High level normalization work

Marlin

2022-12-02

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
library(DSIWastewater)
data(Case_data, package = "DSIWastewater")
data(pop_data, package = "DSIWastewater")
#restrict Case data to only Madison data
Case_DF <- Case_data


#restrict Case data to dates after 2021-02-01
Case_DF <- Case_DF[Case_DF$date  >= as.Date("2020-9-10"),]
Case_DF <- Case_DF%>%
  left_join(pop_data)%>%
  rename(population_served = pop)
```

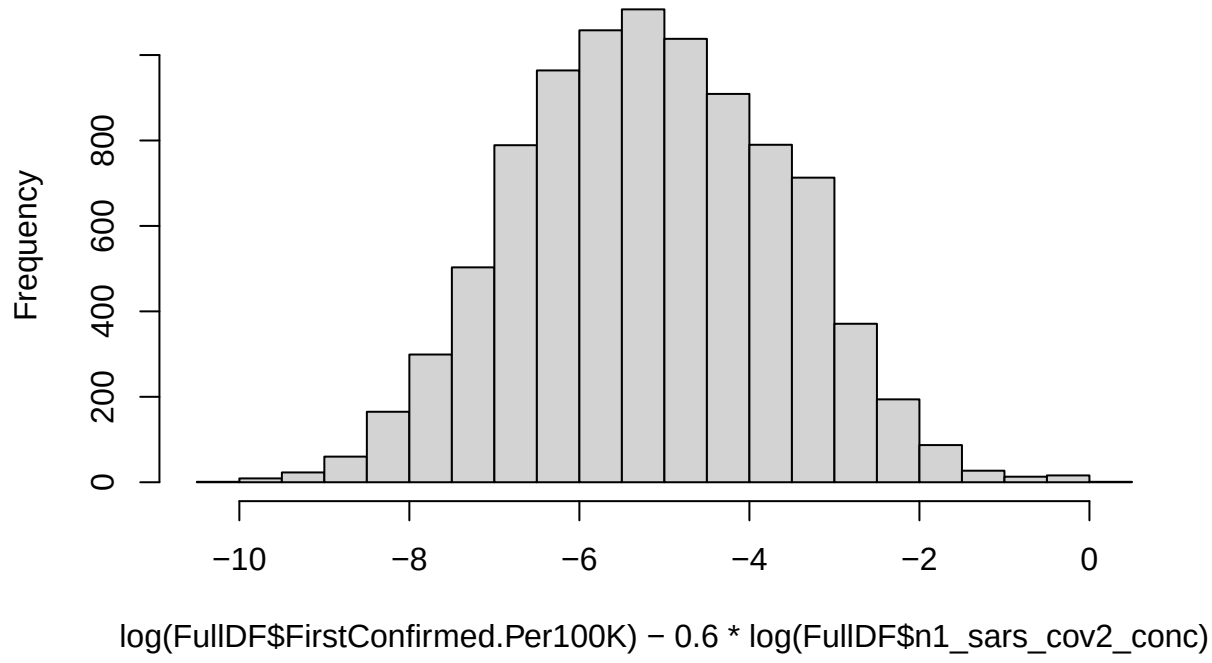
```
## Joining with `by = join_by(site)`
```

```
## Warning in left_join(., pop_data): Detected an unexpected many-to-many relationship between `x` and
## i Row 20501 of `x` matches multiple rows in `y`.
## i Row 70 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```r
#get the case flags
Case_DF <- buildCaseAnalysisDF(Case_DF)

Case_DF <- Case_DF#[,c(1:2, 5, 7)]

data(WasteWater_data, package = "DSIWastewater")

#restrict Waste data to only Madison data
baseWaste_DF <- WasteWater_data

Temp <- baseWaste_DF%>%
  mutate(n1_sars_cov2_lod = gsub(" ", "", tolower(n1_sars_cov2_lod)) ==
         "yes",
         n2_sars_cov2_lod = gsub(" ", "", tolower(n2_sars_cov2_lod)) ==
         "yes",
         n1_sars_cov2_conc = ifelse(n1_sars_cov2_lod, as.numeric(n1_lod)/2,
         N1),
         n2_sars_cov2_conc = ifelse(n2_sars_cov2_lod,
         as.numeric(n2_lod)/2, N2))%>%
  rename()%>%
  select(site, date,
         pop, n1_sars_cov2_conc, n2_sars_cov2_conc,
         flow, PMMoV) %>%
  mutate(site = ifelse(site == "Madison MSD WWTF" , "Madison", site))

