Announcement

P8101S20-Homework_06

Neal Kar

3/10/2020

Announcement

Please do not add code folding (code_folding: hide) to your YAML Header or echo = FALSE to your RMD code chunk options. In order to accurately grade your HTML files we need to be able to see all of your code. Thank you!

Question 1

You have received data for a small observational study on OCD. Patients were measured using the YBOCS scale over eight months. The YBOCS data is saved in the `data/ocd_longitudinal.csv` *file and some basic demographics are stored in the* `data/ocd_demos.csv` *file.*

a)

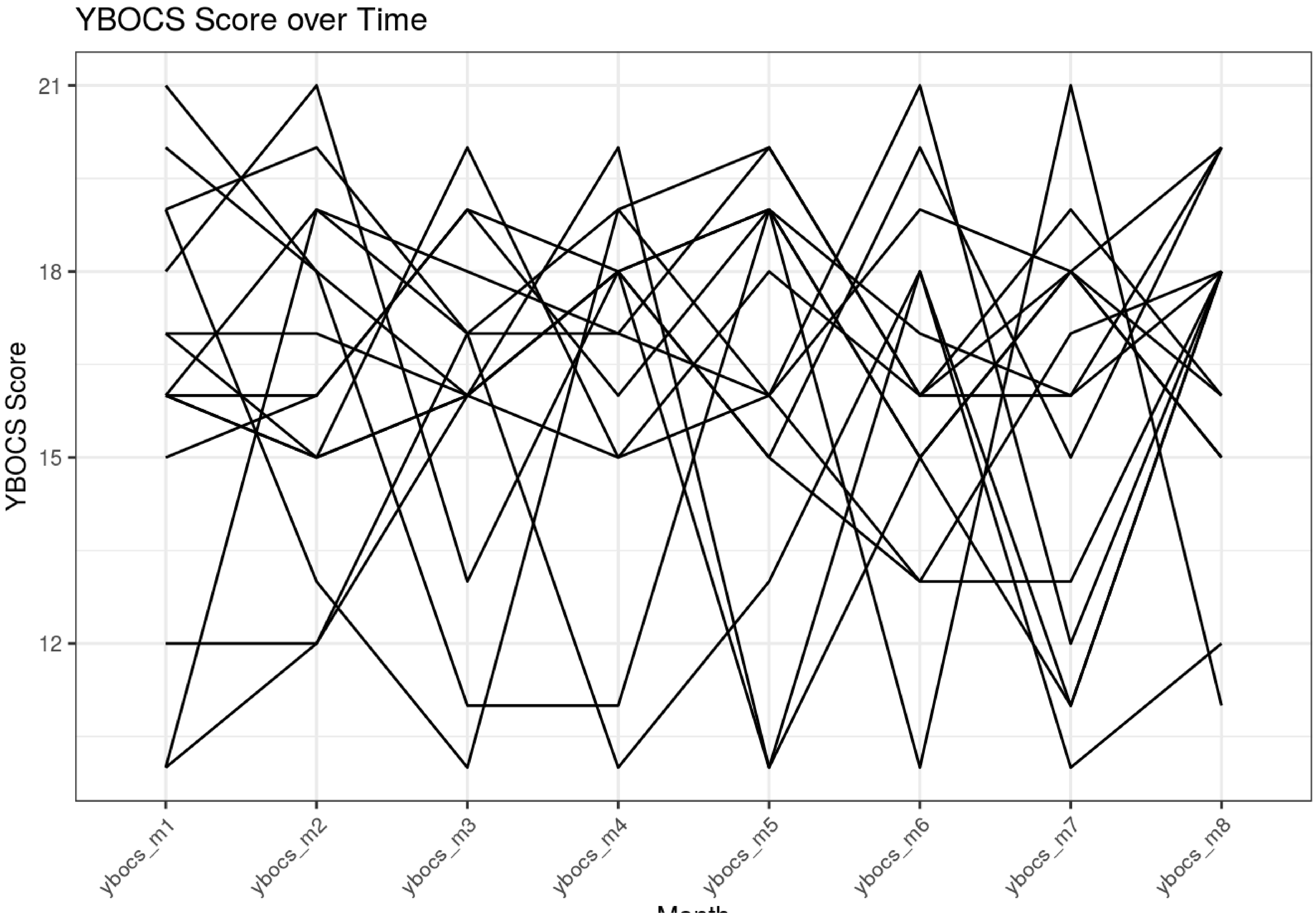
Load both datasets into R and join them using a method that will drop any individuals who are not in both datasets. Report the number of variables and the number of observations in this joined dataset.

```r
#Read in the files
ocd_demos_df <- read_csv("data/ocd_demos.csv")
ocd_obs_df <- read_csv("data/ocd_longitudinal.csv")

#Use inner join to combine df's and drop obs not in both
ocd_df_wide <- inner_join(ocd_demos_df, ocd_obs_df, by = "study_id")
```

b)

You want to produce a spaghetti plot which shows YBOCS scores over time for all individuals. In order to do this, you must transform the data from wide to long format. Use `pivot_longer()` to get the data into long format, and then create a spaghetti plot that displays all individuals’ YBOCS trajectories over the eight months.

```
#Create a long format df
ocd_df_long <- pivot_longer(data = ocd_df_wide,
                             cols = starts_with("ybocs"),
                             names_to = "month",
                             values_to = "ybocs")


#Produce the spaghetti plot
ggplot(data = ocd_df_long) +
  geom_line(aes(x=month, y=ybocs, group=study_id)) +
  theme_bw() +
  theme(text = element_text(size=10),
        axis.text.x = element_text(angle=45, hjust=1)) +
  labs(title="YBOCS Score over Time", x="Month", y="YBOCS Score")
```


YBOCS Score over Time
YBOCS Score
21
18
15
12
ybocs_m1
ybocs_m2
ybocs_m3
ybocs_m4
ybocs_m5
ybocs_m6
ybocs_m7
ybocs_m8
Month

Month

c)

