Assignment 3

Kyle yeo

Due Friday, September 18 at 11:59 PM CT

Problems

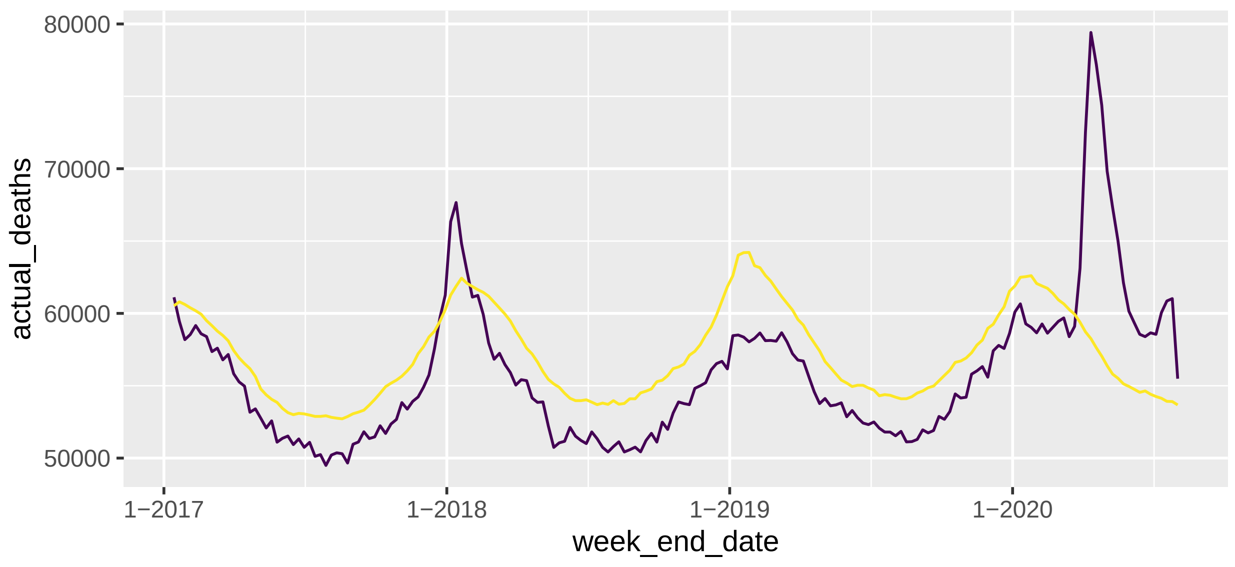
Choose to do either the set of three problems 1A, 2A, and 3A with the US Death data or the three problems 1B, 2B, 3B with the S&P 500 data.

1A

Based on death counts and population and demographic data, a department within the US Center for Disease Control (CDC) makes a prediction interval for what the total number of US deaths is expected to be. This interval accounts for changes in the population, the age structure, and several other demographic variables. The prediction interval accounts for variation at typical levels judged over many years. When the actual death count exceeds the maximum of the prediction interval for several consecutive weeks, this is evidence that there is some cause of excess deaths.

The following code is a simple line plot of the actual deaths and the maximum number of predicted deaths under normal conditions since January, 2017. Note that the command `scale_x_date()` is useful to have more control over the labeling of dates on the x-axis. Here, the pattern "%b-%Y" specifies pattern with an abbreviation for the month, a dash, and the four-digit year. See the help page by typing `?scale_x_date` for more details.

```r
ggplot(us_deaths, aes(x = week_end_date)) +
  geom_line(aes(y = actual_deaths), color = viridis3[1]) +
  geom_line(aes(y = maximum_expected_deaths), color = viridis3[3]) +
  scale_x_date(date_labels = "%b-%Y")
```



What do you thing explains the cyclical pattern in the line plot of the upper end of the prediction interval? How does the death rate tend to change with season?

