Session 4: Homework 2 - Group 23

Leif Beckers, Dung Tran, Salman Abdullah, Andjela Bozinovic, Xiwen Wang

2020-10-03

Table of Contents

# Climate change and temperature anomalies

If we wanted to study climate change, we can find data on the *Combined Land-Surface Air and Sea-Surface Water Temperature Anomalies* in the Northern Hemisphere at [NASA’s Goddard Institute for Space Studies](https://data.giss.nasa.gov/gistemp). The [tabular data of temperature anomalies can be found here](https://data.giss.nasa.gov/gistemp/tabledata_v3/NH.Ts+dSST.txt)

To define temperature anomalies you need to have a reference, or base, period which NASA clearly states that it is the period between 1951-1980.

weather <-   
 read\_csv("https://data.giss.nasa.gov/gistemp/tabledata\_v3/NH.Ts+dSST.csv",   
 skip = 1,   
 na = "\*\*\*")

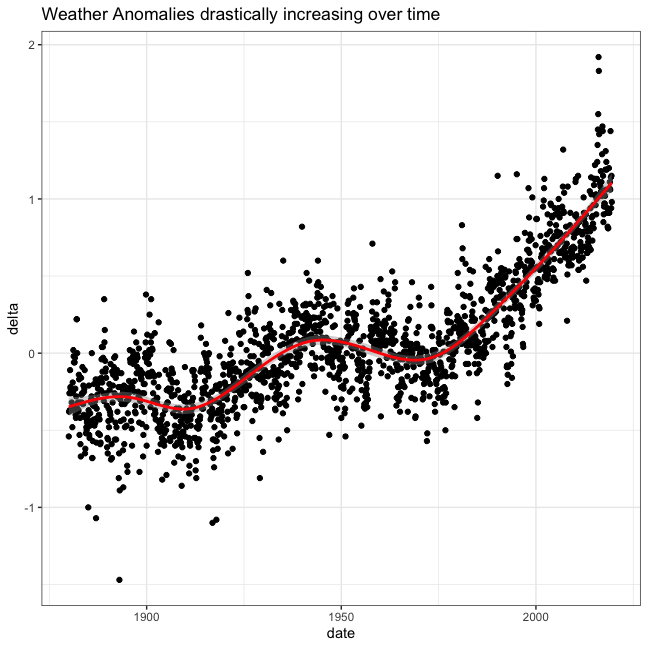
tidyweather <- weather %>%  
 select(-c("J-D", "D-N", "DJF", "MAM", "JJA", "SON")) %>%  
 pivot\_longer(cols = 2:13, names\_to = "Month", values\_to = "delta")

## Plotting Information

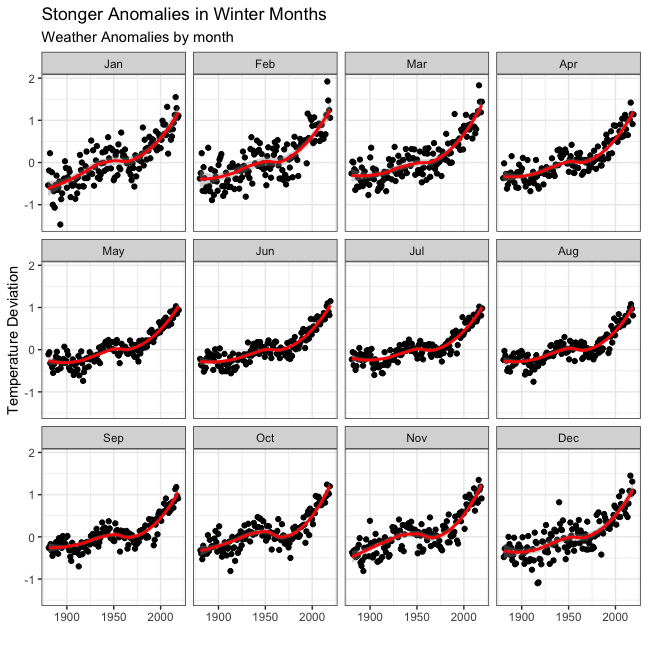
Let us plot the data using a time-series scatter plot, and add a trendline. To do that, we first need to create a new variable called date in order to ensure that the delta values are plot chronologically.

In the following chunk of code, I used the eval=FALSE argument, which does not run a chunk of code; I did so that you can knit the document before tidying the data and creating a new dataframe tidyweather. When you actually want to run this code and knit your document, you must delete eval=FALSE, **not just here but in all chunks were eval=FALSE appears.**

tidyweather <- tidyweather %>%  
 mutate(date = ymd(paste(as.character(Year), Month, "1")),  
 month = month(date), #had to get rid of label = true. TA advised could be a package loading problem  
 year = year(date))  
  
ggplot(tidyweather, aes(x=date, y = delta))+  
 geom\_point()+  
 geom\_smooth(color="red") +  
 theme\_bw() +  
 labs (  
 title = "Weather Anomalies drastically increasing over time"  
 )



Is the effect of increasing temperature more pronounced in some months? Use facet\_wrap() to produce a seperate scatter plot for each month, again with a smoothing line. Your chart should human-readable labels; that is, each month should be labeled “Jan”, “Feb”, “Mar” (full or abbreviated month names are fine), not 1, 2, 3.



It is sometimes useful to group data into different time periods to study historical data. For example, we often refer to decades such as 1970s, 1980s, 1990s etc. to refer to a period of time. NASA calcuialtes a temperature anomaly, as difference form the base periof of 1951-1980. The code below creates a new data frame called comparison that groups data in five time periods: 1881-1920, 1921-1950, 1951-1980, 1981-2010 and 2011-present.

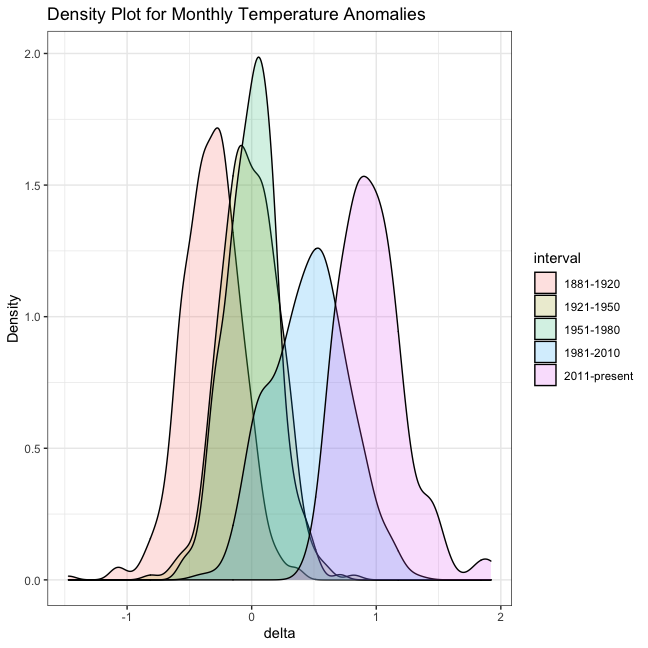
We remove data before 1800 and before using filter. Then, we use the mutate function to create a new variable interval which contains information on which period each observation belongs to. We can assign the different periods using case\_when().

comparison <- tidyweather %>%   
 filter(Year>= 1881) %>% #remove years prior to 1881  
 #create new variable 'interval', and assign values based on criteria below:  
 mutate(interval = case\_when(  
 Year %in% c(1881:1920) ~ "1881-1920",  
 Year %in% c(1921:1950) ~ "1921-1950",  
 Year %in% c(1951:1980) ~ "1951-1980",  
 Year %in% c(1981:2010) ~ "1981-2010",  
 TRUE ~ "2011-present"  
 ))

Inspect the comparison dataframe by clicking on it in the Environment pane.

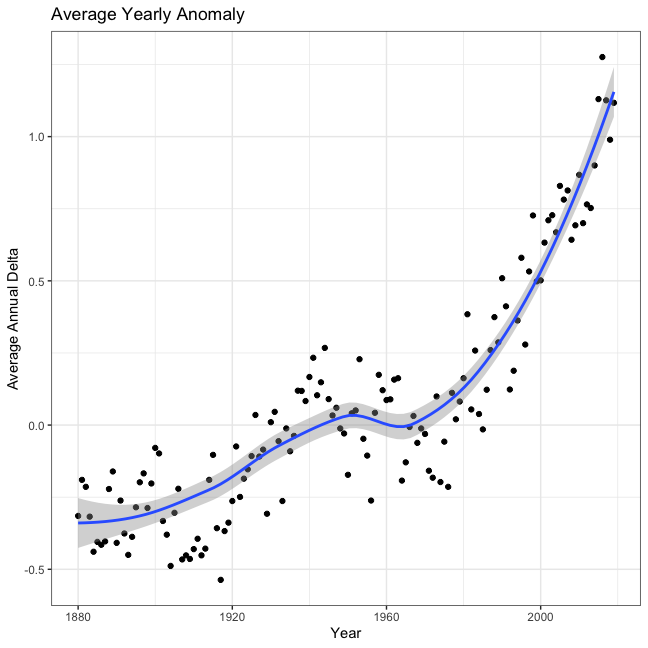
Now that we have the interval variable, we can create a density plot to study the distribution of monthly deviations (delta), grouped by the different time periods we are interested in. Set fill to interval to group and colour the data by different time periods.

ggplot(comparison, aes(x=delta, fill=interval))+  
 geom\_density(alpha=0.2) + #density plot with tranparency set to 20%  
 theme\_bw() + #theme  
 labs (  
 title = "Density Plot for Monthly Temperature Anomalies",  
 y = "Density" #changing y-axis label to sentence case  
 )



So far, we have been working with monthly anomalies. However, we might be interested in average annual anomalies. We can do this by using group\_by() and summarise(), followed by a scatter plot to display the result.

#creating yearly averages  
average\_annual\_anomaly <- tidyweather %>%   
 group\_by(Year) %>% #grouping data by Year  
   
 # creating summaries for mean delta   
 # use `na.rm=TRUE` to eliminate NA (not available) values   
 summarise(annual\_average\_delta = mean(delta, na.rm=TRUE))   
  
#plotting the data:  
ggplot(average\_annual\_anomaly, aes(x=Year, y= annual\_average\_delta))+  
 geom\_point()+  
   
 #Fit the best fit line, using LOESS method  
 geom\_smooth() +  
   
 #change to theme\_bw() to have white background + black frame around plot  
 theme\_bw() +  
 labs (  
 title = "Average Yearly Anomaly",  
 y = "Average Annual Delta"  
 )



## Confidence Interval for delta

[NASA points out on their website](https://earthobservatory.nasa.gov/world-of-change/decadaltemp.php) that

A one-degree global change is significant because it takes a vast amount of heat to warm all the oceans, atmosphere, and land by that much. In the past, a one- to two-degree drop was all it took to plunge the Earth into the Little Ice Age.

Your task is to construct a confidence interval for the average annual delta since 2011, both using a formula and using a bootstrap simulation with the infer package. Recall that the dataframe comparison has already grouped temperature anomalies according to time intervals; we are only interested in what is happening between 2011-present.

formula\_ci <- comparison %>%   
 # clean NAs and choose the interval 2011-present  
 drop\_na(delta) %>%   
 filter(Year >= 2011) %>%   
 # calculate yearly mean temperature deviation (delta)   
 group\_by(Year) %>%   
 summarise(year\_mean\_delta = mean(delta)) %>%   
   
 # Confidence Interval (CI) using the formula mean +- MoE  
 summarise(mean\_delta = mean(year\_mean\_delta), # calculate summary statistics for yearly mean temperature deviation (delta)   
 sd\_delta = sd(year\_mean\_delta),  
 count = n(),  
 t\_critical = qt(0.975, count-1), # get t-critical value with (n-1) degrees of freedom  
 se\_delta = sd(year\_mean\_delta)/sqrt(count), # calculate mean, SD, count, SE, lower/upper 95% CI  
 margin\_of\_error = t\_critical \* se\_delta,  
 delta\_low = mean\_delta - margin\_of\_error,  
 delta\_high = mean\_delta + margin\_of\_error)  
  
  
formula\_ci

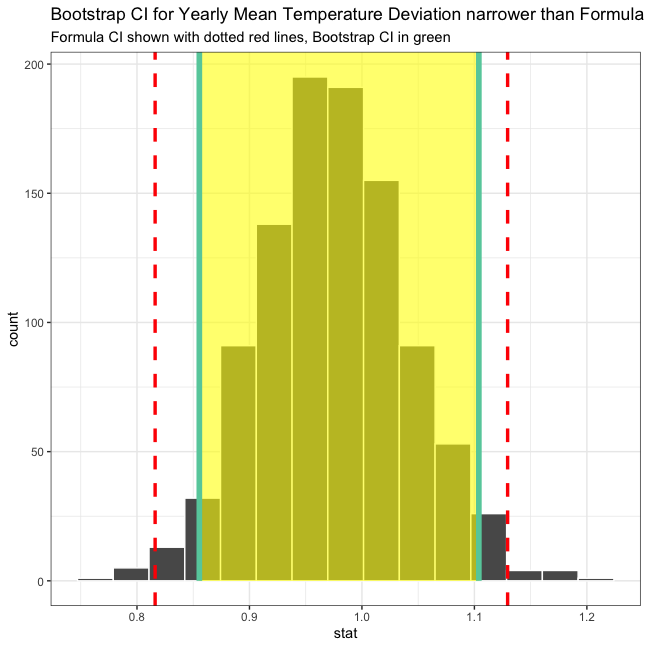
## # A tibble: 1 x 8  
## mean\_delta sd\_delta count t\_critical se\_delta margin\_of\_error delta\_low  
## <dbl> <dbl> <int> <dbl> <dbl> <dbl> <dbl>  
## 1 0.973 0.204 9 2.31 0.0680 0.157 0.816  
## # … with 1 more variable: delta\_high <dbl>

set.seed(1234)  
  
boot\_yearly\_mean\_delta <- comparison %>%  
   
 # Get rid of NAs in 2019  
 drop\_na(delta) %>%   
   
 # Choose only 2011 and following  
 filter(Year >= 2011) %>%  
   
 # Create yearly mean deltas  
 group\_by(Year) %>%   
 summarise(year\_mean\_delta = mean(delta)) %>%  
   
 # Specify the variable of interest  
 specify(response = year\_mean\_delta) %>%  
   
 # Generate a bunch of bootstrap samples  
 generate(reps = 1000, type = "bootstrap") %>%  
   
 # Find the mean of each sample  
 calculate(stat = "mean")  
  
# Calculate bootstrap method confidence intervals  
percentile\_ci <- boot\_yearly\_mean\_delta %>%   
 get\_confidence\_interval(level = 0.95, type = "percentile")  
percentile\_ci

