Final Reflection STA 418/518

Lindsay Bandyk

12/01/2021

This is my Final reflection and I first off wanted to give you information regarding the project we have been working on all year. It is me and Jenn Hwang’s (Jung-Hye Hwang) project. We have been working on the project together and did most of the coding cohesively. The URL for our project it is as follows!! <https://github.com/Snowballtheblackcat/LBJH_distill.git>! In our project we explored all different types of new packages in R, though this was exciting it also was really interesting to see how each package actually made it possible for us to run the code. “I can explore new functions or packages and implement them into analyses.” If you would like to see the code for our graphs it is in graphs.Rmd. Our website can be found by the following link, however it only works with the login through gvsu format we found. <https://rstudio.gvsu.edu/s/53e3bdf49a46c0413cc6e/rmd_output/1/graphs.html#stem-majors> <https://rstudio.gvsu.edu/s/53e3bdf49a46c0413cc6e/rmd_output/1/graphs.html#stem-majors> , <https://rstudio.gvsu.edu/s/53e3bdf49a46c0413cc6e/rmd_output/1/graphs.html#vaccination-data> <https://rstudio.gvsu.edu/s/53e3bdf49a46c0413cc6e/rmd_output/2/graphs.html#leaflet-vaccination-graph-with-longitudes-and-latitudes>

These links take you to specific parts of the project I completed and believe are worth mentioning. One includes pivoting data, which is a very important skill, one includes mutating and cleaning the data to get rid of outliers, one shows the high level of skill with visual graphs that clearly represent our purpose without needing explanation. The last graph in our project actually is an attempt at a leaflet map and we wanted to make it more interactive, unfortunately this didnt work out but we were able to put the cursor over the graph and have it show specific vaccination numbers for those continents. Our project already covers most of the learning objectives for the semester, as I mentioned we cleaned the data, made tables of numerical summaries, wrote extremely clear and efficient code, and made many simulations on randomization based experiments.

The work I have done in the latter half of the class greatly exceeds the previous work I have done! I have gone back through and completed all of the previous activities beginning with data summarization! I have the first three activities and my portraying of knowledge in the midterm which you can go back to if needed to see my initiala progress. I have really honed into the course objectives which include:  
**import, manage,and clean data, create graphical displays and numerical summaries of data for exploratory analysis and presentations, write R programs for simulations from probability models and randomization based experiments, used source documentation and I can identify and correct common errors in R programs and write clear, efficient , and well documented R programs.**

