Homework #3

Gbemisola Talabi

1 November 2022

Instructions

1. Please submit your knitted `.pdf` file to the assignment drop box on eLC. If you are still having trouble knitting your file you can submit your `.Rmd` file.
2. The assignment is due November 1, 2022 by 7:00pm EST. This assignment will be graded for accuracy. Please reach out to us if you need help before this time!
3. Please add your name as "author" to the YAML header above.
4. Below each question is a `r` code chunk that can be used to explore the question. Use the space below the code chunk to directly answer the question.

```
## you can add more, or change...
rm(list=ls(all=T))

library(tidyverse)
library(RColorBrewer)
library(naniar)
```

Problem Set

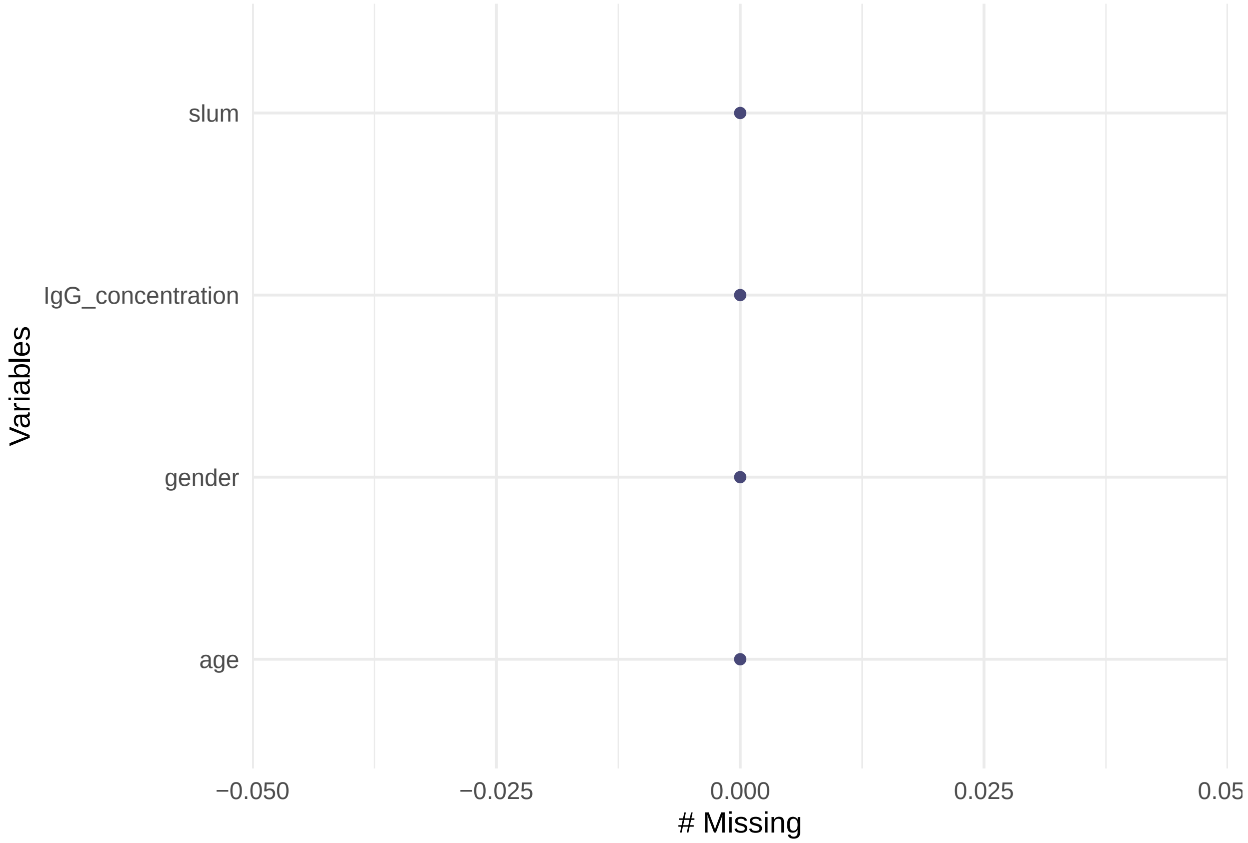
1. Import the dataset from the filename "serodata.rds" from eLC. Assign it to `sero` object and use `glimpse()` to check out the data. Each row is a different individual whom we serological data: `IgG_concentration` represents virus-specific IgG antibody concentrations from a fully immunizing infection based on an enzyme immunoassay; `age` is age in years; `sex` is sex of the individual; and `slum` characterizes the residence of the individual. How many rows, how many columns, what are the data types of the variables? Check the data for missingness using `gg_miss_var()` from the `naniar` package.

```
sero <- read_rds("../data/serodata.rds")
glimpse(sero)
```

```
## Rows: 651
## Columns: 4
## $ IgG_concentration <dbl> 0.31768953, 3.43682310, 0.30000000, 143.23630137, 0.~
## $ age               <dbl> 2, 4, 4, 4, 1, 4, 4, 2, 4, 2, 3, 15, 8, 12, 15, 9, 8~
## $ gender            <chr> "Female", "Female", "Male", "Male", "Male", "Male", ~
## $ slum              <chr> "Non slum", "Non slum", "Non slum", "Non slum", "Non~
```

```
naniar::gg_miss_var(sero)
```

```
## Warning: It is deprecated to specify ‘guide = FALSE‘ to remove a guide. Please
## use ‘guide = "none"‘ instead.
```



```
# 651 rows, 4 columns, IgG_concentration and age have numeric or double variables
#gender and slum have character variables
```

2. Create a new variable called `log_IgG_concentration` which is the log10 of `IgG_concentration` - hint: use `log10(IgG_concentration)` nested inside `mutate()` and make sure reassign it back to `sero`