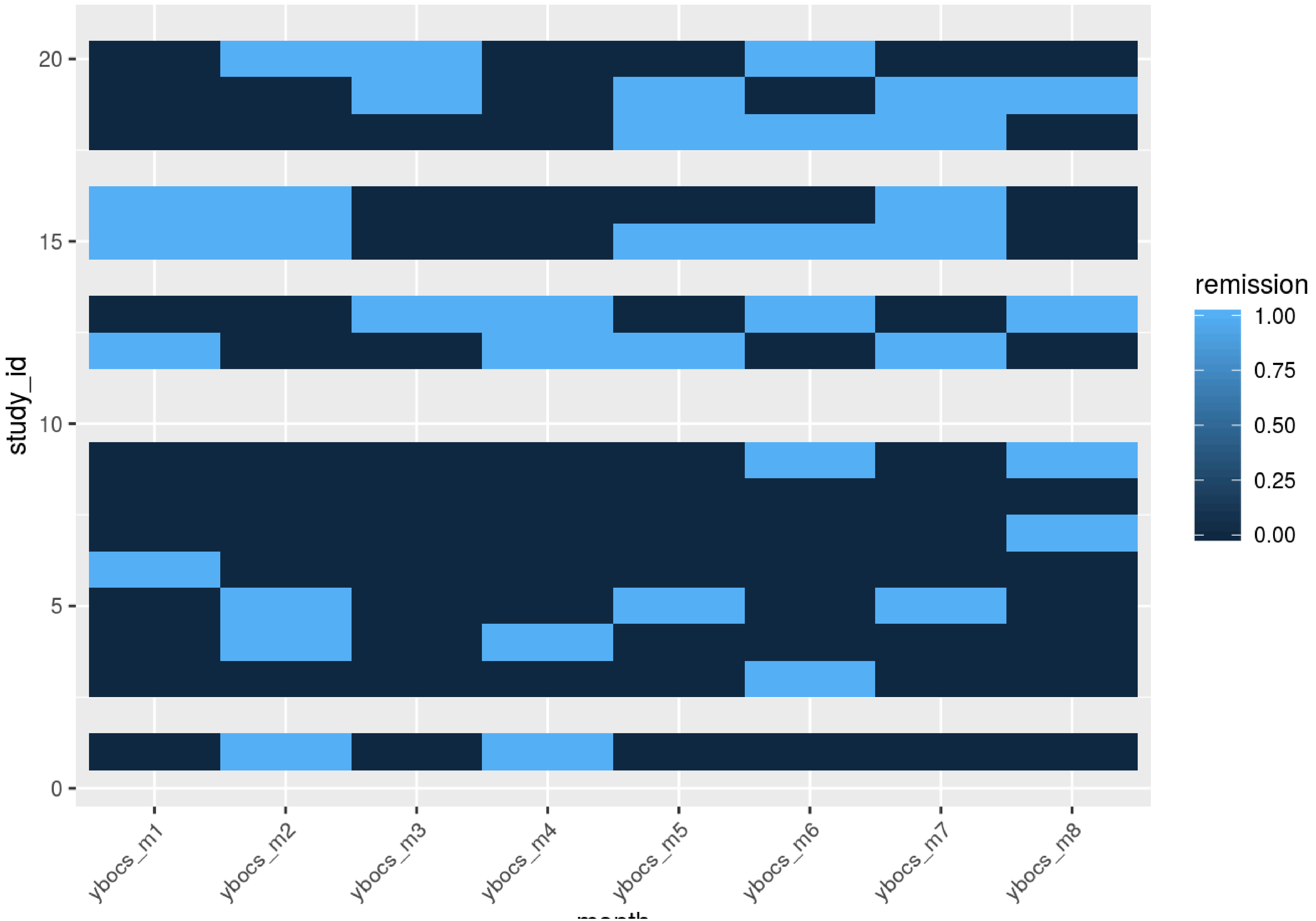
The investigators you are working with want to create a new categorical variable called `remission` which equals 1 if the YBOCS score is less than or equal to 15 and 0 otherwise. Use `mutate()` to create this new categorical variable in your long dataset.

```
ocd_df_long <- ocd_df_long %>%
  mutate(remission = if_else(ybocs <= 15, 1, 0))
```

d)

Next, create a plot that shows the binary `remission` variable over time for each individual. You can use `geom_tile()` to do this, with month mapped to the X axis, patient ID mapped to the Y axis, and fill mapped to the `remission` variable. This type of plot for a categorical variable is called a “lasagna” plot in contrast to the earlier “spaghetti” plot you created.

```
ggplot(ocd_df_long) +
  geom_tile(aes(x=month, y=study_id, fill=remission)) +
  theme(text = element_text(size=10),
        axis.text.x = element_text(angle=45, hjust=1))
```

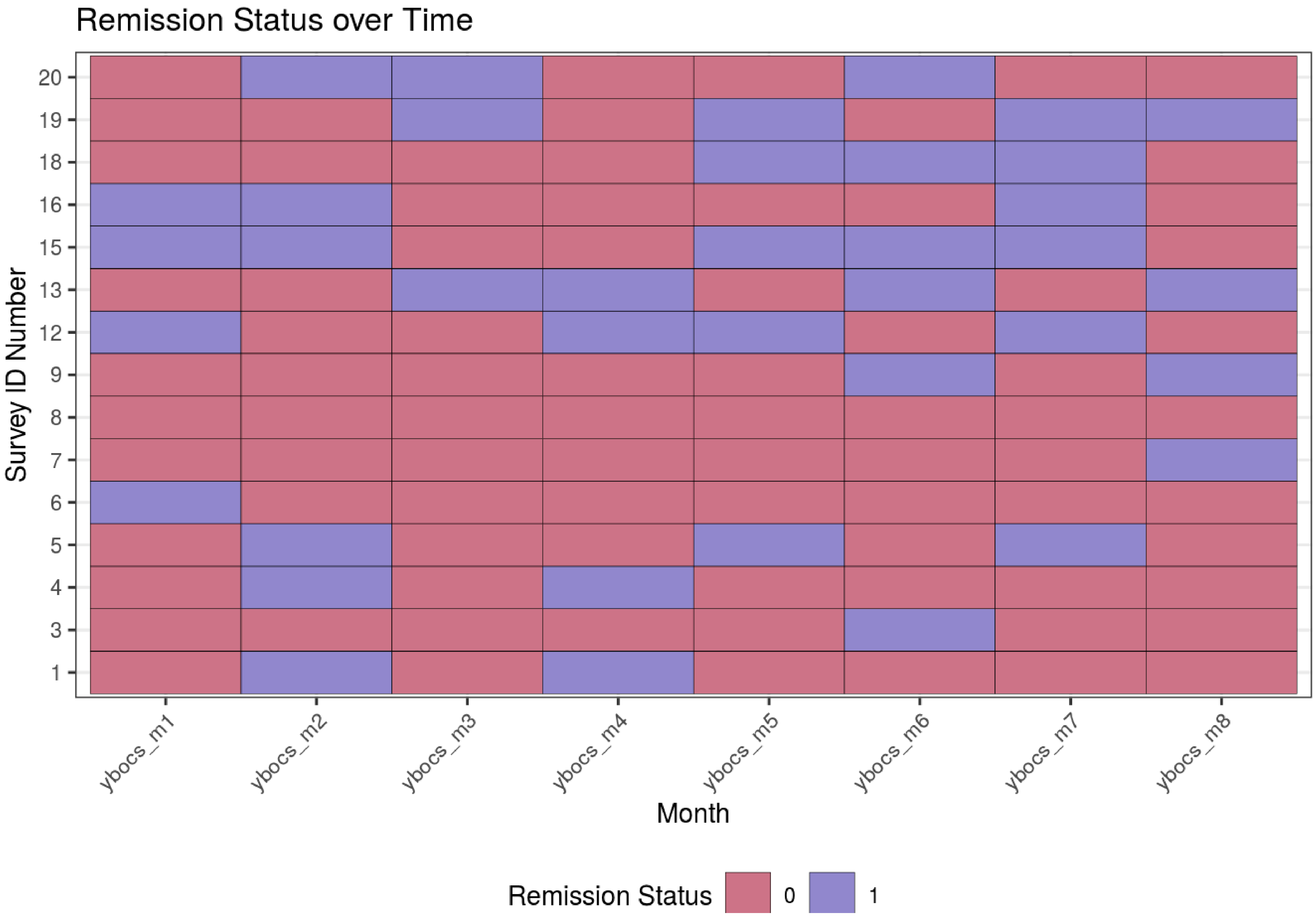

20
15
10
5
0
study_id
ybocs_m1
ybocs_m2
ybocs_m3
ybocs_m4
ybocs_m5
ybocs_m6
ybocs_m7
ybocs_m8
month
remission
1.00
0.75
0.50
0.25
0.00

month

e)