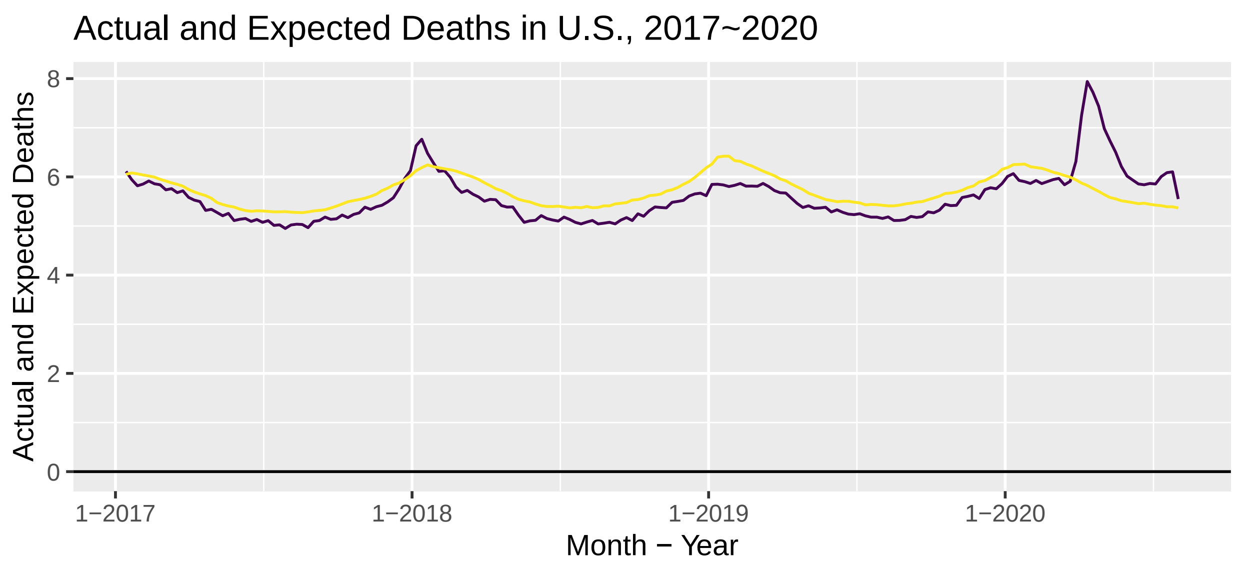
The maximum expected number of deaths. The death rate tends to increase from the end of summer to winter, especially January and decrease from winter to summer and the prediction interval well represents the actual data. However, based on the recent year, the increasing term relatively long and the number of actual deaths also exceeds the maximum expected number of deaths.

2A

Make changes to the code so that:

- The value zero appears on the y-axis so we can better compare relative rates.
- The values on the y-axis are divided by 10,000
- There are clear and informative axis labels and a title.

```r
ggplot(us_deaths, aes(x = week_end_date)) +
  geom_line(aes(y = actual_deaths/10000), color = viridis3[1]) +
  geom_hline(yintercept=0) + #or ylim(0, 8)
  geom_line(aes(y = maximum_expected_deaths/10000), color = viridis3[3]) +
  scale_x_date(date_labels = "%b-%Y") +
  xlab("Month - Year") +
  ylab("Actual and Expected Deaths") +
  ggtitle("Actual and Expected Deaths in U.S., 2017~2020")
```

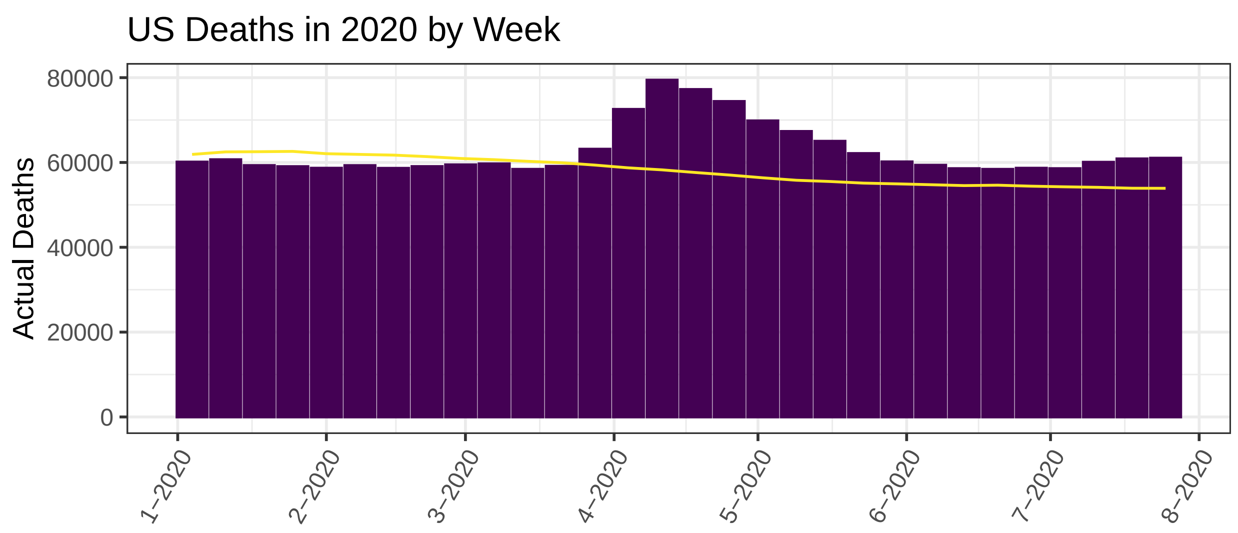


When are the two time periods when the number of actual death greatly exceeds the maximum expected number of deaths over several consecutive weeks?