Through this summarization activity I learned how to create code chunks and summarize them based on the mean/median/etc. I also learned how to group data by specific categories in this activity specifically. I did it by majors and then also made it so we are able to have it alphabetically in order. I was able to summarize, mutate, and arrange the data based on women share in the majors. Here is my github url for the data summarization from activity 5 <https://github.com/lindsaybandyk/activity05-data-summarization.git> . This is a perfect representation of the learning object 1 and 2. I had to import data in and also work with the numerical summaries. I also worked with pivoting the data, this can be seen on my final project to make the data more organized and easier to understand! An example of this is shown at the end of this document with the code and the output for pivoting data on mortality rate of covid 19 in different continents. I also have a URL to my finished pivot activity 6 <https://github.com/lindsaybandyk/activity06-pivoting.git> . By pivoting the data we can create rows for each month and individual columns for continents and number of deaths. It makes it much easier to interpret the data. We then learned how to combine datasets in R using either the bind function or the join function. We use the bind\_\* when we have similar structured datasets and the \*\_join function when joining data sets of different structures. We also did conditioning where we used the case\_when function which is directly related to the “if” and “else” statements. An example of the case\_when would be using a condition so if someone has let’s say cancer, case\_when(lung\_cancer ~ output-1, breast\_cancer ~ output -2, brain\_cancer ~ output -3, TRUE ~ output ~ cancer). In the joint activity we grouped by data and summarized the number of locations to find the least number of denny’s or la quinta’s in certain areas across the US. We can also look at location and area so joining the states and then the area as well to find how many of each restaurant there was per 1000 miles^2. We also joined the two data sets together for each establishment by bind\_rows function. It also made me want to try to use the map function for our project with covid around the world! The URL from my join activity is here <https://github.com/lindsaybandyk/activity07-joins.git> . The next activity we did was joined with data summarization when finding the top charts for the music billboards. We did a random sample of 200 rows and arranged them with titles and mutated them. This was more a review activity with data and being able to find patterns.. My URL is as follows:<https://github.com/lindsaybandyk/activity08-special-variables.git> . Activity 9 focused on functions, such as calculating the range, min, max, and of course means and median. We can utilize these functions to find their quartile ranges and their 5 number summaries! The URL is attached here from my completed activity. <https://github.com/lindsaybandyk/activity09-functions-intro.git> . The next activity was my personal favorite beside the data visualization! It was an iteration! It was the repetition of a process to get an outcome so it copies the data! I also love squid game so this activity was actually fun!The link to my completed activity can be found here: <https://github.com/lindsaybandyk/activity10-iteration.git> I can also do the first option as well if you are looking for more completed data for this activity. The next thing we worked on was simulation of data which is more with creating different sample sizes and using tibble to generate a normal distribution! We did this in our project when we found the std deviation of vaccinated individuals in each continent (seen below) . I will also attach my activity here that shows my understanding and completion of simulating data! <https://github.com/lindsaybandyk/activity11-simulation.git> . Activity 12 focused on bootstrapping which also used some mathematical calculations! Bootstrapping resamples a single dataset to create many simulated samples! We used std error, Confidence interval, and hypothesis testing. I also just finished the challenge portion of the activities so hopefully that also shows a deep understanding of CI and bootstrapping! <https://github.com/lindsaybandyk/activity12-bootstrap.git> . Working with github this semester wasn’t always a breeze but being able to embed these URLs from github and committ and push properly proves I “I can identify and correct common collaboration errors when working with Git/GitHub”. The final activity that we worked on this semester outside of the project where we had lots of freedom in our focuses was shiny apps! Shiny apps were less of an activity and more of an adventure in learning about applying statistics and R to applications! I think of shiny apps as more of a spreadsheet where there are values and formulas that are evaluated based on other sections of the application. Where if one cell changes the values of the other cells also change. It makes it so your apps will be in a controlled environment and private instead of on the web while creating them! There were many different folders in the activity so I did only a couple of them to get a good idea of what shiny apps were and how to create them! I didn’t love the level of difficulty that came with the shiny apps but I still proceeded to try my best and attempt to make them.

^^BELOW I have code and graphical examples of all of the learning objectives and subset learning objectives being completed, outside of our own projects, activities, or preparations. ^^

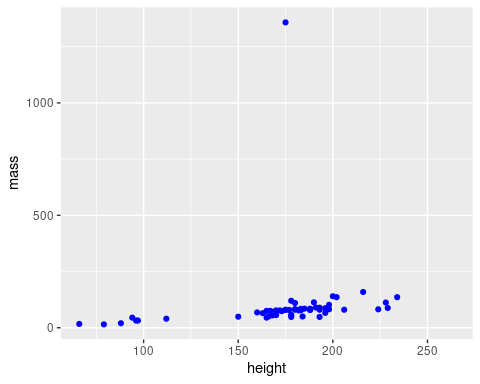
I can explore new functions or packages and implement them into analyses.

library(ggplot2)  
library(tidyverse)  
library(ggridges)  
library(readr)  
library(forcats)  
library(ggrepel)  
library(dplyr)  
library(tidyr)

I can write a function that accomplishes a common analysis task.

data=starwars  
ggplot(data= starwars, mapping= aes(x=height , y= mass)) + geom\_point(colour= "blue")

## Warning: Removed 28 rows containing missing values (geom\_point).



starwarss <- filter(starwars, name != "Jabba Desilijic Tiure")

I can apply a function to groupings within a data source.

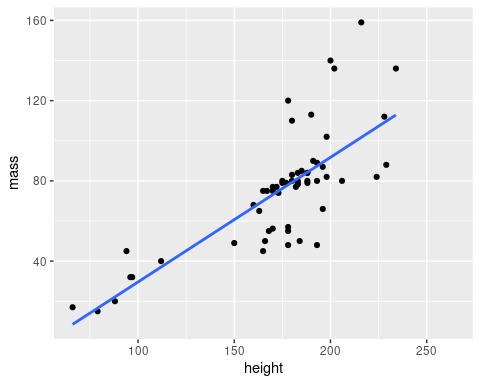
I can fit a regression model for descriptive analysis.

