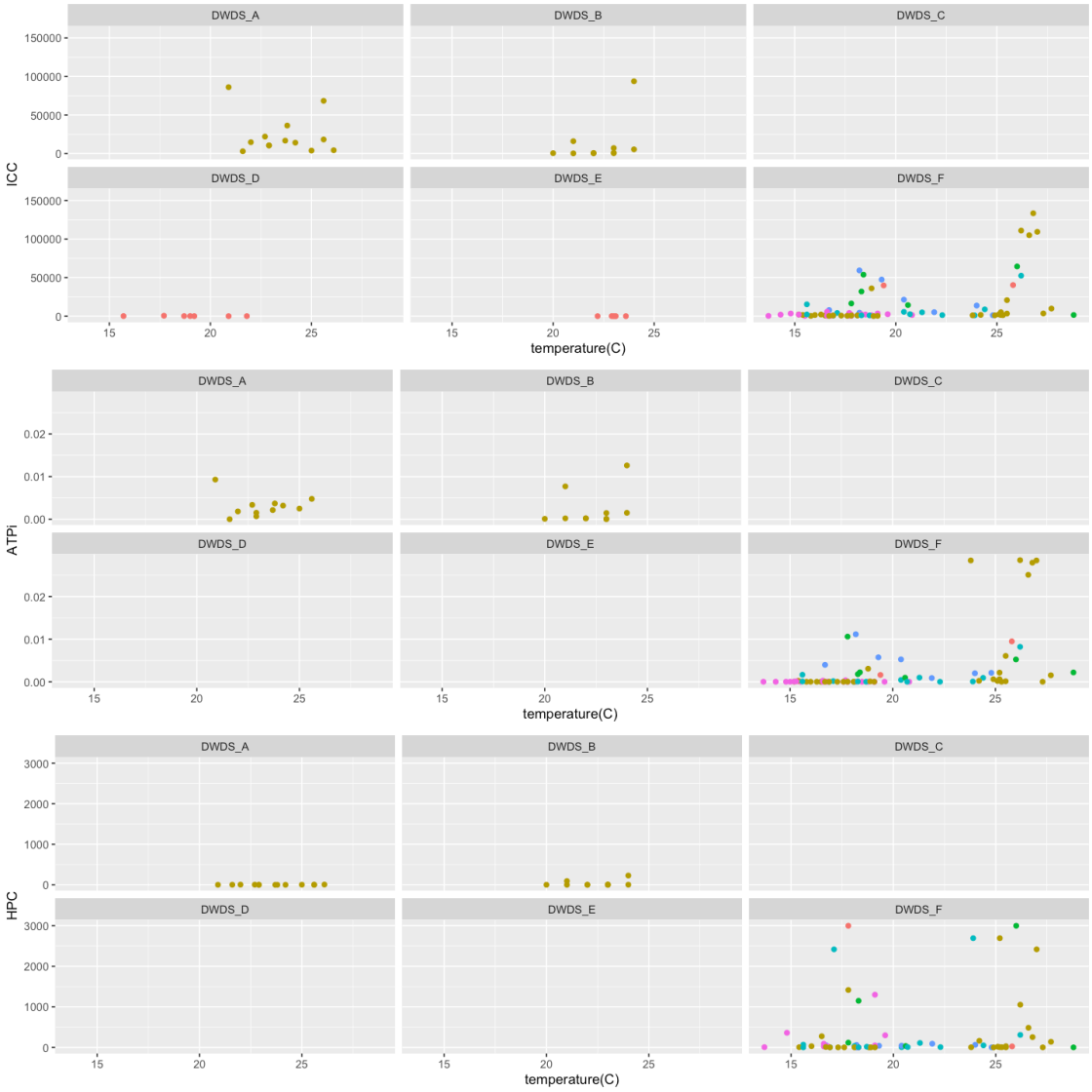
```
options(repr.plot.width = 12, repr.plot.height = 12) #for plotting size in␣
↪jupyter

aa<-ggplot(a, aes(y = SGPI, x = rw, color= count)) +
    plt+
    ylab("ICC") +
    xlabel +
    wrap
b<-ggplot(a, aes(y = intra_ATP_gmean_nM, x = rw, color= count)) +
    plt+
    ylab("ATPi") +
    xlabel +
    wrap
c<-ggplot(a, aes(y = HPC_gmean_MPN_per_100mL, x = rw, color= count)) +
    plt+
    ylab("HPC") +
    xlabel +
    wrap


ggarrange(aa,  b,c,
          ncol = 1, nrow = 3, legend="none")
```



no clear relationship

[29]:
```
a<-raw_dat
a$rw<-a$temp_C
raw_dat_num<-raw_dat[,c("SGPI", "intra_ATP_gmean_nM","HPC_gmean_MPN_per_100mL",␣
↪"temp_C","pH","total_Cl2_mg.L","free_Cl2_mg.L")]


xlabel<-xlab("temperature(C)")
wrap<-facet_wrap(~broad_location)
plt<-geom_point()
options(repr.plot.width = 12, repr.plot.height = 12) #for plotting size in␣
↪jupyter
```

```
aa<-ggplot(a, aes(y = SGPI, x = rw, color= count)) +
    plt+
    ylab("ICC") +
    xlabel +
    wrap
b<-ggplot(a, aes(y = intra_ATP_gmean_nM, x = rw, color= count)) +
    plt+
    ylab("ATPi") +
    xlabel +
    wrap
c<-ggplot(a, aes(y = HPC_gmean_MPN_per_100mL, x = rw, color= count)) +
    plt+
    ylab("HPC") +
    xlabel +
    wrap


ggarrange(aa,  b,c,
          ncol = 1, nrow = 3, legend="none")
```

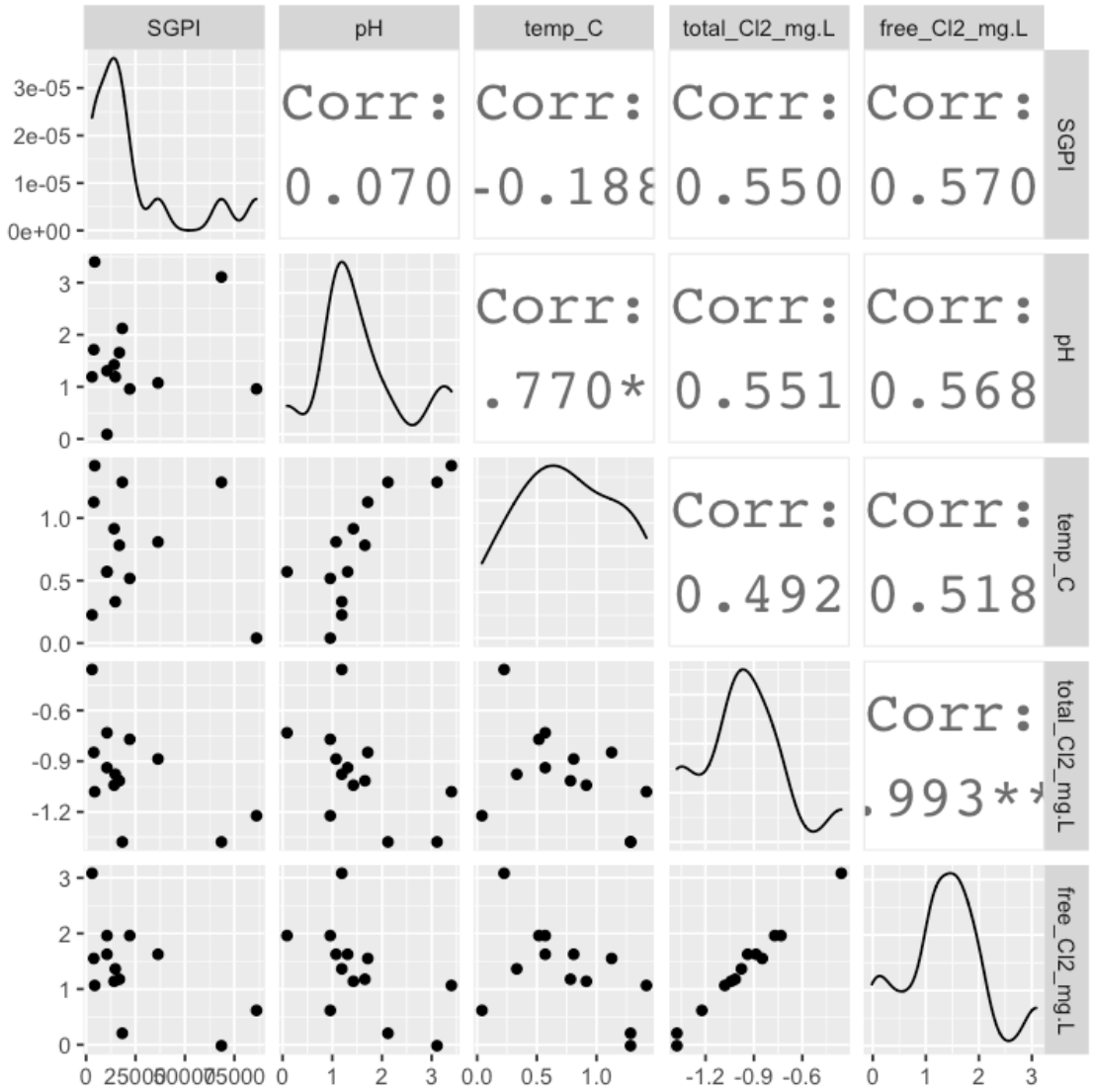


no clear relationship

Also checking for correlations between data from the same broad location and location code (dependency)

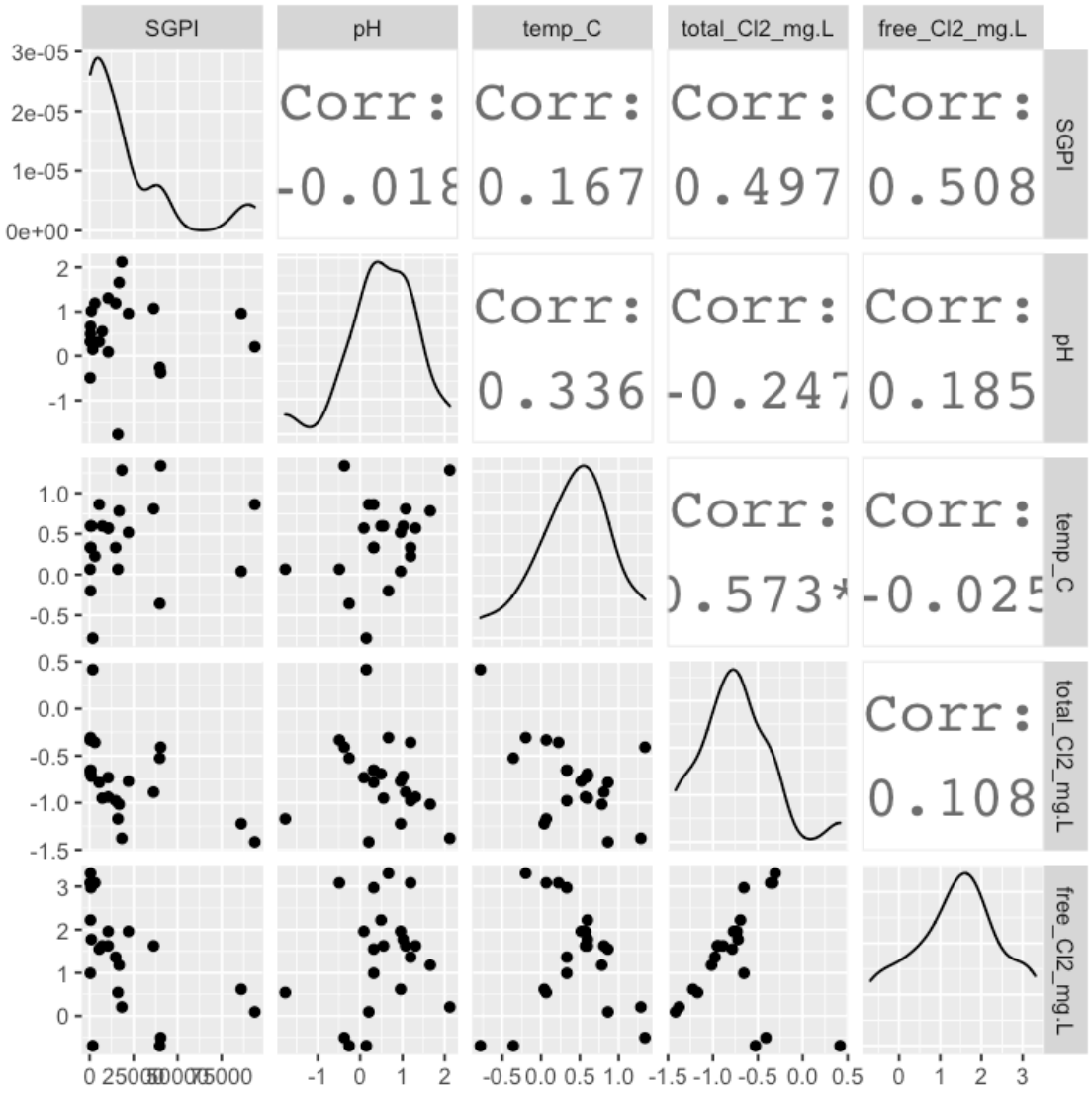
[30]:
```
options(repr.plot.width =6, repr.plot.height = 6) #for plotting size in jupyter
df<-raw_ICC

col<-c('SGPI','pH','temp_C','total_Cl2_mg.L','free_Cl2_mg.L')
a<-df[df$broad_location=="DWDS_A",]
a<-a[,col]

ggpairs(a, upper = list(continuous = wrap('cor', size = 8) ) )
```

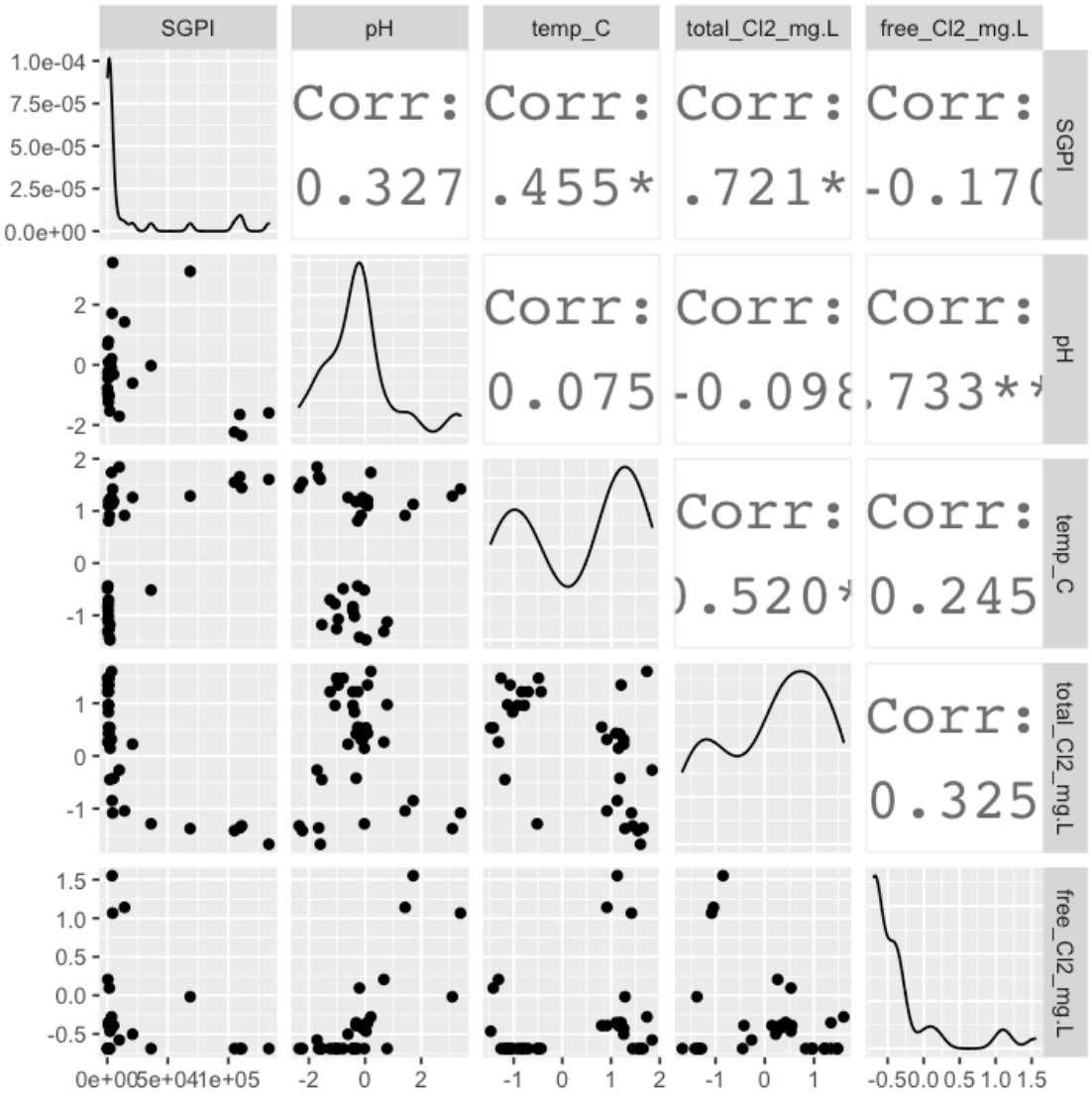
```
# b<-df[df$broad_location=="DWDS_B",]
# b<-b[,col]
#  ggpairs(b, upper = list(continuous = wrap('cor', size = 8) ) )

# f<-df[df$broad_location=="DWDS_F",]
# f<-f[,col]
#  ggpairs(f, upper = list(continuous = wrap('cor', size = 8) ) )

# g<-df[df$location_code=="site_15",]
# g<-g[,col]
#  ggpairs(g, upper = list(continuous = wrap('cor', size = 8) ) )

# h<-df[df$location_code=="site_10",]
# h<-h[,col]
#  ggpairs(h, upper = list(continuous = wrap('cor', size = 8) ) )

# i<-df[df$location_code=="site_24",]
# i<-i[,col]
#  ggpairs(i, upper = list(continuous = wrap('cor', size = 8) ) )

# j<-df[df$location_code=="site_ut",]
# j<-j[,col]
#  ggpairs(j, upper = list(continuous = wrap('cor', size = 8) ) )
```

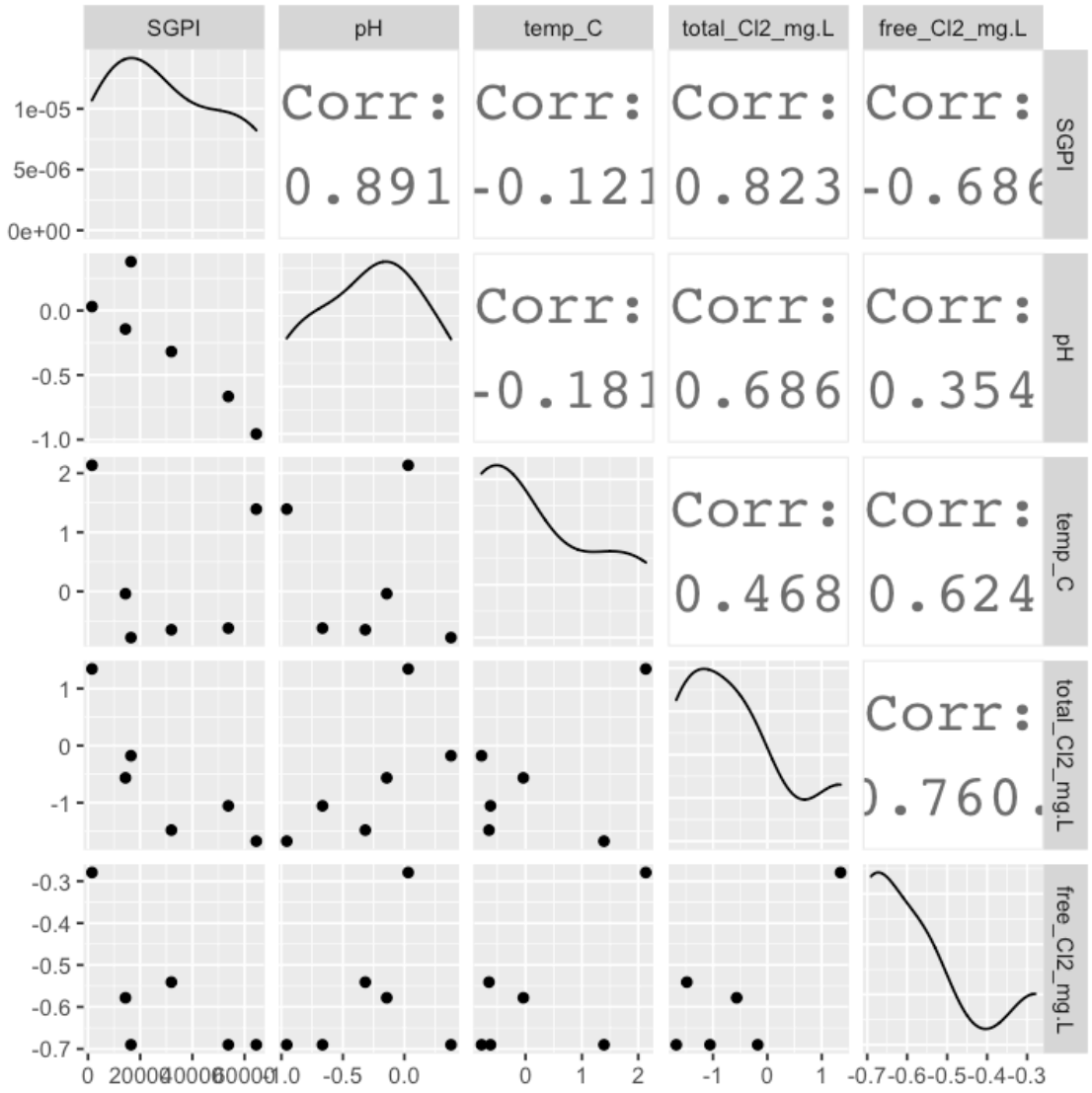


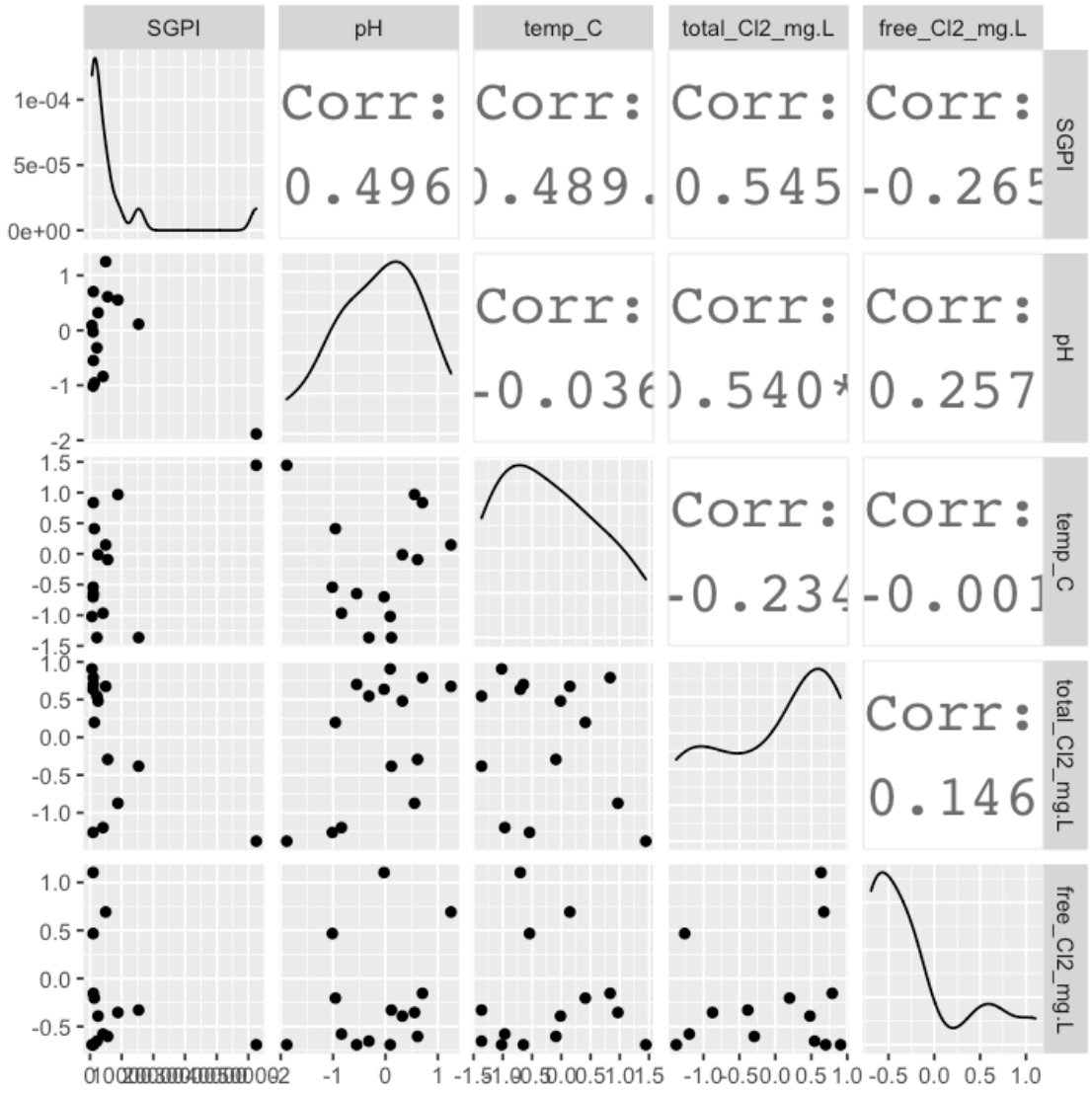
[31]:
```
g<-df[df$count== 1,]
g<-g[,col]
 ggpairs(g, upper = list(continuous = wrap('cor', size = 8) ) )

h<-df[df$count==2,]
h<-h[,col]
 ggpairs(h, upper = list(continuous = wrap('cor', size = 8) ) )

i<-df[df$count==3,]
i<-i[,col]
 ggpairs(i, upper = list(continuous = wrap('cor', size = 8) ) )

j<-df[df$count==7,]
```

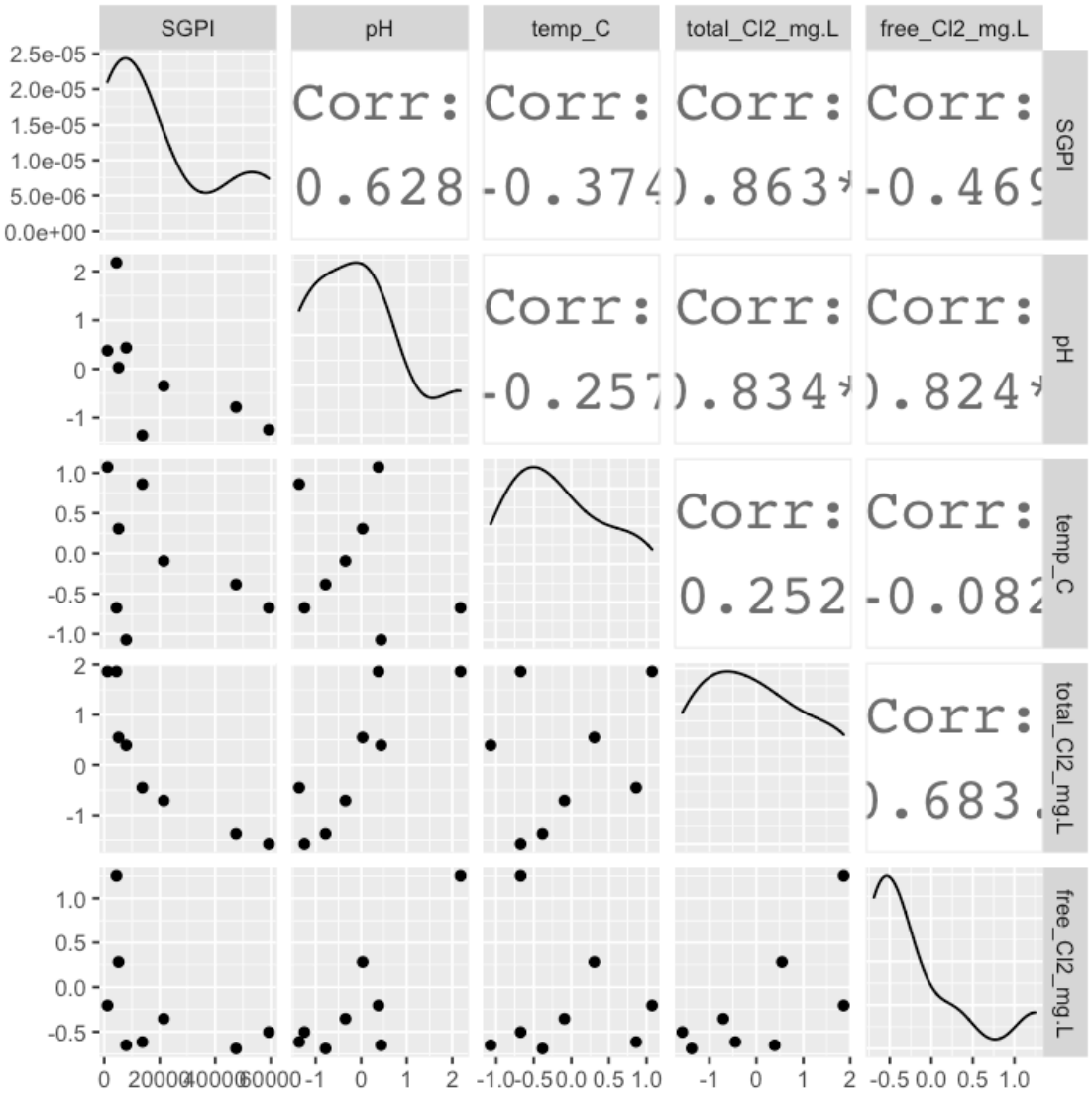
```
j<-j[,col]
 ggpairs(j, upper = list(continuous = wrap('cor', size = 8) ) )

k<-df[df$count==8,] #includes 2 sites but still correlated
k<-k[,col]
 ggpairs(k, upper = list(continuous = wrap('cor', size = 8) ) )

l<-df[df$count==19,]
l<-l[,col]
 ggpairs(l, upper = list(continuous = wrap('cor', size = 8) ) )
```

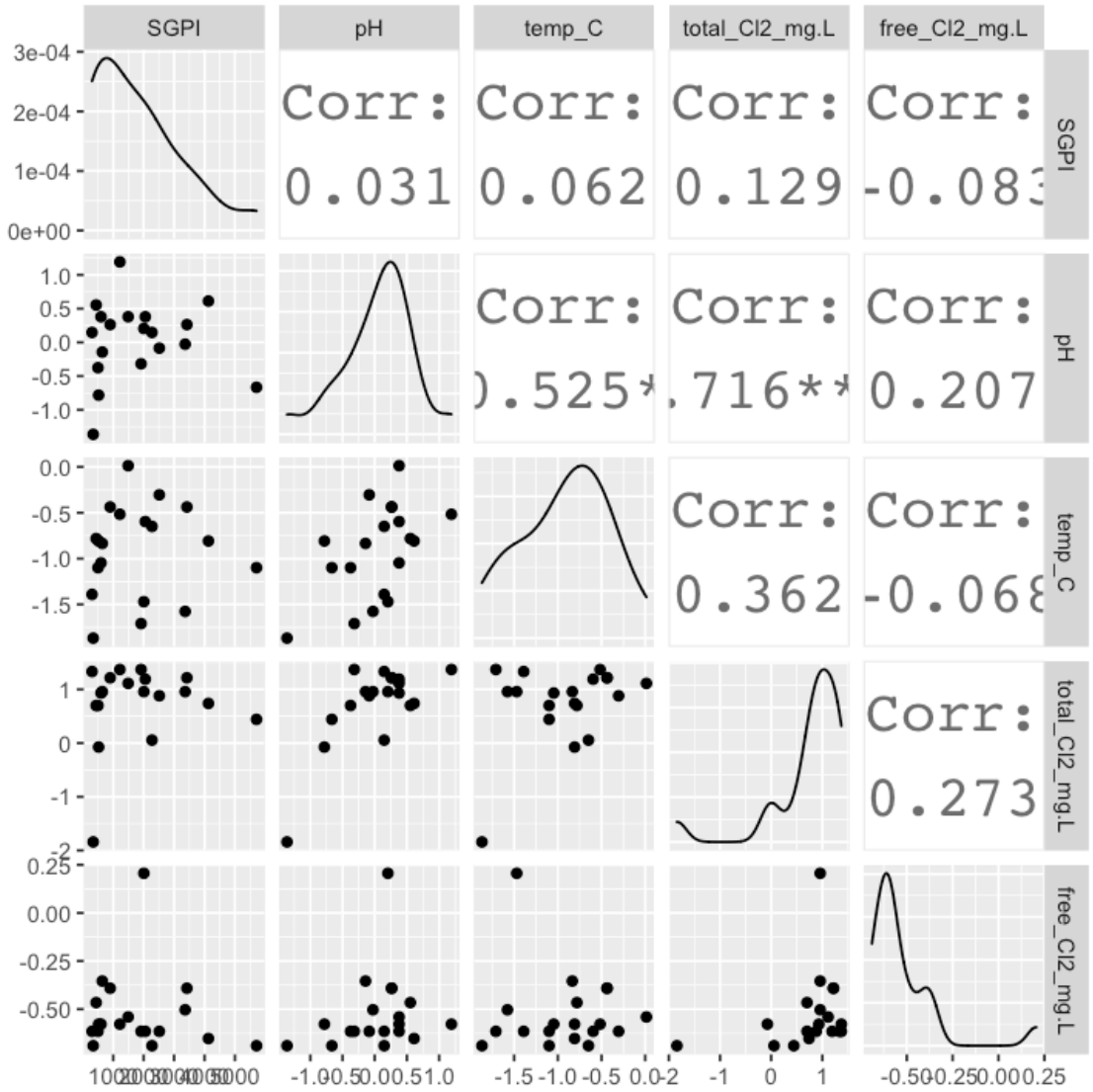



SGPI
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr: 0.327
Corr: .455*
Corr: .721*
Corr: -0.17
Corr: 0.075
Corr: -0.09
Corr: .733**
Corr: .520*
Corr: 0.245
Corr: 0.325
1.0e-04
7.5e-05
5.0e-05
2.5e-05
0.0e+00
0e+00
5e+04
1e+05
-2
0
2
-1
1
-0.5
0.0
0.5
1.0
1.5


SGPI
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
0.891
Corr:
-0.121
Corr:
0.823
Corr:
-0.686
Corr:
-0.181
Corr:
0.686
Corr:
0.354
Corr:
0.468
Corr:
0.624
Corr:
0.760
1e-05
5e-06
0e+00
0.0
-0.5
-1.0
2
1
0
1
0
-1
-0.3
-0.4
-0.5
-0.6
-0.7
0
20000
40000
60000
1.0
-0.5
0.0
0
1
2
-1
0
1
-0.7
-0.6
-0.5
-0.4
-0.3


SGPI
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
0.496
Corr:
).489.
Corr:
0.545
Corr:
-0.265
Corr:
-0.036
Corr:
).540*
Corr:
0.257
Corr:
-0.234
Corr:
-0.001
Corr:
0.146


SGPI
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
0.628
-0.374
0.863*
-0.469
Corr:
-0.257
0.834*
0.824*
Corr:
0.252
-0.082
Corr:
0.683*



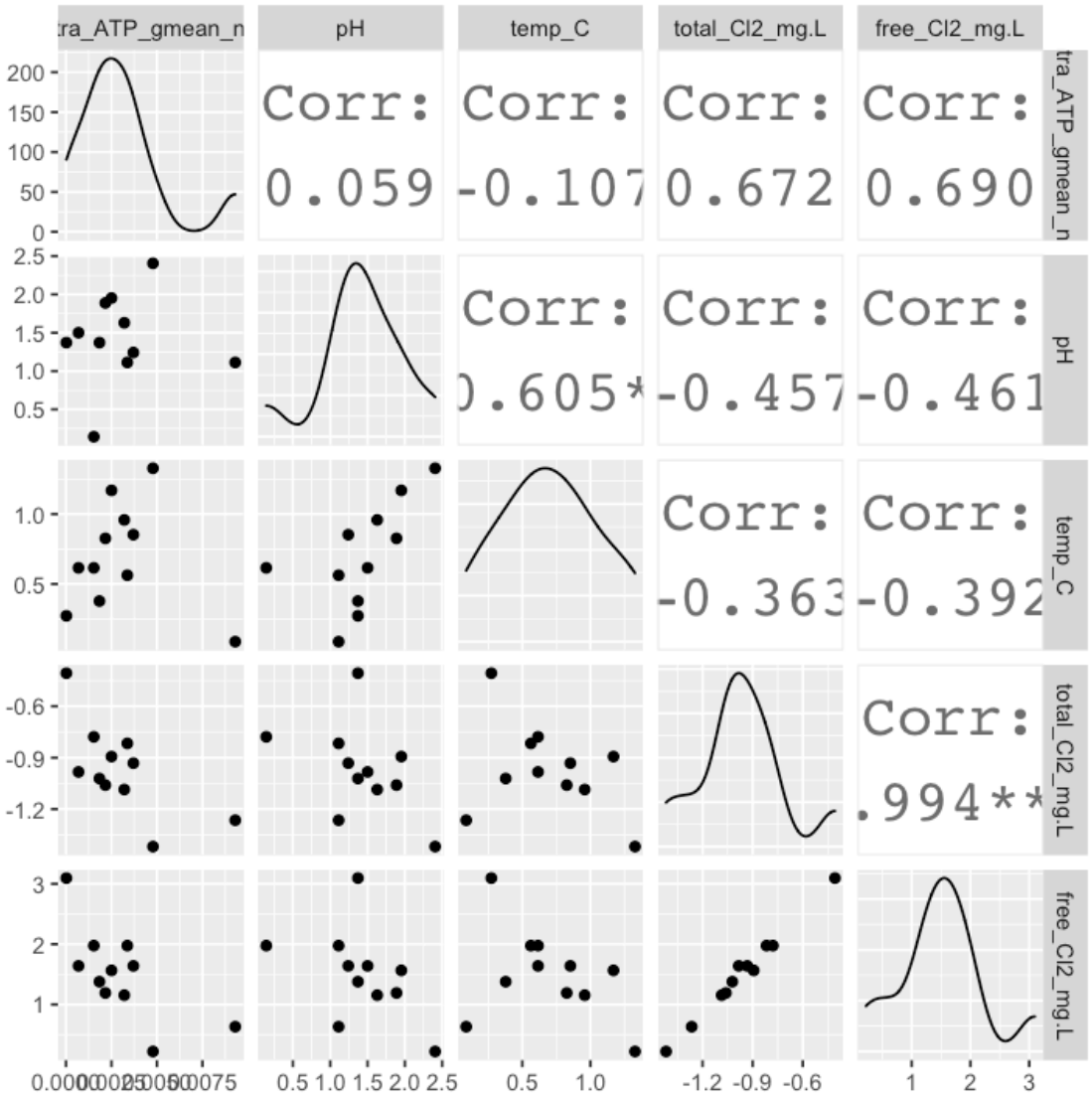
[32]:
```
unique(raw_ATPi$count)
```

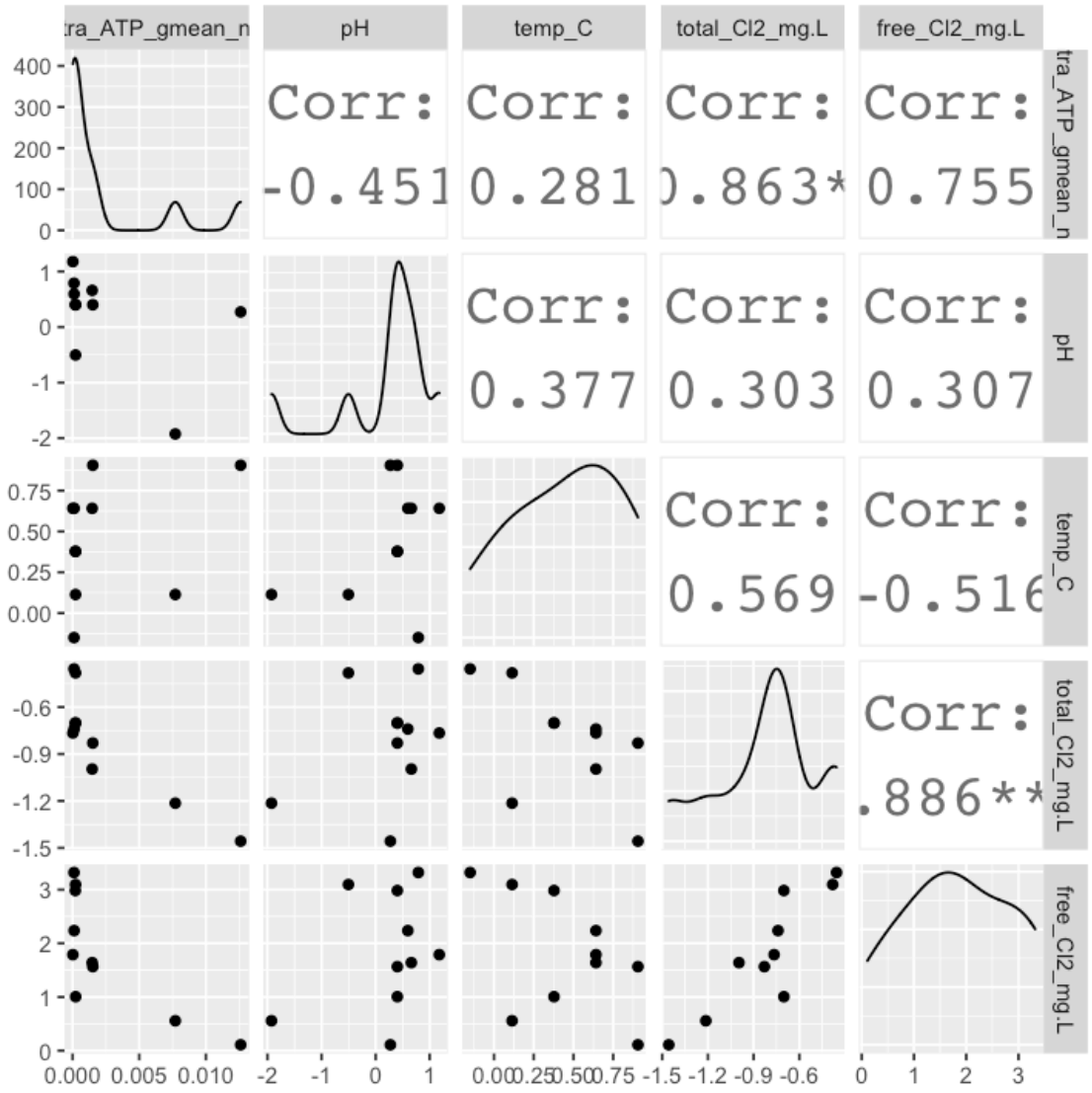
1. 1 2. 2 3. 3 4. 8 5. 7 6. 20

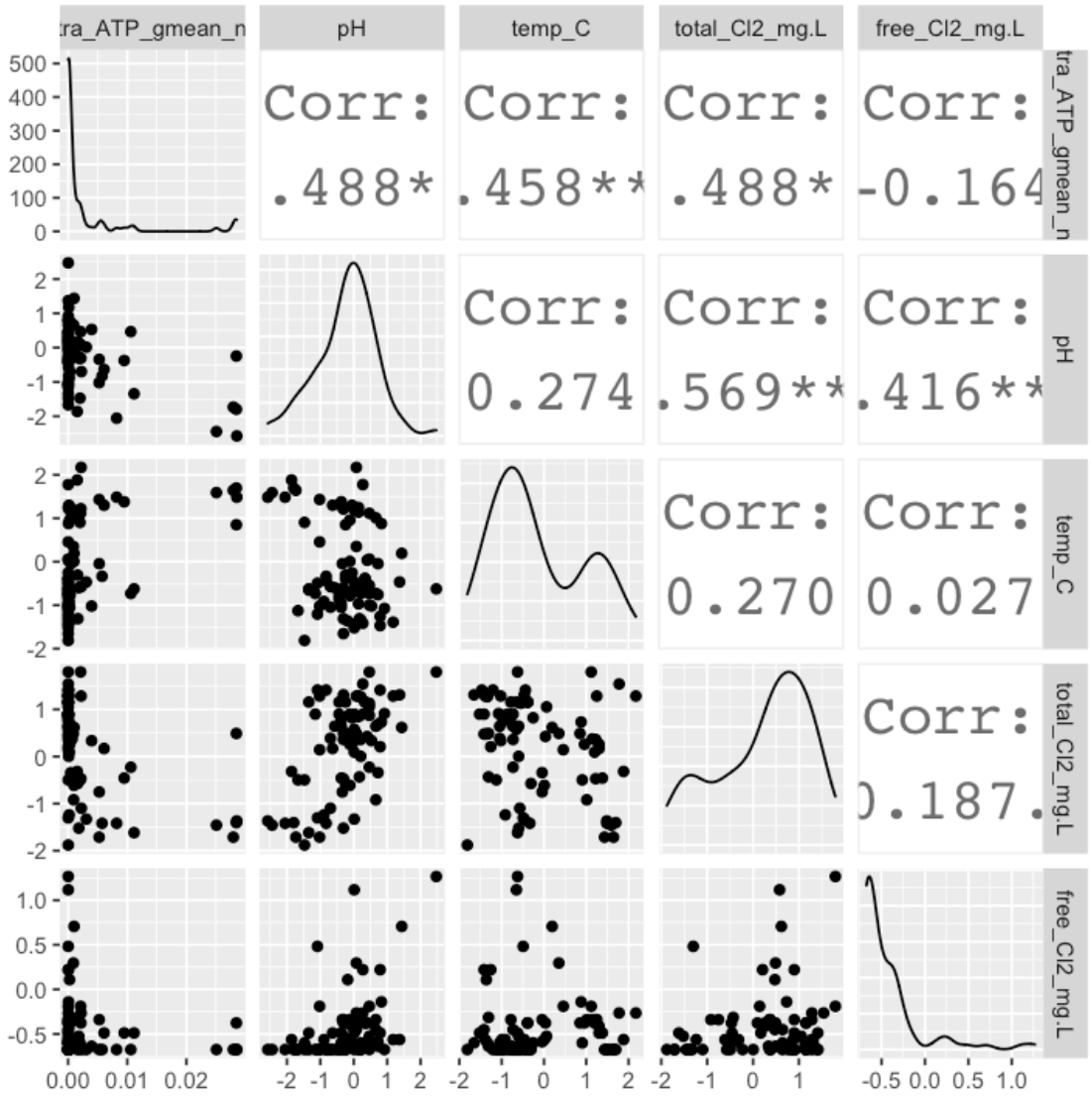
Levels: 1. '1' 2. '2' 3. '3' 4. '7' 5. '8' 6. '20'