Improve the lasagna plot from part (d) in the following ways: (1) If you have not already, make your ID variable and your remission variable into factor variables when plotting. (2) Add an appropriate title to the plot and add titles for the x and y axes. (3) Give the legend an appropriate title and place it at the bottom of the plot. (4) Make the panel background white or blank. (5) Manually set custom colors for the fill aesthetic. (6) Set the color aesthetic in geom_tile() equal to “black”. Explain what you think this did!

```
#Convert ID and remission variables into factor data types
ocd_df_long <- ocd_df_long %>%
  mutate(study_id = as.factor(study_id),
         remission = as.factor(remission))

#Improve the graph
ggplot(ocd_df_long) +
  geom_tile(aes(x=month, y=study_id, fill=remission), color="black")+
  scale_fill_discrete(h=c(0,270), c=60, l=60,
                      h.start=0,direction =1, aesthetics="fill") +
  theme_bw() +
  theme(text = element_text(size=10), axis.text.x = element_text(angle=45, hjust=1),
legend.position = "bottom") +
  labs(title = "Remission Status over Time", x = "Month",
       y = "Survey ID Number", fill="Remission Status")
```

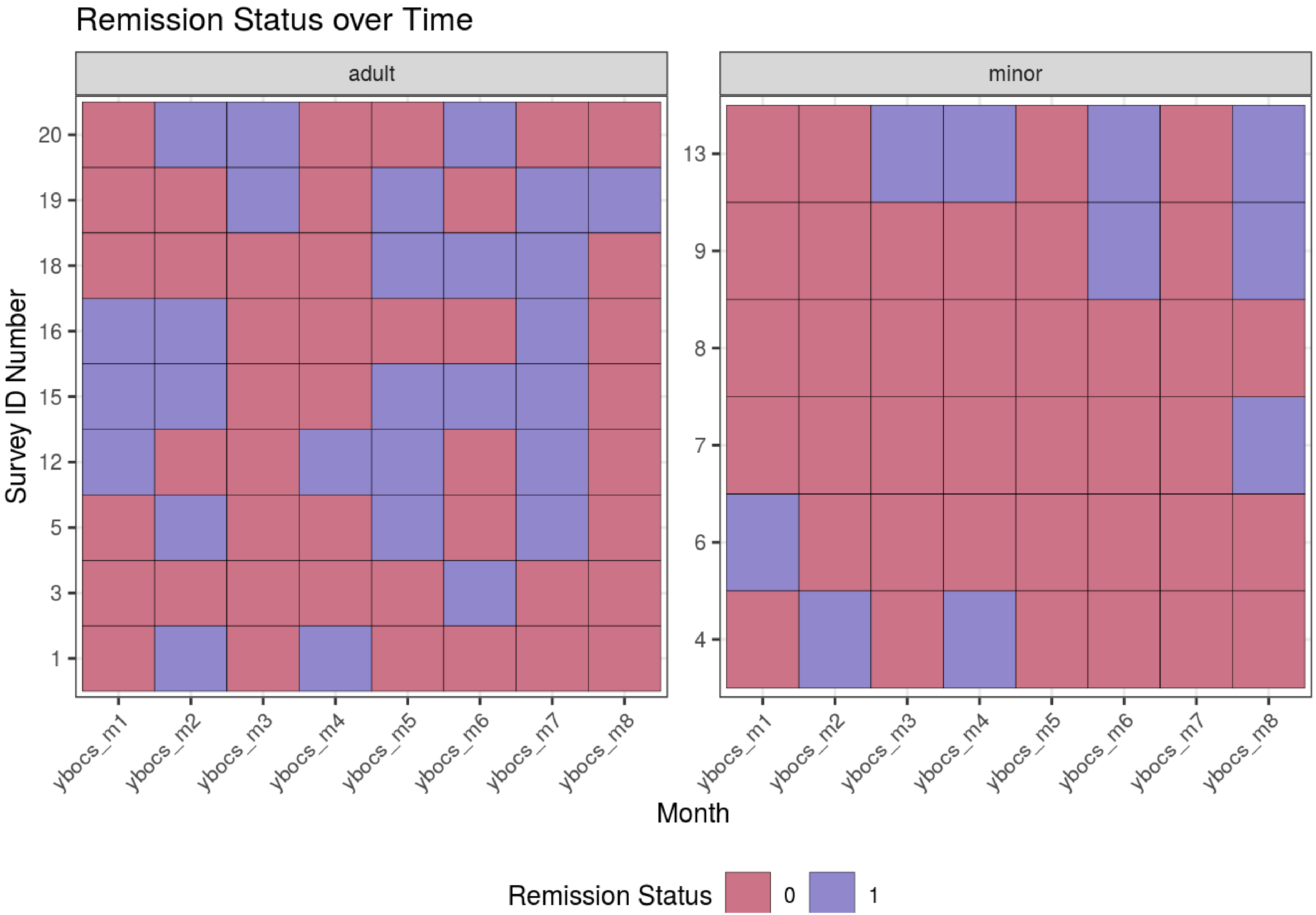

Remission Status over Time
Survey ID Number
20
19
18
16
15
13
12
9
8
7
6
5
4
3
1
ybocs_m1
ybocs_m2
ybocs_m3
ybocs_m4
ybocs_m5
ybocs_m6
ybocs_m7
ybocs_m8
Month
Remission Status
0
1

Setting the color aesthetic in geom_tile() equal to “black” makes the color of the tile borders black.

f)

Create a new categorical age variable called `age_cat` with two categories: “minor” if the individual is less than 18 years old, and “adult” if the individual is 18 years old or greater. Now facet your plot from (e) by this `age_cat` variable. In your facet_wrap() statement, add a scales = “free” argument. Explain what this did.

```
#Create new dichotomous variable
ocd_df_long <- ocd_df_long %>%
  mutate(age_cat = if_else(age < 18, "minor", "adult"))

#Facet wrap
ggplot(ocd_df_long) +
  geom_tile(aes(x=month, y=study_id, fill=remission), color="black") +
  facet_wrap(~age_cat, scales = "free") +
  scale_fill_discrete(h=c(0,270), c=60, l=60,
                      h.start=0,direction =1, aesthetics="fill") +
  theme_bw() +
  theme(text = element_text(size=10), axis.text.x = element_text(angle=45, hjust=1),
legend.position = "bottom") +
  labs(title = "Remission Status over Time", x = "Month",
       y = "Survey ID Number", fill = "Remission Status")
```