FullDF <- full_join(Case_DF, Temp, by = c("date","site"))%>%
  group_by(site)%>%
  mutate(pop = mean(pop, na.rm = TRUE),
          FirstConfirmed.Per100K = pastwk.avg.casesperday.Per100K)%>%
  ungroup()
```

```
## Warning in full_join(Case_DF, Temp, by = c("date", "site")): Detected an unexpected many-to-many rela
## i Row 1340 of `x` matches multiple rows in `y`.
## i Row 1467 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```r
hist(log(FullDF$FirstConfirmed.Per100K) - .6*log(FullDF$n1_sars_cov2_conc))
```



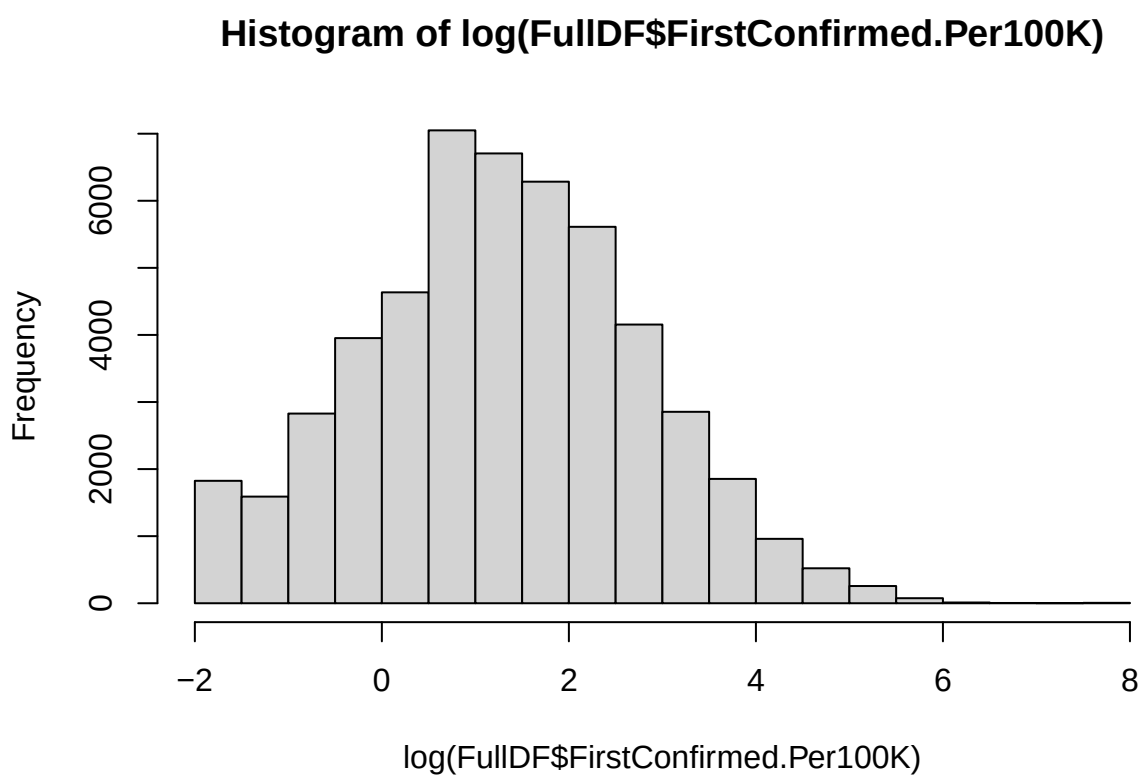

```
diff <- log(FullDF$FirstConfirmed.Per100K) - .6*log(FullDF$n1_sars_cov2_conc)