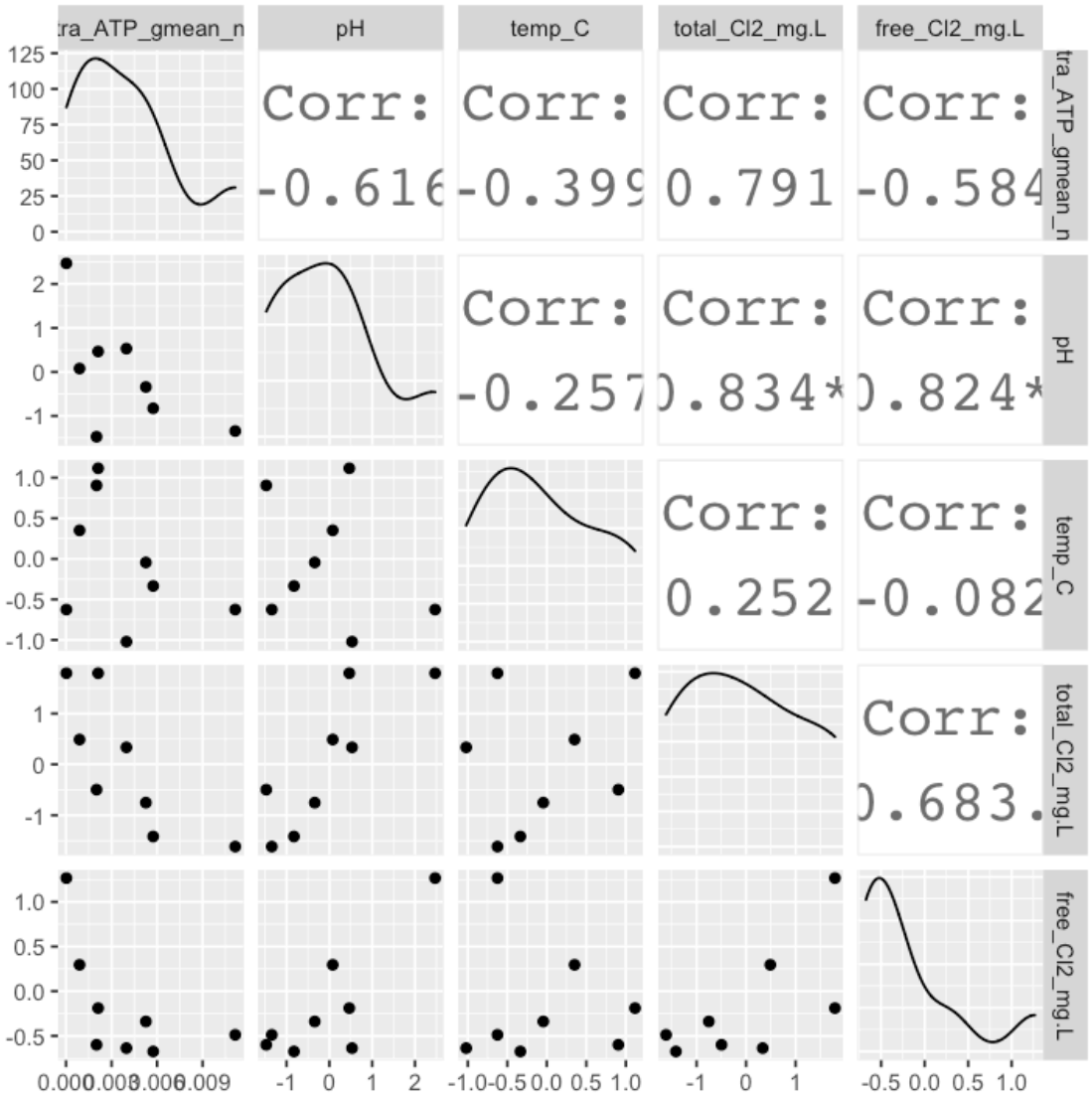
[33]:
```
options(repr.plot.width =6, repr.plot.height = 6) #for plotting size in jupyter
df<-raw_ATPi
col<-c('intra_ATP_gmean_nM','pH','temp_C','total_Cl2_mg.L','free_Cl2_mg.L')
a<-df[df$broad_location=="DWDS_A",]
a<-a[,col]
ggpairs(a, upper = list(continuous = wrap('cor', size = 8) ) )

b<-df[df$broad_location=="DWDS_B",]
```

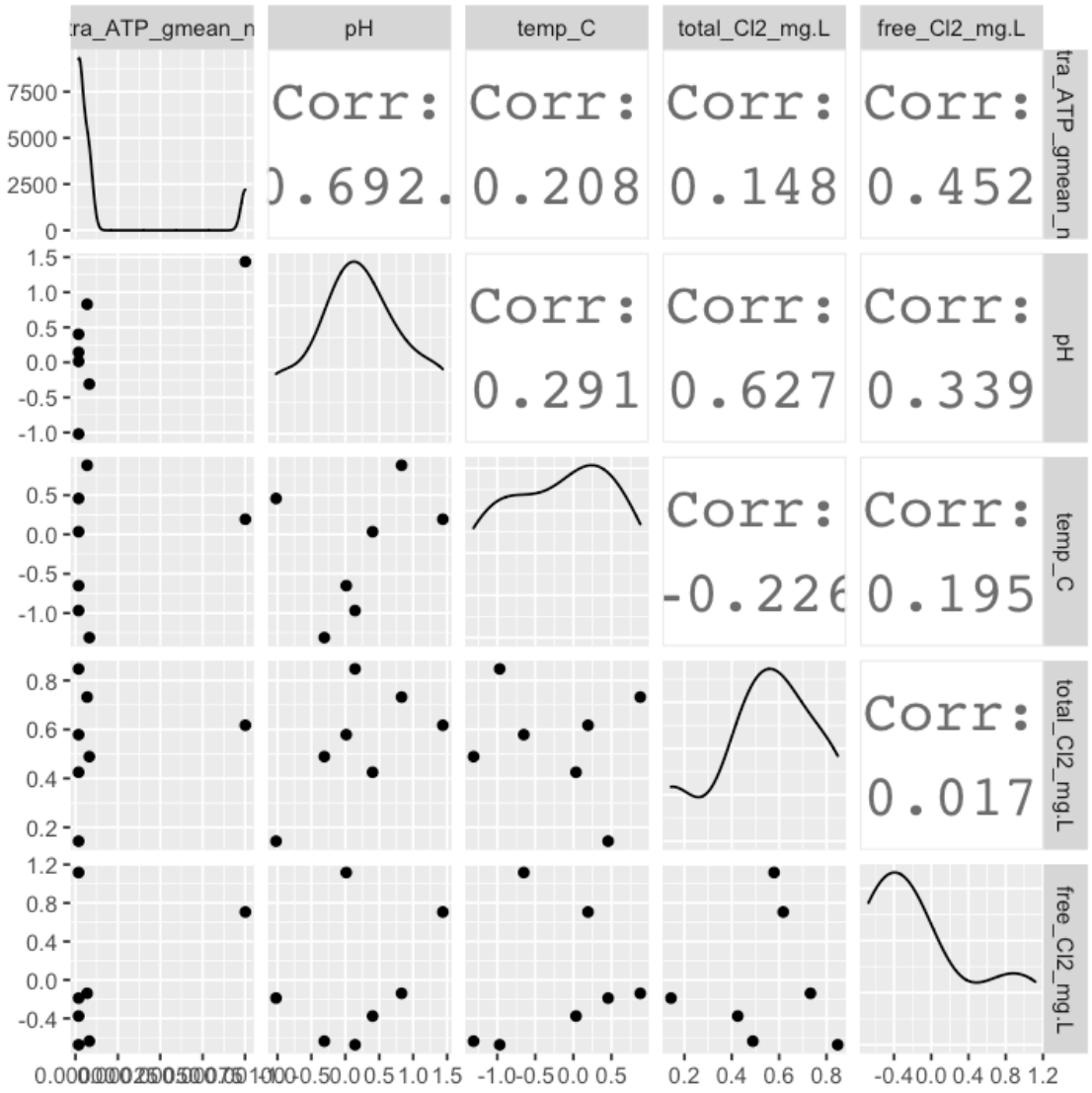
```
b<-b[,col]
 ggpairs(b, upper = list(continuous = wrap('cor', size = 8) ) )

f<-df[df$broad_location=="DWDS_F",]
f<-f[,col]
 ggpairs(f, upper = list(continuous = wrap('cor', size = 8) ) )

g<-df[df$location_code=="site_15",]
g<-g[,col]
 ggpairs(g, upper = list(continuous = wrap('cor', size = 8) ) )

h<-df[df$location_code=="site_10",]
h<-h[,col]
 ggpairs(h, upper = list(continuous = wrap('cor', size = 8) ) )

i<-df[df$location_code=="site_24",]
i<-i[,col]
 ggpairs(i, upper = list(continuous = wrap('cor', size = 8) ) )

j<-df[df$location_code=="site_ut",]
j<-j[,col]
 ggpairs(j, upper = list(continuous = wrap('cor', size = 8) ) )
```

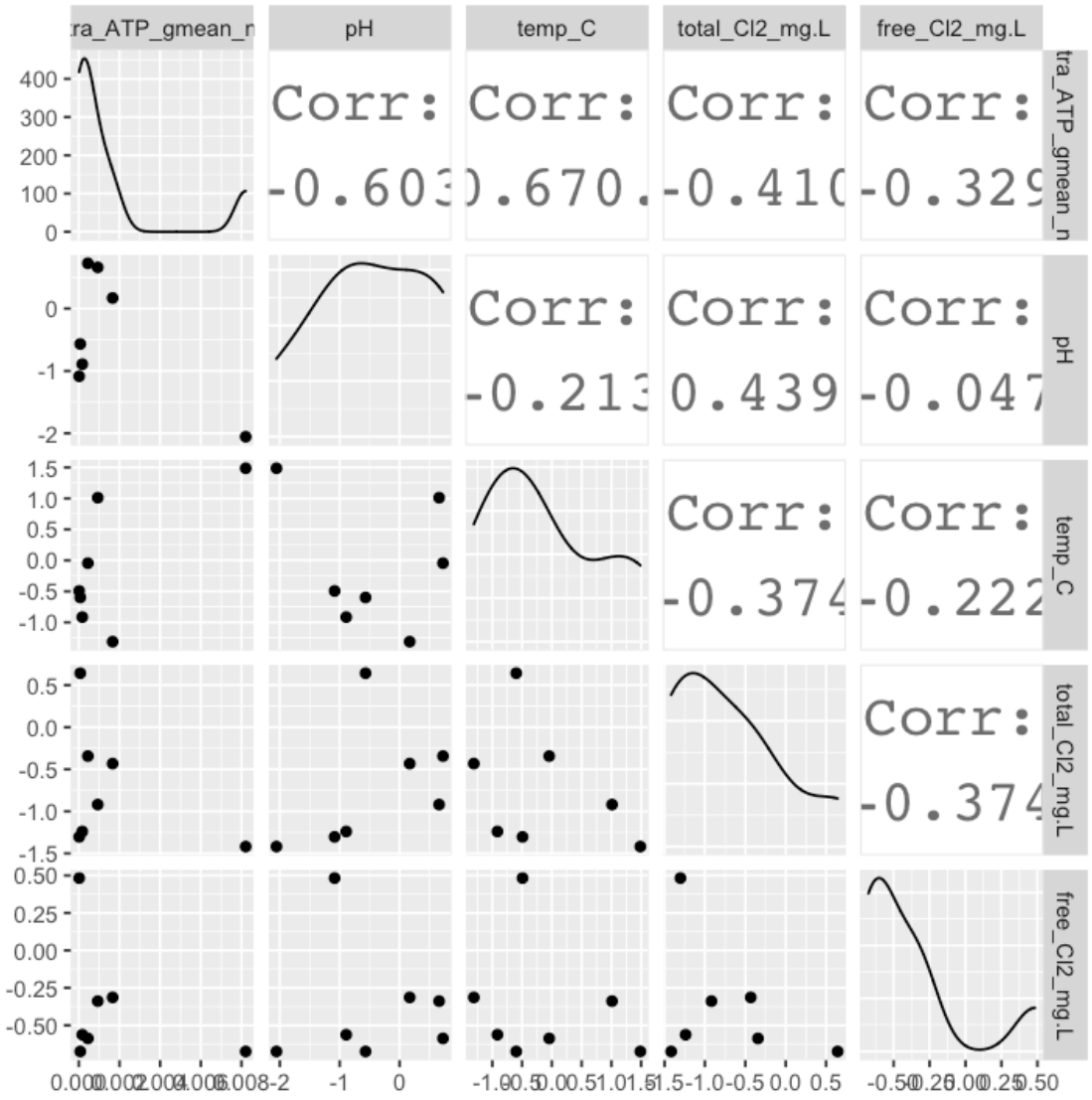

ra_ATP_gmean_n
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
0.059
Corr:
-0.107
Corr:
0.672
Corr:
0.690
Corr:
.605*
Corr:
-0.457
Corr:
-0.461
Corr:
-0.363
Corr:
-0.392
Corr:
.994**

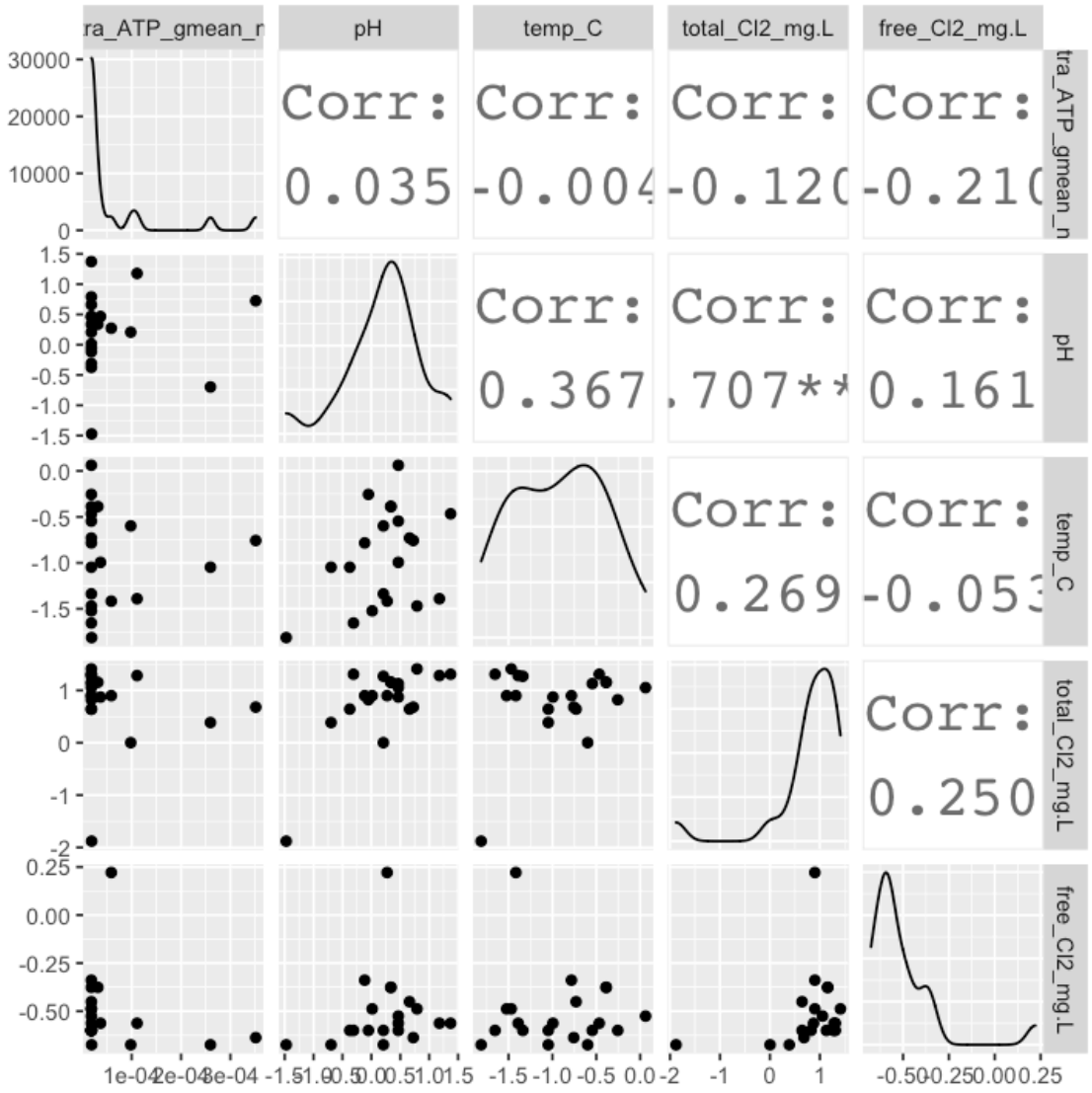

ra_ATP_gmean_n
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
-0.451
Corr:
0.281
Corr:
0.863*
Corr:
0.755
Corr:
0.377
Corr:
0.303
Corr:
0.307
Corr:
0.569
Corr:
-0.516
Corr:
.886**
0.000 0.005 0.010
-2 -1 0 1
0.00 0.25 0.50 0.75
-1.5 -1.2 -0.9 -0.6
0 1 2 3


ra_ATP_gmean_n
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
.488*
.458**
.488*
-0.164
Corr:
0.274
.569**
.416**
Corr:
0.270
Corr:
0.027
Corr:
0.187.


ra_ATP_gmean_n
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
-0.616
Corr:
-0.399
Corr:
0.791
Corr:
-0.584
Corr:
-0.257
Corr:
0.834*
Corr:
0.824*
Corr:
0.252
Corr:
-0.082
Corr:
0.683.

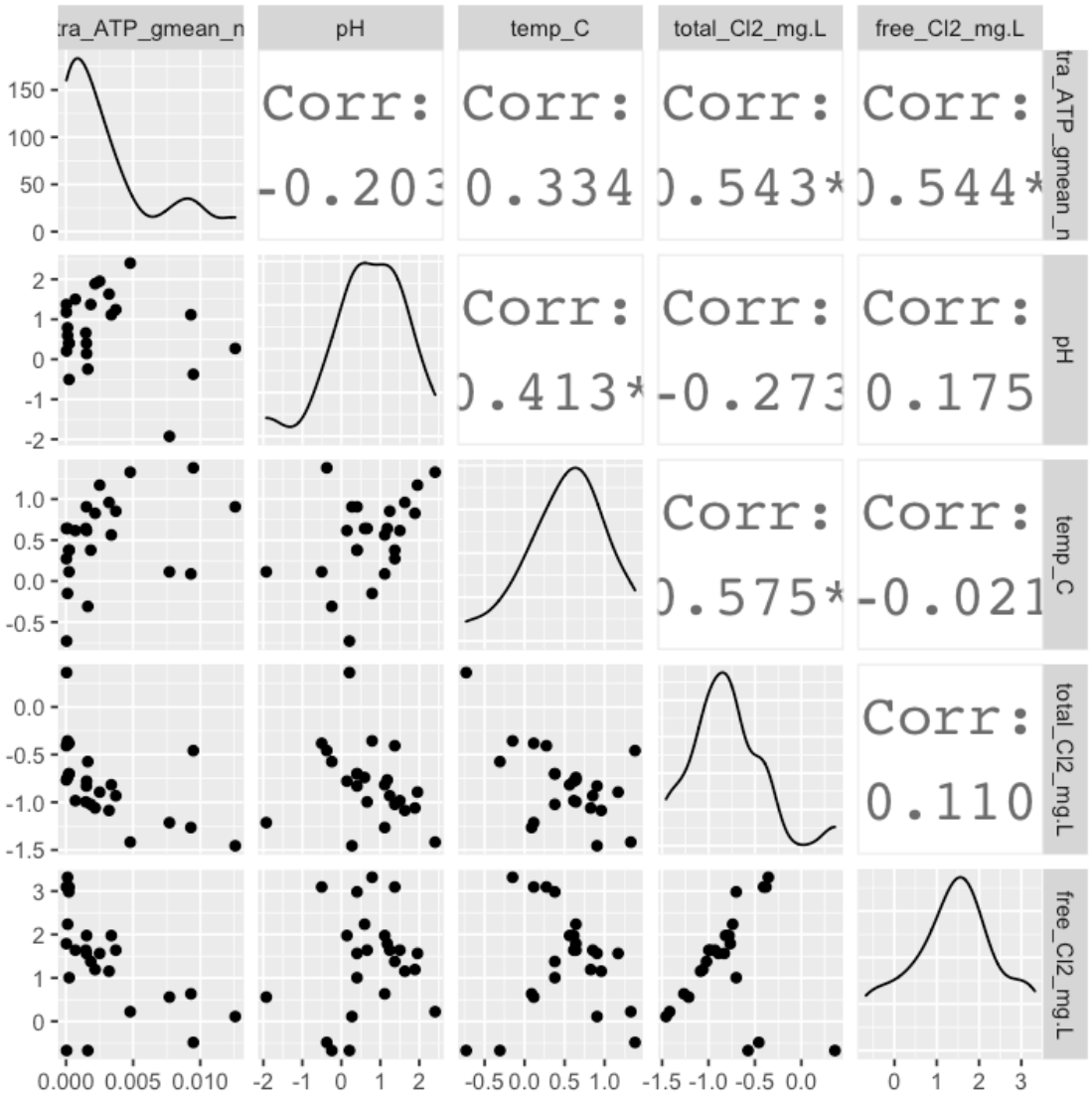

ra_ATP_gmean_n
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
.692.
0.208
0.148
0.452
Corr:
0.291
0.627
0.339
Corr:
-0.226
0.195
Corr:
0.017

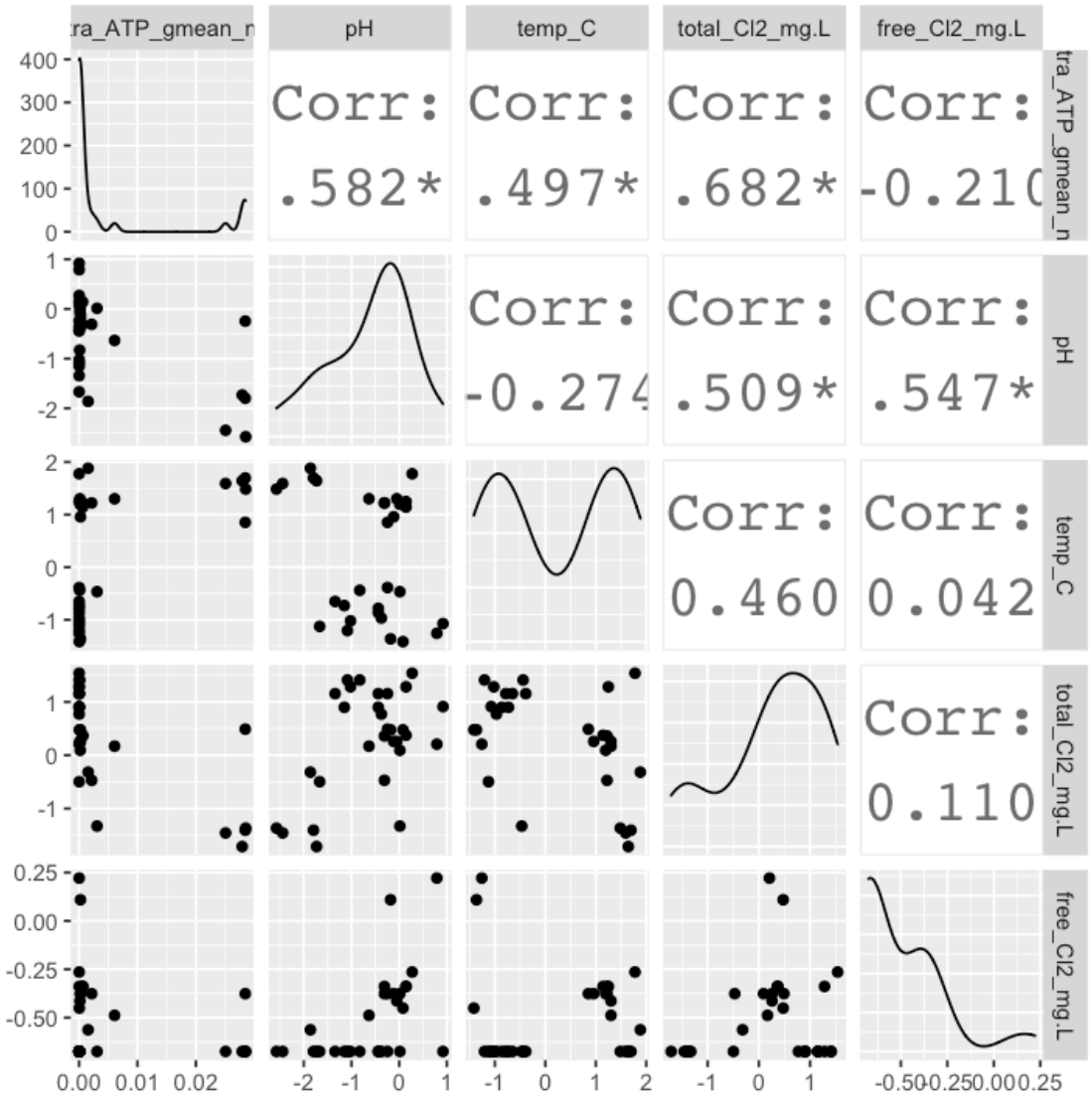

ra_ATP_gmean_n
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
-0.603
Corr:
0.670
Corr:
-0.410
Corr:
-0.329
Corr:
-0.213
Corr:
0.439
Corr:
-0.047
Corr:
-0.374
Corr:
-0.222
Corr:
-0.374

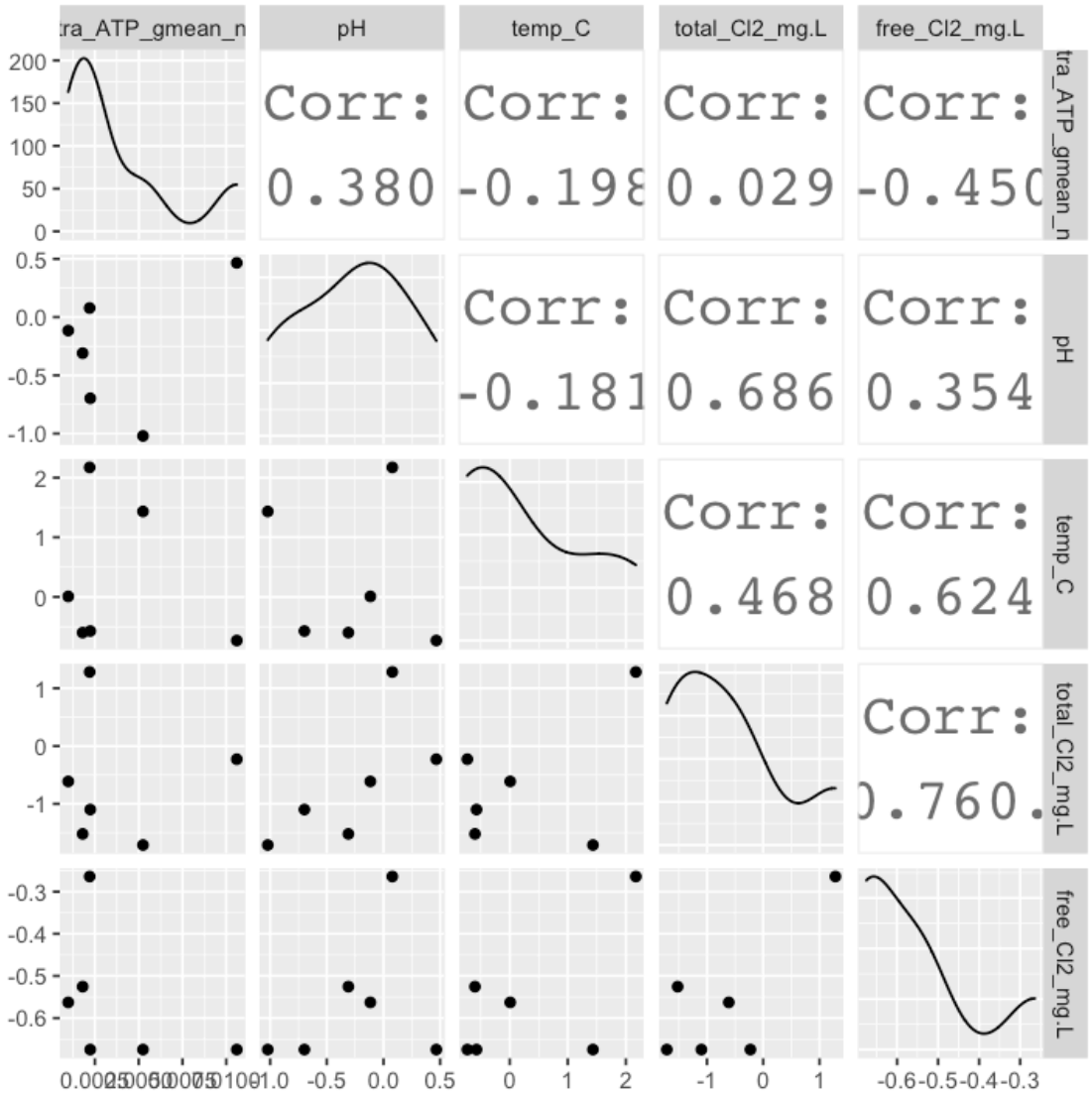


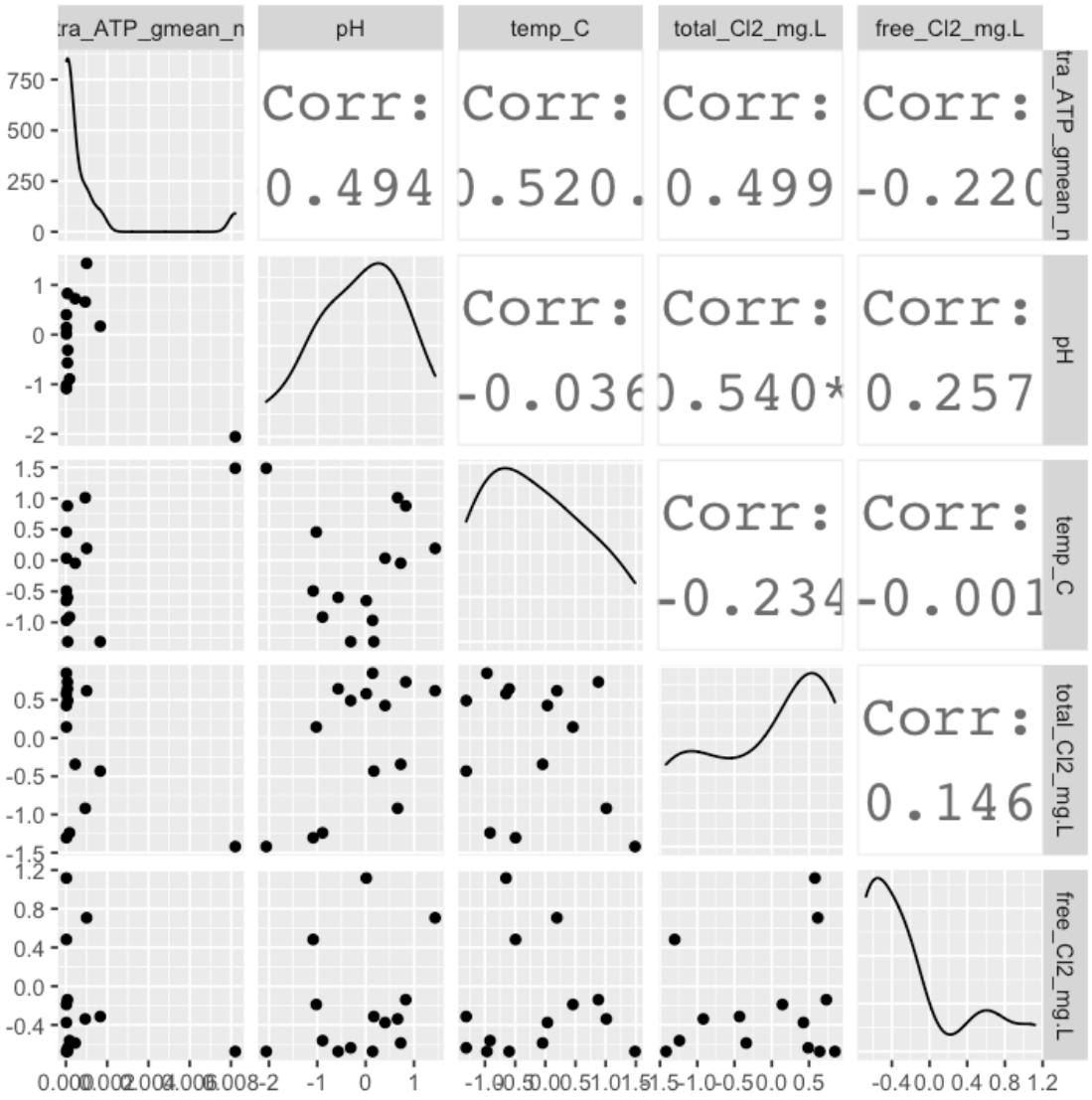
[34]:
```
g<-df[df$count== 1,]
g<-g[,col]
 ggpairs(g, upper = list(continuous = wrap('cor', size = 8) ) )

h<-df[df$count==2,]
h<-h[,col]
 ggpairs(h, upper = list(continuous = wrap('cor', size = 8) ) )

i<-df[df$count==3,]
i<-i[,col]
 ggpairs(i, upper = list(continuous = wrap('cor', size = 8) ) )

j<-df[df$count==7,]
```

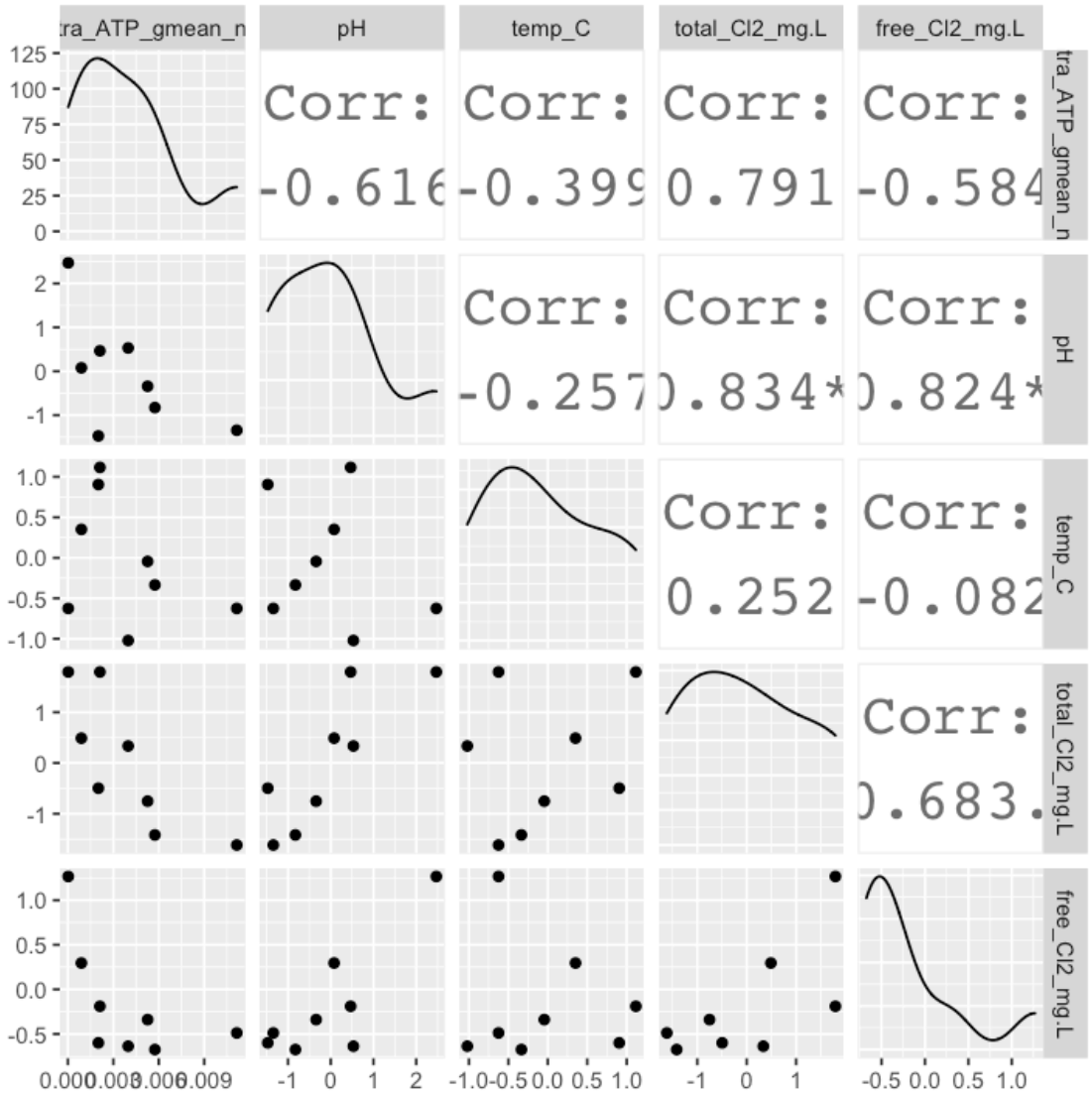
```
j<-j[,col]
 ggpairs(j, upper = list(continuous = wrap('cor', size = 8) ) )

k<-df[df$count==8,] #includes 2 sites but still correlated
k<-k[,col]
 ggpairs(k, upper = list(continuous = wrap('cor', size = 8) ) )

# l<-df[df$count==19,]
# l<-l[,col]
#  ggpairs(l, upper = list(continuous = wrap('cor', size = 8) ) )
```




ra_ATP_gmean_n
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
.582*
Corr:
.497*
Corr:
.682*
Corr:
-0.210
Corr:
-0.274
Corr:
.509*
Corr:
.547*
Corr:
0.460
Corr:
0.042
Corr:
0.110

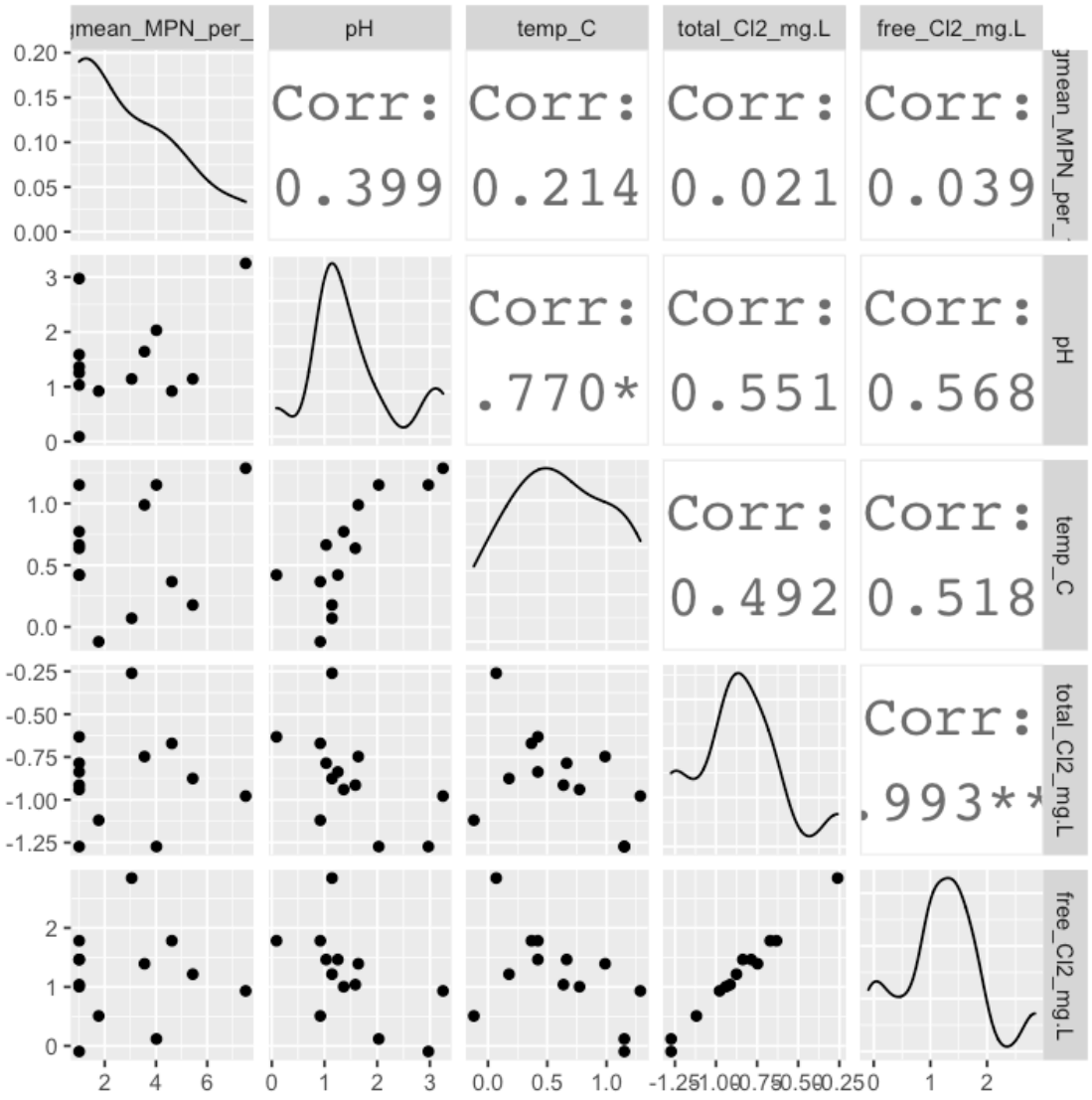

ra_ATP_gmean_n
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
0.380
Corr:
-0.198
Corr:
0.029
Corr:
-0.450
Corr:
-0.181
Corr:
0.686
Corr:
0.354
Corr:
0.468
Corr:
0.624
Corr:
0.760.