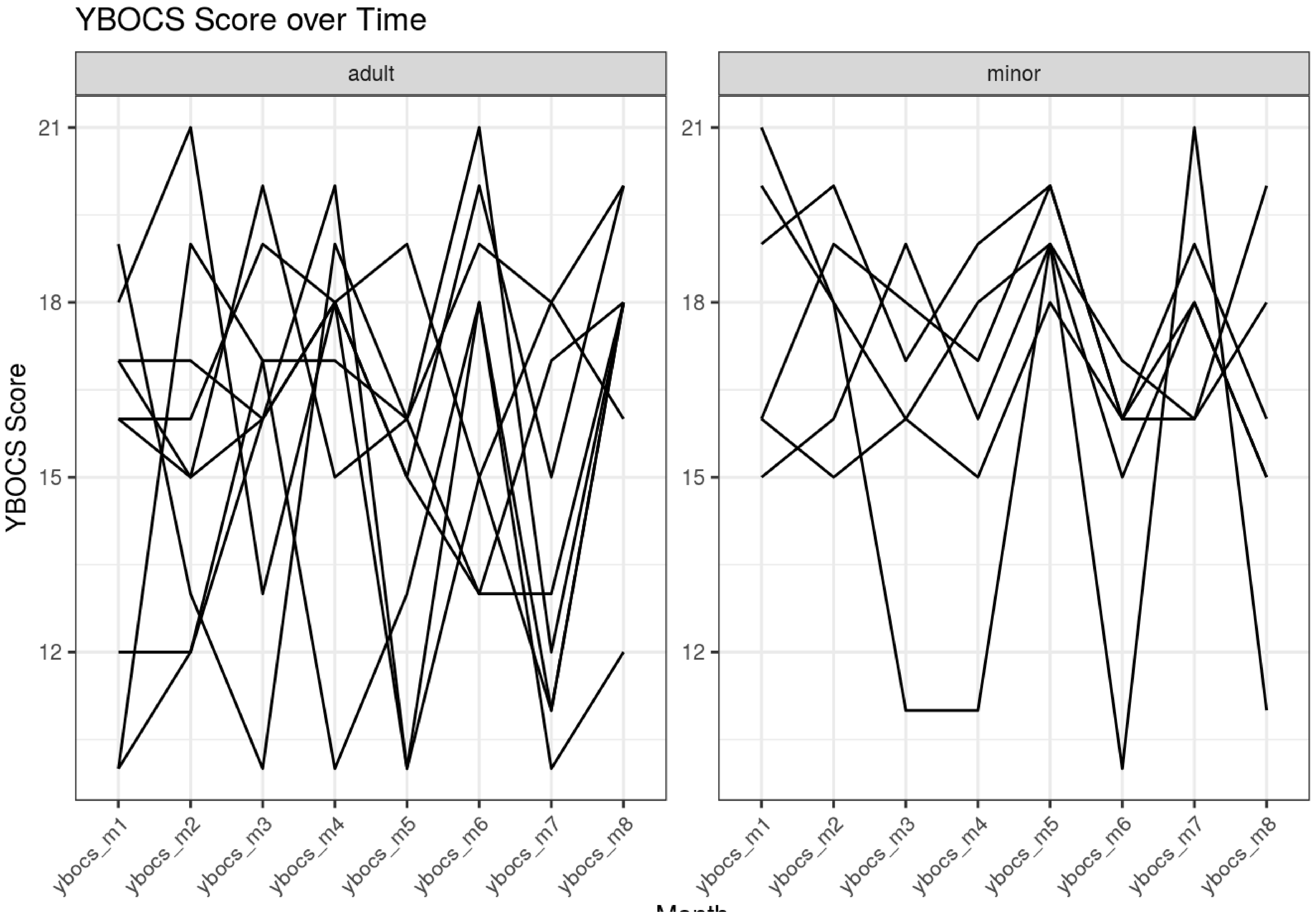

Remission Status over Time
adult
minor
Survey ID Number
20
19
18
16
15
12
5
3
1
13
9
8
7
6
4
ybocs_m1
ybocs_m2
ybocs_m3
ybocs_m4
ybocs_m5
ybocs_m6
ybocs_m7
ybocs_m8
Month
Remission Status
0
1

Adding a scales = “free” argument to the facet_wrap() statement made it so that the y-axis on each heatmap only showed the y-axis values pertaining to that heatmap. Before that, each heatmap shared the same y-axis. However, each y-axis value did not pertain to both heatmaps, so there were empty spaces in each heatmap. The additional statement separates out the relevant y-axis values.

g)

Facet your spaghetti plot in (b) by the new `age_cat` variable. Do you notice any differences in YBOCS score trajectories between adults and minors based on these spaghetti plots? Answer in a sentence.

```
#Facet wrap spaghetti plot
ggplot(data = ocd_df_long) +
  geom_line(aes(x=month, y=ybocs, group=study_id)) +
  facet_wrap(~age_cat, scales = "free") +
  theme_bw() +
  theme(text = element_text(size=10),
        axis.text.x = element_text(angle=45, hjust=1)) +
  labs(title="YBOCS Score over Time", x="Month", y="YBOCS Score")
```


YBOCS Score over Time
adult
minor
YBOCS Score
21
18
15
12
ybocs_m1
ybocs_m2
ybocs_m3
ybocs_m4
ybocs_m5
ybocs_m6
ybocs_m7
ybocs_m8
Month

Month

No, I don’t notice any difference in trajectories based on the spaghetti plots.