ggplot(starwarss, aes(height,mass)) +geom\_point(colour="black") +geom\_smooth(method=lm, se= FALSE)

## `geom\_smooth()` using formula 'y ~ x'

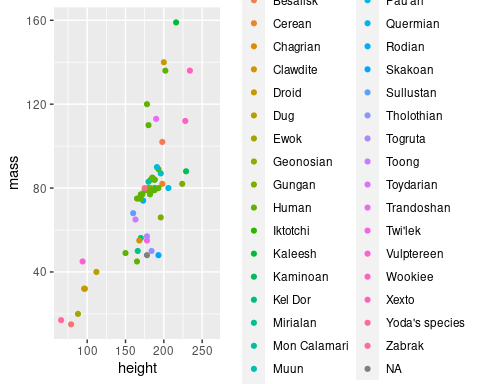
## Warning: Removed 28 rows containing non-finite values (stat\_smooth).

## Warning: Removed 28 rows containing missing values (geom\_point).

 I can implement resampling methods to make conclusions about data.

ggplot(starwarss, aes(height, mass, colour=species)) + geom\_point()

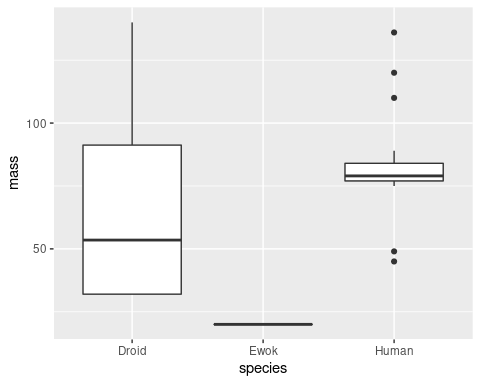
## Warning: Removed 28 rows containing missing values (geom\_point).



lib<- starwars %>% subset(species == "Human" | species == "Droid" | species == "Ewok")

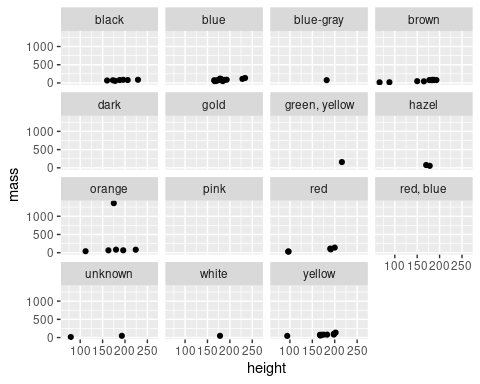
ggplot (lib, aes(species,mass)) + geom\_boxplot()

## Warning: Removed 15 rows containing non-finite values (stat\_boxplot).

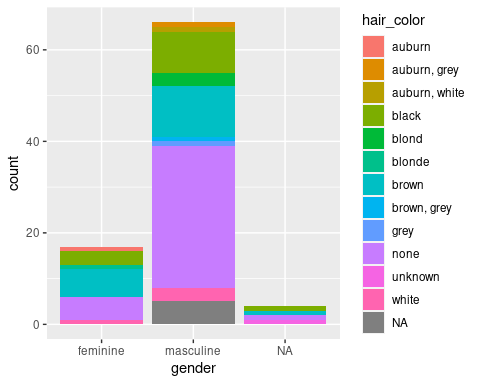


ggplot(data=starwars, mapping= aes(x=height, y=mass)) + geom\_point() + facet\_wrap(~eye\_color)

## Warning: Removed 28 rows containing missing values (geom\_point).

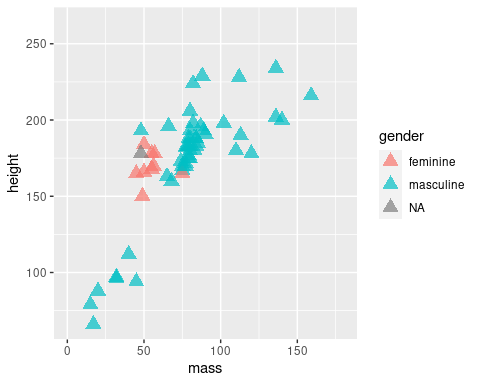


ggplot(data = starwars, mapping = aes(x=gender, fill = hair\_color)) + geom\_bar()



ggplot(data=starwarss) + geom\_point(mapping = aes(x=mass, y= height, colour= gender), alpha= 0.7, shape= 17 , size =4) + coord\_cartesian(xlim=c(0,180))