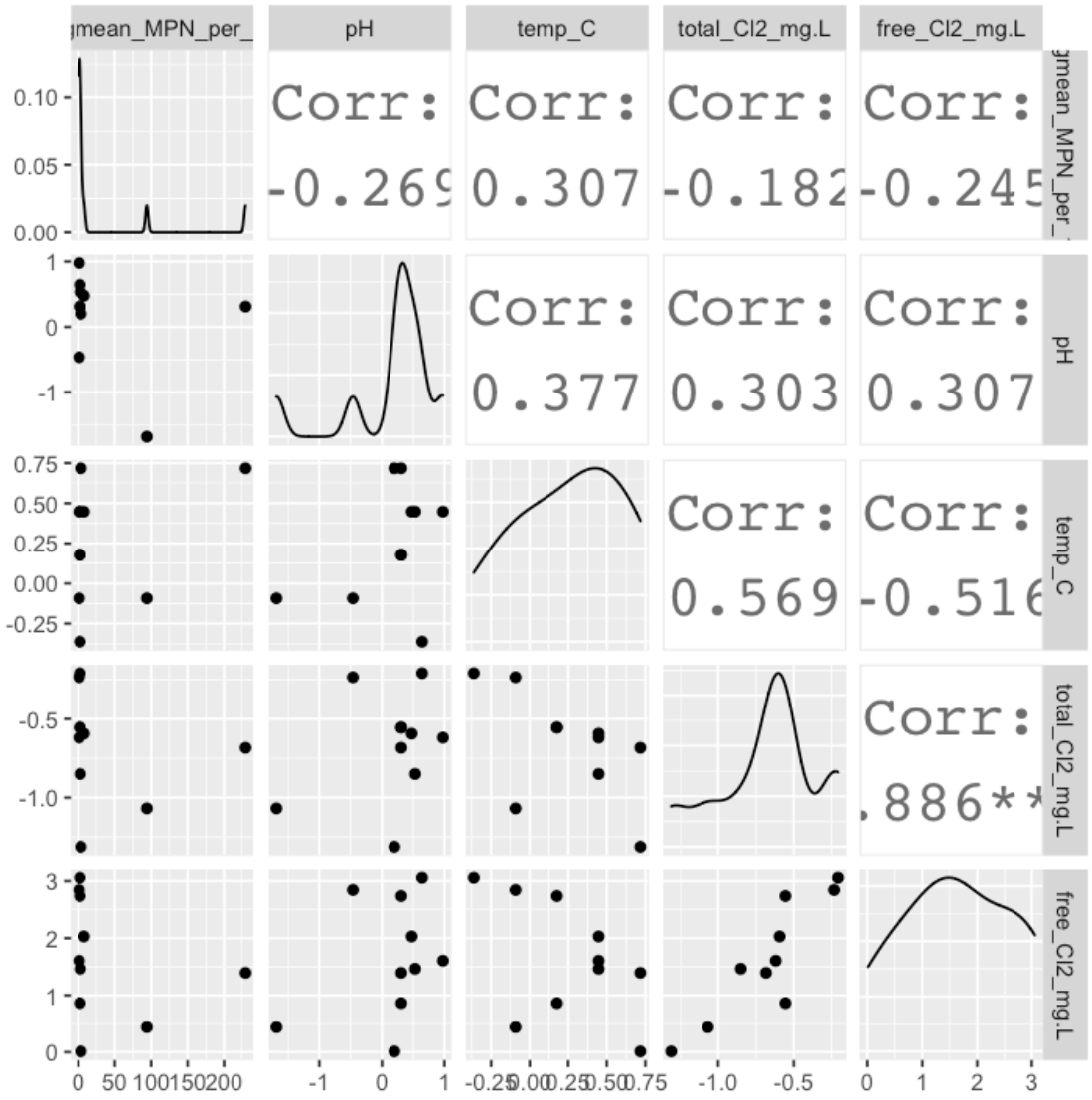

ra_ATP_gmean_n
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
0.494
Corr:
.520.
Corr:
0.499
Corr:
-0.220
Corr:
-0.036
Corr:
.540*
Corr:
0.257
Corr:
-0.234
Corr:
-0.001
Corr:
0.146

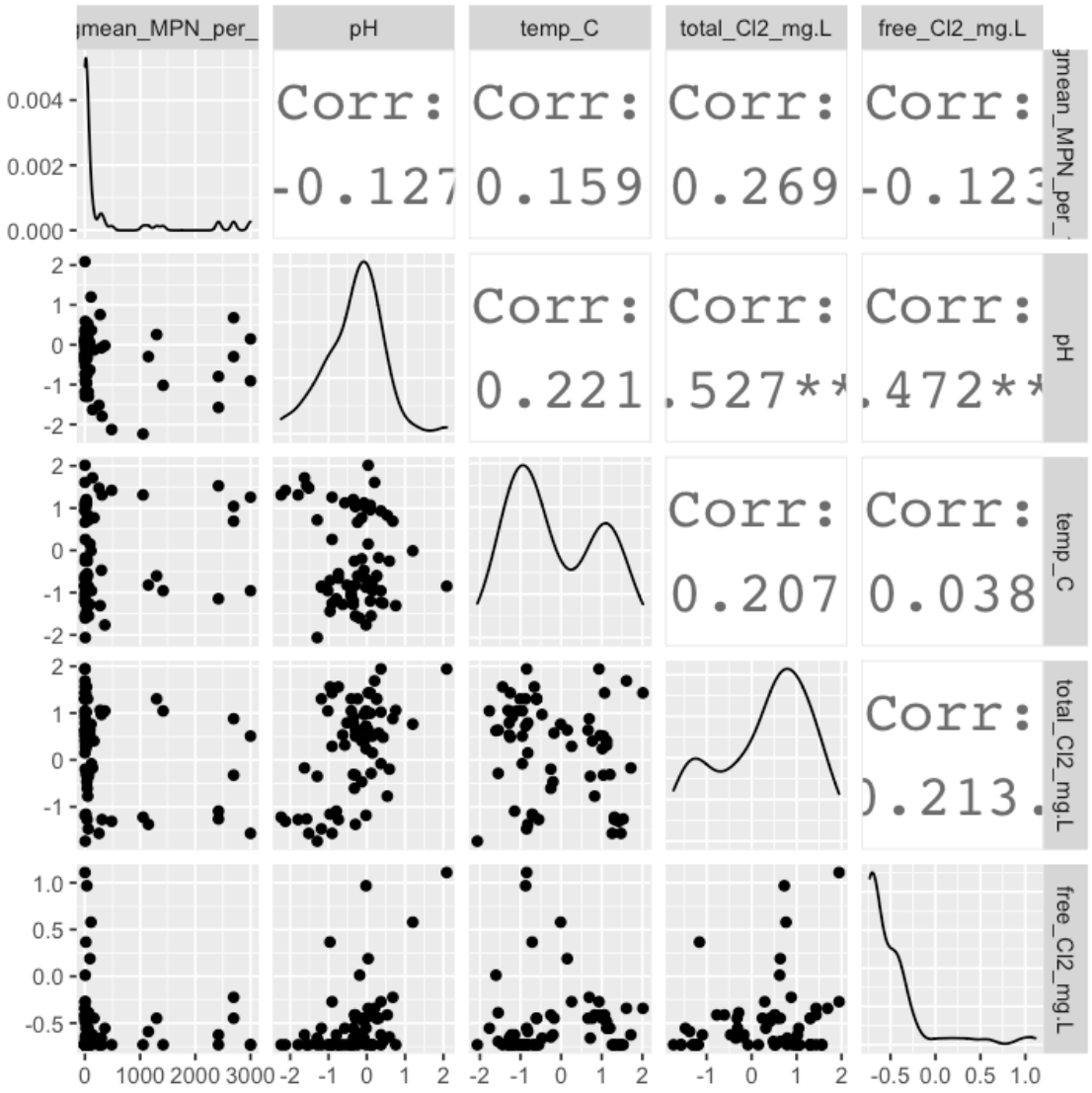


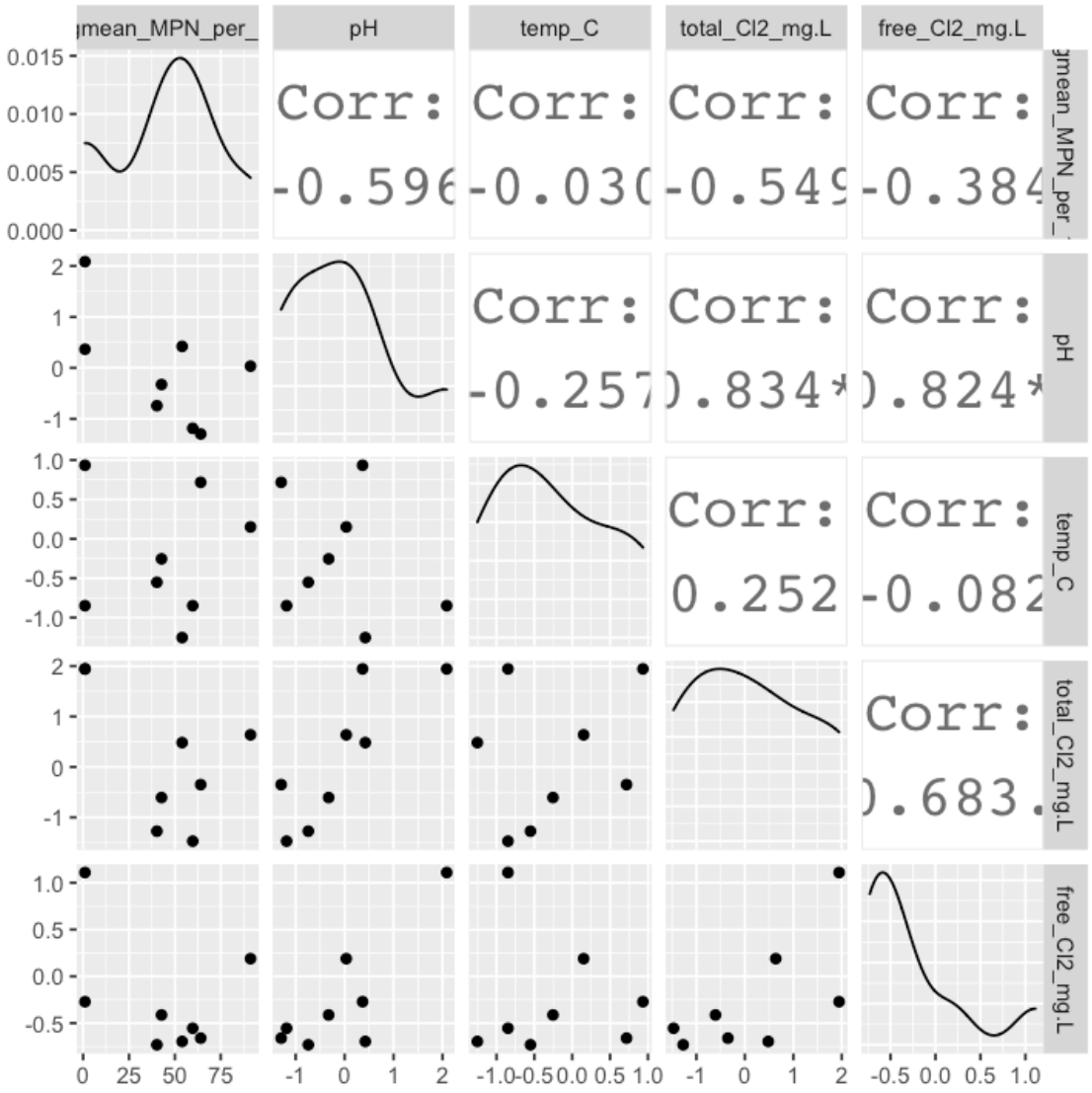
[35]:
```
options(repr.plot.width =6, repr.plot.height = 6) #for plotting size in jupyter
df<-raw_HPC
col<-c('HPC_gmean_MPN_per_100mL','pH','temp_C','total_Cl2_mg.L','free_Cl2_mg.L')
a<-df[df$broad_location=="DWDS_A",]
a<-a[,col]
ggpairs(a, upper = list(continuous = wrap('cor', size = 8) ) )

b<-df[df$broad_location=="DWDS_B",]
b<-b[,col]
 ggpairs(b, upper = list(continuous = wrap('cor', size = 8) ) )

f<-df[df$broad_location=="DWDS_F",]
f<-f[,col]
```

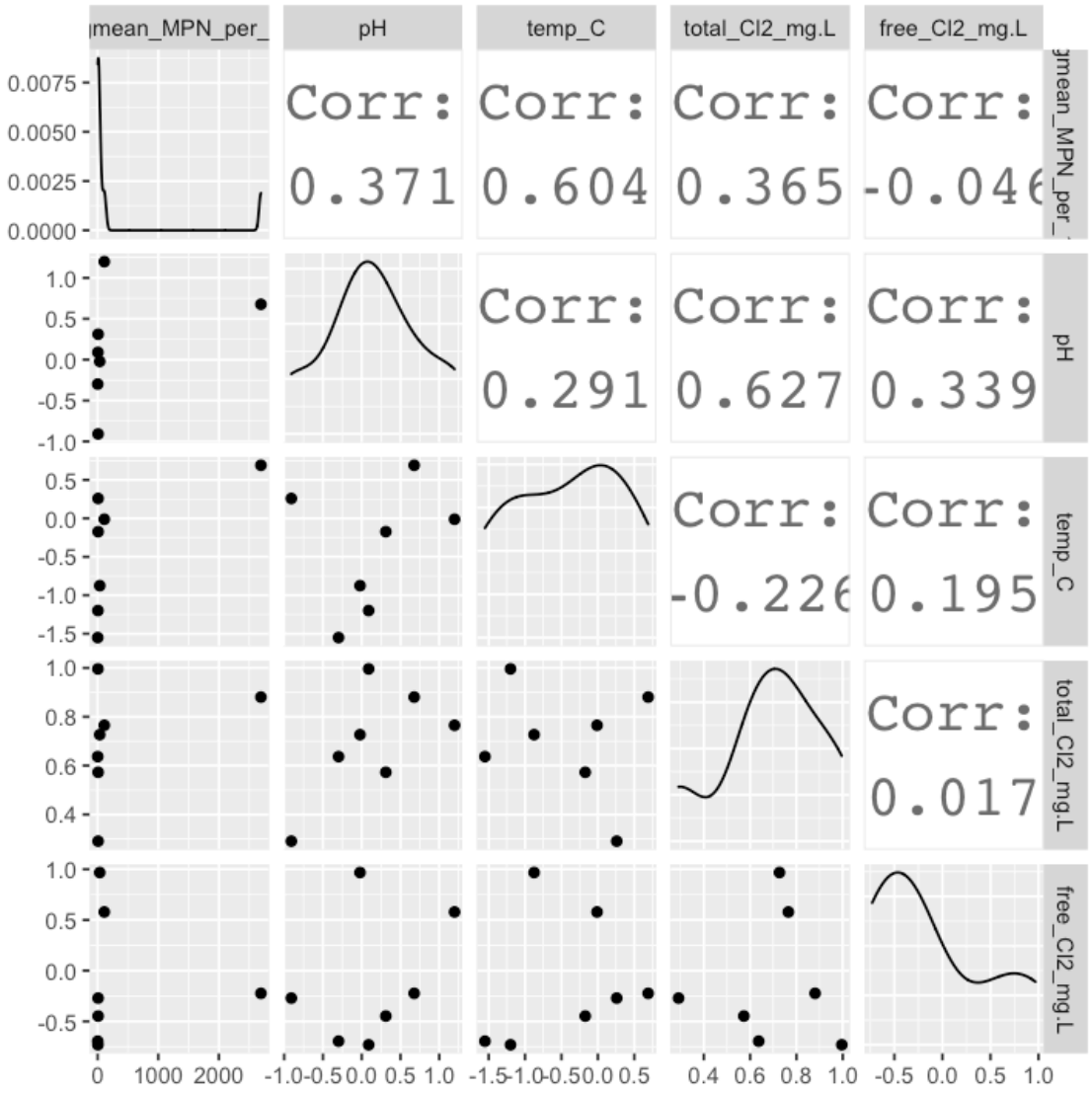
```
 ggpairs(f, upper = list(continuous = wrap('cor', size = 8) ) )

g<-df[df$location_code=="site_15",]
g<-g[,col]
 ggpairs(g, upper = list(continuous = wrap('cor', size = 8) ) )

h<-df[df$location_code=="site_10",]
h<-h[,col]
 ggpairs(h, upper = list(continuous = wrap('cor', size = 8) ) )

i<-df[df$location_code=="site_24",]
i<-i[,col]
 ggpairs(i, upper = list(continuous = wrap('cor', size = 8) ) )

j<-df[df$location_code=="site_ut",]
j<-j[,col]
 ggpairs(j, upper = list(continuous = wrap('cor', size = 8) ) )
```

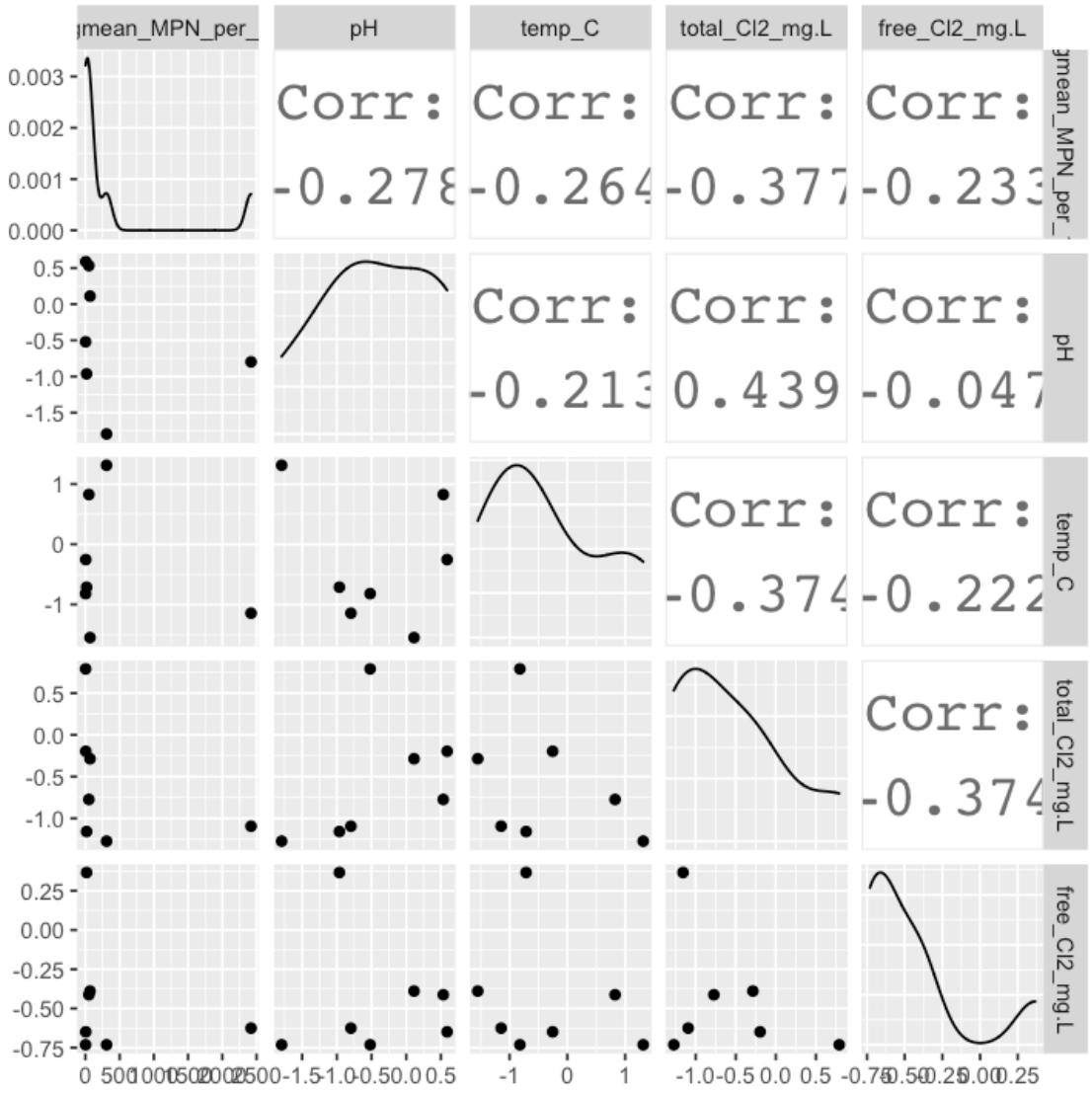

gmean_MPN_per_
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr: 0.399
Corr: 0.214
Corr: 0.021
Corr: 0.039
Corr: .770*
Corr: 0.551
Corr: 0.568
Corr: 0.492
Corr: 0.518
Corr: .993**

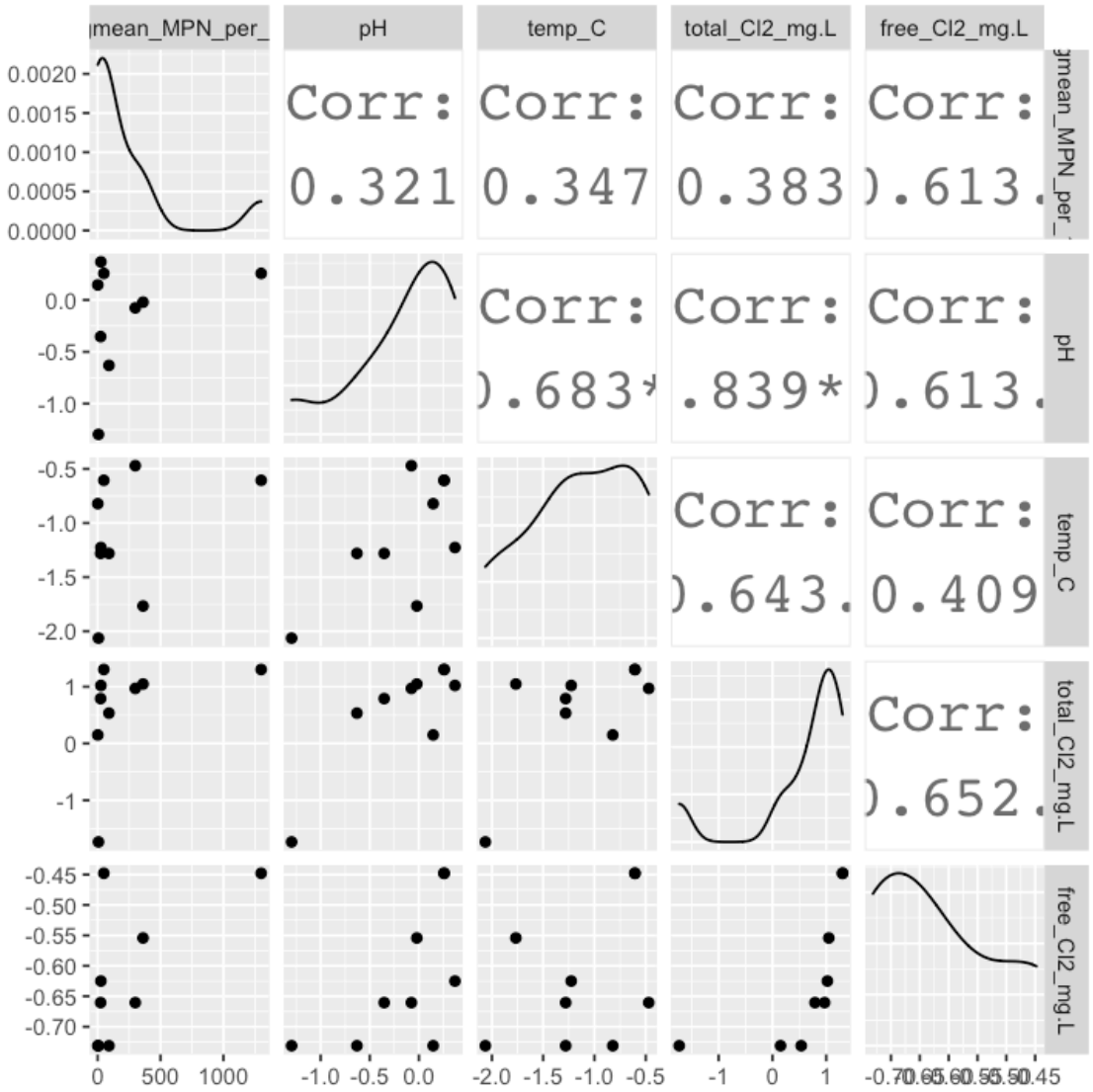

pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
-0.269
0.307
-0.182
-0.245
0.377
0.303
0.307
0.569
-0.516
.886**
0.10
0.05
0.00
1
0
-1
0.75
0.50
0.25
0.00
-0.25
-0.5
-1.0
3
2
1
0
0 50 100150200
-1 0 1
-0.25 0.00 0.25 0.50 0.75
-1.0 -0.5
0 1 2 3


gmean_MPN_per_
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
-0.127
0.159
0.269
-0.123
0.221
.527**
.472**
0.207
0.038
.213.
0.004
0.002
0.000
2
1
0
-1
-2
1.0
0.5
0.0
-0.5
0
1000
2000
3000


gmean_MPN_per_
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
-0.596
-0.030
-0.549
-0.384
Corr:
-0.257
0.834*
0.824*
Corr:
0.252
-0.082
Corr:
0.683.

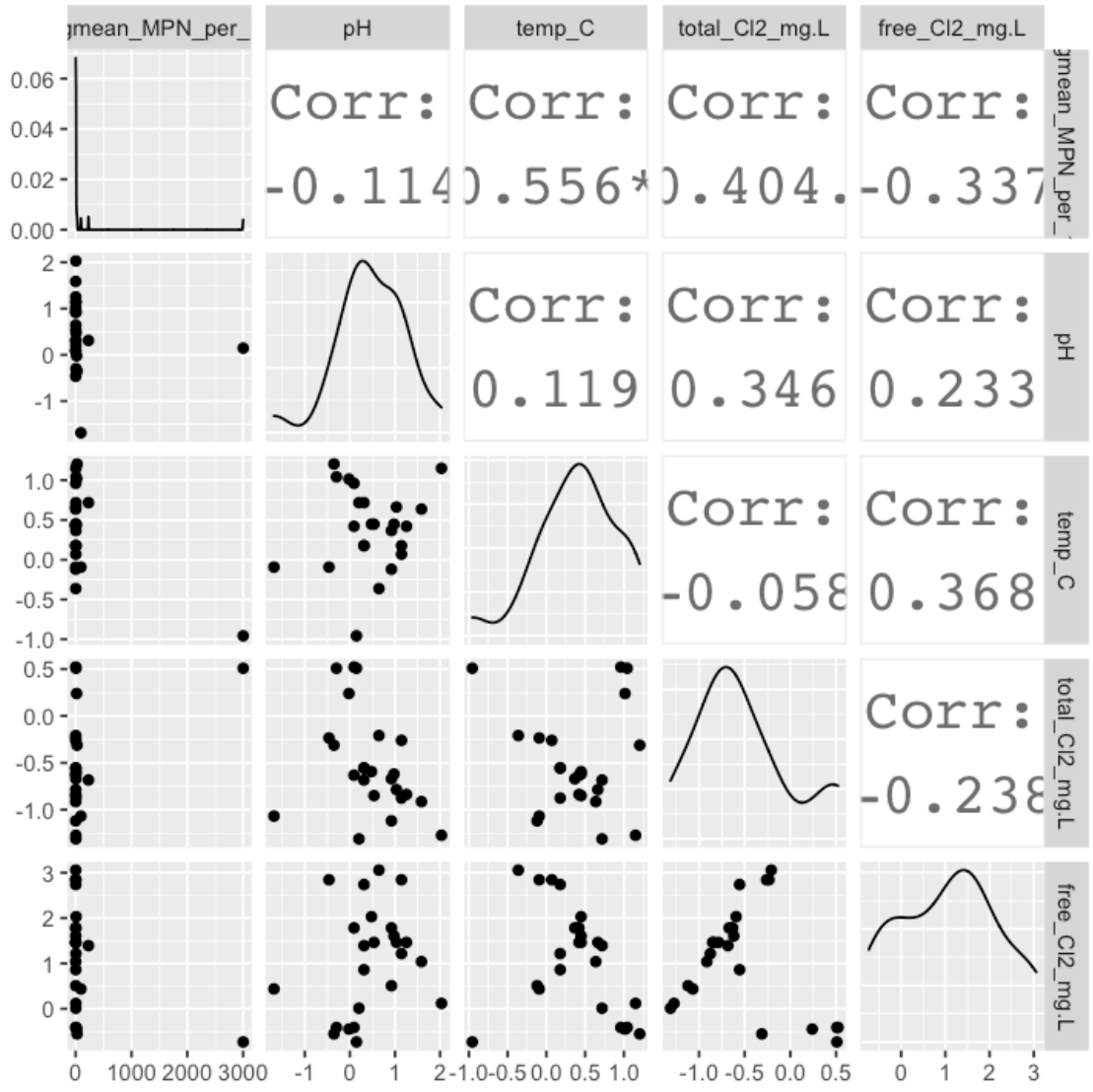

mean_MPN_per_
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr: 0.371
Corr: 0.604
Corr: 0.365
Corr: -0.046
Corr: 0.291
Corr: 0.627
Corr: 0.339
Corr: -0.226
Corr: 0.195
Corr: 0.017

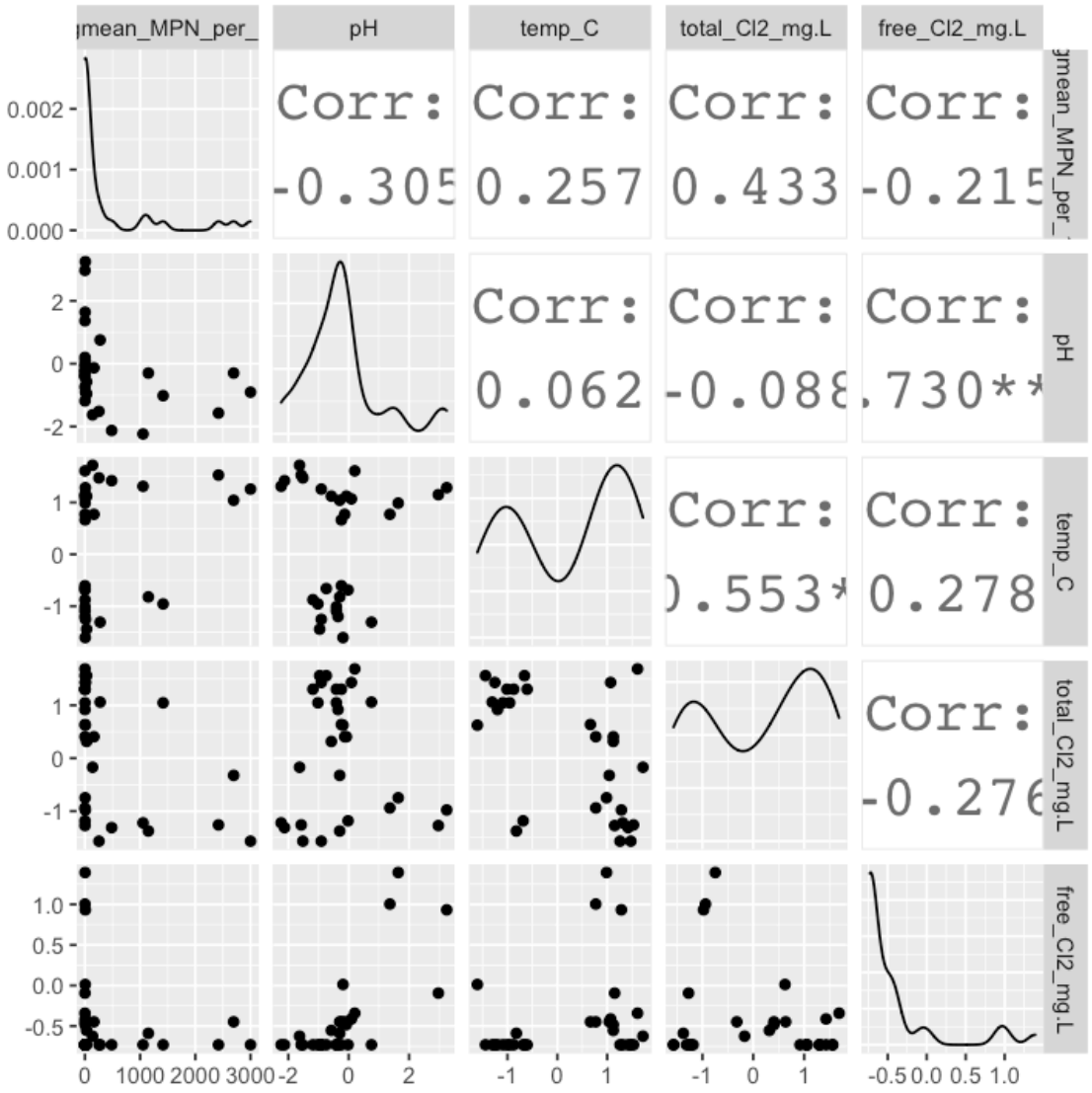

gmean_MPN_per_
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
-0.278
-0.264
-0.377
-0.233
Corr:
-0.213
0.439
-0.047
Corr:
-0.374
-0.222
Corr:
-0.374

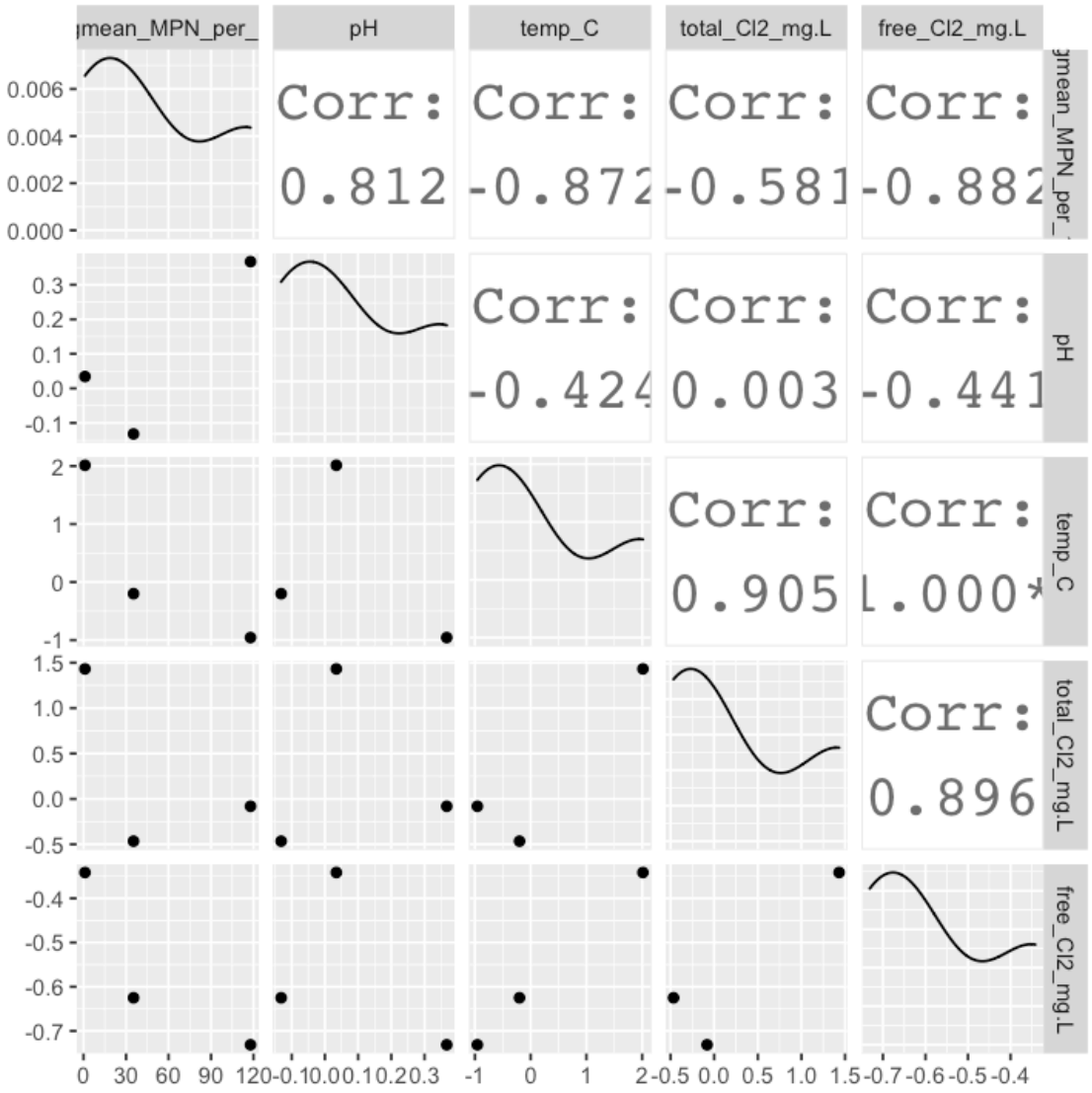


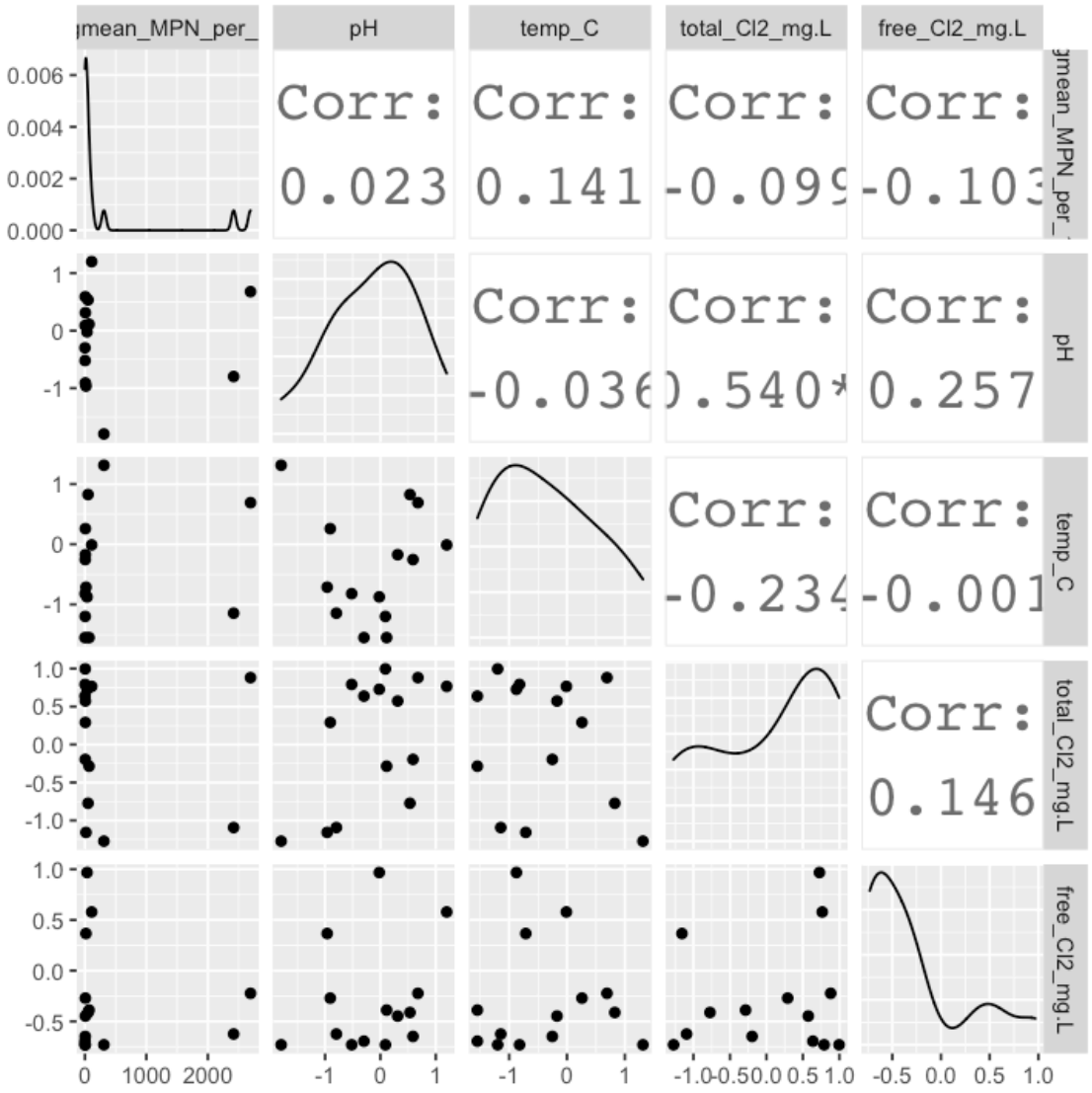
[36]:
```
g<-df[df$count== 1,]
g<-g[,col]
 ggpairs(g, upper = list(continuous = wrap('cor', size = 8) ) )

h<-df[df$count==2,]
h<-h[,col]
 ggpairs(h, upper = list(continuous = wrap('cor', size = 8) ) )

i<-df[df$count==3,]
i<-i[,col]
 ggpairs(i, upper = list(continuous = wrap('cor', size = 8) ) )

j<-df[df$count==7,]
```