cor(diff, log(FullDF$n1_sars_cov2_conc), use  = "pairwise.complete.obs")
```

```
## [1] NaN
```

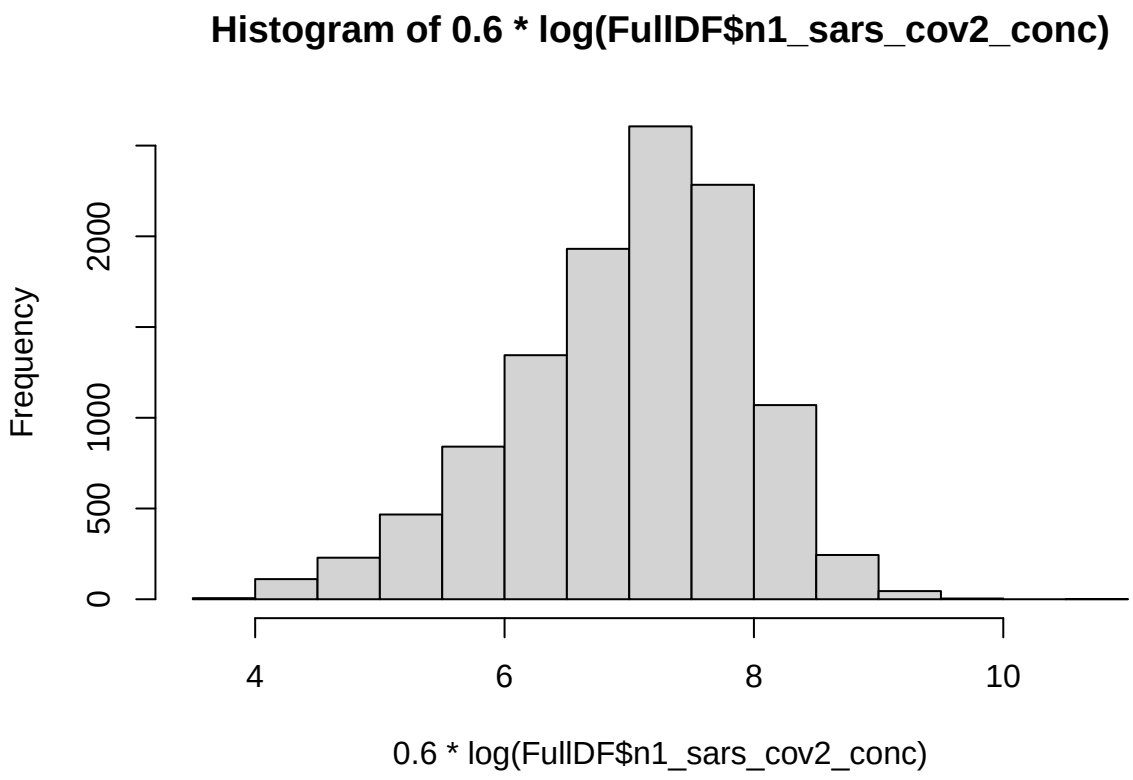
```
cor(diff, log(FullDF$FirstConfirmed.Per100K), use  = "pairwise.complete.obs")
```

```
## [1] NaN
```

```
hist(log(FullDF$FirstConfirmed.Per100K))
```



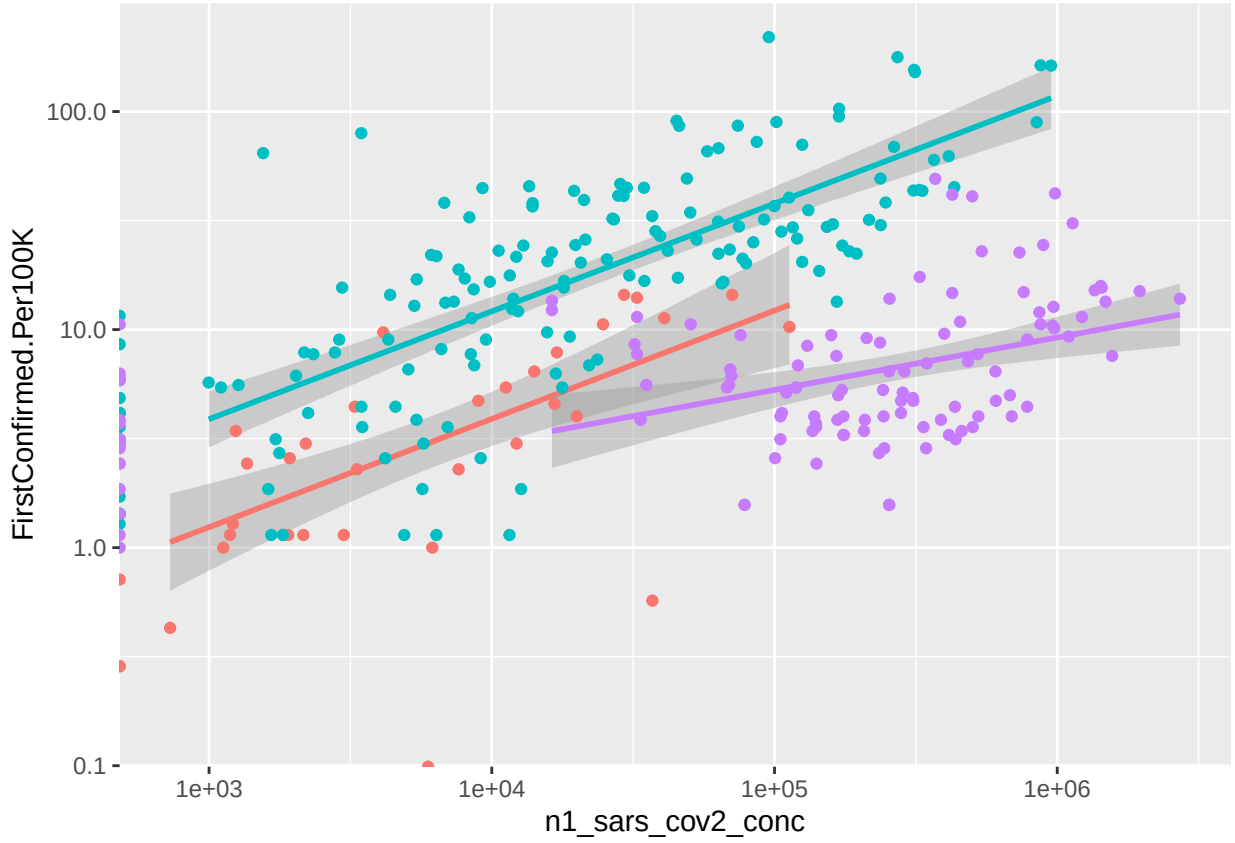