Question 2

You have been given a dataset that describe a study on the association between a new biomarker for inflammation (biomarker_A15) as measured by four different assays and body temperature. The `biomarker_details.csv` *includes medical information such as temperature, infection status (0/1), and biomarker data.*

a)

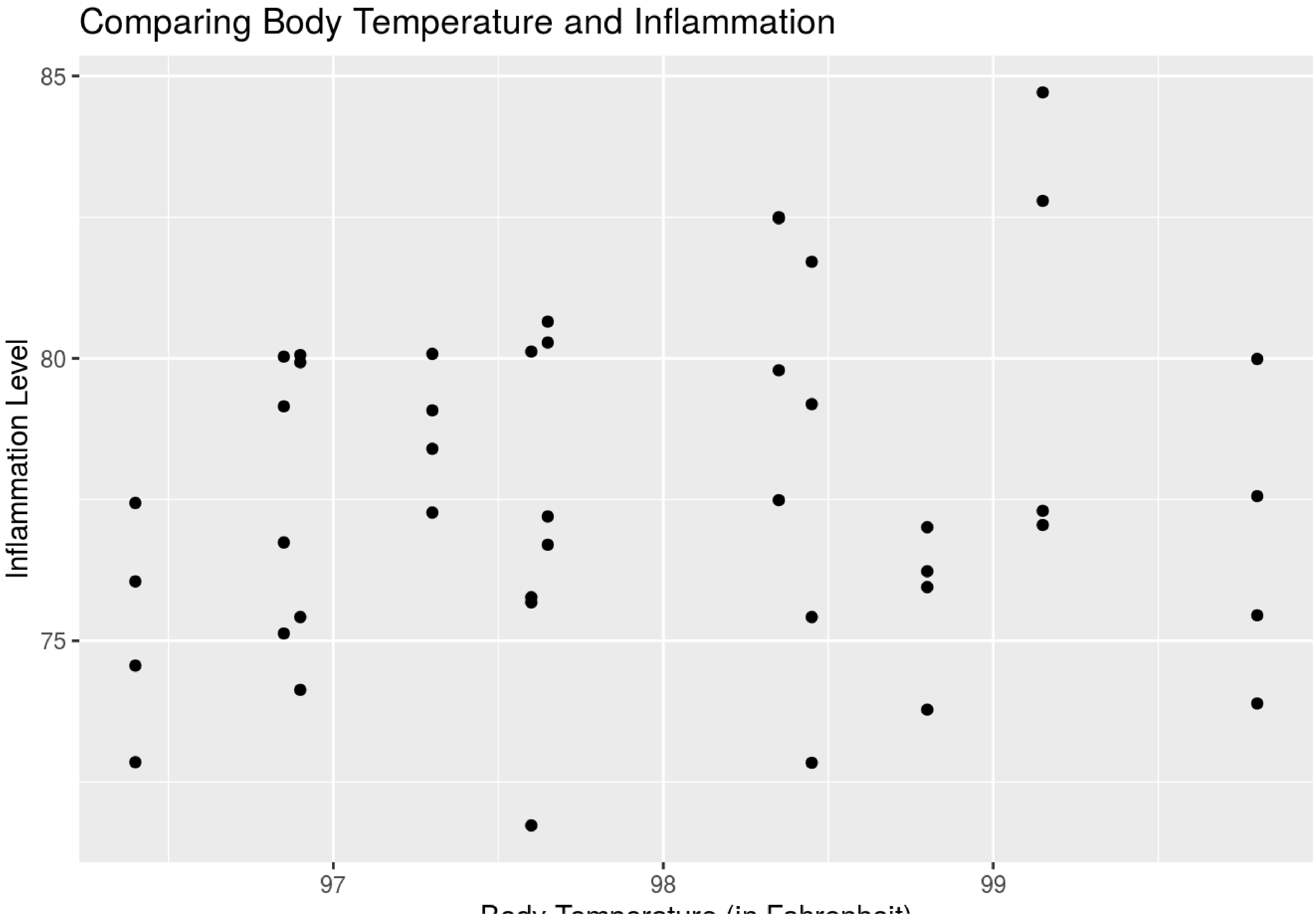
Load in the dataset and transform the data from wide to long format. Make sure to check your data – each individual should have four biomarker readings.

```
#Read in data
biomarker_df_wide <- read_csv("data/biomarker_details.csv")

#Convert to long format
biomarker_df_long <- pivot_longer(data = biomarker_df_wide,
                                  cols = starts_with("bio_assay"),
                                  names_to = "assay",
                                  values_to = "inflammation")
```

b)

Graph the relationship between temperature and the biomarker. What do you think, does there seem to be a linear relationship?

```r
ggplot(data = biomarker_df_long) +
  geom_point(aes(x=body_temp, y=inflammation)) +
  labs(title="Comparing Body Temperature and Inflammation",
       x="Body Temperature (in Fahrenheit)", y="Inflammation Level")
```


Comparing Body Temperature and Inflammation
Inflammation Level
85
80
75
97
98
99
Body Temperature (in Fahrenheit)

Body Temperature (in Fahrenheit)

No, there doesn't seem to be a linear relationship between body temperature and biomarker inflammation level.

c)

Fit a simple linear regression with the new biomarker as the outcome and body temperature as the sole predictor. Report the model's R-squared and omnibus F test p-value in a sentence using in-line coding. Hint: use the glance() function to easily obtain what you need.

```
#Regression model
biomarker_model <- lm(inflammation ~ body_temp,
                      data = biomarker_df_long)

#Put fit statistics in a df
biomarker_model_fit <- glance(biomarker_model)
```

The model's R-squared value is 0.0198069, and the model's omnibus F test p-value is 0.3621819.

d)

What percentage of the variance in biomarker is explained by body temperature? Is the model from part (d) a useful model? Answer in one or two sentences and use information from part d in your answer.

```
biomarker_model_r_squared <- biomarker_model_fit$r.squared
```

About 1.98% of the variance in biomarker inflammation level is explained by body temperature. Since that’s a very small percentage, the model in part (c) is not a useful model.

e)

Fit a linear regression model with the new biomarker as outcome and both body temperature and infection status as predictors. Report this new model’s R-squared and omnibus F test p-value in a sentence using in-line coding.

```
#Regression model
biomarker_model_multi <- lm(inflammation ~ body_temp + infection_status, data = biom
arker_df_long)

#Put fit statistics in a df
multi_model_fit <- glance(biomarker_model_multi)
```

The model’s R-squared value is 0.2565346, and the model’s omnibus F test p-value is 0.0022953.

f)

Report a table of the parameter estimates from the model in (f) using tidy() and kable(). Interpret the parameter estimates for body temperature and infection status in sentences using the words of the problem.

```
#Put parameter estimates in a df
param_df_multi <- tidy(biomarker_model_multi)

#Print the df
kable(param_df_multi)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-49.109259	44.8629858	-1.094650	0.2800593
body_temp	1.307618	0.4600382	2.842411	0.0069477
infection_status	-3.508568	0.9710550	-3.613151	0.0008183

On average, a 1 degree Fahrenheit increase in body temperature increases the biomarker inflammation level by 1.31 units, controlling for infection status.

Compared to a patient with no infection, a patient with an infection has, on average, a lower inflammation level by 3.51 units, controlling for body temperature.

Question 3

a)

A clinical trial is conducted to evaluate the efficacy of a new drug which claims to induce a growth spurt in adolescents. The researchers recruited 50 subjects for the study and they were randomized to either receive the study drug or a sugar pill. Height measurement (in inches) were recorded once a month, beginning at baseline, for 4 consective months. The data is contained in two files: `data/growth_demographics.csv` *which contains the age and sex of the participants and* `data/growth_outcomes.csv` *which contains the treatment group and height measurements for each month*

Read in the datasets and combine them to create a dataframe that contains all the study data.

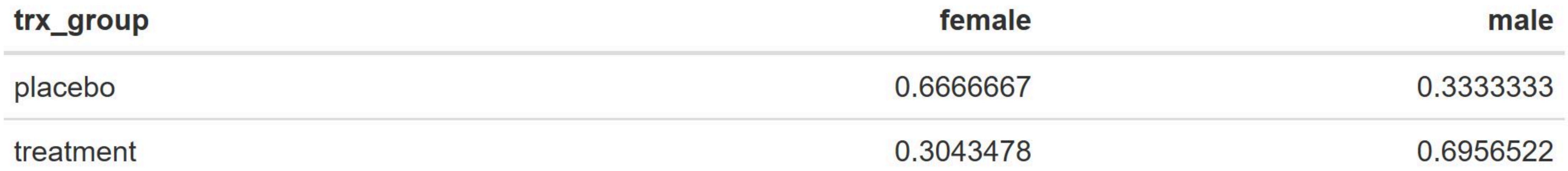
How many variables and observations does the new dataframe have? Please report this in a sentence.

