Comprehensive Random Analysis of Covid-19

2023-01-24

```
library(data.table)
load("../../dhs_code/Normalization/data/data.H12.backup.RData")
'%!in%' <- function(x,y)!('%in%'(x,y))
'%!like%' <- function(x,y)!('%like%'(x,y))
data.H12<-data.H12.backup
### Prep H12 data
names(data.H12) <- c("wwtp_name", "epaid", "zipcode", "county_names", "state", "capacity_mgd", "populati

# Clean samples
data.H12 <- data.H12 %>% filter(wwtp_name %!like% "Madison-P")
data.H12 <- data.H12 %>% filter(wwtp_name %!in% c("Keshena", "Lac du Flambeau", "Menomonie", "Mondovi",
data.H12 <- data.H12 %>% filter(wwtp_name != "")

# Fix wwtp names
data.H12$wwtp_name <- gsub(" WWTP", "", data.H12$wwtp_name)
data.H12$wwtp_name <- gsub("WWTP", "", data.H12$wwtp_name)
data.H12$wwtp_name <- gsub("WWTF", "", data.H12$wwtp_name)
data.H12$wwtp_name <- gsub("BlackRiverFalls", "Black River Falls", data.H12$wwtp_name)
data.H12$wwtp_name <- gsub(" Metro", "", data.H12$wwtp_name)
data.H12$wwtp_name <- gsub("WI Rapids", "Wisconsin Rapids", data.H12$wwtp_name)
#print("List wwtps investigated (in H12 extract):")
#levels(as.factor(data.H12$wwtp_name))

# Add ID
data.H12$sample_collect_date<-as.Date(data.H12$sample_collect_date, format="%m/%d/%Y")
data.H12$ID<-paste0(data.H12$wwtp_name, data.H12$sample_collect_date)


data.meta<-read.table("../../dhs_code/Normalization/data/WWDataRequestDNR.csv", sep = ",", h=T)
data.meta<-data.meta %>% filter(lab_submitter == "SLH")
### Preparation of metadata
# Fix wwtp names
data.meta$wwtp_name<-gsub(" WWTF", "", data.meta$wwtp_name)
data.meta$wwtp_name<-gsub(" WWTP", "", data.meta$wwtp_name)
data.meta$wwtp_name<-gsub(" Sewage Utility", "", data.meta$wwtp_name)
data.meta$wwtp_name<-gsub(" WW Utility", "", data.meta$wwtp_name)
data.meta$wwtp_name<-gsub(" Wastewater Utility", "", data.meta$wwtp_name)
data.meta$wwtp_name<-gsub(" WPCF", "", data.meta$wwtp_name)
data.meta$wwtp_name<-gsub(" MSD", "", data.meta$wwtp_name)
data.meta$wwtp_name<-gsub(" Utilities", "", data.meta$wwtp_name)
data.meta$wwtp_name<-gsub(" Municipal Utility", "", data.meta$wwtp_name)
data.meta$wwtp_name<-gsub(" Water Works", "", data.meta$wwtp_name)
#levels(as.factor(data.meta$wwtp_name))
#print("List wwtps with metadata (should be the same than above):")
```

```r
#levels(as.factor(data.H12$wwtp_name))

# Add ID
data.meta$sample_collect_date <- as.Date(data.meta$sample_collect_date, format="%m/%d/%Y")
data.meta$ID <- paste0(data.meta$wwtp_name, data.meta$sample_collect_date)

# Metadata #1
data.meta.1 <- unique(data.meta[, c("ID", "sars_cov2_adj_load", "cases", "case_rate")])

# Metadata #2
data.meta.2 <- data.meta[, c("ID", "result_amt", "storet_parm_desc", "parm_unit_type")]
data.meta.2$storet_parm_desc <- gsub("CBOD5", "BOD5", data.meta.2$storet_parm_desc)
data.meta.2$storet_parm_desc <- gsub("BOD5, Total", "BOD5", data.meta.2$storet_parm_desc)
data.meta.2$storet_parm_desc <- gsub("Suspended Solids, Total", "TSS", data.meta.2$storet_parm_desc)
data.meta.2 <- reshape2::dcast(data.meta.2, ID~storet_parm_desc, value.var = "result_amt", fun.aggregat
data.meta.2 <- data.meta.2 %>% select("ID", "BOD5", "Flow Rate", "TSS")

# Metadata - merged #1 and #2
data.meta <- left_join(data.meta.1, data.meta.2, by="ID")

data <- inner_join(data.H12, data.meta, by="ID")
#data<-data.H12

data <-data %>% filter(pcr_type == "qPCR")

data$avg_sars_cov2_conc <- as.numeric(as.character(data$avg_sars_cov2_conc))
data$average_flow_rate <- as.numeric(as.character(data$average_flow_rate))
data$population_served <- as.numeric(as.character(data$population_served))
data$pmmov_conc <- as.numeric(as.character(data$pmmov_conc))
data$hf183_conc <- as.numeric(as.character(data$hf183_conc))
data$bcov_rec_rate <- as.numeric(as.character(data$bcov_rec_rate))
data$tss <- as.numeric(as.character(data$tss))
data$ph <- as.numeric(as.character(data$ph))
data$temperature <- as.numeric(as.character(data$temperature))
data$conductivity <- as.numeric(as.character(data$conductivity))
data$sars_cov2_adj_load <- as.numeric(as.character(data$sars_cov2_adj_load))
data$cases <- as.numeric(as.character(data$cases))
data$case_rate <- as.numeric(as.character(data$case_rate))
data$BOD5 <- as.numeric(as.character(data$BOD5))
data$`Flow Rate` <- as.numeric(as.character(data$`Flow Rate`))
data$TSS <- as.numeric(as.character(data$TSS))

data$wwtp_name <- as.factor(data$wwtp_name)

#data <- data%>%
#  mutate(across(c("pmmov_conc", "bcov_rec_rate", "average_flow_rate", "TSS") , ~log(.x)))%>%
#  filter(across(c("pmmov_conc", "bcov_rec_rate", "average_flow_rate", "TSS") , ~is.finite(.x)))

####GROSS CODE
# Difference between value and average of the value for a given WWTP

data.select.1 <- data%>%
  select(wwtp_name, case_rate, avg_sars_cov2_conc,everything())
```

```
#names(data.select.1)[1:3] <- c("wwtp_name", "case_rate", "avg_sars_cov2_conc")


mean(log(data.select.1$n1_lod), na.rm = TRUE)
mean(data.select.1$n1_lod, na.rm = TRUE)
```

```
lm_plotly_model <- function(DF){
  plotly_plot <- DF%>%
  ggplot(aes(x = (avg_sars_cov2_conc),
             y = (case_rate),
             color = n1_sars_cov2_lod,
             fill = n2_sars_cov2_lod))+
  geom_point()+
  geom_smooth()+
  ggtitle("Strong Linear Relationship Between Log Cases and Log COVID Concentration")+
  xlab("log(Covid-19 concentration)")+
  ylab("log(case rate)")

  return(plotly::ggplotly(plotly_plot))#
}
```

```
library(DSIWastewater)
data("Case_data", package = "DSIWastewater")
data("WasteWater_data", package = "DSIWastewater")

roll_Case_data <- Case_data%>%
  mutate(case_rate = conf_case + prob_death)%>%
  group_by(site)%>%
  arrange(date)%>%
  mutate(case_rate = zoo::rollmean(case_rate, 7, fill = NA, align = "right"))


data.select.1 <- WasteWater_data%>%
  mutate(avg_sars_cov2_conc = exp(.5*(log(N1+2) + log(N2+2))))%>%
  left_join(roll_Case_data )%>%
  rename(sample_collect_date = date,
         wwtp_name = site)%>%
  select(-pop)
```

```
library(DSIWastewater)
data(pop_data, package = "DSIWastewater")
data(Covariants_data, package = "DSIWastewater")
Graph_DF <- data.select.1%>%
  gen_Variant_info("sample_collect_date")%>%#join variant data
  left_join(rename(pop_data, wwtp_name = site), by = c("wwtp_name"))%>%#join pop data
  filter(case_rate != 0,
         avg_sars_cov2_conc != 0,
         !is.na(pop))%>%
  mutate(flow_avg_conc = log1p(flow * avg_sars_cov2_conc),
         avg_sars_cov2_conc = log1p(avg_sars_cov2_conc),
         case_rate = log(case_rate + 1),
```

```
        LOD = (n1_sars_cov2_lod == "No") & (n2_sars_cov2_lod == "No"))

Graph_DF$pop_group <- as.factor(ntile(Graph_DF$pop, 4))



df_LOD <- Graph_DF%>%
  filter(!n1_sars_cov2_lod & !n2_sars_cov2_lod)



summary(lm(case_rate ~ avg_sars_cov2_conc, data = df_LOD))
```

```
##
## Call:
## lm(formula = case_rate ~ avg_sars_cov2_conc, data = df_LOD)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6110 -1.0416 -0.1479  0.9475  4.6796
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.11844    0.16972   0.698    0.485
## avg_sars_cov2_conc  0.19096    0.01373  13.906   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.276 on 6327 degrees of freedom
## Multiple R-squared:  0.02966,    Adjusted R-squared:  0.0295
## F-statistic: 193.4 on 1 and 6327 DF,  p-value: < 2.2e-16
```

```
summary(lm(case_rate ~ flow_avg_conc, data = df_LOD))
```

```
##
## Call:
## lm(formula = case_rate ~ flow_avg_conc, data = df_LOD)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1133 -0.5612 -0.0917  0.4737  6.0000
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.74411    0.07616  -62.29   <2e-16 ***
## flow_avg_conc  0.53052    0.00555   95.60   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8282 on 6327 degrees of freedom
## Multiple R-squared:  0.5909, Adjusted R-squared:  0.5908
## F-statistic:  9139 on 1 and 6327 DF,  p-value: < 2.2e-16
```

```r
summary(lm(case_rate ~ avg_sars_cov2_conc:pop_group, data = df_LOD))
```

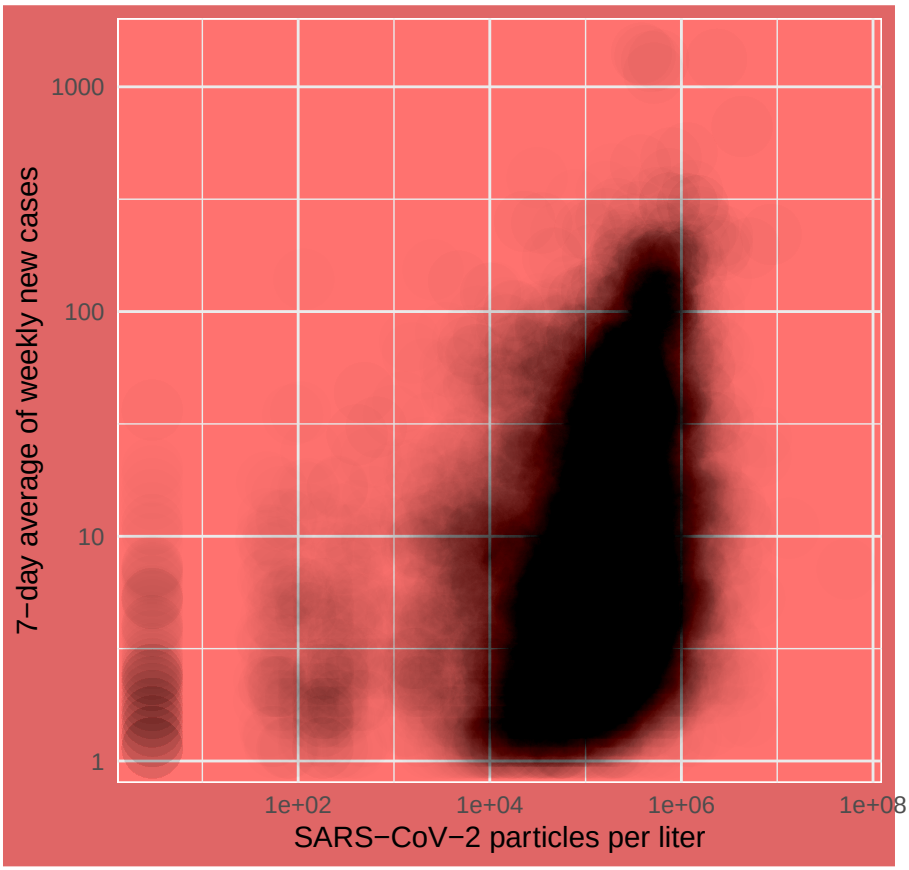
```
##
## Call:
## lm(formula = case_rate ~ avg_sars_cov2_conc:pop_group, data = df_LOD)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4692 -0.5295 -0.0220  0.4827  3.2351
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -0.508338   0.103503  -4.911 9.27e-07 ***
## avg_sars_cov2_conc:pop_group1  0.141261   0.008346  16.926  < 2e-16 ***
## avg_sars_cov2_conc:pop_group2  0.191234   0.008546  22.377  < 2e-16 ***
## avg_sars_cov2_conc:pop_group3  0.271754   0.008576  31.687  < 2e-16 ***
## avg_sars_cov2_conc:pop_group4  0.352087   0.008482  41.511  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7734 on 6324 degrees of freedom
## Multiple R-squared:  0.6434, Adjusted R-squared:  0.6432
## F-statistic:  2853 on 4 and 6324 DF,  p-value: < 2.2e-16
```

```r
summary(lm(case_rate ~ flow_avg_conc:pop_group, data = df_LOD))
```