## Warning: Removed 28 rows containing missing values (geom\_point).



starwars %>% filter (height <100)

## # A tibble: 7 x 14  
## name height mass hair\_color skin\_color eye\_color birth\_year sex gender  
## <chr> <int> <dbl> <chr> <chr> <chr> <dbl> <chr> <chr>   
## 1 R2-D2 96 32 <NA> white, bl… red 33 none mascu…  
## 2 R5-D4 97 32 <NA> white, red red NA none mascu…  
## 3 Yoda 66 17 white green brown 896 male mascu…  
## 4 Wicket S… 88 20 brown brown brown 8 male mascu…  
## 5 Dud Bolt 94 45 none blue, grey yellow NA male mascu…  
## 6 Ratts Ty… 79 15 none grey, blue unknown NA male mascu…  
## 7 R4-P17 96 NA none silver, r… red, blue NA none femin…  
## # … with 5 more variables: homeworld <chr>, species <chr>, films <list>,  
## # vehicles <list>, starships <list>

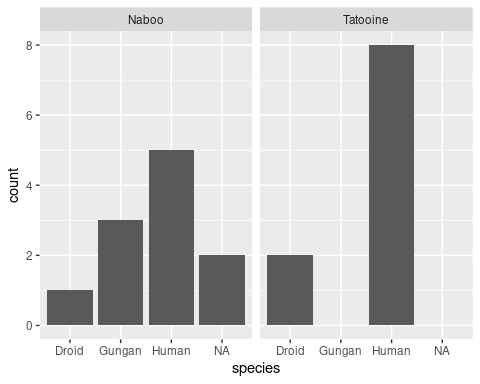
starwars %>% filter(eye\_color == "red") %>% select(name)

## # A tibble: 5 x 1  
## name   
## <chr>   
## 1 R2-D2   
## 2 R5-D4   
## 3 IG-88   
## 4 Bossk   
## 5 Nute Gunray

starwars %>%  
 mutate(height\_plus\_mass = height + mass) %>%  
 select(name, height, mass, height\_plus\_mass)

## # A tibble: 87 x 4  
## name height mass height\_plus\_mass  
## <chr> <int> <dbl> <dbl>  
## 1 Luke Skywalker 172 77 249  
## 2 C-3PO 167 75 242  
## 3 R2-D2 96 32 128  
## 4 Darth Vader 202 136 338  
## 5 Leia Organa 150 49 199  
## 6 Owen Lars 178 120 298  
## 7 Beru Whitesun lars 165 75 240  
## 8 R5-D4 97 32 129  
## 9 Biggs Darklighter 183 84 267  
## 10 Obi-Wan Kenobi 182 77 259  
## # … with 77 more rows

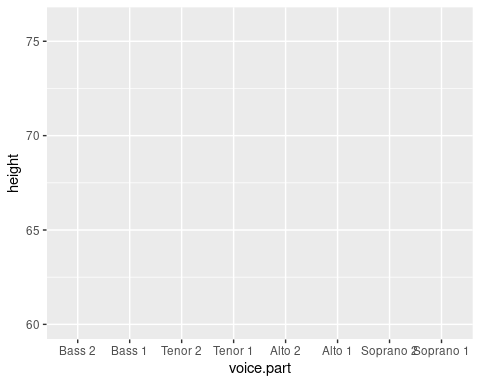
starwars %>%  
 filter(homeworld %in% c("Naboo", "Tatooine")) %>%  
 ggplot(aes(species)) +  
 geom\_bar() +  
 facet\_grid(. ~ homeworld)



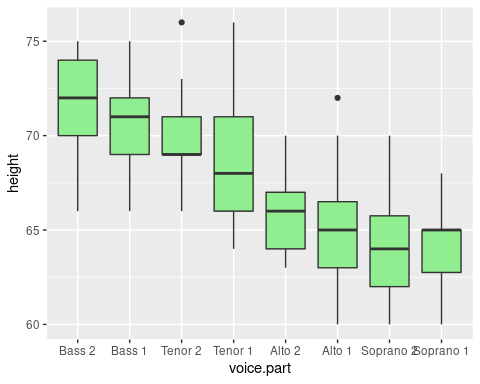
data=USArrests

data(singer, package='lattice')