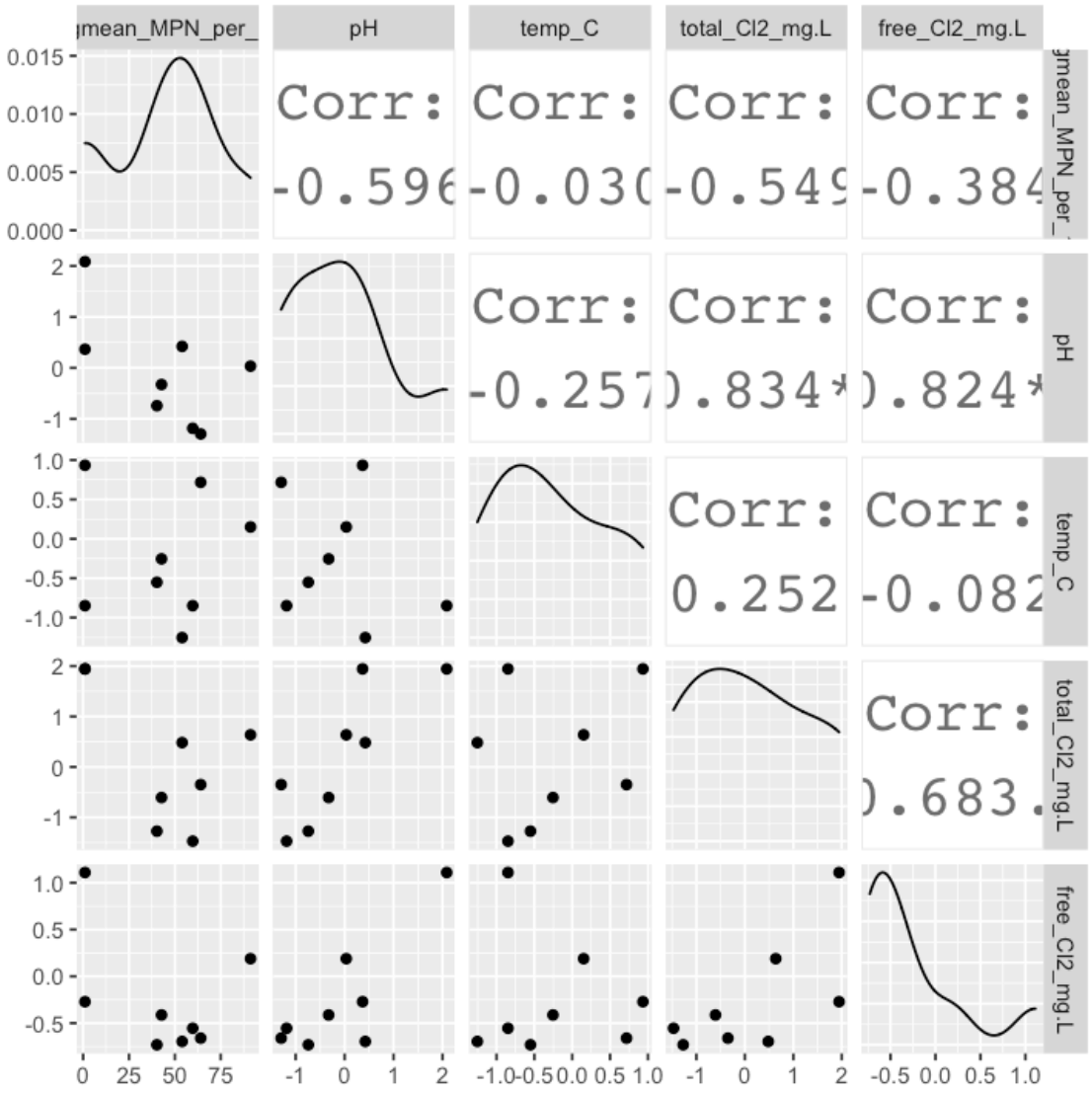
```
j<-j[,col]
 ggpairs(j, upper = list(continuous = wrap('cor', size = 8) ) )

k<-df[df$count==8,] #includes 2 sites but still correlated
k<-k[,col]
 ggpairs(k, upper = list(continuous = wrap('cor', size = 8) ) )

l<-df[df$count==9,]
l<-l[,col]
 ggpairs(l, upper = list(continuous = wrap('cor', size = 8) ) )
```

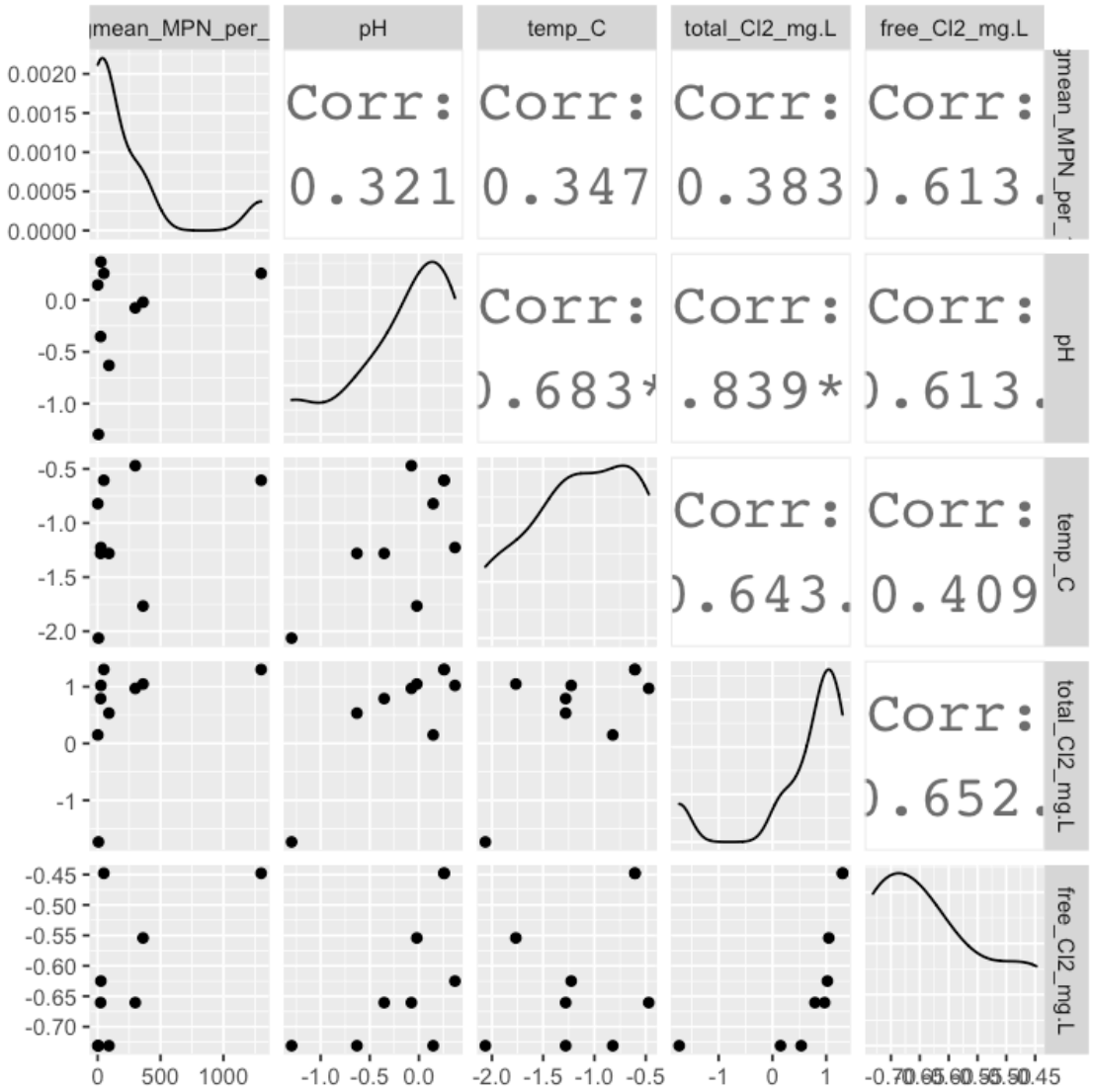



gmean_MPN_per_
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
-0.305
0.257
0.433
-0.215
0.062
-0.088
.730**
0.553*
0.278
-0.276


gmean_MPN_per_
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr: 0.812
Corr: -0.872
Corr: -0.581
Corr: -0.882
Corr: -0.424
Corr: 0.003
Corr: -0.441
Corr: 0.905
Corr: 1.000*
Corr: 0.896
0.006
0.004
0.002
0.000
0.3
0.2
0.1
0.0
-0.1
2
1
0
-1
1.5
1.0
0.5
0.0
-0.5
-0.4
-0.5
-0.6
-0.7
0 30 60 90 120
-0.1 0.0 0.1 0.2 0.3
-1 0 1 2
-0.5 0.0 0.5 1.0 1.5
-0.7 -0.6 -0.5 -0.4


mean_MPN_per_
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
0.023
Corr:
0.141
Corr:
-0.099
Corr:
-0.103
Corr:
-0.036
Corr:
0.540*
Corr:
0.257
Corr:
-0.234
Corr:
-0.001
Corr:
0.146


gmean_MPN_per_
pH
temp_C
total_Cl2_mg.L
free_Cl2_mg.L
Corr:
-0.596
-0.030
-0.549
-0.384
Corr:
-0.257
0.834*
0.824*
Corr:
0.252
-0.082
Corr:
0.683.



0.4.2 B. Develop Models (Zuur et al. 2013)

[37]:
```
#develop functions for later

fix_tab<- function(dfm, sci_lis, dci_lis){
    for (i in sci_lis) {
        dfm[,i]<-unlist(lapply(as.list(dfm[,i]), function (x) if(is.
↪na(x)==FALSE){formatC(x, format='e', digits =2)} else{x=x}))
    }
    for (j in dci_lis) {
        dfm[,j]<-unlist(lapply(as.list(dfm[,j]), function (x) if(is.
↪na(x)==FALSE){formatC(x, digits = 3)} else{x=x}))
```

```
    }

    return(dfm)
}

 make_stars <- function(pval) {
  stars = ""
  if(!is.na(pval)==TRUE){
      if(pval <= 0.0001)
          stars = "<0.0001"
      if(pval > 0.0001 & pval <= 0.001)
          stars = "<0.001"
      if(pval > 0.001 & pval <= 0.01)
          stars = "<0.01"
      if(pval > 0.01 & pval <= 0.05)
          stars = "<0.05"

  }
  else{
      stars=as.character(pval)
  }
stars
}
```

Model selection using AIC

stepwise model selection

[38]:
```
# number of datapoints in the model
length(CAM_ICC$SGPI)
```

80

[39]:
```
glmm_ICC.1<- glmer(SGPI~total_Cl2_mg.L+free_Cl2_mg.L+pH+temp_C+free_Cl2_mg.L:pH␣
 ↪ +total_Cl2_mg.L:pH + free_Cl2_mg.L:temp_C  +total_Cl2_mg.L:temp_C␣
 ↪+(1|location_code), data=CAM_ICC, family=Gamma(link="log"))
ss<-getME(glmm_ICC.1,c("theta","fixef"))
glmm_ICC.1<- update(glmm_ICC.1,start=ss)
# summary(glmm_ICC.1)
# allFit(glmm_ICC.1)
```

[40]:
```
#remove free chlorine
glmm_ICC.2a<- update(glmm_ICC.1, ~ . -free_Cl2_mg.L, start=NULL)
ss<-getME(glmm_ICC.2a,c("theta","fixef"))
glmm_ICC.2a<- update(glmm_ICC.2a,start=ss)

# remove temp
```

```
glmm_ICC.2b<- update(glmm_ICC.1, ~ . -temp_C, start=NULL)
ss<-getME(glmm_ICC.2b,c("theta","fixef"))
glmm_ICC.2b<- update(glmm_ICC.2b,start=ss)

#remove total:pH
glmm_ICC.2c<- update(glmm_ICC.1, ~ . -total_Cl2_mg.L:pH , start=NULL)
ss<-getME(glmm_ICC.2c,c("theta","fixef"))
glmm_ICC.2c<- update(glmm_ICC.2c,start=ss)

#remove free chlorine:temp
glmm_ICC.2d<- update(glmm_ICC.1, ~ . -free_Cl2_mg.L:temp_C , start=NULL)
ss<-getME(glmm_ICC.2d,c("theta","fixef"))
glmm_ICC.2d<- update(glmm_ICC.2d,start=ss)

#total chlorine:temp
glmm_ICC.2e<- update(glmm_ICC.1, ~ . -total_Cl2_mg.L:temp_C , start=NULL)
ss<-getME(glmm_ICC.2e,c("theta","fixef"))
glmm_ICC.2e<- update(glmm_ICC.2e,start=ss)

#remove free:pH
glmm_ICC.2f<- update(glmm_ICC.1, ~ . -free_Cl2_mg.L:pH , start=NULL)
ss<-getME(glmm_ICC.2f,c("theta","fixef"))
glmm_ICC.2f<- update(glmm_ICC.2f,start=ss)

#remove pH
glmm_ICC.2g<- update(glmm_ICC.1, ~ . -pH , start=NULL)
ss<-getME(glmm_ICC.2g,c("theta","fixef"))
glmm_ICC.2g<- update(glmm_ICC.2g,start=ss)


sel_t1=model.sel(glmm_ICC.1,glmm_ICC.2a,glmm_ICC.2b,glmm_ICC.2c,glmm_ICC.
↪2d,glmm_ICC.2e,glmm_ICC.2f,glmm_ICC.2g)
# summary(glmm_ICC.2c)
```

[41]:
```
#remove free chlorine
glmm_ICC.3a<- update(glmm_ICC.2c, ~ . -free_Cl2_mg.L , start=NULL)
ss<-getME(glmm_ICC.3a,c("theta","fixef"))
glmm_ICC.3a<- update(glmm_ICC.3a,start=ss)

# remove temp
glmm_ICC.3b<- update(glmm_ICC.2c, ~ . -temp_C , start=NULL)
ss<-getME(glmm_ICC.3b,c("theta","fixef"))
glmm_ICC.3b<- update(glmm_ICC.3b,start=ss)

# remove free:temp
glmm_ICC.3c<- update(glmm_ICC.2c, ~ . -free_Cl2_mg.L:temp_C , start=NULL)
ss<-getME(glmm_ICC.3c,c("theta","fixef"))
```

```
glmm_ICC.3c<- update(glmm_ICC.3c,start=ss)

# remove total:temp
glmm_ICC.3d<- update(glmm_ICC.2c, ~ . -total_Cl2_mg.L:temp_C , start=NULL)
ss<-getME(glmm_ICC.3d,c("theta","fixef"))
glmm_ICC.3d<- update(glmm_ICC.3d,start=ss)

# remove free:pH
glmm_ICC.3e<- update(glmm_ICC.2c, ~ . -free_Cl2_mg.L:pH , start=NULL)
ss<-getME(glmm_ICC.3e,c("theta","fixef"))
glmm_ICC.3e<- update(glmm_ICC.3e,start=ss)

# remove total:pH
glmm_ICC.3f<- update(glmm_ICC.2c, ~ . -total_Cl2_mg.L:pH , start=NULL)
ss<-getME(glmm_ICC.3f,c("theta","fixef"))
glmm_ICC.3f<- update(glmm_ICC.3f,start=ss)

# remove pH
glmm_ICC.3g<- update(glmm_ICC.2c, ~ . -pH , start=NULL)
ss<-getME(glmm_ICC.3g,c("theta","fixef"))
glmm_ICC.3g<- update(glmm_ICC.3g,start=ss)



sel_t2=model.sel(glmm_ICC.2c,glmm_ICC.3a,glmm_ICC.3b,glmm_ICC.3c,glmm_ICC.
↪3d,glmm_ICC.3e,glmm_ICC.3f,glmm_ICC.3g)
# summary(glmm_ICC.3c)
```

[42]:
```
#remove free chlorine
glmm_ICC.4a<- update(glmm_ICC.3c, ~ . -free_Cl2_mg.L, start=NULL)
ss<-getME(glmm_ICC.4a,c("theta","fixef"))
glmm_ICC.4a<- update(glmm_ICC.4a,start=ss)

# remove temp
glmm_ICC.4b<- update(glmm_ICC.3c, ~ . -temp_C, start=NULL)
ss<-getME(glmm_ICC.4b,c("theta","fixef"))
glmm_ICC.4b<- update(glmm_ICC.4b,start=ss)

# remove total:temp
glmm_ICC.4c<- update(glmm_ICC.3c, ~ . -total_Cl2_mg.L:temp_C, start=NULL)
ss<-getME(glmm_ICC.4c,c("theta","fixef"))
glmm_ICC.4c<- update(glmm_ICC.4c,start=ss)

# remove free:pH
glmm_ICC.4d<- update(glmm_ICC.3c, ~ . -free_Cl2_mg.L:pH, start=NULL)
ss<-getME(glmm_ICC.4d,c("theta","fixef"))
glmm_ICC.4d<- update(glmm_ICC.4d,start=ss)
```

```
# remove total:pH
glmm_ICC.4e<- update(glmm_ICC.3c, ~ . -total_Cl2_mg.L:pH, start=NULL)
ss<-getME(glmm_ICC.4e,c("theta","fixef"))
glmm_ICC.4e<- update(glmm_ICC.4e,start=ss)

# remove pH
glmm_ICC.4f<- update(glmm_ICC.3f, ~ . -pH, start=NULL)
ss<-getME(glmm_ICC.4f,c("theta","fixef"))
glmm_ICC.4f<- update(glmm_ICC.4f,start=ss)

sel_t3=model.sel(glmm_ICC.3c,glmm_ICC.4a,glmm_ICC.4b,glmm_ICC.4c,glmm_ICC.
↪4d,glmm_ICC.4e,glmm_ICC.4f)
# summary(glmm_ICC.4a)
```

[43]:
```
# check by removing next term
glmm_ICC.5<- update(glmm_ICC.3c, ~ . -free_Cl2_mg.L:pH, start=NULL)
ss<-getME(glmm_ICC.5,c("theta","fixef"))
glmm_ICC.5<- update(glmm_ICC.5,start=ss)

sel_t4=model.sel(glmm_ICC.4a,glmm_ICC.5)
# summary(glmm_ICC.4a)
```

0.5 Table S3

Final model output

[44]:
```
table_fin=rbind(sel_t1,sel_t2,sel_t3,sel_t4)
table_fin

stargazer(table_fin,type="html", summary=FALSE,  out=paste(path_tab,"Table_S3.
↪doc"))
summary(glmm_ICC.2c)
summary(glmm_ICC.3c)
summary(glmm_ICC.4a)
```

	(Intercept) <dbl>	free_Cl2_mg.L <dbl>	pH <dbl>	temp_C <dbl>	total_Cl2 <dbl>
glmm_ICC.4a	8.634886	NA	0.3978666	0.34530163	-1.310262
glmm_ICC.4a1	8.634886	NA	0.3978666	0.34530163	-1.310262
glmm_ICC.3c	8.460388	-0.397228282	0.5242192	0.35741346	-1.302844
glmm_ICC.3c1	8.460388	-0.397228282	0.5242192	0.35741346	-1.302844
glmm_ICC.4e	8.460388	-0.397228282	0.5242192	0.35741346	-1.302844
glmm_ICC.3b	8.244307	-0.868041059	0.5655500	NA	-1.225143
glmm_ICC.4c	8.467650	-0.508532806	0.5203894	0.30145193	-1.329605
glmm_ICC.3a	8.636443	NA	0.4177871	0.40217811	-1.322224
glmm_ICC.3d	8.287585	-0.863421217	0.5584155	0.05364703	-1.257774
glmm_ICC.2c	8.342619	-0.648384125	0.5484776	0.19376017	-1.261160
glmm_ICC.2c1	8.342619	-0.648384125	0.5484776	0.19376017	-1.261160
glmm_ICC.3f	8.342619	-0.648384125	0.5484776	0.19376017	-1.261160
glmm_ICC.4d	8.705280	-0.002713199	0.3377636	0.38791869	-1.298339
glmm_ICC.5	8.705280	-0.002713199	0.3377636	0.38791869	-1.298339
glmm_ICC.2d	8.433819	-0.400490617	0.4802220	0.32938549	-1.260422
glmm_ICC.2b	8.242876	-0.854191074	0.5499711	NA	-1.213378
glmm_ICC.2a	8.606293	NA	0.3693416	0.39053566	-1.275764
glmm_ICC.2e	8.287976	-0.836055972	0.5332802	0.05959847	-1.241108
glmm_ICC.3e	8.620837	-0.192438622	0.3482131	0.25659834	-1.269300
glmm_ICC.1	8.343099	-0.619998192	0.5231414	0.19911997	-1.243844
glmm_ICC.2f	8.511057	-0.239277345	0.2957379	0.26793466	-1.182219
glmm_ICC.3g	8.513812	0.017928886	NA	0.23390813	-1.135292
glmm_ICC.4f	8.513812	0.017928886	NA	0.23390813	-1.135292
glmm_ICC.2g	8.481886	0.014222615	NA	0.26923075	-1.086753
glmm_ICC.4b	8.521356	-0.331218730	0.4017245	NA	-1.339975

A model.selection: 25 × 14


```
<table style="text-align:center"><tr><td colspan="15" style="border-bottom: 1px
solid black"></td></tr><tr><td style="text-align:left"></td><td>(Intercept)</td>
<td>free_Cl2_mg.L</td><td>pH</td><td>temp_C</td><td>total_Cl2_mg.L</td><td>free_
Cl2_mg.L:pH</td><td>free_Cl2_mg.L:temp_C</td><td>pH:total_Cl2_mg.L</td><td>temp_
C:total_Cl2_mg.L</td><td>df</td><td>logLik</td><td>AICc</td><td>delta</td><td>we
ight</td></tr>
<tr><td colspan="15" style="border-bottom: 1px solid black"></td></tr><tr><td
style="text-align:left">glmm_ICC.4a</td><td>8.635</td><td></td><td>0.398</td><td
>0.345</td><td>-1.310</td><td>0.385</td><td></td><td></td><td>-0.237</td><td>8</
td><td>-754.175</td><td>1,526.379</td><td>0</td><td>0.141</td></tr>
<tr><td style="text-align:left">glmm_ICC.4a1</td><td>8.635</td><td></td><td>0.39
8</td><td>0.345</td><td>-1.310</td><td>0.385</td><td></td><td></td><td>-0.237</t
d><td>8</td><td>-754.175</td><td>1,526.379</td><td>0</td><td>0.141</td></tr>
<tr><td style="text-align:left">glmm_ICC.3c</td><td>8.460</td><td>-0.397</td><td
>0.524</td><td>0.357</td><td>-1.303</td><td>0.534</td><td></td><td></td><td>-0.2
14</td><td>9</td><td>-753.443</td><td>1,527.458</td><td>1.079</td><td>0.082</td>
</tr>
<tr><td style="text-align:left">glmm_ICC.3c1</td><td>8.460</td><td>-0.397</td><t
d>0.524</td><td>0.357</td><td>-1.303</td><td>0.534</td><td></td><td></td><td>-0.
```

214</td><td>9</td><td>-753.443</td><td>1,527.458</td><td>1.079</td><td>0.082</td
></tr>
<tr><td style="text-align:left">glmm_ICC.4e</td><td>8.460</td><td>-0.397</td><td
>0.524</td><td>0.357</td><td>-1.303</td><td>0.534</td><td></td><td></td><td>-0.2
14</td><td>9</td><td>-753.443</td><td>1,527.458</td><td>1.079</td><td>0.082</td>
</tr>
<tr><td style="text-align:left">glmm_ICC.3b</td><td>8.244</td><td>-0.868</td><td
>0.566</td><td></td><td>-1.225</td><td>0.589</td><td>-0.703</td><td></td><td>-0.
135</td><td>9</td><td>-753.500</td><td>1,527.571</td><td>1.192</td><td>0.078</td
></tr>
<tr><td style="text-align:left">glmm_ICC.4c</td><td>8.468</td><td>-0.509</td><td
>0.520</td><td>0.301</td><td>-1.330</td><td>0.741</td><td></td><td></td><td></td
><td>8</td><td>-754.969</td><td>1,527.967</td><td>1.588</td><td>0.064</td></tr>
<tr><td style="text-align:left">glmm_ICC.3a</td><td>8.636</td><td></td><td>0.418
</td><td>0.402</td><td>-1.322</td><td>0.412</td><td>0.116</td><td></td><td>-0.24
4</td><td>9</td><td>-754.120</td><td>1,528.811</td><td>2.432</td><td>0.042</td><
/tr>
<tr><td style="text-align:left">glmm_ICC.3d</td><td>8.288</td><td>-0.863</td><td
>0.558</td><td>0.054</td><td>-1.258</td><td>0.708</td><td>-0.566</td><td></td><t
d></td><td>9</td><td>-754.162</td><td>1,528.895</td><td>2.516</td><td>0.040</td>
</tr>
<tr><td style="text-align:left">glmm_ICC.2c</td><td>8.343</td><td>-0.648</td><td
>0.548</td><td>0.194</td><td>-1.261</td><td>0.553</td><td>-0.355</td><td></td><t
d>-0.181</td><td>10</td><td>-753.155</td><td>1,529.497</td><td>3.119</td><td>0.0
30</td></tr>
<tr><td style="text-align:left">glmm_ICC.2c1</td><td>8.343</td><td>-0.648</td><t
d>0.548</td><td>0.194</td><td>-1.261</td><td>0.553</td><td>-0.355</td><td></td><
td>-0.181</td><td>10</td><td>-753.155</td><td>1,529.497</td><td>3.119</td><td>0.
030</td></tr>
<tr><td style="text-align:left">glmm_ICC.3f</td><td>8.343</td><td>-0.648</td><td
>0.548</td><td>0.194</td><td>-1.261</td><td>0.553</td><td>-0.355</td><td></td><t
d>-0.181</td><td>10</td><td>-753.155</td><td>1,529.497</td><td>3.119</td><td>0.0
30</td></tr>
<tr><td style="text-align:left">glmm_ICC.4d</td><td>8.705</td><td>-0.003</td><td
>0.338</td><td>0.388</td><td>-1.298</td><td></td><td></td><td></td><td>-0.330</t
d><td>8</td><td>-755.800</td><td>1,529.629</td><td>3.250</td><td>0.028</td></tr>
<tr><td style="text-align:left">glmm_ICC.5</td><td>8.705</td><td>-0.003</td><td>
0.338</td><td>0.388</td><td>-1.298</td><td></td><td></td><td></td><td>-0.330</td
><td>8</td><td>-755.800</td><td>1,529.629</td><td>3.250</td><td>0.028</td></tr>
<tr><td style="text-align:left">glmm_ICC.2d</td><td>8.434</td><td>-0.400</td><td
>0.480</td><td>0.329</td><td>-1.260</td><td>0.458</td><td></td><td>0.078</td><td
>-0.208</td><td>10</td><td>-753.320</td><td>1,529.829</td><td>3.451</td><td>0.02
5</td></tr>
<tr><td style="text-align:left">glmm_ICC.2b</td><td>8.243</td><td>-0.854</td><td
>0.550</td><td></td><td>-1.213</td><td>0.563</td><td>-0.681</td><td>0.027</td><t
d>-0.134</td><td>10</td><td>-753.486</td><td>1,530.161</td><td>3.782</td><td>0.0
21</td></tr>
<tr><td style="text-align:left">glmm_ICC.2a</td><td>8.606</td><td></td><td>0.369

</td><td>0.391</td><td>-1.276</td><td>0.328</td><td>0.164</td><td>0.095</td><td>
-0.239</td><td>10</td><td>-753.951</td><td>1,531.090</td><td>4.712</td><td>0.013
</td></tr>
<tr><td style="text-align:left">glmm_ICC.2e</td><td>8.288</td><td>-0.836</td><td
>0.533</td><td>0.060</td><td>-1.241</td><td>0.667</td><td>-0.522</td><td>0.041</
td><td></td><td>10</td><td>-754.131</td><td>1,531.450</td><td>5.071</td><td>0.01
1</td></tr>
<tr><td style="text-align:left">glmm_ICC.3e</td><td>8.621</td><td>-0.192</td><td
>0.348</td><td>0.257</td><td>-1.269</td><td></td><td>-0.282</td><td></td><td>-0.
301</td><td>9</td><td>-755.628</td><td>1,531.828</td><td>5.449</td><td>0.009</td
></tr>
<tr><td style="text-align:left">glmm_ICC.1</td><td>8.343</td><td>-0.620</td><td>
0.523</td><td>0.199</td><td>-1.244</td><td>0.511</td><td>-0.312</td><td>0.041</t
d><td>-0.181</td><td>11</td><td>-753.123</td><td>1,532.129</td><td>5.751</td><td
>0.008</td></tr>
<tr><td style="text-align:left">glmm_ICC.2f</td><td>8.511</td><td>-0.239</td><td
>0.296</td><td>0.268</td><td>-1.182</td><td></td><td>-0.076</td><td>0.205</td><t
d>-0.257</td><td>10</td><td>-754.510</td><td>1,532.209</td><td>5.830</td><td>0.0
08</td></tr>
<tr><td style="text-align:left">glmm_ICC.3g</td><td>8.514</td><td>0.018</td><td>
</td><td>0.234</td><td>-1.135</td><td>0.235</td><td>-0.168</td><td></td><td>-0.2
00</td><td>9</td><td>-756.943</td><td>1,534.458</td><td>8.079</td><td>0.002</td>
</tr>
<tr><td style="text-align:left">glmm_ICC.4f</td><td>8.514</td><td>0.018</td><td>
</td><td>0.234</td><td>-1.135</td><td>0.235</td><td>-0.168</td><td></td><td>-0.2
00</td><td>9</td><td>-756.943</td><td>1,534.458</td><td>8.079</td><td>0.002</td>
</tr>
<tr><td style="text-align:left">glmm_ICC.2g</td><td>8.482</td><td>0.014</td><td>
</td><td>0.269</td><td>-1.087</td><td>0.079</td><td>0.056</td><td>0.229</td><td>
-0.199</td><td>10</td><td>-755.784</td><td>1,534.757</td><td>8.378</td><td>0.002
</td></tr>
<tr><td style="text-align:left">glmm_ICC.4b</td><td>8.521</td><td>-0.331</td><td
>0.402</td><td></td><td>-1.340</td><td>0.727</td><td></td><td></td><td>-0.049</t
d><td>8</td><td>-759.682</td><td>1,537.391</td><td>11.013</td><td>0.001</td></tr
>
<tr><td colspan="15" style="border-bottom: 1px solid black"></td></tr></table>

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: Gamma  ( log )
Formula:
SGPI ~ total_Cl2_mg.L + free_Cl2_mg.L + pH + temp_C + (1 | location_code) +
    free_Cl2_mg.L:pH + free_Cl2_mg.L:temp_C + total_Cl2_mg.L:temp_C
   Data: CAM_ICC

     AIC      BIC   logLik deviance df.resid
  1526.3   1550.1   -753.2   1506.3       70
```

```
Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.2730 -0.7054 -0.1998  0.4558  3.2482

Random effects:
 Groups        Name        Variance Std.Dev.
 location_code (Intercept) 0.2380   0.4879
 Residual                  0.5875   0.7665
Number of obs: 80, groups:  location_code, 24

Fixed effects:
                      Estimate Std. Error t value Pr(>|z|)
(Intercept)             8.3426     0.2755  30.283  < 2e-16 ***
total_Cl2_mg.L         -1.2612     0.1418  -8.893  < 2e-16 ***
free_Cl2_mg.L          -0.6484     0.4503  -1.440  0.14990
pH                      0.5485     0.2009   2.731  0.00632 **
temp_C                  0.1938     0.2341   0.828  0.40780
free_Cl2_mg.L:pH        0.5525     0.2599   2.126  0.03353 *
free_Cl2_mg.L:temp_C   -0.3554     0.4627  -0.768  0.44242
total_Cl2_mg.L:temp_C  -0.1805     0.1256  -1.438  0.15051
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
            (Intr) tt_C2_.L fr_C2_.L pH     temp_C f_C2_.L:H f_C2_.L:_
ttl_Cl2_m.L -0.444
fr_Cl2_mg.L  0.721 -0.299
pH          -0.151 -0.296   -0.469
temp_C       0.466 -0.317    0.614   -0.090
fr_Cl2_.L:H -0.368  0.038   -0.374    0.491 -0.159
fr_C2_.L:_C  0.532 -0.377    0.708   -0.169  0.910 -0.101
tt_C2_.L:_C -0.182  0.016   -0.348    0.051 -0.440  0.410    -0.344


Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: Gamma  ( log )
Formula:
SGPI ~ total_Cl2_mg.L + free_Cl2_mg.L + pH + temp_C + (1 | location_code) +
    free_Cl2_mg.L:pH + total_Cl2_mg.L:temp_C
   Data: CAM_ICC

     AIC      BIC   logLik deviance df.resid
  1524.9   1546.3   -753.4   1506.9       71

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.2767 -0.7013 -0.1678  0.4620  3.3463
```

```
Random effects:
 Groups        Name        Variance Std.Dev.
 location_code (Intercept) 0.2283   0.4778
 Residual                  0.5865   0.7658
Number of obs: 80, groups:  location_code, 24

Fixed effects:
                      Estimate Std. Error t value Pr(>|z|)
(Intercept)            8.46039    0.23068  36.676  < 2e-16 ***
total_Cl2_mg.L        -1.30284    0.13225  -9.851  < 2e-16 ***
free_Cl2_mg.L         -0.39723    0.31625  -1.256 0.209100
pH                     0.52422    0.19786   2.649 0.008062 **
temp_C                 0.35741    0.09774   3.657 0.000255 ***
free_Cl2_mg.L:pH       0.53448    0.25648   2.084 0.037169 *
total_Cl2_mg.L:temp_C -0.21379    0.11938  -1.791 0.073327 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
            (Intr) tt_C2_.L fr_C2_.L pH     temp_C f_C2_.L:
ttl_Cl2_m.L -0.310
fr_Cl2_mg.L  0.570 -0.040
pH          -0.060 -0.414   -0.502
temp_C      -0.027  0.054   -0.087    0.165
fr_Cl2_.L:H -0.373 -0.020   -0.444    0.472 -0.169
tt_C2_.L:_C -0.005 -0.141   -0.168    0.004 -0.321  0.428


Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: Gamma  ( log )
Formula:
SGPI ~ total_Cl2_mg.L + pH + temp_C + (1 | location_code) + free_Cl2_mg.L:pH +
    total_Cl2_mg.L:temp_C
   Data: CAM_ICC

     AIC      BIC   logLik deviance df.resid
  1524.4   1543.4   -754.2   1508.4       72

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.2783 -0.6711 -0.1629  0.4668  3.2411

Random effects:
 Groups        Name        Variance Std.Dev.
 location_code (Intercept) 0.2608   0.5107
 Residual                  0.5822   0.7630
```

```
Number of obs: 80, groups:  location_code, 24

Fixed effects:
                      Estimate Std. Error t value Pr(>|z|)
(Intercept)            8.63489    0.19228  44.909  < 2e-16 ***
total_Cl2_mg.L        -1.31026    0.13233  -9.902  < 2e-16 ***
pH                     0.39787    0.17138   2.322 0.020259 *
temp_C                 0.34530    0.09709   3.557 0.000376 ***
pH:free_Cl2_mg.L       0.38531    0.23019   1.674 0.094160 .
total_Cl2_mg.L:temp_C -0.23737    0.11665  -2.035 0.041862 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
            (Intr) tt_C2_.L pH     temp_C pH:_C2
ttl_Cl2_m.L -0.336
pH           0.303 -0.499
temp_C       0.018  0.065    0.124
pH:fr_C2_.L -0.161 -0.047    0.337 -0.227
tt_C2_.L:_C  0.094 -0.148   -0.094 -0.342  0.391
```

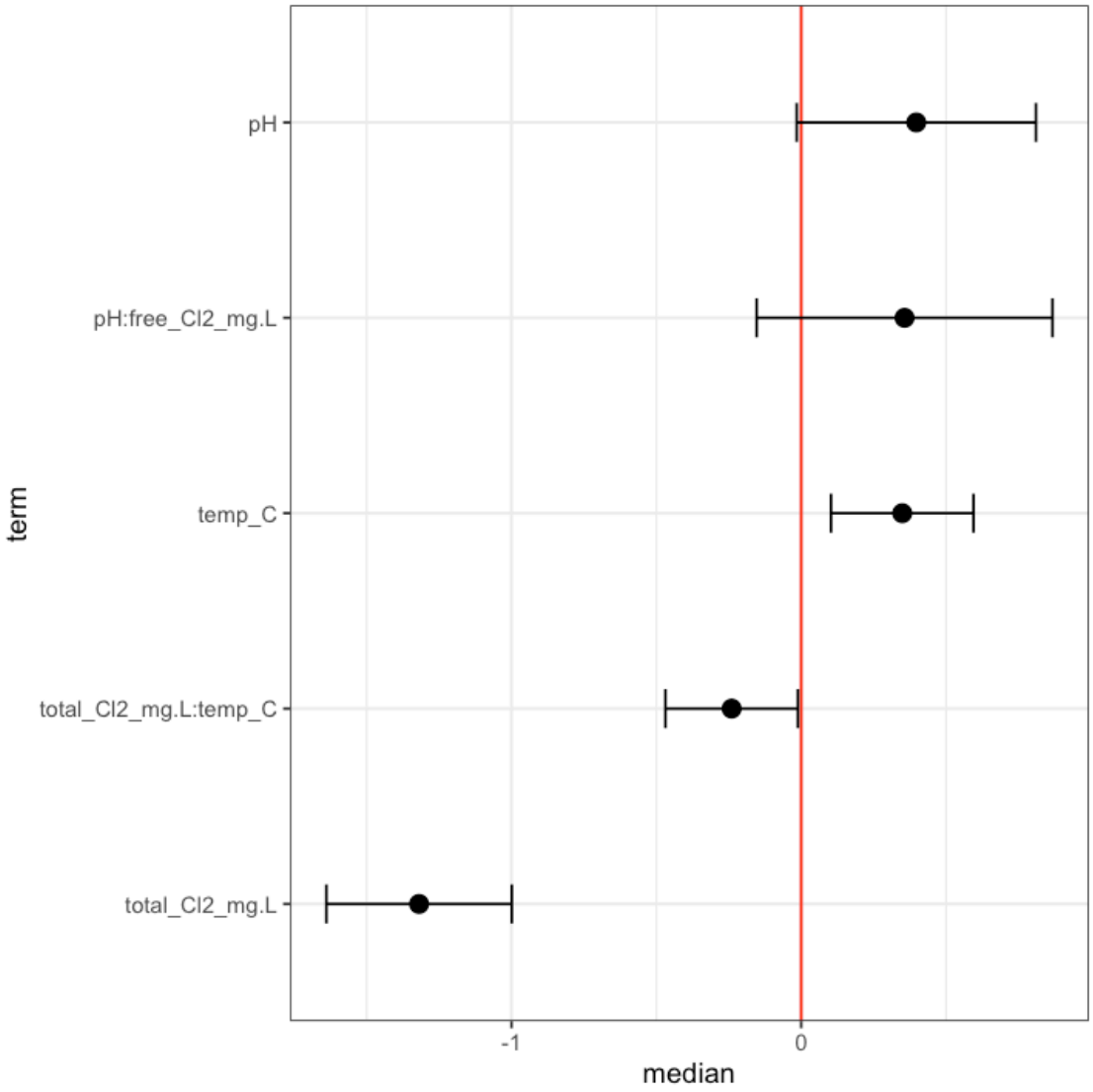
[45]:
```
glmm_ICC<-glmm_ICC.4a
summary(glmm_ICC)
getME(glmm_ICC,"sigma")
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: Gamma  ( log )
Formula:
SGPI ~ total_Cl2_mg.L + pH + temp_C + (1 | location_code) + free_Cl2_mg.L:pH +
    total_Cl2_mg.L:temp_C
   Data: CAM_ICC

     AIC      BIC   logLik deviance df.resid
  1524.4   1543.4   -754.2   1508.4       72

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.2783 -0.6711 -0.1629  0.4668  3.2411

Random effects:
 Groups        Name        Variance Std.Dev.
 location_code (Intercept) 0.2608   0.5107
 Residual                  0.5822   0.7630
Number of obs: 80, groups:  location_code, 24

Fixed effects:
```

```
                      Estimate Std. Error t value Pr(>|z|)
(Intercept)            8.63489    0.19228  44.909  < 2e-16 ***
total_Cl2_mg.L        -1.31026    0.13233  -9.902  < 2e-16 ***
pH                     0.39787    0.17138   2.322 0.020259 *
temp_C                 0.34530    0.09709   3.557 0.000376 ***
pH:free_Cl2_mg.L       0.38531    0.23019   1.674 0.094160 .
total_Cl2_mg.L:temp_C -0.23737    0.11665  -2.035 0.041862 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
            (Intr) tt_C2_.L pH     temp_C pH:_C2
ttl_Cl2_m.L -0.336
pH           0.303 -0.499
temp_C       0.018  0.065    0.124
pH:fr_C2_.L -0.161 -0.047    0.337 -0.227
tt_C2_.L:_C  0.094 -0.148   -0.094 -0.342  0.391
```

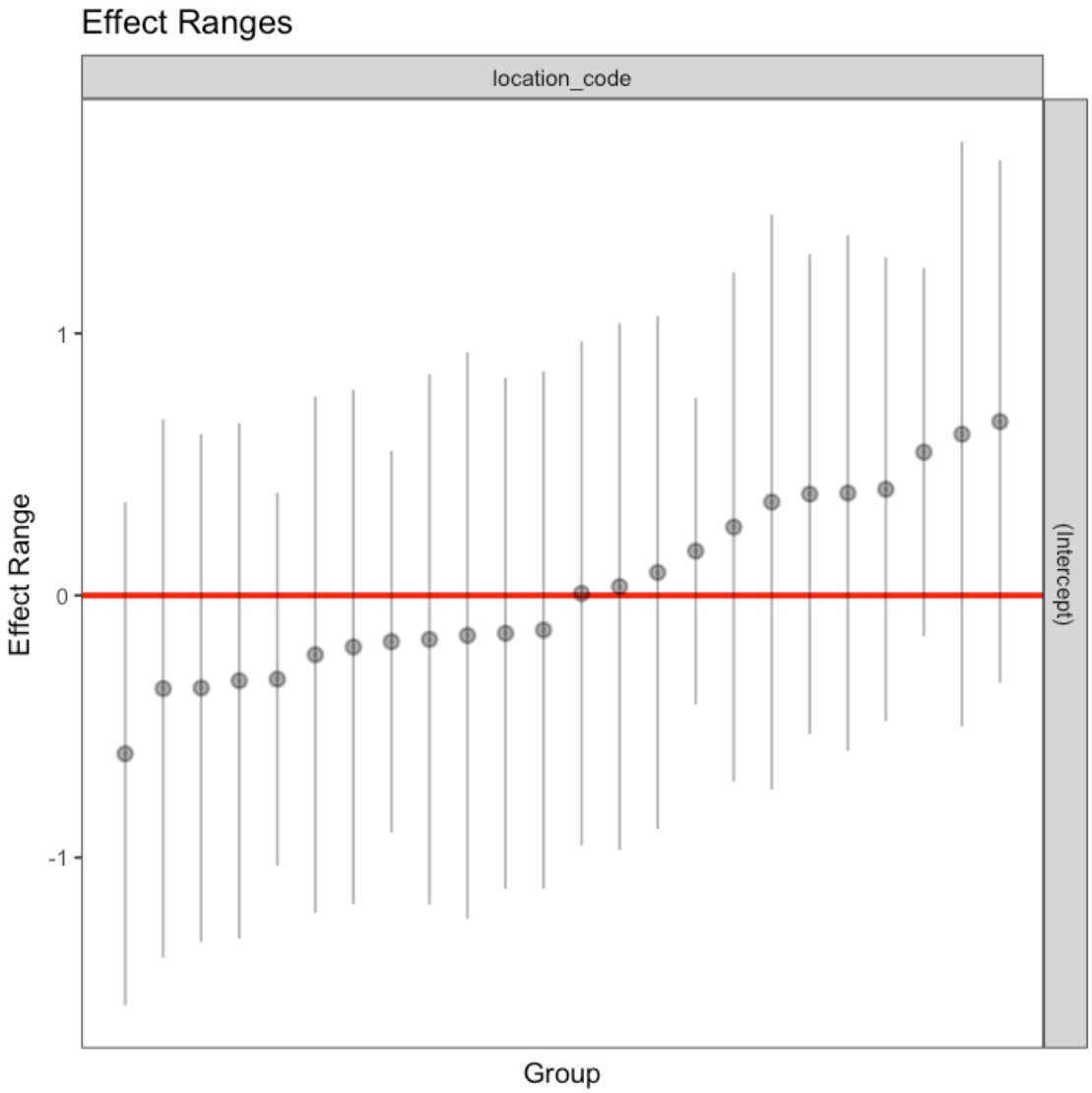
0.763007947415628

[46]:
```
#following this guide https://cran.r-project.org/web/packages/merTools/
 ↪vignettes/merToolsIntro.html
fe<-FEsim(glmm_ICC,1000, seed=30)
plotFEsim(fe, level=0.95)+
  theme_bw()
```



most fixed effects are different than 0 except pH:total chlorine

[47]:
```
re<-REsim(glmm_ICC, 1000, seed=30)
plotREsim(re,level=0.95)
```



random effects were indistinguishable from 0 for all sites

0.6 Table 2

model summary

[48]:
```
#get info
allv.itdf<- tidy(glmm_ICC, effects = c("ran_pars","fixed"), scales =␣
↪c("vcov",NA)) %>% mutate(signif = sapply(p.value, function(x) make_stars(x)))
wald_ci<-as.data.frame(confint(glmm_ICC, method="Wald") )
wald_ci$term<- as.list(rownames(wald_ci))
rownames(wald_ci)<-c()
wald_ci$term<-gsub( ".sig01", "var__(Intercept)",wald_ci$term)
```