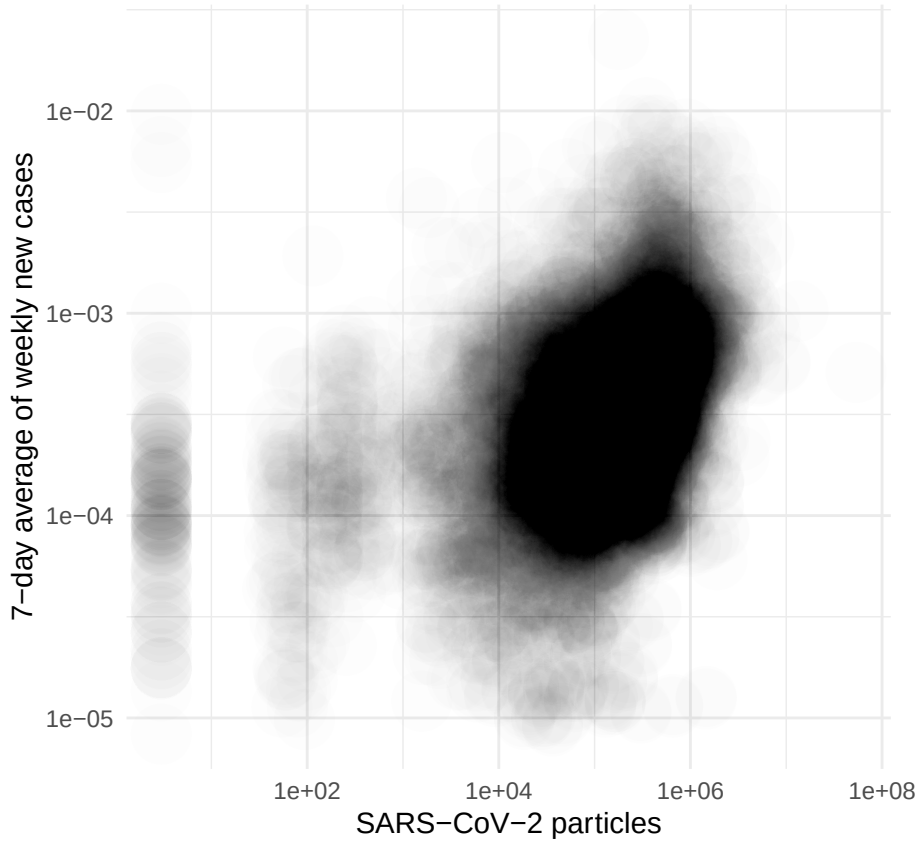
```
##
## Call:
## lm(formula = case_rate ~ flow_avg_conc:pop_group, data = df_LOD)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0409 -0.4944 -0.0497  0.4215  3.2404
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -1.681951   0.098044  -17.16   <2e-16 ***
## flow_avg_conc:pop_group1   0.246145   0.008263   29.79   <2e-16 ***
## flow_avg_conc:pop_group2   0.275346   0.007773   35.42   <2e-16 ***
## flow_avg_conc:pop_group3   0.322068   0.007113   45.28   <2e-16 ***
## flow_avg_conc:pop_group4   0.354316   0.006362   55.69   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.728 on 6324 degrees of freedom
## Multiple R-squared:  0.6841, Adjusted R-squared:  0.6839
## F-statistic:  3423 on 4 and 6324 DF,  p-value: < 2.2e-16
```

```r
size = 10
alpha = .008
#Graph_DF,  df_LOD
Graph_DF%>%
```

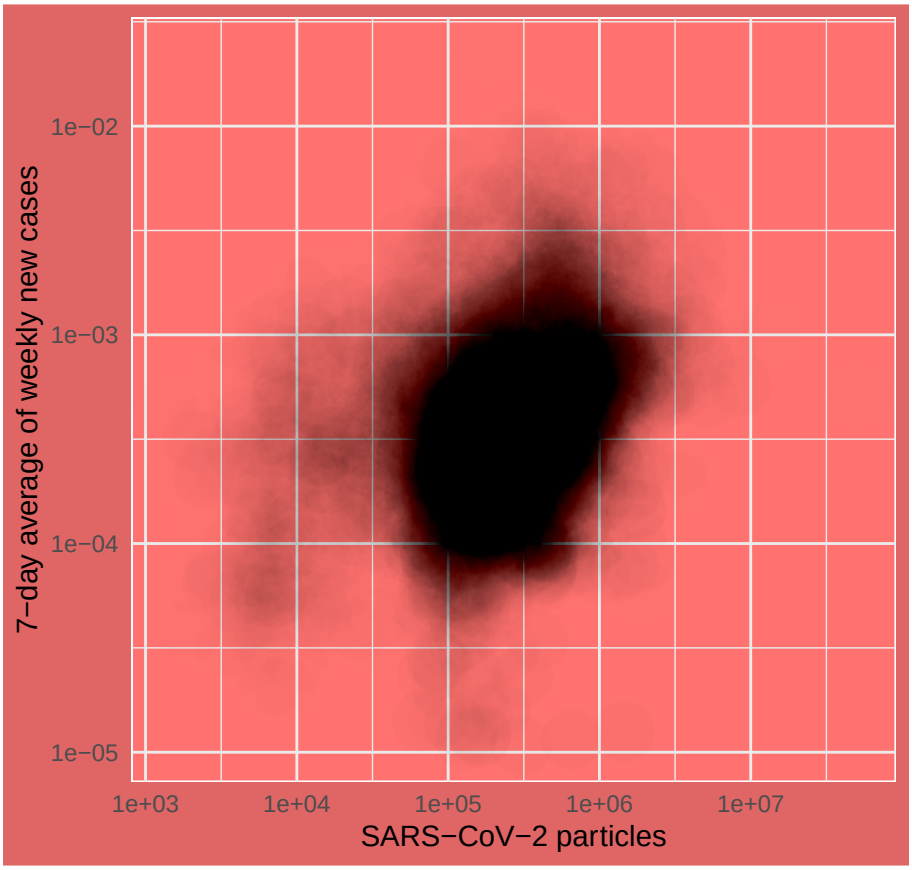
```
  ggplot(aes(x = exp(avg_sars_cov2_conc), y = exp(case_rate)))+
  geom_point(size = size, alpha = alpha, color = "black")+
  scale_y_log10()+
  scale_x_log10()+
  theme_minimal()+
  theme(panel.background = element_rect(fill = '#ff726f', color = 'white'),
        plot.background = element_rect(fill = '#e06666ff', color = 'white'),
        aspect.ratio=1)+
  labs(y = "7-day average of weekly new cases", x = "SARS-CoV-2 particles per liter")
```



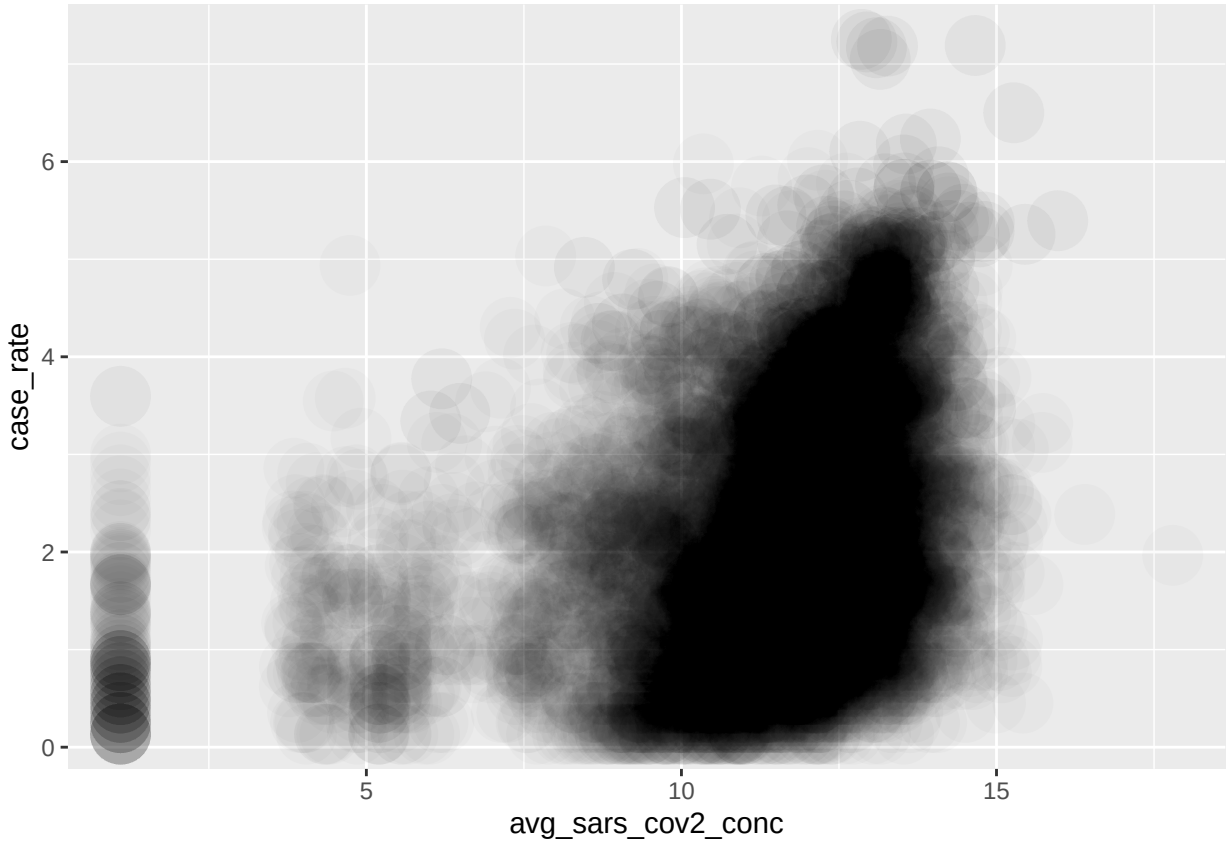

```
Graph_DF%>%
  ggplot(aes(x = exp(avg_sars_cov2_conc), y = exp(case_rate) / pop))+
  geom_point(size = size, alpha = alpha, color = "black")+
  scale_y_log10()+
  scale_x_log10()+
  theme_minimal()+
  theme(aspect.ratio=1)+
  labs(y = "7-day average of weekly new cases", x = "SARS-CoV-2 particles")
```




```
df_LOD%>%
  filter(flow_avg_conc > 5)%>%
  ggplot(aes(x = exp(avg_sars_cov2_conc), y = exp(case_rate) / pop))+
  geom_point(size = size, alpha = alpha)+
  scale_y_log10()+
  scale_x_log10()+
  theme_minimal()+
  theme(panel.background = element_rect(fill = '#ff726f', color = 'white'),
        plot.background = element_rect(fill = '#e06666ff', color = 'white'),
        aspect.ratio=1)+
  labs(y = "7-day average of weekly new cases", x = "SARS-CoV-2 particles")
```