```
#Read in files
growth_dems_df <- read_csv("data/growth_demographics.csv")
growth_outcomes_df <- read_csv("data/growth_outcomes.csv")

#Use full join to combine df's and retain all obs
growth_study_df_wide <- full_join(growth_dems_df,
                                  growth_outcomes_df,
                                  by = "id")
```

The new dataframe has 8 variables and 50 observations.

b)

What are the proportions of males and females within each treatmemt group? Use the kable() function to reproduce the table below (you can see the table without knitting by clicking on the “table_temp.JPG” in the file viewer in the lower right. Hint: calculate the proportions and then use pivot_wider().



trx_group	female	male
placebo	0.6666667	0.3333333
treatment	0.3043478	0.6956522

```
#Group by trx_group and sex, then calculate proportions
growth_gender_grp_long <- growth_study_df_wide %>%
  group_by(trx_group, sex) %>%
  summarize(n = n()) %>%
  mutate(proportion = n/sum(n))

#Convert to wide format
growth_gender_grp_wide <- pivot_wider(growth_gender_grp_long,
                                 names_from = sex,
                                 values_from = proportion)

#Filter out female proporiton in a separate df
growth_gender_fem <- growth_gender_grp_wide %>%
  filter(!is.na(female)) %>%
  select(-n, -male)

#Filter out male proporiton in a separate df
growth_gender_male <- growth_gender_grp_wide %>%
  filter(!is.na(male)) %>%
  select(-n, -female)

#Use inner join to combine newly created gender df's on commong fields
growth_gender_grp_prop <- inner_join(growth_gender_fem,
                                     growth_gender_male,
                                     by = "trx_group")

#Print the newly combined df
kable(growth_gender_grp_prop)
```

trx_group	female	male
placebo	0.6666667	0.3333333
treatment	0.3043478	0.6956522

In the placebo group, the proportion of females is 0.67, and the proportion of males is 0.33. In the treatment group, the proportion of females is 0.3, and the proportion of males is 0.7.

c)

To facilitate visualization, the researchers want the data to be changed from wide to long. Use the appropriate pivot function to create two new variables: “month” and “height” which contains the height measurements for each month. How many variables and observations does the new data frame contain?

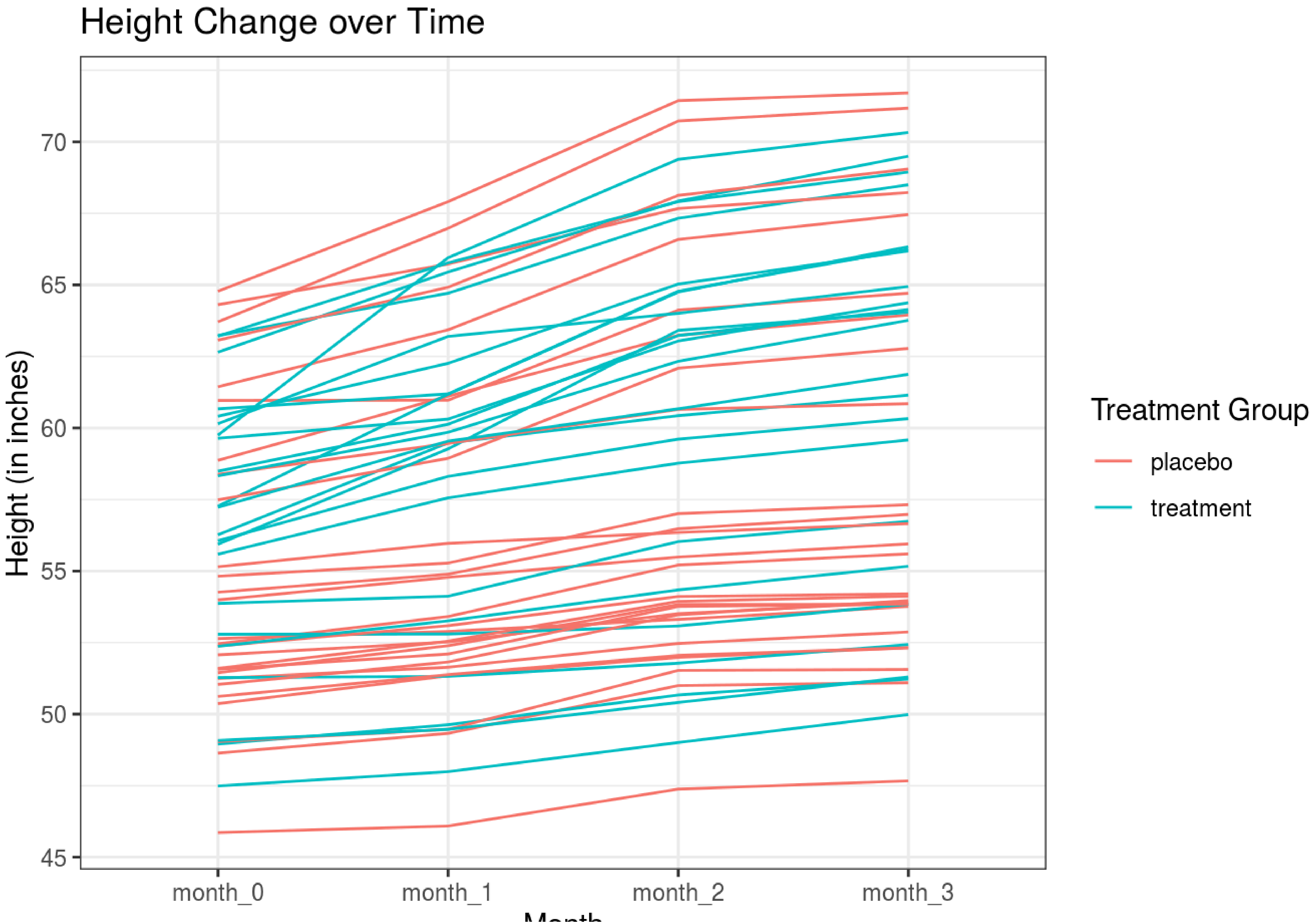
```
growth_study_df_long <- pivot_longer(growth_study_df_wide,
                                     cols = starts_with("month"),
                                     names_to = "month",
                                     values_to = "height")
```

The new dataframe contains 6 variables and 200 observations.

d)

Using the long data, create a spaghetti plot to visualize the longitudinal trends in height across month, and use a color mapping to differentiate between treatment treatment groups. Describe any differences you see between individuals in the treatment groups.

