Announcement

P8101S20-Homework_07

Neal Kar (ink2105)

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

Researchers are interested in the relationship of demographic factors and alcohol consumption on cirrhosis death rates. You have been given a dataset containing population, drinking data, and cirrhosis death rates for 46 US states in the `data/wine_data.txt` *file. The variables of interest are:*

- `urban_pop` *the percentage of urban population in the state*

- *`late_births` a measure of the number of births to women between 45 to 49*

- *`wine_consumption` the consumption of wine per capita*

- *`liquor_consumption` the consumption of hard liquor per capita*

- *`cirrhosis_death` the death rate from cirrhosis (outcome)*

a)

Read the data into your environment using the `read_table2()` function. Take a look at the original data file and describe it's format. Explain why we needed to use the `read_table2()` function – you can take a look at the help file for `read_table2()` for help.

```
help("read_table2")

wine_df <- read_table2("data/wine_data.txt")
```

The format of the original file is .txt, indicating the data is arranged in columns separated by whitespace. We use the "read_table2" command because that enables us to read a .txt file with any amount of whitespace between columns.

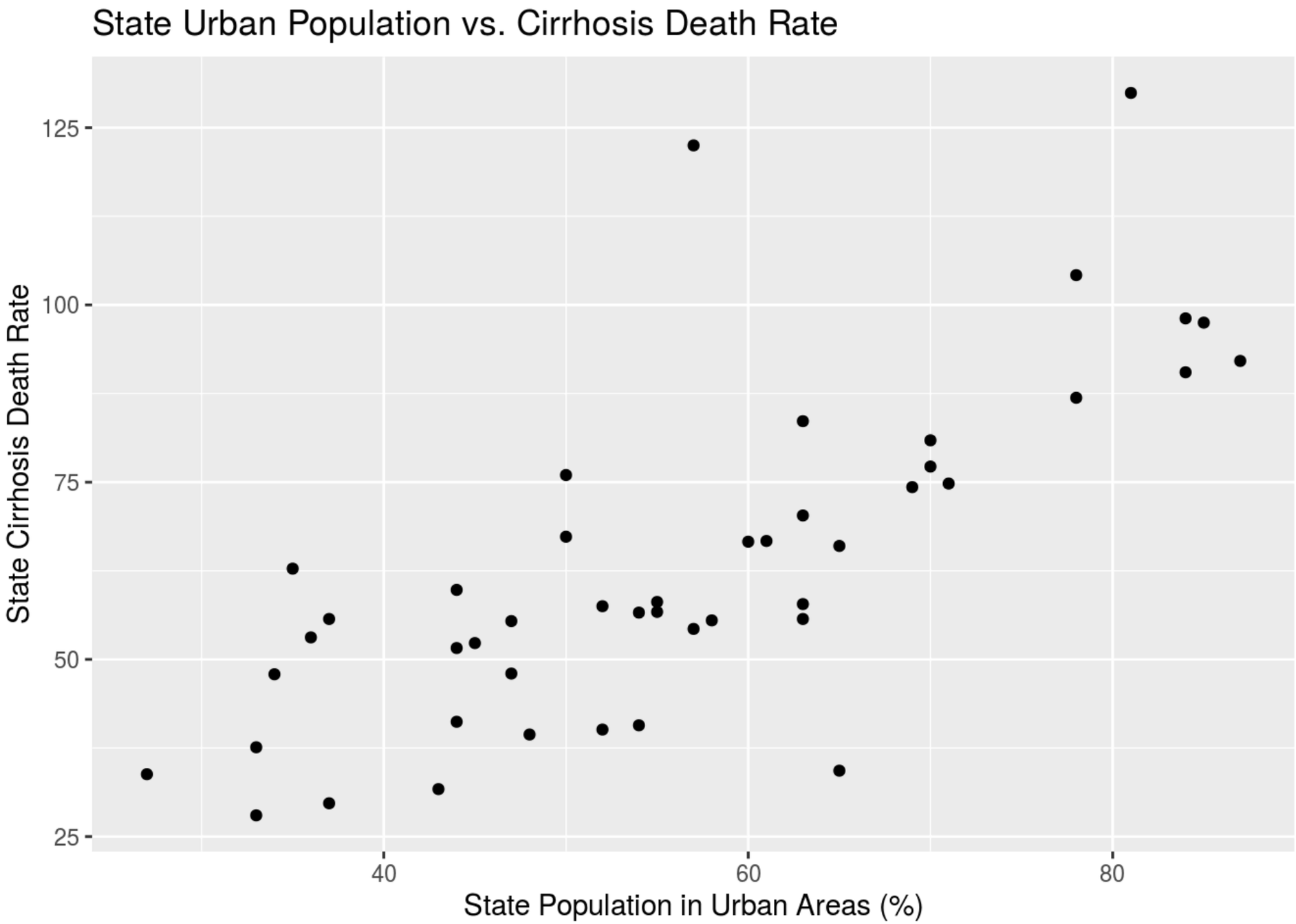
b)

Create scatterplots that display the relationship between each continuous variable (`urban_pop` , `late_births` , `wine_consumption` , and `liquor_consumption`) with the dependent variable of interest `cirrhosis_death` .