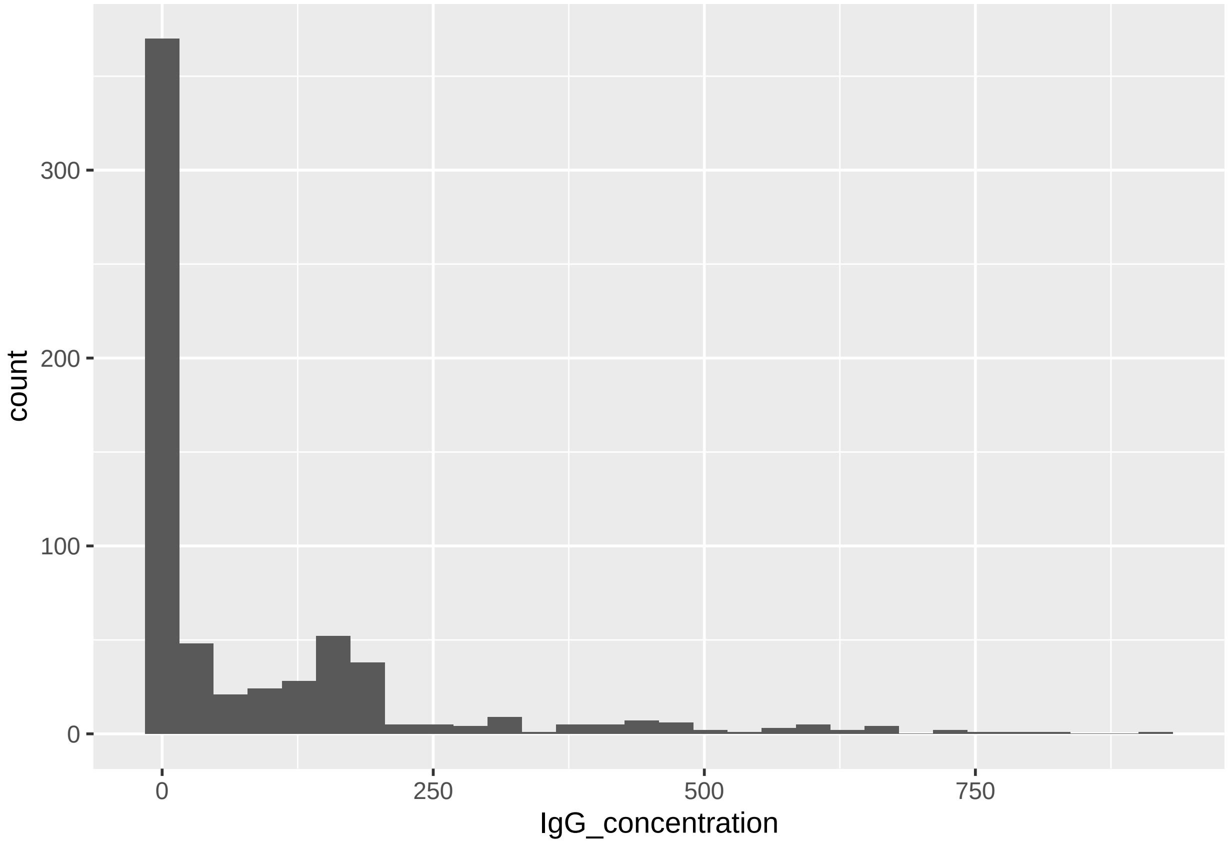
```
sero <- sero %>% mutate(log_IgG_concentration = log10(IgG_concentration))
head(sero)
```

```
##   IgG_concentration age gender     slum log_IgG_concentration
## 1        0.31768953   2 Female Non slum            -0.4979971
## 2        3.43682310   4 Female Non slum             0.5361572
## 3        0.30000000   4   Male Non slum            -0.5228787
## 4      143.23630137   4   Male Non slum             2.1560531
## 5        0.44765343   1   Male Non slum            -0.3490581
## 6        0.02527076   4   Male Non slum            -1.5973817
```

3. Use `ggplot2` to create a histogram of `IgG_concentration` using `stat_bin()`. Describe the distribution of the data based on the histogram. What do we learn about the distribution of `IgG_concentration`

```
ggplot(sero) +
  stat_bin(aes(x= IgG_concentration))
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```

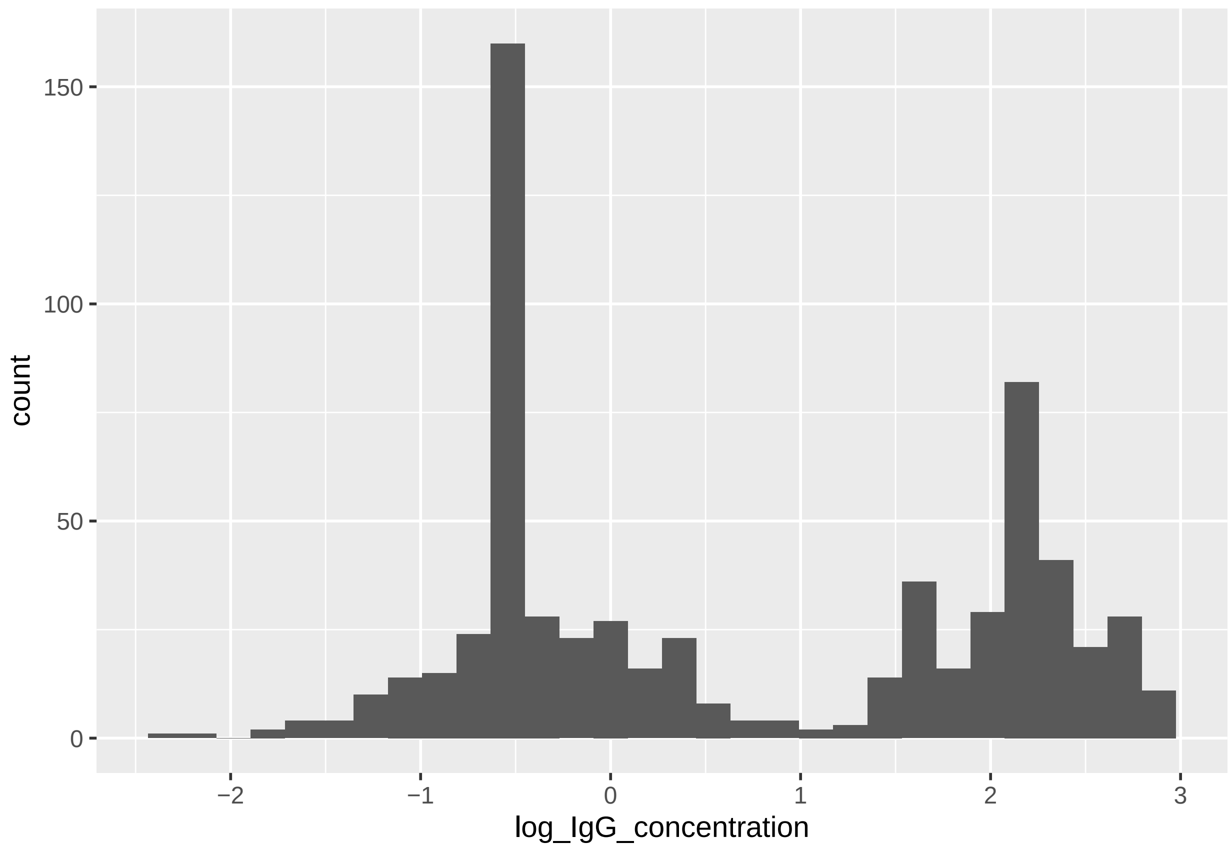


```
#The distribution is right or positively skewed and unimodal.
```

4. Create a histogram of `log_IgG_concentration` using `stat_bin()`. IgG antibodies above a certain threshold generally represent immunity in an individual. If you were to split this bimodal distribution into two normal distributions where would you draw the threshold?

```
ggplot(sero) +
  stat_bin(aes(x= log_IgG_concentration))
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



```
#the threshold will be drawn at 1 to split this bimodal distribution into two normal distributions.
```

5a. Create a new variable named `sero_status` as a binary variable in which observations below the threshold are defined as 0 and observations above or equal to the threshold are defined as 1. Make sure to reassign it back to `sero` - use the `log_IgG_concentration` threshold of 1.0

```
attach(sero)
sero$sero_status[log_IgG_concentration< 1] <- 0
sero$sero_status[log_IgG_concentration>=1] <- 1
```

5b. Output the min and max of `log_IgG_concentration` by `sero_status` to make sure the code you used in #5a worked correctly. - hint: use `group_by()` and `summarise()` creating a minimum and maximum variable using `min()` and `max()`

```
sero %>%
  group_by(sero_status) %>%
  summarise(min(log_IgG_concentration), max(log_IgG_concentration))
```

```
## # A tibble: 2 x 3
##   sero_status `min(log_IgG_concentration)` `max(log_IgG_concentration)`
##         <dbl>                        <dbl>                        <dbl>
## 1           0                        -2.27                        0.980
## 2           1                         1.09                        2.96
```

6a. Create a new factor variable named `slum_binary` as a binary variable in which observations with `slum` and `mixed` values (from the `slum` variables) are defined as `slum/mixed` and observations with the `Non slum` value remain defined as `Non slum`. Make sure to reassign it back to `sero`

```
sero <- sero %>%
  mutate(slum = factor(slum,
                       levels = c("Slum", "Mixed", "Non slum")))
levels(pull(sero, slum))
```

```
## [1] "Slum"     "Mixed"    "Non slum"
```

```
sero <- sero %>%
  mutate(slum_binary = (recode(slum,
                       "Slum" = "slum/mixed",
                       "Mixed" = "slum/mixed",
                       "Non slum" = "Non slum")))
```