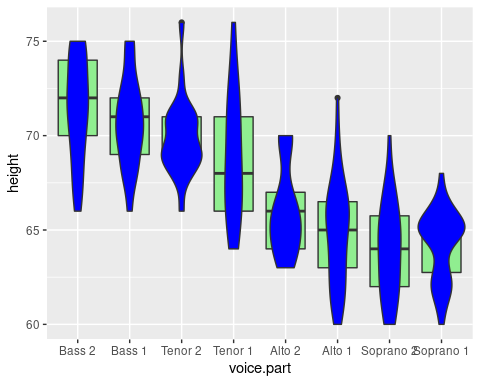
lm <- ggplot(singer, aes(x=voice.part, y=height))  
  
lm



lm <- lm + geom\_boxplot(fill="lightgreen")  
lm



lm <- lm + geom\_violin(fill="blue")  
lm

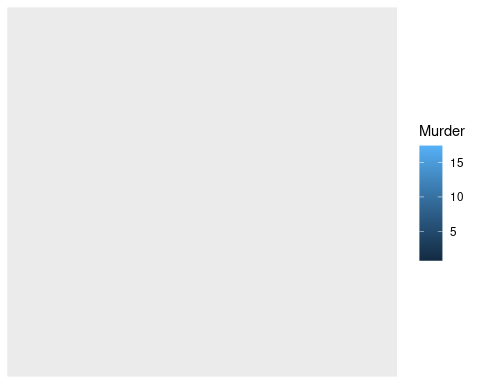


data("USArrests")  
crimes <- data.frame(state=tolower(rownames(USArrests)), USArrests)

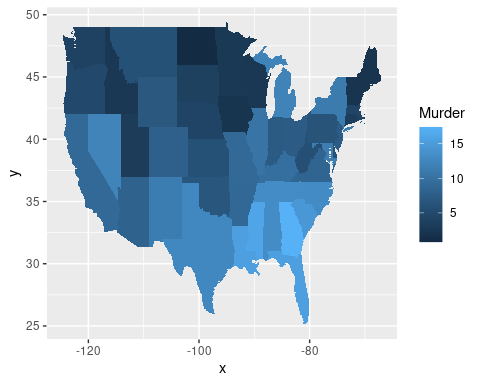
gg <- ggplot(crimes, aes(map\_id=state, fill=Murder))  
gg



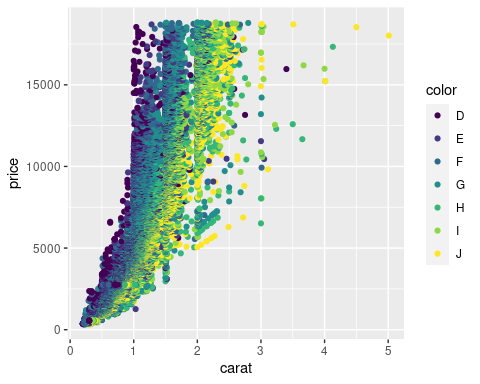
gg <- gg + geom\_map(map=map\_data("state"))  
gg



gg <- gg + expand\_limits(x=map\_data("state")$long, y=map\_data("state")$lat)  
gg



diamond.plot <- ggplot(data=diamonds, aes(x=carat, y=price, colour = color))  
diamond.plot + geom\_point()



birthwt <- as\_tibble(MASS::birthwt)  
  
birthwt <- birthwt %>%  
 rename(birthwt.below.2500 = low,   
 mother.age = age,  
 mother.weight = lwt,  
 mother.smokes = smoke,  
 previous.prem.labor = ptl,  
 hypertension = ht,  
 uterine.irr = ui,  
 physician.visits = ftv,  
 birthwt.grams = bwt)  
  
# Assign better labels to categorical variables  
birthwt <- birthwt %>%  
 mutate(race = recode\_factor(race, `1` = "white", `2` = "black", `3` = "other")) %>%  
 mutate\_at(c("mother.smokes", "hypertension", "uterine.irr", "birthwt.below.2500"),  
 ~ recode\_factor(.x, `0` = "no", `1` = "yes"))