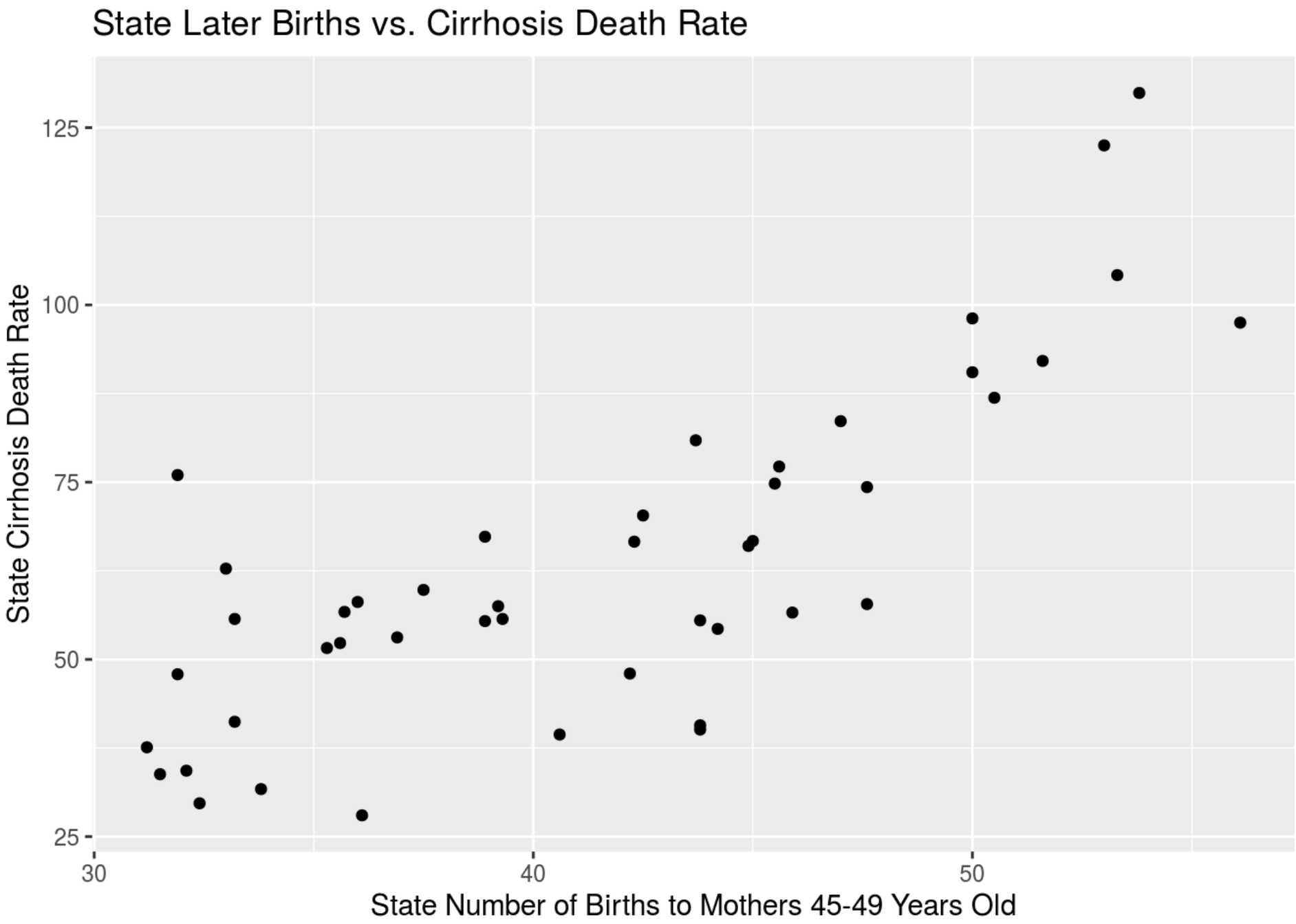
Make sure to add a proper title to each graph, and descriptive titles to the x and y axes (not just the variable names).

Based on these scatterplots, which continuous independent variables have a linear relationship with `cirrhosis_death` ? Please answer in a sentence.

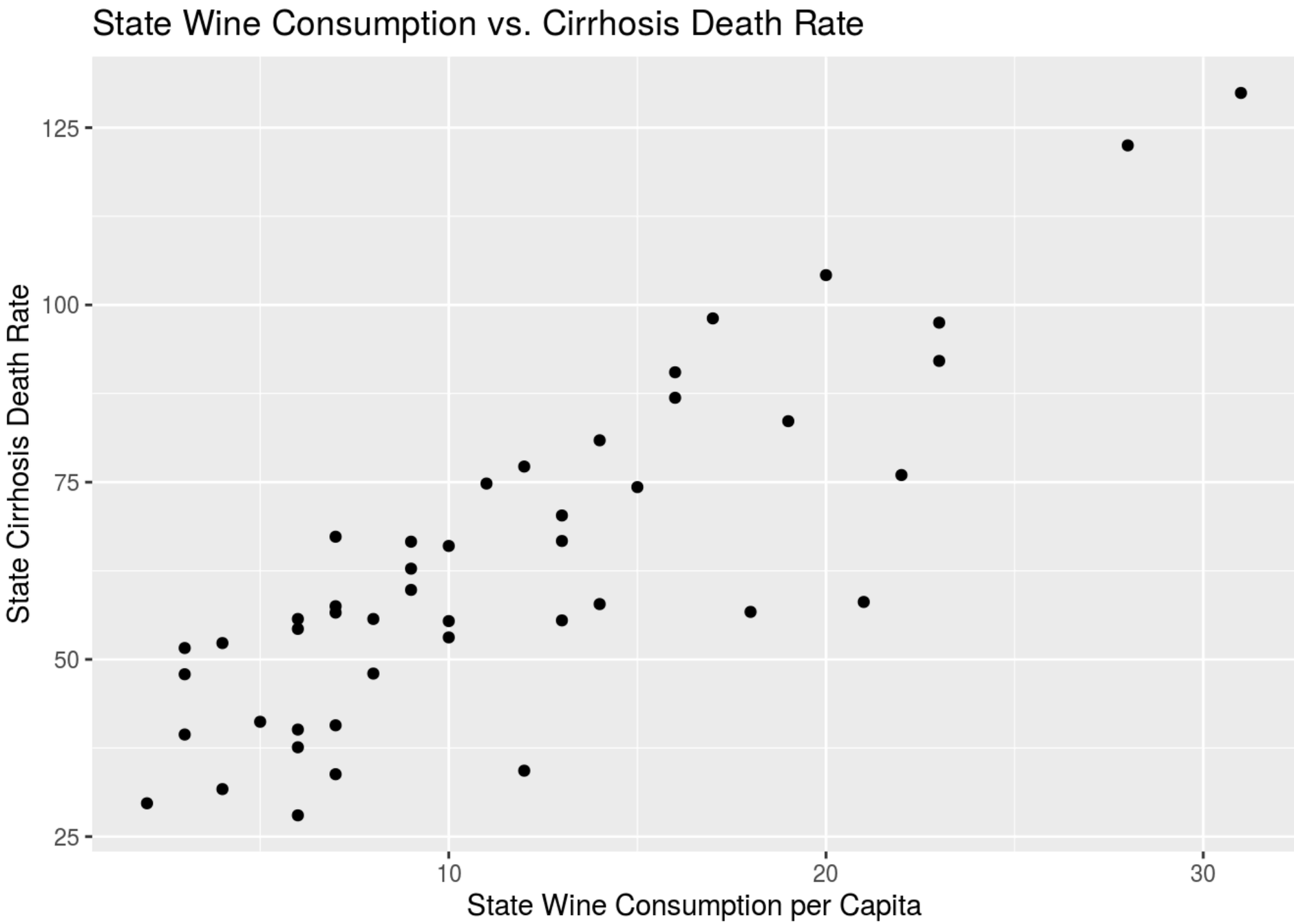
```
#Urban population
ggplot(data = wine_df) +
  geom_point(aes(x=urban_pop, y=cirrhosis_death)) +
  labs(title="State Urban Population vs. Cirrhosis Death Rate",
       x="State Population in Urban Areas (%)",
       y="State Cirrhosis Death Rate")
```