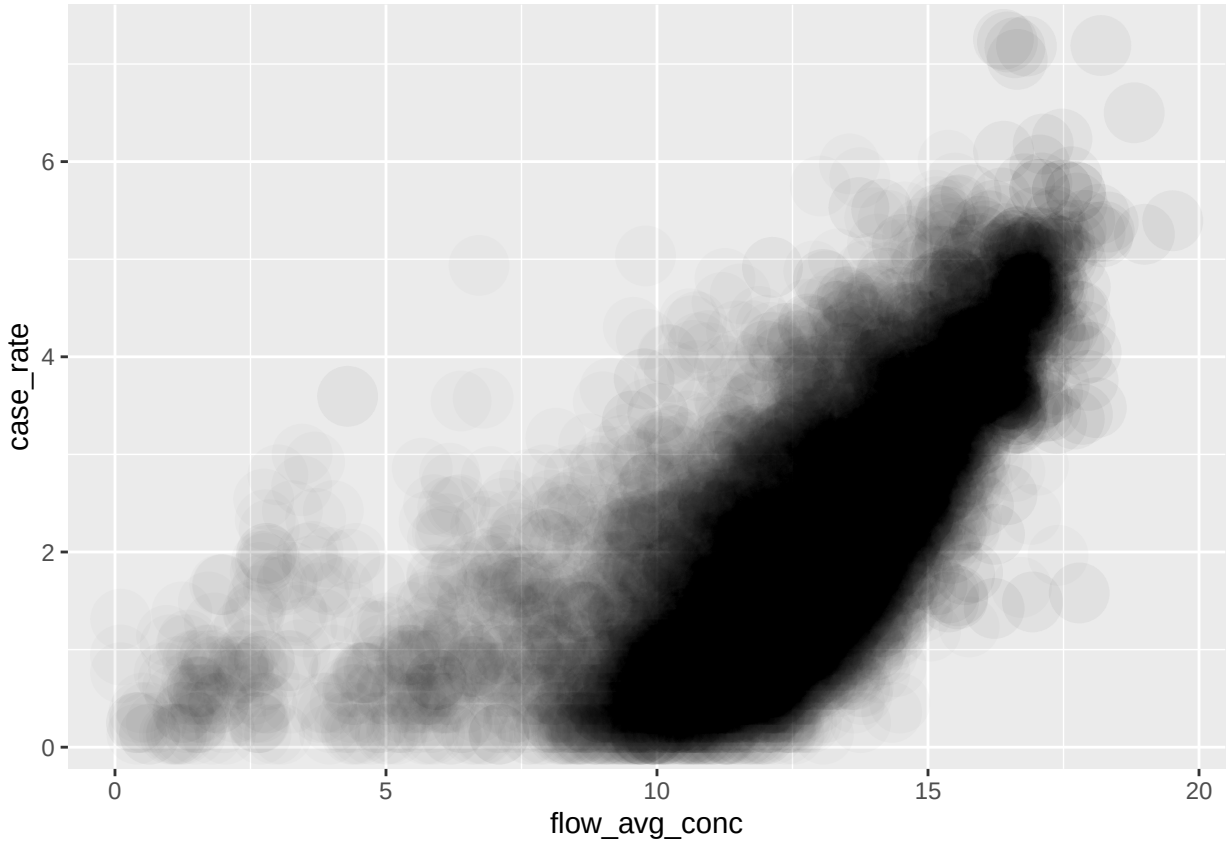

```
size = 10
alpha = .02
#Graph_DF,  df_LOD
Graph_DF%>%
  ggplot(aes(x = avg_sars_cov2_conc, y = case_rate))+
  geom_point(size = size, alpha = alpha)
```



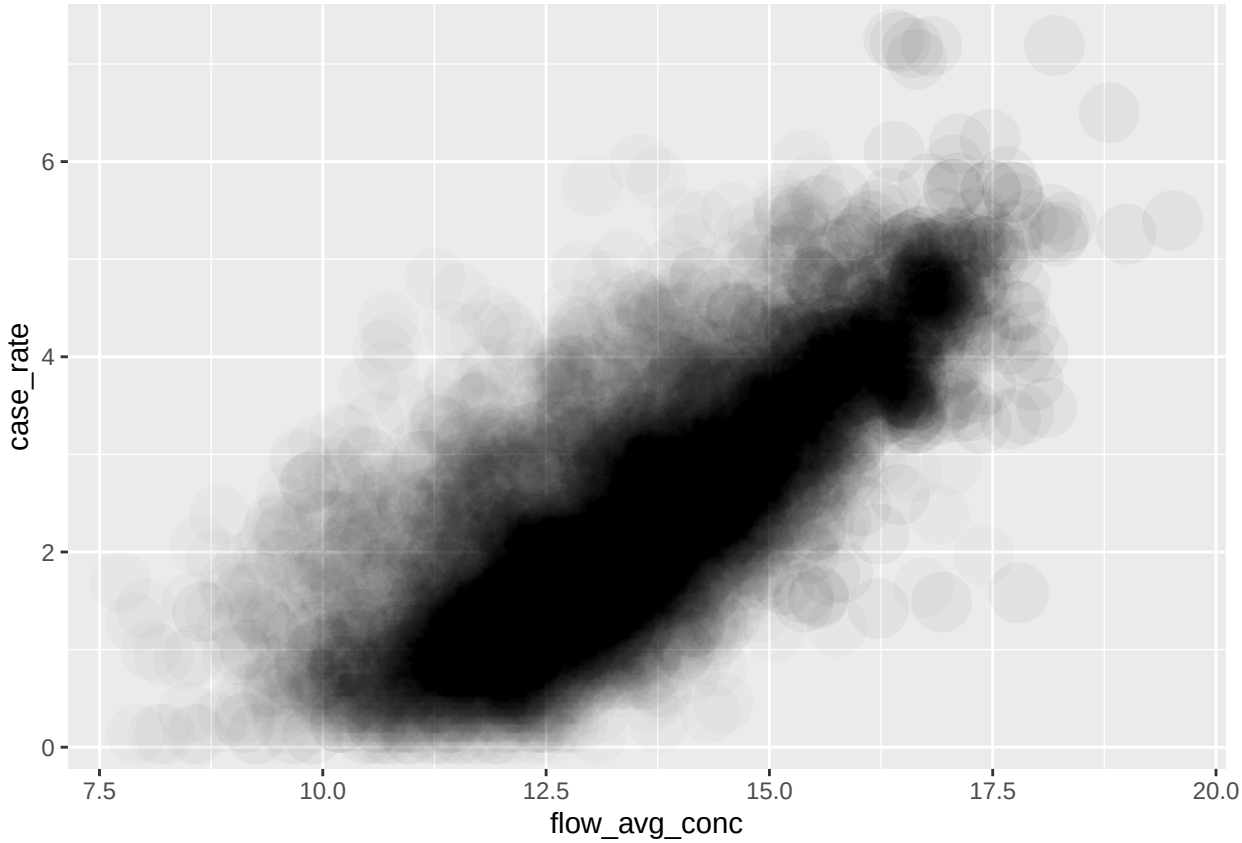

```
summary(lm(case_rate ~ avg_sars_cov2_conc, data = Graph_DF))
```

```
##
## Call:
## lm(formula = case_rate ~ avg_sars_cov2_conc, data = Graph_DF)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4786 -1.0003 -0.1675  0.8605  4.7966
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.079002   0.065870   1.199     0.23
## avg_sars_cov2_conc 0.184926   0.005678  32.571   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.233 on 9012 degrees of freedom
## Multiple R-squared:  0.1053, Adjusted R-squared:  0.1052
## F-statistic:  1061 on 1 and 9012 DF,  p-value: < 2.2e-16
```

```
Graph_DF%>%
  ggplot(aes(x = flow_avg_conc, y = case_rate))+
  geom_point(size = size, alpha = alpha)
```



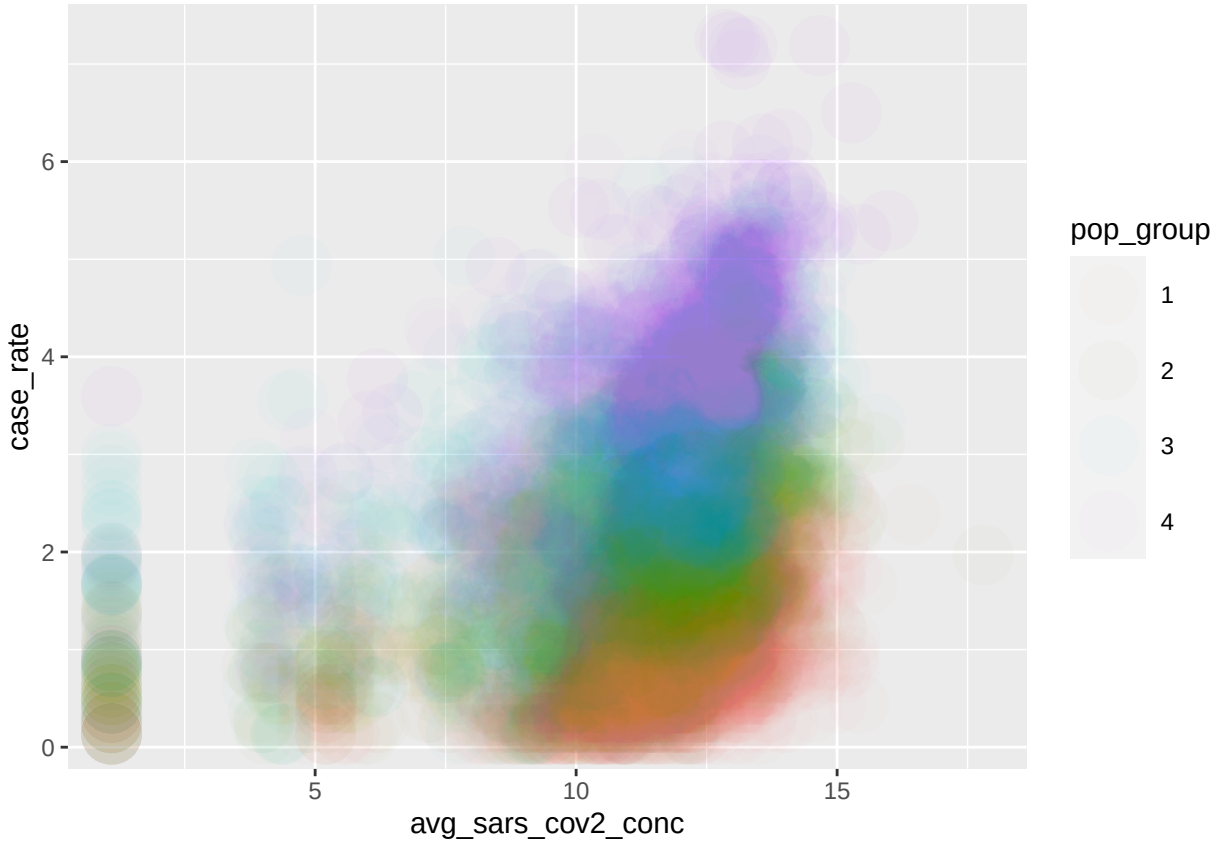

```
df_LOD%>%
  filter(flow_avg_conc > 5)%>%
  ggplot(aes(x = flow_avg_conc, y = case_rate))+
  geom_point(size = size, alpha = alpha)
```




```
summary(lm(case_rate ~ flow_avg_conc, data = df_LOD))
```

```
##
## Call:
## lm(formula = case_rate ~ flow_avg_conc, data = df_LOD)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1133 -0.5612 -0.0917  0.4737  6.0000
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.74411    0.07616  -62.29   <2e-16 ***
## flow_avg_conc  0.53052    0.00555   95.60   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8282 on 6327 degrees of freedom
## Multiple R-squared:  0.5909, Adjusted R-squared:  0.5908
## F-statistic:  9139 on 1 and 6327 DF,  p-value: < 2.2e-16
```

