In Class Exam

Matthew Hashim

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(readxl)
library(stringr)
library(lubridate)
```

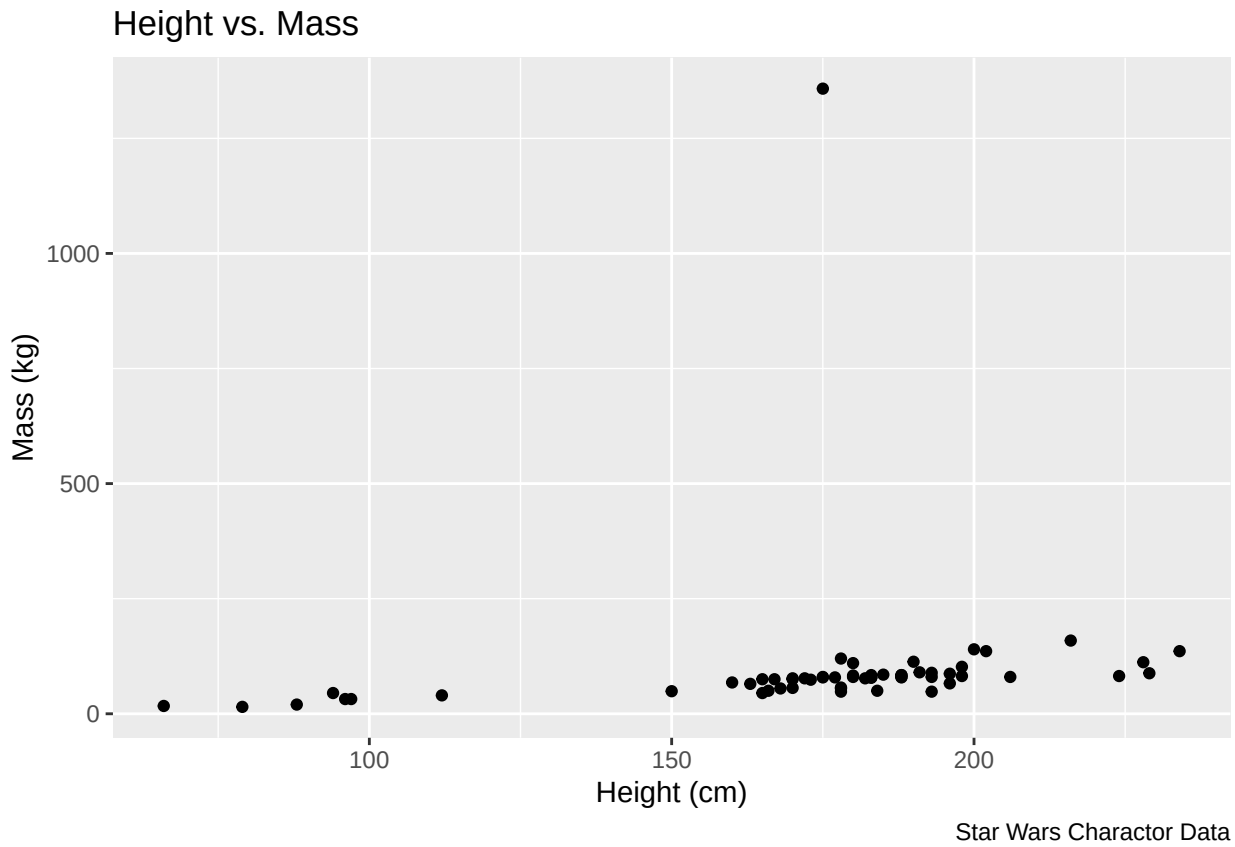
2. (10 pts) Download the file starwars.xlsx to your file system. Write an R command to create an R data frame (or tibble) called "StarWars" from that file. The result should be a data set with 11 columns and 87 rows. Make sure that missing data is handled properly and that the proper values are loaded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