6b. Output a table of `slum` and `slum_binary` to make sure the code you used in #6a worked correctly. - hint: use `count(slum, slum_binary)`

```
sero %>%
  mutate(slum_binary = (recode(slum,
                       "Slum" = "slum/mixed",
                       "Mixed" = "slum/mixed",
                       "Non slum" = "Non slum"))) %>%
  count(slum, slum_binary)
```

```
##       slum slum_binary   n
## 1     Slum  slum/mixed  45
## 2    Mixed  slum/mixed 135
## 3 Non slum    Non slum 471
```

7. Now let's summarize the `sero` dataset: Calculate the proportion of observations where `sero_status` is equal to 1 grouped by `age` and `slum_binary` variables. Call the new variable `prop` & assign the new tibble to `sero_group`. Check that the number of rows in the new datset `sero_group` is 30 using `nrow()` - hint: you will need to use `group_by()` and then `summarise(prop = ...)` to create a new variable `prop`

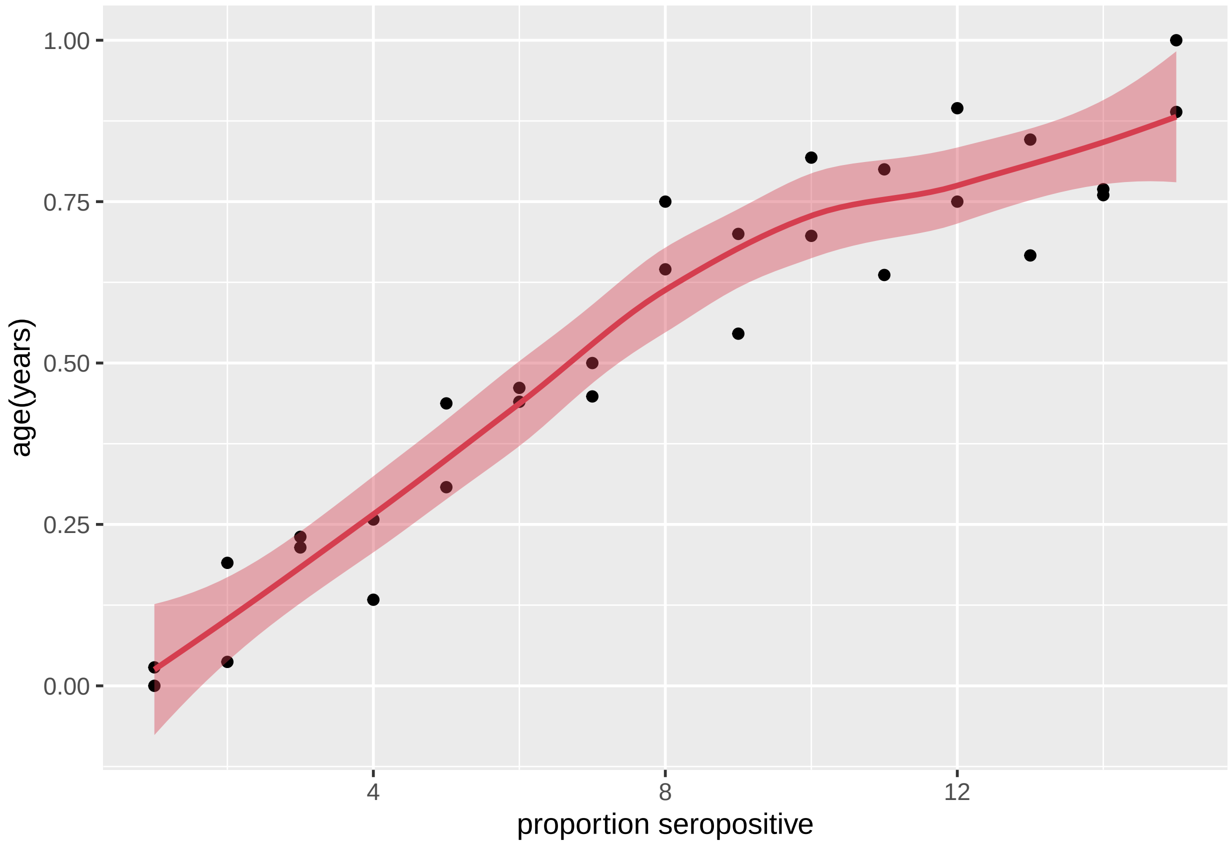
```
sero_group <- sero %>%
  group_by(age, slum_binary) %>%
  summarise(prop = sum(sero_status == 1)/n(),
            .groups ="drop")

nrow(sero_group)
```

```
## [1] 30
```

8. Using the `sero_group` dataset, create a scatter plot of `age` (x axis) and `prop` (y axis) and then add fit a line using LOESS method. - change the LOESS fit line color and fit line ribbon fill to the color HEX code "#d53e4f" - change the y axis label to "proportion seropositive" using `labs()` - change the x axis label to "age (years)" using `labs()`

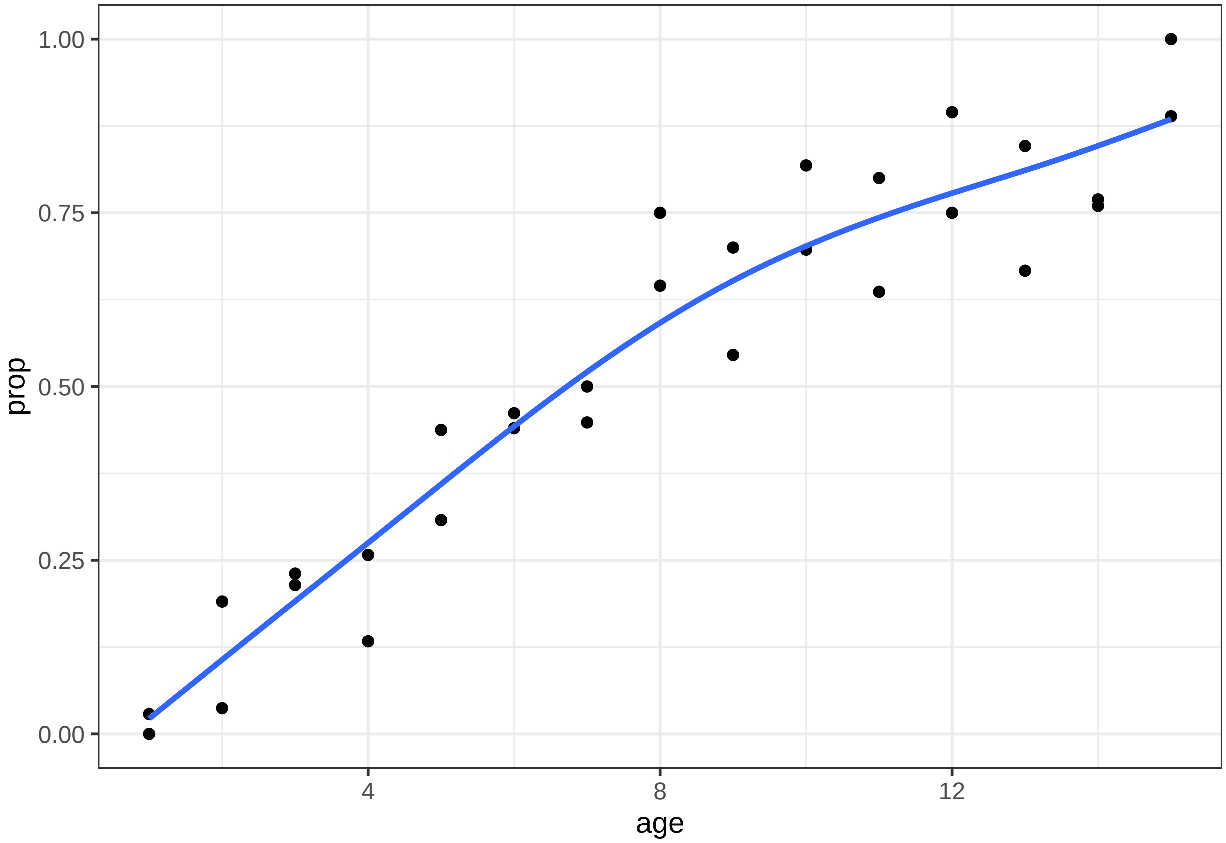
```
ggplot(sero_group, aes(x=age, y=prop)) +
  geom_point() +
  stat_smooth(method="loess", col="#d53e4f", fill="#d53e4f") +
  labs(x = "proportion seropositive",
       y = "age(years)")
```