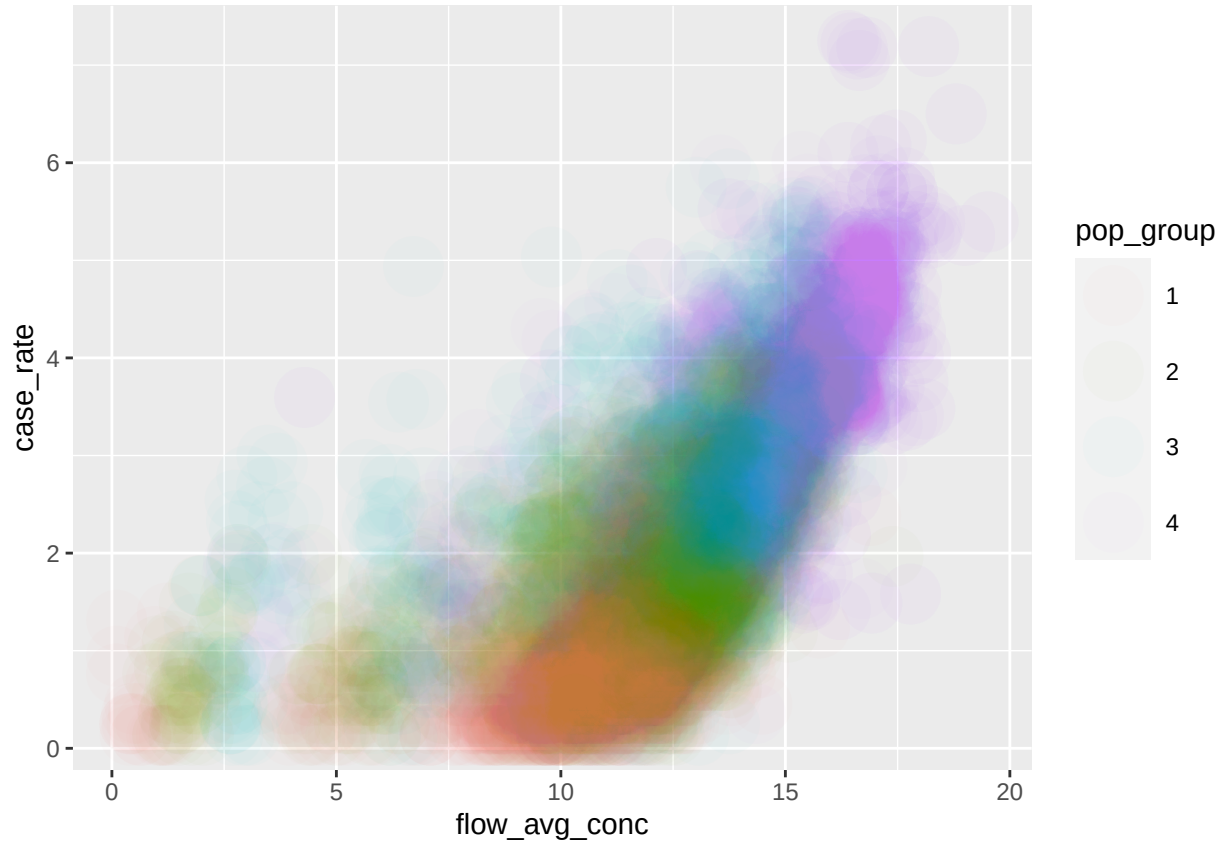
```
Graph_DF%>%
  ggplot(aes(x = avg_sars_cov2_conc, y = case_rate, color = pop_group))+
  geom_point(size = size, alpha = alpha)
```




```
summary(lm(case_rate ~ avg_sars_cov2_conc:pop_group, data = df_LOD))
```

```
##
## Call:
## lm(formula = case_rate ~ avg_sars_cov2_conc:pop_group, data = df_LOD)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4692 -0.5295 -0.0220  0.4827  3.2351
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -0.508338   0.103503  -4.911 9.27e-07 ***
## avg_sars_cov2_conc:pop_group1  0.141261   0.008346  16.926  < 2e-16 ***
## avg_sars_cov2_conc:pop_group2  0.191234   0.008546  22.377  < 2e-16 ***
## avg_sars_cov2_conc:pop_group3  0.271754   0.008576  31.687  < 2e-16 ***
## avg_sars_cov2_conc:pop_group4  0.352087   0.008482  41.511  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7734 on 6324 degrees of freedom
## Multiple R-squared:  0.6434, Adjusted R-squared:  0.6432
## F-statistic:  2853 on 4 and 6324 DF,  p-value: < 2.2e-16
```

```
Graph_DF%>%
  ggplot(aes(x = flow_avg_conc, y = case_rate, color = pop_group))+
  geom_point(size = size, alpha = alpha)
```

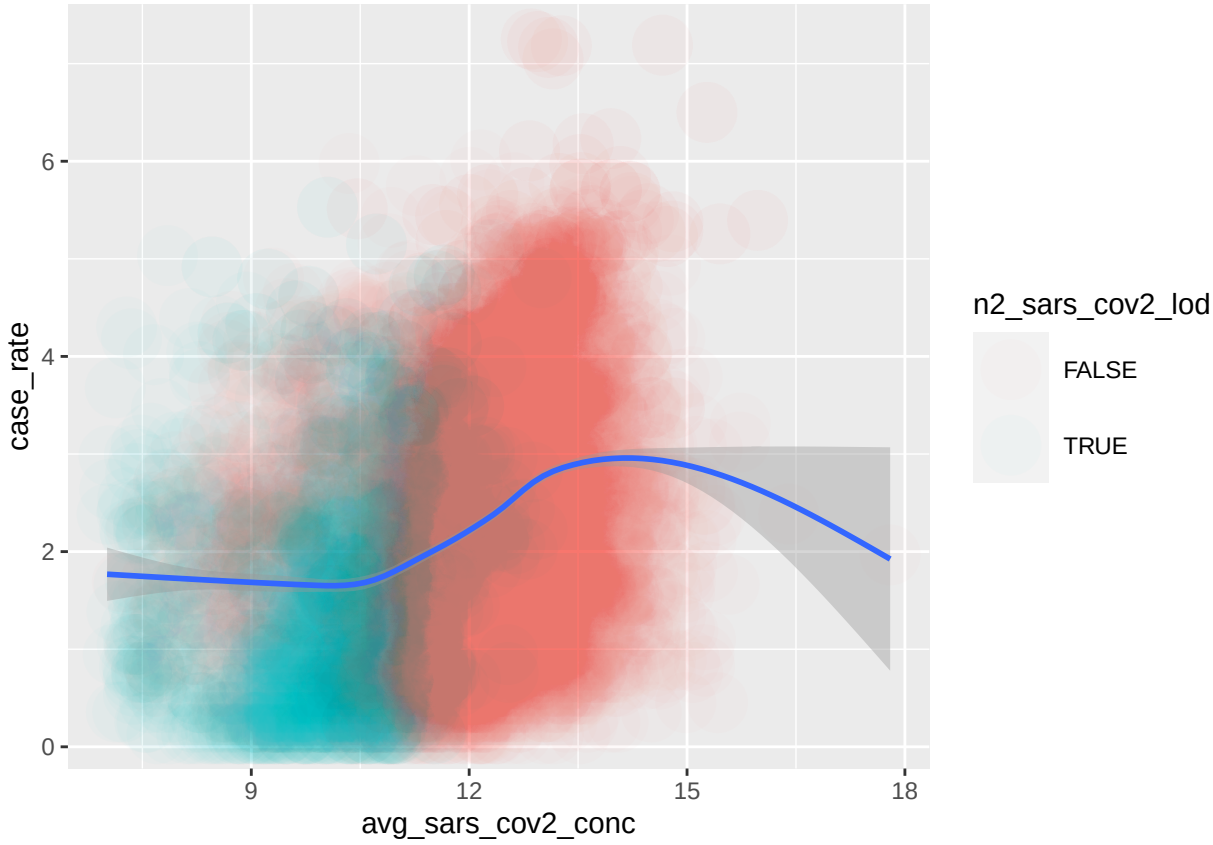



```
summary(lm(case_rate ~ flow_avg_conc:pop_group, data = df_LOD))
```

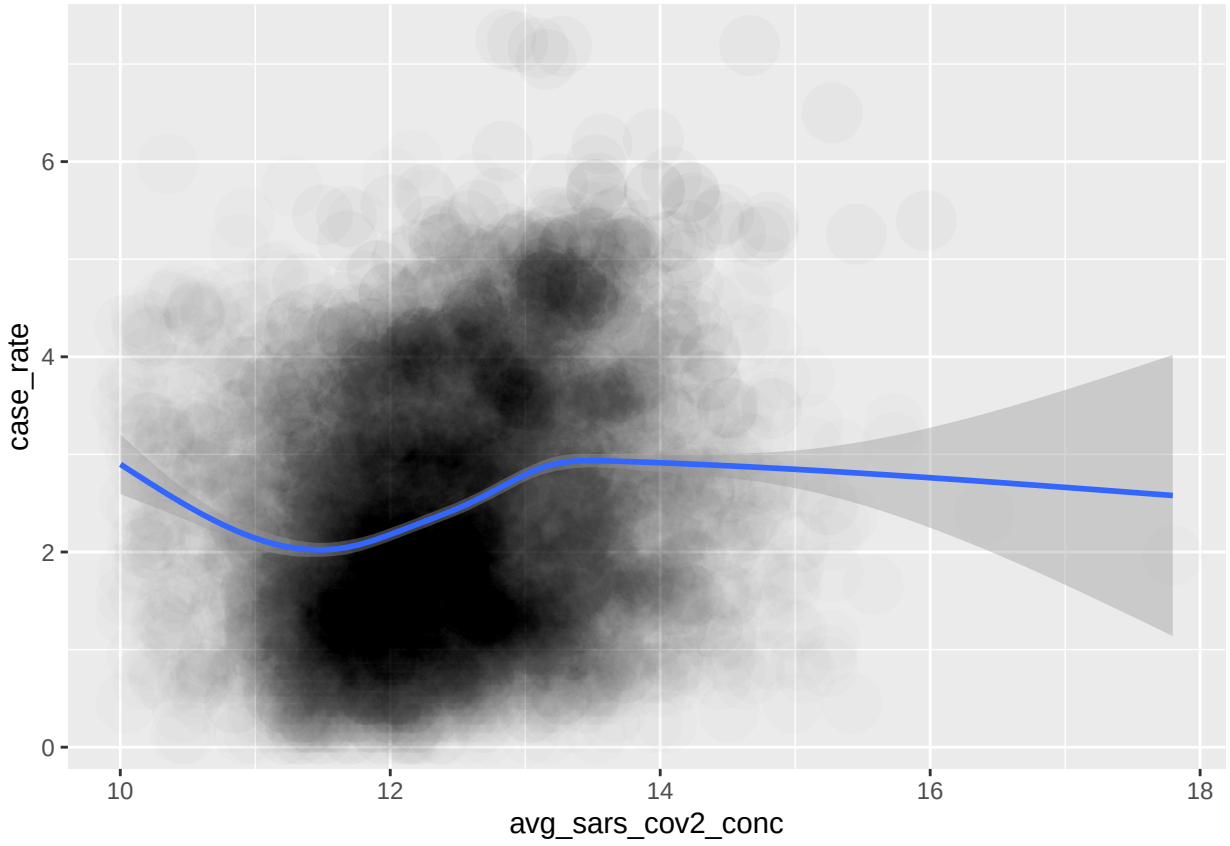
```
##
## Call:
## lm(formula = case_rate ~ flow_avg_conc:pop_group, data = df_LOD)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0409 -0.4944 -0.0497  0.4215  3.2404
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -1.681951   0.098044  -17.16   <2e-16 ***
## flow_avg_conc:pop_group1   0.246145   0.008263   29.79   <2e-16 ***
## flow_avg_conc:pop_group2   0.275346   0.007773   35.42   <2e-16 ***
## flow_avg_conc:pop_group3   0.322068   0.007113   45.28   <2e-16 ***
## flow_avg_conc:pop_group4   0.354316   0.006362   55.69   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.728 on 6324 degrees of freedom
```

```
## Multiple R-squared:  0.6841, Adjusted R-squared:  0.6839
## F-statistic:  3423 on 4 and 6324 DF,  p-value: < 2.2e-16
```