```r
StarWars <- read_excel("starwars.xlsx", sheet = "Data", skip = 2)

str(StarWars)
```

```
## tibble [87 x 11] (S3: tbl_df/tbl/data.frame)
##  $ name      : chr [1:87] "Luke Skywalker" "C-3PO" "R2-D2" "Darth Vader" ...
##  $ height    : num [1:87] 172 167 96 202 150 178 165 97 183 182 ...
##  $ mass      : num [1:87] 77 75 32 136 49 120 75 32 84 77 ...
##  $ hair_color: chr [1:87] "blond" NA NA "none" ...
##  $ skin_color: chr [1:87] "fair" "gold" "white, blue" "white" ...
##  $ eye_color : chr [1:87] "blue" "yellow" "red" "yellow" ...
##  $ birth_year: num [1:87] 19 112 33 41.9 19 52 47 NA 24 57 ...
##  $ sex       : chr [1:87] "male" "none" "none" "male" ...
##  $ gender    : chr [1:87] "masculine" "masculine" "masculine" "masculine" ...
##  $ homeworld : chr [1:87] "Tatooine" "Tatooine" "Naboo" "Tatooine" ...
##  $ species   : chr [1:87] "Human" "Droid" "Droid" "Human" ...
```

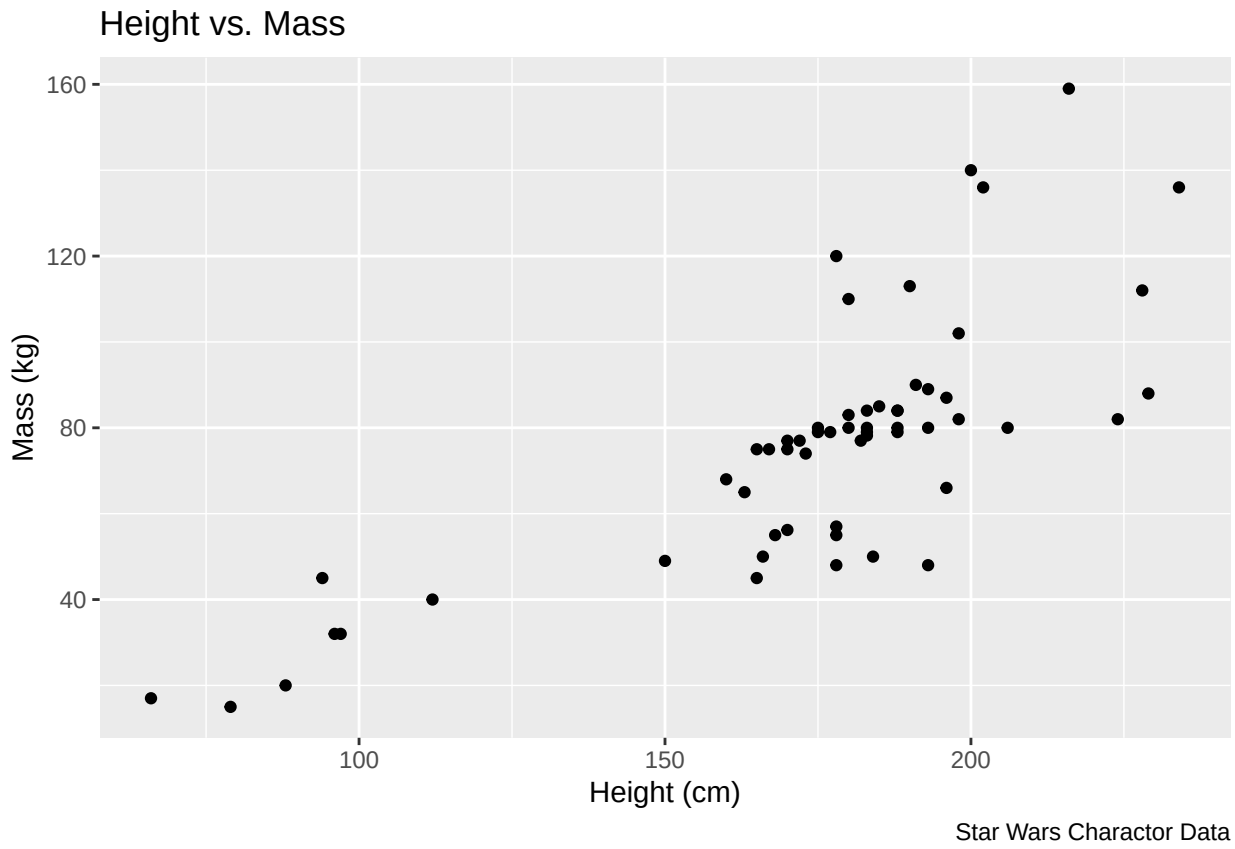
3. (10 pts) Create a scatterplot of mass (y) vs height(x). Filter the data first so that characters with missing masses or heights are removed. Include an appropriate title to the graph and label axes appropriately.

```r
StarWars.clean <- StarWars %>% drop_na(height, mass)

ggplot(StarWars.clean, aes(x = height, y = mass)) +
  geom_point() +
  labs( title='Height vs. Mass' ) +
  labs( x="Height (cm)", y="Mass (kg)" ) +
  labs( caption = "Star Wars Charactor Data" )
```



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.