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



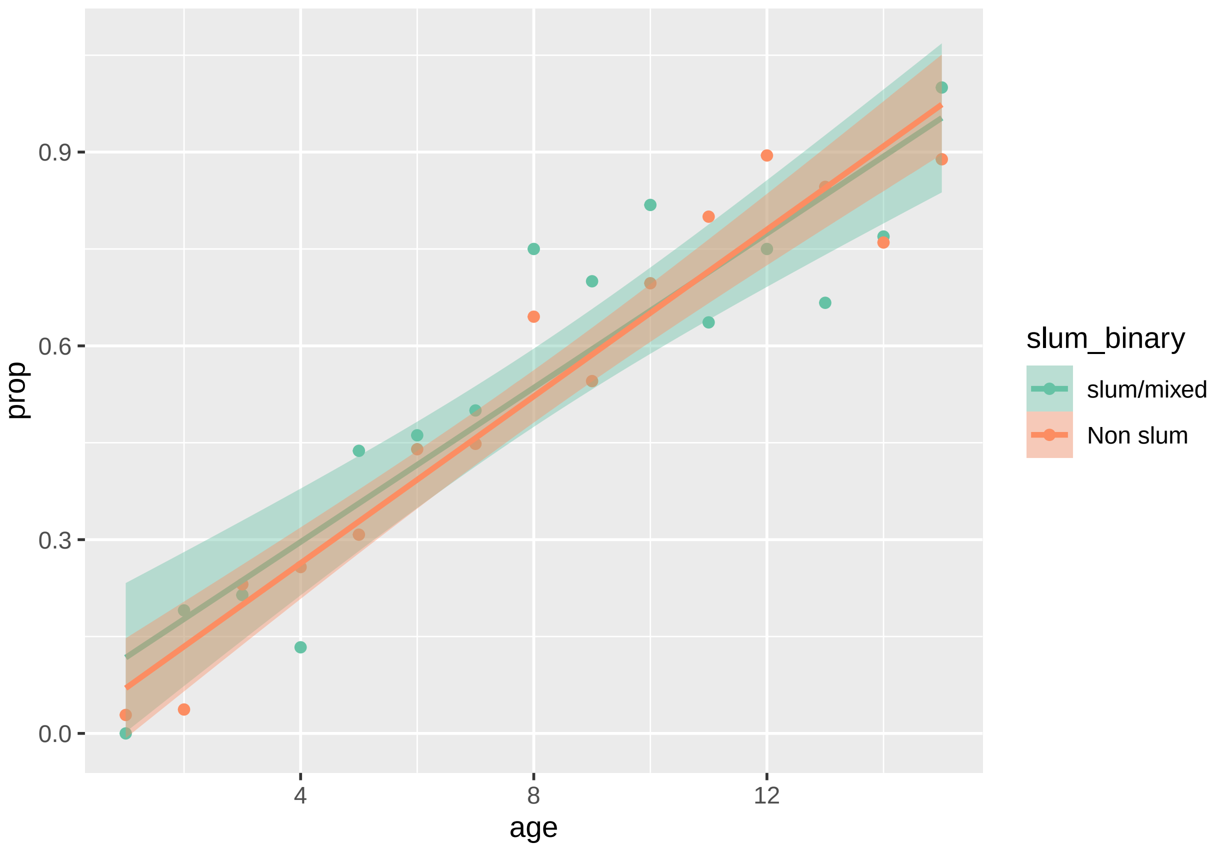
9. Using the `sero_group` dataset, create a scatter plot of `age` (x axis) and `prop` (y axis) and then add fit a line using GAM method. - use `formula = y ~ s(x, bs="cr")` - remove the 95% CI ribbon from the GAM fit - use built in ggplot theme `theme_bw()`

```
ggplot(sero_group, aes(x=age, y=prop)) +
  geom_point() +
  stat_smooth(method="gam", se =FALSE, formula = y ~ s(x, bs="cr")) +
  theme_bw()
```



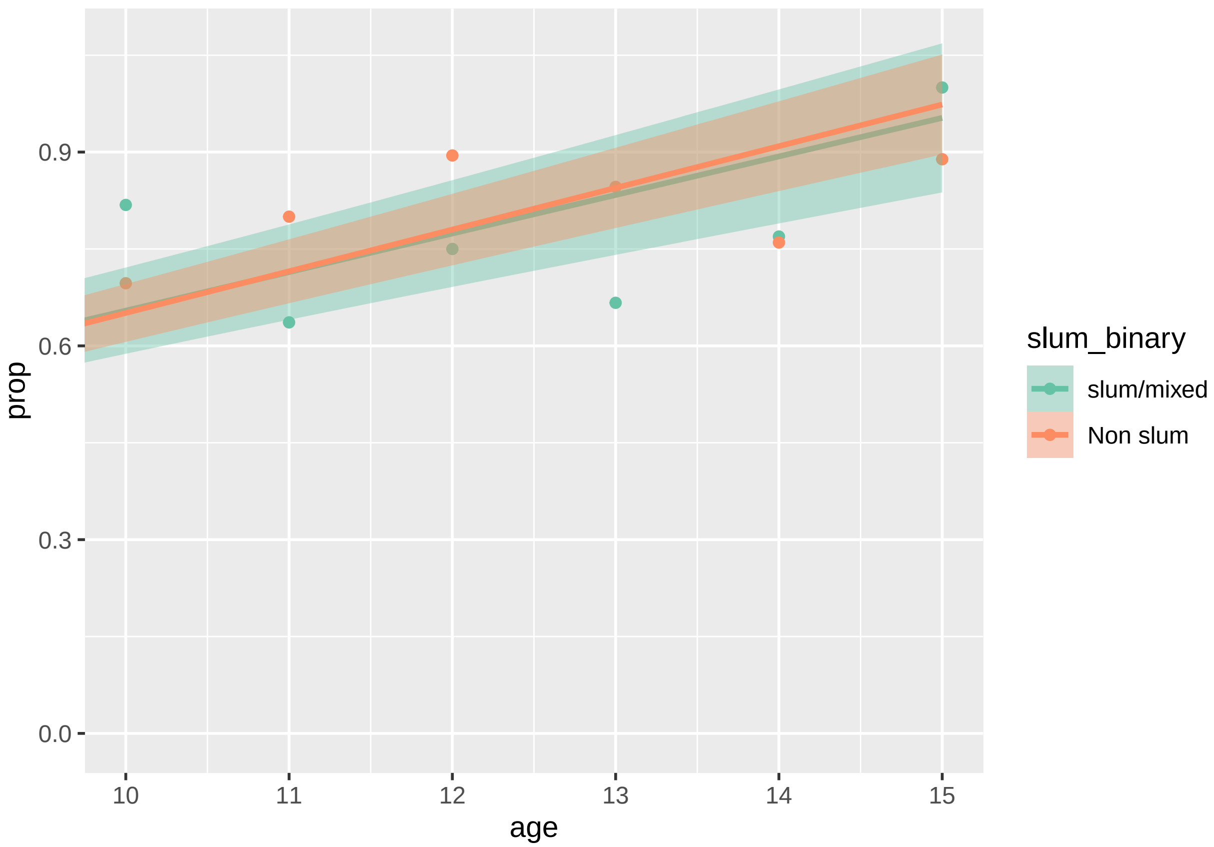
10. Using the `sero_group` dataset, create a scatter plot of `age` (x axis) and `prop` (y axis) grouped by `slum_binary` and then add a fit line using GAM method also for each value of the `slum_binary` variable. - hint: use `color=slum_binary` and `fill=slum_binary` as mapped aesthetics - use the RColorBrewer color palette "Set2" to for both the colors (point and fit line) and the fills (95% CI ribbon).

```
 ggplot(sero_group, aes(x=age, y=prop, col=slum_binary, fill=slum_binary)) +
  geom_point() +
  stat_smooth(method="gam", formula = y ~ x) +
  scale_color_brewer(palette = "Set2") +
scale_fill_brewer(palette = "Set2")
```