```
wald_ci$term<-gsub(".sigma", "var__Observation",wald_ci$term)

allv.itdf<-left_join(allv.itdf,wald_ci)
allv.itdf



allv_tdf<-allv.itdf[c("term","estimate","std.error","p.value","signif", "2.5␣
↪%", "97.5 %")]
colnames(allv_tdf)<-c( "parameter", "estimate", "std.error", "p.
↪value","signif", "2.5", "97.5")
allv_tdf<-fix_tab(allv_tdf,NULL, c("estimate","97.5","2.5","std.error","p.
↪value"))
allv_tdf$parameter<-gsub("_", " ", allv_tdf$parameter)
allv_tdf$parameter<-gsub("Cl2", "chlorine", allv_tdf$parameter)
allv_tdf$parameter<-gsub("mg.L", "", allv_tdf$parameter)
allv_tdf$parameter<-gsub("(Intercept)", "intercept", allv_tdf$parameter)
allv_tdf$parameter<-gsub("(intercept)", "intercept", allv_tdf$parameter)
allv_tdf$parameter<-gsub("sd", "standard deviation of the", allv_tdf$parameter)

allv_tdf
stargazer(allv_tdf ,type="html", summary=FALSE,  out=paste(path_tab,"Table_2.
↪doc"))
```

```
Joining, by = "term"
```

A tibble: 8 × 10

effect <chr>	group <chr>	term <chr>	estimate <dbl>	std.error <dbl>	statistic <dbl>	
fixed	NA	(Intercept)	8.6348860	0.19227716	44.908536	
fixed	NA	total_Cl2_mg.L	-1.3102620	0.13232694	-9.901703	
fixed	NA	pH	0.3978666	0.17138239	2.321514	
fixed	NA	temp_C	0.3453016	0.09708671	3.556631	
fixed	NA	pH:free_Cl2_mg.L	0.3853111	0.23019439	1.673851	
fixed	NA	total_Cl2_mg.L:temp_C	-0.2373743	0.11665247	-2.034885	
ran_pars	location_code	var___(Intercept)	0.2607637	NA	NA	
ran_pars	Residual	var___Observation	0.5821811	NA	NA	

A tibble: 8 × 7

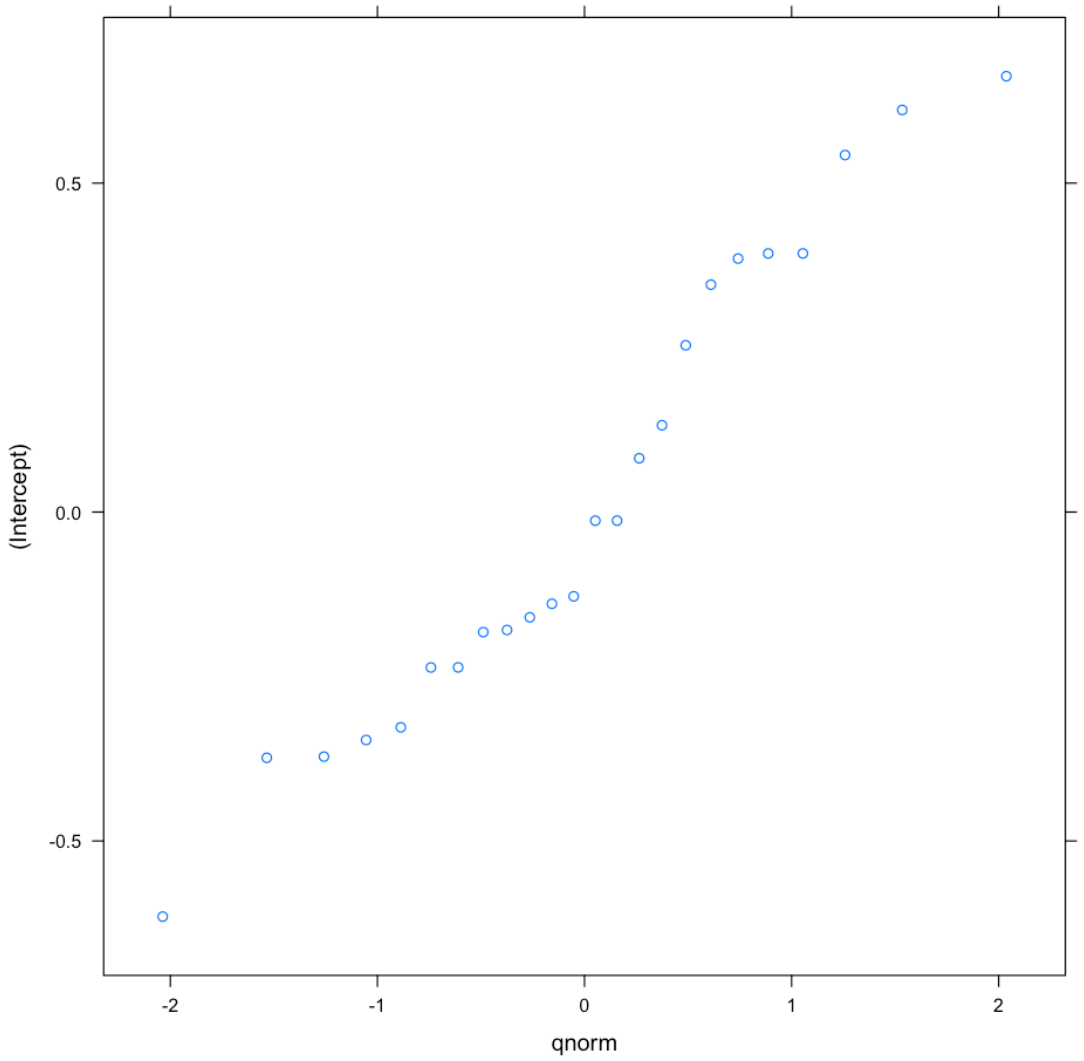
parameter <chr>	estimate <chr>	std.error <chr>	p.value <chr>	signif <chr>	2.5 <chr>	97.5 <chr>
(intercept)	8.63	0.192	0	<0.0001	8.26	9.01
total chlorine	-1.31	0.132	4.09e-23	<0.0001	-1.57	-1.05
pH	0.398	0.171	0.0203	<0.05	0.062	0.734
temp C	0.345	0.0971	0.000376	<0.001	0.155	0.536
pH:free chlorine	0.385	0.23	0.0942		-0.0659	0.836
total chlorine :temp C	-0.237	0.117	0.0419	<0.05	-0.466	-0.00874
var (intercept)	0.261	NA	NA	NA	NA	NA
var Observation	0.582	NA	NA	NA	NA	NA

<table style="text-align:center"><tr><td colspan="8" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"></td><td>parameter</td><td>estimate</td><td>std.error</td><td>p.value</td><td>signif</td><td>2.5</td><td>97.5</td></tr>
<tr><td colspan="8" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left">1</td><td>(intercept)</td><td>8.63</td><td>0.192</td><td>0</td><td>< 0.0001</td><td>8.26</td><td>9.01</td></tr>
<tr><td style="text-align:left">2</td><td>total chlorine</td><td>-1.31</td><td>0.132</td><td>4.09e-23</td><td>< 0.0001</td><td>-1.57</td><td>-1.05</td></tr>
<tr><td style="text-align:left">3</td><td>pH</td><td>0.398</td><td>0.171</td><td>0.0203</td><td>< 0.05</td><td>0.062</td><td>0.734</td></tr>
<tr><td style="text-align:left">4</td><td>temp C</td><td>0.345</td><td>0.0971</td><td>0.000376</td><td>< 0.001</td><td>0.155</td><td>0.536</td></tr>
<tr><td style="text-align:left">5</td><td>pH:free chlorine</td><td>0.385</td><td>0.23</td><td>0.0942</td><td></td><td>-0.0659</td><td>0.836</td></tr>
<tr><td style="text-align:left">6</td><td>total chlorine :temp C</td><td>-0.237</td><td>0.117</td><td>0.0419</td><td>< 0.05</td><td>-0.466</td><td>-0.00874</td></tr>
<tr><td style="text-align:left">7</td><td>var (intercept)</td><td>0.261</td><td>NA</td><td>NA</td><td></td><td>NA</td><td>NA</td></tr>
<tr><td style="text-align:left">8</td><td>var Observation</td><td>0.582</td><td>NA</td><td>NA</td><td></td><td>NA</td><td>NA</td></tr>
<tr><td colspan="8" style="border-bottom: 1px solid black"></td></tr></table>

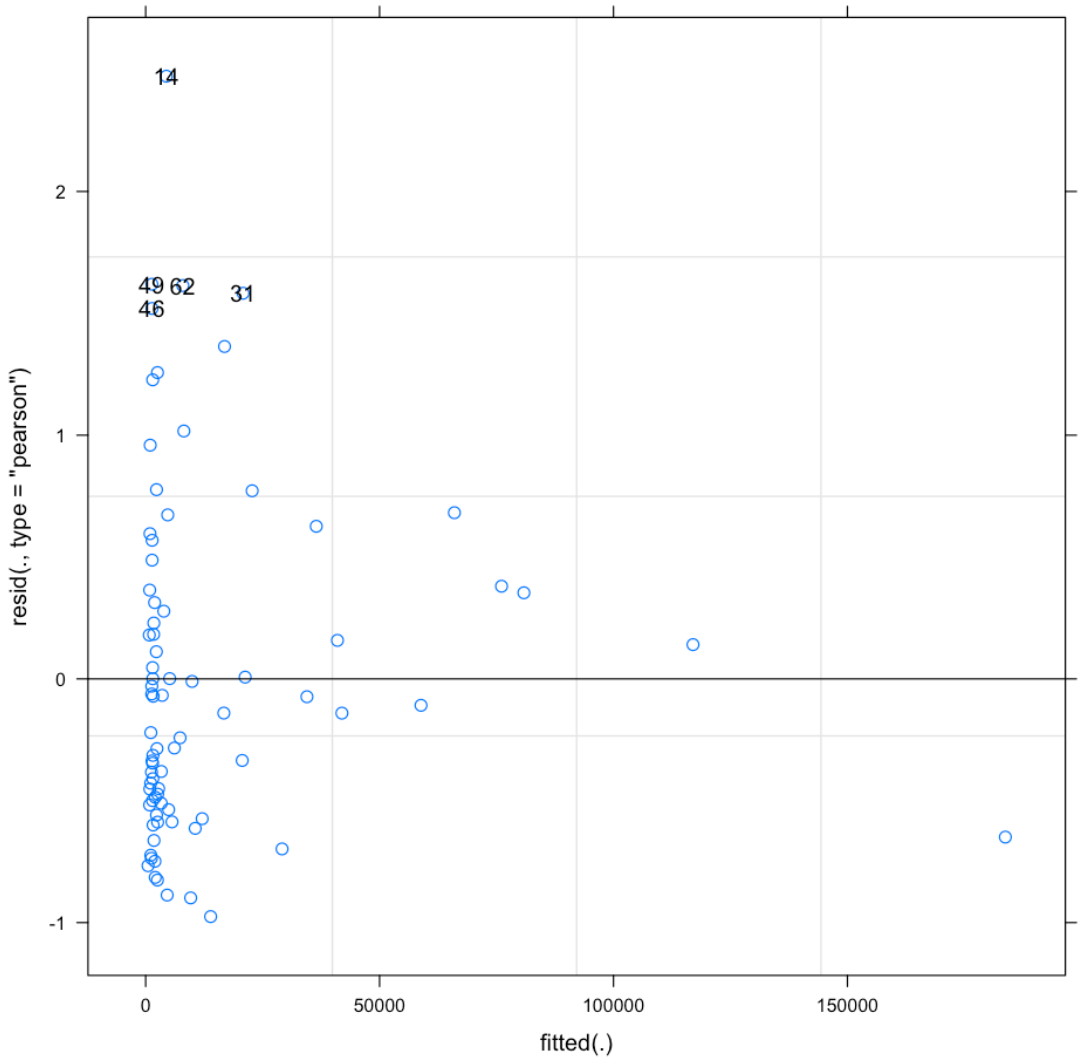
0.6.1 C. Model validation (Zuur 2015)

[49]:
```
options(repr.plot.width =8, repr.plot.height = 8) #for plotting size in jupyter
plot(ranef(glmm_ICC))
```

$location_code

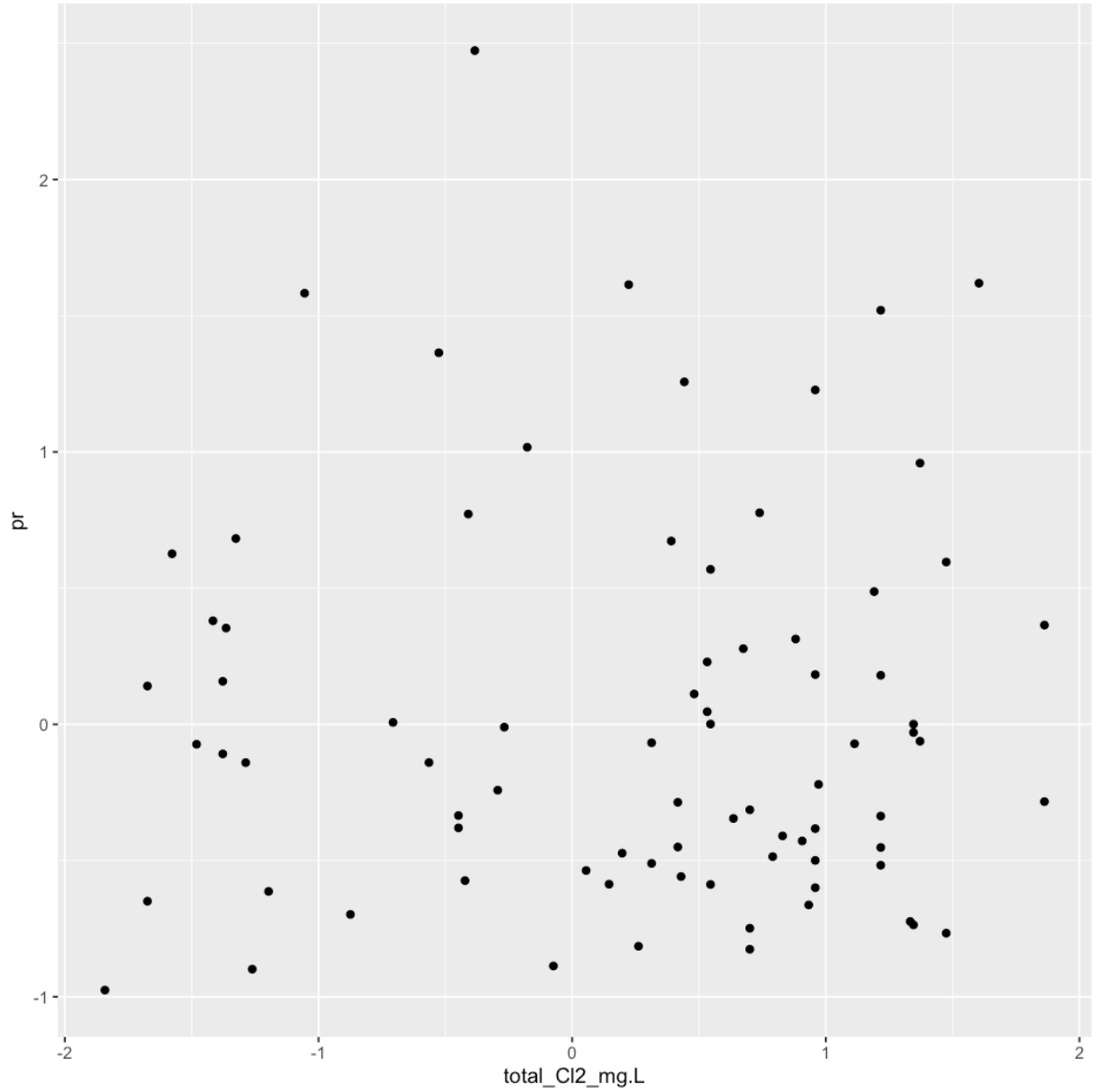


[50]:
```
options(repr.plot.width =8, repr.plot.height = 8) #for plotting size in jupyter
plot(glmm_ICC,id=0.05,idLabels=~.obs)
```

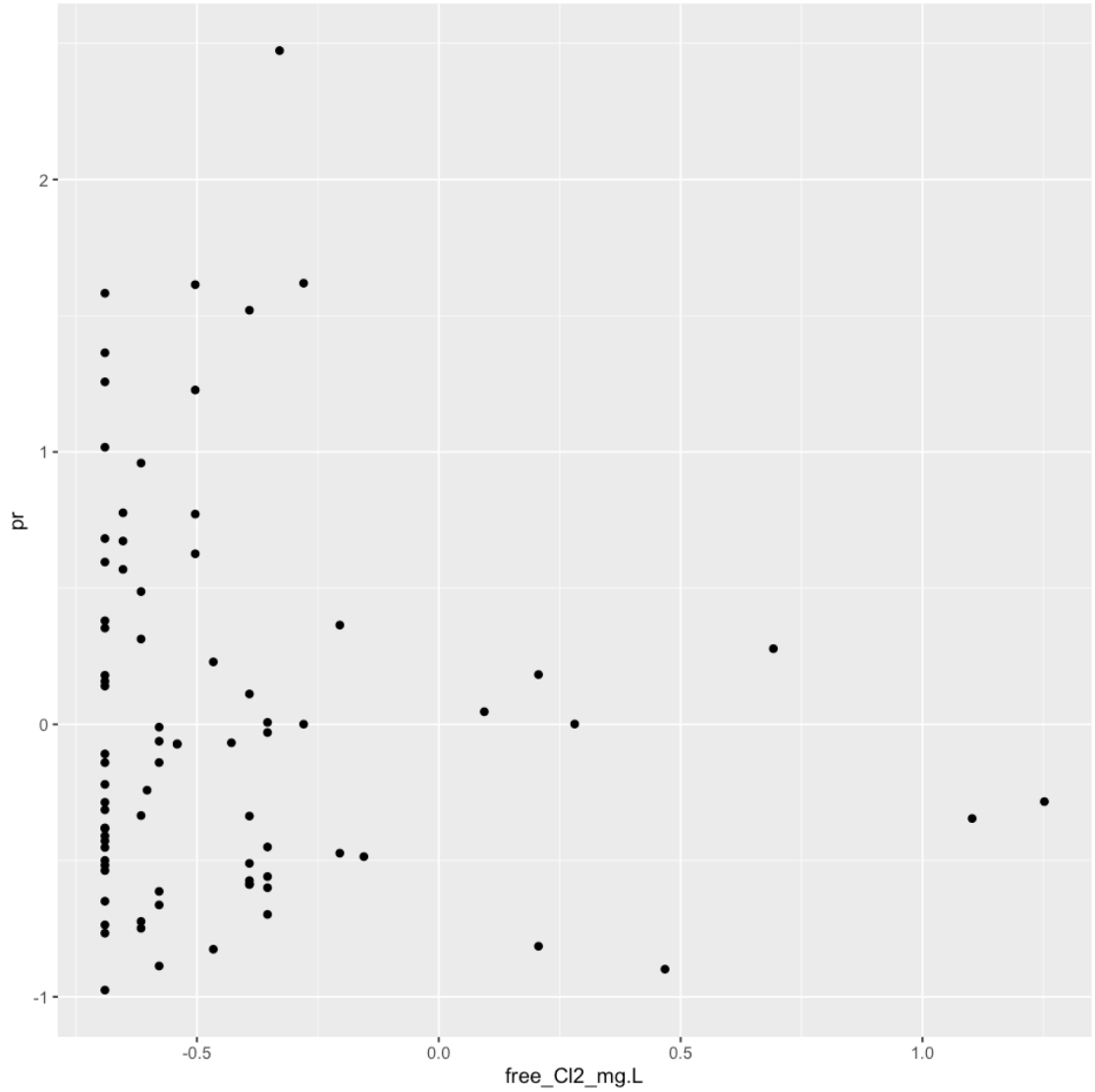


This plot looks similar to Zuur (A beginner's guide to glmm in R) pg 194 figure 6.14A (but more sparse)

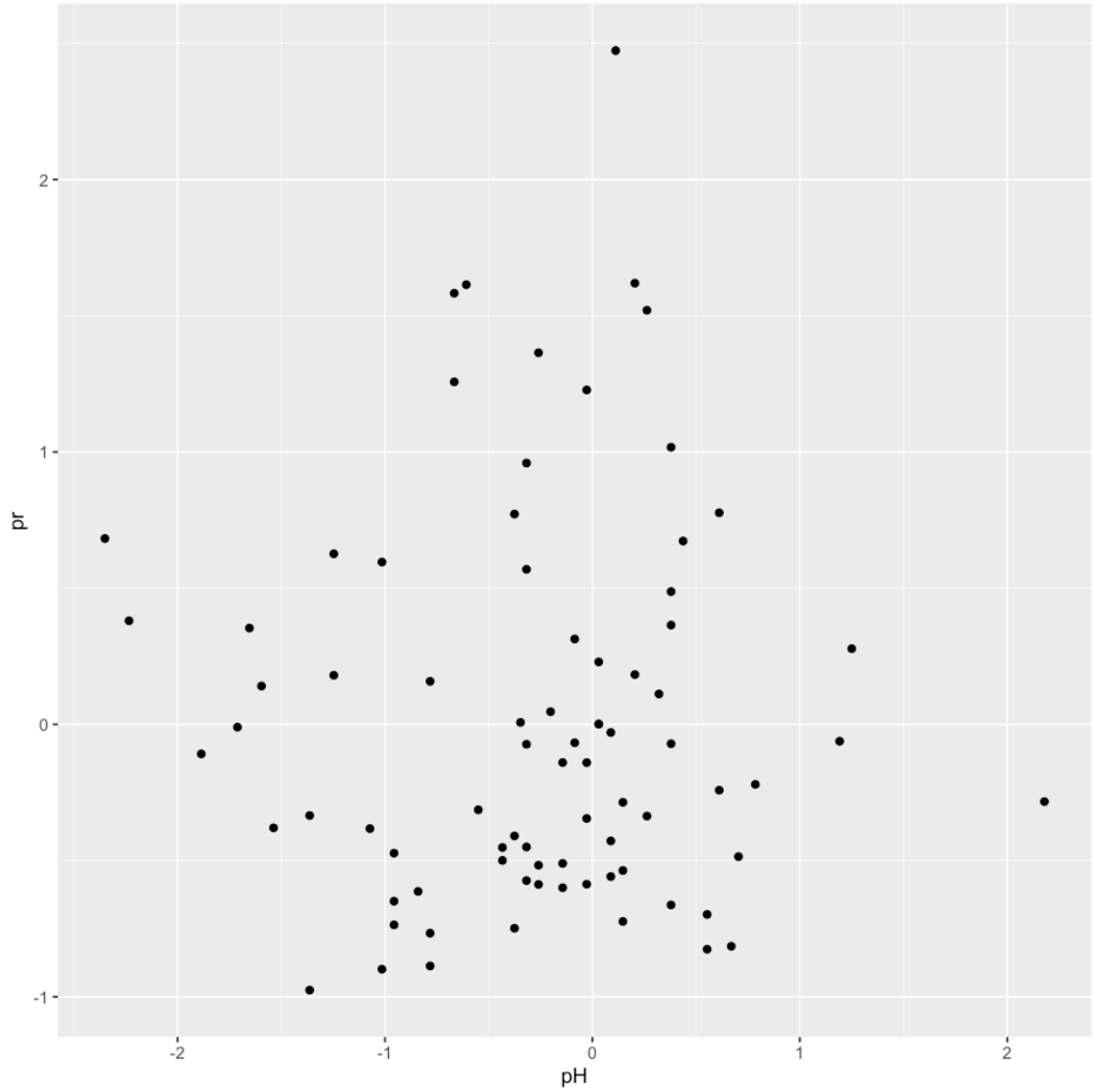
[51]:
```
pr<-residuals(glmm_ICC, "pearson")

ggplot(data=CAM_ICC,aes( x=total_Cl2_mg.L,y=pr) )+geom_point()
```



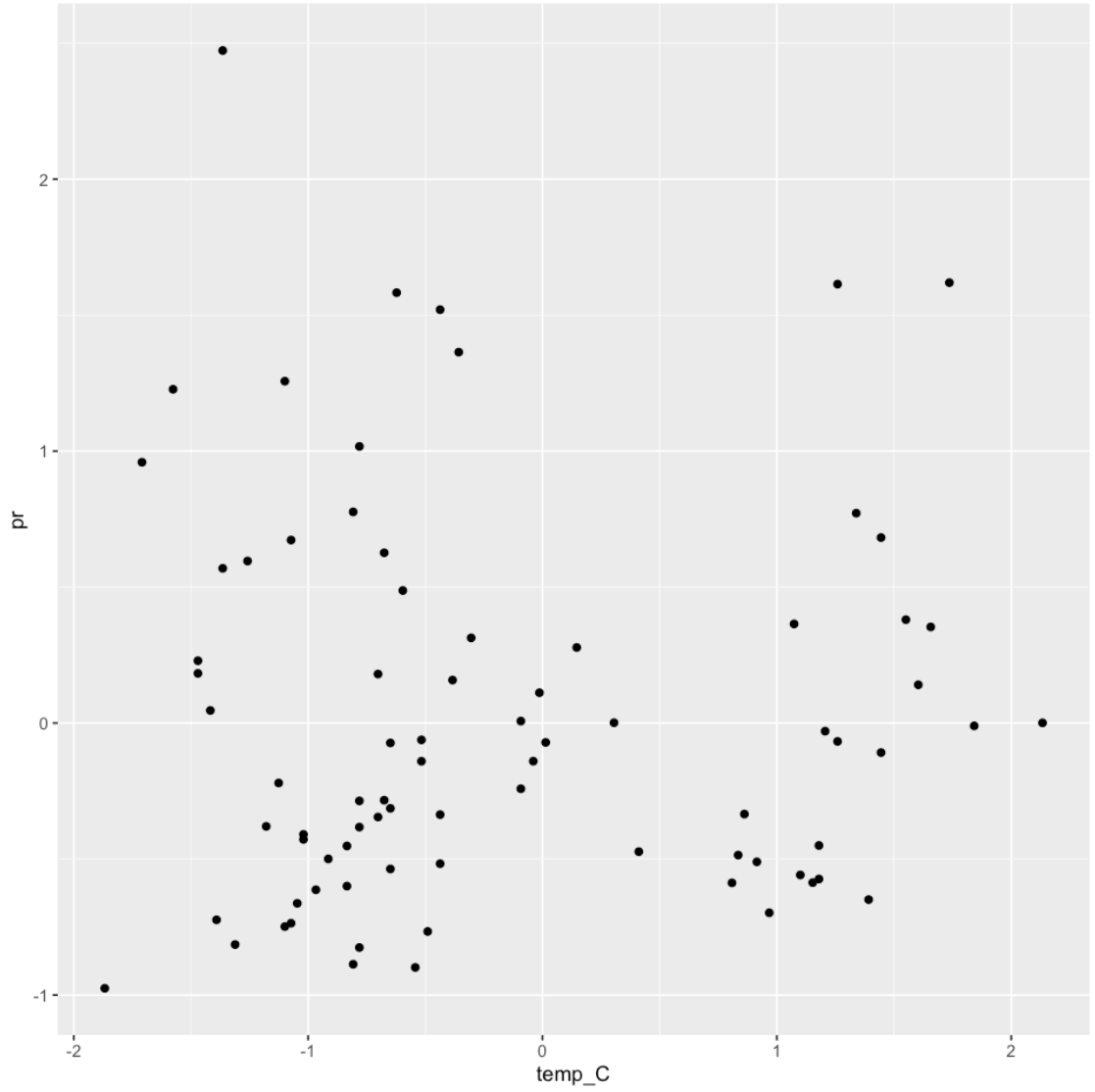
[52]:
```
# total_Cl2_mg.L+free_Cl2_mg.L+pH+temp_C+free_Cl2_mg.L:pH  +total_Cl2_mg.L:pH +␣
↪free_Cl2_mg.L:temp_C  +total_Cl2_mg.L:temp_C

ggplot(data=CAM_ICC,aes( x=free_Cl2_mg.L,y=pr) )+geom_point()
```



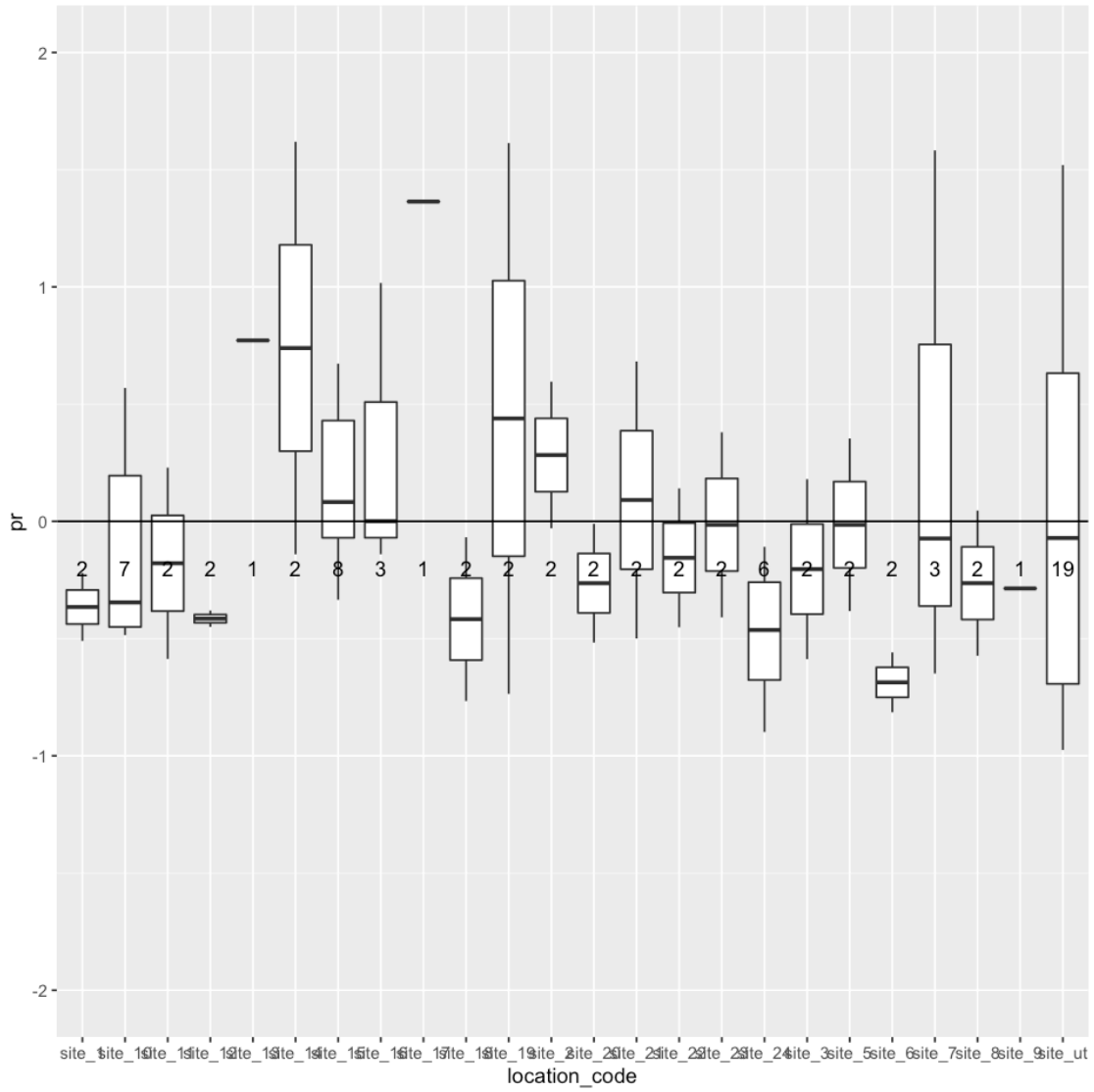
[53]:
```
# total_Cl2_mg.L+free_Cl2_mg.L+pH+temp_C+free_Cl2_mg.L:pH  +total_Cl2_mg.L:pH +␣
↪free_Cl2_mg.L:temp_C  +total_Cl2_mg.L:temp_C

ggplot(data=CAM_ICC,aes( x=pH,y=pr) )+geom_point()
```



[54]:
```
# total_Cl2_mg.L+free_Cl2_mg.L+pH+temp_C+free_Cl2_mg.L:pH  +total_Cl2_mg.L:pH +␣
↪free_Cl2_mg.L:temp_C  +total_Cl2_mg.L:temp_C

ggplot(data=CAM_ICC,aes( x=temp_C,y=pr) )+geom_point()
```

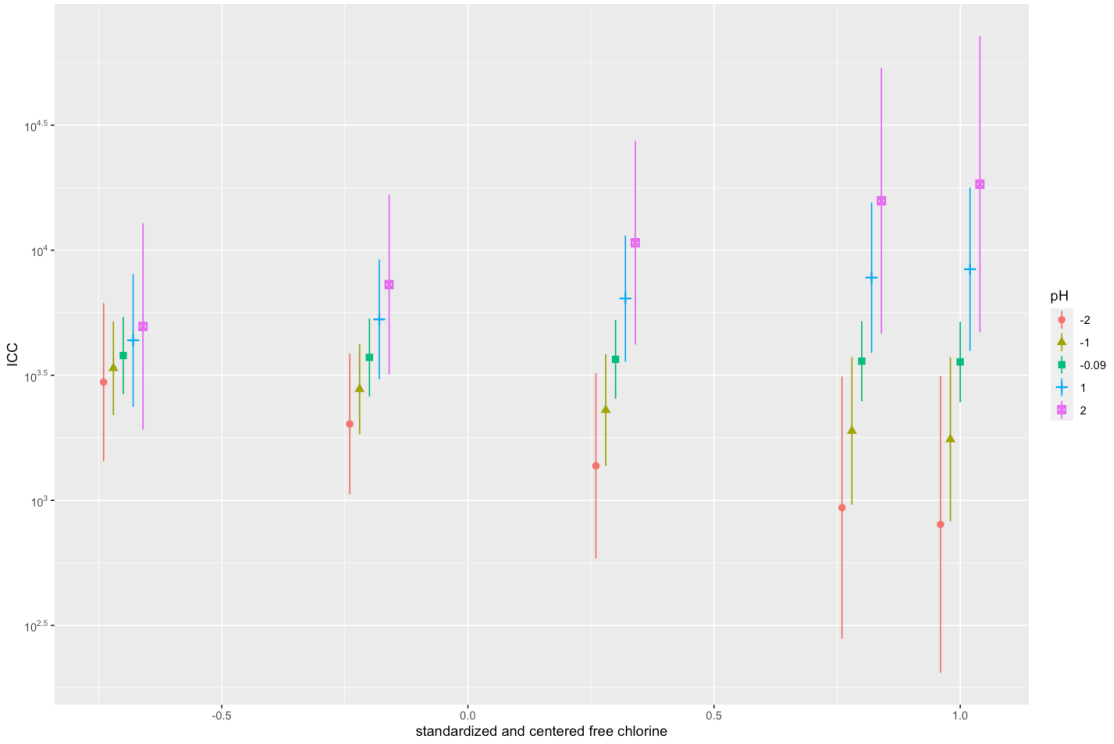


[55]:
```
ggplot(data=CAM_ICC,aes( x=location_code,y=pr) )+geom_boxplot()+
geom_hline(yintercept=0)+ylim(-2,2)+stat_summary(fun.data = give.n, geom =␣
↪"text", position = position_dodge(width = 0.75))
```

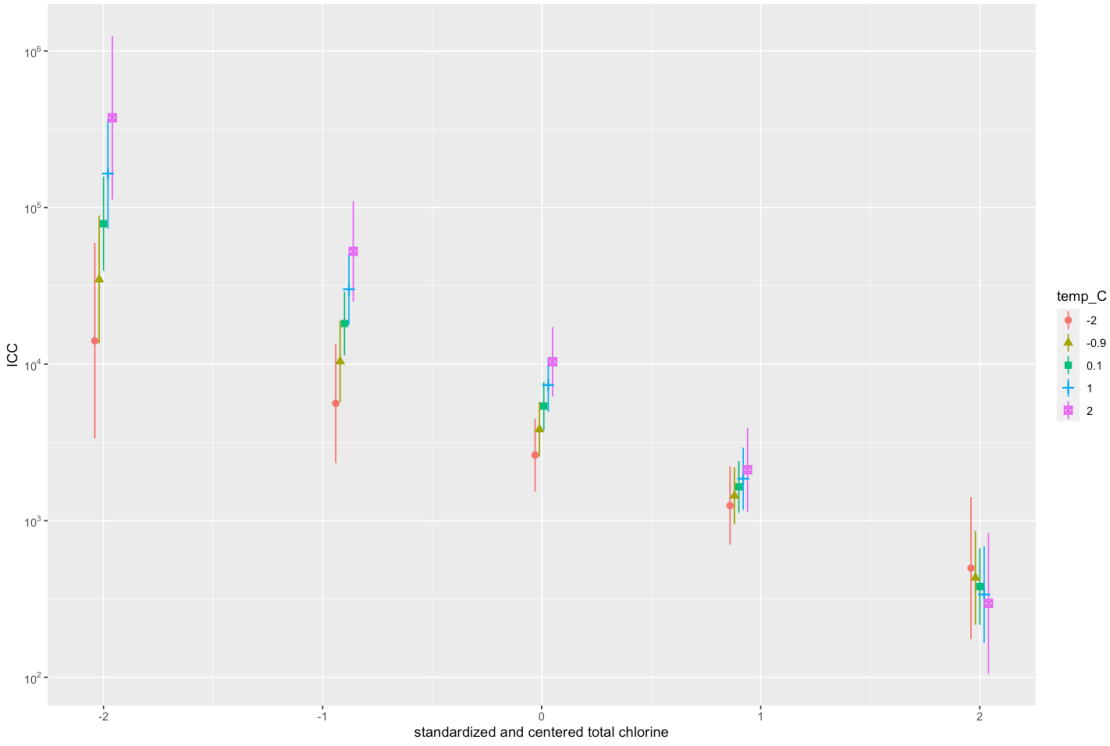


[56]:
```
options(repr.plot.width =12, repr.plot.height = 8) #for plotting size in jupyter

e<-allEffects(glmm_ICC)
e.df1 <- as.data.frame(e)[1]
e.df1 <- as.data.frame(e.df1)
colnames(e.df1)<-c("pH","free_Cl2_mg.L","fit","se","lower","upper")
e.df1$pH<-as.factor(e.df1$pH)

g <- ggplot(e.df1,aes(x=free_Cl2_mg.
 ↪L,y=fit,color=pH,shape=pH,ymin=lower,ymax=upper)) +
    geom_pointrange(position=position_dodge(width=.1)) +
    scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
                  labels = trans_format("log10", math_format(10^.x))) +
```

```
    xlab("standardized and centered free chlorine") + ylab("ICC")
plot(g)
```



[57]:
```
options(repr.plot.width =12, repr.plot.height = 8) #for plotting size in jupyter

e<-allEffects(glmm_ICC)
e.df2 <- as.data.frame(e)[2]
e.df2 <- as.data.frame(e.df2)
colnames(e.df2)<-c("total_Cl2_mg.L","temp_C","fit","se","lower","upper")
e.df2$temp_C<-as.factor(e.df2$temp_C)

g <- ggplot(e.df2,aes(x=total_Cl2_mg.
 ↪L,y=fit,color=temp_C,shape=temp_C,ymin=lower,ymax=upper)) +
    geom_pointrange(position=position_dodge(width=.1)) +
    scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
                  labels = trans_format("log10", math_format(10^.x))) +
    xlab("standardized and centered total chlorine") + ylab("ICC")
plot(g)
```



0.7 Figures 3 & S3

model visualization

predictions

[58]:
```
#predictions

#using the example here https://www.rdocumentation.org/packages/lme4/versions/1.
 ↪1-23/topics/predict.merMod
# model visualization total chlorine on the x axis where temp, pH and free␣
 ↪chlorine are fixed
options(repr.plot.width =12, repr.plot.height = 8) #for plotting size in jupyter
l<-length(unique(CAM_ICC$location_code))
obs.points<-1:l
df.predicted <- data.frame(location_code = NA, x = obs.points,
                total_Cl2_mg.L = seq(min(CAM_ICC$total_Cl2_mg.L),␣
 ↪max(CAM_ICC$total_Cl2_mg.L),length =l),
                temp_C = mean(CAM_ICC$temp_C),
                free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                pH = mean(CAM_ICC$pH))
predict.fun <- function(my.lmm) {
    predict(my.lmm, newdata = df.predicted, re.form = NA, type="response") } #␣
 ↪This is predict.merMod
```

```
#     make_predictions(glmm_ICC,total_Cl2_mg.L, interval=FALSE, )}
#
df.predicted$ml.value <- predict.fun(glmm_ICC)

# Make predictions in 100 bootstraps of the LMM. Use these to get confidence␣
 ↪intervals.
glmm.boots <- bootMer(glmm_ICC, predict.fun, nsim = 100) #this takes awhile!
df.predicted <- cbind(df.predicted, confint(glmm.boots))

quant<-as.data.frame(quantile(CAM_ICC$temp_C))[,1]
code<-factor( c("0%","25%","50%", "75%","100%"),levels=c("0%","25%","50%",␣
 ↪"75%","100%"))
num=100
obs.points<-1:num
quant
df.p1 <- data.frame(location_code = NA, x = obs.points,
                    total_Cl2_mg.L = seq(min(CAM_ICC$total_Cl2_mg.L),␣
 ↪max(CAM_ICC$total_Cl2_mg.L),length =num),
                    temp_C = quant[1],
                    free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                    pH = mean(CAM_ICC$pH))

df.p2 <- data.frame(location_code = NA, x = obs.points,
                    total_Cl2_mg.L = seq(min(CAM_ICC$total_Cl2_mg.L),␣
 ↪max(CAM_ICC$total_Cl2_mg.L),length =num),
                    temp_C = quant[2],
                    free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                    pH = mean(CAM_ICC$pH))

df.p3 <- data.frame(location_code = NA, x = obs.points,
                    total_Cl2_mg.L = seq(min(CAM_ICC$total_Cl2_mg.L),␣
 ↪max(CAM_ICC$total_Cl2_mg.L),length =num),
                    temp_C = quant[3],
                    free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                    pH = mean(CAM_ICC$pH))

df.p4 <- data.frame(location_code = NA, x = obs.points,
                    total_Cl2_mg.L = seq(min(CAM_ICC$total_Cl2_mg.L),␣
 ↪max(CAM_ICC$total_Cl2_mg.L),length =num),
                    temp_C = quant[4],
                    free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                    pH = mean(CAM_ICC$pH))

df.p5 <- data.frame(location_code = NA, x = obs.points,
                    total_Cl2_mg.L = seq(min(CAM_ICC$total_Cl2_mg.L),␣
 ↪max(CAM_ICC$total_Cl2_mg.L),length =num),
```