1-2018 and 4-2020

3A

```r
us_deaths %>%
  filter(week_end_date > ymd("2020-01-01") & week_end_date < ymd("2020-08-01")) %>%
ggplot(aes(x = week_end_date)) +
  geom_col(aes(y = actual_deaths), color = viridis3[1], fill = viridis3[1]) +
  geom_line(aes(y = maximum_expected_deaths), color = viridis3[3]) +
  scale_x_date(date_labels = "%b-%Y", date_break = "1 month") +
  xlab("") +
  ylab("Actual Deaths") +
  ggtitle("US Deaths in 2020 by Week") +
  theme_bw() +
  theme(axis.text.x = element_text(angle=60, hjust=1))
```



> When during the year did the number of actual deaths begin to greatly exceed the maximum expected number? When did the number of excess deaths peak? Is the current trend of excess deaths increasing or decreasing?

The number of actual deaths begin to greatly exceed the maximum expected number since 4-2020.

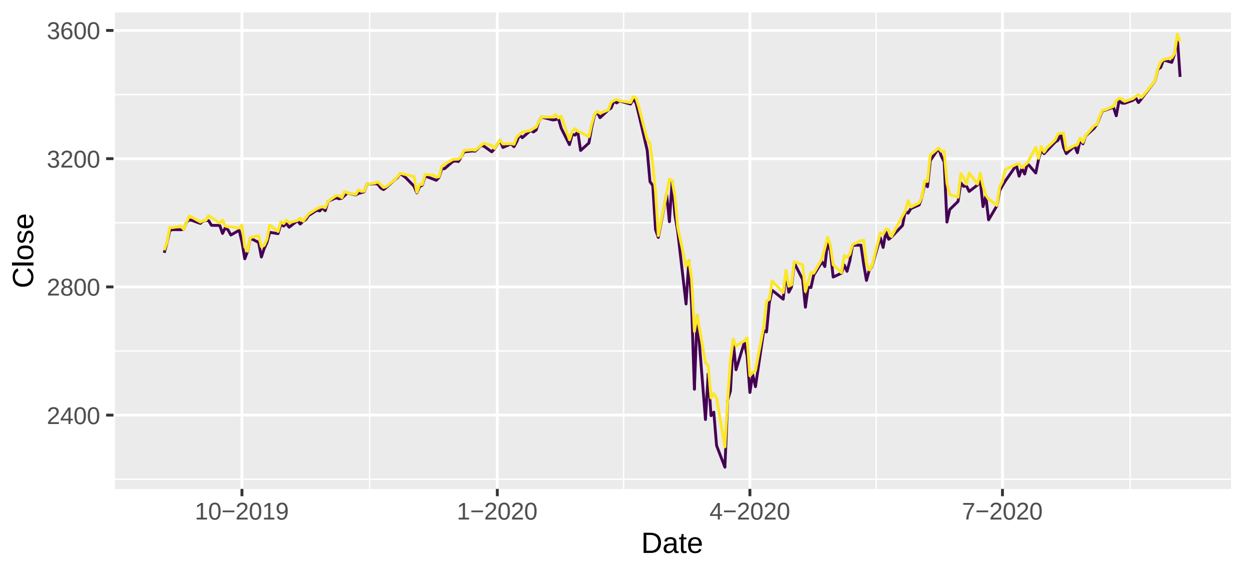
During the April, the number of actual deaths reached its peak.

The current trend of excess deaths is steadily increasing, but it cannot be said clearly.

1B

The following code is a simple line plot of the S&P 500 Index closing value and the daily high.

```
ggplot(sp500, aes(x = Date)) +
  geom_line(aes(y = Close), color = viridis3[1]) +
  geom_line(aes(y = High), color = viridis3[3]) +
  scale_x_date(date_labels = "%b-%Y")
```



Provide a description of the pattern you see.