```
hist(.6*log(FullDF$n1_sars_cov2_conc))
```




```
toShow <- sample(unique(FullDF$site), 4)

quick_plot <- function(df, x){
  df%>%
    filter(site %in% toShow)%>%
    ggplot(aes(y = FirstConfirmed.Per100K,
               x = !!sym(x),
               color = site))+
    geom_smooth(method = "lm")+
    geom_point()+
    guides(color = FALSE)+
    scale_y_log10()+
    scale_x_log10()
}
FullDF%>%
  quick_plot("n1_sars_cov2_conc")
```

```
## Warning: The ‘<scale>‘ argument of ‘guides()‘ cannot be ‘FALSE‘. Use "none" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous x-axis

## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 2205 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 2170 rows containing missing values (`geom_point()`).
```

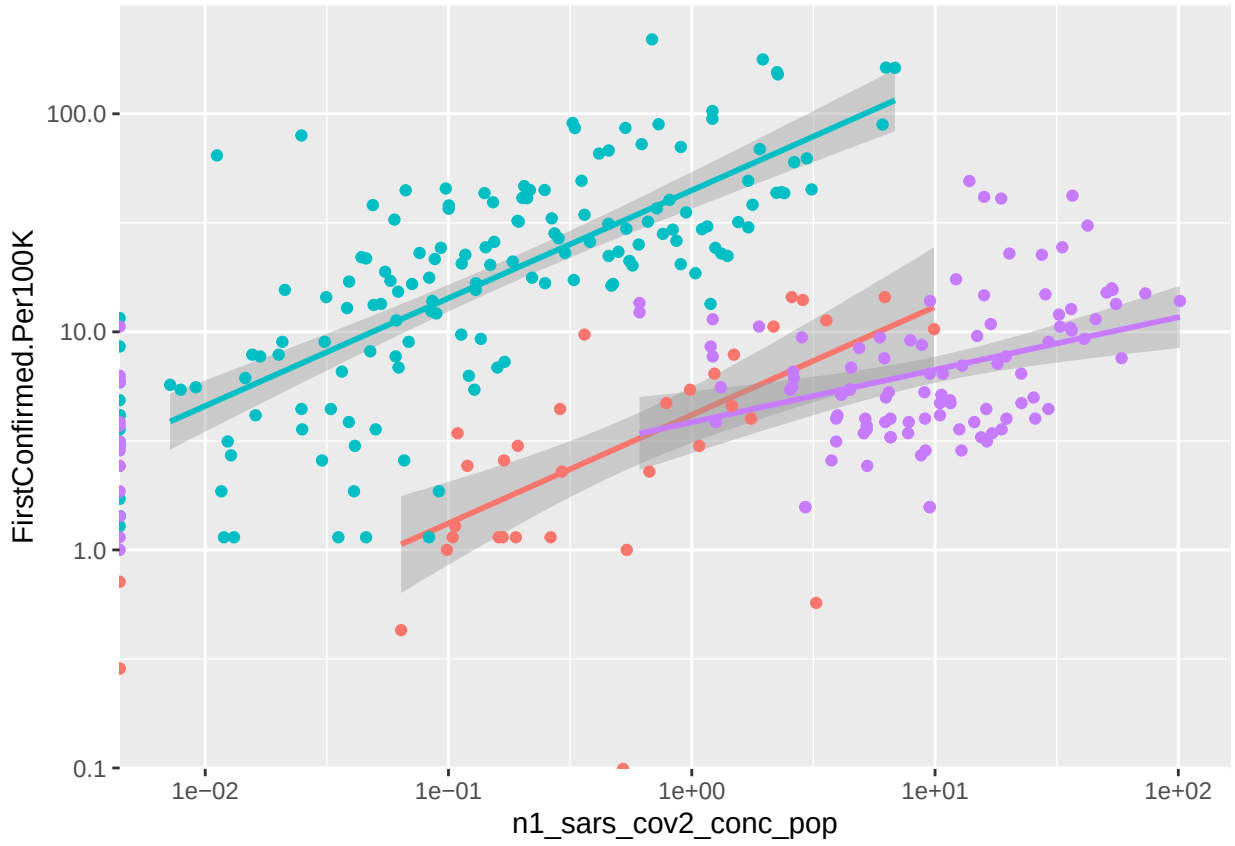