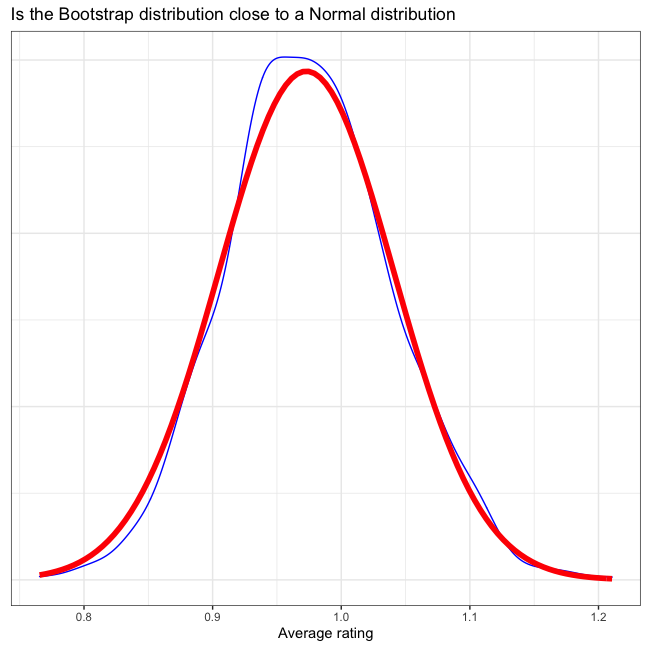
## # A tibble: 1 x 2  
## lower\_ci upper\_ci  
## <dbl> <dbl>  
## 1 0.855 1.10

What is the data showing us? Please type your answer after (and outside!) this blockquote. You have to explain what you have done, and the interpretation of the result. One paragraph max, please!

# Visualise bootstrap CI vs formula CI  
visualize(boot\_yearly\_mean\_delta) +  
 shade\_ci(endpoints = percentile\_ci,fill = "yellow")+  
 labs(title='Bootstrap CI for Yearly Mean Temperature Deviation narrower than Formula CI',  
 subtitle = 'Formula CI shown with dotted red lines, Bootstrap CI in green')+  
 geom\_vline(xintercept = formula\_ci$delta\_low, colour = "red", linetype="dashed", size=1.2)+  
 geom\_vline(xintercept = formula\_ci$delta\_high, colour = "red", linetype="dashed", size=1.2)+  
 theme\_bw()+  
 NULL



# compare bootstrap distribution with a Normal distribution with parameters estimated from the sample  
ggplot(boot\_yearly\_mean\_delta, aes(x = stat)) +  
 geom\_density(color="blue") +  
 stat\_function(  
 fun = dnorm,  
 color = "red",  
 size = 2,  
 args = list(mean = formula\_ci$mean\_delta, sd = formula\_ci$se\_delta)  
 )+  
 theme\_bw()+  
 labs(title = "Is the Bootstrap distribution close to a Normal distribution",  
 x= 'Average rating', y = "")+  
 theme(axis.title.y=element\_blank(),  
 axis.text.y=element\_blank(),  
 axis.ticks.y=element\_blank())+  
 NULL



# General Social Survey (GSS)

The [General Social Survey (GSS)](http://www.gss.norc.org/) gathers data on American society in order to monitor and explain trends in attitudes, behaviours, and attributes. Many trends have been tracked for decades, so one can see the evolution of attitudes, etc in American Society.

In this assignment we analyze data from the **2016 GSS sample data**, using it to estimate values of *population parameters* of interest about US adults. The GSS sample data file has 2867 observations of 935 variables, but we are only interested in very few of these variables and you are using a smaller file.

gss <- read\_csv(here::here("data", "smallgss2016.csv"),   
 na = c("", "Don't know",  
 "No answer", "Not applicable"))

You will also notice that many responses should not be taken into consideration, like “No Answer”, “Don’t Know”, “Not applicable”, “Refused to Answer”.

We will be creating 95% confidence intervals for population parameters. The variables we have are the following:

* hours and minutes spent on email weekly. The responses to these questions are recorded in the emailhr and emailmin variables. For example, if the response is 2.50 hours, this would be recorded as emailhr = 2 and emailmin = 30.
* snapchat, instagrm, twitter: whether respondents used these social media in 2016
* sex: Female - Male
* degree: highest education level attained

## Instagram and Snapchat, by sex

Can we estimate the *population* proportion of Snapchat or Instagram users in 2016?

1. Create a new variable, snap\_insta that is *Yes* if the respondent reported using any of Snapchat (snapchat) or Instagram (instagrm), and *No* if not. If the recorded value was NA for both of these questions, the value in your new variable should also be NA.
2. Calculate the proportion of Yes’s for snap\_insta among those who answered the question, i.e. excluding NAs.
3. Using the CI formula for proportions, please construct 95% CIs for men and women who used either Snapchat or Instagram

gss\_new <- gss %>%  
 mutate(snap\_insta = if\_else(snapchat == "Yes", "Yes", if\_else(instagrm == "Yes", "Yes", "No", "NA"), "NA")) %>%  
 mutate(snap\_insta = na\_if(snapchat, "NA"))  
  
  
Insta\_snap <- gss\_new %>%  
 group\_by(snap\_insta) %>%  
 filter(!is.na(snap\_insta)) %>%  
 summarise(count = n()) %>%  
 mutate(usage\_proportion = count/sum(count)) %>%  
 filter(snap\_insta == "Yes")  
  
CI\_by\_sex <- gss\_new %>%  
 filter(!is.na(snap\_insta)) %>%  
 group\_by(sex, snap\_insta) %>%  
 summarise(count = n()) %>%  
 mutate(usage\_proportion = count/sum(count),  
 t\_critical = qt(0.975, sum(count)-1),  
 se\_proportion = sqrt(usage\_proportion \* (1 - usage\_proportion)/sum(count)),  
 margin\_of\_error = t\_critical \* se\_proportion,  
 proportion\_low = usage\_proportion - margin\_of\_error,  
 proportion\_high = usage\_proportion + margin\_of\_error) %>%  
 filter(snap\_insta == "Yes") %>%  
 select(sex, usage\_proportion, proportion\_low, proportion\_high)  
  
gss\_new

## # A tibble: 2,867 x 8  
## emailmin emailhr snapchat instagrm twitter sex degree snap\_insta  
## <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>   
## 1 0 12 NA NA NA Male Bachelor <NA>   
## 2 30 0 No No No Male High school No   
## 3 NA NA No No No Male Bachelor No   
## 4 10 0 NA NA NA Female High school <NA>   
## 5 NA NA Yes Yes No Female Graduate Yes   
## 6 0 2 No Yes No Female Junior college No   
## 7 0 40 NA NA NA Male High school <NA>   
## 8 NA NA Yes Yes No Female High school Yes   
## 9 0 0 NA NA NA Male High school <NA>   
## 10 NA NA No No No Male Junior college No   
## # … with 2,857 more rows

Insta\_snap

## # A tibble: 1 x 3  
## snap\_insta count usage\_proportion  
## <chr> <int> <dbl>  
## 1 Yes 311 0.227

CI\_by\_sex

## # A tibble: 2 x 4  
## # Groups: sex [2]  
## sex usage\_proportion proportion\_low proportion\_high  
## <chr> <dbl> <dbl> <dbl>  
## 1 Female 0.234 0.204 0.264  
## 2 Male 0.217 0.184 0.250

## Twitter, by education level

Can we estimate the *population* proportion of Twitter users by education level in 2016?.

There are 5 education levels in variable degree which, in ascneding order of years of education, are Lt high school, High School, Junior college, Bachelor, Graduate.

1. Turn degree from a character variable into a factor variable. Make sure the order is the correct one and that levels are not sorted alphabetically which is what R by default does.
2. Create a new variable, bachelor\_graduate that is *Yes* if the respondent has either a Bachelor or Graduate degree. As before, if the recorded value for either was NA, the value in your new variable should also be NA.
3. Calculate the proportion of bachelor\_graduate who do (Yes) and who don’t (No) use twitter.
4. Using the CI formula for proportions, please construct two 95% CIs for bachelor\_graduate vs whether they use (Yes) and don’t (No) use twitter.
5. Do these two Confidence Intervals overlap?

gss$degree <- as.factor(gss$degree)  
gss$degree <- factor(gss$degree, levels = c("Lt high school", "High school", "Junior college", "Bachelor", "Graduate"))  
  
gss\_edu <- gss %>%  
 mutate(bachelor\_graduate = if\_else(degree %in% c("Bachelor", "Graduate"), "Yes", "No", "NA"),  
 bachelor\_graduate = na\_if(bachelor\_graduate, "NA"))  
  
twitter\_usage <- gss\_edu %>%   
 filter(bachelor\_graduate == "Yes",  
 twitter != "NA") %>%  
 group\_by(twitter) %>%  
 summarise(count = n()) %>%  
 mutate(usage\_proportion = count/sum(count))  
  
CI\_twitter\_by\_edu <- gss\_edu %>%  
 filter(twitter != "NA",  
 !is.na(bachelor\_graduate)) %>%  
 group\_by(bachelor\_graduate, twitter) %>%  
 summarise(count = n()) %>%  
 mutate(usage\_proportion = count/sum(count),  
 t\_critical = qt(0.975, sum(count)-1),  
 se\_proportion = sqrt(usage\_proportion \* (1 - usage\_proportion)/sum(count)),  
 margin\_of\_error = t\_critical \* se\_proportion,  
 proportion\_low = usage\_proportion - margin\_of\_error,  
 proportion\_high = usage\_proportion + margin\_of\_error) %>%   
 filter(twitter == "Yes")  
  
gss\_edu

## # A tibble: 2,867 x 8  
## emailmin emailhr snapchat instagrm twitter sex degree bachelor\_gradua…  
## <chr> <chr> <chr> <chr> <chr> <chr> <fct> <chr>   
## 1 0 12 NA NA NA Male Bachelor Yes   
## 2 30 0 No No No Male High scho… No   
## 3 NA NA No No No Male Bachelor Yes   
## 4 10 0 NA NA NA Female High scho… No   
## 5 NA NA Yes Yes No Female Graduate Yes   
## 6 0 2 No Yes No Female Junior co… No   
## 7 0 40 NA NA NA Male High scho… No   
## 8 NA NA Yes Yes No Female High scho… No   
## 9 0 0 NA NA NA Male High scho… No   
## 10 NA NA No No No Male Junior co… No   
## # … with 2,857 more rows

twitter\_usage

## # A tibble: 2 x 3  
## twitter count usage\_proportion  
## <chr> <int> <dbl>  
## 1 No 375 0.767  
## 2 Yes 114 0.233

CI\_twitter\_by\_edu

## # A tibble: 2 x 9  
## # Groups: bachelor\_graduate [2]  
## bachelor\_gradua… twitter count usage\_proportion t\_critical se\_proportion  
## <chr> <chr> <int> <dbl> <dbl> <dbl>  
## 1 No Yes 141 0.160 1.96 0.0123  
## 2 Yes Yes 114 0.233 1.96 0.0191  
## # … with 3 more variables: margin\_of\_error <dbl>, proportion\_low <dbl>,  
## # proportion\_high <dbl>

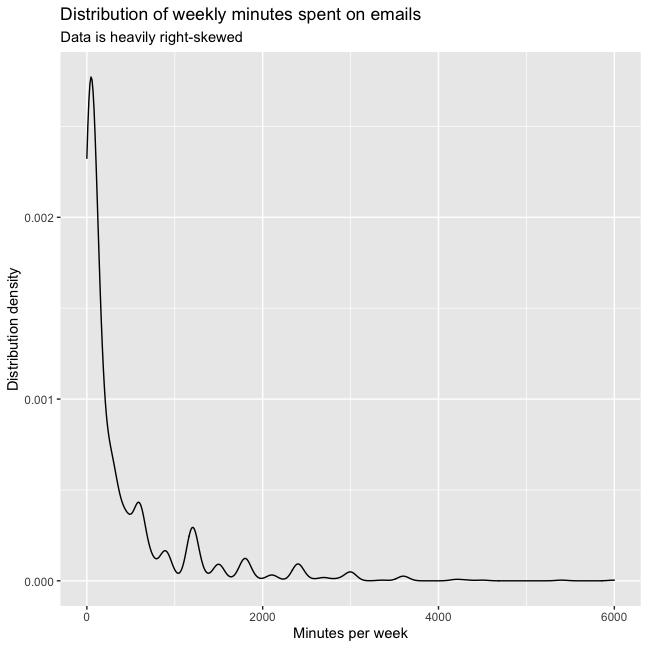
The confidence intervals between the less-educated and more-educated groups do not overlap. In fact, the data suggest that we can be 95% confident that those with bachelor degrees or above are more likely to to use twitter than those without.

## Email usage

Can we estimate the *population* parameter on time spent on email weekly?