State Urban Population vs. Cirrhosis Death Rate
State Cirrhosis Death Rate
125
100
75
50
25
40
60
80
State Population in Urban Areas (%)

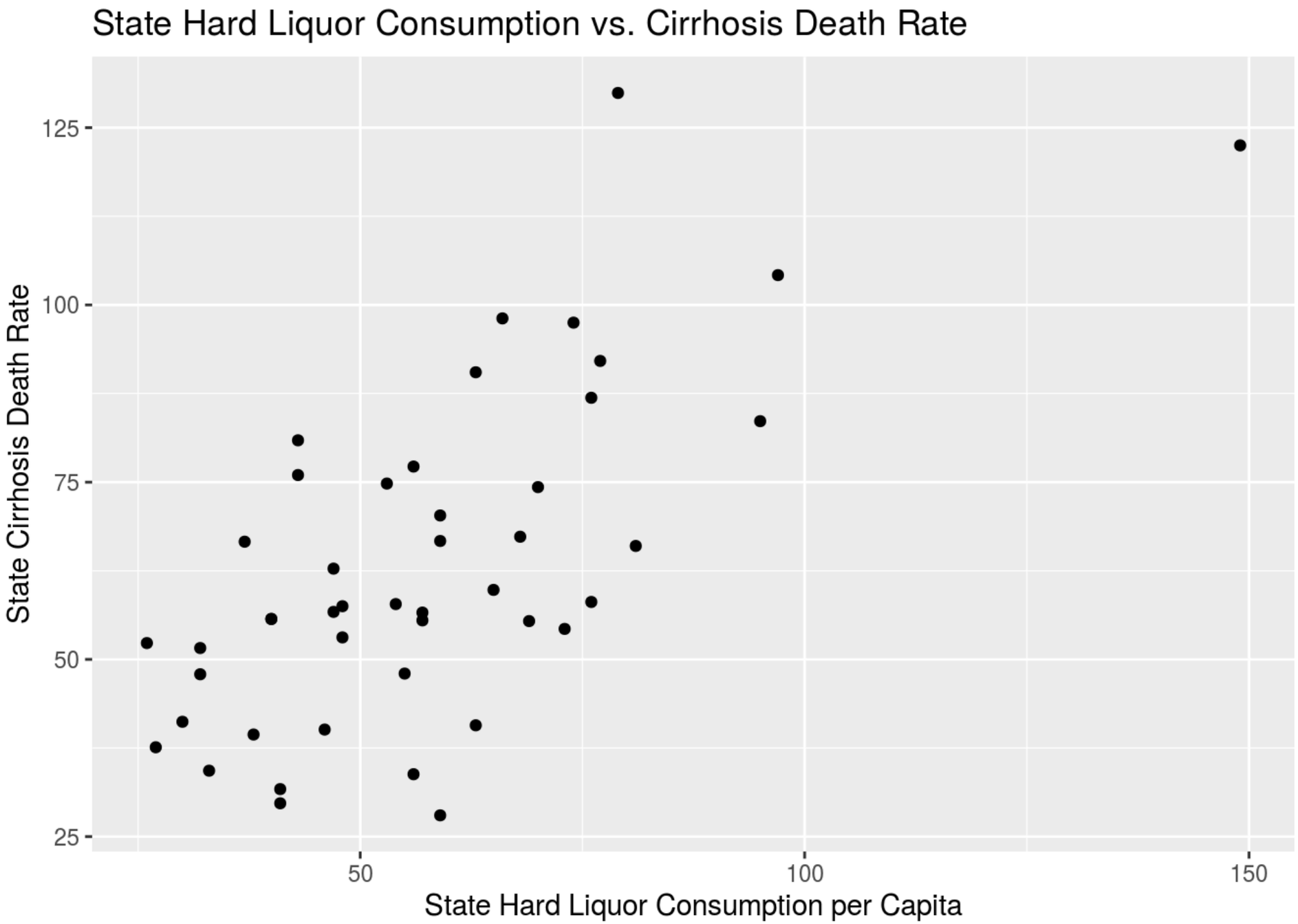
```
#Late births
ggplot(data = wine_df) +
  geom_point(aes(x=late_births, y=cirrhosis_death)) +
  labs(title="State Later Births vs. Cirrhosis Death Rate",
       x="State Number of Births to Mothers 45-49 Years Old",
       y="State Cirrhosis Death Rate")
```


State Later Births vs. Cirrhosis Death Rate
State Cirrhosis Death Rate
125
100
75
50
25
30
40
50
State Number of Births to Mothers 45-49 Years Old

```
#Wine consumption
ggplot(data = wine_df) +
  geom_point(aes(x=wine_consumption, y=cirrhosis_death)) +
  labs(title="State Wine Consumption vs. Cirrhosis Death Rate",
       x="State Wine Consumption per Capita",
       y="State Cirrhosis Death Rate")
```


State Wine Consumption vs. Cirrhosis Death Rate
State Cirrhosis Death Rate
125
100
75
50
25
10
20
30
State Wine Consumption per Capita

```
#Liquor consumption
ggplot(data = wine_df) +
  geom_point(aes(x=liquor_consumption, y=cirrhosis_death)) +
  labs(title="State Hard Liquor Consumption vs. Cirrhosis Death Rate",
       x="State Hard Liquor Consumption per Capita",
       y="State Cirrhosis Death Rate")
```


State Hard Liquor Consumption vs. Cirrhosis Death Rate
State Cirrhosis Death Rate
125
100
75
50
25
50
100
150
State Hard Liquor Consumption per Capita

Based on the scatterplots, it seems all four variables have a linear relationship with cirrhosis death rate.

c)

Fit a series of simple regressions with `cirrhosis_death` as an outcome and each continuous variable as a sole predictor. Report each model's parameter estimates using tidy() and kable(). You do not need to provide interpretations for these estimates.