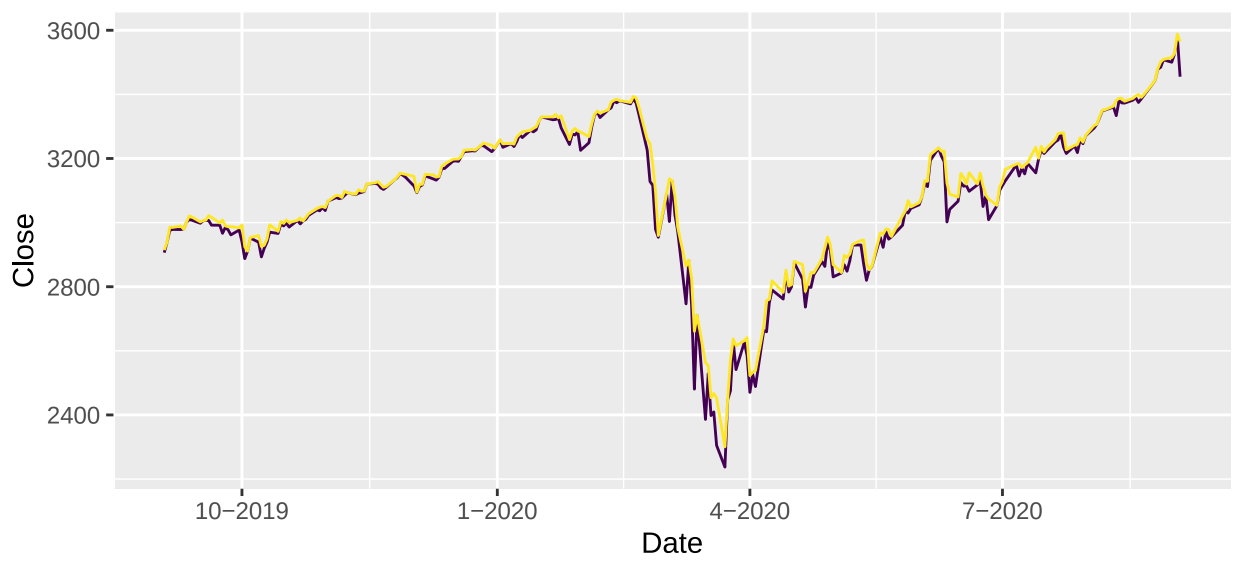
RESPONSE

2B

Make changes to the code so that:

- The value 0 appears on the y-axis.
- There are clear and informative axis labels and a title.
- Change the colors of the lines to something different.

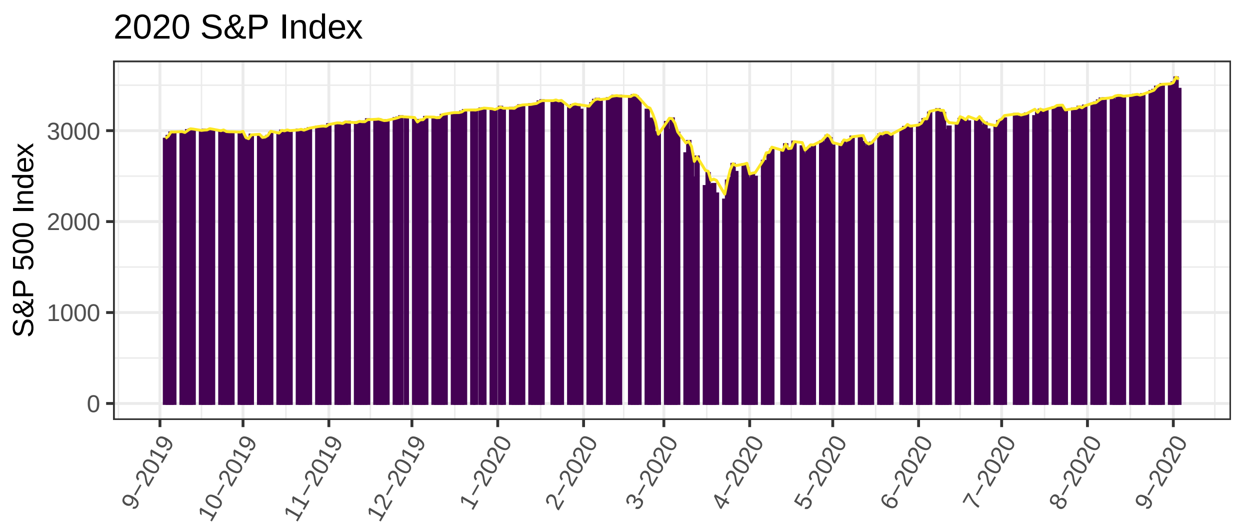
```r
ggplot(sp500, aes(x = Date)) +
  geom_line(aes(y = Close), color = viridis3[1]) +
  geom_line(aes(y = High), color = viridis3[3]) +
  scale_x_date(date_labels = "%b-%Y")
```



About what percentage of the value of S&P dropped from its peak in mid March to the lowest point in late March?