1. Create a new variable called email that combines emailhr and emailmin to reports the number of minutes the respondents spend on email weekly.
2. Visualise the distribution of this new variable. Find the mean and the median number of minutes respondents spend on email weekly. Is the mean or the median a better measure of the typical amoung of time Americans spend on email weekly? Why?
3. Using the infer package, calculate a 95% bootstrap confidence interval for the mean amount of time Americans spend on email weekly. Interpret this interval in context of the data, reporting its endpoints in “humanized” units (e.g. instead of 108 minutes, report 1 hr and 8 minutes). If you get a result that seems a bit odd, discuss why you think this might be the case.
4. Would you expect a 99% confidence interval to be wider or narrower than the interval you calculated above? Explain your reasoning.

gss\_email <- gss %>%   
 mutate(email = as.numeric(emailhr) \* 60 + as.numeric(emailmin))   
  
ggplot(gss\_email, aes(x = email)) +  
 geom\_density() +  
 labs(title = "Distribution of weekly minutes spent on emails",  
 subtitle = "Data is heavily right-skewed",  
 x = "Minutes per week",  
 y = "Distribution density") +  
 NULL



skim(gss\_email$email)

Data summary

|  |  |
| --- | --- |
| Name | gss\_email$email |
| Number of rows | 2867 |
| Number of columns | 1 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| numeric | 1 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: numeric**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| data | 1218 | 0.58 | 417 | 680 | 0 | 50 | 120 | 480 | 6000 | ▇▁▁▁▁ |

Here, the data is heavily right-skewed. This means that calculating the mean would incorporate all values in the data, including outliers whose values are far from the desired ‘average’. As this can easily sway the data, using the median would be a better option.

set.seed(1234)  
  
boot\_email <- gss\_email %>%  
   
 # Get rid of NAs  
 drop\_na(email) %>%   
   
 # Specify the variable of interest  
 specify(response = email) %>%  
   
 # Generate a bunch of bootstrap samples  
 generate(reps = 1000, type = "bootstrap") %>%  
   
 # Find the mean of each sample  
 calculate(stat = "mean")   
  
#Calculate bootstrap method confidence intervals  
email\_CI\_95 <- boot\_email %>%  
 get\_confidence\_interval(level = 0.95, type = "percentile") %>%   
 mutate(Lower\_hour = floor(lower\_ci/60),  
 Lower\_minutes = round((lower\_ci - 60 \* Lower\_hour), digits = 0),  
 Upper\_hour = floor(upper\_ci/60),  
 Upper\_minutes = round((upper\_ci - 60 \* Upper\_hour), digits = 0),  
 Lower = paste(Lower\_hour, "hr", Lower\_minutes, "mins"),  
 Upper = paste(Upper\_hour, "hr" , Upper\_minutes, "mins")) %>%  
 select(Lower, Upper)  
  
email\_CI\_95

## # A tibble: 1 x 2  
## Lower Upper   
## <chr> <chr>   
## 1 6 hr 25 mins 7 hr 33 mins

We are 95% certain that i the mean weekly hours that Americans spent on email is between 6 hr 25 minutes - 7 hr 33 minutes.

email\_CI\_99 <- boot\_email %>%   
 get\_confidence\_interval(level = 0.99, type = "percentile") %>%  
 mutate(Lower\_hour = floor(lower\_ci/60),  
 Lower\_minutes = round((lower\_ci - 60 \* Lower\_hour), digits = 0),  
 Upper\_hour = floor(upper\_ci/60),  
 Upper\_minutes = round((upper\_ci - 60 \* Upper\_hour), digits = 0),  
 Lower = paste(Lower\_hour, "hr", Lower\_minutes, "mins"),  
 Upper = paste(Upper\_hour, "hr" , Upper\_minutes, "mins")) %>%  
 select(Lower, Upper)  
  
  
email\_CI\_99

## # A tibble: 1 x 2  
## Lower Upper   
## <chr> <chr>   
## 1 6 hr 15 mins 7 hr 45 mins

As expected, the 99% confidence interval is wider than the 95% interval. The reason is that with a 99% confidence interval, we can be more confident that the actual mean is within the calculated interval, as compared to the 95% case. For this to be the case, the calculated range should be wider to incorporate additional values that would otherwise be excluded in the 95% confidence interval.

# Trump’s Approval Margins

As we saw in class, fivethirtyeight.com has detailed data on [all polls that track the president’s approval](https://projects.fivethirtyeight.com/trump-approval-ratings)

# Import approval polls data  
approval\_polllist <- read\_csv(here::here('data', 'approval\_polllist.csv'))  
  
# or directly off fivethirtyeight website  
# approval\_polllist <- read\_csv('https://projects.fivethirtyeight.com/trump-approval-data/approval\_polllist.csv')   
  
glimpse(approval\_polllist)

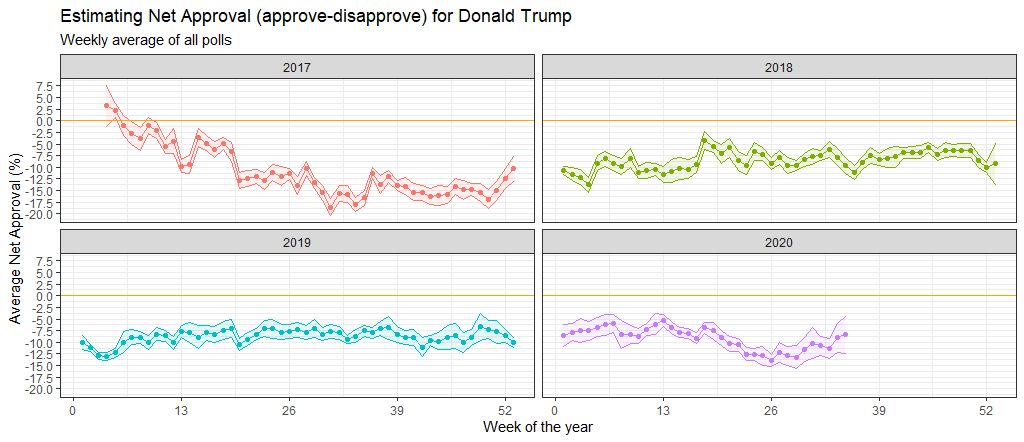
## Rows: 15,619  
## Columns: 22  
## $ president <chr> "Donald Trump", "Donald Trump", "Donald Trump", "…  
## $ subgroup <chr> "All polls", "All polls", "All polls", "All polls…  
## $ modeldate <chr> "9/27/2020", "9/27/2020", "9/27/2020", "9/27/2020…  
## $ startdate <chr> "1/20/2017", "1/20/2017", "1/20/2017", "1/21/2017…  
## $ enddate <chr> "1/22/2017", "1/22/2017", "1/24/2017", "1/23/2017…  
## $ pollster <chr> "Gallup", "Morning Consult", "Ipsos", "Gallup", "…  
## $ grade <chr> "B", "B/C", "B-", "B", "B-", "C+", "B+", "B", "C+…  
## $ samplesize <dbl> 1500, 1992, 1632, 1500, 1651, 1500, 1190, 1500, 1…  
## $ population <chr> "a", "rv", "a", "a", "a", "lv", "rv", "a", "lv", …  
## $ weight <dbl> 0.262, 0.680, 0.153, 0.243, 0.142, 0.200, 1.514, …  
## $ influence <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…  
## $ approve <dbl> 45.0, 46.0, 42.1, 45.0, 42.3, 57.0, 36.0, 46.0, 5…  
## $ disapprove <dbl> 45.0, 37.0, 45.2, 46.0, 45.8, 43.0, 44.0, 45.0, 4…  
## $ adjusted\_approve <dbl> 45.7, 45.3, 43.2, 45.7, 43.4, 51.5, 37.6, 46.7, 5…  
## $ adjusted\_disapprove <dbl> 43.6, 38.3, 43.9, 44.6, 44.5, 44.5, 42.8, 43.6, 4…  
## $ multiversions <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N…  
## $ tracking <lgl> TRUE, NA, TRUE, TRUE, TRUE, TRUE, NA, TRUE, TRUE,…  
## $ url <chr> "http://www.gallup.com/poll/201617/gallup-daily-t…  
## $ poll\_id <dbl> 49253, 49249, 49426, 49262, 49425, 49266, 49260, …  
## $ question\_id <dbl> 77265, 77261, 77599, 77274, 77598, 77278, 77272, …  
## $ createddate <chr> "1/23/2017", "1/23/2017", "3/1/2017", "1/24/2017"…  
## $ timestamp <chr> "00:45:20 27 Sep 2020", "00:45:20 27 Sep 2020", "…

# Use `lubridate` to fix dates, as they are given as characters.

## Create a plot

What I would like you to do is to calculate the average net approval rate (approve- disapprove) for each week since he got into office. I want you plot the net approval, along with its 95% confidence interval. There are various dates given for each poll, please use enddate, i.e., the date the poll ended.

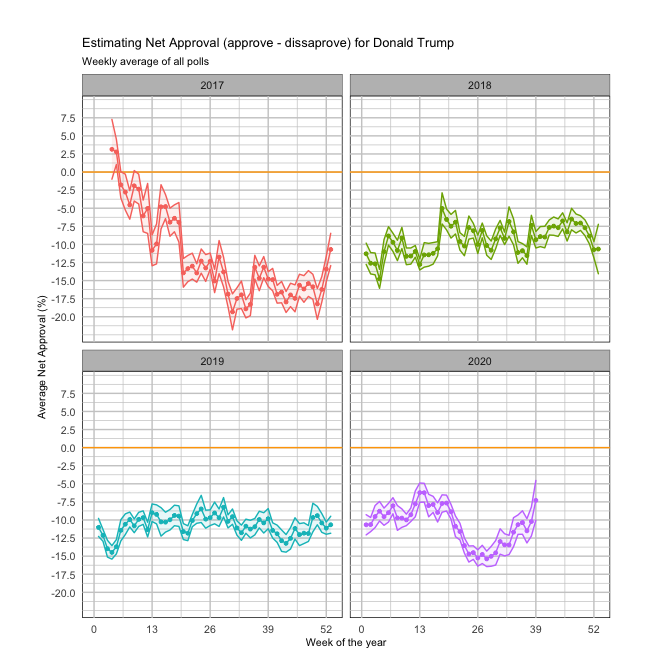
You can facet by year, and add an orange line at zero. Your plot should look like this:



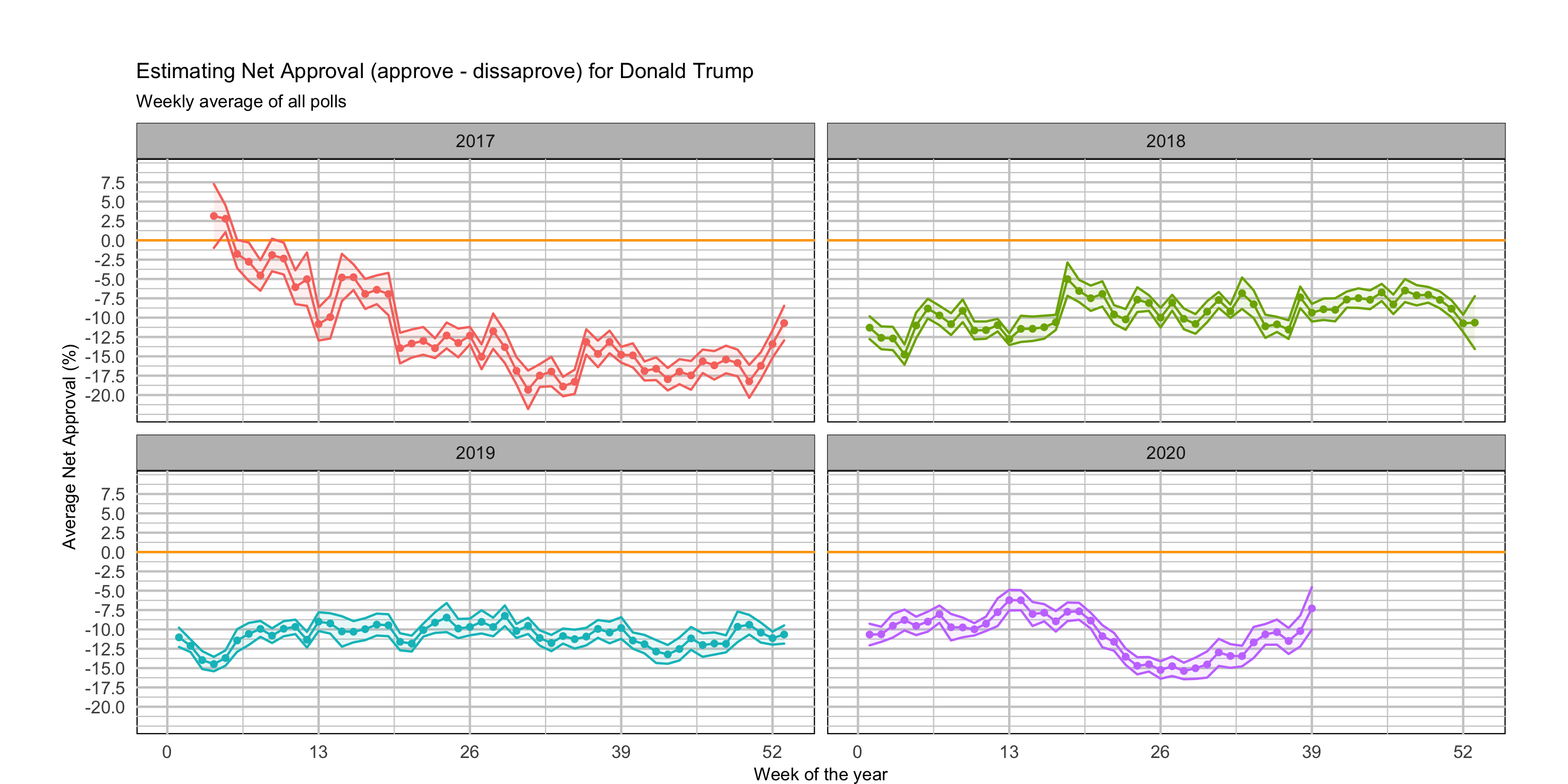
glimpse(approval\_polllist)