Which continuous variables were significantly associated with `cirrhosis_death` ?

```
#Urban population
urb_pop_lm = lm(cirrhosis_death ~ urban_pop, data=wine_df)
urb_pop_lm_est <- tidy(urb_pop_lm)
print("Urban Population Bivariate Paramater Estimates:")
```

```
## [1] "Urban Population Bivariate Paramater Estimates:"
```

```
kable(urb_pop_lm_est)
```

term	estimate	std.error	statistic	p.value
(Intercept)	0.7407338	8.6814780	0.0853235	0.9323913

term	estimate	std.error	statistic	p.value
urban_pop	1.1153888	0.1487141	7.5002216	0.0000000

```
#Later births
late_birth_lm = lm(cirrhosis_death ~ late_births, data=wine_df)
late_birth_lm_est <- tidy(late_birth_lm)
print("Later Births Bivariate Paramater Estimates:")
```

```
## [1] "Later Births Bivariate Paramater Estimates:"
```

```
kable(late_birth_lm_est)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-44.568187	13.1349020	-3.393111	0.0014717
late_births	2.605397	0.3123103	8.342335	0.0000000

```
#Wine consumption
wine_lm = lm(cirrhosis_death ~ wine_consumption, data=wine_df)
wine_lm_est <- tidy(wine_lm)
print("Wine Consumption Bivariate Paramater Estimates:")
```

```
## [1] "Wine Consumption Bivariate Paramater Estimates:"
```

```
kable(wine_lm_est)
```

term	estimate	std.error	statistic	p.value
(Intercept)	30.334670	3.6802585	8.242538	0
wine_consumption	2.861736	0.2734694	10.464557	0

```
#Hard liquor consumption
liquor_lm = lm(cirrhosis_death ~ liquor_consumption, data=wine_df)
liquor_lm_est <- tidy(liquor_lm)
print("Hard Liquor Consumption Bivariate Paramater Estimates:")
```

```
## [1] "Hard Liquor Consumption Bivariate Paramater Estimates:"
```

```
kable(liquor_lm_est)
```

term	estimate	std.error	statistic	p.value
(Intercept)	21.9649475	7.1847242	3.057173	0.0037919
liquor_consumption	0.7222353	0.1167699	6.185114	0.0000002

d)

Fit a linear regression model that includes all of the continuous variables from parts (b) and (c). Report the model's parameter estimates using tidy() and kable().

Are any continuous variables significantly associated with `cirrhosis_death` *in this fully adjusted model?*

If there are any continuous variables significantly associated with `cirrhosis_death` *, give an interpretation of the significant parameter estimate using the words of the problem.*

```
#Fully adjusted model
full_cirrh_lm <- lm(cirrhosis_death ~ urban_pop + late_births
                    + wine_consumption + liquor_consumption,
                    data = wine_df)

full_cirrh_lm_est <- tidy(full_cirrh_lm)

print("Paramater Estimates of Fully Adjusted Model:")
```

```
## [1] "Paramater Estimates of Fully Adjusted Model:"
```

```
kable(full_cirrh_lm_est)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-13.9631001	11.4003481	-1.2247959	0.2276426
urban_pop	0.0982859	0.2440656	0.4027028	0.6892590
late_births	1.1483771	0.5829952	1.9697881	0.0556435
wine_consumption	1.8578610	0.4009582	4.6335532	0.0000361
liquor_consumption	0.0481702	0.1333558	0.3612156	0.7197931

Only wine consumption is significantly associated with cirrhosis death rate in the fully adjusted model. The model indicates that for every one unit increase in wine consumption per capita in a state, that state's cirrhosis death rate is predicted to increase by 1.86 units, controlling for urban population, late births, and hard liquor consumption.

Question 2

You have been given a small data file of the average weight (in pounds) for two strains of guinea pig from a series of years in the early 20th century. This data is stored in `data/gpig_dat.csv` *and has a variable for year and a variable for each strain. The value in each cell of the strain variable is the weight for that strain in that year.*

a)

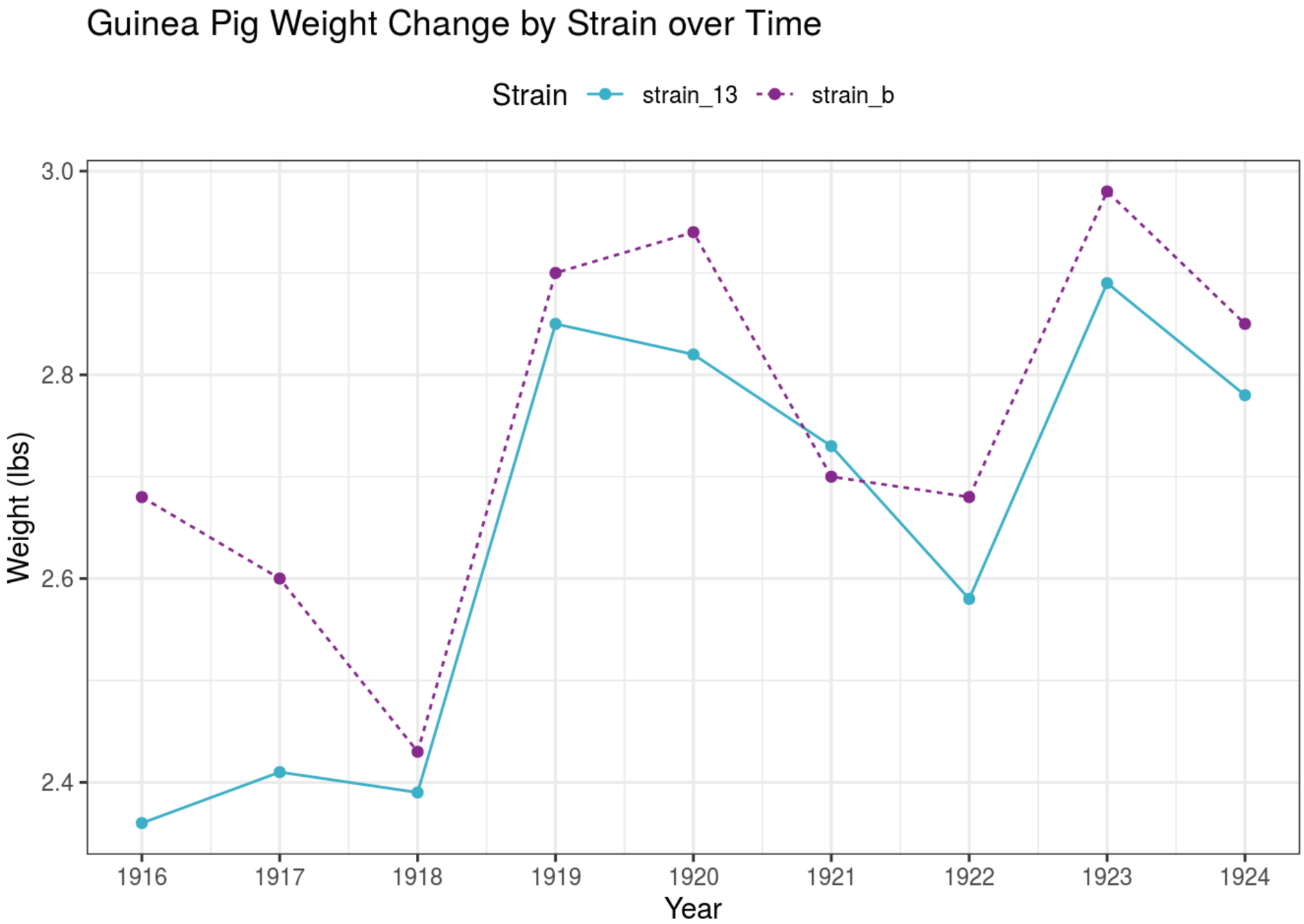
Graph guinea pig weights by year in a line plot (using both line and point geoms).

