In Class Exam

Connor Phillips

April 11, 2024

Instructions

- **You may use an 8.5 by 11 handwritten note sheet and the help in R during the exam.**
- **You are allowed to use our textbook.**
- **You may not refer to homework assignments, google, chat gpt or other outside resources.**
- **You may not share this exam with anyone. Any attempt to do so will lead to an automatic zero in the class.**
- **Change the header information within the RMD to contain your own name. (5 points)**
- **Answer all exercise prompts within the RMD. All code must be shown**
- **Make sure to show appropriate output. Think about what I need to see to grade your solution.**
- **Place answers into the blank R chunks given for each required response.**
- **Compile the RMD into a PDF when finished.(5 points)**
- **Ensure all code is visible within the PDF.**
- **Submit the PDF through our bblearn portal.**

1. (10 pts) Look at the exam questions and make sure you have installed the packages you will need (like tidyverse and readxl). Write the library commands you will need for the exam in this R chunk. Suppress warning messages and start up messages.

```
library(tidyverse)
library(readr)
library(readxl)
library(ggplot2)
library(stringr)
library(lubridate)
```

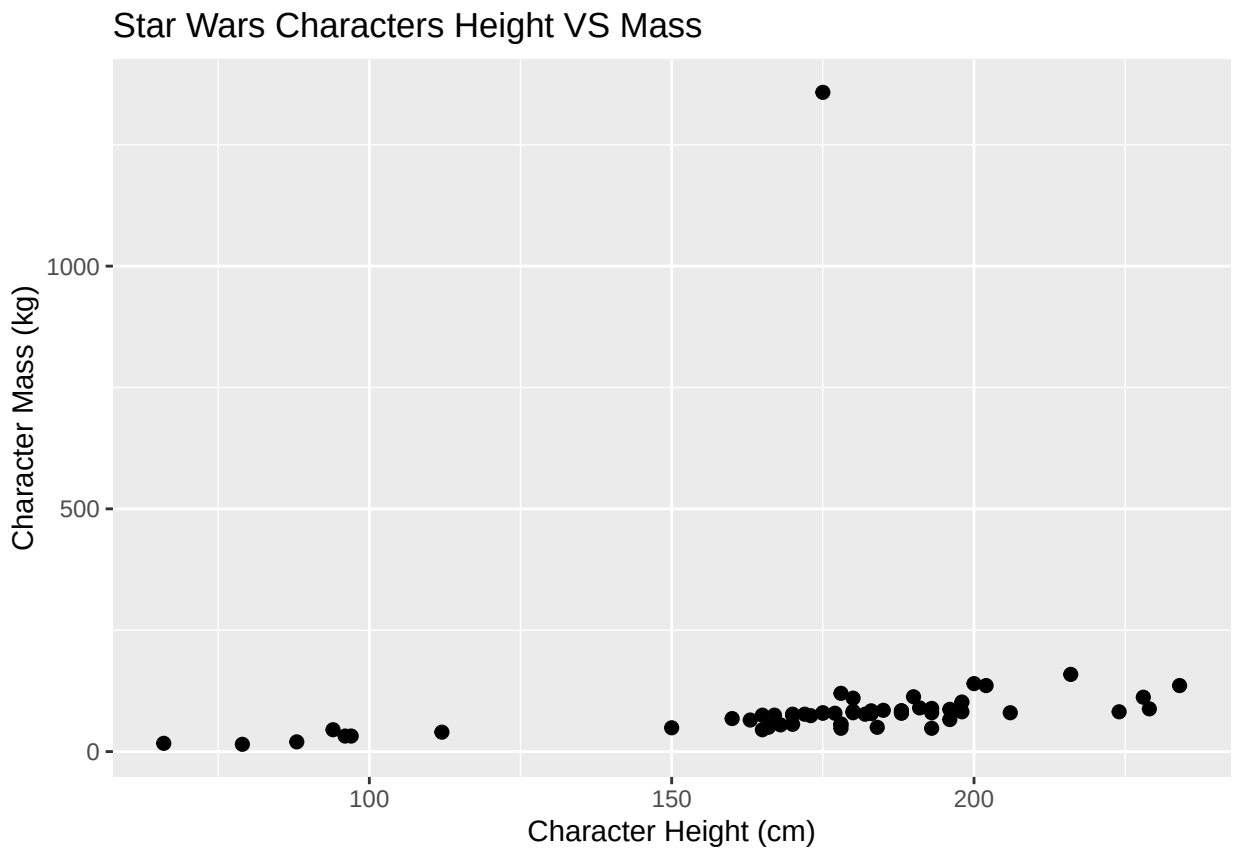
2. (10 pts) Download the file starwars.xlsx to your file system. Write an R command to create an R data frame (or tibble) called "StarWars" from that file. The result should be a data set with 11 columns and 87 rows. Make sure that missing data is handled properly and that the proper values are loaded as column names. Use str(StarWars) at the end to show that the data has been loaded properly. Do not print the dataset.

```
StarWars = data.frame(read_excel('starwars.xlsx',sheet = 'Data', range = 'A3:K90'))
str(StarWars)
```

```
## 'data.frame':    87 obs. of  11 variables:
##  $ name      : chr  "Luke Skywalker" "C-3PO" "R2-D2" "Darth Vader" ...
##  $ height    : num  172 167 96 202 150 178 165 97 183 182 ...
##  $ mass      : num  77 75 32 136 49 120 75 32 84 77 ...
##  $ hair_color: chr  "blond" NA NA "none" ...
##  $ skin_color: chr  "fair" "gold" "white, blue" "white" ...
##  $ eye_color : chr  "blue" "yellow" "red" "yellow" ...
##  $ birth_year: num  19 112 33 41.9 19 52 47 NA 24 57 ...
##  $ sex       : chr  "male" "none" "none" "male" ...
##  $ gender    : chr  "masculine" "masculine" "masculine" "masculine" ...
##  $ homeworld : chr  "Tatooine" "Tatooine" "Naboo" "Tatooine" ...
##  $ species   : chr  "Human" "Droid" "Droid" "Human" ...
```