```
FullDF%>%
  mutate(n1_sars_cov2_conc_pop = n1_sars_cov2_conc / pop)%>%
  quick_plot("n1_sars_cov2_conc_pop")
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous x-axis

## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 2205 rows containing non-finite values (‘stat_smooth()‘).
```

```
## Warning: Removed 2170 rows containing missing values (‘geom_point()‘).
```




```
FullDF%>%
  mutate(n1_sars_cov2_conc_ppmov = n1_sars_cov2_conc / PMMoV)%>%
  quick_plot("n1_sars_cov2_conc_ppmov")
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

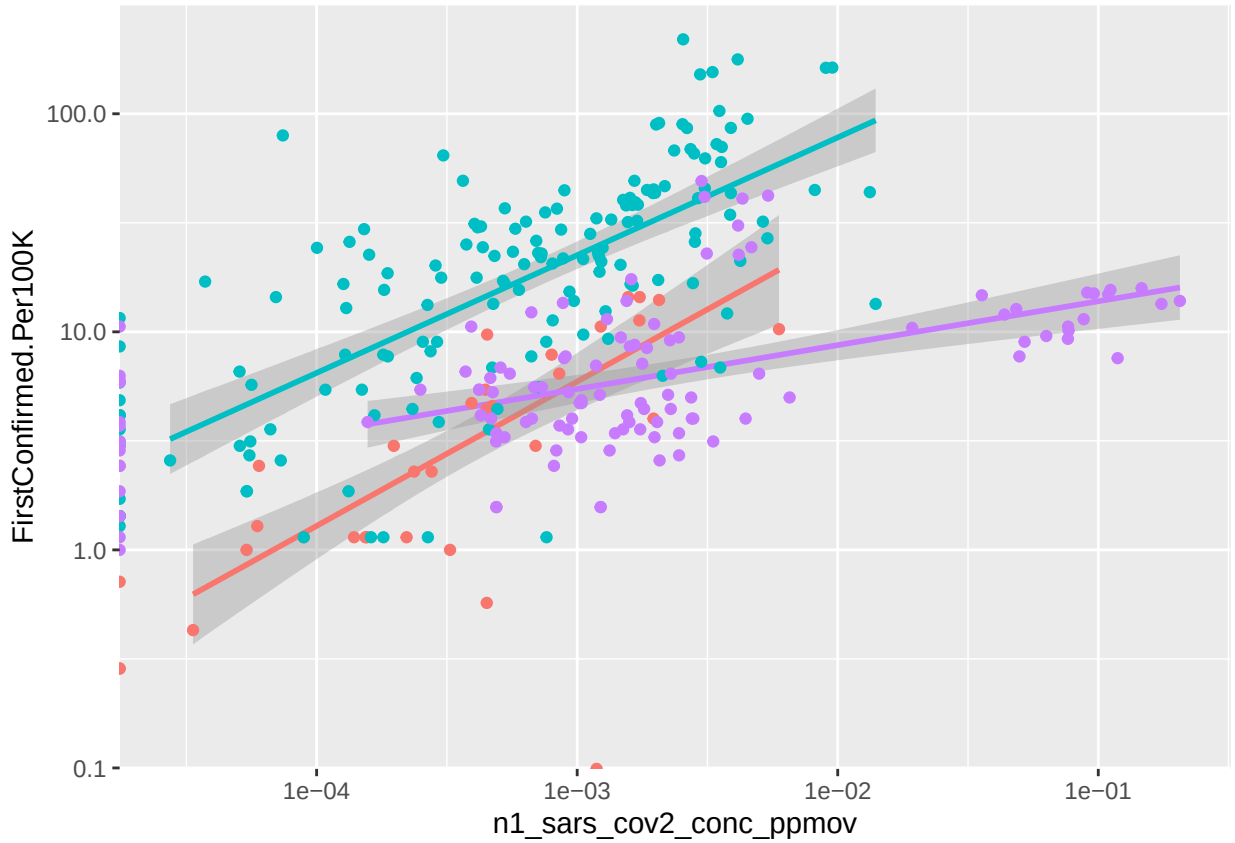
```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
## Warning: Removed 2207 rows containing non-finite values (‘stat_smooth()‘).
```