3B

```r
ggplot(sp500, aes(x = Date)) +
  geom_col(aes(y = Close), color = viridis3[1]) +
  geom_line(aes(y = High), color = viridis3[3]) +
  scale_x_date(date_labels = "%b-%Y", date_break = "1 month") +
  xlab("") +
  ylab("S&P 500 Index") +
  ggtitle("2020 S&P Index") +
  theme_bw() +
  theme(axis.text.x = element_text(angle=60, hjust=1))
```



Make the plot as above and again with the `scale_x_date()` line commented out.

> What do you think this line of code does?

RESPONSE

> Repeat for the last line of code with `theme(...)`

RESPONSE

4

> Explain the difference between the `geom_col()` and `geom_bar()` functions.

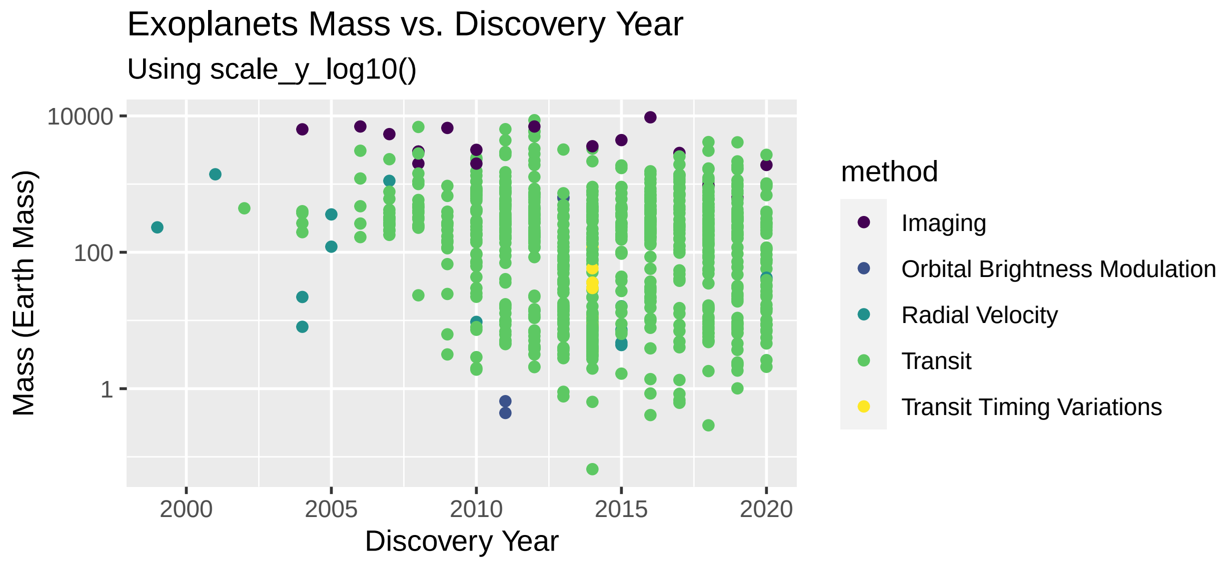
The function geom_col() represents values based on the heights of the bars, while the function geom_bar() counts values at each x position.