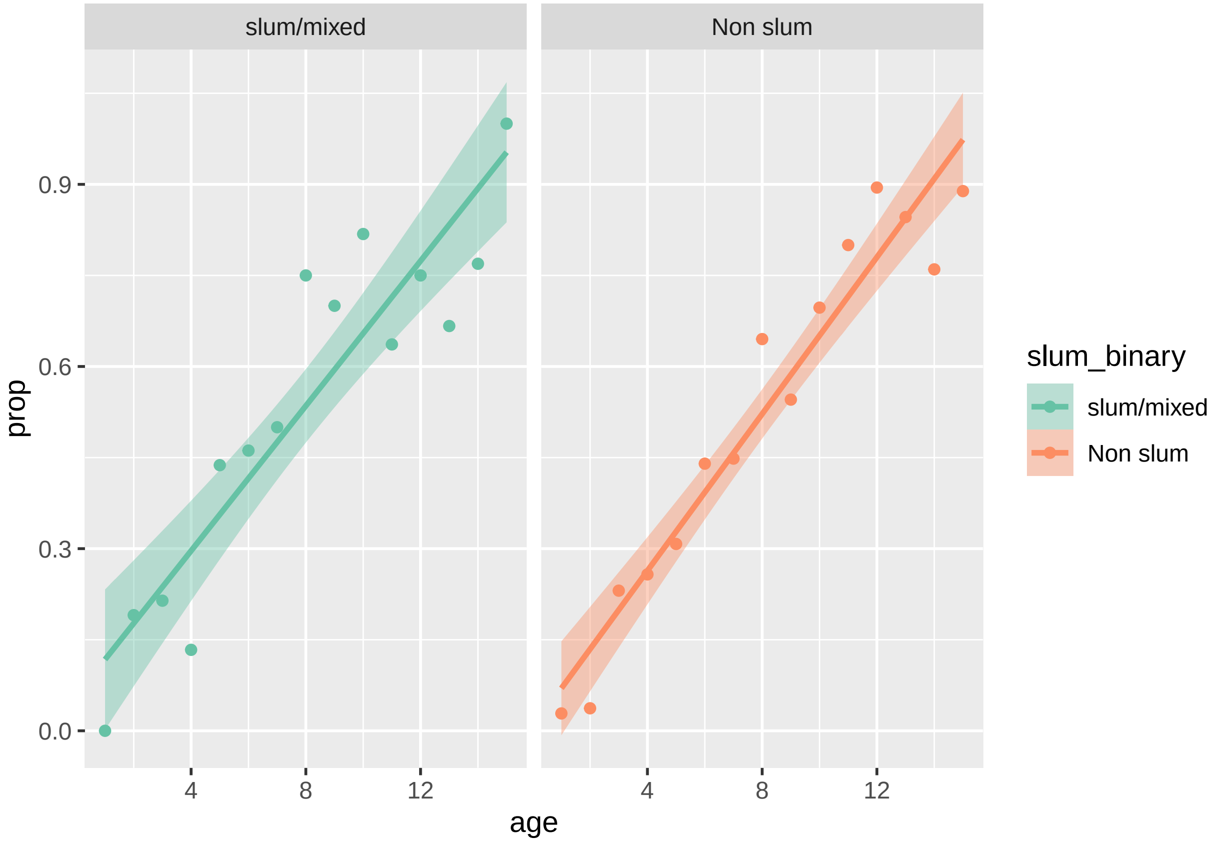
11. Zoom in on the plot from #10 for ages 10 to 15 using `coord_cartesian()`

```
ggplot(sero_group, aes(x=age, y=prop, col=slum_binary, fill=slum_binary)) +
  geom_point() +
  stat_smooth(method="gam", formula = y ~ x) +
  scale_color_brewer(palette = "Set2") +
scale_fill_brewer(palette = "Set2") +
  coord_cartesian(xlim = c(10, 15))
```



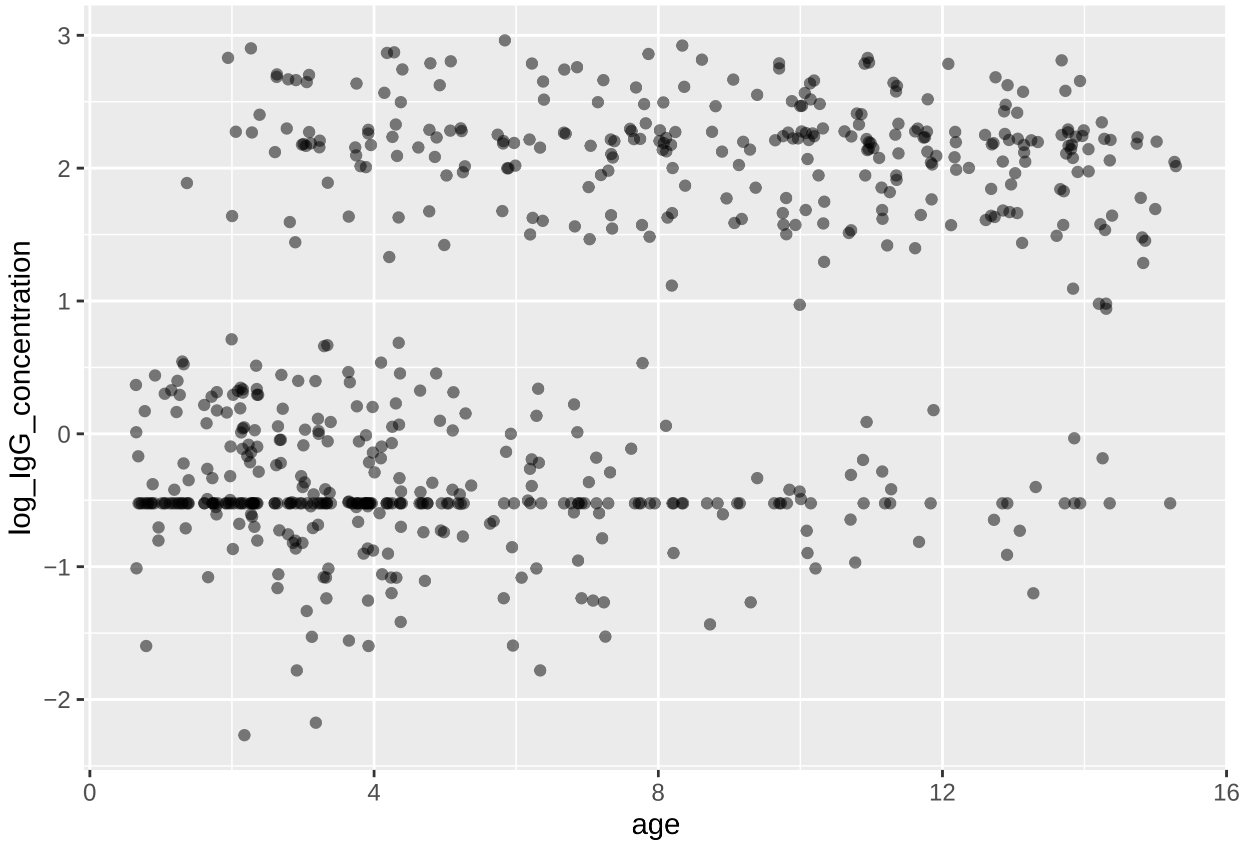
12. Make the plot from #10 easier to read using `facet_wrap()` to create a plot for each `slum_binary` value

```
ggplot(sero_group, aes(x=age, y=prop, col=slum_binary, fill=slum_binary)) +
  geom_point() +
  stat_smooth(method="gam", formula = y ~ x) +
  scale_color_brewer(palette = "Set2") +
scale_fill_brewer(palette = "Set2") +
  facet_wrap(. ~slum_binary)
```