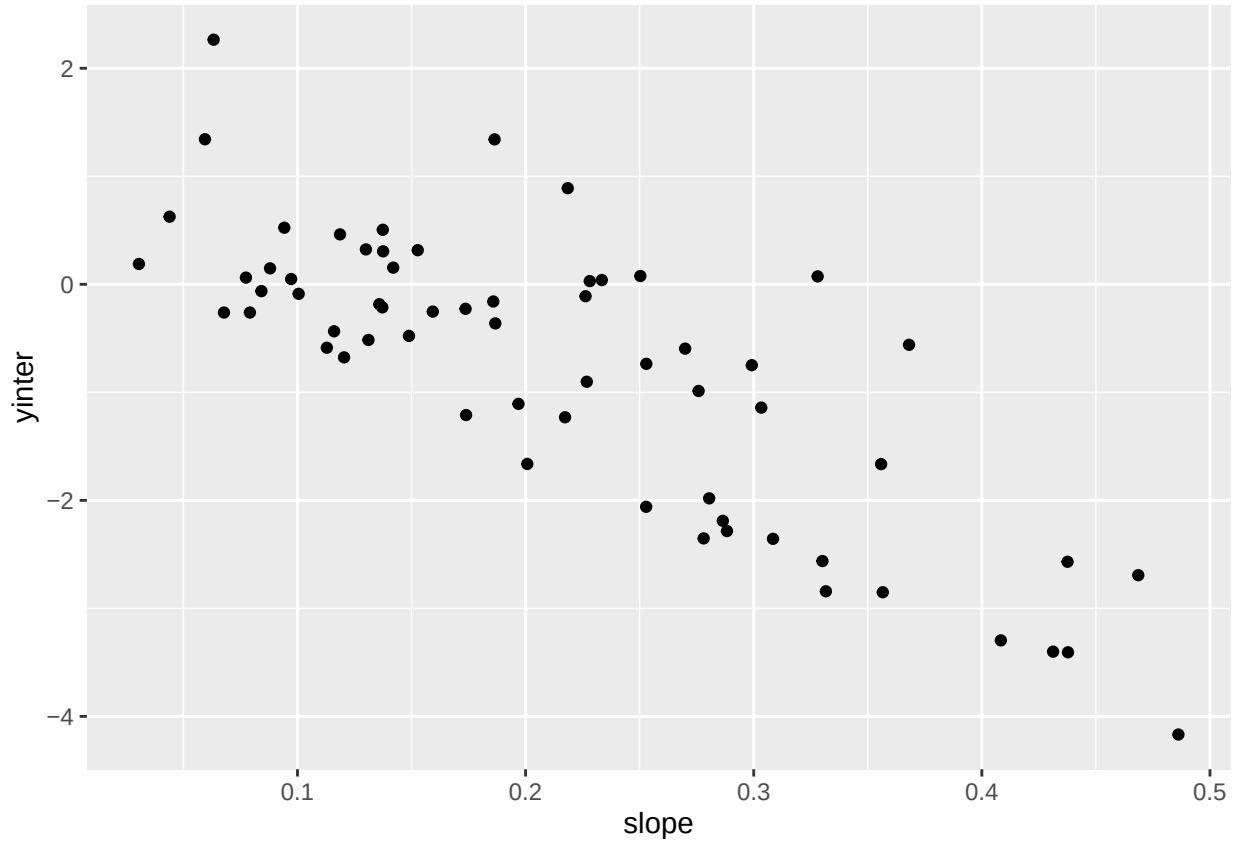
```
## Warning: Removed 2172 rows containing missing values (‘geom_point()‘).
```