5

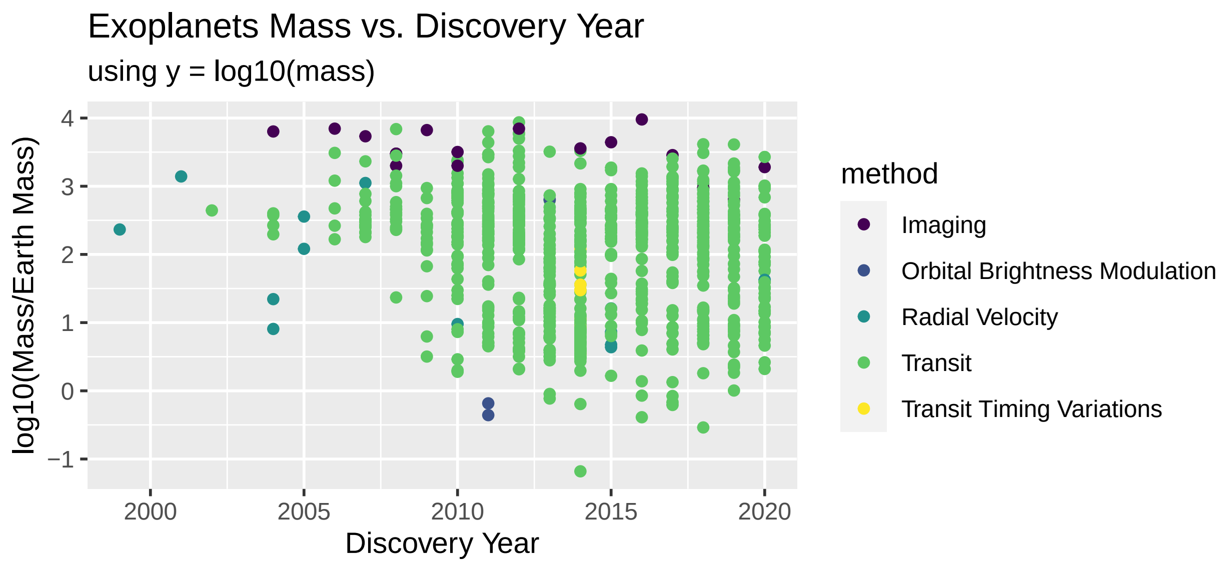
```r
## Read in the csv file
## There is one row per exoplanet after applying `filter(default_flag != 0)`
## Select some variables that we will work with and rename them
## Remove very massive planet (only to improve plot visuals)
## Drop missing values; the remaining exoplanets will have estimates of both mass and radius
planets <- read_csv("C:/stat_240/data/exoplanets-3sept2020.csv") %>%
  filter(default_flag != 0) %>%
  select(pl_name, discoverymethod, disc_year, sy_pnum, pl_rade, pl_bmasse) %>%
  rename(name=pl_name, method=discoverymethod,year=disc_year, number=sy_pnum, radius=pl_rade, mass=pl_br
  filter(mass <10000) %>%
  drop_na()
```

The following block of code make two different scatter plots of mass on a log scale versus year.

```
ggplot(planets, aes(x = year, y = mass, color = method)) +
  geom_point() +
  scale_y_log10() +
  xlab("Discovery Year") +
  ylab("Mass (Earth Mass)") +
  ggtitle("Exoplanets Mass vs. Discovery Year",
          subtitle="Using scale_y_log10()")
```



```
ggplot(planets, aes(x = year, y = log10(mass), color = method)) +
  geom_point() +
  xlab("Discovery Year") +
  ylab("log10(Mass/Earth Mass)") +
  ggtitle("Exoplanets Mass vs. Discovery Year",
          subtitle="using y = log10(mass)")
```



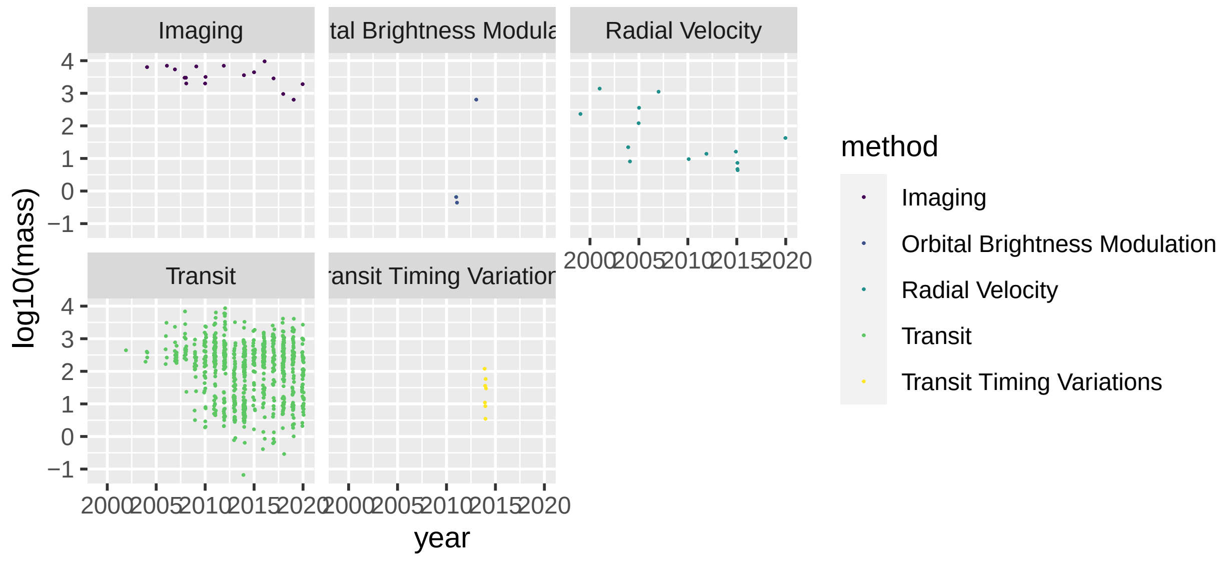
Describe the differences between the two plots. Which do you think is more effective and why?