For your graph: (1) Each strain should be a different color and a different line type. For the color, don't use default R choices, but choose your own custom colors. (2) The background of your plot should be white, not the default gray. (3) Your graph should have a title, and the x and y axes should also have titles (not just the variable names). (4) The legend should have a title and should be at the top of your graph.

Hint: you may have to transform the data to plot this correctly.

```
#Create wide format df
gpig_df_wide <- read_csv("data/gpig_dat.csv")

#Clean variable names
gpig_df_wide <- clean_names(gpig_df_wide)

#Convert into long format
gpig_df_long <- pivot_longer(data = gpig_df_wide,
                             cols = starts_with("strain"),
                             names_to = "strain",
                             values_to = "weight")

#Make lineplot
ggplot(data = gpig_df_long) +
  geom_point(aes(x=year, y=weight, color=strain)) +
  geom_line(aes(x=year, y=weight, color=strain,
                linetype=strain)) +
  scale_color_manual(values = c("strain_13" = "#3CB3C8",
                                "strain_b" = "#892890")) +
  scale_x_continuous(breaks = seq(1916, 1924, by=1)) +
  theme_bw() +
  theme(legend.position = "top") +
    labs(title="Guinea Pig Weight Change by Strain over Time",
```

```
        x="Year", y="Weight (lbs)", color="Strain",
        linetype="Strain")
```


Guinea Pig Weight Change by Strain over Time
Strain
strain_13
strain_b
3.0
2.8
2.6
2.4
Weight (lbs)
1916
1917
1918
1919
1920
1921
1922
1923
1924
Year

b)

Use a two-sample t-test to test the hypothesis that mean guinea pig weights are equal, at a two-sided level of significance of 5%. Use `tidy()` and `kable()` functions to provide the summary results of your t-test and also report the test statistic, degrees of freedom, and p-value in a sentence that explains whether you reject or do not reject the null hypothesis.

```
#Conduct t-test
ttest_gpig <- t.test(weight ~ strain, data = gpig_df_long)

#Store results in a tibble
ttest_gpig_tib <- tidy(ttest_gpig)

#Display table of results
kable(ttest_gpig_tib)
```

estimate	estimate1	estimate2	statistic	p.value	parameter	conf.low	conf.high	method	alternative
-0.1055556	2.645556	2.751111	-1.135204	0.2734787	15.5563	-0.3031303	0.0920192	Welch Two Sample t-test	two.sided

Test statistic: -1.14 Degrees of freedom: 15.56 p-value: 0.27

Since the p-value of the t-test is greater than 0.05, we do not reject the null hypothesis of mean weights being equal over time between the two guinea pig strains (i.e. their mean weights did not significantly differ over time).

c)

Use a permutation test to test whether the minimum weight of guinea pigs are equal between strains, at a two-sided level of significance of 5%. Your permutation test should simulate at least 2000 test statistics.

In your solution, be sure to (1) provide a histogram of simulated test statistics with your observed statistic clearly marked, and (2) include a final interpretation sentence that reports your test p-value and decision to reject or not reject the null hypothesis.