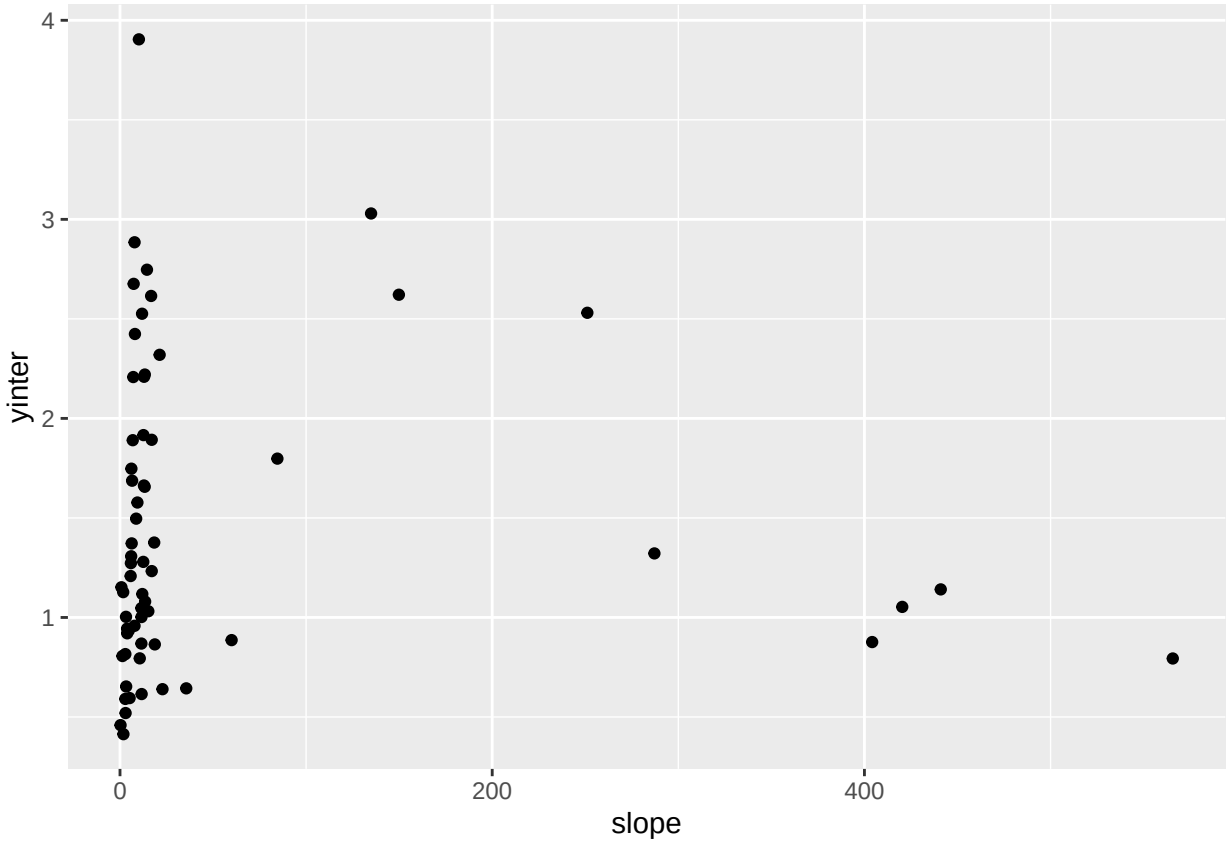

```
lm_DF1 <- FullDF%>%
  group_by(site)%>%
  filter(!is.na(log(FirstConfirmed.Per100K)), !is.na(log(n1_sars_cov2_conc)))%>%
  summarise(yinter = coef(lm(log(FirstConfirmed.Per100K + 1) ~ log(n1_sars_cov2_conc + 1)))[1],
            slope = coef(lm(log(FirstConfirmed.Per100K + 1) ~ log(n1_sars_cov2_conc + 1)))[2])



lm_DF2 <- FullDF%>%
  group_by(site)%>%
  filter(!is.na(log(1 + FirstConfirmed.Per100K)),
         !is.na(log(1 + n1_sars_cov2_conc/PMMoV)),
         !is.na(log(PMMoV)),
         PMMoV != 0)%>%
  summarise(yinter = coef(lm(log(1 + FirstConfirmed.Per100K) ~ log(1 + n1_sars_cov2_conc/abs(PMMoV))))[
            slope = coef(lm(log(1 + FirstConfirmed.Per100K) ~ log(1 + n1_sars_cov2_conc/abs(PMMoV))))[2
```

```
## Warning: There were 4 warnings in ‘filter()‘.
## The first warning was:
## i In argument: ‘!is.na(log(1 + n1_sars_cov2_conc/PMMoV))‘.
## i In group 59: ‘site = "Rib Lake"‘.
## Caused by warning in ‘log()‘:
## ! NaNs produced
## i Run ‘dplyr::last_dplyr_warnings()‘ to see the 3 remaining warnings.
```