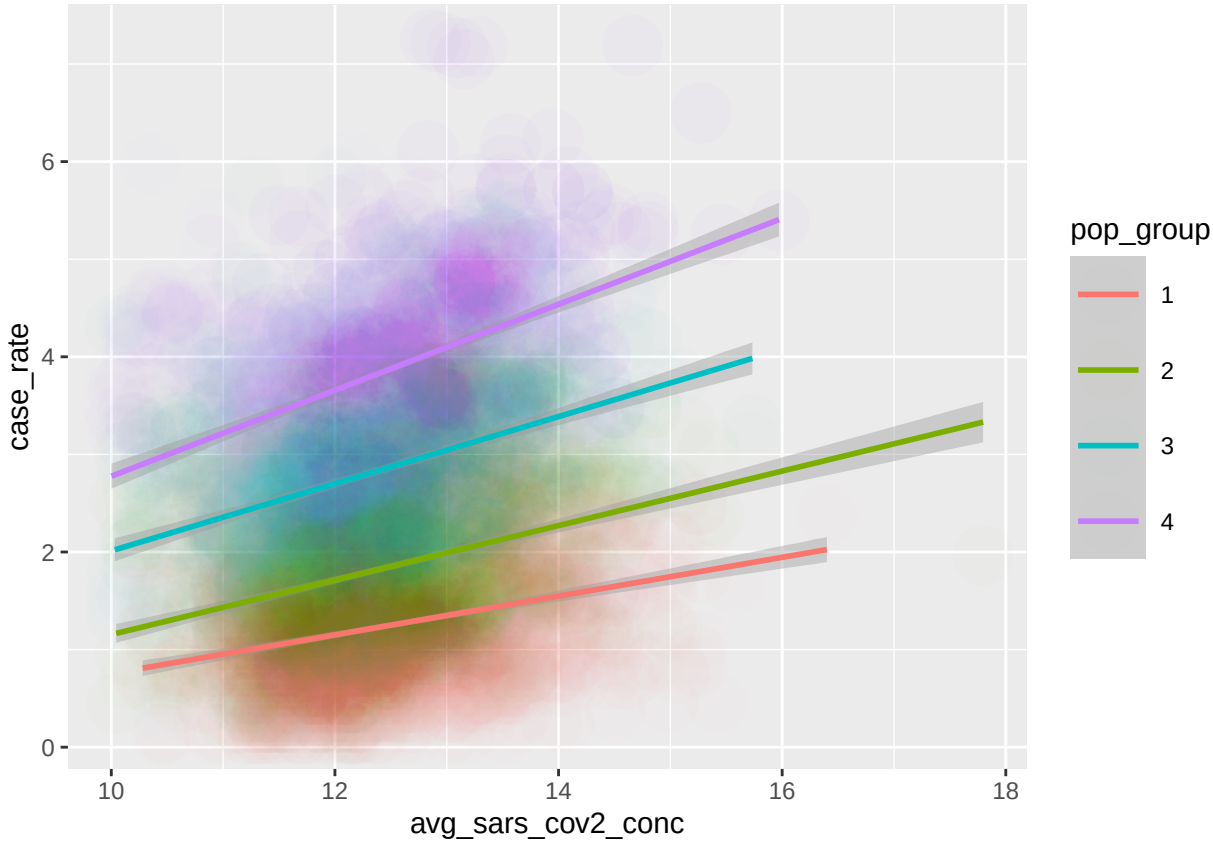
```
Graph_DF%>%#n2_sars_cov2_lod
  filter(7 < avg_sars_cov2_conc)%>%
  ggplot(aes(x = avg_sars_cov2_conc, y = case_rate))+
  geom_point(aes(color = n2_sars_cov2_lod), size = 10, alpha = .02)+
  geom_smooth()
```




```
df_LOD%>%#n2_sars_cov2_lod
  filter(avg_sars_cov2_conc > 10)%>%
  ggplot(aes(x = avg_sars_cov2_conc, y = case_rate))+
  geom_point(size = 10, alpha = .01)+
  geom_smooth()
```




```
df_LOD%>%#n2_sars_cov2_lod
  filter(avg_sars_cov2_conc > 10)%>%
  ggplot(aes(x = avg_sars_cov2_conc, y = case_rate, color = pop_group))+
  geom_point(size = 10, alpha = .01)+
  geom_smooth(method = "lm")
```




```
best_mod <- lm(case_rate ~ avg_sars_cov2_conc:pop_group, data = df_LOD)

summary(lm(case_rate ~ avg_sars_cov2_conc, data = Graph_DF))
```

```
##
## Call:
## lm(formula = case_rate ~ avg_sars_cov2_conc, data = Graph_DF)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4786 -1.0003 -0.1675  0.8605  4.7966
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.079002   0.065870   1.199     0.23
## avg_sars_cov2_conc 0.184926   0.005678  32.571   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.233 on 9012 degrees of freedom
## Multiple R-squared:  0.1053, Adjusted R-squared:  0.1052
## F-statistic:  1061 on 1 and 9012 DF,  p-value: < 2.2e-16
```

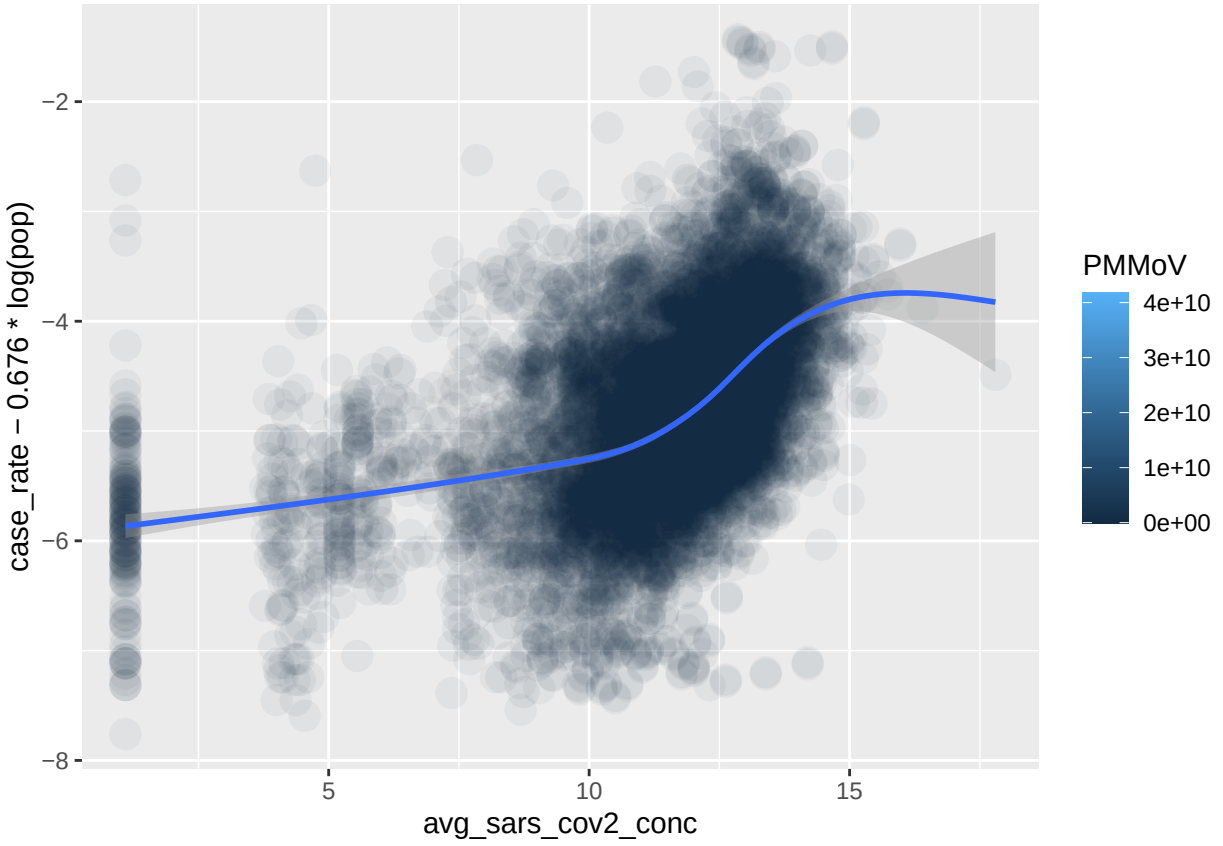
```
summary(lm(case_rate ~ avg_sars_cov2_conc, data = df_LOD))
```

```
##
## Call:
## lm(formula = case_rate ~ avg_sars_cov2_conc, data = df_LOD)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6110 -1.0416 -0.1479  0.9475  4.6796
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.11844    0.16972   0.698    0.485
## avg_sars_cov2_conc   0.19096    0.01373  13.906   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.276 on 6327 degrees of freedom
## Multiple R-squared:  0.02966,    Adjusted R-squared:  0.0295
## F-statistic: 193.4 on 1 and 6327 DF,  p-value: < 2.2e-16
```