## Rows: 15,619  
## Columns: 22  
## $ president <chr> "Donald Trump", "Donald Trump", "Donald Trump", "…  
## $ subgroup <chr> "All polls", "All polls", "All polls", "All polls…  
## $ modeldate <chr> "9/27/2020", "9/27/2020", "9/27/2020", "9/27/2020…  
## $ startdate <chr> "1/20/2017", "1/20/2017", "1/20/2017", "1/21/2017…  
## $ enddate <chr> "1/22/2017", "1/22/2017", "1/24/2017", "1/23/2017…  
## $ pollster <chr> "Gallup", "Morning Consult", "Ipsos", "Gallup", "…  
## $ grade <chr> "B", "B/C", "B-", "B", "B-", "C+", "B+", "B", "C+…  
## $ samplesize <dbl> 1500, 1992, 1632, 1500, 1651, 1500, 1190, 1500, 1…  
## $ population <chr> "a", "rv", "a", "a", "a", "lv", "rv", "a", "lv", …  
## $ weight <dbl> 0.262, 0.680, 0.153, 0.243, 0.142, 0.200, 1.514, …  
## $ influence <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0…  
## $ approve <dbl> 45.0, 46.0, 42.1, 45.0, 42.3, 57.0, 36.0, 46.0, 5…  
## $ disapprove <dbl> 45.0, 37.0, 45.2, 46.0, 45.8, 43.0, 44.0, 45.0, 4…  
## $ adjusted\_approve <dbl> 45.7, 45.3, 43.2, 45.7, 43.4, 51.5, 37.6, 46.7, 5…  
## $ adjusted\_disapprove <dbl> 43.6, 38.3, 43.9, 44.6, 44.5, 44.5, 42.8, 43.6, 4…  
## $ multiversions <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N…  
## $ tracking <lgl> TRUE, NA, TRUE, TRUE, TRUE, TRUE, NA, TRUE, TRUE,…  
## $ url <chr> "http://www.gallup.com/poll/201617/gallup-daily-t…  
## $ poll\_id <dbl> 49253, 49249, 49426, 49262, 49425, 49266, 49260, …  
## $ question\_id <dbl> 77265, 77261, 77599, 77274, 77598, 77278, 77272, …  
## $ createddate <chr> "1/23/2017", "1/23/2017", "3/1/2017", "1/24/2017"…  
## $ timestamp <chr> "00:45:20 27 Sep 2020", "00:45:20 27 Sep 2020", "…

# Clean data set and prepare to analyse on weekly basis and across years  
approval\_polllist\_clean <- approval\_polllist %>%   
 mutate(enddate = mdy(enddate)) %>% # Use `lubridate` to fix dates, as they are given as characters.  
 mutate(week\_count = week(enddate)) %>% # get week counts  
 mutate(year = year(enddate)) %>% # get year for facetting  
 filter(subgroup == "Voters") # Keeping only "Voters" as the subgroup   
  
# Adding net rate to the data set as %  
approval\_polllist\_clean <- approval\_polllist\_clean %>%   
 mutate(net\_rate = (adjusted\_approve - adjusted\_disapprove) / (adjusted\_approve + adjusted\_disapprove) \* 100)   
  
#Calculating average net rate, SD, SE & DF to create CIs on a weekly basis  
weekly\_approval\_polllist <- approval\_polllist\_clean %>%   
 group\_by(year,week\_count) %>%   
 summarise(count =n(),  
 MEAN = mean(net\_rate),  
 SD = sd(net\_rate),   
 SE = SD/sqrt(count),   
 DF = count - 1) %>%   
 mutate(CI\_upper = MEAN + qt(.975, DF) \* SE,   
 CI\_lower = MEAN - qt(.975, DF) \* SE)  
  
# colour hex codes from: https://html-color-codes.info/colors-from-image/  
graph\_colouring <- c("#F8766D" ,"#7CAE00", "#00BFC4", "#C77CFF")  
  
#Plotting the data  
ggplot(weekly\_approval\_polllist, aes(x = week\_count, y = MEAN, color = as.factor(year))) +   
 geom\_line() +   
 facet\_wrap(~year) +   
 geom\_point(size = 1) +  
 geom\_hline(yintercept = 0, color = "orange") +   
 scale\_x\_continuous (limits = c(0, 53),  
 breaks = c(0, 13, 26, 39, 52),  
 labels = c("0", "13","26","39","52")) +   
 scale\_y\_continuous (limits=c(-22, 9),  
 breaks=c(-20, -17.5, -15, -12.5, -10, -7.5, -5, -2.5, 0, 2.5, 5, 7.5)) +  
 geom\_ribbon(aes(ymin = CI\_lower, ymax = CI\_upper, fill = as.factor(year)), alpha = .1) +  
 labs(y = "Average Net Approval (%)",   
 x = "Week of the year") +   
 ggtitle(label = "Estimating Net Approval (approve - dissaprove) for Donald Trump",   
 subtitle = "Weekly average of all polls") +  
 theme(title = element\_text(size=8),  
 axis.title = element\_text(size=8),  
 axis.text = element\_text(size=8),  
 axis.ticks = element\_blank(),  
 strip.text = element\_text(size=8),  
 panel.background = element\_rect(color="black", fill = "white"),  
 panel.border = element\_blank(),  
 strip.background = element\_rect(color="black", fill="grey", size=.25),  
 panel.grid = element\_line(color = "#CCCCCC"),  
 legend.position = "none") +  
 scale\_colour\_manual(aesthetics = "custom\_color\_palette") +  
 #coord\_fixed(ratio= 3/5, clip = "on") +   
 theme(plot.margin = unit(c(1,1,0,1), "cm"))



ggsave("plot1.png",  
 plot = last\_plot(),  
 scale = 1,  
 width = 25,  
 height = 12.5,  
 units = "cm",  
 dpi = 300,  
 limitsize = TRUE)  
  
knitr::include\_graphics(here::here("plot1.png"), error = FALSE)



## Compare Confidence Intervals

Compare the confidence intervals for week 15 (6-12 April 2020) and week 34 (17-23 August 2020). Can you explain what’s going on? One paragraph would be enough.

# Gapminder revisited

Recall the gapminder data frame from the gapminder package. That data frame contains just six columns from the larger [data in Gapminder World](https://www.gapminder.org/data/). In this part, you will join a few dataframes with more data than the ‘gapminder’ package. Specifically, you will look at data on

* Life expectancy at birth (life\_expectancy\_years.csv)
* GDP per capita in constant 2010 US$ (<https://data.worldbank.org/indicator/NY.GDP.PCAP.KD>)
* Female fertility: The number of babies per woman (<https://data.worldbank.org/indicator/SP.DYN.TFRT.IN>)
* Primary school enrollment as % of children attending primary school (<https://data.worldbank.org/indicator/SE.PRM.NENR>)
* Mortality rate, for under 5, per 1000 live births (<https://data.worldbank.org/indicator/SH.DYN.MORT>)
* HIV prevalence (adults\_with\_hiv\_percent\_age\_15\_49.csv): The estimated number of people living with HIV per 100 population of age group 15-49.

You must use the wbstats package to download data from the World Bank. The relevant World Bank indicators are SP.DYN.TFRT.IN, SE.PRM.NENR, NY.GDP.PCAP.KD, and SH.DYN.MORT

# load gapminder HIV data  
hiv <- read\_csv(here::here("data","adults\_with\_hiv\_percent\_age\_15\_49.csv"))  
life\_expectancy <- read\_csv(here::here("data","life\_expectancy\_years.csv"))  
  
# get World bank data using wbstats  
indicators <- c("SP.DYN.TFRT.IN","SE.PRM.NENR", "SH.DYN.MORT", "NY.GDP.PCAP.KD")  
  
  
library(wbstats)  
  
worldbank\_data <- wb\_data(country="countries\_only", #countries only- no aggregates like Latin America, Europe, etc.  
 indicator = indicators,   
 start\_date = 1960,   
 end\_date = 2016)  
  
# get a dataframe of information regarding countries, indicators, sources, regions, indicator topics, lending types, income levels, from the World Bank API   
countries <- wbstats::wb\_cachelist$countries

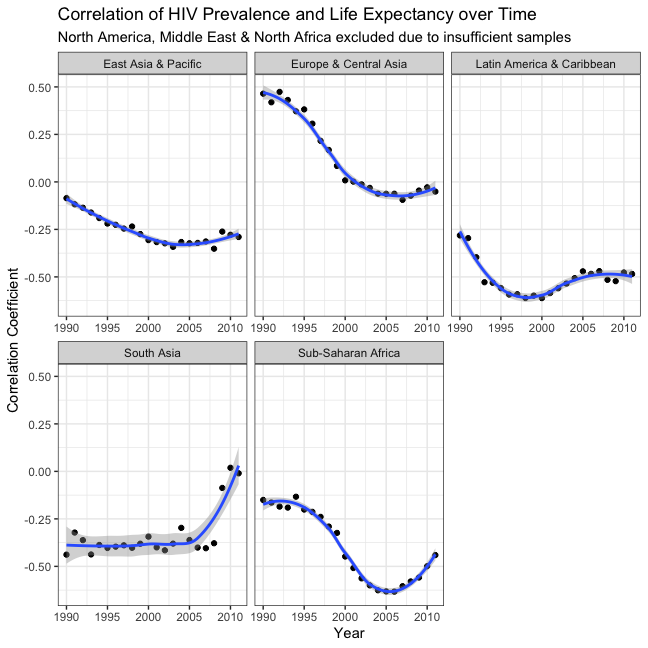
You have to join the 3 dataframes (life\_expectancy, worldbank\_data, and HIV) into one. You may need to tidy your data first and then perform [join operations](http://r4ds.had.co.nz/relational-data.html). Think about what type makes the most sense **and explain why you chose it**.

hiv\_cleaned <- hiv %>%  
 select(country, c(13:34)) %>%  
 pivot\_longer(cols = 2:23, names\_to = "Year", values\_to = "HIV\_rate\_per\_100") %>%  
 mutate(Year = as.numeric(Year))  
  
life\_expectancy\_cleaned <- life\_expectancy %>%  
 select(country, c(192:213)) %>%  
 pivot\_longer(cols = 2:23, names\_to = "Year", values\_to = "life\_expectancy") %>%  
 mutate(Year = as.numeric(Year))  
   
  
worldbank\_cleaned <- worldbank\_data %>%  
 filter(date >= 1990,  
 date <= 2011) %>%  
 rename(Year = date) %>%  
 select(c(3:8)) %>%  
 rename(gdpPerCap = NY.GDP.PCAP.KD,  
 school\_enrollment\_rate = SE.PRM.NENR,  
 mortality\_rate\_under5 = SH.DYN.MORT,  
 fertility\_rate = SP.DYN.TFRT.IN)  
  
compiled\_countries\_data <- worldbank\_cleaned %>%  
 left\_join(hiv\_cleaned, by = c("country", "Year")) %>%  
 left\_join(life\_expectancy\_cleaned, by = c("country", "Year")) %>%  
 left\_join(countries, by = "country") %>%  
 select(region, country, Year, gdpPerCap, school\_enrollment\_rate, mortality\_rate\_under5, fertility\_rate, HIV\_rate\_per\_100, life\_expectancy)

* The hiv and life\_expectancy data frames are reformatted using pivot\_longer to match the format of the worldbank\_data data frame. Its longer format is chosen because this allows multiple indicators to be recorded over time, whereas in the other two data frames only one indicator can be included. In addition, in order to allow for faceting the data by region where necessary, the region column in the countries data frame has also been merged to the new data frame.
* Moreover, the time range of all data frames are also synchronised to ensure consistency. Whilst hiv, the data frame with the shortest time span, ranges from 1979 until 2011, in the earlier years hiv data do not exist for almost all countries in the list. In fact, data for most countries were only recorded from 1990. For this reason, data in all the data frames have been filtered to only include entries between 1990 and 2011.

## What is the relationship between HIV prevalence and life expectancy? Generate a scatterplot with a smoothing line to report your results. You may find faceting useful

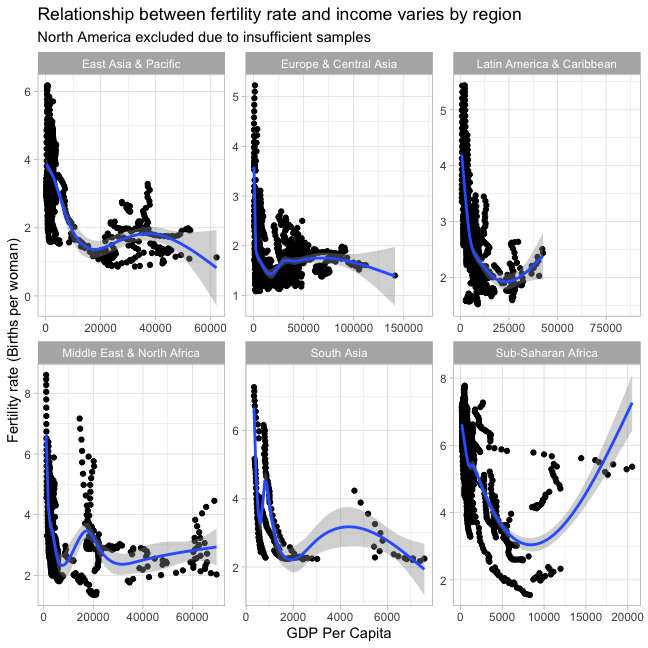
hiv\_life <- compiled\_countries\_data %>%  
 group\_by(Year, region) %>%  
 drop\_na(HIV\_rate\_per\_100, life\_expectancy) %>%  
 summarise(correlation = cor(HIV\_rate\_per\_100, life\_expectancy)) %>%  
 filter(region != c("North America", "Middle East & North Africa"))  
  
ggplot(hiv\_life, aes(x = Year, y = correlation)) +  
 geom\_point() +  
 facet\_wrap(~region, scales = "free\_x") +  
 geom\_smooth() +   
 theme\_bw() +  
 labs(title = "Correlation of HIV Prevalence and Life Expectancy over Time",  
 subtitle = "North America, Middle East & North Africa excluded due to insufficient samples",  
 y = "Correlation Coefficient",  
 x = "Year")



In most region, there has been a negative correlation between HIV prevalence and life expectancy over time. This suggests that the higher the HIV rate, the lower the life expectancy. However, this seems to be the case only for less developed regions. In Europe & Central Asia where living conditions are significantly higher, the correlation coefficient between 1990-2000 is actually positive. Since this is counter-intuitive, the only reasonable explanation would be that as living conditions increase, HIV rate becomes much less relevant in deciding life expectancy. This also explains why Sub-Saharan Africa and South Asia, where economic development was high between 2005-2011, experienced significant decreases in the absolute value of correlation coefficients.