The first plot sets y-intercepts with the scale of log10, but the second plot sets y-intercepts by converting actual values(mass) to log10(actual values). The second plot seems to be more effective since it has more specific y-intercepts so that it can help us to read data more accurately.

6

Using the exoplanet data, make a plot with discovery year on the x axis, mass on the log10 scale on the y-axis, and with a different facet for each method. Set the argument `position` within the `geom_point()` command using the command `position_jitter()` so that points are jittered horizontally, but not vertically. Choose an amount of jitter so that there is a reduction in overplotting, but all points for a single year appear as a band without overlap with other years.

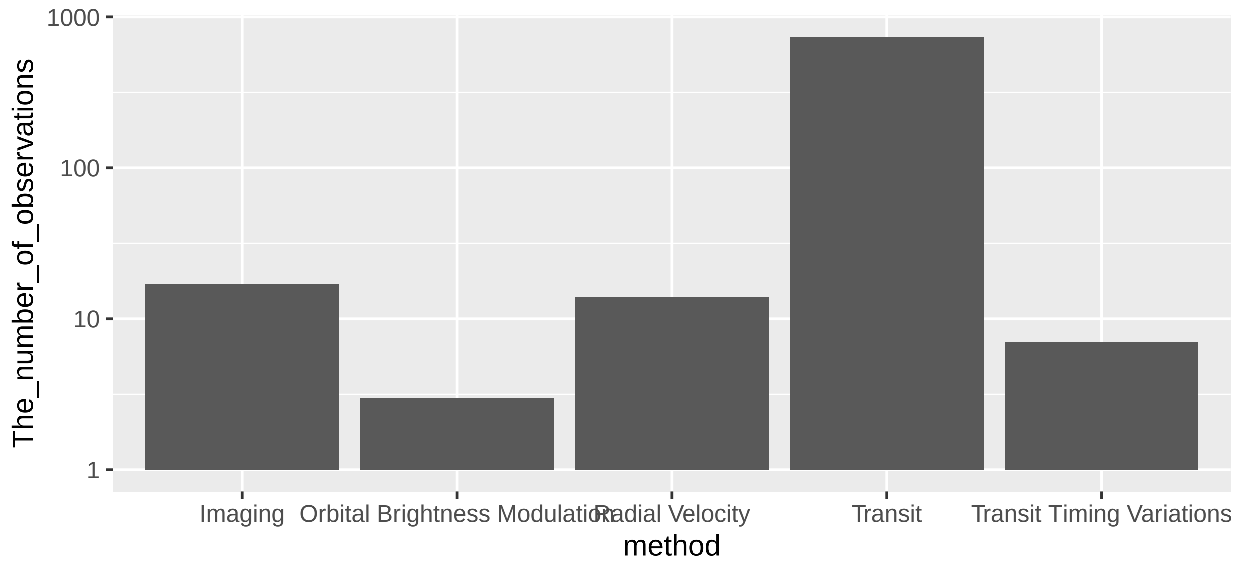
```r
ggplot(planets, aes(x=year,y=log10(mass), color = method)) +
  geom_point(position = position_jitter(w = 0.1, h = 0), size =0) +
  facet_wrap(~method)
```



7

With the exoplanet data, make a bar graph of the method variable that displays the count of the number of observations for each method.

```
planets3 <- planets %>%
  group_by(method) %>%
  summarise(The_number_of_observations = n())
ggplot(planets3, aes(x=method, y=The_number_of_observations)) +
  scale_y_log10() +
  geom_col()
```