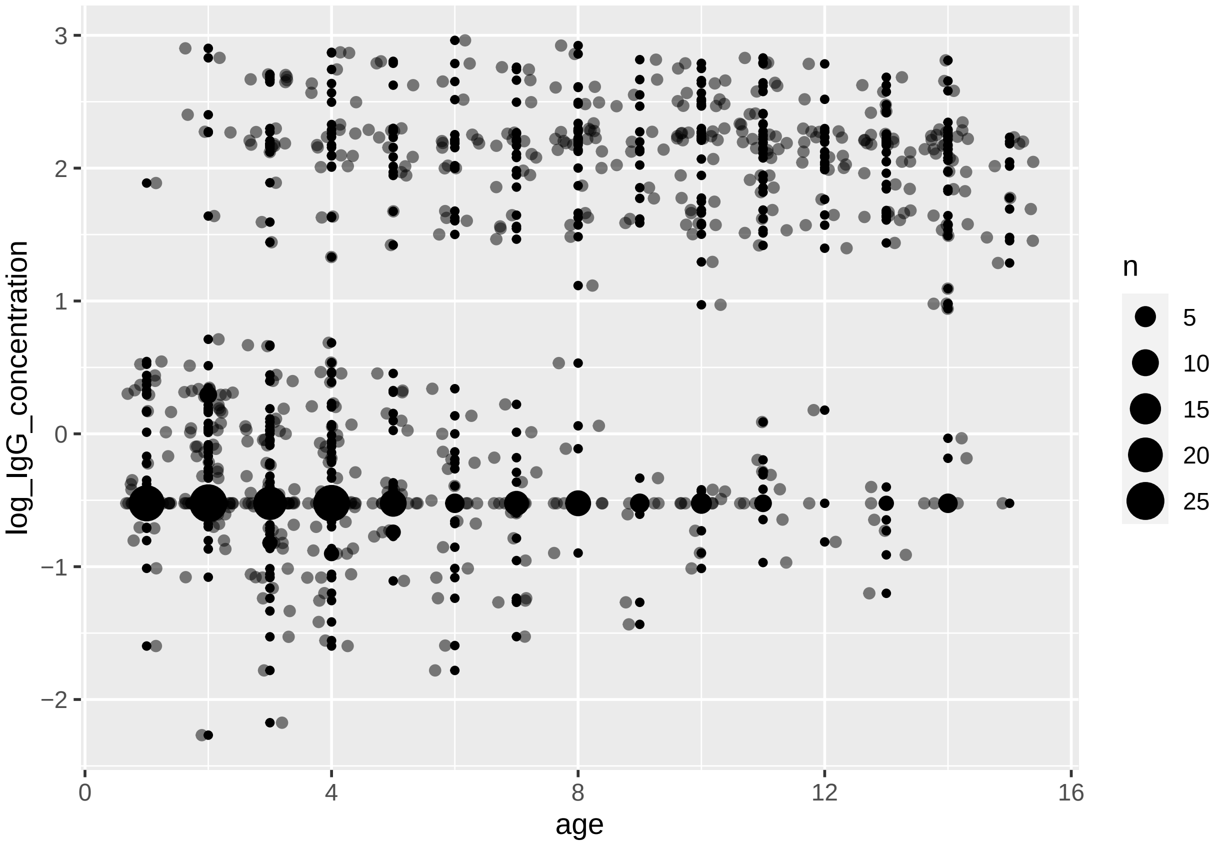
13. Using the sero dataset, create a scatter plot of log_IgG_concentration (y axis) and age (x axis) using geom_jitter() where the alpha attribute is geom_jitter() is set to 0.5

```
ggplot(sero, aes(x=age, y=log_IgG_concentration)) +
  geom_jitter(alpha = 0.5)
```



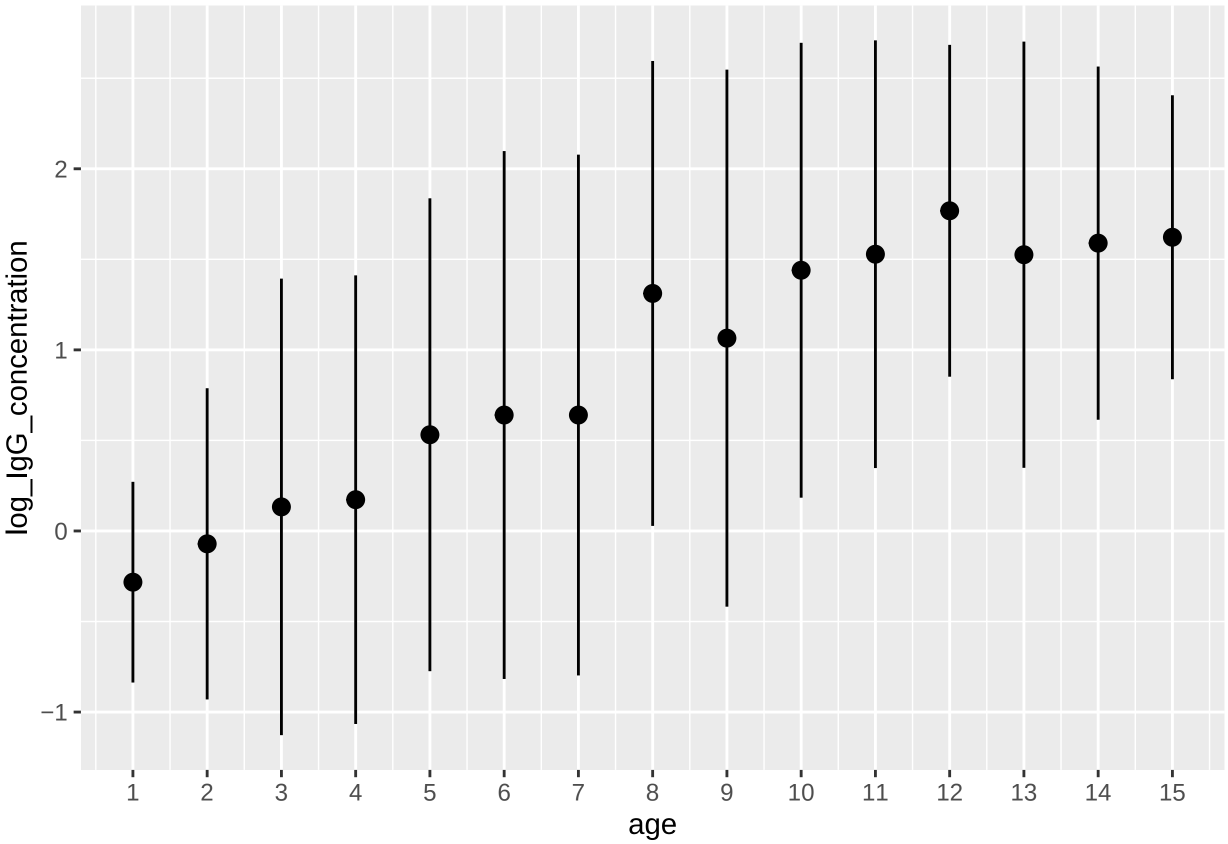
14. Add an additional plot layer to #13 of `stat_sum()` to count the number of observations at each `age` and then map the count onto size as the point area.

```
ggplot(sero, aes(x=age, y=log_IgG_concentration)) +
  stat_sum() +
  geom_jitter(alpha = 0.5)
```