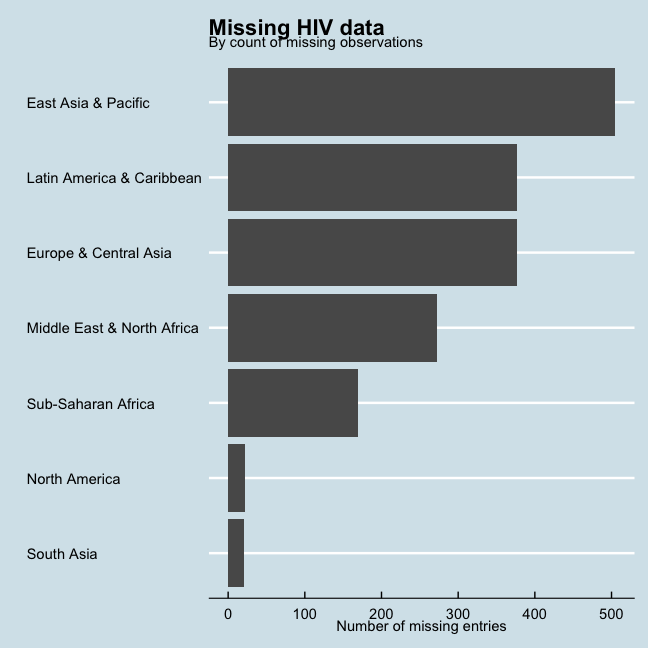
## What is the relationship between fertility rate and GDP per capita? Generate a scatterplot with a smoothing line to report your results. You may find facetting by region useful

compiled\_countries\_data %>%  
 filter(region != "North America") %>%  
 ggplot(aes(x = gdpPerCap, y = fertility\_rate)) +  
 geom\_point() +  
 facet\_wrap(~region, scales = "free") +  
 theme\_light() +  
 geom\_smooth() +  
 labs(title = "Relationship between fertility rate and income varies by region",  
 subtitle = "North America excluded due to insufficient samples",  
 x = "GDP Per Capita",  
 y = "Fertility rate (Births per woman)")

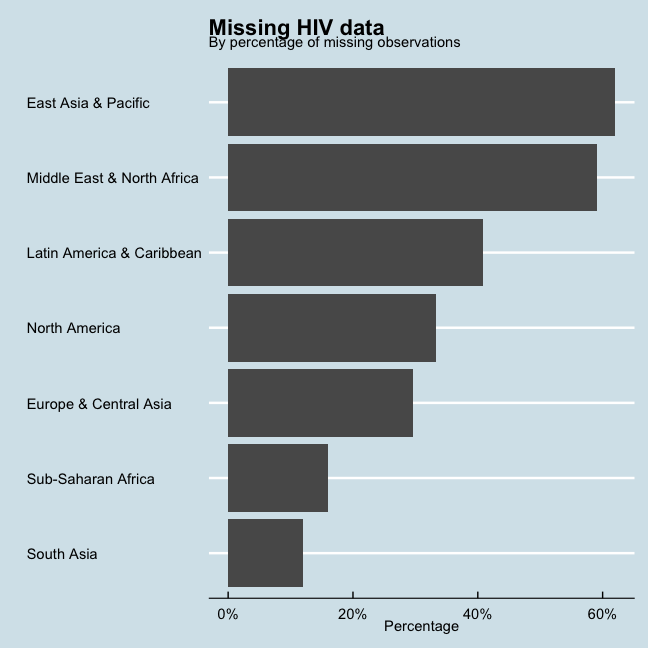
 At lower income levels, there is a consistent trend that the higher the average income, the lower the fertility rate. Conversely, at mid-income levels, birth rates actually increase with GDP per capita for most countries, although the degree by which it increases varies greatly by region. Finally, at higher income levels, correlation becomes negative again.

## Which regions have the most observations with missing HIV data? Generate a bar chart (geom\_col()), in descending order.

HIV\_missing\_data <- compiled\_countries\_data %>%  
 group\_by(region) %>%  
 summarise(missing\_entries = sum(is.na(HIV\_rate\_per\_100)))  
  
ggplot(HIV\_missing\_data, aes(x = missing\_entries, y = reorder(region, missing\_entries))) +  
 geom\_col() +  
 theme\_economist() +  
 labs(title = "Missing HIV data",  
 subtitle = "By count of missing observations",  
 y = "",  
 x = "Number of missing entries") +  
 NULL

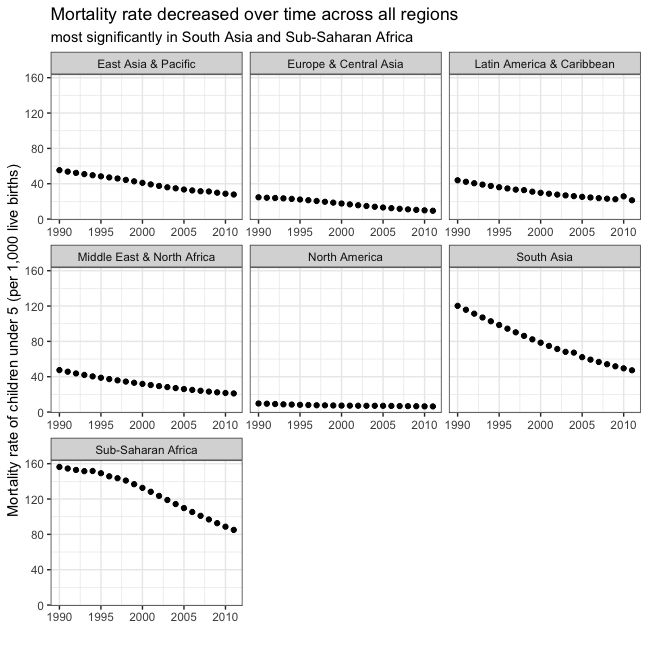


HIV\_missing\_data\_2 <- compiled\_countries\_data %>%  
 group\_by(region) %>%  
 summarise(missing\_entries = sum(is.na(HIV\_rate\_per\_100)),  
 count = n()) %>%  
 mutate(percentage\_missing = missing\_entries/count)  
  
ggplot(HIV\_missing\_data\_2, aes(x = percentage\_missing, y = reorder(region, percentage\_missing))) +  
 geom\_col() +  
 theme\_economist() +  
 scale\_x\_continuous(labels = scales::percent) +  
 labs(title = "Missing HIV data",  
 subtitle = "By percentage of missing observations",  
 y = "",  
 x = "Percentage") +  
 NULL

 East Asia & Pacific has the most observations with missing HIV data, by both count of missing entries and percentage of missing entries.

## How has mortality rate for under 5 changed by region? In each region, find the top 5 countries that have seen the greatest improvement, as well as those 5 countries where mortality rates have had the least improvement or even deterioration.

mortality\_plot <- compiled\_countries\_data %>%  
 group\_by(region, Year) %>%  
 summarise(mean\_mortality\_rate = mean(mortality\_rate\_under5, na.rm = TRUE))  
  
ggplot(mortality\_plot, aes(x = Year, y = mean\_mortality\_rate)) +  
 geom\_point() +  
 facet\_wrap(~region, scales = "free\_x") +  
 theme\_bw() +  
 labs(title = "Mortality rate decreased over time across all regions",  
 subtitle = "most significantly in South Asia and Sub-Saharan Africa",  
 y = "Mortality rate of children under 5 (per 1,000 live births)",  
 x = "") +  
 NULL



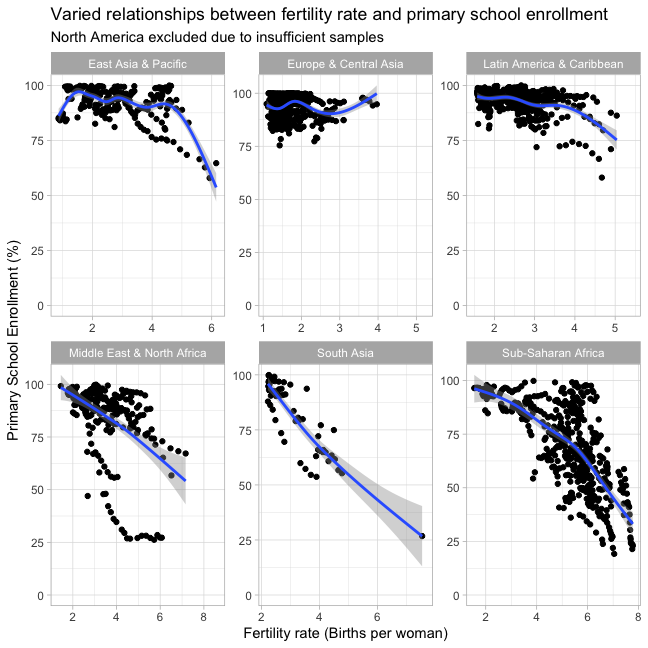
mortality\_head <- compiled\_countries\_data %>%  
 filter(Year == c(1990, 2011)) %>%  
 select(region, country, Year, mortality\_rate\_under5) %>%  
 pivot\_wider(names\_from = Year, values\_from = mortality\_rate\_under5) %>%  
 rename(start = 3,  
 end = 4) %>%  
 mutate(change = end - start) %>%  
 arrange(region, change, na.rm = TRUE) %>%  
 group\_by(region) %>%  
 slice\_head(n=5)  
  
mortality\_tail <- compiled\_countries\_data %>%  
 filter(Year == c(1990, 2011)) %>%  
 select(region, country, Year, mortality\_rate\_under5) %>%  
 pivot\_wider(names\_from = Year, values\_from = mortality\_rate\_under5) %>%  
 rename(start = 3,  
 end = 4) %>%  
 mutate(change = end - start) %>%  
 drop\_na(change) %>%  
 arrange(region, change, na.rm = TRUE) %>%  
 group\_by(region) %>%  
 slice\_tail(n=5)  
  
mortality\_top\_and\_bottom <- bind\_rows(mortality\_head, mortality\_tail) %>%  
 arrange(region, change) %>%  
 select(region, country, change) %>%  
 rename(change\_in\_mortality\_rate = change) %>%  
 drop\_na()  
   
mortality\_top\_and\_bottom

## # A tibble: 64 x 3  
## # Groups: region [7]  
## region country change\_in\_mortality\_rate  
## <chr> <chr> <dbl>  
## 1 East Asia & Pacific Timor-Leste -116.   
## 2 East Asia & Pacific Lao PDR -88.2   
## 3 East Asia & Pacific Mongolia -80.2   
## 4 East Asia & Pacific Cambodia -75.3   
## 5 East Asia & Pacific Myanmar -53.9   
## 6 East Asia & Pacific New Zealand -5.20  
## 7 East Asia & Pacific Singapore -4.9   
## 8 East Asia & Pacific Australia -4.70  
## 9 East Asia & Pacific Brunei Darussalam -3.5   
## 10 East Asia & Pacific Japan -3.10  
## # … with 54 more rows

Here, a negative change in mortality rate implies an improvement. Hence, the 5 countries that improved the most are those with the most negative changes in mortality rate, and vice versa for the least improved. The table compiled is for the 1990-2011 period.

## Is there a relationship between primary school enrollment and fertility rate?

compiled\_countries\_data %>%  
 filter(region != "North America") %>%  
 ggplot(aes(x = fertility\_rate, y = school\_enrollment\_rate)) +  
 geom\_point() +  
 facet\_wrap(~region, scales = "free") +  
 expand\_limits(y = c(0,100)) +  
 theme\_light() +  
 geom\_smooth() +  
 labs(title = "Varied relationships between fertility rate and primary school enrollment",  
 subtitle = "North America excluded due to insufficient samples",  
 x = "Fertility rate (Births per woman)",  
 y = "Primary School Enrollment (%)")

 Whilst the relationship between fertility rate and primary school enrollment varies by region, on overall there seems to be a negative relationship between the 2 variables in less developed regions (Latin America & Caribbean, Middle East & North Africa, Sub-Saharan Africa, South Asia, and countries with really high birth rates in East Asia & Pacific). The correlation, however, fades in more developed countries (Europe & Central Asia, and most of East Asia & Pacific). As with the HIV prevalence - life expectancy case, the most plausible explanation is that as living conditions increase, fertility rates become much less relevant in deciding primary school enrollment.