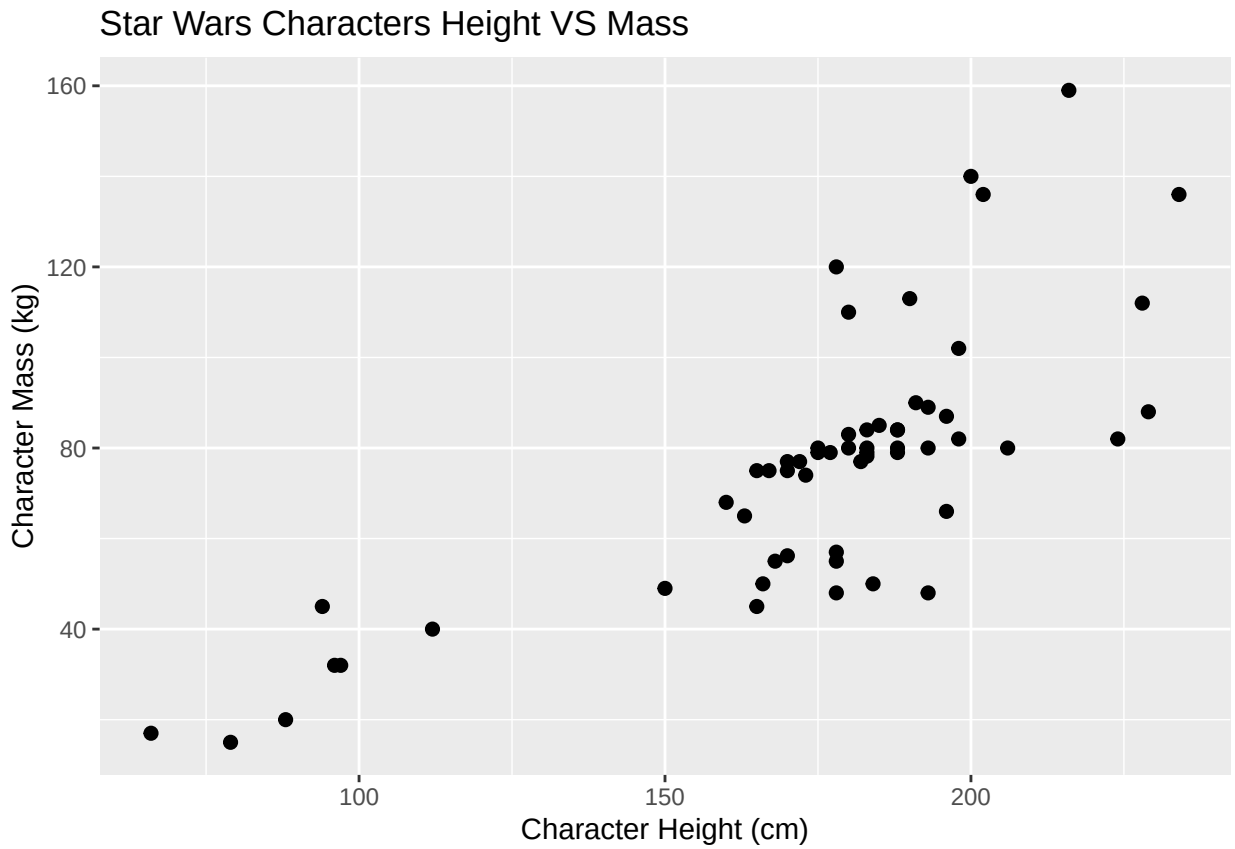
3. (10 pts) Create a scatterplot of mass (y) vs height(x). Filter the data first so that characters with missing masses or heights are removed. Include an appropriate title to the graph and label axes appropriately.

```
characters_noNA = StarWars %>%
  filter(is.na(height) == FALSE) %>%
  filter(is.na(mass) == FALSE)

ggplot(data = characters_noNA, aes(x=height, y=mass)) +
  geom_point(size=2) +
  labs(title = 'Star Wars Characters Height VS Mass',
       x = 'Character Height (cm)', y = 'Character Mass (kg)')
```