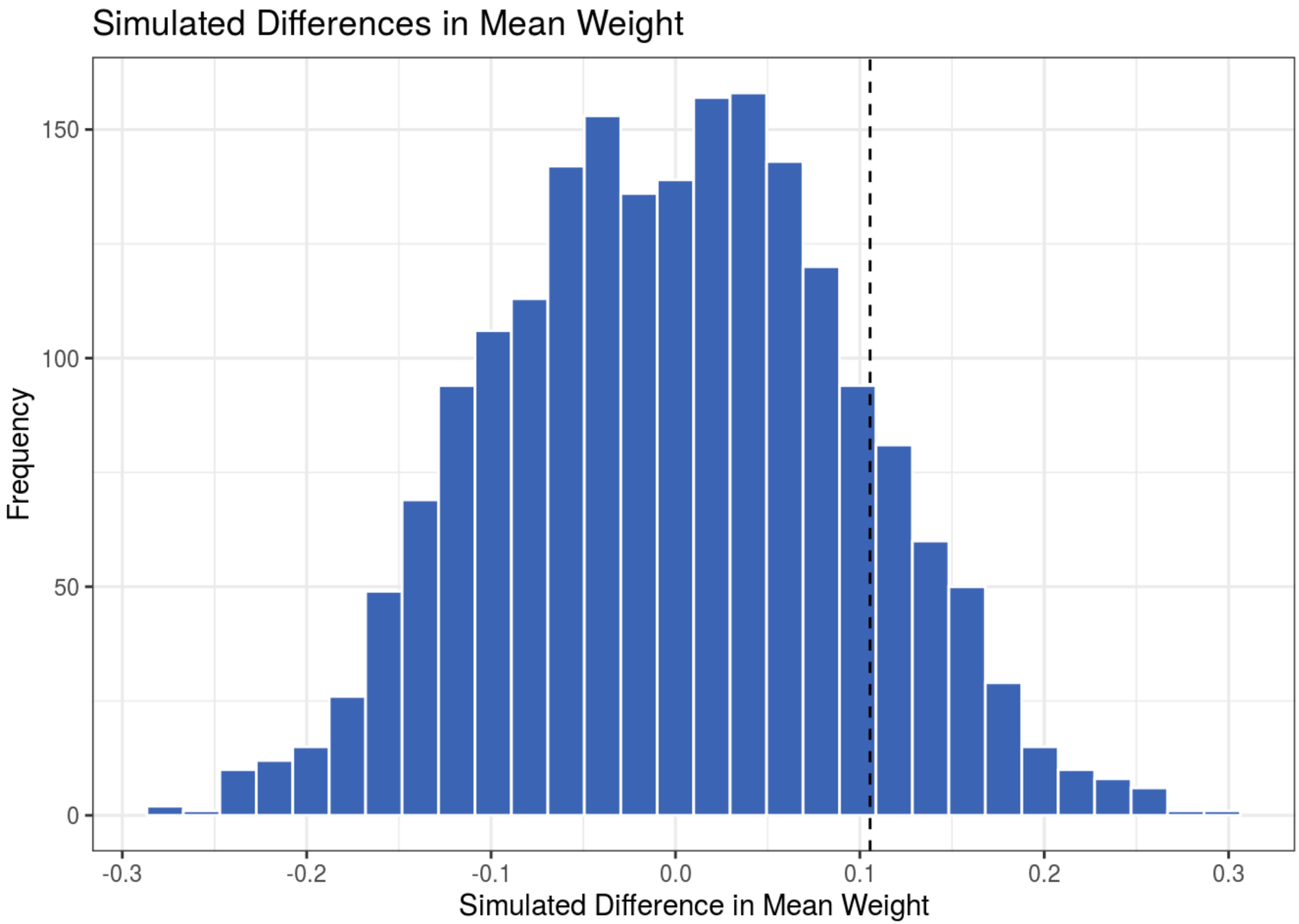
```r
# Create a function to calculate the test statistic

calculate_ts <- function(df){
  summary <- df %>%
    group_by(strain) %>%
    summarize(mean_weight = mean(weight))

  strain_b <- summary %>%
    filter(strain == "strain_b") %>%
    pull(mean_weight)

  strain_13 <- summary %>%
    filter(strain == "strain_13") %>%
    pull(mean_weight)

  difference <- strain_b - strain_13
  return(difference)
}

obs_stat <- calculate_ts(gpig_df_long)
obs_stat
```

```
## [1] 0.1055556
```

```r
# Create a function that performs a single permutation.

perform_permutation <- function(df){
  permuted <- df %>%
    mutate(weight = sample(weight))
  return(permuted)
}

permuted <- perform_permutation(gpig_df_long)



# Combine the permutation function and the calculation of the test statistic
do_permutation_test <- function(df){
  permed <- perform_permutation(df)
  ts <- calculate_ts(permed)
  return(ts)
}

do_permutation_test(gpig_df_long)
```

```
## [1] 0.09222222
```

```r
# Now we want to do this 2000 times

results <- map_dbl(1:2000, function(x) do_permutation_test(gpig_df_long))

# Put into a tibble for graphing:
res_tibble <- tibble(sim_stat = results)



# Graph all of these permuted test statistics to the observed difference in means

ggplot(data = res_tibble) +
  geom_histogram(aes(x = sim_stat), bins = 30,
                 fill = "#4068B7", color = "white") +
  geom_vline(aes(xintercept = obs_stat), linetype = "dashed") +
  theme_bw() +
  labs(title="Simulated Differences in Mean Weight",
       x="Simulated Difference in Mean Weight", y="Frequency")
```


Simulated Differences in Mean Weight
Frequency
150
100
50
0
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
Simulated Difference in Mean Weight

```
# Compare simulated test statistics to observed test statistic to calculate p-value. T
he proportion of simulated test statistics whose absolute value is greater than or equ
al to the observed test statistic will be our p-value:

greater_than <- res_tibble %>%
  mutate(abs_val_greater = if_else(abs(sim_stat) >=
                                     abs(obs_stat), 1, 0)) %>%
  filter(abs_val_greater == 1) %>%
  summarize(n = n()) %>%
  pull(n)

permute_pval <- greater_than/2000

permute_pval
```

```
## [1] 0.2895
```

The p-value of the simulation is 0.29, and since it's greater than 0.05, we do not reject the null hypothesis of mean weights being equal over time between the two guinea pig strains (i.e. their mean weights did not significantly differ over time).

Question 3

You have been given six datasets with repeated measures of MIRECC GAF Social Functioning scores for an observational study of individuals with schizophrenia. Each dataset includes a GAF SF measurement at a different follow-up visit. These files are stored in the `data/gaf_files` *folder.*

a)

Combine all six datasets using a method that will remove any individuals who are not present in ALL datasets.

```
#Read in all 6 files
visit1_df <- read_csv("data/gaf_files/visit_1.csv")
visit2_df <- read_csv("data/gaf_files/visit_2.csv")
visit3_df <- read_csv("data/gaf_files/visit_3.csv")
visit4_df <- read_csv("data/gaf_files/visit_4.csv")
visit5_df <- read_csv("data/gaf_files/visit_5.csv")
visit6_df <- read_csv("data/gaf_files/visit_6.csv")

#Merge visit1 and visit2 df's
first_merge <- inner_join(visit1_df, visit2_df,
                          by = c("patient_id" = "id"))

#Merge visit3
second_merge <- inner_join(first_merge, visit3_df,
                           by = c("patient_id" = "study_id"))

#Merge visit4
third_merge <- inner_join(second_merge, visit4_df,
                          by = c("patient_id" = "study_id"))

#Merge visit5
fourth_merge <- inner_join(third_merge, visit5_df,
                           by = "patient_id")

#Merge visit6 to create wide format df
```