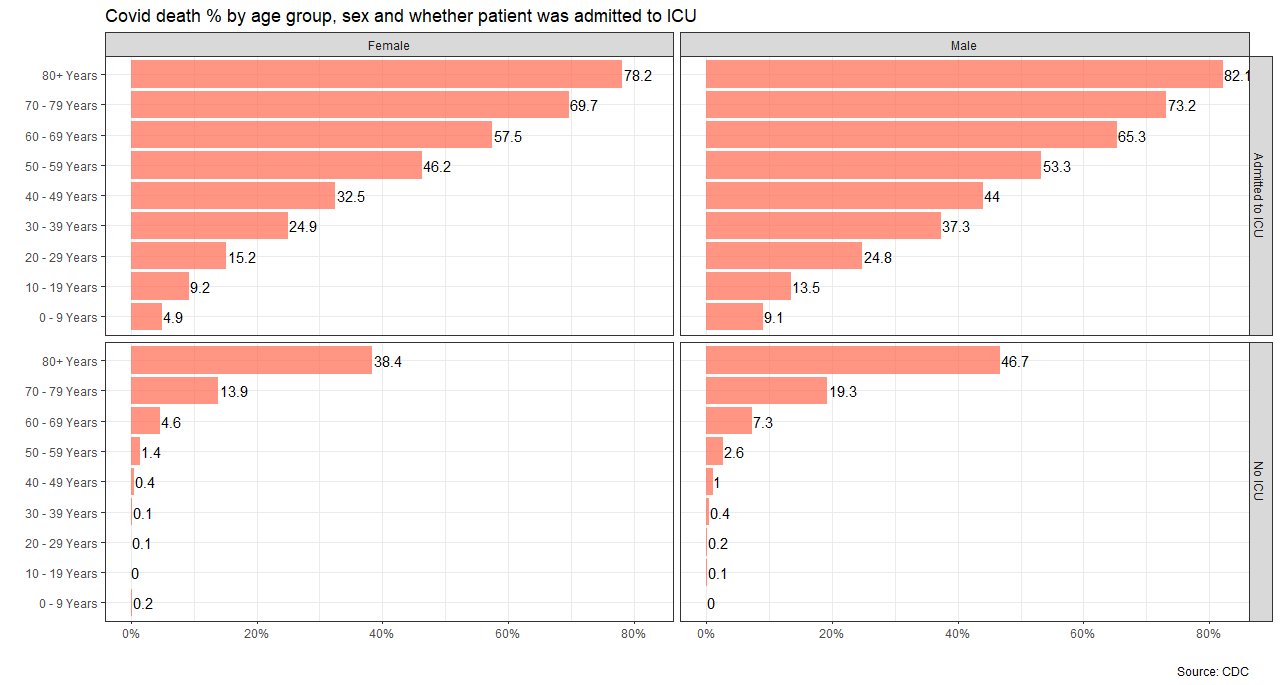
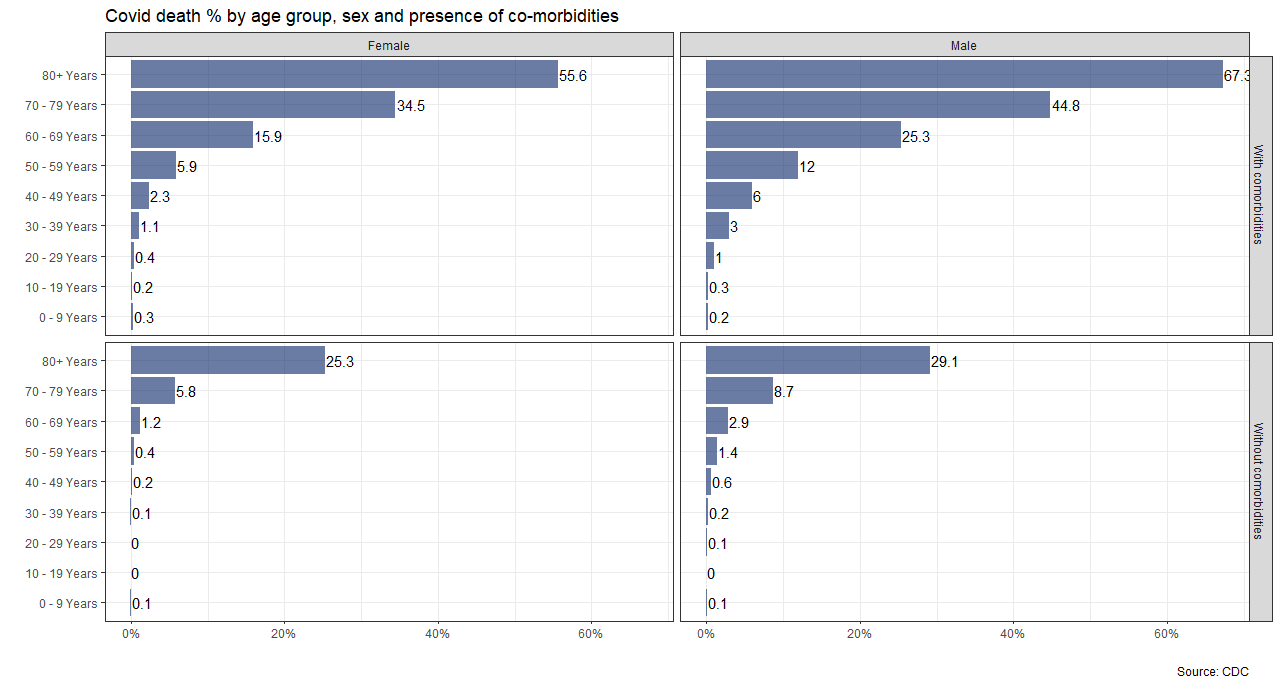
# Challenge 1: CDC COVID-19 Public Use Data

Let us revisit the [CDC Covid-19 Case Surveillance Data](https://data.cdc.gov/Case-Surveillance/COVID-19-Case-Surveillance-Public-Use-Data/vbim-akqf). There are well over 3 million entries of individual, de-identified patient data. Since this is a large file, I suggest you use vroom to load it and you keep cache=TRUE in the chunk options.

# file contains 11 variables and 3.66m rows and is well over 380Mb.   
# It will take time to download  
  
# URL link to CDC to download data  
url <- "https://data.cdc.gov/api/views/vbim-akqf/rows.csv?accessType=DOWNLOAD"  
  
covid\_data <- vroom::vroom(url)%>% # If vroom::vroom(url) doesn't work, use read\_csv(url)  
 clean\_names()

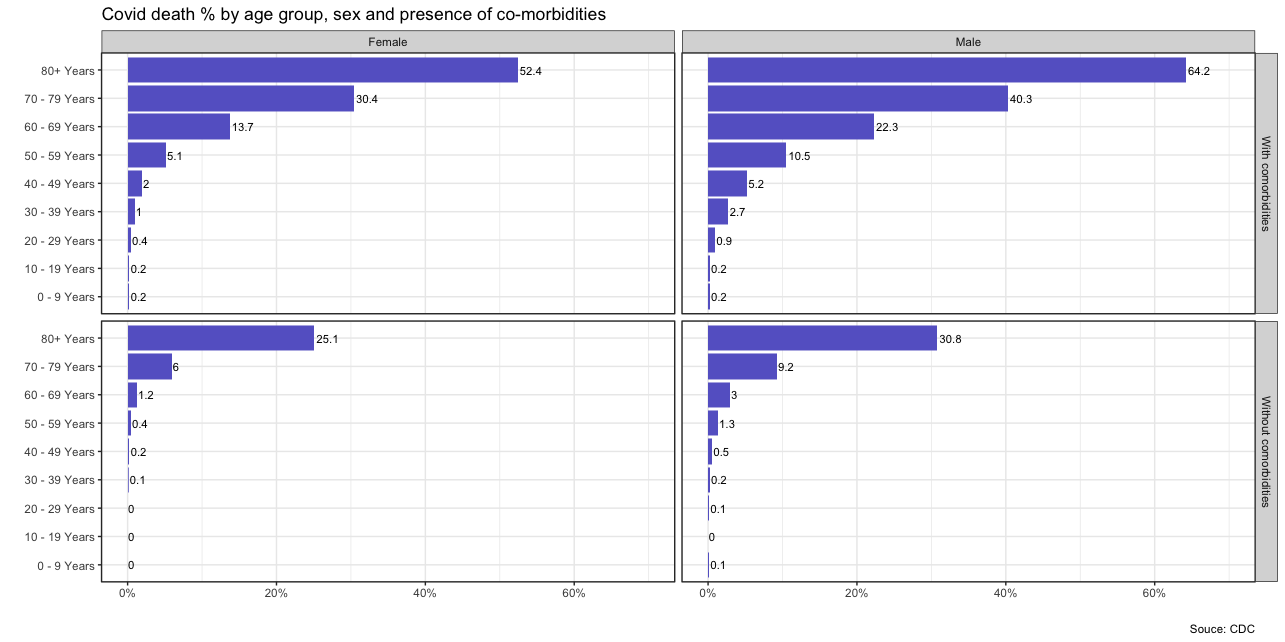
Given the data we have, I would like you to produce two graphs that show death % rate:

1. by age group, sex, and whether the patient had co-morbidities or not
2. by age group, sex, and whether the patient was admited to Intensive Care Unit (ICU) or not.

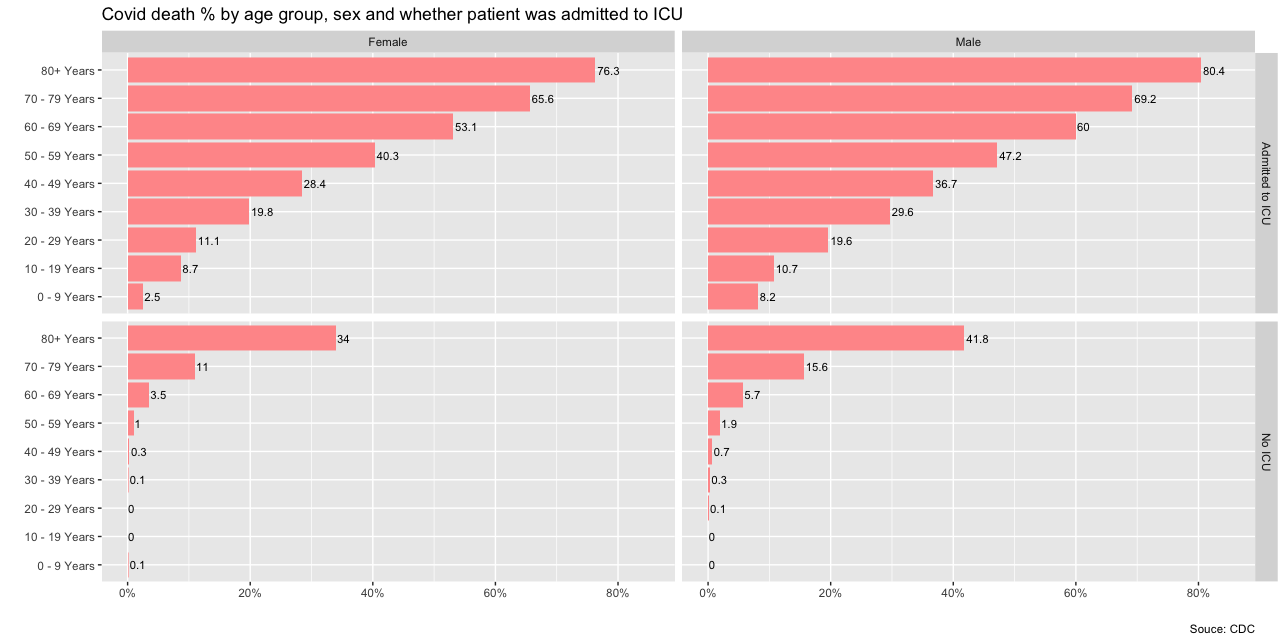


Besides the graphs, make sure your code is easy to read and understand– imagine if you revisit this six months from now. you should be able to follow what you were doing!

covid\_plot1\_data <- covid\_data %>%  
 mutate\_if(is.character, list(~na\_if(., "Unknown"))) %>%  
 mutate\_if(is.character, list(~na\_if(., "Missing"))) %>%  
 mutate(sex = na\_if(sex, "Other")) %>%  
 group\_by(age\_group, sex, medcond\_yn, death\_yn) %>%  
 drop\_na(age\_group, sex, medcond\_yn, death\_yn) %>%  
 summarise(count = n()) %>%  
 ungroup() %>%  
 group\_by(age\_group, sex, medcond\_yn) %>%  
 mutate(subgroup\_total = sum(count)) %>%  
 ungroup() %>%  
 filter(death\_yn == "Yes") %>%  
 mutate(death\_percentage = round(count/subgroup\_total, digits = 3),  
 medcond\_yn = factor(medcond\_yn, levels = c("Yes", "No"),  
 labels = c("With comorbidities", "Without comorbidities")))  
  
new <- theme\_set(theme\_bw())  
theme\_replace(plot.tag.position = c(0.95,0.01),  
 plot.tag = element\_text(size = 7))  
  
covid\_plot1 <- ggplot(covid\_plot1\_data, aes(x = death\_percentage, y = reorder(age\_group, age\_group))) +  
 geom\_col(fill = "#6666CC") +  
 facet\_grid(medcond\_yn ~ sex) +  
 coord\_cartesian(xlim = c(0, 0.7), clip = "off") +  
 scale\_x\_continuous(labels = scales::percent) +  
 theme\_set(new) +  
 labs(title = "Covid death % by age group, sex and presence of co-morbidities",  
 x = "",  
 y = "",  
 caption = "Souce: CDC") +  
 geom\_text(aes(label = death\_percentage\*100), size = 3, hjust = -0.1) +  
 NULL  
  
ggsave("covid\_plot1.png", width = 34, height = 17, units = "cm", dpi = 96)  
  
covid\_plot2\_data <- covid\_data %>%  
 mutate\_if(is.character, list(~na\_if(., "Unknown"))) %>%  
 mutate\_if(is.character, list(~na\_if(., "Missing"))) %>%  
 mutate(sex = na\_if(sex, "Other")) %>%  
 group\_by(age\_group, sex, icu\_yn, death\_yn) %>%  
 drop\_na(age\_group, sex, icu\_yn, death\_yn) %>%  
 summarise(count = n()) %>%  
 ungroup() %>%  
 group\_by(age\_group, sex, icu\_yn) %>%  
 mutate(subgroup\_total = sum(count)) %>%  
 ungroup() %>%  
 filter(death\_yn == "Yes") %>%  
 mutate(death\_percentage = round(count/subgroup\_total, digits = 3),  
 icu\_yn = factor(icu\_yn, levels = c("Yes", "No"),  
 labels = c("Admitted to ICU", "No ICU")))  
  
covid\_plot2 <- ggplot(covid\_plot2\_data, aes(x = death\_percentage, y = reorder(age\_group, age\_group))) +  
 geom\_col(fill = "#FF9999") +  
 facet\_grid(icu\_yn ~ sex) +  
 coord\_cartesian(xlim = c(0, 0.85), clip = "off") +  
 scale\_x\_continuous(labels = scales::percent) +  
 theme\_set(new) +  
 labs(title = "Covid death % by age group, sex and whether patient was admitted to ICU",  
 x = "",  
 y = "",  
 caption = "Souce: CDC") +  
 geom\_text(aes(label = death\_percentage\*100), size = 3, hjust = -0.1) +  
 NULL  
  
ggsave("covid\_plot2.png", width = 34, height = 17, units = "cm", dpi = 96)  
  
knitr::include\_graphics(here::here("covid\_plot1.png"), error = FALSE)



knitr::include\_graphics(here::here("covid\_plot2.png"), error = FALSE)