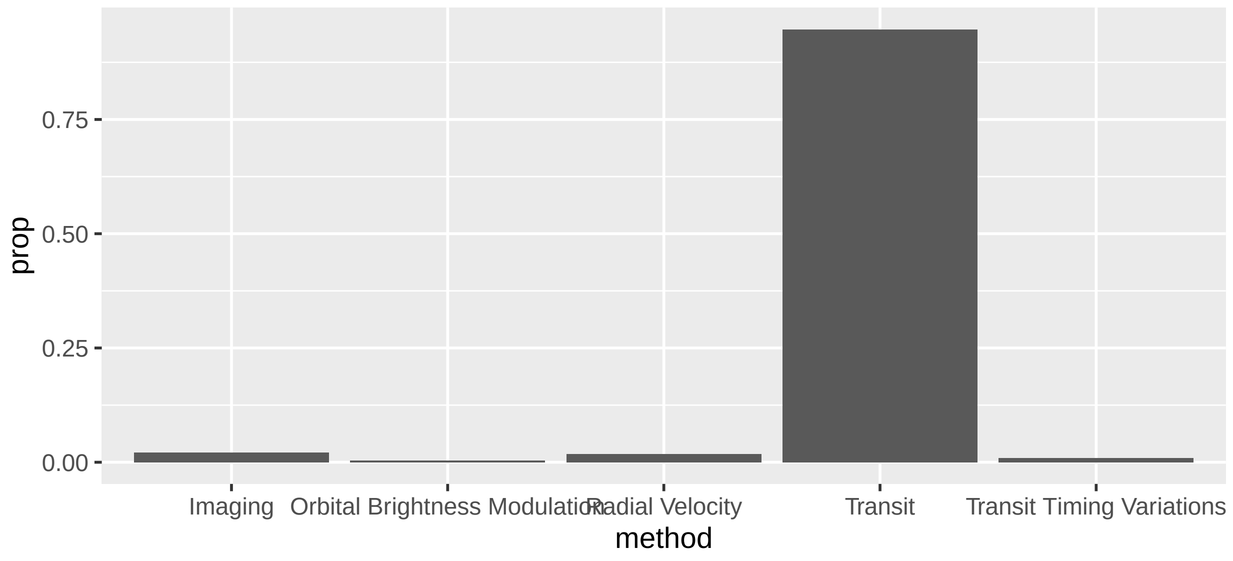


8

Repeat the previous problem, but replace the counts on the y-axis with proportions. (*Hint:* You need to set the y aesthetic to `stat(prop)` and the group aesthetic to `1`. See the second example in *R for Data Science* section 3.7.)

```
planets3 <- planets %>%
  group_by(method)

ggplot(planets3) +
  geom_bar(mapping = aes(x=method, y = stat(prop), group = 1))
```



9

The following block of code reads in the raw Lake Mendota data and makes several transformations. Examine indicated sections of the code and answer the corresponding questions.

```
mendota_interval = read_csv("C:/stat_240/data/lake-mendota-raw.csv")%>%

## question (a) begin
  select(-days) %>%
## question (a) end
## question (b) begin
  drop_na() %>%
#str(mendota_interval)
## question (b) end
## question (c) begin
  separate(winter,into = c("year1","year2"), remove = FALSE) %>%
  mutate(year1 = as.numeric(year1)) %>%
  mutate(year2 = year1+1) %>%
## question (c) end
  mutate(closed = case_when(
    str_detect(closed,"Oct|Nov|Dec") ~ str_c(closed,' ',year1),
    str_detect(closed,"Jan|Feb|Mar|Apr|May") ~ str_c(closed,' ',year2),
    TRUE ~ NA_character_
  )) %>%
  mutate(closed = dmy(closed)) %>%
  mutate(open = case_when(
    str_detect(open,"Oct|Nov|Dec") ~ str_c(open,' ',year1),
    str_detect(open,"Jan|Feb|Mar|Apr|May") ~ str_c(open,' ',year2),
    TRUE ~ NA_character_
  )) %>%
  mutate(open = dmy(open)) %>%
  mutate(days = open - closed)

mendota = mendota_interval %>%
## question (d) begin
```

```
  group_by(winter) %>%
  summarize(intervals = n(),
            days = sum(days),
            first_freeze = min(closed),
            last_thaw = max(open)) %>%
## questions (d) end
  mutate(year1 = as.numeric(str_sub(winter,1,4))) %>%
  mutate(decade = floor(year1 / 10) * 10) %>%
  select(winter,year1,everything())
```

(a)

What does the line `select(-days)` do to the data set?

select variables except 'days'

(b)

What does the command `drop_na()` do? How many rows (observations) are in the data set when it is first read in and how many rows remain after this code is executed?

The command 'drop_na()' drops all the empty data. There were 700 rows (1:175 * 4) in the data set when it is first read in and 660 rows remain after the code is executed.

(c)

Describe the effect these three lines of code have on the data set

The first line of code seperates 'winter' variable into the vector of year1 and year2. The second and third lines of code make new columns which are year1 as numeric and year2 as year1+1, respectively.

(d)

Explain what the effect of these two commands are. In your response, describe what the effect of the `group_by(winter)` command is, what the function `n()` does, and what the functions `sum()`, `min()`, and `max()` do.

The group_by command takes an original tibble and converts it into a grouped data frame. The summarize command reduces a data set to a summary of just one vector or value. 'group_by(winter)' command converts an existing tibble into tibble grouped with the value 'winter'. 'n()' command count the number of variables in interval. 'sum()' gives us the sum of the variables. 'min()' gives us the minimum value among the variables. 'max()' gives us the maximum value among the variables.