```
ggplot(data = growth_study_df_long) +
  geom_line(aes(x=month, y=height, group=id, color=trx_group)) +
  theme_bw() +
  labs(title="Height Change over Time", x="Month",
       y="Height (in inches)", color="Treatment Group")
```

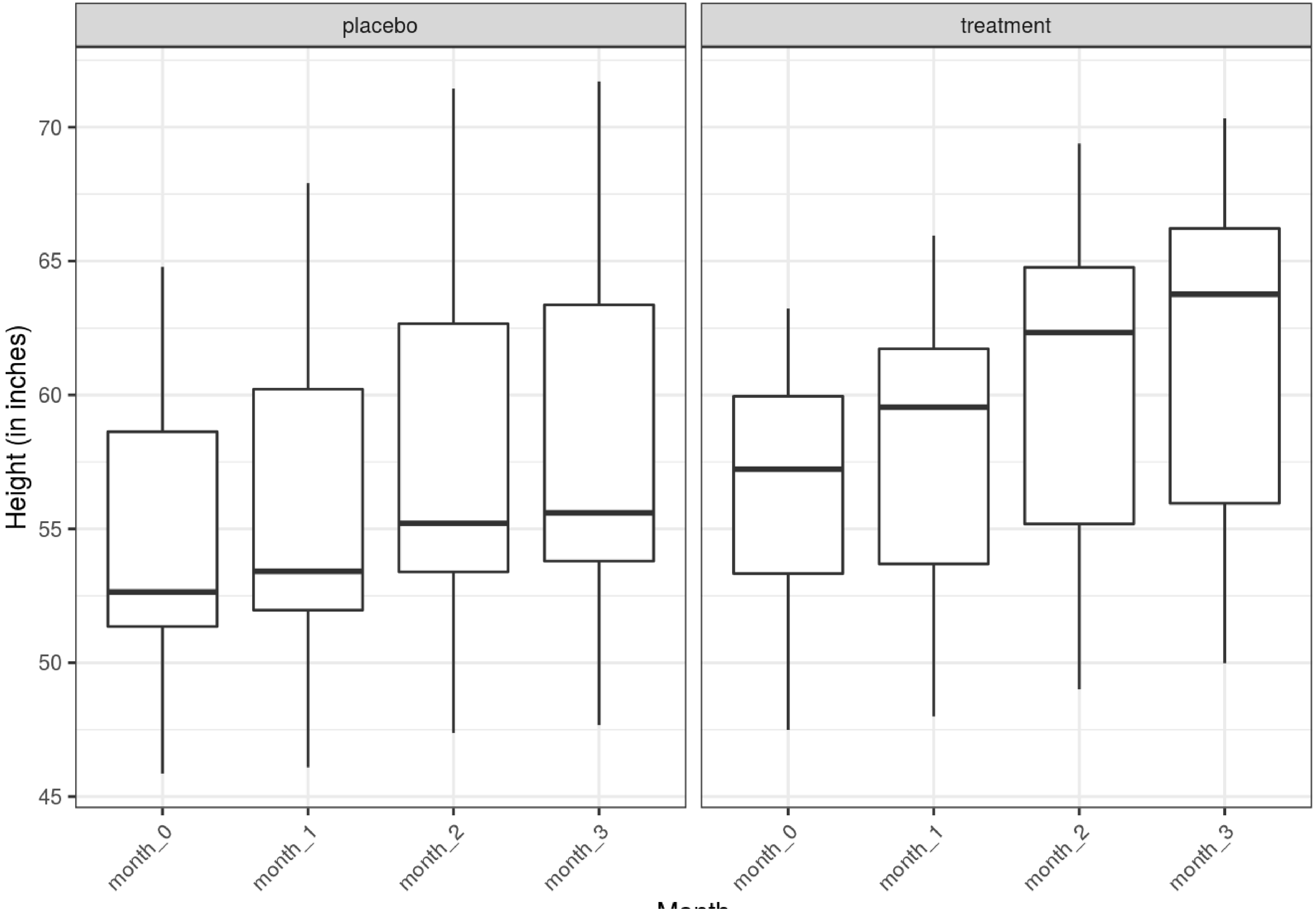

Height Change over Time
Height (in inches)
70
65
60
55
50
45
month_0
month_1
month_2
month_3
Month
Treatment Group
placebo
treatment

It seems the participants in the treatment group, on average, were taller overall than the participants in the placebo group. It also seems participants in the treatment group, on average, experienced larger increases in height over time than participants in the placebo group.

e)

Make two boxplots using facet_wrap() to visualize the distribution of height across month and treatment group.

```
ggplot(data = growth_study_df_long) +
  geom_boxplot(aes(x=month, y=height)) +
  facet_wrap(~trx_group) +
  theme_bw() +
  theme(text = element_text(size=10),
        axis.text.x = element_text(angle=45, hjust=1)) +
  labs("Distribution of Height by Treatment Group",
       x="Month", y="Height (in inches)")
```


placebo
treatment
Height (in inches)
70
65
60
55
50
45
month_0
month_1
month_2
month_3
month_0
month_1
month_2
month_3
Month

Month

f)

The researchers are interested in whether taking the growth drug is associated with larger heights. Create a new variable which represents change in height between baseline (month 0) and month 3 and use linear regression to model the relationship between the change in height and treatment group while accounting for baseline height (month 0). This means your model should have both treatment group an baseline height (month 0) as predictors. Is the overall model significant? Report the r-squared and p-value for the omnibus F test using the glance() function.

```
#Use wide file and create height change variable
growth_study_df_wide <- growth_study_df_wide %>%
  mutate(height_change = month_3 - month_0)

#Run regression model
growth_chg_lm <- lm(height_change ~ trx_group + month_0,
                    data = growth_study_df_wide)

#Put fit statistics in a df
growth_chg_lm_fit <- glance(growth_chg_lm)
```

The model’s R-squared value is 0.5806525, and the model’s omnibus F test p-value is 1.350483510^{-9}. Since $p < 0.05$ for the omnibus F test, the overall model is statistically significant. Treatment group assignment and baseline height explain about 58.07% of the variation in height change.

g)

Report the parameter estimates from your model in (f) in a table using tidy() and kable() functions. Include the 95% confidence interval in your table. Is the treatment group variable significant? Provide an interpretation of the treatment group variable.