# Challenge 2: Excess rentals in TfL bike sharing

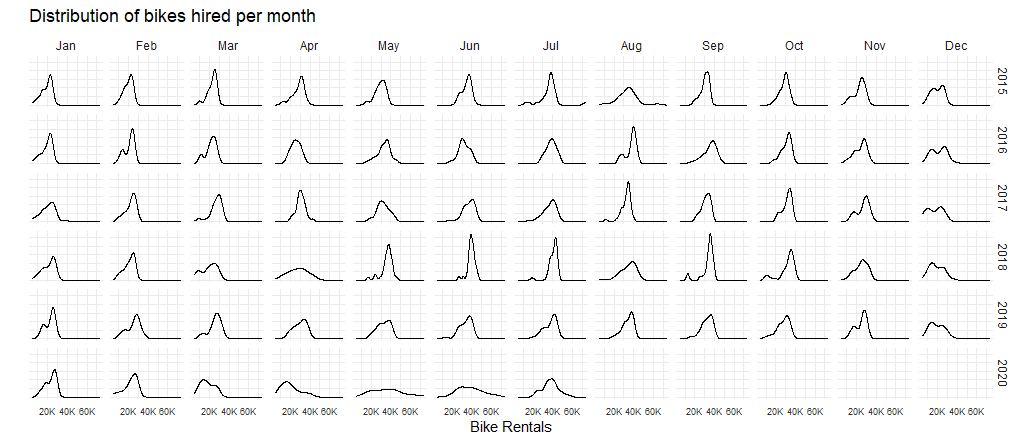
Recall the TfL data on how many bikes were hired every single day. We can get the latest data by running the following

url <- "https://data.london.gov.uk/download/number-bicycle-hires/ac29363e-e0cb-47cc-a97a-e216d900a6b0/tfl-daily-cycle-hires.xlsx"  
  
# Download TFL data to temporary file  
httr::GET(url, write\_disk(bike.temp <- tempfile(fileext = ".xlsx")))

## Response [https://airdrive-secure.s3-eu-west-1.amazonaws.com/london/dataset/number-bicycle-hires/2020-09-18T09%3A06%3A54/tfl-daily-cycle-hires.xlsx?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAJJDIMAIVZJDICKHA%2F20201003%2Feu-west-1%2Fs3%2Faws4\_request&X-Amz-Date=20201003T162202Z&X-Amz-Expires=300&X-Amz-Signature=65ab470a7dbd8afa74a806c14d834968dbf2a934162347ade59901275760ce59&X-Amz-SignedHeaders=host]  
## Date: 2020-10-03 16:22  
## Status: 200  
## Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet  
## Size: 165 kB  
## <ON DISK> /var/folders/dw/7jxg1bl50110hhppf2gnfvkm0000gn/T//Rtmpz2iQCJ/file1187633920027.xlsx

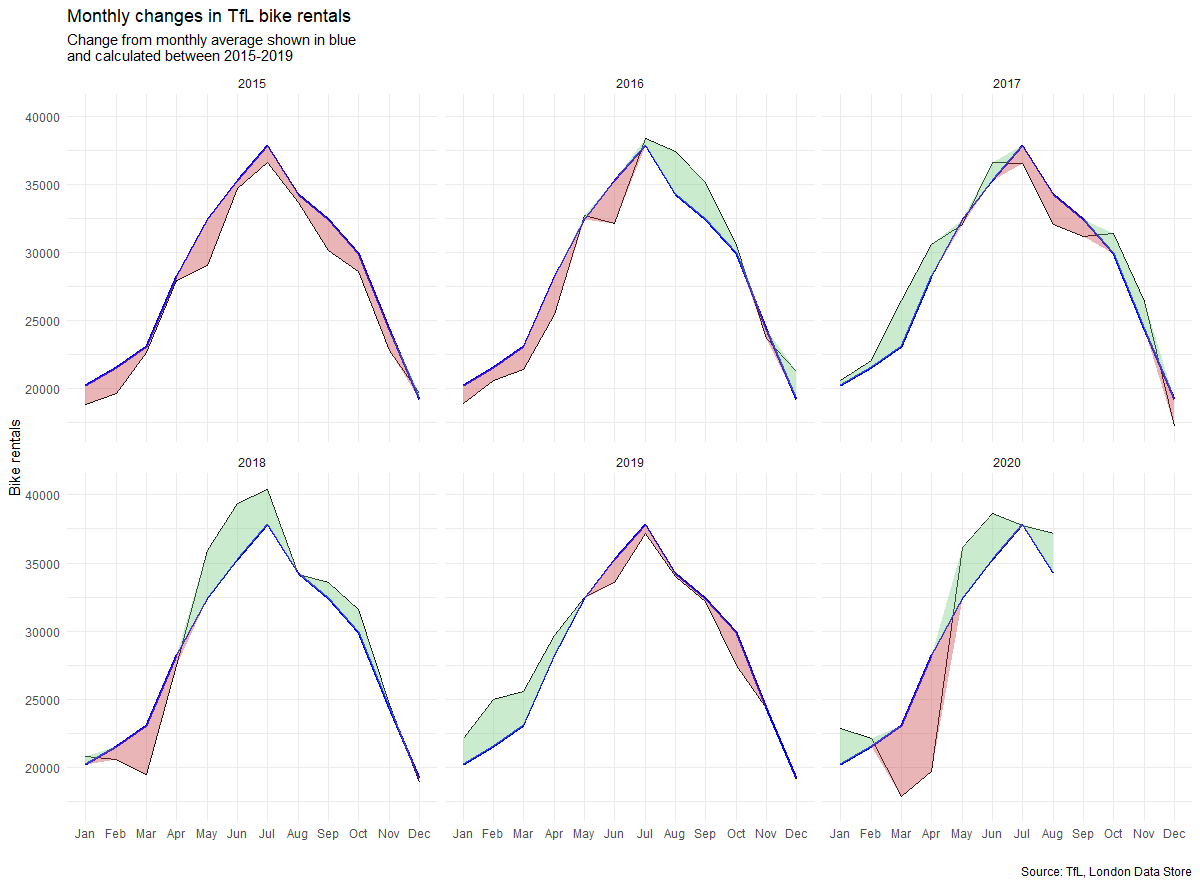
# Use read\_excel to read it as dataframe  
bike0 <- read\_excel(bike.temp,  
 sheet = "Data",  
 range = cell\_cols("A:B"))  
  
# change dates to get year, month, and week  
bike <- bike0 %>%   
 clean\_names() %>%   
 rename (bikes\_hired = number\_of\_bicycle\_hires) %>%   
 mutate (year = year(day),  
 month = lubridate::month(day, label = TRUE),  
 week = isoweek(day))

We can easily create a facet grid that plots bikes hired by month and year.

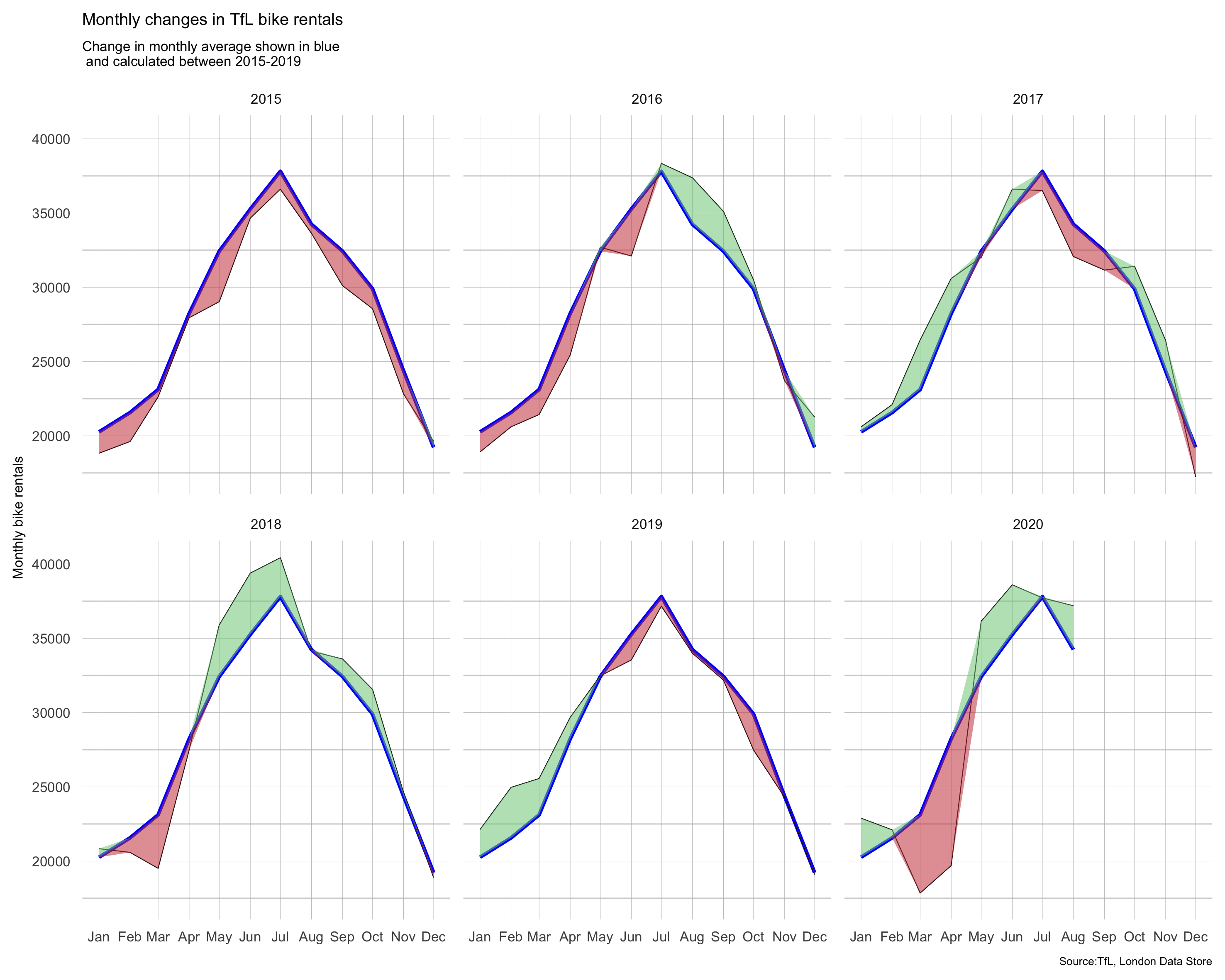


Look at May and Jun and compare 2020 with the previous years. What’s happening?