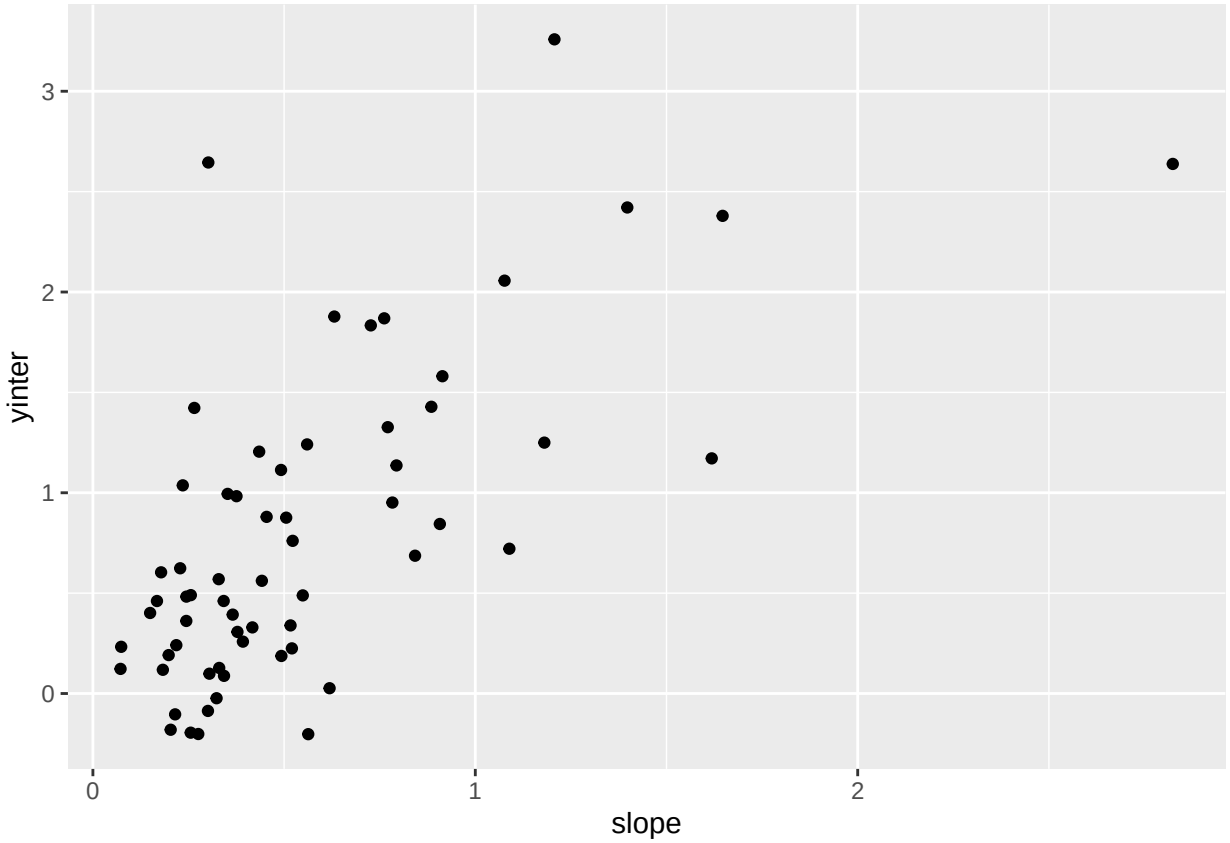
```r
lm_DF3 <- FullDF%>%
  group_by(site)%>%
  filter(!is.na(log(1 + FirstConfirmed.Per100K)),
         !is.na(log(1 + n1_sars_cov2_conc)),
         !is.na(log(1 + pop)))%>%
  summarise(yinter = coef(lm(log(1 + FirstConfirmed.Per100K) ~
                               log(1 + n1_sars_cov2_conc / pop)))[1],
            slope = coef(lm(log(1 + FirstConfirmed.Per100K) ~
                            log(1 + n1_sars_cov2_conc / pop)))[2])

lm_DF1%>%
  ggplot(aes(x = slope, y = yinter))+
  geom_point()
```




```r
lm_DF2%>%
  ggplot(aes(x = slope, y = yinter))+
  geom_point()
```




```
lm_DF3%>%
  ggplot(aes(x = slope, y = yinter))+
  geom_point()
```


yinter
slope
0
1
2
3
0
1
2