```
                temp_C = quant[5],
                free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                pH = mean(CAM_ICC$pH))



df.p1$pred<-predict(glmm_ICC, newdata = df.p1, re.form = NA, type="response")
df.p1$code<-code[1]
df.p2$pred<-predict(glmm_ICC, newdata = df.p2, re.form = NA, type="response")
df.p2$code<-code[2]
df.p3$pred<-predict(glmm_ICC, newdata = df.p3, re.form = NA, type="response")
df.p3$code<-code[3]
df.p4$pred<-predict(glmm_ICC, newdata = df.p4, re.form = NA, type="response")
df.p4$code<-code[4]
df.p5$pred<-predict(glmm_ICC, newdata = df.p5, re.form = NA, type="response")
df.p5$code<-code[5]

df.predicted.quant<-rbind(df.p1,df.p2,df.p3,df.p4,df.p5)

df.predicted.a=df.predicted
df.predicted.quant.a=df.predicted.quant
```

1. -1.86783637074696 2. -0.980099539168121 3. -0.529606221650499 4. 0.874872944727968 5. 2.13360427308603

[59]:
```
# total_Cl2_mg.L:temp_C interaction demonstration with total chlorine on the x␣
 ↪axis where  pH and free chlorine are fixed
quant<-as.data.frame(quantile(CAM_ICC$temp_C))[,1]
code<-factor( c("0%","25%","50%", "75%","100%"),levels=c("0%","25%","50%",␣
 ↪"75%","100%"))
num=100
obs.points<-1:num
quant
df.p1 <- data.frame(location_code = NA, x = obs.points,
                total_Cl2_mg.L = seq(min(CAM_ICC$total_Cl2_mg.L),␣
 ↪max(CAM_ICC$total_Cl2_mg.L),length =num),
                temp_C = quant[1],
                free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                pH = mean(CAM_ICC$pH))

df.p2 <- data.frame(location_code = NA, x = obs.points,
                total_Cl2_mg.L = seq(min(CAM_ICC$total_Cl2_mg.L),␣
 ↪max(CAM_ICC$total_Cl2_mg.L),length =num),
                temp_C = quant[2],
                free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                pH = mean(CAM_ICC$pH))
```

```
df.p3 <- data.frame(location_code = NA, x = obs.points,
                    total_Cl2_mg.L = seq(min(CAM_ICC$total_Cl2_mg.L),␣
 ↪max(CAM_ICC$total_Cl2_mg.L),length =num),
                    temp_C = quant[3],
                    free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                    pH = mean(CAM_ICC$pH))

df.p4 <- data.frame(location_code = NA, x = obs.points,
                    total_Cl2_mg.L = seq(min(CAM_ICC$total_Cl2_mg.L),␣
 ↪max(CAM_ICC$total_Cl2_mg.L),length =num),
                    temp_C = quant[4],
                    free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                    pH = mean(CAM_ICC$pH))

df.p5 <- data.frame(location_code = NA, x = obs.points,
                    total_Cl2_mg.L = seq(min(CAM_ICC$total_Cl2_mg.L),␣
 ↪max(CAM_ICC$total_Cl2_mg.L),length =num),
                    temp_C = quant[5],
                    free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                    pH = mean(CAM_ICC$pH))



df.p1$pred<-predict(glmm_ICC, newdata = df.p1, re.form = NA, type="response")
df.p1$code<-code[1]
df.p2$pred<-predict(glmm_ICC, newdata = df.p2, re.form = NA, type="response")
df.p2$code<-code[2]
df.p3$pred<-predict(glmm_ICC, newdata = df.p3, re.form = NA, type="response")
df.p3$code<-code[3]
df.p4$pred<-predict(glmm_ICC, newdata = df.p4, re.form = NA, type="response")
df.p4$code<-code[4]
df.p5$pred<-predict(glmm_ICC, newdata = df.p5, re.form = NA, type="response")
df.p5$code<-code[5]

df.predicted<-rbind(df.p1,df.p2,df.p3,df.p4,df.p5)

df.predicted.b=df.predicted
```

1. -1.86783637074696 2. -0.980099539168121 3. -0.529606221650499 4. 0.874872944727968 5. 2.13360427308603

[60]:
```
# total_Cl2_mg.L:temp_C interaction demonstration with temp on the x axis where␣
 ↪ pH and free chlorine are fixed
quant<-as.data.frame(quantile(CAM_ICC$total_Cl2_mg.L))[,1]
quant
code<-factor( c("0%","25%","50%", "75%","100%"),levels=c("0%","25%","50%",␣
 ↪"75%","100%"))
```

```
num=100
obs.points<-1:num

df.p1 <- data.frame(location_code = NA, x = obs.points,
                temp_C = seq(min(CAM_ICC$temp_C), max(CAM_ICC$temp_C),length␣
 ↪=num),
                total_Cl2_mg.L = quant[1],
                free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                pH = mean(CAM_ICC$pH))

df.p2 <- data.frame(location_code = NA, x = obs.points,
                temp_C = seq(min(CAM_ICC$temp_C), max(CAM_ICC$temp_C),length␣
 ↪=num),
                total_Cl2_mg.L= quant[2],
                free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                pH = mean(CAM_ICC$pH))

df.p3 <- data.frame(location_code = NA, x = obs.points,
                temp_C = seq(min(CAM_ICC$temp_C), max(CAM_ICC$temp_C),length␣
 ↪=num),
                total_Cl2_mg.L = quant[3],
                free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                pH = mean(CAM_ICC$pH))

df.p4 <- data.frame(location_code = NA, x = obs.points,
                temp_C = seq(min(CAM_ICC$temp_C), max(CAM_ICC$temp_C),length␣
 ↪=num),
                total_Cl2_mg.L = quant[4],
                free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                pH = mean(CAM_ICC$pH))

df.p5 <- data.frame(location_code = NA, x = obs.points,
                temp_C = seq(min(CAM_ICC$temp_C), max(CAM_ICC$temp_C),length␣
 ↪=num),
                total_Cl2_mg.L = quant[5],
                free_Cl2_mg.L= mean(CAM_ICC$free_Cl2_mg.L),
                pH = mean(CAM_ICC$pH))



df.p1$pred<-predict(glmm_ICC, newdata = df.p1, re.form = NA, type="response")
df.p1$code<-code[1]
df.p2$pred<-predict(glmm_ICC, newdata = df.p2, re.form = NA, type="response")
df.p2$code<-code[2]
df.p3$pred<-predict(glmm_ICC, newdata = df.p3, re.form = NA, type="response")
df.p3$code<-code[3]
df.p4$pred<-predict(glmm_ICC, newdata = df.p4, re.form = NA, type="response")
```

```
df.p4$code<-code[4]
df.p5$pred<-predict(glmm_ICC, newdata = df.p5, re.form = NA, type="response")
df.p5$code<-code[5]

df.predicted<-rbind(df.p1,df.p2,df.p3,df.p4,df.p5)
df.predicted.c=df.predicted
```

1. -1.84206746826039 2. -0.428894802833405 3. 0.506767464230122 4. 0.958466489709067 5. 1.86186454066696

[61]:
```
# free chlorine:pH interaction demonstration with pH on the x axis where  temp␣
 ↪and total chlorine are fixed
quant<-as.data.frame(quantile(CAM_ICC$free_Cl2_mg.L))[,1]
quant
code<-factor( c("0%","25%","50%", "75%","100%"),levels=c("0%","25%","50%",␣
 ↪"75%","100%"))
num=100
obs.points<-1:num

df.p1 <- data.frame(location_code = NA, x = obs.points,
                pH= seq(min(CAM_ICC$pH), max(CAM_ICC$pH),length =num),
                free_Cl2_mg.L = quant[1],
                total_Cl2_mg.L= mean(CAM_ICC$total_Cl2_mg.L),
                temp_C = mean(CAM_ICC$temp_C))

df.p2 <- data.frame(location_code = NA, x = obs.points,
                pH= seq(min(CAM_ICC$pH), max(CAM_ICC$pH),length =num),
                free_Cl2_mg.L = quant[2],
                total_Cl2_mg.L= mean(CAM_ICC$total_Cl2_mg.L),
                temp_C = mean(CAM_ICC$temp_C))

df.p3 <- data.frame(location_code = NA, x = obs.points,
                pH= seq(min(CAM_ICC$pH), max(CAM_ICC$pH),length =num),
                free_Cl2_mg.L = quant[3],
                total_Cl2_mg.L= mean(CAM_ICC$total_Cl2_mg.L),
                temp_C = mean(CAM_ICC$temp_C))

df.p4 <- data.frame(location_code = NA, x = obs.points,
                pH= seq(min(CAM_ICC$pH), max(CAM_ICC$pH),length =num),
                free_Cl2_mg.L = quant[4],
                total_Cl2_mg.L= mean(CAM_ICC$total_Cl2_mg.L),
                temp_C = mean(CAM_ICC$temp_C))

df.p5 <- data.frame(location_code = NA, x = obs.points,
                pH= seq(min(CAM_ICC$pH), max(CAM_ICC$pH),length =num),
                free_Cl2_mg.L = quant[5],
                total_Cl2_mg.L= mean(CAM_ICC$total_Cl2_mg.L),
```

```
                temp_C = mean(CAM_ICC$temp_C))



df.p1$pred<-predict(glmm_ICC, newdata = df.p1, re.form = NA, type="response")
df.p1$code<-code[1]
df.p2$pred<-predict(glmm_ICC, newdata = df.p2, re.form = NA, type="response")
df.p2$code<-code[2]
df.p3$pred<-predict(glmm_ICC, newdata = df.p3, re.form = NA, type="response")
df.p3$code<-code[3]
df.p4$pred<-predict(glmm_ICC, newdata = df.p4, re.form = NA, type="response")
df.p4$code<-code[4]
df.p5$pred<-predict(glmm_ICC, newdata = df.p5, re.form = NA, type="response")
df.p5$code<-code[5]

df.predicted<-rbind(df.p1,df.p2,df.p3,df.p4,df.p5)
df.predicted.d=df.predicted
```

1. -0.690214788809883 2. -0.690214788809883 3. -0.5781606540922 4. -0.382065918336256 5. 1.25205687962995

[62]:
```
# free chlorine:pH interaction demonstration with free chlorine on the x axis␣
 ↪where  temp and total chlorine are fixed
quant<-as.data.frame(quantile(CAM_ICC$pH))[,1]
quant
code<-factor( c("0%","25%","50%", "75%","100%"),levels=c("0%","25%","50%",␣
 ↪"75%","100%"))
num=100
obs.points<-1:num

df.p1 <- data.frame(location_code = NA, x = obs.points,
                free_Cl2_mg.L= seq(min(CAM_ICC$free_Cl2_mg.L),␣
 ↪max(CAM_ICC$free_Cl2_mg.L),length =num),
                pH = quant[1],
                total_Cl2_mg.L= mean(CAM_ICC$total_Cl2_mg.L),
                temp_C = mean(CAM_ICC$temp_C))

df.p2 <- data.frame(location_code = NA, x = obs.points,
                free_Cl2_mg.L= seq(min(CAM_ICC$free_Cl2_mg.L),␣
 ↪max(CAM_ICC$free_Cl2_mg.L),length =num),
                pH = quant[2],
                total_Cl2_mg.L= mean(CAM_ICC$total_Cl2_mg.L),
                temp_C = mean(CAM_ICC$temp_C))

df.p3 <-data.frame(location_code = NA, x = obs.points,
                free_Cl2_mg.L= seq(min(CAM_ICC$free_Cl2_mg.L),␣
 ↪max(CAM_ICC$free_Cl2_mg.L),length =num),
```

```
                pH = quant[3],
                total_Cl2_mg.L= mean(CAM_ICC$total_Cl2_mg.L),
                temp_C = mean(CAM_ICC$temp_C))

df.p4 <- data.frame(location_code = NA, x = obs.points,
                free_Cl2_mg.L= seq(min(CAM_ICC$free_Cl2_mg.L),␣
 ↪max(CAM_ICC$free_Cl2_mg.L),length =num),
                pH = quant[4],
                total_Cl2_mg.L= mean(CAM_ICC$total_Cl2_mg.L),
                temp_C = mean(CAM_ICC$temp_C))

df.p5 <- data.frame(location_code = NA, x = obs.points,
                free_Cl2_mg.L= seq(min(CAM_ICC$free_Cl2_mg.L),␣
 ↪max(CAM_ICC$free_Cl2_mg.L),length =num),
                pH = quant[5],
                total_Cl2_mg.L= mean(CAM_ICC$total_Cl2_mg.L),
                temp_C = mean(CAM_ICC$temp_C))



df.p1$pred<-predict(glmm_ICC, newdata = df.p1, re.form = NA, type="response")
df.p1$code<-code[1]
df.p2$pred<-predict(glmm_ICC, newdata = df.p2, re.form = NA, type="response")
df.p2$code<-code[2]
df.p3$pred<-predict(glmm_ICC, newdata = df.p3, re.form = NA, type="response")
df.p3$code<-code[3]
df.p4$pred<-predict(glmm_ICC, newdata = df.p4, re.form = NA, type="response")
df.p4$code<-code[4]
df.p5$pred<-predict(glmm_ICC, newdata = df.p5, re.form = NA, type="response")
df.p5$code<-code[5]

df.predicted<-rbind(df.p1,df.p2,df.p3,df.p4,df.p5)

df.predicted.e=df.predicted
```

1. -2.35002933144051 2. -0.782215592508988 3. -0.172510249591176 4. 0.204926391262706 5. 2.1792103588061

plot code

[63]:
```
w2=6
h2=4
l=0.5
options(repr.plot.width = w2, repr.plot.height = h2) #for plotting size in␣
 ↪jupyter
theme_set(theme_classic(base_size=7, base_family="Arial"))#, base_line_size= 1))
```

[64]:
```
# Plot the ML prediction and its confidence intervals
a<-ggplot(data=CAM_ICC) +
geom_line(data=df.predicted.a, aes(x=total_Cl2_mg.L, y=ml.value),␣
↪linetype="dashed",lwd=1) +
 geom_ribbon(data=df.predicted.a, aes(x=total_Cl2_mg.L, ymin=`2.5 %`,
                                      ymax=`97.5 %`),
                  fill="gray", alpha=0.5, inherit.aes = FALSE)+
   geom_line(data=df.predicted.quant.a, aes(x=total_Cl2_mg.L, y=pred,␣
↪color=code), lwd=1) +
   ylim(-1, max(CAM_ICC$SGPI+1000))+
   ylab("intact cell count\n(cells per mL)")+
   xlab("total chlorine concentration\n(standardized and centered)")+
   labs(color = " quantiles of temperature\n(standardized and centered)")+
   scale_color_viridis_d()+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
↪element_text(hjust = 0.5),legend.position="right")
```

[65]:
```
# Plot the predictions based on
b<-ggplot(data=CAM_ICC)+ geom_line(data=df.predicted.b, aes(x=total_Cl2_mg.L,␣
↪y=pred, color=code), lwd=1) +
   ylim(-1, max(CAM_ICC$SGPI+1000))+
   ylab("intact cell count\n(cells per mL)")+
   xlab("total chlorine concentration\n(standardized and centered)")+
   labs(color = " quantiles\nof temperature\n(standardized\nand centered)")+
   scale_color_viridis_d()+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
↪element_text(hjust = 0.5),legend.position="right")
```

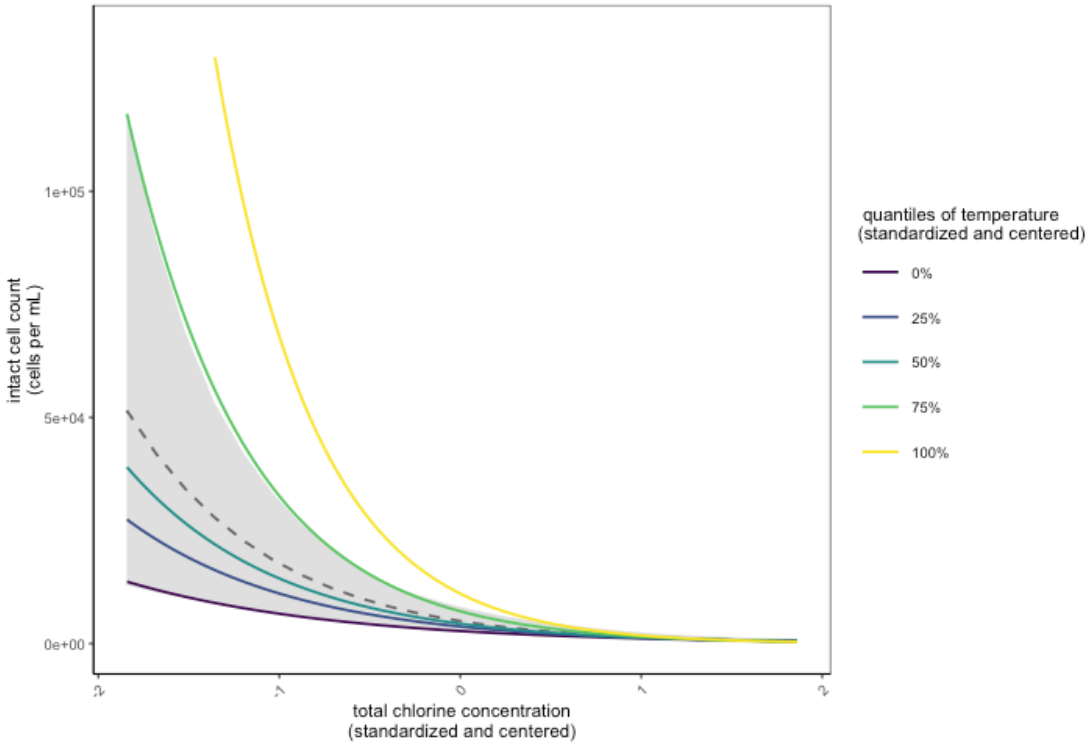
[66]:
```
# Plot the predictions based on
c<-ggplot(data=CAM_ICC) + geom_line(data=df.predicted.c, aes(x=temp_C, y=pred,␣
↪color=code), lwd=1) +
   ylim(-1, max(CAM_ICC$SGPI+1000))+
   ylab("intact cell count\n(cells per mL)")+
   xlab("temperature\n(standardized and centered)")+
   labs(color = " quantiles\nof total chlorine\n(standardized\nand centered)")+
   scale_color_viridis_d()+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
↪element_text(hjust = 0.5),legend.position="right")
```

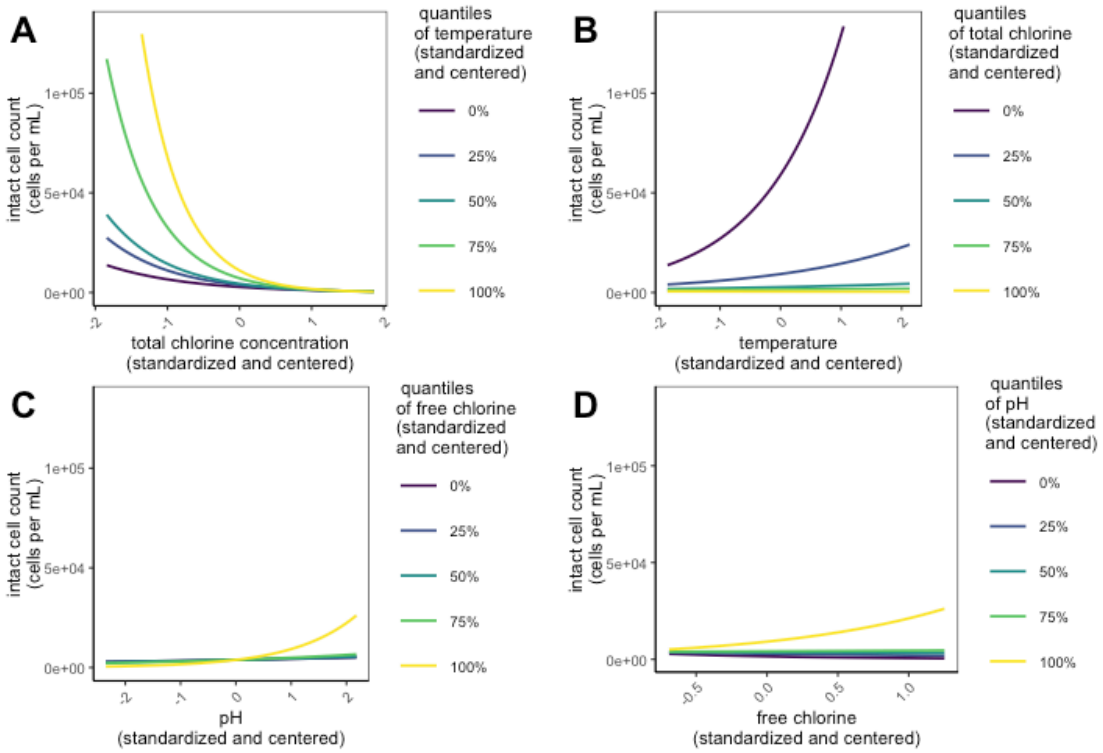
[67]:
```
# Plot the predictions based on
d<-ggplot(data=CAM_ICC)  + geom_line(data=df.predicted.d, aes(x=pH, y=pred,␣
 ↪color=code), lwd=1) +
    ylim(-1, max(CAM_ICC$SGPI+1000))+
    ylab("intact cell count\n(cells per mL)")+
    xlab("pH\n(standardized and centered)")+
    labs(color = " quantiles\nof free chlorine\n(standardized\nand centered)")+
    scale_color_viridis_d()+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
 ↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
 ↪element_text(hjust = 0.5),legend.position="right")
```

[68]:
```
# Plot the predictions based on
e<-ggplot(data=CAM_ICC)  + geom_line(data=df.predicted.e, aes(x=free_Cl2_mg.L,␣
 ↪y=pred, color=code), lwd=1) +
    ylim(-1, max(CAM_ICC$SGPI+1000))+
    ylab("intact cell count\n(cells per mL)")+
    xlab("free chlorine\n(standardized and centered)")+
    labs(color = " quantiles\nof pH\n(standardized\nand centered)")+
    scale_color_viridis_d()+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
 ↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
 ↪element_text(hjust = 0.5),legend.position="right")
```

0.7.1 plot

[69]:
```
#Figure 3
a
ggsave(paste(path_fig,"Figure_3.pdf",sep=""), device= cairo_pdf, units='mm',␣
 ↪width=single.col_w, height=max_h/5)

#Figure S3
ggarrange(b,c,d,e,
          labels = c("A", "B","C","D"),
           ncol = 2, nrow = 2, common.legend=FALSE)
#           ncol = 1, nrow = 3, common.legend=TRUE, legend= NULL, align="v")


ggsave(paste(path_fig,"Figure_S3.pdf",sep=""), device= cairo_pdf, units='mm',␣
 ↪width=max_w, height=(max_h/2))
```


intact cell count
(cells per mL)
1e+05
5e+04
0e+00
-2
-1
0
1
2
total chlorine concentration
(standardized and centered)
quantiles of temperature
(standardized and centered)
0%
25%
50%
75%
100%


A
intact cell count
(cells per mL)
1e+05
5e+04
0e+00
-2
-1
0
1
2
total chlorine concentration
(standardized and centered)
quantiles
of temperature
(standardized
and centered)
0%
25%
50%
75%
100%
B
intact cell count
(cells per mL)
1e+05
5e+04
0e+00
-2
-1
0
1
2
temperature
(standardized and centered)
quantiles
of total chlorine
(standardized
and centered)
0%
25%
50%
75%
100%
C
intact cell count
(cells per mL)
1e+05
5e+04
0e+00
-2
-1
0
1
2
pH
(standardized and centered)
quantiles
of free chlorine
(standardized
and centered)
0%
25%
50%
75%
100%
D
intact cell count
(cells per mL)
1e+05
5e+04
0e+00
-0.5
0.0
0.5
1.0
free chlorine
(standardized and centered)
quantiles
of pH
(standardized
and centered)
0%
25%
50%
75%
100%

[70]:
```
#https://r-sig-mixed-models.r-project.narkive.com/4jQJ7hfz/
 ↪r-sig-me-residual-variance-or-dispersion-of-gamma-glmer
glmm_ICC
sigma(glmm_ICC) #this is an estimate of the coefficient of variation or 1/
 ↪sqrt(shape parameter)
tau<-(1/sigma(glmm_ICC))^2
tau
VarCorr(glmm_ICC)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: Gamma  ( log )
Formula:
SGPI ~ total_Cl2_mg.L + pH + temp_C + (1 | location_code) + free_Cl2_mg.L:pH +
    total_Cl2_mg.L:temp_C
   Data: CAM_ICC
      AIC       BIC    logLik  deviance  df.resid
1524.3504 1543.4066 -754.1752 1508.3504        72
Random effects:
 Groups        Name        Std.Dev.
 location_code (Intercept) 0.5107
 Residual                  0.7630
Number of obs: 80, groups:  location_code, 24
Fixed Effects:
          (Intercept)         total_Cl2_mg.L                     pH
               8.6349                -1.3103                 0.3979
               temp_C       pH:free_Cl2_mg.L  total_Cl2_mg.L:temp_C
               0.3453                 0.3853                -0.2374
```

0.763007947415628

1.71767848907359

```
 Groups        Name        Std.Dev.
 location_code (Intercept) 0.51065
 Residual                  0.76301
```

[71]:
```
#Using code from this site to test my assumption above: https://github.com/lme4/
 ↪lme4/issues/290

set.seed(101)
d <- expand.grid(block=LETTERS[1:26], rep=1:100, KEEP.OUT.ATTRS = FALSE)
d$x <- runif(nrow(d))  ## sd=1
reff_f <- rnorm(length(levels(d$block)),sd=1)
## need intercept large enough to avoid negative values
d$eta0 <- 4+3*d$x  ## fixed effects only
d$eta <- d$eta0+reff_f[d$block]
```

```
shapevec <- tau
res <- expand.grid(rep=1:10,shape=shapevec,est=NA) ## order matters
k <- 1
i <- 1
numOK <- 0
## for (i in seq_along(shapevec)) {
## cat(".")
for (j in 1:10) {
    dgl <- d
    dgl$mu <- exp(d$eta)
    dgl$y <- rgamma(nrow(d),scale=dgl$mu/2,shape=shapevec[i])
    ggl1 <- try(glmer(y ~ x + (1|block), data=dgl, family=Gamma(link="log")))
    if (!is(ggl1,"try-error")) numOK <- numOK+1
    ## res[k,"est"] <- 1/sigma(ggl1)^2
    k <- k+1
}

sigma(ggl1)#checks out
set.seed(30)
```

0.76263229812164

0.8 4. Rename

Rename columns for plotting

[72]:
```
utility_name<- "site ut"
SG_all$location_code <- gsub("site_ut", utility_name, SG_all$location_code)
SGPI_all$location_code <- gsub("site_ut", utility_name, SGPI_all$location_code)
ATP_long$location_code <- gsub("site_ut", utility_name, ATP_long$location_code)
fcm_all_long$location_code <- gsub("site_ut", utility_name,␣
 ↪fcm_all_long$location_code)

#call raw
raw_name<- "raw water"
SG_all$location_code <- gsub("raw_water", raw_name, SG_all$location_code)
SGPI_all$location_code <- gsub("raw_water", raw_name, SGPI_all$location_code)
ATP_long$location_code <- gsub("raw_water", raw_name, ATP_long$location_code)
fcm_all_long$location_code <- gsub("raw_water", raw_name,␣
 ↪fcm_all_long$location_code)

#call eff
eff_name<- "finished water"
SG_all$location_code <- gsub("finished_water", eff_name, SG_all$location_code)
SGPI_all$location_code <- gsub("finished_water", eff_name,␣
 ↪SGPI_all$location_code)
```

```
ATP_long$location_code <- gsub("finished_water", eff_name,␣
 ↪ATP_long$location_code)
fcm_all_long$location_code <- gsub("finished_water", eff_name,␣
 ↪fcm_all_long$location_code)


#paper
D_A<-"system A"
D_B<-"system B"
D_C<-"system C"
D_D<-"system D"
D_F<-"system F"
D_E<-"system E"

#call other descriptors
TCC_l<-"total cell count"
ICC_l<-"intact cell count"

ATPi_l<- "intracellular ATP"
ATPt_l<- "total ATP"

lc<- "sampling_site"
bl<- "sampling_location"
atp<-"ATP assay"
ccl<-"cell count"

#call other descriptors plotting!
TCC_lp<-"`total cell count`"
ICC_lp<-"`intact cell count`"

ATPi_lp<- "`intracellular ATP`"
ATPt_lp<- "`total ATP`"

lcp<- "`sampling_site`"
blp<- "`sampling_location`"
atpp<-"`ATP assay`"
cclp<-"`cell count`"

#quantification Limits & label BDL values where needed
TCC_d=geom_hline(yintercept=TCC_lim, linetype="solid", color = "black", size =␣
 ↪0.25) #lower limit of quantification
ICC_d=geom_hline(yintercept=ICC_lim, linetype="dashed", color = "black", size =␣
 ↪0.5) #lower limit of quantification
ATPt_d=geom_hline(yintercept=ATPt_lim, linetype="solid", color = "black", size␣
 ↪= 0.25)  # Total ATP lower limit of quantification
ATPi_d=geom_hline(yintercept=ATPi_lim, linetype="dashed", color = "black", size␣
 ↪= 0.5)  # Intracellular ATP limit of quantification
```

```
HPC_d=geom_hline(yintercept=HPC_lim, linetype="dashed", color = "black", size =␣
 ↪0.5)

#axis labels- x
flow_x=xlab("flow cytometer")
free_x= xlab("free chlorine residual (mg/L)")
total_x= xlab("total chlorine residual (mg/L)")
date_x= xlab("sampling date")
lc_x<- xlab("sampling site")

#axis labels- y
IT_y= ylab("proportion of intact cells")
TCC_y=ylab("total cell count (cells/mL)")
ICC_y=ylab("intact cell count (cells/mL)")
CC_y=ylab("cell count (cells/mL)")
ATPi_y= ylab("intracellular ATP (nM)")
 ATPt_y= ylab("total ATP (nM)")
R_y=ylab("regrowth potential")
HPC_y= ylab("HPC\n(most probable number per 100 mL)")
ATP_y= ylab("ATP (nM)")
regf_y=ylab("increase in total cell count (\u0025)")
cl2_y= ylab("chlorine (mg/L)")
```

[73]:
```
#rename everything for plots
SG_all$site <- gsub("other_site_in_", "other site\nin ", SG_all$site)
SGPI_all$site <- gsub("other_site_in_", "other site\nin ", SGPI_all$site)
ATP_long$site <- gsub("other_site_in_", "other site\nin ", ATP_long$site)
fcm_all_long$site <- gsub("other_site_in_", "other site\nin ",␣
 ↪fcm_all_long$site)

SG_all$site <- gsub("DWDS_A", D_A, SG_all$site)
SGPI_all$site <- gsub("DWDS_A", D_A, SGPI_all$site)
ATP_long$site <- gsub("DWDS_A", D_A, ATP_long$site)
fcm_all_long$site <- gsub("DWDS_A", D_A, fcm_all_long$site)

SG_all$site <- gsub("DWDS_B", D_B, SG_all$site)
SGPI_all$site <- gsub("DWDS_B", D_B, SGPI_all$site)
ATP_long$site <- gsub("DWDS_B", D_B, ATP_long$site)
fcm_all_long$site <- gsub("DWDS_B", D_B, fcm_all_long$site)

SG_all$site <- gsub("DWDS_C", D_C, SG_all$site)
SGPI_all$site <- gsub("DWDS_C", D_C, SGPI_all$site)
ATP_long$site <- gsub("DWDS_C", D_C, ATP_long$site)
fcm_all_long$site <- gsub("DWDS_C", D_C,fcm_all_long$site)


SG_all$site <- gsub("DWDS_F", D_F, SG_all$site)
```

```
SGPI_all$site <- gsub("DWDS_F", D_F, SGPI_all$site)
ATP_long$site <- gsub("DWDS_F", D_F, ATP_long$site)
fcm_all_long$site <- gsub("DWDS_F", D_F, fcm_all_long$site)

SG_all$broad_location <- gsub("DWDS_A", D_A, SG_all$broad_location)
SGPI_all$broad_location <- gsub("DWDS_A", D_A, SGPI_all$broad_location)
ATP_long$broad_location <- gsub("DWDS_A", D_A, ATP_long$broad_location)
mwq_all$broad_location <- gsub("DWDS_A", D_A, mwq_all$broad_location)
mwq_quant$broad_location <- gsub("DWDS_A", D_A, mwq_quant$broad_location)
fcm_all_long$broad_location <- gsub("DWDS_A", D_A, fcm_all_long$broad_location)

SG_all$broad_location <- gsub("DWDS_B", D_B, SG_all$broad_location)
SGPI_all$broad_location <- gsub("DWDS_B", D_B, SGPI_all$broad_location)
ATP_long$broad_location <- gsub("DWDS_B", D_B, ATP_long$broad_location)
mwq_all$broad_location <- gsub("DWDS_B", D_B, mwq_all$broad_location)
mwq_quant$broad_location <- gsub("DWDS_B", D_B, mwq_quant$broad_location)
fcm_all_long$broad_location <- gsub("DWDS_B", D_B, fcm_all_long$broad_location)

SG_all$broad_location <- gsub("DWDS_C", D_C, SG_all$broad_location)
SGPI_all$broad_location <- gsub("DWDS_C", D_C, SGPI_all$broad_location)
ATP_long$broad_location <- gsub("DWDS_C", D_C, ATP_long$broad_location)
mwq_all$broad_location <- gsub("DWDS_C", D_C, mwq_all$broad_location)
mwq_quant$broad_location <- gsub("DWDS_C", D_C, mwq_quant$broad_location)
fcm_all_long$broad_location <- gsub("DWDS_C", D_C,fcm_all_long$broad_location)

SG_all$broad_location <- gsub("DWDS_D", D_D, SG_all$broad_location)
SGPI_all$broad_location <- gsub("DWDS_D", D_D, SGPI_all$broad_location)
ATP_long$broad_location <- gsub("DWDS_D", D_D, ATP_long$broad_location)
mwq_all$broad_location <- gsub("DWDS_D", D_D, mwq_all$broad_location)
mwq_quant$broad_location <- gsub("DWDS_D", D_D, mwq_quant$broad_location)
fcm_all_long$broad_location <- gsub("DWDS_D", D_D, fcm_all_long$broad_location)

SG_all$broad_location <- gsub("DWDS_E", D_E, SG_all$broad_location)
SGPI_all$broad_location <- gsub("DWDS_E", D_E, SGPI_all$broad_location)
ATP_long$broad_location <- gsub("DWDS_E", D_E, ATP_long$broad_location)
mwq_all$broad_location <- gsub("DWDS_E", D_E, mwq_all$broad_location)
mwq_quant$broad_location <- gsub("DWDS_E", D_E, mwq_quant$broad_location)
fcm_all_long$broad_location <- gsub("DWDS_E", D_E, fcm_all_long$broad_location)

SG_all$broad_location <- gsub("DWDS_F", D_F, SG_all$broad_location)
SGPI_all$broad_location <- gsub("DWDS_F", D_F, SGPI_all$broad_location)
ATP_long$broad_location <- gsub("DWDS_F", D_F, ATP_long$broad_location)
mwq_all$broad_location <- gsub("DWDS_F", D_F, mwq_all$broad_location)
mwq_quant$broad_location <- gsub("DWDS_F", D_F, mwq_quant$broad_location)
fcm_all_long$broad_location <- gsub("DWDS_F", D_F, fcm_all_long$broad_location)

ATP_long$variable <- gsub("total_ATP_gmean_nM", ATPt_1, ATP_long$variable)
```

```
ATP_long$variable <- gsub("intra_ATP_gmean_nM", ATPi_l, ATP_long$variable)

SG_all$stain <- gsub("SG", TCC_l, SG_all$stain)
SGPI_all$stain <- gsub("SGPI", ICC_l, SGPI_all$stain)
fcm_all_long$stain <- gsub("SGPI", ICC_l, fcm_all_long$stain)
fcm_all_long$stain <- gsub("SG", TCC_l, fcm_all_long$stain)

names(SGPI_all)[names(SGPI_all) == "location_code"] <- lc
names(SG_all)[names(SG_all) == "location_code"] <- lc
names(ATP_long)[names(ATP_long) == "location_code"] <- lc
names(mwq_all)[names(mwq_all) == "location_code"] <- lc
names(mwq_quant)[names(mwq_quant) == "location_code"] <- lc
names(fcm_all_long)[names(fcm_all_long) == "location_code"] <- lc

names(SGPI_all)[names(SGPI_all) == "broad_location"] <- bl
names(SG_all)[names(SG_all) == "broad_location"] <- bl
names(ATP_long)[names(ATP_long) == "broad_location"] <- bl
names(mwq_all)[names(mwq_all) == "broad_location"] <- bl
names(mwq_quant)[names(mwq_quant) == "broad_location"] <- bl
names(fcm_all_long)[names(fcm_all_long) == "broad_location"] <- bl

names(mwq_all)[names(mwq_all) == "SG"] <- TCC_l
names(mwq_all)[names(mwq_all) == "SGPI"] <- ICC_l
names(mwq_quant)[names(mwq_quant) == "SG"] <- TCC_l
names(mwq_quant)[names(mwq_quant) == "SGPI"] <- ICC_l

names(SGPI_all)[names(SGPI_all) == "intra_ATP_gmean_nM"] <- ATPi_l
names(SG_all)[names(SG_all) == "intra_ATP_gmean_nM"] <- ATPi_l
names(mwq_all)[names(mwq_all) == "intra_ATP_gmean_nM"] <- ATPi_l
names(mwq_quant)[names(mwq_quant) == "intra_ATP_gmean_nM"] <- ATPi_l

names(SGPI_all)[names(SGPI_all) == "total_ATP_gmean_nM"] <- ATPt_l
names(SG_all)[names(SG_all) == "total_ATP_gmean_nM"] <- ATPt_l
names(mwq_all)[names(mwq_all) == "total_ATP_gmean_nM"] <- ATPt_l
names(mwq_quant)[names(mwq_quant) == "total_ATP_gmean_nM"] <- ATPt_l

names(SGPI_all)[names(SGPI_all) == "stain"] <- ccl
names(SG_all)[names(SG_all) == "stain"] <- ccl
names(fcm_all_long)[names(fcm_all_long) == "stain"] <- ccl

names(ATP_long)[names(ATP_long) == "variable"] <- atp
```

0.9 Tables 3 & S5

Assessment of detection limits

[74]:
```
row_nam<-c("intact cell count","total cell count", "intracellular ATP", "total␣
 ↪ATP","heterotrophic plate counts")
col_nam<- c("assay","n", 'percent quantifiable', 'percent below quantification␣
 ↪limit','percent above quantification limit')

#totals (n)
tot_ATPi<-length(mwq_all_c[!is.
 ↪na(mwq_all_c$intra_ATP_gmean_nM),"intra_ATP_gmean_nM"])
tot_ATPt<-length(mwq_all_c[!is.
 ↪na(mwq_all_c$total_ATP_gmean_nM),"total_ATP_gmean_nM"])
tot_SGPI<-length(mwq_all_c[!is.na(mwq_all_c$SGPI),"SGPI"])
tot_SG<-length(mwq_all_c[!is.na(mwq_all_c$SG),"SG"])
tot_HPC<-length(mwq_all_c[!is.
 ↪na(mwq_all_c$HPC_gmean_MPN_per_100mL),"HPC_gmean_MPN_per_100mL"])
tots<-c(tot_SGPI,tot_SG,tot_ATPi,tot_ATPt, tot_HPC)

#number of quantifiable samples
nq_ATPi<-length(mwq_quant_ATPi[!is.
 ↪na(mwq_quant_ATPi$intra_ATP_gmean_nM),"intra_ATP_gmean_nM"])
nq_ATPt<-length(mwq_quant_ATPt[!is.
 ↪na(mwq_quant_ATPt$total_ATP_gmean_nM),"total_ATP_gmean_nM"])
nq_SGPI<-length(mwq_quant_SGPI[!is.na(mwq_quant_SGPI$SGPI),"SGPI"])
nq_SG<-length(mwq_quant_SG[!is.na(mwq_quant_SG$SG),"SG"])
nq_HPC<-length(mwq_quant_HPC[!is.
 ↪na(mwq_quant_HPC$HPC_gmean_MPN_per_100mL),"HPC_gmean_MPN_per_100mL"])
pq<-(c(nq_SGPI,nq_SG,nq_ATPi,nq_ATPt,nq_HPC)*100)/tots

#number below quantification limit
bdl_ATPi<- tot_ATPi-nq_ATPi
bdl_ATPt<- tot_ATPt-nq_ATPt
bdl_SGPI<-tot_SGPI-nq_SGPI
bdl_SG<-tot_SG-nq_SG
bdl_HPC<-length(mwq_all[((mwq_all$HPC_label=="BDL")&(!is.
 ↪na(mwq_all$HPC_label))),]$HPC_gmean_MPN_per_100mL)
pbdl<-(c(bdl_SGPI,bdl_SG,bdl_ATPi,bdl_ATPt,bdl_HPC)*100)/tots

#number above quantification limit (only HPC)
adl_ATPi<- 0
adl_ATPt<- 0
adl_SGPI<-0
adl_SG<-0
adl_HPC<-length(mwq_all[((mwq_all$HPC_label=="ADL")&(!is.
 ↪na(mwq_all$HPC_label))),]$HPC_gmean_MPN_per_100mL)
padl<-(c(adl_SGPI,adl_SG, adl_ATPi,adl_ATPt,adl_HPC)*100)/tots

#check
```

```
adl_HPC + bdl_HPC +nq_HPC ==tot_HPC
pq+pbdl+padl
```

TRUE

1. 100 2. 100 3. 100 4. 100 5. 100

0.9.1 Table 3

quantifiable samples by assay

[75]:
```
#generate table

table1<- as.data.frame(cbind(row_nam, tots,pq,pbdl,padl))
names(table1)<- col_nam
table1

stargazer(table1,type="html", summary=FALSE,  out=paste(path_tab,"Table_3.doc"))
```

A data.frame: 5 × 5

assay <fct>	n <fct>	percent quantifiable <fct>	percent below quantification lim <fct>
intact cell count	166	97.5903614457831	2.40963855421687
total cell count	166	100	0
intracellular ATP	115	69.5652173913043	30.4347826086957
total ATP	115	100	0
heterotrophic plate counts	102	81.3725490196078	14.7058823529412

```
<table style="text-align:center"><tr><td colspan="6" style="border-bottom: 1px
solid black"></td></tr><tr><td style="text-
align:left"></td><td>assay</td><td>n</td><td>percent
quantifiable</td><td>percent below quantification limit</td><td>percent above
quantification limit</td></tr>
<tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr><tr><td
style="text-align:left">1</td><td>intact cell count</td><td>166</td><td>97.59036
14457831</td><td>2.40963855421687</td><td>0</td></tr>
<tr><td style="text-align:left">2</td><td>total cell
count</td><td>166</td><td>100</td><td>0</td><td>0</td></tr>
<tr><td style="text-align:left">3</td><td>intracellular ATP</td><td>115</td><td>
69.5652173913043</td><td>30.4347826086957</td><td>0</td></tr>
<tr><td style="text-align:left">4</td><td>total
ATP</td><td>115</td><td>100</td><td>0</td><td>0</td></tr>
<tr><td style="text-align:left">5</td><td>heterotrophic plate counts</td><td>102
</td><td>81.3725490196078</td><td>14.7058823529412</td><td>3.92156862745098</td>
</tr>
<tr><td colspan="6" style="border-bottom: 1px solid black"></td></tr></table>
```

0.9.2 Table S5

quantifiable samples by DWDS

[76]:
```
#same as above but by broad location
locations<-sort(unique(mwq_all_c$broad_location))[1:6]#DWDS

locations
table<- row_nam
for (i in locations) {
    a<-mwq_all_c[mwq_all_c$broad_location== i,]