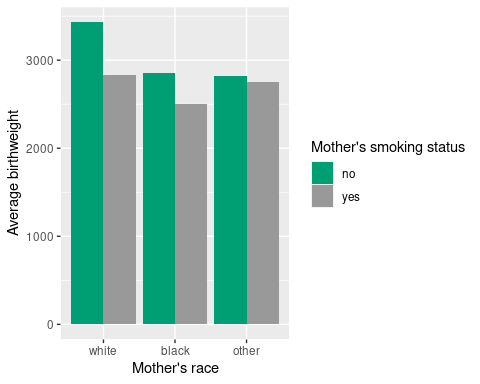
bwt.summary <- birthwt %>%  
 group\_by(race, mother.smokes) %>%  
 summarize(mean.birthwt = round(mean(birthwt.grams), 0))

## `summarise()` has grouped output by 'race'. You can override using the `.groups` argument.

bwt.summary

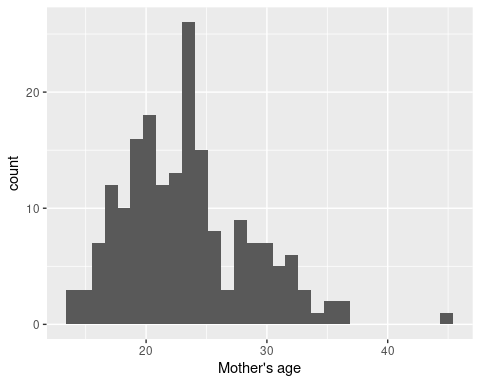
## # A tibble: 6 x 3  
## # Groups: race [3]  
## race mother.smokes mean.birthwt  
## <fct> <fct> <dbl>  
## 1 white no 3429  
## 2 white yes 2827  
## 3 black no 2854  
## 4 black yes 2504  
## 5 other no 2816  
## 6 other yes 2757

p.bwt <- ggplot(data = bwt.summary,   
 aes(y = mean.birthwt, x = race, fill = mother.smokes))  
  
# Pick colors for the bars  
bwt.colors <- c("#009E73", "#999999")  
  
# Display barchart  
p.bwt + geom\_bar(stat = "identity", position = "dodge") +  
 ylab("Average birthweight") +   
 xlab("Mother's race") +  
 guides(fill = guide\_legend(title = "Mother's smoking status")) +   
 scale\_fill\_manual(values=bwt.colors)



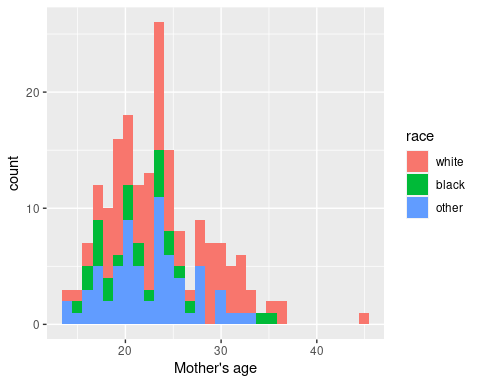
base.plot <- ggplot(birthwt, aes(x = mother.age)) +  
 xlab("Mother's age")   
base.plot + geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

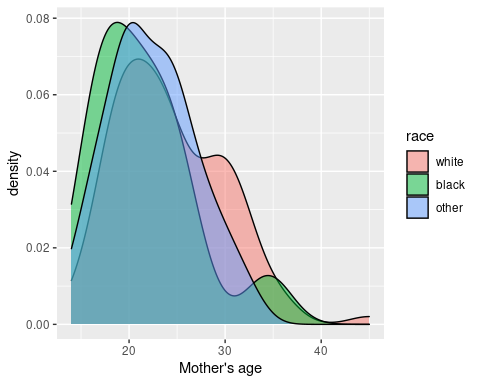


base.plot + geom\_histogram(aes(fill = race))

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



base.plot + geom\_density(aes(fill = race), alpha = 0.5)



ggplot(birthwt, aes(x=mother.age, y=birthwt.grams, shape=mother.smokes, color=mother.smokes)) +   
 geom\_point() + # Adds points (scatterplot)  
 geom\_smooth(method = "lm") + # Adds regression lines  
 ylab("Birth Weight (grams)") + # Changes y-axis label  
 xlab("Mother's Age (years)") + # Changes x-axis label  
 ggtitle("Birth Weight in relation to Mother's Age")

## `geom\_smooth()` using formula 'y ~ x'