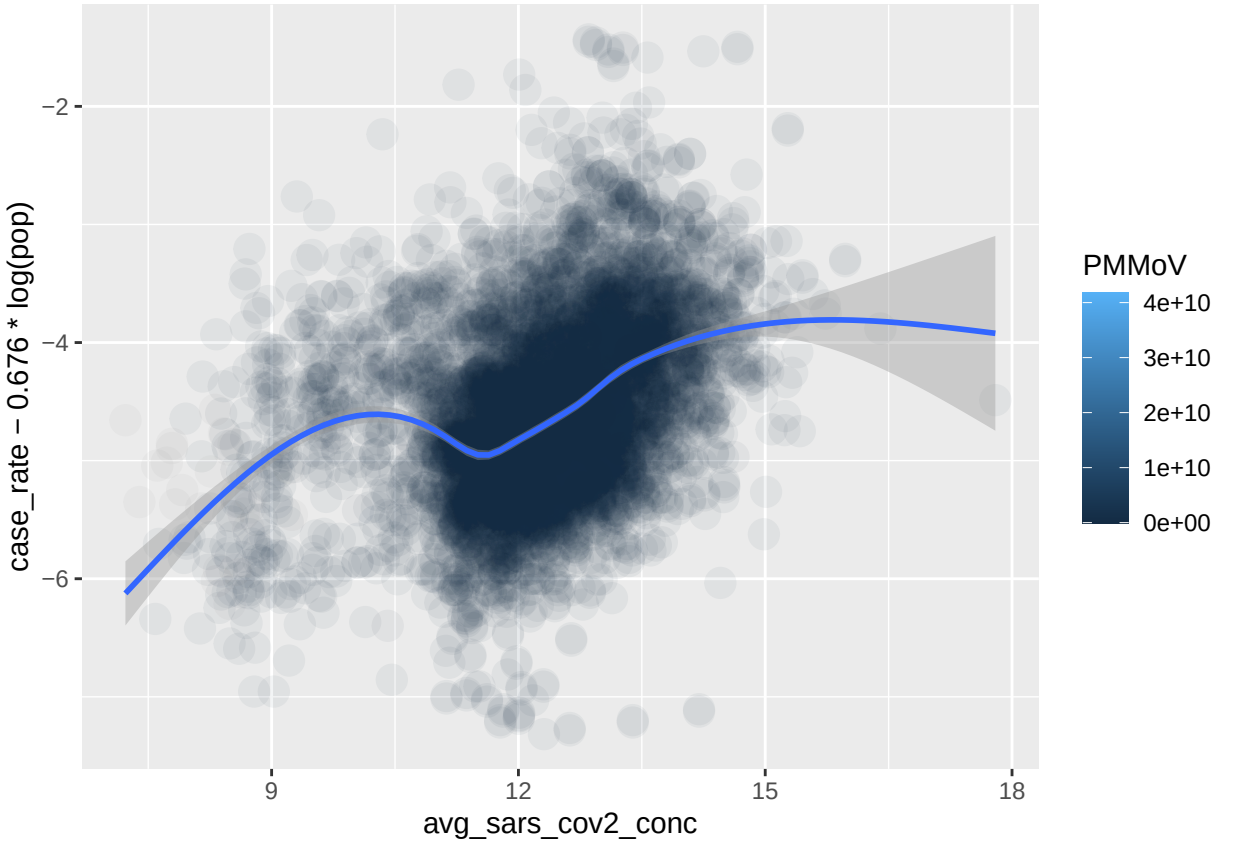
```
summary(lm(case_rate ~ avg_sars_cov2_conc + pop_group, data = df_LOD))
```

```
##
## Call:
## lm(formula = case_rate ~ avg_sars_cov2_conc + pop_group, data = df_LOD)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4628 -0.5369 -0.0232  0.4875  3.2967
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -1.886231   0.107872  -17.49   <2e-16 ***
## avg_sars_cov2_conc   0.250857   0.008455   29.67   <2e-16 ***
## pop_group2           0.654168   0.028213   23.19   <2e-16 ***
## pop_group3           1.635104   0.028399   57.58   <2e-16 ***
## pop_group4           2.617573   0.027218   96.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7796 on 6324 degrees of freedom
## Multiple R-squared:  0.6377, Adjusted R-squared:  0.6374
## F-statistic:  2782 on 4 and 6324 DF,  p-value: < 2.2e-16
```

```
Graph_DF%>%
  ggplot(aes(x = avg_sars_cov2_conc, y = case_rate - .676*log(pop)))+
  geom_point(aes(color = PMMoV), size = 5, alpha = .05)+
  geom_smooth()
```




```
df_LOD%>%
  filter(5 < avg_sars_cov2_conc)%>%
  ggplot(aes(x = avg_sars_cov2_conc, y = case_rate - .676*log(pop)))+
  geom_point(aes(color = PMMoV), size = 5, alpha = .05)+
  geom_smooth()
```




```
summary(lm(case_rate ~ avg_sars_cov2_conc + log(pop), data  = df_LOD))
```

```
##
## Call:
## lm(formula = case_rate ~ avg_sars_cov2_conc + log(pop), data = df_LOD)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9662 -0.4952 -0.0512  0.4306  4.3759
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -7.335831   0.116258  -63.10   <2e-16 ***
## avg_sars_cov2_conc  0.223741   0.007815   28.63   <2e-16 ***
## log(pop)            0.676036   0.005876  115.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7254 on 6326 degrees of freedom
## Multiple R-squared:  0.6862, Adjusted R-squared:  0.6861
## F-statistic:  6916 on 2 and 6326 DF,  p-value: < 2.2e-16
```

```
#log(case) = mlog(cov) + ylog(pop) + x
#log(case) - ylog(pop)
#log(case/pop)
```

```r
df_LOD%>%
  #filter(5 < avg_sars_cov2_conc)%>%
  mutate(n_case_rate = case_rate  - .676*log(pop))%>%
  lm(n_case_rate ~ avg_sars_cov2_conc + main_variant, data = .)%>%
  summary()
```

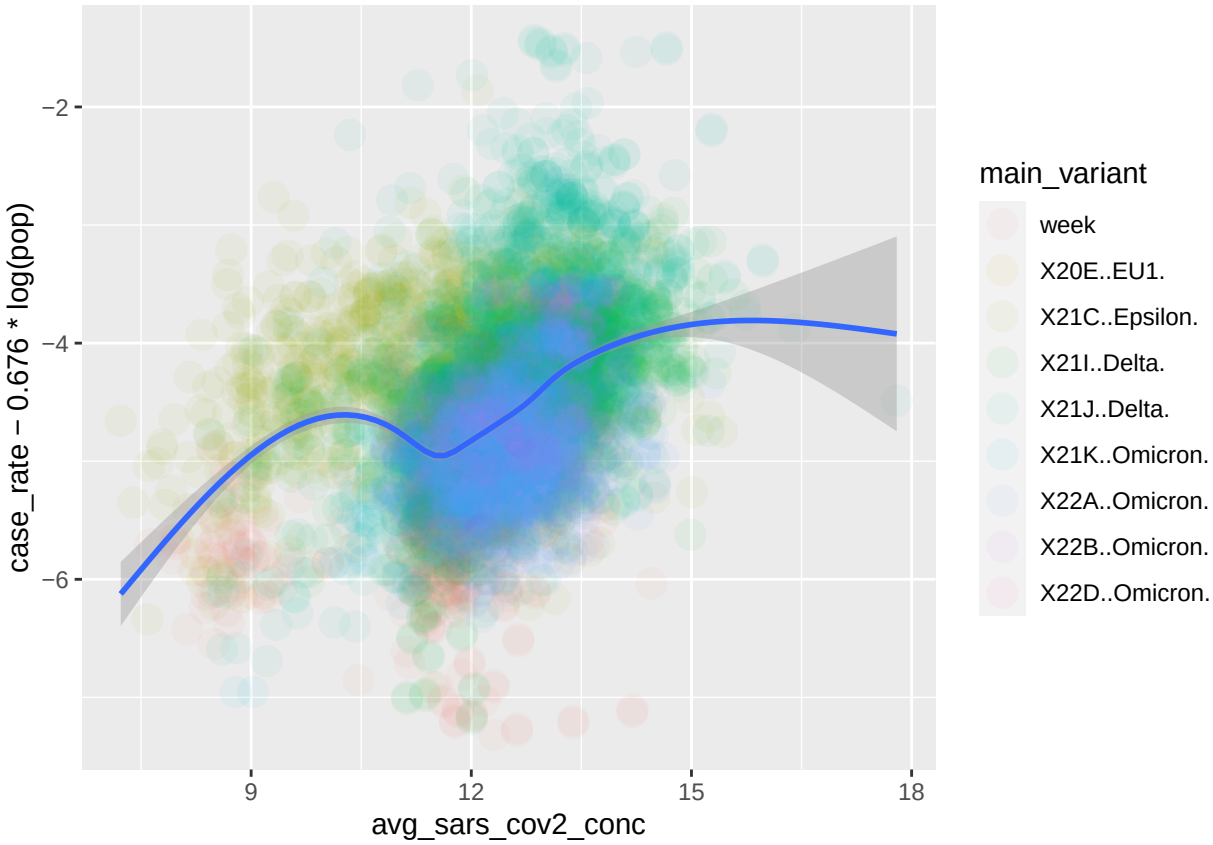
```
##
## Call:
## lm(formula = n_case_rate ~ avg_sars_cov2_conc + main_variant,
##     data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5369 -0.3475 -0.0009  0.3888  3.6082
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -7.988847   0.084980 -94.009  < 2e-16 ***
## avg_sars_cov2_conc          0.210653   0.006961  30.261  < 2e-16 ***
## main_variantX20E..EU1.      0.262835   0.064505   4.075 4.66e-05 ***
## main_variantX21C..Epsilon.  1.196394   0.033264  35.966  < 2e-16 ***
## main_variantX21I..Delta.    0.755421   0.033725  22.400  < 2e-16 ***
## main_variantX21J..Delta.    1.435257   0.036670  39.140  < 2e-16 ***
## main_variantX21K..Omicron.  0.717779   0.046805  15.336  < 2e-16 ***
## main_variantX22A..Omicron.  0.431566   0.033923  12.722  < 2e-16 ***
## main_variantX22B..Omicron.  0.874015   0.049896  17.517  < 2e-16 ***
## main_variantX22D..Omicron.  0.238599   0.201675   1.183    0.237
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5982 on 6319 degrees of freedom
## Multiple R-squared:  0.3989, Adjusted R-squared:  0.398
## F-statistic: 465.8 on 9 and 6319 DF,  p-value: < 2.2e-16
```

```r
mod_df <- df_LOD%>%
  mutate(n_case_rate = case_rate  - .676*log(pop))
plot_lm_model <- lm(n_case_rate ~ avg_sars_cov2_conc, data = mod_df)
summary(plot_lm_model)
```

```
##
## Call:
## lm(formula = n_case_rate ~ avg_sars_cov2_conc, data = mod_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9661 -0.4951 -0.0512  0.4306  4.3757
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -7.335432   0.096519  -76.00   <2e-16 ***
## avg_sars_cov2_conc  0.223739   0.007809   28.65   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7254 on 6327 degrees of freedom
## Multiple R-squared:  0.1148, Adjusted R-squared:  0.1147
## F-statistic: 820.8 on 1 and 6327 DF,  p-value: < 2.2e-16
```

```
(mod_df%>%
  modelr::add_predictions(plot_lm_model)%>%
  filter(5 < avg_sars_cov2_conc)%>%
  ggplot(aes(x = avg_sars_cov2_conc, y = case_rate - .676*log(pop)))+
  geom_point(aes(color = main_variant), size = 5, alpha = .05)+
  geom_smooth())#%>%
```




```
  #plotly::ggplotly()
```

```
rm_data <- df_LOD%>%
  mutate(n_case_rate = case_rate  - .676*log(pop))

simple_lm_model <- lm(n_case_rate ~ avg_sars_cov2_conc + log(pop), data  = rm_data)
library(randomForest)
rm_data <- rm_data%>%
  modelr::add_predictions(simple_lm_model)%>%
  mutate(resid = n_case_rate - pred,
         PMMoV = log(PMMoV + 2),
         tss = as.numeric(tss),
```

```
        pcr_type = as.factor(pcr_type),
        regions = as.factor(regions))%>%
  select(- n_case_rate)%>%
  select(-case_rate, -pred, -avg_sars_cov2_conc)%>%
  select(wwtp_name, resid, regions, PMMoV, flow, ph, pcr_type, main_variant, pop_group)%>%
  group_by(wwtp_name, pcr_type)%>%
   mutate(across(c(where(is.numeric), -resid), scale))%>%
  ungroup()%>%

  select(-wwtp_name)%>%
  na.roughfix()

#n1_sars_cov2_lod, n2_sars_cov2_lod, LOD
  #pcr_type
resid_mod <- randomForest(resid ~ ., data = rm_data, ntree=500,
                    importance=TRUE, keep.inbag=TRUE)


summary(simple_lm_model)
```

```
##
## Call:
## lm(formula = n_case_rate ~ avg_sars_cov2_conc + log(pop), data = rm_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9662 -0.4952 -0.0512  0.4306  4.3759
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -7.3358311  0.1162581 -63.100   <2e-16 ***
## avg_sars_cov2_conc  0.2237410  0.0078151  28.629   <2e-16 ***
## log(pop)            0.0000362  0.0058764   0.006    0.995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7254 on 6326 degrees of freedom
## Multiple R-squared:  0.1148, Adjusted R-squared:  0.1146
## F-statistic: 410.4 on 2 and 6326 DF,  p-value: < 2.2e-16
```

```
resid_mod
```

```
##
## Call:
##  randomForest(formula = resid ~ ., data = rm_data, ntree = 500,      importance = TRUE, keep.inbag =
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 0.2288545
##                     % Var explained: 56.49
```

```
as.data.frame(randomForest::importance(resid_mod))%>%
  arrange(IncNodePurity)
```