15. Using the `sero` dataset, plot the mean `log_IgG_concentration` (y axis) as points and 1 standard deviation plus or minus the mean `log_IgG_concentration` as vertical lines for each age in years (x axis). - hint: `stat_summary(fun.data = mean_sdl)` - hint: use the `mult` argument to specify 1 standard deviation - change the x axis ticks so that all ages 1 though 15 are displayed using `breaks()`

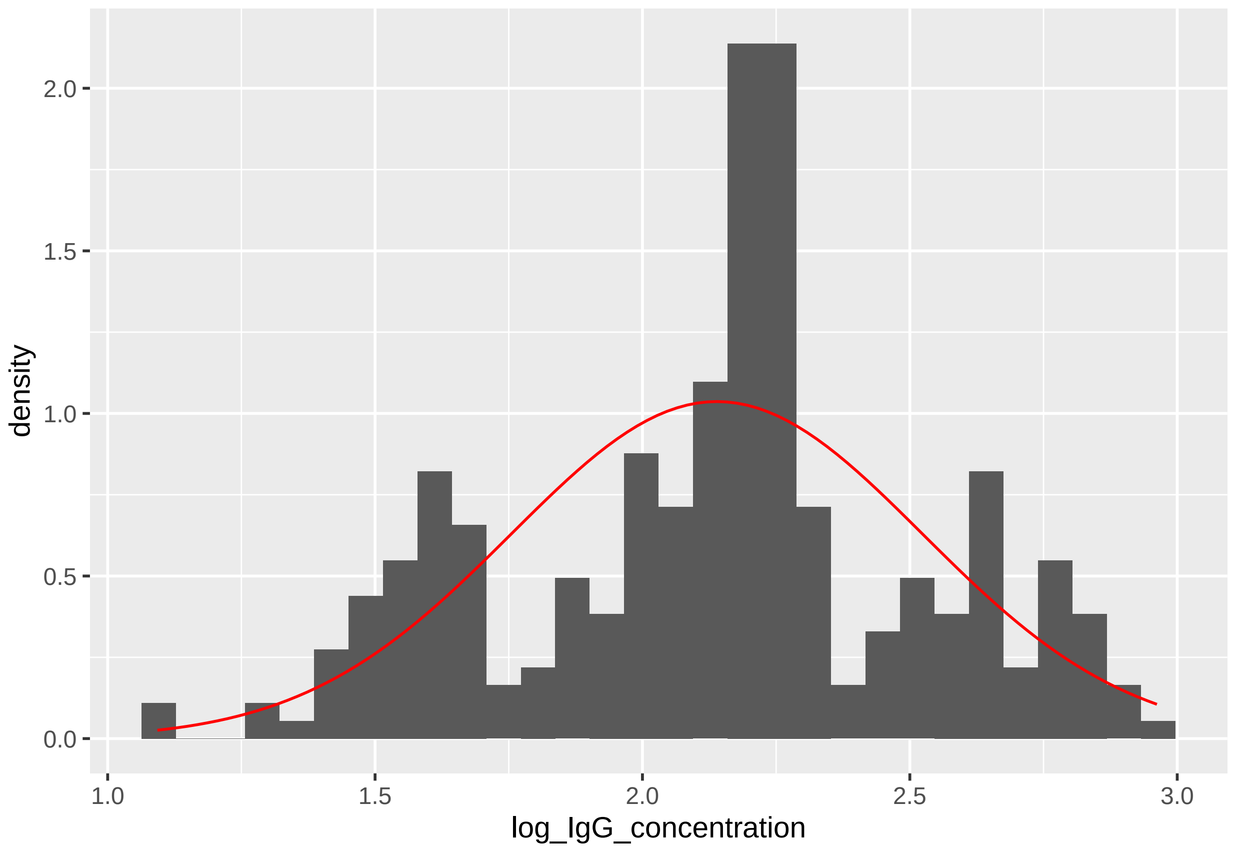
```
ggplot(sero, aes(x = age, y = log_IgG_concentration)) +
  stat_summary(fun.data = mean_sdl, fun.args = list(mult=1)) +
    scale_x_continuous(breaks=seq(1:15))
```



16. Using the sero dataset, evaluate whether log_IgG_concentration is normally distributed among observations in which sero_status is equal to 1. Answer below the R chunk if it appears to be normally distributed? - hint: subset data using filter() to only include samples where sero_status is equal to 1 then use stat_function()

```
sero_filter <- filter(sero, sero_status == 1)
ggplot(sero_filter, aes(x=log_IgG_concentration)) +
  geom_histogram(aes(y = ..density..)) +
  stat_function(fun = dnorm, color="red", args = list(mean = mean(sero_filter$log_IgG_concentration),
                                                       sd = sd(sero_filter$log_IgG_concentration)))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# log_IgG_concentration appears to be normally distributed
```

17. Using the `sero` dataset, compute the correlation between the `age` and `log_IgG_concentration` variables. What does this tell you about the relationship between the two variables?

```
x_age <- sero %>% pull(age)
y_log_IgG_concentration <- sero %>% pull(log_IgG_concentration)
cor(x_age, y_log_IgG_concentration)
```

```
## [1] 0.4859566
```

```
# r= 0.49 indicates a positive weak correlation between age and log_IgG_concentration.
```

18. Using the `sero` dataset, perform a t-test to determine if there is evidence of a difference between the mean `log_IgG_concentration` and 2.0. Is this a one or two sample t-test? Interpret the results.