```r
#Put parameter estimates in a df
param_df_growth <- tidy(growth_chg_lm)

####Calculate 95% CI for parameter estimates
#Create a function to start and use t=1.960
calculate_95_CI <- function(df) {

  param1 <- df$estimate[1]
  param2 <- df$estimate[2]
  param3 <- df$estimate[3]

  se1 <- df$std.error[1]
  se2 <- df$std.error[2]
  se3 <- df$std.error[3]

  lower_bound1 <- param1 - (1.960*se1)
  upper_bound1 <- param1 + (1.960*se1)

  lower_bound2 <- param2 - (1.960*se2)
  upper_bound2 <- param2 + (1.960*se2)

  lower_bound3 <- param3 - (1.960*se3)
  upper_bound3 <- param3 + (1.960*se3)
  lm_CI_df <- tibble(estimate = c(param1, param2, param3),
                     lower_bound_95_CI = c(lower_bound1,
                                            lower_bound2,
                                            lower_bound3),
                      upper_bound_95_CI = c(upper_bound1,
                                            upper_bound2,
                                            upper_bound3))

  return(lm_CI_df)
}

#Now use the function on the paramater estimate df
growth_lm_CI_df <- calculate_95_CI(param_df_growth)

#Next, merge the CI estimates with the parameter estimate df
param_df_growth <- inner_join(param_df_growth, growth_lm_CI_df,
                             by = "estimate")

#Print the new parameter estimate df
kable(param_df_growth)
```

term	estimate	std.error	statistic	p.value	lower_bound_95_CI	upper_bound_95_CI
(Intercept)	-13.2301816	2.3407465	-5.652121	0.0000009	-17.8180448	-8.6423185

term	estimate	std.error	statistic	p.value	lower_bound_95_CI	upper_bound_95_CI
trx_grouptreatment	1.1228776	0.4202245	2.672090	0.0103272	0.2992376	1.9465176
month_0	0.2988186	0.0423305	7.059186	0.0000000	0.2158509	0.3817863

Compared to the placebo group, the treatment group, on average, grew 1.12 inches more from baseline to the end of the study, controlling for baseline height. Since the p-value of this estimate (0.01) is less than 0.05, the estimate for treatment group assignment is statistically signficant. In other words, this model suggests the drug may increase height.

h)

To control for effects of demographic characteristics on height, add age and sex into the model in part f and g. Describe your findings and give a plausible explanation for the change in your results.

```
#Run regression model
growth_chg_lm_v2 <- lm(height_change ~ trx_group + month_0
                       + age + sex, data = growth_study_df_wide)

#Put fit statistics in a df
growth_chg_lm_fit_v2 <- glance(growth_chg_lm_v2)

#Put parameter estimates in a df
param_df_growth_v2 <- tidy(growth_chg_lm_v2)

#Print the new parameter estimate df
kable(param_df_growth_v2)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-0.7195109	4.1103918	-0.1750468	0.8618281
trx_grouptreatment	0.4253746	0.4304748	0.9881521	0.3283618
month_0	0.0698241	0.0780885	0.8941663	0.3759905
age	-0.0551516	0.0781359	-0.7058423	0.4839245
sexmale	2.8102421	0.8124228	3.4590884	0.0011969

The new model’s R-squared value is 0.6724061, and the new model’s omnibus F test p-value is 2.007548110^{-10}. Since p < 0.05 for the omninbus F test, the new overall model is statistically significant. Treatmemt group assignment, baseline height, age, and sex explain about 67.24% of the variation in height change.

The parameter estimate for treatment group assignment decreased to 0.43. This means compared to the placebo group, the treatment group, on average, grew 0.43 inches more, controlling for baseline height, age, and sex. Also, the paramater estimate for treatment group assignment is no longer significant since it’s p-value (0.33) is greater than 0.05. The explanation for these changes may be that age and sex confounded the association between treatment group assignment and height change.

Sex is the only significant predictor in the model. The estimate indicates that compared to female participants, male participants, on average, grew 2.81 inches more, controlling for treatment group assignment, baseline height, and age.

i)

Finally, use broom::tidy() to create 95% confidence intervals around your parameter estimates from part (h) and output a table using kable()

```
#Create a new function to clauclate 95% CI and use t=1.960
calculate_95_CI_v2 <- function(df) {

  param1 <- df$estimate[1]
  param2 <- df$estimate[2]
  param3 <- df$estimate[3]
  param4 <- df$estimate[4]
  param5 <- df$estimate[5]

  se1 <- df$std.error[1]
  se2 <- df$std.error[2]
  se3 <- df$std.error[3]
  se4 <- df$std.error[4]
  se5 <- df$std.error[5]

  lower_bound1 <- param1 - (1.960*se1)
  upper_bound1 <- param1 + (1.960*se1)

  lower_bound2 <- param2 - (1.960*se2)
  upper_bound2 <- param2 + (1.960*se2)

  lower_bound3 <- param3 - (1.960*se3)
  upper_bound3 <- param3 + (1.960*se3)
  lower_bound4 <- param4 - (1.960*se4)
  upper_bound4 <- param4 + (1.960*se4)

  lower_bound5 <- param5 - (1.960*se5)
  upper_bound5 <- param5 + (1.960*se5)

  lm_CI_df <- tibble(estimate = c(param1, param2, param3, param4,
                                  param5),
                     lower_bound_95_CI = c(lower_bound1,
                                            lower_bound2,
                                            lower_bound3,
                                            lower_bound4,
                                           lower_bound5),
                      upper_bound_95_CI = c(upper_bound1,
                                            upper_bound2,
                                            upper_bound3,
                                            upper_bound4,
                                            upper_bound5))

  return(lm_CI_df)
}

#Use the new function on the new paramater estimate df
growth_lm_CI_df_v2 <- calculate_95_CI_v2(param_df_growth_v2)
```

```
#Next, merge the CI estimates with the new parameter estimate df
param_df_growth_v2 <- inner_join(param_df_growth_v2,
                                 growth_lm_CI_df_v2, by = "estimate")

#Print table of new parameter estimates
kable(param_df_growth_v2)
```

term	estimate	std.error	statistic	p.value	lower_bound_95_CI	upper_bound_95_CI
(Intercept)	-0.7195109	4.1103918	-0.1750468	0.8618281	-8.7758787	7.3368570
trx_grouptreatment	0.4253746	0.4304748	0.9881521	0.3283618	-0.4183560	1.2691052
month_0	0.0698241	0.0780885	0.8941663	0.3759905	-0.0832293	0.2228775
age	-0.0551516	0.0781359	-0.7058423	0.4839245	-0.2082981	0.0979948
sexmale	2.8102421	0.8124228	3.4590884	0.0011969	1.2178935	4.4025907

```
####Note: tidy() was used in part h)
```

j)

Given your findings, do you think there is enough evidence to support the use of this drug? Explain in a sentence or two.

Since only the parameter estimate for sex is significant in the full model, sex may have been the most robust predictor of height change in the experiment. Since the estimate for treatment group is not significant in the full model, there isn’t enough evidence to support the use of the drug.