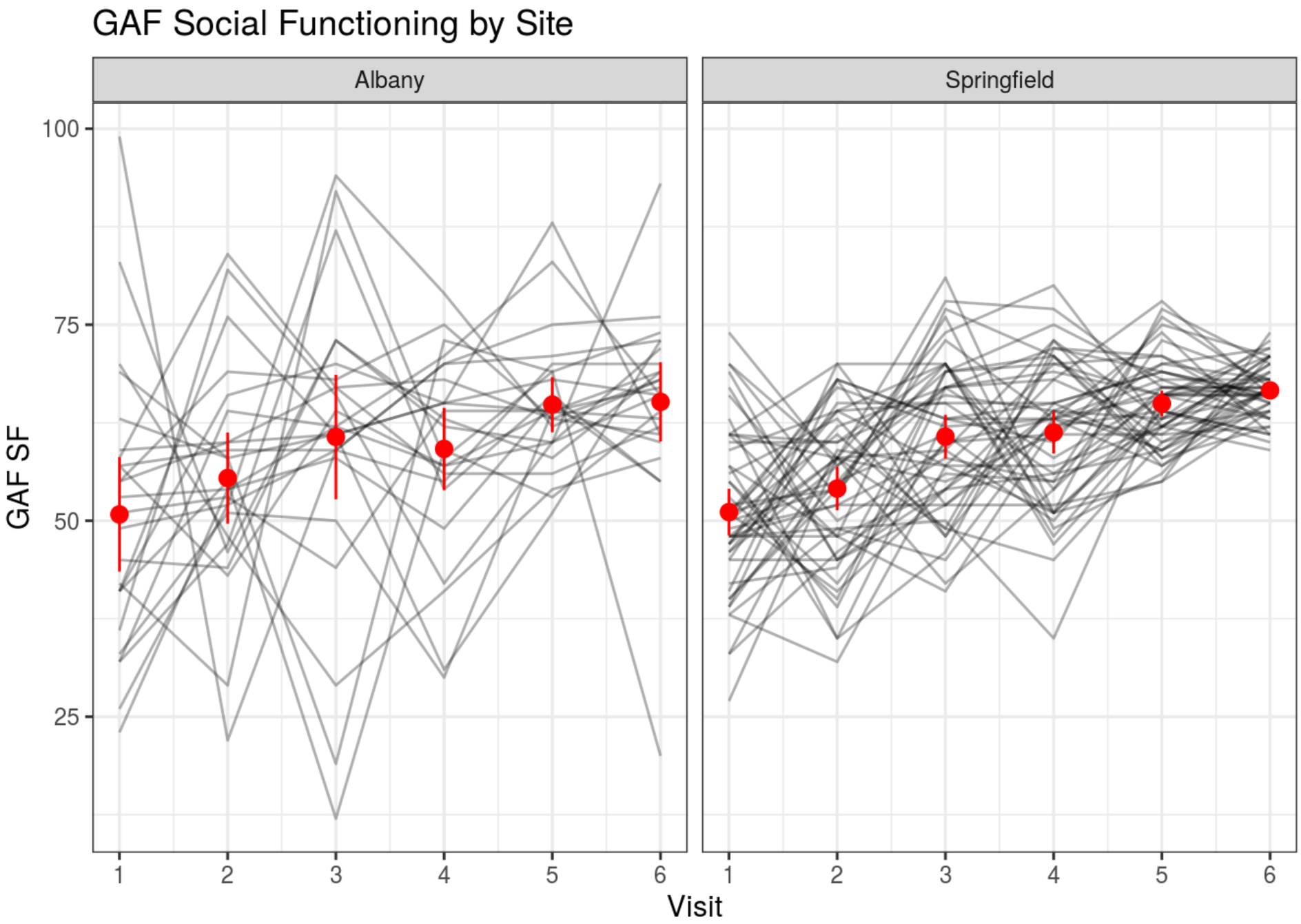
```
soc_func_df_wide <- inner_join(fourth_merge, visit6_df,
                               by = c("patient_id" = "id"))
```

b)

Reproduce the graph shown in the `gaf_plot.png` file. You can see this file by clicking on it in your file explorer. Notice that the graph is faceted by site and has the mean values and their confidence interval overlaid on each plot in red. Hint: if you are having trouble remembering how to overlay a second dataset onto a graph, check out Lab 04.

```
#Convert to long format
soc_func_df_long <- pivot_longer(data = soc_func_df_wide,
                                 cols = starts_with("visit"),
                                 names_to = "visit",
                                 names_prefix = "visit_",
                                 names_ptypes =
                                   list(visit=integer()),
                                 values_to = "gaf_score")


#Create separate df with mean GAF scores by site and visit
site_means <- soc_func_df_long %>%
  group_by(site, visit) %>%
  summarize(mean_gaf = mean(gaf_score), sd_gaf = sd(gaf_score),
            count = n()) %>%
  mutate(se_gaf = sd_gaf / sqrt(count),
    lower_95_ci = round(mean_gaf - qt(1-(0.05/2),
                                      count-1)*se_gaf, 2),
    upper_95_ci = round(mean_gaf + qt(1-(0.05/2),
                                      count-1)*se_gaf, 2))


#Create plot
ggplot(data = soc_func_df_long) +
  geom_line(aes(x=visit, y=gaf_score, group=patient_id),
```

```
          alpha=0.3) +
geom_pointrange(data = site_means,
                aes(x=visit, y=mean_gaf,
                    ymin = lower_95_ci, ymax = upper_95_ci),
                color="red") +
facet_wrap(~site) +
theme_bw() +
theme(legend.position = "none") +
scale_x_continuous(breaks=1:6, labels=1:6) +
labs(title="GAF Social Functioning by Site", x="Visit",
     y="GAF SF")
```


GAF Social Functioning by Site
Albany
Springfield
GAF SF
100
75
50
25
1
2
3
4
5
6
Visit

Question 4

Investigators are interested in determining a sample size for a study on the effect of a mobile health app on mood score. They expect that the mean mood score before treatment will be 8 and the mean score after treatment will be 11, with a known population standard deviation of 4.

a)