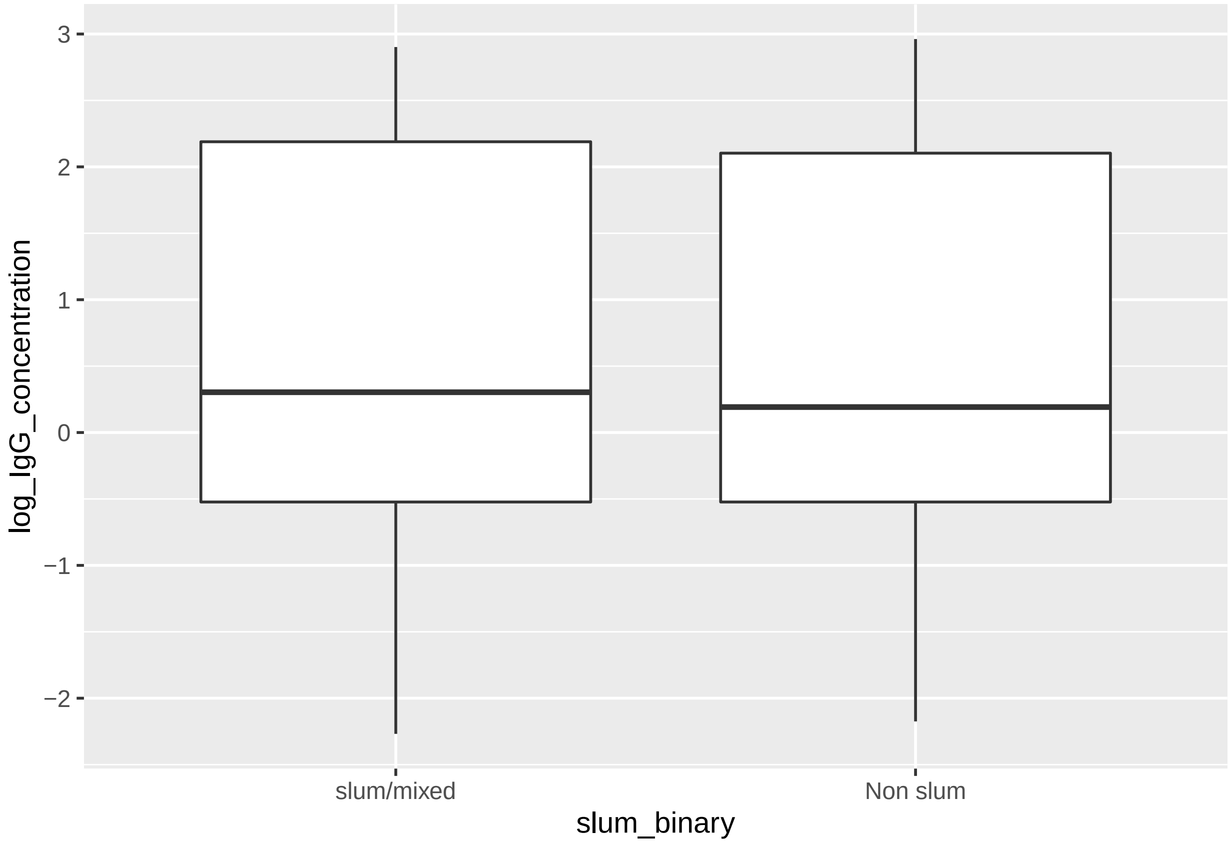
```
t.test(sero$log_IgG_concentration, mu = 2)
```

```
##
##  One Sample t-test
##
## data:  sero$log_IgG_concentration
## t = -24.817, df = 650, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 2
## 95 percent confidence interval:
##  0.5881397 0.7951780
## sample estimates:
## mean of x
## 0.6916588
```

```
# it is a one sample t-test, we reject the null hypothesis(t = -24.82, d.f =650, p<0.01) that the mean
#log_IgG_concentration is equal to 2, given alpha 0.05.
```

19. Using `sero` dataset, draw a box plot of `log_IgG_concentration` for the two different values in `slum_binary`.

```
sero %>%
  ggplot(aes(x = slum_binary, y = log_IgG_concentration)) +
         geom_boxplot()
```



20. Using the `sero` dataset, perform a t-test to determine if there is evidence of a difference between the mean `log_IgG_concentration` among individuals with residence in `slum/mixed` and individuals with residence in `Non slum` per the `slum_binary` variable. Is this a one or two sample t-test? Interpret the results.

```
slumone <- sero %>% filter(slum_binary == "slum/mixed") %>% pull(log_IgG_concentration)
slumtwo <- sero %>% filter(slum_binary == "Non slum") %>% pull(log_IgG_concentration)
t.test(slumone, slumtwo)
```

```
##
##  Welch Two Sample t-test
##
## data:  slumone and slumtwo
## t = 0.46034, df = 317.44, p-value = 0.6456
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1795809  0.2892832
## sample estimates:
## mean of x mean of y
## 0.7313437 0.6764926
```

```
# this is a two-sample t-test, we fail to reject the null hypothesis(t=0.460, d.f= 317.44, p=0.65) that
# the difference in the mean log_IgG_conentration among individuals with residence in slum/mixed and
# individuals with residence in Non slum is 0.
```

21. Using the `sero` dataset, fit a linear regression model with `log_IgG_concentration` as the dependent variable (outcome) and `age`, `gender`, and `slum_binary` as independent variables (covariates). Save the model fit in an object called "lmfit_sero" and display the summary table. Interpret the results.

```
lmfit_sero <- glm(log_IgG_concentration ~ age + gender + slum_binary, data = sero)
summary(lmfit_sero)
```

```
##
## Call:
## glm(formula = log_IgG_concentration ~ age + gender + slum_binary,
##     data = sero)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.88718  -0.83069  -0.09624   0.82590   2.94467
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -0.24249    0.12602  -1.924   0.0548 .
## age                  0.15721    0.01116  14.085   <2e-16 ***
## genderMale          -0.11467    0.09242  -1.241   0.2151
## slum_binaryNon slum -0.06110    0.10319  -0.592   0.5540
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for gaussian family taken to be 1.384501)
##
##     Null deviance: 1176.04  on 650  degrees of freedom
## Residual deviance:  895.77  on 647  degrees of freedom
## AIC: 2065.2
##
## Number of Fisher Scoring iterations: 2
```

```
#log_IgG_concentration decreases with increasing age.
# we found a significant relationship between log_IgG_concentration and age(p<0.01).
# we found an insignificant relationship between log_IgG_concentration and gender(p = 0.22)
# we found an insignificant relationship between log_IgG_concentration and slum_binary(p=0.55)
```

22. Using the `sero` dataset, fit a logistic regression model with `sero_status` as the dependent variable (outcome) and `age`, `gender`, and `slum_binary` as independent variables (covariates). Save the model fit in an object called "logfit_sero" and display the summary table. Interpret the results.

```
logfit_sero <- glm(sero_status ~ age + gender + slum_binary, data = sero)
summary(logfit_sero)
```

```
##
## Call:
## glm(formula = sero_status ~ age + gender + slum_binary, data = sero)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.96138  -0.26060  -0.09882   0.31522   0.95013
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.033708   0.044611   0.756    0.450
## age                    0.065108   0.003951  16.477   <2e-16 ***
## genderMale            -0.033541   0.032718  -1.025    0.306
## slum_binaryNon slum -0.015406   0.036530  -0.422    0.673
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1735031)
##
##     Null deviance: 159.98  on 650  degrees of freedom
## Residual deviance: 112.26  on 647  degrees of freedom
## AIC: 713.18
##
## Number of Fisher Scoring iterations: 2
```

```
#log_IgG_concentration increases with increasing age.
# we found a significant relationship between sero_status and age(p<0.01).
# we found an insignificant relationship between sero_status and gender(p = 0.31)
# we found an insignificant relationship between sero_status and slum_binary(p=0.68)
```

23. Based on the object called "logfit_sero" calculate odds ratio and 95% confidence intervals of the odds ratio for `slum_binary`.

```
logfit_sero2 <- glm(sero_status ~ slum_binary, data = sero, family = binomial(link = "logit"))
summary(logfit_sero2)
```

```
##
## Call:
## glm(formula = sero_status ~ slum_binary, family = binomial(link = "logit"),
##     data = sero)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.084  -1.062  -1.062   1.297   1.297
##
```

```
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.22314    0.15000  -1.488    0.137
## slum_binaryNon slum -0.05464    0.17651  -0.310    0.757
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 891.35  on 650  degrees of freedom
## Residual deviance: 891.25  on 649  degrees of freedom
## AIC: 895.25
##
## Number of Fisher Scoring iterations: 3
```

```
exp(logfit_sero2$coefficients)
```

```
##         (Intercept) slum_binaryNon slum
##           0.8000000           0.9468284
```