4. (10 pts) Continuing with your dataset with NAs removed, there is one character that is particularly massive and not all that tall, making it hard to see the relationship between height and weight. Filter that character out and redo the graph. Show the new graph.

```
characters_nJabba = characters_noNA %>%
  filter(mass <=1000)

ggplot(data = characters_nJabba, aes(x=height, y=mass)) +
  geom_point(size=2) +
  labs(title = 'Star Wars Characters Height VS Mass',
       x = 'Character Height (cm)', y = 'Character Mass (kg)')
```



5. (10 pts) Now that Jabba has been removed, you should be able to see an increasing relationship between height and mass. Fit a linear model to the data (without Jabba) and save the model object as "HtWtModel". Extract and show the slope and intercept parameters from the model.

```
HtWtModel = lm(data=characters_nJabba, mass ~ height)

sw_slope = HtWtModel$coefficients[2]
sw_intercept = HtWtModel$coefficients[1]

sw_slope
```

```
##    height
## 0.6213599
```

```
sw_intercept
```

```
## (Intercept)
##   -32.54076
```

6. (10 pts) Continuing with your dataset with Jabba removed, filter the data again so that characters with missing gender are removed. Then add a column to your StarWars data set that contains the characters height in inches (cm*.393701 = in).

```
characters_Genders = characters_nJabba %>%
  filter(is.na(gender) == FALSE) %>%
  mutate('height.in' = height * .393701)
```

7. (10 pts) Continuing with your dataset, produce a table that shows the average height in inches for each gender as well as the number of characters present in the data for each gender. The resulting data frame should have 2 rows and 3 columns. Make sure that data frame is displayed.

```
tableSW = characters_Genders %>%
  group_by(gender) %>%
  summarise('Average.Height' = mean(height.in),
            'num.Characters' = n()
            )
tableSW
```

```
## # A tibble: 2 x 3
##   gender    Average.Height num.Characters
##   <chr>              <dbl>          <int>
## 1 feminine            66.7              9
## 2 masculine           68.8             48
```

8. (10 pts) You are thinking of doing some analysis using hair color. Unfortunately the are a few characters who have two hair colors separated by a comma. I think the color after the comma is the color of the hair after they have aged. I want to get rid of the part after the comma and retain the younger hair color. Use mutate, stringr functions, and regular expressions to alter the hair color column so that just one hair color is listed for each character. Start with a "fresh" Star Wars dataset with all the characters included. Remove any characters whose hair color is NA. To show this change has been completed, print out the entire column of hair colors as a vector using $ to extract that column.

```
# charactersHair = StarWars %>%
#   filter(is.na(hair_color) == FALSE) %>%
#   mutate('trueHairColor' = str_replace_all(string, pattern = ', grey' | ', white', replacement=''))
#
# charactersHair$trueHairColor  --couldn't finish this one
```

9. (10 pts) In the metadata for the Star Wars data, the birth year is explained to be the number of years before the battle of Yavin. For this exercise, we will assume that the battle of Yavin occurred December 7th, 1999. Create a DateOfBirth Column in the StarWars dataset giving the date of birth for each character that has a birth_year listed(Remove any NAs). Start with a fresh Star Wars dataset. Show the first 10 rows of the data frame that only includes birth_year and DateOfBirth as columns. You will need to use lubridate functions.

```
Yavin = mdy('December 7th, 1999')

Birthdays = StarWars %>%
  select(name, birth_year) %>%
  filter(is.na(birth_year) == FALSE) %>%
  mutate('DateOfBirth' = Yavin - years(as.integer((birth_year))))

head(Birthdays, n =10)
```

```
##                   name birth_year DateOfBirth
## 1      Luke Skywalker       19.0  1980-12-07
## 2                C-3PO      112.0  1887-12-07
## 3                R2-D2       33.0  1966-12-07
## 4          Darth Vader       41.9  1958-12-07
## 5          Leia Organa       19.0  1980-12-07
## 6            Owen Lars       52.0  1947-12-07
## 7  Beru Whitesun lars       47.0  1952-12-07
## 8   Biggs Darklighter       24.0  1975-12-07
## 9       Obi-Wan Kenobi       57.0  1942-12-07
## 10   Anakin Skywalker       41.9  1958-12-07
```