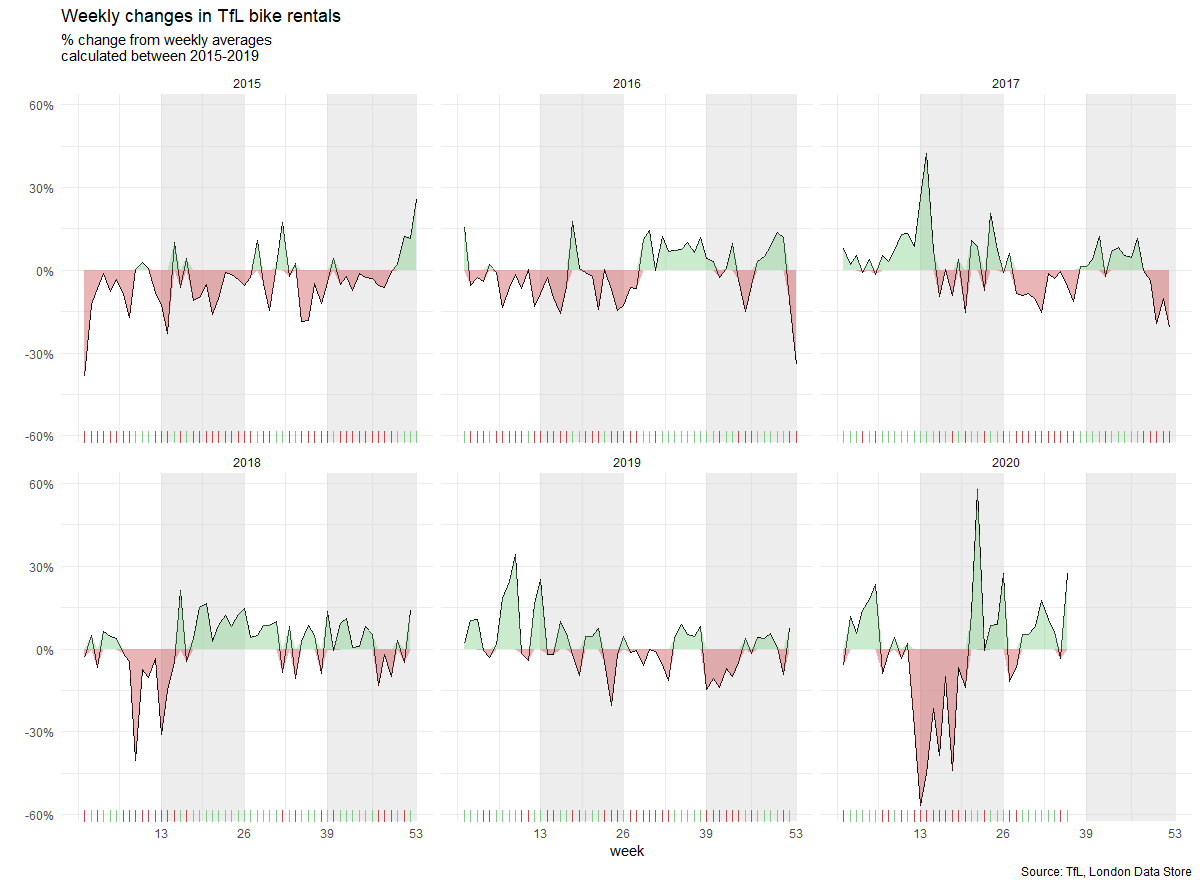
However, the challenge I want you to work on is to reproduce the following two graphs.



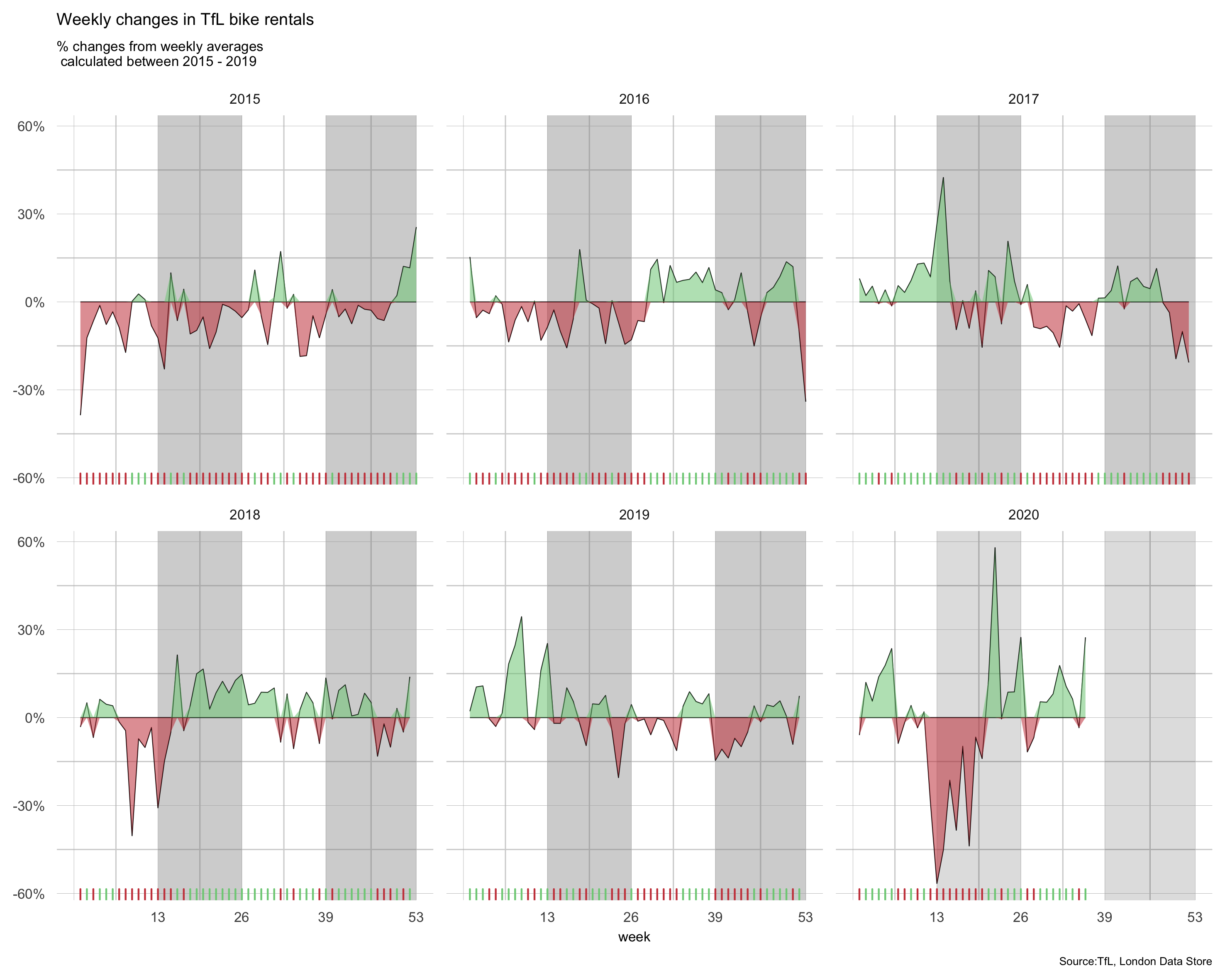
bike\_exp <- bike %>%   
 filter(year %in% c(2015:2019)) %>% # exclude 2020 to avoid outliers  
 group\_by(month) %>%   
 summarise(expected\_rentals = mean(bikes\_hired))  
  
bike\_avg <- bike %>%   
 filter(year >= "2015") %>%   
 group\_by(month, year) %>%   
 summarise(actual\_rentals = mean(bikes\_hired)) %>%   
 left\_join(y = bike\_exp, join\_by = month) %>%   
 mutate(excess\_rentals = actual\_rentals - expected\_rentals,  
 date\_month = as.Date(paste0("2015-", match(month, month.abb),"-01"),"%Y-%m-%d"))  
  
  
  
bike\_month\_surplus <- ggplot(bike\_avg, aes(x = date\_month, y = actual\_rentals)) +  
 geom\_line(aes(y = expected\_rentals), color = "blue", size = 1) +  
 geom\_line(aes(y = actual\_rentals), color = "black", size = 0.25) +  
 geom\_ribbon(aes(ymin = expected\_rentals + pmin(excess\_rentals,0), ymax = expected\_rentals, fill = "Deficit", alpha = .2)) +   
 geom\_ribbon(aes(ymin = expected\_rentals, ymax = expected\_rentals + pmax(excess\_rentals,0), fill = "Surplus", alpha = .2)) +   
 scale\_fill\_manual(values=c("#CB454A","#7DCD85"), name="Deficit vs. Surplus") +  
 facet\_wrap( ~year) +  
 theme\_bw(base\_size = 15) +  
 scale\_x\_date(date\_labels = "%b", date\_breaks ="months") +   
 theme(title = element\_text(size=8),  
 axis.text = element\_text(size=8),  
 axis.ticks = element\_blank(),  
 axis.title.x= element\_blank(),  
 panel.background = element\_rect(color="white", fill = "white"),  
 panel.border = element\_blank(),  
 panel.grid.minor.x = element\_blank(),  
 panel.grid.major = element\_line(color="grey", size = 0.1),  
 strip.background = element\_rect(color="white", fill="white"),  
 strip.text = element\_text(size=8),  
 panel.grid = element\_line(color = "#D3D3D3"),  
 legend.position = "none") +  
 labs(y = "Monthly bike rentals",  
 caption = "Source:TfL, London Data Store") +   
 ggtitle(label = "Monthly changes in TfL bike rentals",   
 subtitle = "Change in monthly average shown in blue \n and calculated between 2015-2019")   
  
ggsave("plot2.png",  
 plot = last\_plot(),  
 scale = 1,  
 width = 25,  
 height = 20,  
 units = "cm",  
 dpi = 300,  
 limitsize = TRUE)  
  
knitr::include\_graphics(here::here("plot2.png"), error = FALSE)



The second one looks at percentage changes from the expected level of weekly rentals. The two grey shaded rectangles correspond to the second (weeks 14-26) and fourth (weeks 40-52) quarters.



bike\_exp\_week <- bike %>%   
 filter(year %in% c(2015:2019)) %>% # exclude 2020 to avoid outliers  
 group\_by(week) %>%   
 summarise(expected\_rentals = mean(bikes\_hired))  
  
bike\_avg\_week <- bike %>%   
 filter(year >= "2015") %>%   
 group\_by(week, year) %>%   
 summarise(actual\_rentals = mean(bikes\_hired)) %>%   
 left\_join(y = bike\_exp\_week, join\_by = week) %>%   
 mutate(excess\_rentals = actual\_rentals - expected\_rentals,  
 perc\_change = excess\_rentals / expected\_rentals)  
  
  
bike\_week\_deviation <- ggplot(bike\_avg\_week, aes(x = week, y = perc\_change)) +  
 geom\_rect(xmin=13, xmax=26, ymin=-1, ymax=Inf,fill = "grey",alpha=0.01) +   
 geom\_rect(xmin=39, xmax=53, ymin=-1, ymax=Inf,fill = "grey",alpha=0.01) +  
 geom\_line(color = "black", size = 0.25) +  
 geom\_line(aes(y = 0), color = "black", size = 0.25) +   
 geom\_ribbon(aes(ymin = pmin(perc\_change,0), ymax = 0, fill = "Deficit", alpha = .2)) +  
 geom\_ribbon(aes(ymin = 0, ymax = pmax(perc\_change,0), fill = "Surplus", alpha = .2)) +  
 scale\_fill\_manual(values=c("#CB454A","#7DCD85"), name="Deficit vs. Surplus") +  
 geom\_rug(aes(colour = ifelse(actual\_rentals >= expected\_rentals,">=0","<0")), sides = "b")+  
 scale\_colour\_manual(values=c("#CB454A","#7DCD85"),name="Deficit vs. Surplus", guide=FALSE) +  
 facet\_wrap( ~year) +  
 theme\_bw(base\_size = 15) +  
 scale\_y\_continuous(labels = scales::percent) +  
 scale\_x\_continuous (limits = c(0, 53),  
 breaks = c(0, 13, 26, 39, 53),  
 labels = c("", "13","26","39","53")) +   
 theme(title = element\_text(size=8),  
 axis.text = element\_text(size=8),  
 axis.ticks = element\_blank(),  
 axis.title.y= element\_blank(),  
 panel.background = element\_rect(color="white", fill = "white"),  
 panel.border = element\_blank(),  
 panel.grid.major = element\_line(color="grey", size = 0.1),  
 strip.background = element\_rect(color="white", fill="white"),  
 strip.text = element\_text(size=8),  
 panel.grid = element\_line(color = "#D3D3D3"),  
 legend.position = "none") +  
 labs(x = "week",  
 caption = "Source:TfL, London Data Store") +  
 ggtitle(label = "Weekly changes in TfL bike rentals",  
 subtitle = "% changes from weekly averages \n calculated between 2015 - 2019")  
  
ggsave("plot3.png",  
 plot = last\_plot(),  
 scale = 1,  
 width = 25,  
 height = 20,  
 units = "cm",  
 dpi = 300,  
 limitsize = TRUE)  
  
knitr::include\_graphics(here::here("plot3.png"), error = FALSE)



For both of these graphs, you have to calculate the expected number of rentals per week or month between 2015-2019 and then, see how each week/month of 2020 compares to the expected rentals. Think of the calculation excess\_rentals = actual\_rentals - expected\_rentals.

Should you use the mean or the median to calculate your expected rentals? Why?

In creating your plots, you may find these links useful:

* <https://ggplot2.tidyverse.org/reference/geom_ribbon.html>
* <https://ggplot2.tidyverse.org/reference/geom_tile.html>
* <https://ggplot2.tidyverse.org/reference/geom_rug.html>

# Deliverables

As usual, there is a lot of explanatory text, comments, etc. You do not need these, so delete them and produce a stand-alone document that you could share with someone. Knit the edited and completed R Markdown file as an HTML document (use the “Knit” button at the top of the script editor window) and upload it to Canvas.

# Details

* Who did you collaborate with: Leif Beckers, Dung Tran, Salman Abdullah, Andjela Bozinovic, Xiwen Wang
* Approximately how much time did you spend on this problem set: 15 h
* What, if anything, gave you the most trouble: getting the graph to look closer to actual picture (re-scaling, etc)

**Please seek out help when you need it,** and remember the [15-minute rule](https://mam2021.netlify.app/syllabus/#the-15-minute-rule). You know enough R (and have enough examples of code from class and your readings) to be able to do this. If you get stuck, ask for help from others, post a question on Slack– and remember that I am here to help too!

As a true test to yourself, do you understand the code you submitted and are you able to explain it to someone else?

# Rubric

Check minus (1/5): Displays minimal effort. Doesn’t complete all components. Code is poorly written and not documented. Uses the same type of plot for each graph, or doesn’t use plots appropriate for the variables being analyzed.

Check (3/5): Solid effort. Hits all the elements. No clear mistakes. Easy to follow (both the code and the output).

Check plus (5/5): Finished all components of the assignment correctly and addressed both challenges. Code is well-documented (both self-documented and with additional comments as necessary). Used tidyverse, instead of base R. Graphs and tables are properly labelled. Analysis is clear and easy to follow, either because graphs are labeled clearly or you’ve written additional text to describe how you interpret the output.