The investigators are interested how the power to detect a difference between these means will change based on sample size. Calculate different power values for a two-sided test at a level of signifiance of 5% for sample sizes from 5 to 20. Hint: you can use the power calculation function you corrected in Homework 05 Question 2 to do this.

```r
#Create function that calculates power
calculate_power <- function(alt_mu, null_mu, s, n){
  alt_z <-(alt_mu - null_mu)/(s/sqrt(n))
  alt_hypothesis <- rnorm(10000, alt_z, 1)
  alt_tibble <- tibble(obs = alt_hypothesis, scenario = "Alternative Hypothesis") %>%
    mutate(region = if_else(obs < -1.96 | obs > 1.96, "rejection", "non-rejection"))

  power_df <- alt_tibble %>%
    group_by(region) %>%
    summarize(n = n()) %>%
    mutate(proportion = n/sum(n))

  power <- power_df %>%
    filter(region == "rejection") %>%
    pull(proportion) %>%
    round(4)
return(power)
}

#Run power calculations for each sample size
n_5 <- calculate_power(11,8,4,5)
n_6 <- calculate_power(11,8,4,6)
n_7 <- calculate_power(11,8,4,7)
n_8 <- calculate_power(11,8,4,8)
n_9 <- calculate_power(11,8,4,9)
```

```
n_10 <- calculate_power(11,8,4,10)
n_11 <- calculate_power(11,8,4,11)
n_12 <- calculate_power(11,8,4,12)
n_13 <- calculate_power(11,8,4,13)
n_14 <- calculate_power(11,8,4,14)
n_15 <- calculate_power(11,8,4,15)
n_16 <- calculate_power(11,8,4,16)
n_17 <- calculate_power(11,8,4,17)
n_18 <- calculate_power(11,8,4,18)
n_19 <- calculate_power(11,8,4,19)
n_20 <- calculate_power(11,8,4,20)

#Create a tibble of sample sizes and power
power_calcs_tib <- tibble(sample_size = c(5:20),
                          power = c(n_5, n_6, n_7, n_8, n_9,
                                    n_10, n_11, n_12, n_13, n_14,
                                    n_15, n_16, n_17, n_18, n_19,
                                    n_20))

#Print out the tibble to show statistical power at each sample size
print("Power at each sample size:")
```

```
## [1] "Power at each sample size:"
```

```
kable(power_calcs_tib)
```

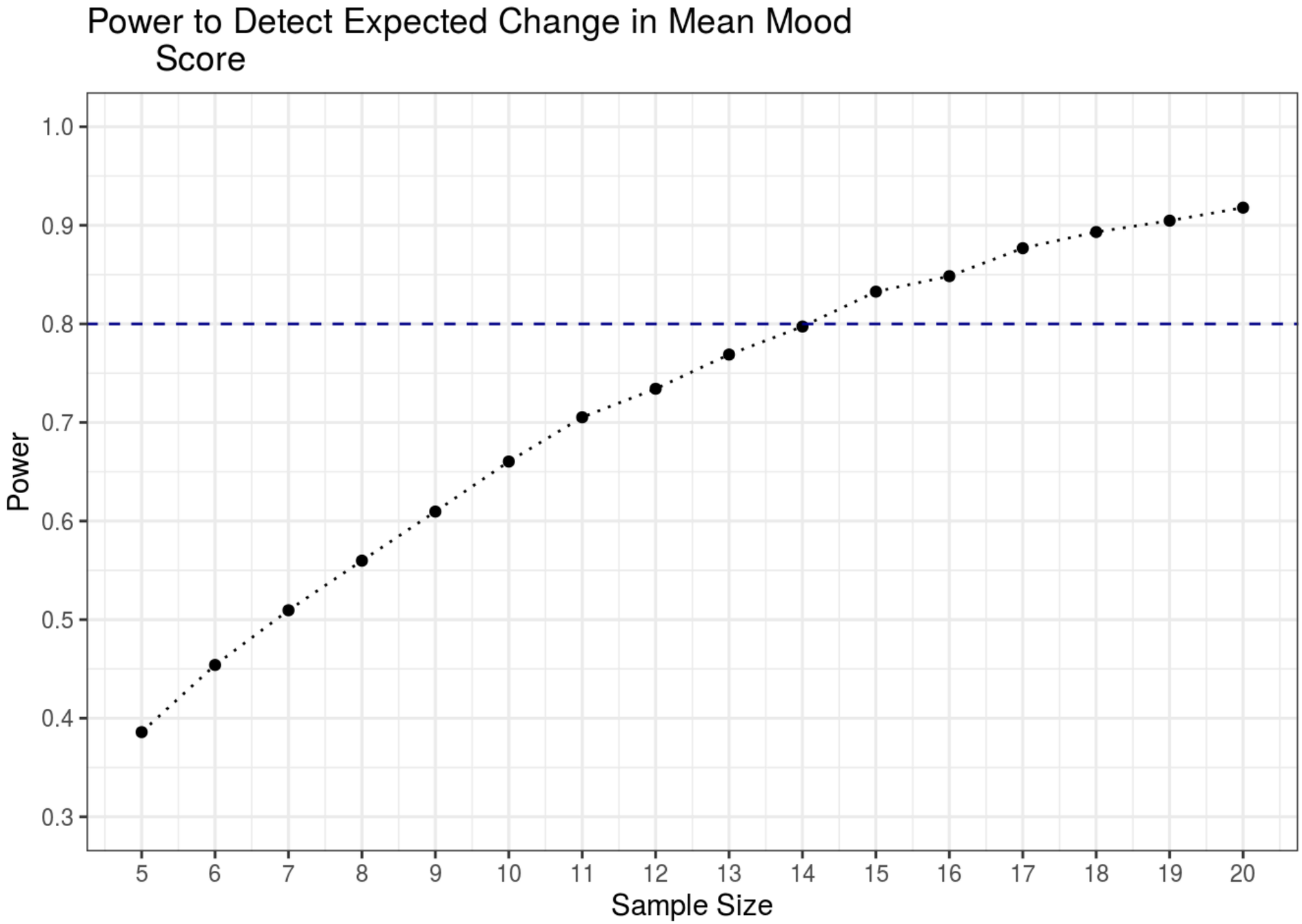
sample_size	power
5	0.3859
6	0.4540
7	0.5095
8	0.5598
9	0.6096
10	0.6604
11	0.7053
12	0.7342
13	0.7689
14	0.7973
15	0.8327
16	0.8484

	sample_size	power
	17	0.8768
	18	0.8932
	19	0.9047
	20	0.9179

b)

Once you have calculated the power across these sample sizes, make a graph of it and include a dashed line at 80% power. Make sure to include a title and appropriate x and y axis labels to your graph. According to your calculations, how big of a sample size is required to guarantee greater than 80% power? Answer in a sentence.

```
ggplot(data = power_calcs_tib) +
  geom_point(aes(x=sample_size, y=power)) +
  geom_line(aes(x=sample_size, y=power), linetype="dotted") +
  geom_hline(aes(yintercept = 0.8), color = "darkblue",
             linetype = "dashed") +
  theme_bw() +
  scale_x_continuous(breaks=5:20, labels=5:20) +
  scale_y_continuous(limits=c(0.3,1.0),
                     breaks=seq(from=0.3,to=1.0,by=0.1)) +
  labs(title="Power to Detect Expected Change in Mean Mood
       Score", x="Sample Size", y="Power")
```


Power to Detect Expected Change in Mean Mood Score
Power
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Sample Size

To detect the expected change in mean mood score, the investigators need a sample size of at least 15 participants to guarantee more than 80% power.