```
StarWars.clean.filtered <- StarWars.clean %>% filter(mass < 500)

ggplot(StarWars.clean.filtered, aes(x = height, y = mass)) +
  geom_point() +
  labs( title='Height vs. Mass' ) +
  labs( x="Height (cm)", y="Mass (kg)" ) +
  labs( caption = "Star Wars Charactor Data" )
```



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(mass~height, data = StarWars.clean.filtered)

HtWtModel$coefficients
```

```
## (Intercept)      height
## -32.5407582   0.6213599
```

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).

```
StarWars.clean.filtered.new <- StarWars.clean.filtered %>%
  drop_na(gender) %>%
  mutate(
    "heightInInches" = mass * 0.393701
  )

head(StarWars.clean.filtered.new)
```

```
## # A tibble: 6 x 12
```

```
##   name      height  mass hair_color skin_color eye_color birth_year sex   gender
##   <chr>      <dbl> <dbl> <chr>      <chr>      <chr>          <dbl> <chr> <chr>
## 1 Luke Sky~    172    77 blond      fair       blue            19   male  mascu~
## 2 C-3PO        167    75 <NA>       gold       yellow         112   none  mascu~
## 3 R2-D2         96    32 <NA>       white, bl~ red             33   none  mascu~
## 4 Darth Va~    202   136 none       white      yellow          41.9 male  mascu~
## 5 Leia Org~    150    49 brown      light      brown           19   fema~ femin~
## 6 Owen Lars    178   120 brown, gr~ light      blue            52   male  mascu~
## # i 3 more variables: homeworld <chr>, species <chr>, heightInInches <dbl>
```

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.

```
table <- data_frame(
  "gender" = c("masculine", "feminine"),
  "height" =c(
    mean(StarWars.clean.filtered.new$heightInInches[StarWars.clean.filtered.new$gender == "masculine"]),
  mean(StarWars.clean.filtered.new$heightInInches[StarWars.clean.filtered.new$gender == "feminine"])),
  "count" = c(
    sum(StarWars.clean.filtered.new$gender == "masculine"),
    sum(StarWars.clean.filtered.new$gender == "feminine")
    )
)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## i Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
table
```

```
## # A tibble: 2 x 3
##   gender    height count
##   <chr>      <dbl> <int>
## 1 masculine   31.5    48
## 2 feminine    21.5     9
```

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.

```
StarWars.hair <- StarWars %>% drop_na(hair_color) %>%
  mutate(
    hair_color = ifelse(str_detect(hair_color, ',') == TRUE, str_sub(hair_color, end = str_locate(hair_
  )

StarWars.hair$hair_color
```

```
##  [1] "blond"   "none"    "brown"   "brown"   "brown"   "black"   "auburn"
##  [8] "blond"   "auburn"  "brown"   "brown"   "brown"   "brown"   "white"
## [15] "grey"    "black"   "none"    "none"    "black"   "none"    "none"
## [22] "auburn"  "brown"   "brown"   "none"    "brown"   "none"    "blond"
## [29] "none"    "none"    "none"    "brown"   "black"   "none"    "black"
## [36] "black"   "none"    "none"    "none"    "none"    "none"    "none"
## [43] "none"    "white"   "none"    "black"   "none"    "none"    "none"
## [50] "none"    "none"    "black"   "brown"   "brown"   "none"    "black"
## [57] "black"   "brown"   "white"   "black"   "black"   "blonde"  "none"
## [64] "none"    "none"    "white"   "none"    "none"    "none"    "none"
## [71] "none"    "none"    "brown"   "brown"   "none"    "none"    "black"
## [78] "brown"   "brown"   "none"    "unknown" "brown"
```

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.

```
StarWars.DOB <- StarWars %>%
  drop_na(birth_year)  %>%
  mutate(
    "DateOfBirth" = as.period(year(birth_year) %--%mdy("December 7th, 1999"))$year
  )
```

```
## Warning: There were 2 warnings in `mutate()`.
## The first warning was:
## i In argument: `DateOfBirth = as.period(year(birth_year) %--% mdy("December
##   7th, 1999"))$year`.
## Caused by warning:
## ! tz(): Don't know how to compute timezone for object of class numeric; returning "UTC".
## i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```
str(StarWars.DOB$DateOfBirth)
```

```
##  num [1:43] 29 29 29 29 29 29 29 29 29 29 ...
```