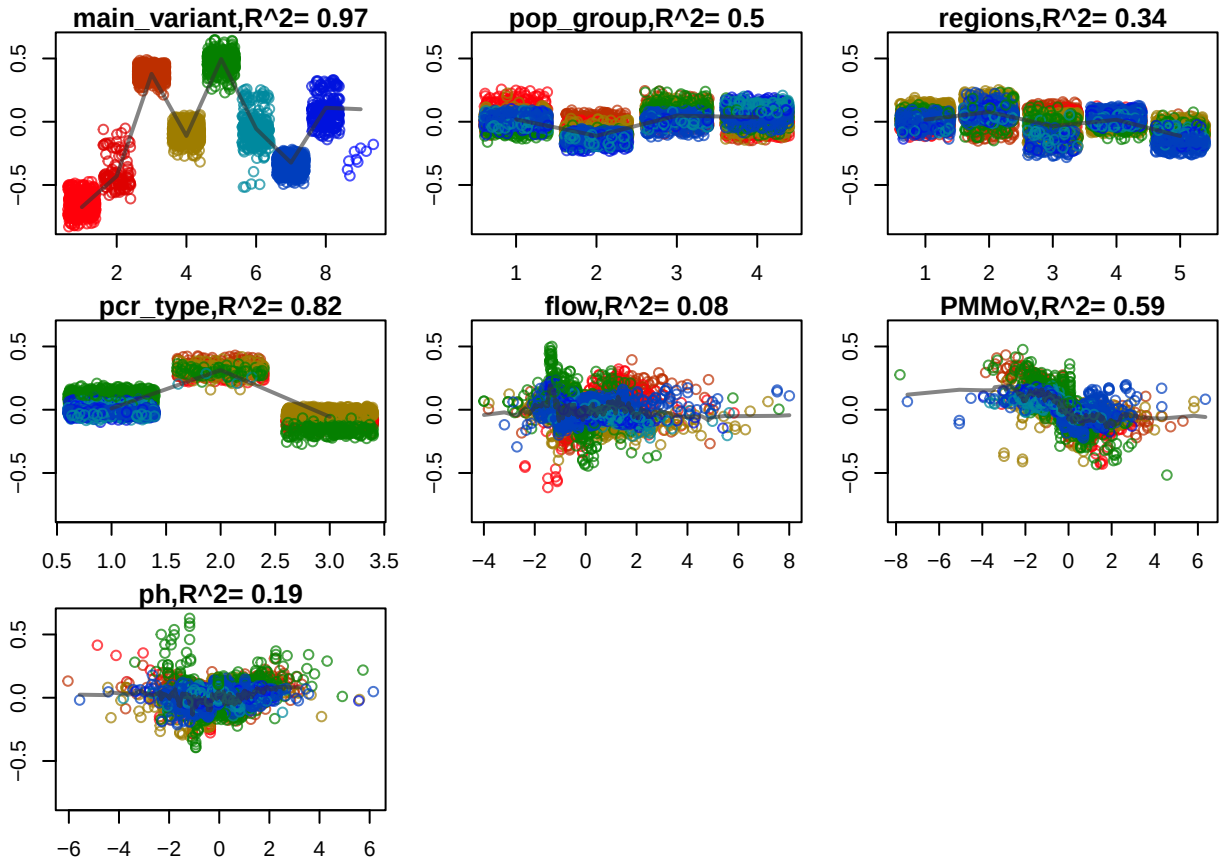
```
##                %IncMSE IncNodePurity
## pop_group     88.75115      169.8696
## regions       81.42375      179.0441
## pcr_type      81.03039      182.7714
## ph            60.72503      350.1511
## PMMoV         63.81313      413.2007
## flow          76.30676      420.8160
## main_variant 166.80296      987.3180
```

```
as.data.frame(randomForest::importance(resid_mod))%>%
   arrange(`%IncMSE`)
```

```
##                %IncMSE IncNodePurity
## ph            60.72503      350.1511
## PMMoV         63.81313      413.2007
## flow          76.30676      420.8160
## pcr_type      81.03039      182.7714
## regions       81.42375      179.0441
## pop_group     88.75115      169.8696
## main_variant 166.80296      987.3180
```

```
ff = forestFloor::forestFloor(resid_mod, na.roughfix(rm_data), calc_np=T)
Col = forestFloor::fcol(ff, cols = 1, outlier.lim = 2.5)
plot(ff, col=Col, plot_GOF = T)
```

```
## [1] "compute goodness-of-fit with leave-one-out k-nearest neighbor(guassian weighting), kknn package
```

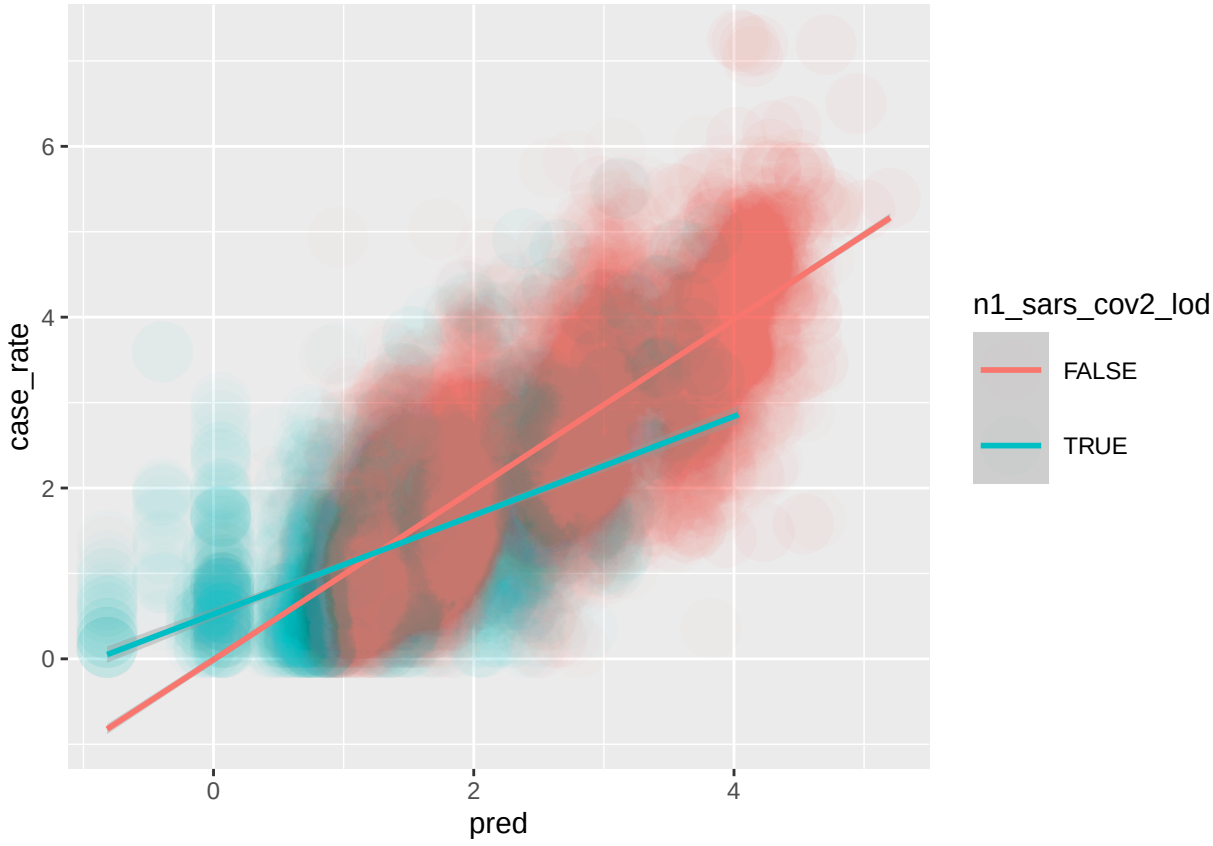



```
#hf183, conductivity, bcov_rec_rate, temperature
```

```
best_mod <- lm(case_rate ~ avg_sars_cov2_conc:pop_group + pop_group, data = df_LOD)


better_mod <- Graph_DF%>%
  modelr::add_predictions(best_mod)

better_mod%>%
  ggplot(aes(x = pred, y = case_rate, color = n1_sars_cov2_lod))+
  geom_point(size = 10, alpha = .02)+
  geom_smooth(method = "lm")
```



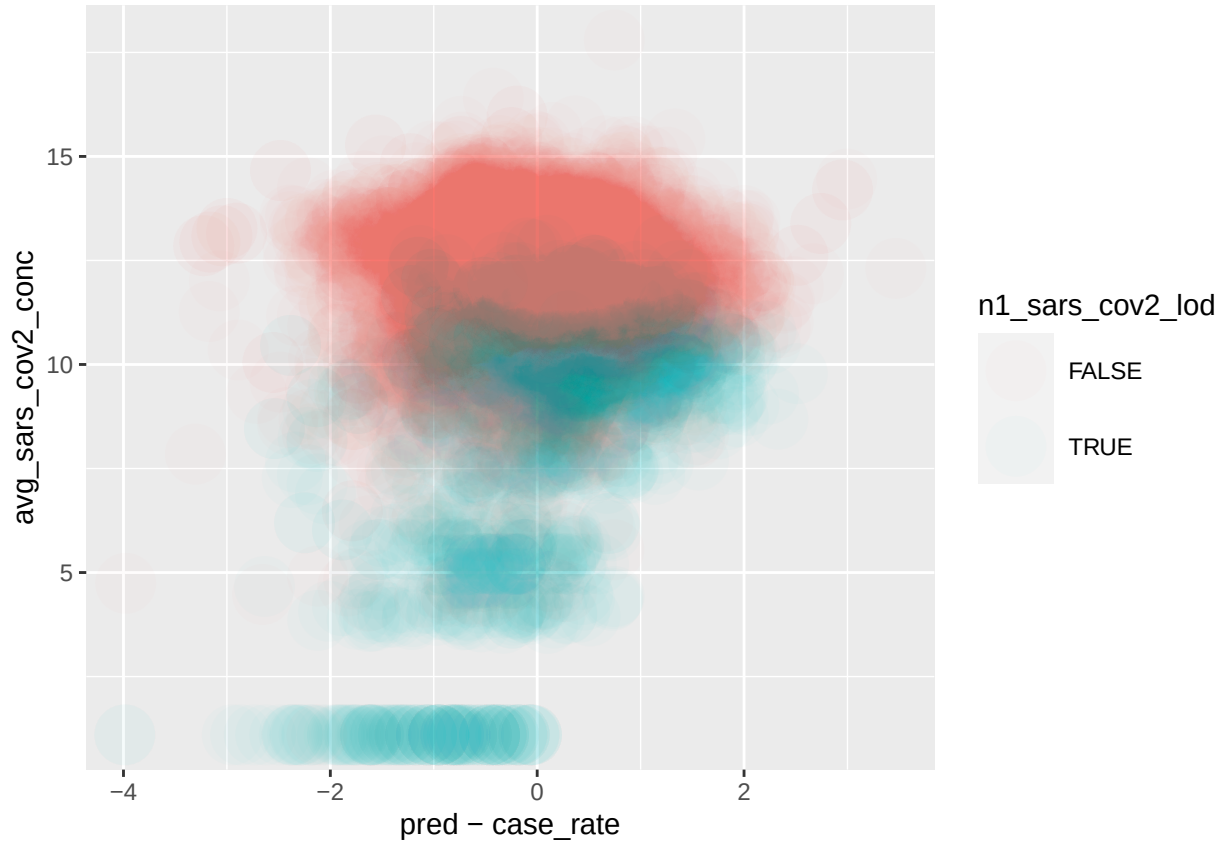

```
#df_LOD%>%

best_mod <- lm(case_rate ~ avg_sars_cov2_conc:pop_group + pop_group, data = df_LOD)
summary(best_mod)
```

```
##
## Call:
## lm(formula = case_rate ~ avg_sars_cov2_conc:pop_group + pop_group,
##     data = df_LOD)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4671 -0.5307 -0.0177  0.4799  3.2242
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -1.01763    0.24530  -4.148 3.39e-05 ***
## pop_group2                      0.92297    0.31456   2.934  0.00336 **
## pop_group3                      0.79942    0.32483   2.461  0.01388 *
## pop_group4                      0.21299    0.30760   0.692  0.48869
## avg_sars_cov2_conc:pop_group1   0.18161    0.01949   9.316  < 2e-16 ***
## avg_sars_cov2_conc:pop_group2   0.15769    0.01605   9.826  < 2e-16 ***
## avg_sars_cov2_conc:pop_group3   0.24816    0.01740  14.265  < 2e-16 ***
## avg_sars_cov2_conc:pop_group4   0.37599    0.01504  24.993  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7728 on 6321 degrees of freedom
## Multiple R-squared:  0.6442, Adjusted R-squared:  0.6438
## F-statistic:  1635 on 7 and 6321 DF,  p-value: < 2.2e-16
```

```
better_mod%>%
  ggplot(aes(x = pred - case_rate, y = avg_sars_cov2_conc, color = n1_sars_cov2_lod))+
  geom_point(size = 10, alpha = .02)
```




```
  #geom_smooth()
```

```
lm_plotly_model(Graph_DF)#plot based N1 vs cases



#M(t) = unknown function of T. Number of sick people at time t
#C(t) = e_l + a * M(t)*e_m + e_l
#W(t) = e_li + b * M(t)*e_e



plotly::ggplotly(
  Graph_DF%>%
    ggplot(aes(x = (avg_sars_cov2_conc),
```

```
              y = (case_rate),
              fill = pop_group,
              color = LOD))+
    geom_point()+
    #geom_smooth(method = "lm")+
    ggtitle("Strong Linear Relationship Between Log Cases and Log COVID Concentration")+
    xlab("log(Covid-19 concentration)")+
    ylab("log(case rate)")
)

plotly::ggplotly(
  Graph_DF%>%
    ggplot(aes(x = (avg_sars_cov2_conc),
              y = (case_rate),
              fill = main_variant,
              color = LOD))+
    geom_point()+
    #geom_smooth(method = "lm")+
    ggtitle("Strong Linear Relationship Between Log Cases and Log COVID Concentration")+
    xlab("log(Covid-19 concentration)")+
    ylab("log(case rate)")
)

gen_R2_params <- function(lm_formula, df){
  lm_run <- lm(lm_formula, data = df)
  return(c(summary(lm_run)$adj.r.squared, length(coef(lm_run))))
}

#main_variant
full_formula <- case_rate ~ avg_sars_cov2_conc:LOD:pop_group:main_variant + LOD:pop_group:main_variant
#
lm_full_inder_formula <- case_rate ~ avg_sars_cov2_conc:LOD:pop_group +  avg_sars_cov2_conc:LOD:main_va
#
lm_true_formula <- case_rate ~ avg_sars_cov2_conc:pop_group:main_variant + pop_group:main_variant
#
lm_true_inder_formula <- case_rate ~ avg_sars_cov2_conc:pop_group + avg_sars_cov2_conc:main_variant + p
##
lm_norm_formula <- case_rate ~ avg_sars_cov2_conc:LOD:main_variant + LOD:main_variant
##
lm_min_formula <- case_rate ~ avg_sars_cov2_conc:main_variant + main_variant
#
simple_formula <- case_rate ~ avg_sars_cov2_conc:LOD
{
  output_DF <- data.frame("Type" = c("adjusted R^2", "num_param"),
              "full model" = gen_R2_params(full_formula, Graph_DF),
              "full indirect model" = gen_R2_params(lm_full_inder_formula, Graph_DF),
              "true model" = gen_R2_params(lm_true_formula, df_LOD),
            "true indirect model"= gen_R2_params(lm_true_inder_formula, df_LOD),
              "norm site model" = gen_R2_params(lm_norm_formula, Graph_DF2),
              "true norm site model" = gen_R2_params(lm_min_formula, df_LOD2),
              "Original relationship" = gen_R2_params(simple_formula, Graph_DF))

  n <- output_DF$Type
```

```
  output_DF <- as.data.frame(t(output_DF[,-1]))
  colnames(output_DF) <- n
  output_DF
}
output_DF%>%
  ggplot(aes(x = num_param, y = `adjusted R^2`))+
  geom_point()+
  geom_smooth(se = FALSE)
```