    #totals (n)
    tot_ATPi<-length(a$intra_ATP_gmean_nM[!is.na(a$intra_ATP_gmean_nM)])
    tot_ATPt<-length(a$total_ATP_gmean_nM[!is.na(a$total_ATP_gmean_nM)])
    tot_SGPI<-length(a$SGPI[!is.na(a$SGPI)])
    tot_SG<-length(a$SG[!is.na(a$SG)])
    tot_HPC<-length(a$HPC_gmean_MPN_per_100mL[!is.
 ↪na(a$HPC_gmean_MPN_per_100mL)])
    tots<-c(tot_SGPI,tot_SG, tot_ATPi,tot_ATPt,tot_HPC)


    #find how many per location
    la<-unique(mwq_quant_ATPi$broad_location)
    li<-unique(mwq_quant_SGPI$broad_location)
    lh<-unique(mwq_quant_HPC$broad_location)

    if (i %in% la){
        qai<-mwq_quant_ATPi[mwq_quant_ATPi$broad_location== i,]
        qat<-mwq_quant_ATPt[mwq_quant_ATPt$broad_location== i,]
    }
    else{
        qai<-"-"
        qat<-"-"
    }
    if (i %in% li){
        qi<-mwq_quant_SGPI[mwq_quant_SGPI$broad_location== i,]
        qt<-mwq_quant_SG[mwq_quant_SG$broad_location== i,]
    }
    else{
        qi<-"-"
        qt<-"-"
    }
    if (i %in% lh){
        qh<-mwq_quant_HPC[mwq_quant_HPC$broad_location== i,]

    }
    else{
        qh<-"-"
    }
```

```
    # calculate number of quantifiable samples
    nq_ATPi<-length(qai$intra_ATP_gmean_nM[!is.na(qai$intra_ATP_gmean_nM)])
    nq_ATPt<-length(qat$total_ATP_gmean_nM[!is.na(qat$total_ATP_gmean_nM)])
    nq_SGPI<-length(qi$SGPI[!is.na(qi$SGPI)])
    nq_SG<-length(qt$SG[!is.na(qt$SG)])
    nq_HPC<-length(qh$HPC_gmean_MPN_per_100mL[!is.
 ↪na(qh$HPC_gmean_MPN_per_100mL)])
    pq<-round(((c(nq_SGPI,nq_SG, nq_ATPi,nq_ATPt,nq_HPC)*100)/tots),1)

    #tack onto table
    new<-as.data.frame(cbind(tots, pq))
    ntot<-paste("n (", i, ")", sep="")
    nnq<- paste("percent quantifiable (", i, ")", sep="")
    names(new)<- c(ntot, nnq)
    table<-cbind(table, new)
}
table
```

1. DWDS_A 2. DWDS_B 3. DWDS_C 4. DWDS_D 5. DWDS_E 6. DWDS_F

Levels: 1. 'DWDS_A' 2. 'DWDS_B' 3. 'DWDS_C' 4. 'DWDS_D' 5. 'DWDS_E' 6. 'DWDS_F'

A data.frame: 5 × 13

table <fct>	n (DWDS_A) <dbl>	percent quantifiable (DWDS_A) <dbl>	n (DWDS <dbl>
intact cell count	22	100.0	20
total cell count	22	100.0	20
intracellular ATP	11	90.9	10
total ATP	11	100.0	10
heterotrophic plate counts	21	76.2	10

[77]:
```
#export
stargazer(table,type="html", summary=FALSE,  out=paste(path_tab,"Table_S5.doc"))
table
```

```
<table style="text-align:center"><tr><td colspan="14" style="border-bottom: 1px
solid black"></td></tr><tr><td style="text-align:left"></td><td>table</td><td>n
(DWDS_A)</td><td>percent quantifiable (DWDS_A)</td><td>n
(DWDS_B)</td><td>percent quantifiable (DWDS_B)</td><td>n
(DWDS_C)</td><td>percent quantifiable (DWDS_C)</td><td>n
(DWDS_D)</td><td>percent quantifiable (DWDS_D)</td><td>n
(DWDS_E)</td><td>percent quantifiable (DWDS_E)</td><td>n
(DWDS_F)</td><td>percent quantifiable (DWDS_F)</td></tr>
<tr><td colspan="14" style="border-bottom: 1px solid black"></td></tr><tr><td
style="text-align:left">1</td><td>intact cell count</td><td>22</td><td>100</td><
td>20</td><td>100</td><td>12</td><td>100</td><td>7</td><td>85.700</td><td>5</td>
<td>40</td><td>100</td><td>100</td></tr>
<tr><td style="text-align:left">2</td><td>total cell count</td><td>22</td><td>10
0</td><td>20</td><td>100</td><td>12</td><td>100</td><td>7</td><td>100</td><td>5<
```

```
/td><td>100</td><td>100</td><td>100</td></tr>
<tr><td style="text-align:left">3</td><td>intracellular ATP</td><td>11</td><td>9
0.900</td><td>10</td><td>90</td><td>0</td><td></td><td>0</td><td></td><td>0</td>
<td></td><td>94</td><td>64.900</td></tr>
<tr><td style="text-align:left">4</td><td>total ATP</td><td>11</td><td>100</td><
td>10</td><td>100</td><td>0</td><td></td><td>0</td><td></td><td>0</td><td></td><
td>94</td><td>100</td></tr>
<tr><td style="text-align:left">5</td><td>heterotrophic plate counts</td><td>21<
/td><td>76.200</td><td>10</td><td>90</td><td>0</td><td></td><td>0</td><td></td><
td>0</td><td></td><td>71</td><td>81.700</td></tr>
<tr><td colspan="14" style="border-bottom: 1px solid black"></td></tr></table>
```

A data.frame: 5 × 13

table <fct>	n (DWDS_A) <dbl>	percent quantifiable (DWDS_A) <dbl>	n (DWDS <dbl>
intact cell count	22	100.0	20
total cell count	22	100.0	20
intracellular ATP	11	90.9	10
total ATP	11	100.0	10
heterotrophic plate counts	21	76.2	10

0.10 5. Remove ADL samples

remove samples that were above the detection limit for HPC for further analysis (4 samples)

[78]:
```
sum(!is.na(SGPI_all$HPC_gmean_MPN_per_100mL))
sum(!is.na(mwq_all$HPC_gmean_MPN_per_100mL))
length(SGPI_all[(!is.na(SG_all$HPC_label)&(SG_all$HPC_label ==␣
 ↪"ADL")),"HPC_gmean_MPN_per_100mL"])#4 samples

SG_all[(!is.na(SG_all$HPC_label)&(SG_all$HPC_label ==␣
 ↪"ADL")),"HPC_gmean_MPN_per_100mL"]<- NA
SGPI_all[(!is.na(SGPI_all$HPC_label)&(SGPI_all$HPC_label ==␣
 ↪"ADL")),"HPC_gmean_MPN_per_100mL"]<- NA
mwq_all[(!is.na(mwq_all$HPC_label)&(mwq_all$HPC_label ==␣
 ↪"ADL")),"HPC_gmean_MPN_per_100mL"]<- NA
mwq_quant[(!is.na(mwq_quant$HPC_label)&(mwq_quant$HPC_label ==␣
 ↪"ADL")),"HPC_gmean_MPN_per_100mL"]<- NA
mwq_all_c[(!is.na(mwq_all_c$HPC_label)&(mwq_all_c$HPC_label ==␣
 ↪"ADL")),"HPC_gmean_MPN_per_100mL"]<- NA
mwq_quant_c[(!is.na(mwq_quant_c$HPC_label)&(mwq_quant_c$HPC_label ==␣
 ↪"ADL")),"HPC_gmean_MPN_per_100mL"]<- NA

SG_all[(!is.na(SG_all$HPC_label)&(SG_all$HPC_label ==␣
 ↪"ADL")),"HPC_gstd_MPN_per_100mL"]<- NA
SGPI_all[(!is.na(SGPI_all$HPC_label)&(SGPI_all$HPC_label ==␣
 ↪"ADL")),"HPC_gstd_MPN_per_100mL"]<- NA
mwq_all[(!is.na(mwq_all$HPC_label)&(mwq_all$HPC_label ==␣
 ↪"ADL")),"HPC_gstd_MPN_per_100mL"]<- NA
```

```
mwq_quant[(!is.na(mwq_quant$HPC_label)&(mwq_quant$HPC_label ==␣
 ↪"ADL")),"HPC_gstd_MPN_per_100mL"]<- NA
mwq_all_c[(!is.na(mwq_all_c$HPC_label)&(mwq_all_c$HPC_label ==␣
 ↪"ADL")),"HPC_gstd_MPN_per_100mL"]<- NA
mwq_quant_c[(!is.na(mwq_quant_c$HPC_label)&(mwq_quant_c$HPC_label ==␣
 ↪"ADL")),"HPC_gstd_MPN_per_100mL"]<- NA

unique(SG_all[(!is.na(SG_all$HPC_label)&(SG_all$HPC_label ==␣
 ↪"ADL")),"HPC_gmean_MPN_per_100mL"])
unique(SGPI_all[(!is.na(SGPI_all$HPC_label)&(SGPI_all$HPC_label ==␣
 ↪"ADL")),"HPC_gmean_MPN_per_100mL"])

sum(!is.na(SGPI_all$HPC_gmean_MPN_per_100mL))
sum(!is.na(mwq_all$HPC_gmean_MPN_per_100mL))
```

102

102

4

<NA>

<NA>

98

98

0.11 Figure 1

ICC, HPC, and ATPi in all DWDSs vs disinfectant concentration

[79]:
```
w2=6
h2=6
p=0.25
c= "other site\nin system C\n"
f= "other site\nin system F\n"
o="site_24\n\n"

options(repr.plot.width = w2, repr.plot.height = h2) #for plotting size in␣
 ↪jupyter
theme_set(theme_classic(base_size=10, base_family="Arial"))#, base_line_size=␣
 ↪1))
```

[80]:
```
a=SGPI_all
a=subset(a, a$disinfectant=="Chloramine")
a=subset(a, a$wtp=="No")

a$site<-gsub("other site\nin system C",c, a$site)
```

```
a$site<-gsub("other site\nin system F",f, a$site)
a$site<-gsub("site_24",o, a$site)
a$site<-factor(a$site, levels=c("site_ut","site_15", "site_10",o,f, c))

#plot
f2_a<-ggplot(a, aes_string(x="total_Cl2_mg.L", y="avg_cells_per_mL_gmean",␣
↪color= blp))+
   geom_pointrange(aes(ymin=avg_cells_per_mL_gmean/avg_cells_per_mL_gstd,
                       ymax=avg_cells_per_mL_gmean *avg_cells_per_mL_gstd,␣
↪shape=site, fill=site),size=p)+
   ICC_d+
   scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
                 labels = trans_format("log10", math_format(10^.
↪x)),limits=c(1,500000)) +
   scale_shape_manual(values= c(21,24,22,23,25,25))+
   scale_fill_manual(values=␣
↪c(grey_s[1],grey_s[1],grey_s[1],grey_s[1],grey_s[1],green_s[2]))+
   scale_color_manual(values= c(green_s[2],grey_s[1]))+
   ICC_y +
   total_x+
   scale_x_continuous(limits=c(0,3))+
    guides(shape=guide_legend(title.position="top",title.hjust=0.
↪5,ncol=2),color=guide_legend(title.position="top",ncol=1))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
↪element_text(hjust = 0.5),legend.position="bottom")
```

[81]:
```
a=SGPI_all
a=subset(a, a$disinfectant=="Chlorine")
a=subset(a, a$wtp=="No")
a$site<-factor(a$site, levels=c("site_ut","site_15", "site_10","site_24","other␣
↪site\nin system F","other site\nin system C"))



f2_d<-ggplot(a, aes_string(x="free_Cl2_mg.L", y="avg_cells_per_mL_gmean",␣
↪color=blp))+
   geom_pointrange(aes(ymin=avg_cells_per_mL_gmean/avg_cells_per_mL_gstd,
                       ymax=avg_cells_per_mL_gmean *avg_cells_per_mL_gstd,␣
↪shape=sampling_year), size=p)+
   ICC_d+
   scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
                 labels = trans_format("log10", math_format(10^.
↪x)),limits=c(1,500000)) +
   scale_shape_manual(values= c(12,8))+
```

```
    scale_color_manual(values= c(blue_s[3],pink_s[1],brown_s[2],pink_s[3]))+
    ICC_y +
    free_x+
    scale_x_continuous(limits=c(0,2.15))+
          guides(shape=guide_legend(title.position="top",title.hjust=0.
 ↪25,ncol=1),color=guide_legend(title.position="top",ncol=1))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
 ↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
 ↪element_text(hjust = 0.5),legend.position="bottom")
```

[82]:
```
a=SG_all
a=subset(a, a$disinfectant=="Chloramine")
a=subset(a, a$wtp=="No")

a$site<-gsub("other site\nin system C",c, a$site)
a$site<-gsub("other site\nin system F",f, a$site)
a$site<-gsub("site_24",o, a$site)
a$site<-factor(a$site, levels=c("site_ut","site_15", "site_10",o,f, c))

f2_c<-ggplot(a, aes_string(x="total_Cl2_mg.L", y="HPC_gmean_MPN_per_100mL",␣
 ↪color=blp))+
    geom_pointrange(aes(ymin=HPC_gmean_MPN_per_100mL/HPC_gstd_MPN_per_100mL,
                        ymax=HPC_gmean_MPN_per_100mL*HPC_gstd_MPN_per_100mL,␣
 ↪shape=site, fill= site),size=p)+
    HPC_d+
    scale_shape_manual(values= c(21,24,22,23,25,25))+
    scale_fill_manual(values=␣
 ↪c(grey_s[1],grey_s[1],grey_s[1],grey_s[1],grey_s[1],green_s[2]))+
    scale_color_manual(values= c(green_s[2],grey_s[1]))+
    scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
                  labels = trans_format("log10", math_format(10^.x)), limits=c(0.
 ↪1,15000)) +
    HPC_y+
    total_x+
    scale_x_continuous(limits=c(0,3))+
          guides(shape=guide_legend(title.position="top",title.hjust=0.
 ↪5,ncol=2),color=guide_legend(title.position="top",ncol=1))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
 ↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
 ↪element_text(hjust = 0.5),legend.position="bottom")
```

[83]:
```
a=SG_all
a=subset(a, a$disinfectant=="Chlorine")
a=subset(a, a$wtp=="No")
```

```
f2_f<-ggplot(a, aes_string(x="free_Cl2_mg.L", y="HPC_gmean_MPN_per_100mL",␣
↪color=blp))+
    geom_pointrange(aes(ymin=HPC_gmean_MPN_per_100mL/HPC_gstd_MPN_per_100mL,
                    ␣
↪ymax=HPC_gmean_MPN_per_100mL*HPC_gstd_MPN_per_100mL,shape=sampling_year),size=p)+
    HPC_d+
    scale_shape_manual(values= c(12,8))+
    scale_color_manual(values= c(blue_s[3],pink_s[1],brown_s[2],pink_s[3]))+
    scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
              labels = trans_format("log10", math_format(10^.x)), limits=c(0.
↪1,15000)) +
    HPC_y+
    free_x+
    scale_x_continuous(limits=c(0,2.15))+
          guides(shape=guide_legend(title.position="top",title.hjust=0.
↪25,ncol=1),color=guide_legend(title.position="top",ncol=1))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
↪element_text(hjust = 0.5),legend.position="bottom")
```

[84]:
```
a=ATP_long
a=subset(a, a$disinfectant=="Chloramine")
a<-a[a[[atp]]==ATPi_1,]

a=subset(a, a$wtp=="No")

a$site<-gsub("other site\nin system C",c, a$site)
a$site<-gsub("other site\nin system F",f, a$site)
a$site<-gsub("site_24",o, a$site)
a$site<-factor(a$site, levels=c("site_ut","site_15", "site_10",o,f, c))


f2_b<- ggplot(a, aes_string(x="total_Cl2_mg.L", y="value", color=blp))+
    geom_pointrange(aes(ymin=value/ATP_stdev,
                        ymax=value*ATP_stdev,shape=site,fill=site),size=p)+
    #ATPt_d+
    ATPi_d+
    scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
                  labels = trans_format("log10", math_format(10^.x)), limits=c(0.
↪000001,0.1)) +
    scale_shape_manual(values= c(21,24,22,23,25,25))+
    scale_fill_manual(values=␣
↪c(grey_s[1],grey_s[1],grey_s[1],grey_s[1],grey_s[1],green_s[2]))+
```

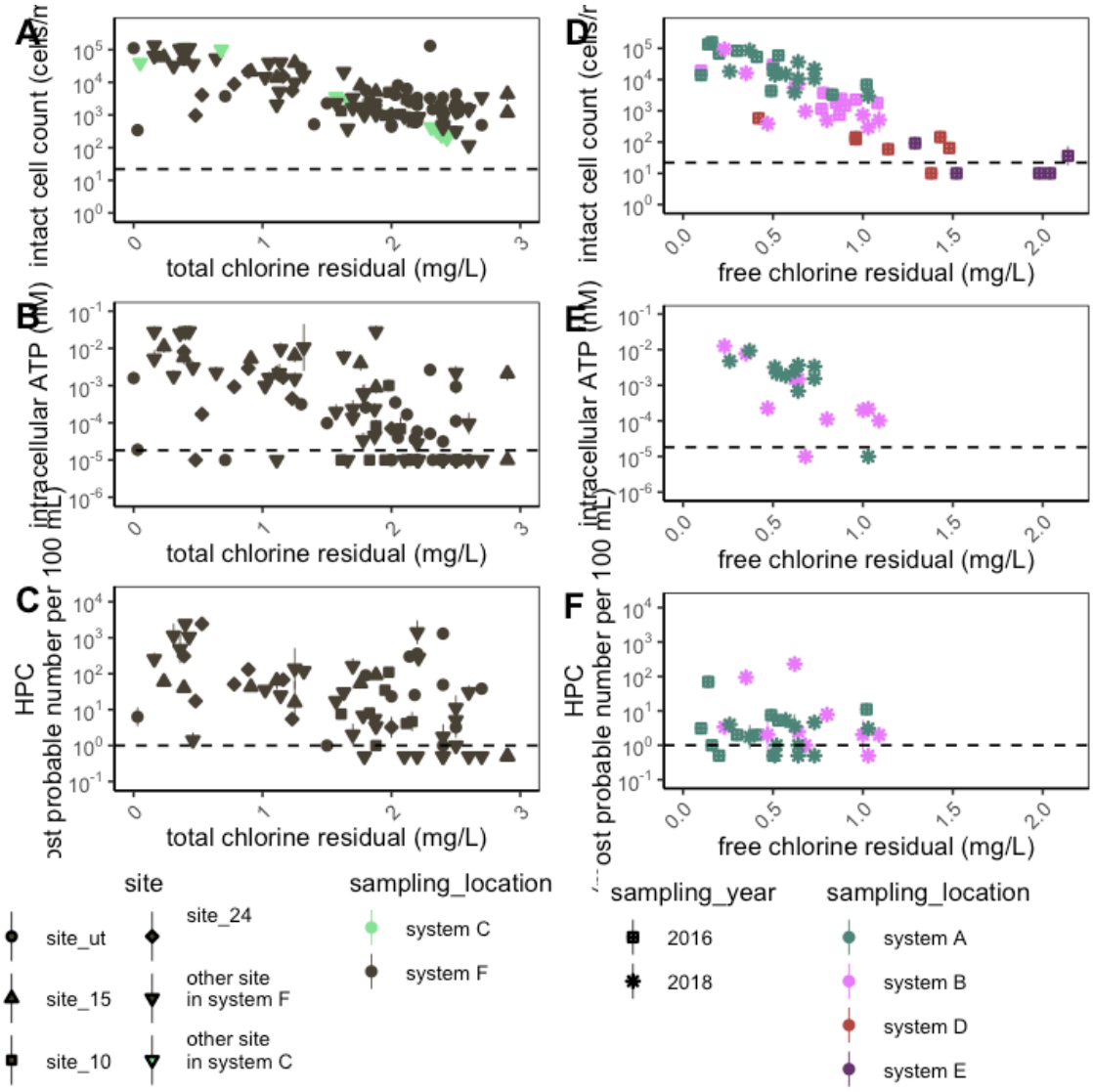
```
    scale_color_manual(values= c(green_s[2],grey_s[1]))+
    ATPi_y+
    total_x+
    scale_x_continuous(limits=c(0,3))+
              guides(shape=guide_legend(title.position="top",title.hjust=0.
 ↪5,ncol=2),color=guide_legend(title.position="top",ncol=1))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
 ↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
 ↪element_text(hjust = 0.5),legend.position="bottom")
```

[85]:
```
a=ATP_long
a=subset(a, a$disinfectant=="Chlorine")
a<-a[a[[atp]]==ATPi_1,]
a=subset(a, a$wtp=="No")



f2_e<-ggplot(a, aes_string(x="free_Cl2_mg.L", y="value", color= blp))+
           geom_pointrange(aes(ymin=value/ATP_stdev,
                     ymax=value * ATP_stdev, shape=sampling_year),size=p)+
    #ATPt_d+
    ATPi_d+
    scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
                  labels = trans_format("log10", math_format(10^.x)),limits=c(0.
 ↪000001,0.1)) +
    scale_shape_manual(values= c(12,8))+
    scale_color_manual(values= c(blue_s[3],pink_s[1],brown_s[2],pink_s[3]))+
    ATPi_y+
    free_x+
    scale_x_continuous(limits=c(0,2.15))+
       guides(shape=guide_legend(title.position="top",title.hjust=0.
 ↪25,ncol=1),color=guide_legend(title.position="top",ncol=1))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
 ↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
 ↪element_text(hjust = 0.5),legend.position="bottom")
```

0.11.1 plot

[86]:
```
# combine into plot2

#paper
pa<-ggarrange(f2_a,  f2_b,f2_c,
          labels = c("A", "B", "C"),
#            ncol = 1, nrow = 3, legend="none")
```

```
          ncol = 1, nrow = 3, common.legend=TRUE, legend= "bottom", align="v")

pb<-ggarrange(f2_d, f2_e, f2_f,
          labels = c("D", "E", "F"),
#          ncol = 1, nrow = 3, legend="none")
          ncol = 1, nrow = 3, common.legend=TRUE, legend= "bottom", align="v")

ggarrange(pa,pb, ncol = 2, nrow = 1)

ggsave(paste(path_fig,"Figure_1.pdf",sep=""),  units='mm',device=␣
 ↪cairo_pdf,width=max_w, height=max_h)
```



0.12 Table S2

geometric and arithmetic means for variables in all DWDS

[87]:
```
a<-subset(mwq_quant, mwq_quant$wtp=="No")
a<-subset(a, a$disinfectant=="Chlorine")

a<-a[,c("sample_date","sampling_site","sampling_location","total cell count",␣
↪"intact cell count", "free_Cl2_mg.L", "total_Cl2_mg.L", "pH", "temp_C",␣
↪"intracellular ATP", "HPC_gmean_MPN_per_100mL", "total ATP")]

colnames(a)<- c("sample_date","location_code","broad_location","B.TCC", "A.
↪ICC", "F.free", "G.total", "H.pH", "I.temp", "C.ATPi", "E.HPC", "D.ATPt")
```

[88]:
```
#summary table output to word
#need to convert to scientific notation by copying and pasting in excel but␣
↪easy after this

df.sum<- a %>% dplyr::select(B.TCC, A.ICC, F.free, G.total, H.pH, I.temp, C.
↪ATPi, D.ATPt, E.HPC ) %>%
    dplyr::summarise_each(funs(n=sum(!is.na(.)),min=min(.,na.
↪rm=TRUE),median=median(.,na.rm=TRUE), max=max(.,na.rm=TRUE), mean=mean(.,na.
↪rm=TRUE), sd=sd(.,na.rm=TRUE), gmean=Gmean(.,na.rm=TRUE), gsd=Gsd(.,na.
↪rm=TRUE)))

df.stats.tidy <- df.sum %>% gather(stat, val, na.rm=TRUE) %>%
    separate(stat, into = c("var", "stat"), sep = "_") %>%
    spread(stat, val) %>%
    dplyr::select(var, n, min, median, max, gmean, gsd, mean, sd)

df.stats.tidy$var<-c("intact cell count (cells/mL)","total cell count (cells/
↪mL)", "intracellular ATP (nM)",  "total ATP (nM)","HPC (MPN/100 mL)","free␣
↪chlorine (mg/L)", "total chlorine (mg/L)", "pH", "temperature (C)")
colnames(df.stats.tidy)<- c("indicator","n","min","median","max", "geometric␣
↪mean", "geometric standard deviation","arithmetic mean", "arithmetic␣
↪standard deviation")

df.stats.tidy$disinfectant='Chlorine'
df.stats.tidy.chl=df.stats.tidy

# stargazer(df.stats.tidy,type="html", summary=FALSE, ␣
↪out=paste(path_tab,"Table_S2.doc"))
```

[89]:
```
a<-subset(mwq_quant, mwq_quant$wtp=="No")
a<-subset(a, a$disinfectant=="Chloramine")

a<-a[,c("sample_date","sampling_site","sampling_location","total cell count",␣
↪"intact cell count", "free_Cl2_mg.L", "total_Cl2_mg.L", "pH", "temp_C",␣
↪"intracellular ATP", "HPC_gmean_MPN_per_100mL", "total ATP")]
```

```
colnames(a)<- c("sample_date","location_code","broad_location","B.TCC", "A.
↪ICC", "F.free", "G.total", "H.pH", "I.temp", "C.ATPi", "E.HPC", "D.ATPt")
```

[90]:
```
#summary table output to word
#need to convert to scientific notation by copying and pasting in excel but␣
↪easy after this

df.sum<- a %>% dplyr::select(B.TCC, A.ICC, F.free, G.total, H.pH, I.temp, C.
↪ATPi, D.ATPt, E.HPC ) %>%
    dplyr::summarise_each(funs(n=sum(!is.na(.)),min=min(.,na.
↪rm=TRUE),median=median(.,na.rm=TRUE), max=max(.,na.rm=TRUE), mean=mean(.,na.
↪rm=TRUE), sd=sd(.,na.rm=TRUE), gmean=Gmean(.,na.rm=TRUE), gsd=Gsd(.,na.
↪rm=TRUE)))

df.stats.tidy <- df.sum %>% gather(stat, val, na.rm=TRUE) %>%
    separate(stat, into = c("var", "stat"), sep = "_") %>%
    spread(stat, val) %>%
    dplyr::select(var, n, min, median, max, gmean, gsd, mean, sd)

df.stats.tidy$var<-c("intact cell count (cells/mL)","total cell count (cells/
↪mL)", "intracellular ATP (nM)",  "total ATP (nM)","HPC (MPN/100 mL)","free␣
↪chlorine (mg/L)", "total chlorine (mg/L)", "pH", "temperature (C)")
colnames(df.stats.tidy)<- c("indicator","n","min","median","max", "geometric␣
↪mean", "geometric standard deviation","arithmetic mean", "arithmetic␣
↪standard deviation")

df.stats.tidy$disinfectant='Chloramine'
df.stats.tidy.cam=df.stats.tidy

# stargazer(df.stats.tidy,type="html", summary=FALSE, ␣
↪out=paste(path_tab,"Table_S2.doc"))
```

[91]:
```
#stack
df.stats.tidy=rbind(df.stats.tidy.cam,df.stats.tidy.chl)
df.stats.tidy=df.stats.
↪tidy[,c("indicator","disinfectant","n","min","median","max","geometric␣
↪mean", "geometric standard deviation","arithmetic mean", "arithmetic␣
↪standard deviation")]
df.stats.tidy
stargazer(df.stats.tidy,type="html", summary=FALSE, ␣
↪out=paste(path_tab,"Table_S2.doc"))
```

A data.frame: 18 × 10

indicator <chr>	disinfectant <chr>	n <dbl>	min <dbl>	median <dbl>	max <db
intact cell count (cells/mL)	Chloramine	112	1.183529e+02	2.420524e+03	1.52
total cell count (cells/mL)	Chloramine	112	3.496986e+02	9.761390e+03	6.22
intracellular ATP (nM)	Chloramine	94	1.830000e-05	1.045635e-04	2.85
total ATP (nM)	Chloramine	94	1.852030e-04	1.447990e-03	3.11
HPC (MPN/100 mL)	Chloramine	67	9.999000e-01	2.471639e+01	2.41
free chlorine (mg/L)	Chloramine	96	2.000000e-02	5.000000e-02	5.40
total chlorine (mg/L)	Chloramine	109	2.000000e-02	1.880000e+00	2.90
pH	Chloramine	84	7.670000e+00	8.045000e+00	8.45
temperature (C)	Chloramine	82	1.370000e+01	1.860000e+01	2.88
intact cell count (cells/mL)	Chlorine	54	2.200000e+01	3.531297e+03	1.57
total cell count (cells/mL)	Chlorine	54	3.169952e+01	7.131288e+03	1.58
intracellular ATP (nM)	Chlorine	21	1.830000e-05	1.527416e-03	1.26
total ATP (nM)	Chlorine	21	3.079871e-03	8.249141e-03	1.54
HPC (MPN/100 mL)	Chlorine	31	9.999000e-01	2.024846e+00	2.29
free chlorine (mg/L)	Chlorine	54	1.000000e-01	7.300000e-01	2.14
total chlorine (mg/L)	Chlorine	32	2.400000e-01	7.100000e-01	1.22
pH	Chlorine	44	7.400000e+00	8.215000e+00	8.74
temperature (C)	Chlorine	35	1.570000e+01	2.290000e+01	2.61


```
<table style="text-align:center"><tr><td colspan="11" style="border-bottom: 1px
solid black"></td></tr><tr><td style="text-align:left"></td><td>indicator</td><t
d>disinfectant</td><td>n</td><td>min</td><td>median</td><td>max</td><td>geometri
c mean</td><td>geometric standard deviation</td><td>arithmetic
mean</td><td>arithmetic standard deviation</td></tr>
<tr><td colspan="11" style="border-bottom: 1px solid black"></td></tr><tr><td
style="text-align:left">1</td><td>intact cell count (cells/mL)</td><td>Chloramin
e</td><td>112</td><td>118.353</td><td>2,420.524</td><td>152,588.900</td><td>3,62
2.185</td><td>6.159</td><td>17,748.420</td><td>33,609.310</td></tr>
<tr><td style="text-align:left">2</td><td>total cell count (cells/mL)</td><td>Ch
loramine</td><td>112</td><td>349.699</td><td>9,761.390</td><td>622,548.900</td><
td>13,230.470</td><td>5.333</td><td>47,701.470</td><td>87,471.400</td></tr>
<tr><td style="text-align:left">3</td><td>intracellular ATP (nM)</td><td>Chloram
ine</td><td>94</td><td>0.00002</td><td>0.0001</td><td>0.029</td><td>0.0002</td><
td>11.613</td><td>0.003</td><td>0.006</td></tr>
<tr><td style="text-align:left">4</td><td>total ATP (nM)</td><td>Chloramine</td>
<td>94</td><td>0.0002</td><td>0.001</td><td>0.031</td><td>0.002</td><td>4.102</t
d><td>0.005</td><td>0.007</td></tr>
<tr><td style="text-align:left">5</td><td>HPC (MPN/100 mL)</td><td>Chloramine</t
d><td>67</td><td>1.000</td><td>24.716</td><td>2,419.699</td><td>20.641</td><td>1
0.041</td><td>199.960</td><td>492.156</td></tr>
<tr><td style="text-align:left">6</td><td>free chlorine (mg/L)</td><td>Chloramin
e</td><td>96</td><td>0.020</td><td>0.050</td><td>0.540</td><td>0.055</td><td>2.4
19</td><td>0.084</td><td>0.095</td></tr>
<tr><td style="text-align:left">7</td><td>total chlorine (mg/L)</td><td>Chlorami
ne</td><td>109</td><td>0.020</td><td>1.880</td><td>2.900</td><td>1.352</td><td>2
```

.492</td><td>1.708</td><td>0.782</td></tr>
<tr><td style="text-align:left">8</td><td>pH</td><td>Chloramine</td><td>84</td><td>7.670</td><td>8.045</td><td>8.450</td><td>8.025</td><td>1.018</td><td>8.026</td><td>0.142</td></tr>
<tr><td style="text-align:left">9</td><td>temperature (C)</td><td>Chloramine</td><td>82</td><td>13.700</td><td>18.600</td><td>28.800</td><td>19.595</td><td>1.214</td><td>19.968</td><td>3.985</td></tr>
<tr><td style="text-align:left">10</td><td>intact cell count (cells/mL)</td><td>Chlorine</td><td>54</td><td>22</td><td>3,531.297</td><td>157,782.700</td><td>2,579.431</td><td>12.939</td><td>19,145.260</td><td>34,371.790</td></tr>
<tr><td style="text-align:left">11</td><td>total cell count (cells/mL)</td><td>Chlorine</td><td>54</td><td>31.700</td><td>7,131.288</td><td>158,474.500</td><td>4,974.886</td><td>10.011</td><td>23,502.530</td><td>36,976.140</td></tr>
<tr><td style="text-align:left">12</td><td>intracellular ATP (nM)</td><td>Chlorine</td><td>21</td><td>0.00002</td><td>0.002</td><td>0.013</td><td>0.001</td><td>6.844</td><td>0.003</td><td>0.003</td></tr>
<tr><td style="text-align:left">13</td><td>total ATP (nM)</td><td>Chlorine</td><td>21</td><td>0.003</td><td>0.008</td><td>0.015</td><td>0.007</td><td>1.728</td><td>0.008</td><td>0.004</td></tr>
<tr><td style="text-align:left">14</td><td>HPC (MPN/100 mL)</td><td>Chlorine</td><td>31</td><td>1.000</td><td>2.025</td><td>229.809</td><td>3.264</td><td>4.018</td><td>15.382</td><td>44.581</td></tr>
<tr><td style="text-align:left">15</td><td>free chlorine (mg/L)</td><td>Chlorine</td><td>54</td><td>0.100</td><td>0.730</td><td>2.140</td><td>0.643</td><td>2.028</td><td>0.790</td><td>0.472</td></tr>
<tr><td style="text-align:left">16</td><td>total chlorine (mg/L)</td><td>Chlorine</td><td>32</td><td>0.240</td><td>0.710</td><td>1.220</td><td>0.664</td><td>1.548</td><td>0.722</td><td>0.276</td></tr>
<tr><td style="text-align:left">17</td><td>pH</td><td>Chlorine</td><td>44</td><td>7.400</td><td>8.215</td><td>8.740</td><td>8.221</td><td>1.038</td><td>8.226</td><td>0.303</td></tr>
<tr><td style="text-align:left">18</td><td>temperature (C)</td><td>Chlorine</td><td>35</td><td>15.700</td><td>22.900</td><td>26.100</td><td>22.103</td><td>1.114</td><td>22.223</td><td>2.264</td></tr>
<tr><td colspan="11" style="border-bottom: 1px solid black"></td></tr></table>

0.13 Table S4

geometric and arithmetic means for variables in DWDS F

[92]:
```
#chloraminated or chlorinated
a<-subset(mwq_quant, mwq_quant$wtp=="No")
a<-subset(a, mwq_quant$disinfectant=="Chloramine")
a<-a[,c("sample_date",lc,bl,"total cell count", "intact cell count",␣
 ↪"free_Cl2_mg.L", "total_Cl2_mg.L", "pH", "temp_C", "intracellular ATP",␣
 ↪"HPC_gmean_MPN_per_100mL", "total ATP")]
colnames(a)<- c("sample_date","location_code","broad_location","B.TCC", "A.
 ↪ICC", "F.free", "G.total", "H.pH", "I.temp", "C.ATPi", "E.HPC", "D.ATPt")
```

[93]:
```
#summary table output to word

df.sum<- a %>% dplyr::select(B.TCC, A.ICC, F.free, G.total, H.pH, I.temp, C.
 ↪ATPi, D.ATPt, E.HPC ) %>%
    dplyr::summarise_each(funs(n=sum(!is.na(.)),min=min(.,na.
 ↪rm=TRUE),median=median(.,na.rm=TRUE), max=max(.,na.rm=TRUE), mean=mean(.,na.
 ↪rm=TRUE), sd=sd(.,na.rm=TRUE), gmean=Gmean(.,na.rm=TRUE), gsd=Gsd(.,na.
 ↪rm=TRUE)))

df.stats.tidy <- df.sum %>% gather(stat, val, na.rm=TRUE) %>%
    separate(stat, into = c("var", "stat"), sep = "_") %>%
    spread(stat, val) %>%
    dplyr::select(var, n, min, median, max, gmean, gsd, mean, sd)

df.stats.tidy$var<-c("intact cell count (cells/mL)","total cell count (cells/
 ↪mL)", "intracellular ATP (nM)",  "total ATP (nM)","HPC (MPN/100 mL)","free␣
 ↪chlorine (mg/L)", "total chlorine (mg/L)", "pH", "temperature (C)")
colnames(df.stats.tidy)<- c("indicator","n","min","median","max", "geometric␣
 ↪mean", "geometric standard deviation","arithmetic mean", "arithmetic␣
 ↪standard deviation")

df.stats.tidy
stargazer(df.stats.tidy,type="html", summary=FALSE, ␣
 ↪out=paste(path_tab,"Table_S4.doc"))
```

A data.frame: 9 × 9

indicator <chr>	n <dbl>	min <dbl>	median <dbl>	max <dbl>	geor <db
intact cell count (cells/mL)	112	1.183529e+02	2.420524e+03	1.525889e+05	3.62
total cell count (cells/mL)	112	3.496986e+02	9.761390e+03	6.225489e+05	1.32
intracellular ATP (nM)	94	1.830000e-05	1.045635e-04	2.851924e-02	2.10
total ATP (nM)	94	1.852030e-04	1.447990e-03	3.116579e-02	1.68
HPC (MPN/100 mL)	67	9.999000e-01	2.471639e+01	2.419699e+03	2.06
free chlorine (mg/L)	96	2.000000e-02	5.000000e-02	5.400000e-01	5.51
total chlorine (mg/L)	109	2.000000e-02	1.880000e+00	2.900000e+00	1.35
pH	84	7.670000e+00	8.045000e+00	8.450000e+00	8.02
temperature (C)	82	1.370000e+01	1.860000e+01	2.880000e+01	1.95

```
<table style="text-align:center"><tr><td colspan="10" style="border-bottom: 1px
solid black"></td></tr><tr><td style="text-align:left"></td><td>indicator</td><t
d>n</td><td>min</td><td>median</td><td>max</td><td>geometric
mean</td><td>geometric standard deviation</td><td>arithmetic
mean</td><td>arithmetic standard deviation</td></tr>
<tr><td colspan="10" style="border-bottom: 1px solid black"></td></tr><tr><td
style="text-align:left">1</td><td>intact cell count (cells/mL)</td><td>112</td><
td>118.353</td><td>2,420.524</td><td>152,588.900</td><td>3,622.185</td><td>6.159
</td><td>17,748.420</td><td>33,609.310</td></tr>
<tr><td style="text-align:left">2</td><td>total cell count (cells/mL)</td><td>11
```

2</td><td>349.699</td><td>9,761.390</td><td>622,548.900</td><td>13,230.470</td><td>5.333</td><td>47,701.470</td><td>87,471.400</td></tr>
<tr><td style="text-align:left">3</td><td>intracellular ATP (nM)</td><td>94</td><td>0.00002</td><td>0.0001</td><td>0.029</td><td>0.0002</td><td>11.613</td><td>0.003</td><td>0.006</td></tr>
<tr><td style="text-align:left">4</td><td>total ATP (nM)</td><td>94</td><td>0.0002</td><td>0.001</td><td>0.031</td><td>0.002</td><td>4.102</td><td>0.005</td><td>0.007</td></tr>
<tr><td style="text-align:left">5</td><td>HPC (MPN/100 mL)</td><td>67</td><td>1.000</td><td>24.716</td><td>2,419.699</td><td>20.641</td><td>10.041</td><td>199.960</td><td>492.156</td></tr>
<tr><td style="text-align:left">6</td><td>free chlorine (mg/L)</td><td>96</td><td>0.020</td><td>0.050</td><td>0.540</td><td>0.055</td><td>2.419</td><td>0.084</td><td>0.095</td></tr>
<tr><td style="text-align:left">7</td><td>total chlorine (mg/L)</td><td>109</td><td>0.020</td><td>1.880</td><td>2.900</td><td>1.352</td><td>2.492</td><td>1.708</td><td>0.782</td></tr>
<tr><td style="text-align:left">8</td><td>pH</td><td>84</td><td>7.670</td><td>8.045</td><td>8.450</td><td>8.025</td><td>1.018</td><td>8.026</td><td>0.142</td></tr>
<tr><td style="text-align:left">9</td><td>temperature (C)</td><td>82</td><td>13.700</td><td>18.600</td><td>28.800</td><td>19.595</td><td>1.214</td><td>19.968</td><td>3.985</td></tr>
<tr><td colspan="10" style="border-bottom: 1px solid black"></td></tr></table>

0.14 Table 4

coefficient of variation

[94]:
```
a<-mwq_quant
#rename cols
a<-a[,c("std_note","HPC_label","sample_date",lc,bl,"SG_gstd", "SGPI_gstd",␣
 ↪"intra_ATP_gstd_nM","intracellular ATP", "total_ATP_gstd_nM", "intact cell␣
 ↪count","total ATP","HPC_gstd_MPN_per_100mL","HPC_gmean_MPN_per_100mL","total␣
 ↪cell count")]
colnames(a)<-␣
 ↪c("std_note","HPC_label","sample_date","location_code","broad_location","B.
 ↪TCC", "A.ICC", "C.ATPi","ATPi_m","E.ATPt","ICC_m","ATPt_m","D.
 ↪HPC","HPC_m","TCC_m")

#subset to remove unquantifiable samps (because have no stdev)
a[(!is.na(a$ATPi_m) & (a$ATPi_m== ATPi_lim)) & (a$C.ATPi==1),"C.ATPi"]<- NA
a[(!is.na(a$ATPt_m) & (a$ATPt_m== ATPt_lim))& (a$E.ATPt==1),"E.ATPt"]<- NA
a[(!is.na(a$ICC_m) & (a$ICC_m== ICC_lim))& (a$A.ICC==1),"A.ICC"]<- NA
a[!is.na(a$HPC_label) & (a$HPC_label=="BDL"),"D.HPC"]<- NA
a[is.na(a$D.HPC)&(!is.na(a$HPC_m))&(is.na(a$HPC_label)),"location_code"] #these␣
 ↪10 samples will be removed because no replicates
```

```
a[(!is.na(a$ATPi_m) & (a$ATPi_m== ATPi_lim)) ,"ATPi_m"]<- NA
a[(!is.na(a$ATPt_m) & (a$ATPt_m== ATPt_lim)),"ATPt_m"]<- NA
a[(!is.na(a$ICC_m) & (a$ICC_m== ICC_lim)),"ICC_m"]<- NA
a[!is.na(a$HPC_label) & (a$HPC_label=="BDL"),"HPC_m"]<- NA