////////////////////////////////////// // HFG Work //////////////////////////////////////

```
data("HFGWaste_data", package = "DSIWastewater")
data(Case_data , package = "DSIWastewater")

#crazy agressive method
hfg_outlier_detection <- function(small_vec){
  sortedVec <- sort(log(small_vec))
  lower_quant <- sortedVec[4]
  upper_quant <- sortedVec[6]
  range <- upper_quant - lower_quant
  retVec = ifelse(log(small_vec) > upper_quant + 1.5 * range,
                  exp(upper_quant + 1.5 * range), small_vec)
  retVec = ifelse(log(small_vec) < lower_quant - 1.5 * range,
                  exp(lower_quant - 1.5 * range), small_vec)
  retVec = ifelse(is.infinite(retVec), NA, retVec)
  return(retVec)
}

Pop_DF <- data.frame(
  site = c("Hudson","Kenosha","Platteville","Madison","Merrill","Plymouth","River Falls","Sun Prairie",
  pop = c(19680,122000,14000,380000,10000,9000,16000,34926,42000,67000,42000)
)

hfg_waste_filt_df <- HFGWaste_data%>%
  select(site, date, Filter, Well, N1, N2, PMMOV, HF183, CrP, everything())%>%
  group_by(date, site)%>%
  mutate(across(N1:CrP, hfg_outlier_detection))%>%
  left_join(Pop_DF)
```

```
trend_df <- hfg_waste_filt_df%>%
  group_by(site)%>%
  group_split()%>%
  lapply(loessSmoothMod, InVar = "N1", OutVar = "Trend_N1")%>%
  lapply(loessSmoothMod, InVar = "N2", OutVar = "Trend_N2")%>%
  lapply(loessSmoothMod, InVar = "PMMOV", OutVar = "Trend_PMMOV")%>%
  lapply(loessSmoothMod, InVar = "HF183", OutVar = "Trend_HF183")%>%
  lapply(loessSmoothMod, InVar = "CrP", OutVar = "Trend_CrP")%>%
  bind_rows()%>%
  mutate(
    Diff_N1 = Trend_N1 - N1,
    Diff_N2 = Trend_N2 - N2,
    Diff_PMMOV = Trend_PMMOV - PMMOV,
    Diff_HF183 = Trend_HF183 - HF183,
```

```
    Diff_CrP = Trend_CrP - CrP
    )%>%
  select(date, site, Filter, Well, pop, Trend_N1:Diff_CrP, N1LOD, N2LOD)
```

```
log_trend_df <- hfg_waste_filt_df%>%
  mutate(log_N1 = log(N1),
         log_N2 = log(N2),
         log_PMMOV = log(PMMOV),
         log_HF183 = log(HF183),
         log_CrP = log(CrP))%>%
  group_by(site)%>%
  group_split()%>%
  lapply(loessSmoothMod, InVar = "log_N1", OutVar = "Trend_N1")%>%
  lapply(loessSmoothMod, InVar = "log_N2", OutVar = "Trend_N2")%>%
  lapply(loessSmoothMod, InVar = "log_PMMOV", OutVar = "Trend_PMMOV")%>%
  lapply(loessSmoothMod, InVar = "log_HF183", OutVar = "Trend_HF183")%>%
  lapply(loessSmoothMod, InVar = "log_CrP", OutVar = "Trend_CrP")%>%
  bind_rows()%>%
  mutate(
    Diff_N1 = Trend_N1 - log_N1,
    Diff_N2 = Trend_N2 - log_N2,
    Diff_PMMOV = Trend_PMMOV - log_PMMOV,
    Diff_HF183 = Trend_HF183 - log_HF183,
    Diff_CrP = Trend_CrP - log_CrP)%>%
  select(site, date, Filter, Well, pop, Trend_N1:Diff_CrP, N1LOD, N2LOD)


diff_norm <- function(df){
  df%>%
    mutate(across(Diff_N1:Diff_CrP, ~.x - mean(.x, na.rm = TRUE)),
                        across(Diff_N1:Diff_CrP, ~ifelse(is.finite(.x),.x,NA)))
}

diff_var <- function(df, name){
  df%>%
    ungroup()%>%
    summarise(across(Diff_N1:Diff_CrP, ~var(.x, na.rm = TRUE)))%>%
    mutate(var_type = name)
}
```

```
gen_vars <- function(df){
  trend_variance_df <- df%>%
    group_by(site)%>%
    diff_norm()

  filter_variance_df <- df%>%
    group_by(site, date)%>%
    diff_norm()

  well_variance_df <- df%>%
    group_by(site, date, Filter)%>%
    diff_norm()
```

```
  bind_DF <- rbind(diff_var(trend_variance_df, "trend var"),
                   diff_var(filter_variance_df, "filter var"),
                   diff_var(well_variance_df, "well var"))%>%
    pivot_longer(Diff_N1:Diff_CrP)%>%
    mutate(name = factor(name, levels=c('Diff_N1', 'Diff_N2', 'Diff_PMMOV',
                                        'Diff_HF183', 'Diff_CrP')),
           var_type = factor(var_type, levels = c("trend var", "filter var",
                                                  "well var")))
  levels(bind_DF$name) <- c("N1", "N2", "PMMoV", "HF183", "CrP")
  return(bind_DF)
}

gen_plot_heat <- function(df, title = NA){
  df%>%
    ggplot(aes(x = name, y = var_type)) +
    geom_tile(aes(fill = value)) +
    geom_text(aes(label = round(value, 3))) +#formatC
    scale_fill_gradient(low = "white", high = "red")+
    scale_x_discrete(position = "top")+
    xlab("signal source")+
    ylab("variance source")+
    ggtitle(title)
}

gen_plot_hist <- function(df, title = NA){
  t_plot <- df%>%
    ggplot(aes(x = name, fill = var_type))+
    geom_col(aes(y = value), position="identity" ) +
    xlab("signal source")+
    ylab("Cumaltive Variance")+
    ggtitle(title)
  return(t_plot)
}
```

```
dis_df <- log_trend_df

var_output_base <- gen_vars(log_trend_df)

var_output_lod <- log_trend_df%>%
  filter(N2LOD | N1LOD)%>%
  gen_vars()


var_output_NLOD <- log_trend_df%>%
  filter(!(N2LOD | N1LOD))%>%
  gen_vars()


gen_plot_heat(var_output_base, "source of variance in HFG data")
gen_plot_heat(var_output_lod, "source of variance in above LOD info HFG data")
gen_plot_heat(var_output_NLOD, "source of variance in bellow LOD info HFG data")

gen_plot_hist(var_output_base, "source of variance in HFG data")
```

```r
gen_plot_hist(var_output_lod, "source of variance in above LOD info HFG data")
gen_plot_hist(var_output_NLOD, "source of variance in bellow LOD info HFG data")


#Trend var: variance points of a day #variance of the signal
#Filter var: variance between collected points #collection variance
#well var: variance of 3 points #PCR test variance / extraction variance

#Gaussian mixture model
#change label to make more clear
#stacked in bar graph
#z = x + y
#var(z) = var(x) + var(y) if independent(
```

```r
temp_func <- function(DF){
  g_ret <- DF%>%
    ggplot(aes(x = Diff_N1))+
    geom_histogram()
  return(g_ret)
}
temp_func(trend_variance_df)
temp_func(filter_variance_df)
temp_func(well_variance_df)

hist_plot <- ggplot(mapping = aes(x = Diff_N1))+
    geom_histogram(data = well_variance_df, fill = "green")+
    geom_histogram(data = filter_variance_df, fill = "blue")+
    geom_histogram(data = trend_variance_df, fill = "red")

plotly::ggplotly(hist_plot)
```

```r
num_samples <- 16000
for(A in c(1,3,5)){
  X <- 2 * rnorm(num_samples) + 10
  yn_exp <- rnorm(num_samples)
  yn_lin <- exp(2*A)*rnorm(num_samples)
  Y <- exp(X + yn_exp) + yn_lin
  xn_exp <- rnorm(num_samples)
  xn_lin <- exp(1*A)*rnorm(num_samples)
  X <- exp(X + xn_exp)# + xn_lin
  #plot(X, Y)
  plot(log(X), log(Y))
}
```