#fin
length(!is.na(a$D.HPC))
a[(!is.na(a$std_note)&(a$std_note== 'exclude')),"D.HPC"]<-NA
length(!is.na(a$D.HPC))
a<-dplyr::select(a,-std_note)
a<-a[,c("sample_date","location_code","broad_location","B.TCC", "A.ICC", "C.
 ↪ATPi", "D.HPC", "E.ATPt")]
a[,c("B.TCC", "A.ICC", "C.ATPi", "D.HPC", "E.ATPt")]<- a[,c("B.TCC", "A.ICC",␣
 ↪"C.ATPi", "D.HPC", "E.ATPt")] -1
colnames(a)<-c("sample_date","location_code","broad_location","B.TCC", "A.ICC",␣
 ↪"C.ATPi", "E.HPC", "D.ATPt")
subset(a, a$A.ICC==0)
subset(a, a$C.ATPi==0)
subset(a, a$E.ATPt==0)
subset(a, a$D.HPC==0)
```

Levels: 1. 'site_1' 2. 'site_10' 3. 'site_11' 4. 'site_12' 5. 'site_13' 6. 'site_14' 7. 'site_15' 8. 'site_16' 9. 'site_17' 10. 'site_18' 11. 'site_19' 12. 'site_2' 13. 'site_20' 14. 'site_21' 15. 'site_22' 16. 'site_23' 17. 'site_24' 18. 'site_3' 19. 'site_5' 20. 'site_6' 21. 'site_7' 22. 'site_8' 23. 'site_9' 24. 'site_A1' 25. 'site_A10' 26. 'site_A11' 27. 'site_A2' 28. 'site_A3' 29. 'site_A4' 30. 'site_A5' 31. 'site_A6' 32. 'site_A7' 33. 'site_A8' 34. 'site_A9' 35. 'site_B1' 36. 'site_B10' 37. 'site_B2' 38. 'site_B3' 39. 'site_B4' 40. 'site_B5' 41. 'site_B6' 42. 'site_B7' 43. 'site_B8' 44. 'site_B9' 45. 'site_C1' 46. 'site_C10' 47. 'site_C11' 48. 'site_C12' 49. 'site_C2' 50. 'site_C3' 51. 'site_C4' 52. 'site_C5' 53. 'site_C6' 54. 'site_C7' 55. 'site_C8' 56. 'site_C9' 57. 'site_D1' 58. 'site_D2' 59. 'site_D3' 60. 'site_D4' 61. 'site_D5' 62. 'site_D6' 63. 'site_D7' 64. 'site_E1' 65. 'site_E2' 66. 'site_E3' 67. 'site_E4' 68. 'site_E5' 69. 'site_ut'

168

168

A data.frame: 0×8

sample_date <date>	location_code <fct>	broad_location <chr>	B.TCC <dbl>	A.ICC <dbl>	C.ATPi <dbl>	E.HPC <dbl>	D.A <d

A data.frame: 0×8

sample_date <date>	location_code <fct>	broad_location <chr>	B.TCC <dbl>	A.ICC <dbl>	C.ATPi <dbl>	E.HPC <dbl>	D.A <d

A data.frame: 0×8

sample_date <date>	location_code <fct>	broad_location <chr>	B.TCC <dbl>	A.ICC <dbl>	C.ATPi <dbl>	E.HPC <dbl>	D.A <d

A data.frame: 0×8

sample_date <date>	location_code <fct>	broad_location <chr>	B.TCC <dbl>	A.ICC <dbl>	C.ATPi <dbl>	E.HPC <dbl>	D.A <d

0.14.1 table

[95]:
```
# consistency calculations-- similar to intra assay CV- average coefficient of␣
↪variation between replicates

df.sum<- a %>% dplyr::select(B.TCC, A.ICC, D.ATPt, C.ATPi, E.HPC) %>%
    dplyr::summarise_each(funs(n=sum(!is.na(.)),min=min(.,na.
↪rm=TRUE),median=median(.,na.rm=TRUE), max=max(.,na.rm=TRUE),mean=mean(.,na.
↪rm=TRUE)))
df.stats.tidy <- df.sum %>% gather(stat, val, na.rm=TRUE) %>%
    separate(stat, into = c("var", "stat"), sep = "_") %>%
    spread(stat, val) %>%
    dplyr::select(var, n, min, median, max, mean)

df.stats.tidy$var<-c("intact cell count","total cell count", "intracellular␣
↪ATP",  "total ATP", "HPC")
colnames(df.stats.tidy)<- c("indicator","n","min","median","max", "intra-assay␣
↪coefficient of variation" )

#report as percentages
df.stats.tidy[,c("min","median","max", "intra-assay coefficient of variation"␣
↪)]<- df.stats.tidy[,c("min","median","max", "intra-assay coefficient of␣
↪variation") ]*100



df.stats.tidy
stargazer(df.stats.tidy,type="html", summary=FALSE, ␣
↪out=paste(path_tab,"Table_4.doc"))
```

A data.frame: 5 × 6

indicator <chr>	n <dbl>	min <dbl>	median <dbl>	max <dbl>	intra-assay coefficient of v <dbl>
intact cell count	162	0.0266000	9.778128	148.1027	16.936797
total cell count	166	0.3179580	6.154096	255.6794	16.953986
intracellular ATP	80	42.9840566	48.630348	327.9336	55.994417
total ATP	115	0.3885756	4.810641	65.9587	9.307352
HPC	73	0.0000000	27.131735	293.2202	49.390096

<table style="text-align:center"><tr><td colspan="7" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"></td><td>indicator</td><td>n</td><td>min</td><td>median</td><td>max</td><td>intra-assay coefficient of variation</td></tr>
<tr><td colspan="7" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left">1</td><td>intact cell count</td><td>162</td><td>0.027</td><td>9.778</td><td>148.103</td><td>16.937</td></tr>
<tr><td style="text-align:left">2</td><td>total cell count</td><td>166</td><td>0.318</td><td>6.154</td><td>255.679</td><td>16.954</td></tr>

```
<tr><td style="text-align:left">3</td><td>intracellular ATP</td><td>80</td><td>4
2.984</td><td>48.630</td><td>327.934</td><td>55.994</td></tr>
<tr><td style="text-align:left">4</td><td>total ATP</td><td>115</td><td>0.389</t
d><td>4.811</td><td>65.959</td><td>9.307</td></tr>
<tr><td style="text-align:left">5</td><td>HPC</td><td>73</td><td>0</td><td>27.13
2</td><td>293.220</td><td>49.390</td></tr>
<tr><td colspan="7" style="border-bottom: 1px solid black"></td></tr></table>
```

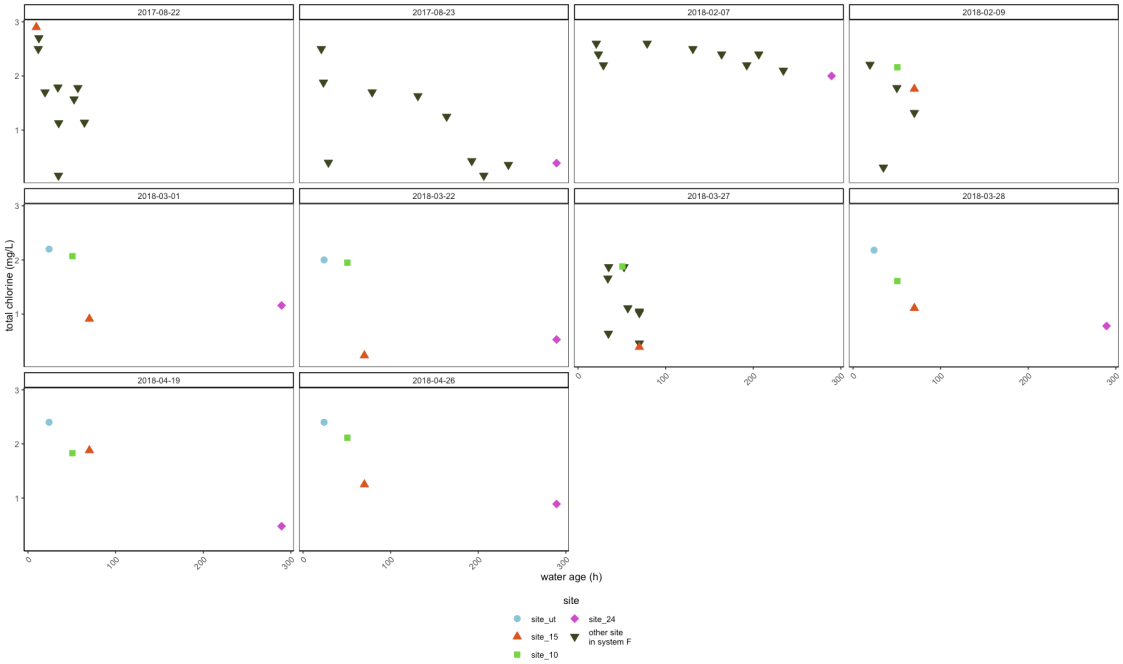
0.15 Figure S4

decay in total chlorine with water age in DWDS F by date

[96]:
```
#plot 2 sizes
w2=6
h2=6
theme_set(theme_classic(base_size=10, base_family="Arial"))#, base_line_size=␣
↪1))
p=2.5
```

[97]:
```
a=SGPI_all
a=subset(a, a$disinfectant=="Chloramine")
a=subset(a, a$wtp=="No")
a$sample_date<- as.factor(a$sample_date)
dates<- a %>%
  count(sample_date)%>%
    subset(n > 2)
dates<-dates$sample_date
a=subset(a, a$sample_date %in% dates)
a$site<-factor(a$site, levels=c("site_ut","site_15", "site_10","site_24","other␣
↪site\nin system F","other site\nin system C"))
f<-a[a[[bl]]==D_F,]
options(repr.plot.width = w2*2.5, repr.plot.height = h2*1.5) #for plotting size␣
↪in jupyter

ggplot(f, aes_string(x="water_age_h", y="total_Cl2_mg.L", color="site",␣
↪shape="site", fill="site"))+
    geom_point(size=p)+
     scale_shape_manual(values= c(21,24,22,23,25,25))+
    scale_fill_manual(values= colors)+
    scale_color_manual(values= colors)+
    ylab("total chlorine (mg/L)")+
    xlab("water age (h)")+
          guides(shape=guide_legend(title.position="top",title.hjust=0.
↪5,ncol=2),color=guide_legend(title.position="top",ncol=1))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
↪element_text(hjust = 0.5),legend.position="bottom")+facet_wrap(~sample_date)
```

```
ggsave(paste(path_fig,"Figure_S4.pdf",sep=""),  units='mm',device=␣
↪cairo_pdf,width=max_w, height=max_h/1.5)
```



0.16 Figure 4

decay in total chlorine with water age in DWDS F

[98]:
```
w2=6
h2=4
p=0.3

options(repr.plot.width = w2, repr.plot.height = h2) #for plotting size in␣
↪jupyter
theme_set(theme_classic(base_size=12, base_family="Arial"))#, base_line_size=␣
↪1))
```

[99]:
```
a=SGPI_all
a=subset(a, a$disinfectant=="Chloramine")
a=subset(a, a$wtp=="No")

a$site<-factor(a$site, levels=c("site_ut","site_15", "site_10","site_24","other␣
↪site\nin system F","other site\nin system C"))
f<-a[a[[bl]]==D_F,]
```

```
wa1<-ggplot(f, aes_string(x="water_age_h", y="avg_cells_per_mL_gmean",␣
↪color="site", shape="site", fill="site"))+ #site
   geom_pointrange(aes(min=avg_cells_per_mL_gmean/avg_cells_per_mL_gstd,
                       ymax=avg_cells_per_mL_gmean *avg_cells_per_mL_gstd ),␣
↪size=p)+
   ICC_d+
   scale_shape_manual(values= c(21,24,22,23,25,25))+
   scale_fill_manual(values= colors)+
   scale_color_manual(values= colors)+
   scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
              labels = trans_format("log10", math_format(10^.x)),␣
↪limits=c(1,1000000)) +
   ylab("intact cell count\n(cells/mL)")+
   xlab("water age (h)")+
         guides(shape=guide_legend(title.position="top",title.hjust=0.
↪5,ncol=2),color=guide_legend(title.position="top",ncol=1))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
↪"black", fill = NA),
     axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
↪element_text(hjust = 0.5),legend.position="right")
```

[100]:
```
a=ATP_long
a=subset(a, a$disinfectant=="Chloramine")
a<-a[a[[atp]]==ATPi_1,]
a=subset(a, a$wtp=="No")

a$site<-factor(a$site, levels=c("site_ut","site_15", "site_10","site_24","other␣
↪site\nin system F","other site\nin system C"))

wa2<- ggplot(a, aes_string(x="water_age_h", y="value", color="site",␣
↪shape="site", fill="site"))+ #site
   geom_pointrange(aes(ymin=value/ATP_stdev,
                       ymax=value*ATP_stdev),size=p)+
   ATPi_d+
   scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
                 labels = trans_format("log10", math_format(10^.x)), limits=c(0.
↪000001,0.1)) +
   scale_shape_manual(values= c(21,24,22,23,25,25))+
   scale_fill_manual(values= colors)+
   scale_color_manual(values= colors)+
   ylab("intracellular ATP\n(nM)")+
   xlab("water age (h)")+
              guides(shape=guide_legend(title.position="top",title.hjust=0.
↪5,ncol=2),color=guide_legend(title.position="top",ncol=1))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
↪"black", fill = NA),
```

```
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
↪element_text(hjust = 0.5),legend.position="right")
```

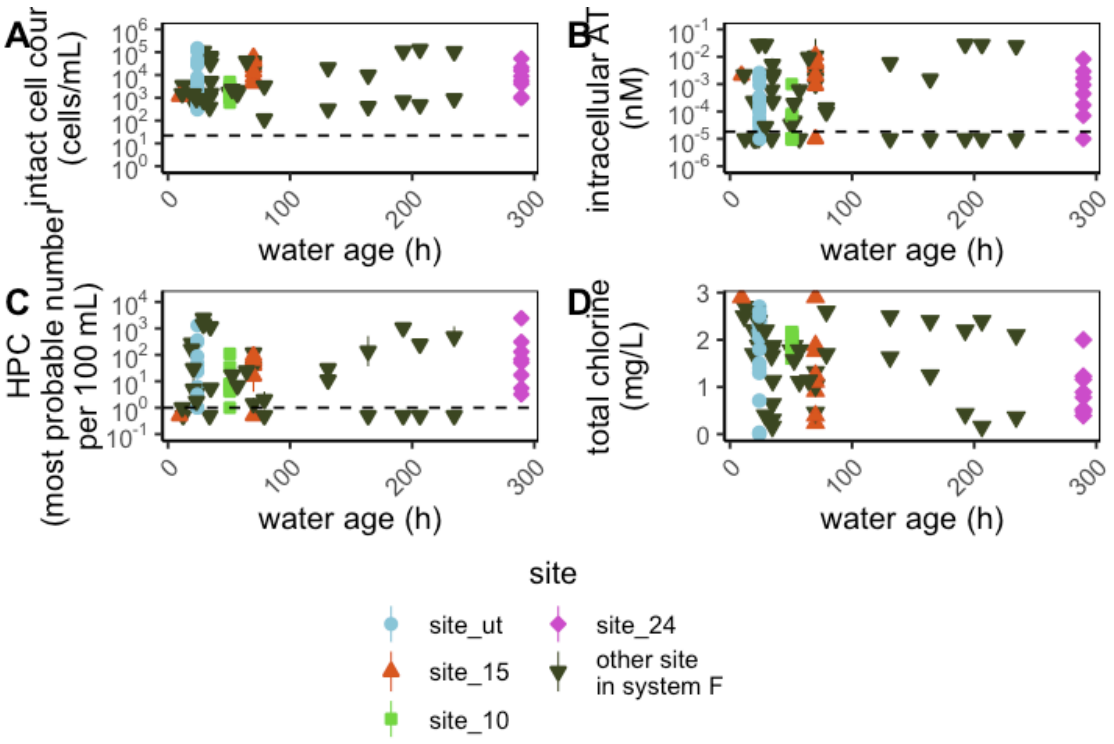
[101]:
```
a=SG_all
a=subset(a, a$disinfectant=="Chloramine")
a=subset(a, a$wtp=="No")

a$site<-factor(a$site, levels=c("site_ut","site_15", "site_10","site_24","other␣
↪site\nin system F","other site\nin system C"))
f<-a[a[[bl]]==D_F,]

wa3<-ggplot(a, aes_string(x="water_age_h", y="HPC_gmean_MPN_per_100mL",␣
↪color="site", shape="site", fill="site"))+ #site
    geom_pointrange(aes(ymin=HPC_gmean_MPN_per_100mL/HPC_gstd_MPN_per_100mL,
                    ␣
↪ymax=HPC_gmean_MPN_per_100mL*HPC_gstd_MPN_per_100mL),size=p)+
    HPC_d+
    scale_shape_manual(values= c(21,24,22,23,25,25))+
    scale_fill_manual(values= colors)+
    scale_color_manual(values= colors)+
    scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
              labels = trans_format("log10", math_format(10^.x)), limits=c(0.
↪1,15000)) +
ylab("HPC\n(most probable number\nper 100 mL)")+
 xlab("water age (h)")+
          guides(shape=guide_legend(title.position="top",title.hjust=0.
↪5,ncol=2),color=guide_legend(title.position="top",ncol=1))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
↪element_text(hjust = 0.5),legend.position="right")
```

[102]:
```
a=SGPI_all
a=subset(a, a$disinfectant=="Chloramine")
a=subset(a, a$wtp=="No")
a$sample_date<- as.factor(a$sample_date)

a$site<-factor(a$site, levels=c("site_ut","site_15", "site_10","site_24","other␣
↪site\nin system F","other site\nin system C"))
f<-a[a[[bl]]==D_F,]

options(repr.plot.width = w2, repr.plot.height = h2) #for plotting size in␣
↪jupyter
```

```
wa4<-ggplot(f, aes_string(x="water_age_h", y="total_Cl2_mg.L", color="site",␣
↪shape="site", fill="site"))+#site
    geom_point(size=2)+
     scale_shape_manual(values= c(21,24,22,23,25,25))+
    scale_fill_manual(values= colors)+
    scale_color_manual(values= colors)+
    ylab("total chlorine\n(mg/L)")+
    xlab("water age (h)")+
          guides(shape=guide_legend(title.position="top",title.hjust=0.
↪5,ncol=2),color=guide_legend(title.position="top",ncol=1))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
↪element_text(hjust = 0.5),legend.position="right")
```

0.16.1 plot

[103]:
```
ggarrange(wa1, wa2, wa3,wa4,
          labels = c("A", "B","C","D"),
          ncol = 2, nrow = 2, common.legend=TRUE, legend="bottom" , align="v")

ggsave(paste(path_fig,"Figure_4.pdf",sep=""), device= cairo_pdf, units='mm',␣
↪width=max_w, height=(max_h/1.5))
```



0.17 Figure S2

ICC/TCC by disinfectant in all DWDSs

[104]:
```
w2=10
h2=4
p=1.5

options(repr.plot.width = w2, repr.plot.height = h2) #for plotting size in␣
↪jupyter
theme_set(theme_classic(base_size=8, base_family="Arial"))#, base_line_size= 1))
```

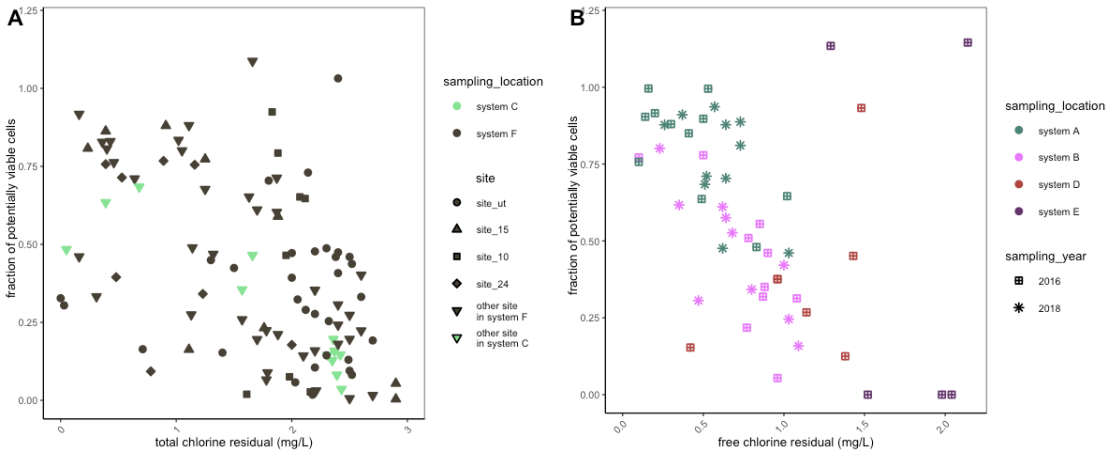
[105]:
```
a=SGPI_all
a=subset(a, a$disinfectant=="Chloramine")
a=subset(a, a$wtp=="No")
a$site<-factor(a$site, levels=c("site_ut","site_15", "site_10","site_24","other␣
↪site\nin system F","other site\nin system C"))
#plot
g2_a<-ggplot(a, aes_string(x="total_Cl2_mg.L", y="ICC_to_TCC", color=␣
↪blp,shape="site", fill="site"))+
    geom_point(size=p, stroke=p/2)+
    scale_shape_manual(values= c(21,24,22,23,25,25))+
    scale_fill_manual(values=␣
↪c(grey_s[1],grey_s[1],grey_s[1],grey_s[1],grey_s[1],green_s[2]))+
    scale_color_manual(values= c(green_s[2],grey_s[1]))+
   ylab("   fraction of potentially viable cells")+
    total_x+
    scale_x_continuous(limits=c(0,3))+
     scale_y_continuous(limits=c(0,1.2))+
     guides(shape=guide_legend(title.position="top",title.hjust=0.
↪5,ncol=1),color=guide_legend(title.position="top",ncol=1))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
↪element_text(hjust = 0.5),legend.position="bottom")
```

[106]:
```
a=SGPI_all
a=subset(a, a$disinfectant=="Chlorine")
a=subset(a, a$wtp=="No")
options(repr.plot.width = w2, repr.plot.height = h2) #for plotting size in␣
↪jupyter
a$site<-factor(a$site, levels=c("site_ut","site_15", "site_10","site_24","other␣
↪site\nin system F","other site\nin system C"))
```

```
g2_b<-ggplot(a, aes_string(x="free_Cl2_mg.L", y="ICC_to_TCC", color= blp,␣
↪shape="sampling_year"))+
    geom_point(size=p,stroke=p/2)+
    scale_shape_manual(values= c(12,8))+
    scale_color_manual(values= c(blue_s[3],pink_s[1],brown_s[2],pink_s[3]))+
   ylab("   fraction of potentially viable cells")+
    free_x+
    scale_x_continuous(limits=c(0,2.15))+
scale_y_continuous(limits=c(0,1.2))+
          guides(shape=guide_legend(title.position="top",title.hjust=0.
↪25,ncol=1),color=guide_legend(title.position="top",ncol=1))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
↪element_text(hjust = 0.5),legend.position="bottom")
```

0.17.1 plot

[107]:
```
ggarrange(g2_a,  g2_b,
          labels = c("A", "B"),
          ncol = 2, nrow = 1,  common.legend=FALSE, legend= "right", align="v")
ggsave(paste(path_fig,"Figure_S2.pdf",sep=""), device= cairo_pdf, units='mm',␣
↪width=max_w, height=max_h/4)
```



0.18 Figure S1

ICC & TCC compared to ATPi & ATPt in all DWDSs by disinfectant concentration

[108]:
```
w2=6
h2=4
p=0.25
```

```
options(repr.plot.width = w2, repr.plot.height = h2) #for plotting size in␣
↪jupyter
theme_set(theme_classic(base_size=10, base_family="Arial"))#, base_line_size=␣
↪1))
```

[109]:
```
a=SG_all
a=subset(a, a$disinfectant=="Chloramine")
a=subset(a, a$wtp=="No")


#plot
s2a<-ggplot(a, aes_string(x="total_Cl2_mg.L", y="avg_cells_per_mL_gmean",␣
↪color=blp))+
     geom_pointrange(aes(ymin=avg_cells_per_mL_gmean/avg_cells_per_mL_gstd,
                    ymax=avg_cells_per_mL_gmean *avg_cells_per_mL_gstd),size=p)+
    TCC_d+
    scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
                  labels = trans_format("log10", math_format(10^.
↪x)),limits=c(1,500000)) +
    scale_fill_manual(values=␣
↪c(grey_s[3],grey_s[4],grey_s[5],grey_s[6],grey_s[2],green_s[2]))+
    scale_color_manual(values= c(green_s[2],grey_s[1]))+
    TCC_y +
    total_x+
    scale_x_continuous(limits=c(0,3))+
     guides(shape=guide_legend(title.position="top",title.hjust=0.
↪5,ncol=1),color=guide_legend(title.position="top",ncol=1))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
↪element_text(hjust = 0.5),legend.position="bottom")
```

[110]:
```
a=SG_all
a=subset(a, a$disinfectant=="Chlorine")
a=subset(a, a$wtp=="No")


s2b<-ggplot(a, aes_string(x="free_Cl2_mg.L", y="avg_cells_per_mL_gmean",␣
↪color=blp))+
geom_pointrange(aes(ymin=avg_cells_per_mL_gmean/avg_cells_per_mL_gstd,
                    ymax=avg_cells_per_mL_gmean␣
↪*avg_cells_per_mL_gstd,),size=p)+
    TCC_d+
    scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
                  labels = trans_format("log10", math_format(10^.
↪x)),limits=c(1,500000)) +
```

```
    scale_color_manual(values= c(blue_s[3],pink_s[1],brown_s[2],pink_s[3]))+
    TCC_y +
    free_x+
    scale_x_continuous(limits=c(0,1.6))+
          guides(shape=guide_legend(title.position="top",title.hjust=0.
 ↪5,ncol=1),color=guide_legend(title.position="top",ncol=2))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
 ↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
 ↪element_text(hjust = 0.5),legend.position="bottom")
```

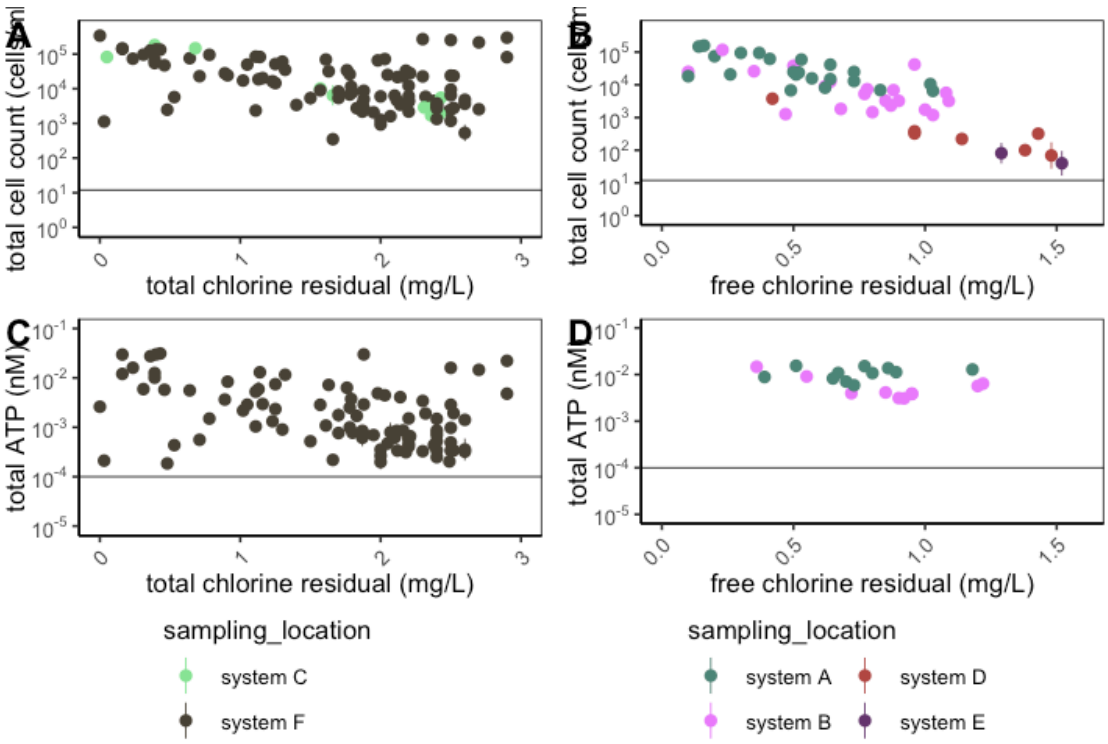
[111]:
```
a=ATP_long
a=subset(a, a$disinfectant=="Chloramine")
a<- a[(a[[atp]]!= "extra_ATP_gmean_nM"),]
a<- a[(a[[atp]]== "total ATP"),]
a=subset(a, a$wtp=="No")
a[[atp]]<-factor(a[[atp]], levels=(c(ATPt_1, ATPi_1)))


s2c<-ggplot(a, aes_string(x="total_Cl2_mg.L", y="value", color= blp))+
    geom_pointrange(aes(ymin=value/ATP_stdev,
                        ymax=value*ATP_stdev),size=p)+
    ATPt_d+
    scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
                  labels = trans_format("log10", math_format(10^.x)), limits=c(0.
 ↪00001,0.1)) +
    scale_fill_manual(values=␣
 ↪c(grey_s[3],grey_s[4],grey_s[5],grey_s[6],grey_s[2],green_s[2]))+
    scale_color_manual(values= c(green_s[2],grey_s[1]))+
    ATPt_y+
    total_x+
    scale_x_continuous(limits=c(0,3))+
             guides(shape=guide_legend(title.position="top",title.hjust=0.
 ↪5,ncol=1),color=guide_legend(title.position="top",ncol=1))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
 ↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
 ↪element_text(hjust = 0.5),legend.position="bottom")
```

[112]:
```
a=ATP_long
a=subset(a, a$disinfectant=="Chlorine")
a<- a[(a[[atp]]!= "extra_ATP_gmean_nM"),]
a<- a[(a[[atp]]== "total ATP"),]
a=subset(a, a$wtp=="No")
a[[atp]]<-factor(a[[atp]], levels=(c(ATPt_1, ATPi_1)))
```

```
s2d<-ggplot(a, aes_string(x="total_Cl2_mg.L", y="value", color= blp))+
    geom_pointrange(aes(ymin=value/ATP_stdev,
                        ymax=value*ATP_stdev),size=p)+
    ATPt_d+
    scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
                  labels = trans_format("log10", math_format(10^.x)), limits=c(0.
 ↪00001,0.1)) +
    scale_color_manual(values= c(blue_s[3],pink_s[1],brown_s[2],pink_s[3]))+
    ATPt_y +
    free_x+
    scale_x_continuous(limits=c(0,1.6))+
          guides(shape=guide_legend(title.position="top",title.hjust=0.
 ↪5,ncol=1),color=guide_legend(title.position="top",ncol=2))+
theme(panel.background=element_blank(), panel.border=element_rect(color =␣
 ↪"black", fill = NA),
      axis.text.x = element_text(angle = 45, hjust = 1),plot.caption =␣
 ↪element_text(hjust = 0.5),legend.position="bottom")
```

0.18.1 plot

[113]:
```
# combine into plot2

pa<-ggarrange(s2a,  s2c,
          labels = c("A", "C"),
          ncol = 1, nrow = 2,  common.legend=TRUE, legend= "bottom", align="v")

pb<-ggarrange(s2b,s2d,
          labels = c("B","D"),
          ncol = 1, nrow = 2, common.legend=TRUE, legend= "bottom", align="v")

ggarrange(pa,pb, ncol = 2, nrow = 1)

ggsave(paste(path_fig,"Figure_S1.pdf",sep=""), device= cairo_pdf, units='mm',␣
 ↪width=max_w, height=(max_h/1.5))
```



0.19 Figure 2

Spearman's correlation plots

[114]:
```
#order for figure
nam<-c("intact cell count", "total cell count", "intracellular ATP", "total␣
 ↪ATP", "HPC", "pH", "temperature", "free chlorine", "total chlorine")

wg<-8
hg<-8
s=2.5
options(repr.plot.width = wg, repr.plot.height = hg) #for plotting size in␣
 ↪jupyter
theme_set(theme_classic(base_size=7, base_family="Arial"))#, base_line_size= 1))
```

[115]:
```
# Get upper triangle of the correlation matrix
#order for figure

  get_upper_tri <- function(cormat){
    cormat[lower.tri(cormat)]<- NA
    diag(cormat)=NA
    return(cormat)
  }
```

```
  get_lower_tri <- function(cormat){
    cormat[upper.tri(cormat)]<- NA
    diag(cormat)=NA
    cormat<-rotate(cormat)
    return(cormat)
  }

rotate <- function(x) t(apply(x, 2, rev))

 reorder_cormat <- function(cormat){
# Use correlation between variables as distance
dd <- as.dist((1-cormat)/2)
hc <- hclust(dd)
cormat <-cormat[hc$order, hc$order]
}


 reorder_cormat_mini <- function(cormat){
# Use correlation between variables as distance
cormat <-cormat[nam,nam]

}

 reorder_cormat_mini2 <- function(cormat){
   cormat <-cormat[nam,nam]
# Use correlation between variables as distance
# cormat <-cormat[order(rownames(cormat)),order(colnames(cormat))]
}
```

[116]:
```
cols=c("HPC_gmean_MPN_per_100mL","total cell count","intact cell count", "total␣
 ↪ATP", "intracellular ATP", "pH", "temp_C", "total_Cl2_mg.L","free_Cl2_mg.L")

#chloraminated
test<-mwq_quant
test<-subset(test, test$wtp=="No")
test<-subset(test, test$disinfectant=="Chloramine")
test<-test[,which(!sapply(test,is.character))]
test<-test[,cols]
names(test)[names(test)=="HPC_gmean_MPN_per_100mL"] <- "HPC"
names(test)[names(test)=="temp_C"] <- "temperature"
names(test)[names(test)=="total_Cl2_mg.L"] <- "total chlorine"
names(test)[names(test)=="free_Cl2_mg.L"] <- "free chlorine"
test<- test[which(complete.cases(test)),]
#log transform
test[,c("HPC","total cell count","intact cell count", "total ATP",␣
 ↪"intracellular ATP") ] <- log10(test[,c("HPC","total cell count","intact␣
 ↪cell count", "total ATP", "intracellular ATP") ])
```

```
cor<-rcorr(as.matrix(test), type="spearman")
head(cor$r)
#reorder matrices
cor_or<-reorder_cormat_mini(cor$r)
corP_or<-cor$P[rownames(cor_or),]
corP_or<-corP_or[,colnames(cor_or)]
upper_tri_r<-get_upper_tri(cor_or)
upper_tri_P<-get_upper_tri(corP_or)

 diag(cor_or)<- NA
melted_r<-melt(upper_tri_r,na.rm=TRUE)
melted_P<-melt(upper_tri_P,na.rm=TRUE)
melted_P$stars <- ifelse(melted_P$value < 0.0001, "\n***",
                  ifelse(melted_P$value  < 0.001, "\n**",
                         ifelse(melted_P$value  < 0.01, "\n*", "")))
unique(melted_r[,1:2]==melted_P[,1:2]) #double check
stars<-melted_P$stars
pvalue<-melted_P$value
melted=cbind(melted_r,stars)
melted=cbind(melted,pvalue)
melted$fill<- paste(round(melted$value,2),melted$stars, sep=" ")
melted_mini<- melted

melted_mini[melted_mini$pvalue>0.01,"value" ]<-NA

melted_mini$text<-0
melted_mini[!is.na(melted_mini$value)&(melted_mini$value>0),"text"]<-1
melted_mini$text<- as.factor(melted_mini$text)
```

A matrix: 6 × 9 of type dbl

	HPC	total cell count	intact cell count	total ATP	int
HPC	1.0000000	0.24291574	0.5291601	0.26325930	0.3
total cell count	0.2429157	1.00000000	0.6965098	0.77859305	0.6
intact cell count	0.5291601	0.69650978	1.0000000	0.75799521	0.7
total ATP	0.2632593	0.77859305	0.7579952	1.00000000	0.7
intracellular ATP	0.3669691	0.62131649	0.7559113	0.78523641	1.0
pH	-0.2227906	0.07272199	-0.1325609	0.01834608	-0.

A matrix: 1 × 2 of type lgl

	Var1	Var2
10	TRUE	TRUE

[117]:
```
grey1<-ggplot(data = melted_mini, aes(Var2, Var1, fill = value))+
 geom_tile(color = "white")+
 scale_fill_viridis(name="spearman\ncorrelation",limit =␣
↪c(-1,1),option="inferno",na.value="#999999")+
    geom_text(aes(Var2, Var1, label = fill, color=text), size = s) +
    ylab("")+
    xlab("")+
```

```
    scale_color_manual(values= c("white","black"))+
    guides(color=FALSE)+
 coord_fixed()+theme(panel.background=element_blank(), panel.
 ↪border=element_rect(color = "black", fill = NA),
axis.text.x = element_text(angle = 45, hjust = 1),legend.position="left")
```

[118]:
```
#chlorinated
test<-mwq_quant
test<-subset(test, test$wtp=="No")
test<-subset(test, test$disinfectant=="Chlorine")
test<-test[,which(!sapply(test,is.character))]
test<-test[,cols]
names(test)[names(test)=="HPC_gmean_MPN_per_100mL"] <- "HPC"
names(test)[names(test)=="temp_C"] <- "temperature"
names(test)[names(test)=="total_Cl2_mg.L"] <- "total chlorine"
names(test)[names(test)=="free_Cl2_mg.L"] <- "free chlorine"
test<- test[which(complete.cases(test)),]
test<-test[,nam]
#log transform
# test[,c("HPC","total cell count","intact cell count", "total ATP",␣
 ↪"intracellular ATP") ] <- log10(test[,c("HPC","total cell count","intact␣
 ↪cell count", "total ATP", "intracellular ATP") ])
cor<-rcorr(as.matrix(test), type="spearman")

#reorder matrices
cor_or<-reorder_cormat_mini2(cor$r)
corP_or<-cor$P[rownames(cor_or),]
corP_or<-corP_or[,colnames(cor_or)]

upper_tri_r<-get_lower_tri(cor_or)
upper_tri_P<-get_lower_tri(corP_or)

melted_r<-melt(upper_tri_r,na.rm=TRUE)
melted_P<-melt(upper_tri_P,na.rm=TRUE)
melted_P$stars <- ifelse(melted_P$value < 0.0001, "\n***",
                  ifelse(melted_P$value  < 0.001, "\n**",
                         ifelse(melted_P$value  < 0.01, "\n*", "")))
unique(melted_r[,1:2]==melted_P[,1:2]) #double check
stars<-melted_P$stars
pvalue<-melted_P$value
melted=cbind(melted_r,stars)
melted=cbind(melted,pvalue)
melted$fill<- paste(round(melted$value,2),melted$stars, sep=" ")
melted_mini<- melted
melted_mini[melted_mini$pvalue>0.01,"value" ]<-NA
melted_mini$text<-0
melted_mini[!is.na(melted_mini$value)&(melted_mini$value>0),"text"]<-1
```

```
melted_mini$text<- as.factor(melted_mini$text)
```

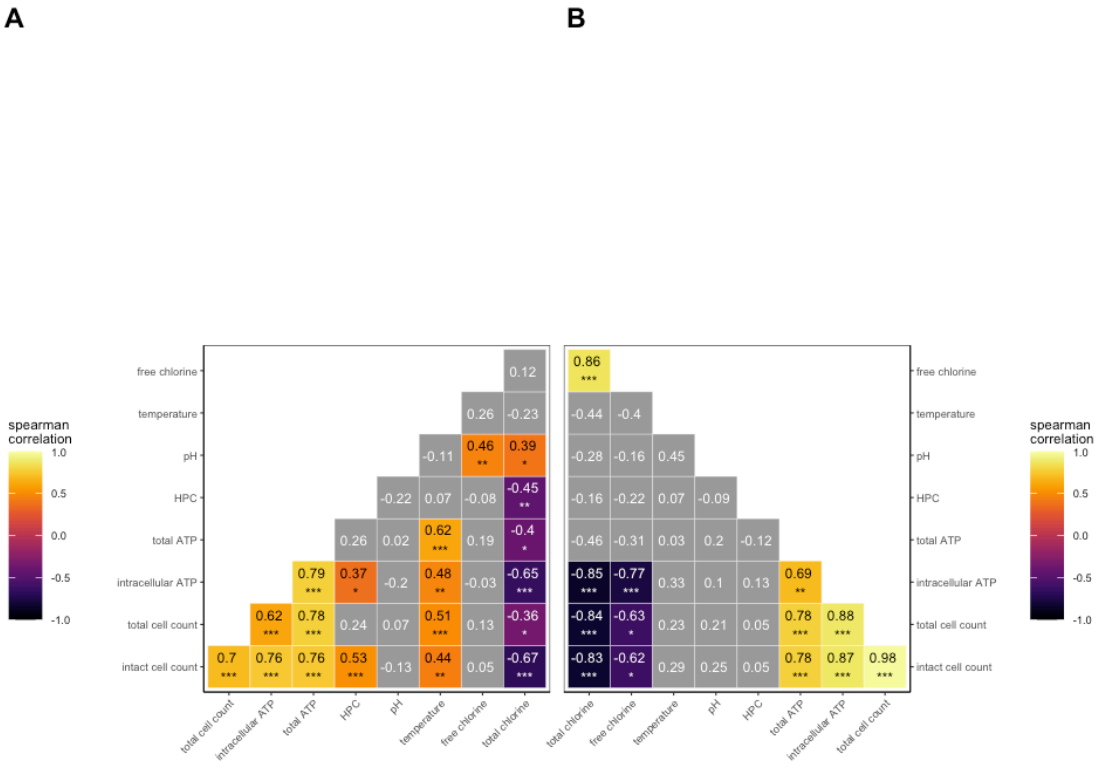
A matrix: 1 × 2 of type lgl

	Var1	Var2
1	TRUE	TRUE

[119]:
```
grey2<-ggplot(data = melted_mini, aes(Var2, Var1, fill = value))+
 geom_tile(color = "white")+
 scale_y_discrete(position="right")+
 scale_fill_viridis(name="spearman\ncorrelation",limit =␣
 ↪c(-1,1),option="inferno",na.value="#999999")+
    geom_text(aes(Var2, Var1, label = fill, color=text), size = s) +
    ylab("")+
    xlab("")+
    scale_color_manual(values= c("white","black"))+
    guides(color=FALSE)+
 coord_fixed()+theme(panel.background=element_blank(), panel.
 ↪border=element_rect(color = "black", fill = NA),
axis.text.x = element_text(angle = 45, hjust = 1),legend.position="right")
```

0.19.1 plot

[120]:
```
ggarrange(grey1,grey2,
          labels = c("A", "B"),
           ncol = 2, nrow = 1)

ggsave(paste(path_fig,"Figure_2.pdf", sep=""), device= cairo_pdf,units='mm',␣
 ↪width=max_w, height=max_h/2.5)
```



0.20 Figure S5

total chlorine at sites in DWDS F over time

[121]:
```
w2=6
h2=4

options(repr.plot.width = w2, repr.plot.height = h2) #for plotting size in␣
↪jupyter
theme_set(theme_classic(base_size=10, base_family="Arial"))#, base_line_size=␣
↪1))
```

[122]:
```
a=DWDS_F_dat
a$location_code<-as.factor(a$site)
a$location_code<-with(a, reorder(x = site , X= total_Cl2_mg.L, FUN =␣
↪function(m) sd(m, na.rm = TRUE)))

ggplot(a, aes(x=site, y=total_Cl2_mg.L, color= water_age_h))+
geom_boxplot()+
    stat_summary(fun.data = give.n, geom = "text", position =␣
↪position_dodge(width = 0.75))  +
   ylab("total chlorine (mg/L)")+
   xlab("site")+
        theme(panel.background=element_blank(), panel.border=element_rect(color␣
↪= "black", fill = NA),
     axis.text.x = element_text(angle = 45, hjust = 1),legend.position="right")

ggsave(paste(path_fig,"Figure_S5.pdf",sep=""), device= cairo_pdf, units='mm',␣
↪width=max_w, height=(max_h/2.5))
```