Exploring Data Integration and Visualization Techniques using R and SQLite

Me

2023-05-01

Question 1

```
### (a) Read data in ###


# Read movies500.csv
movies <- read.csv("movies500.csv")

# Read movies500_genres.csv
movies_genres <- read.csv("movies500_genres.csv")

# Read genres.csv
genres <- read.csv("genres.csv")


### (b) Create a SQLite database ###

library(RSQLite)
```

```
## Warning: package 'RSQLite' was built under R version 4.2.3
```

```
# delete file if it exists
if (file.exists("movies.sqlite")) {
  file.remove("movies.sqlite")
}
```

```
## [1] TRUE
```

```
# Create a connection to the SQLite database
con <- dbConnect(SQLite(), "movies.sqlite")

### (c) Copy the data ###
# Copy movies data to the movies table in SQLite
dbWriteTable(con, "movies", movies, overwrite = TRUE, na.omit())

# Copy movies_genres data to the movies_genres table in SQLite
dbWriteTable(con, "movies_genres", movies_genres, overwrite = TRUE, na.omit())
```

```
--### (d) Count the number of rows ###
SELECT COUNT(*) FROM movies
```

1 records

COUNT(*)
500

```
--### (e) Output a list of movies ###
SELECT title,
  runtime, release_date
  FROM movies
  WHERE runtime > 480
  ORDER BY runtime ASC
```

8 records

title	runtime	release_date
Planet Earth	550	2006-12-10
Tie Xi Qu: West of the Tracks	551	2002-04-26
Shoah	566	1985-11-01
The Godfather Trilogy: 1972-1990	583	1992-10-17
New York: A Documentary Film	600	1999-11-14
The Civil War	680	1990-09-23
The Story of Film: An Odyssey	900	2011-09-03
Heimat: A Chronicle of Germany	925	1984-09-16

```
--### (f) Movies with love in the title  ###
SELECT title
FROM movies
WHERE title
LIKE '%love%'
```

Displaying records 1 - 10

title
Marvin Hamlisch: What He Did For Love
Love at 16
My Future Love
Frankie Boyle: Hurt Like You’ve Never Been Loved
Harold and Lillian: A Hollywood Love Story
Leather Jacket Love Story

title
Love Torn in a Dream
The Loves of Pharaoh
Love You You
From Mexico With Love

```sql
--### (g) Create a table genres  ###
CREATE TABLE genres (
          genre_id INTEGER PRIMARY KEY,
          genre_name TEXT)
```

```r
dbWriteTable(con, "genres", genres, overwrite = TRUE)
```

```sql
--### (h) Copy contents over to table of genres  ###
INSERT INTO genres (genre_id, genre_name)
          SELECT genre_id, genre_name
          FROM genres
```

```sql
--### (i) Add new row  ###
INSERT INTO genres (genre_id, genre_name)
          VALUES (3579, 'University Comedy')
```

```sql
--### (j) Modify the name of genre 3579  ###
UPDATE genres
          SET genre_name = 'University Tragedy'
          WHERE genre_id = 3579
```

```sql
--### (k) Find id' s associated with the movie Running Wild  ###

SELECT genre_id
FROM movies_genres
WHERE tmdbId IN (SELECT tmdbId FROM movies WHERE title = 'Running Wild')
```

2 records

genre_id
12
18

```sql
--### (l) Three way join  ###
SELECT DISTINCT genres.genre_name
FROM movies_genres
INNER JOIN genres ON movies_genres.genre_id = genres.genre_id
WHERE movies_genres.tmdbId IN (SELECT tmdbId FROM movies WHERE title = 'Running Wild');
```

2 records

genre_name
Adventure
Drama

```
--### (m) number of movies by genre
SELECT genres.genre_name, COUNT(*) AS movie_count
FROM movies_genres
LEFT JOIN genres ON movies_genres.genre_id = genres.genre_id
GROUP BY genres.genre_name
HAVING movie_count >= 20
ORDER BY movie_count DESC
```

Displaying records 1 - 10

genre_name	movie_count
Drama	328
Documentary	292
Comedy	226
Romance	80
Music	56
Action	44
Crime	40
Family	38
Animation	38
History	36

```
# Disconnect from the SQLite database
dbDisconnect(con)
```

Question 2

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```
# Read the file into R
vehicles <- read.csv("motor_vehicle_modified.csv", stringsAsFactors = FALSE)
```

```
### (a) “4-gear auto” and are make Kia or Honda  ###
# Find the number of vehicles with transmission type "4-gear auto" and make Kia or Honda
count_vehicles <- nrow(filter(vehicles, transmission_type == "4-gear auto" & make %in% c("Ki
a", "Honda")))
count_vehicles
```

```
## [1] 13
```

```
### (b) Drop columns  ###
# Drop the columns vehicle_usage and vehicle_type
vehicles <- select(vehicles, -vehicle_usage, -vehicle_type)
```

```
### (c) Create, a contingency  ###
# Create the contingency table vehicles_country_status
vehicles_country_status <- table(vehicles$original_country, vehicles$import_status)

### (d) top 3 countries used  ###
# Calculate the total number of used cars by country
used_cars_by_country <- vehicles_country_status[, "used"]

# Sort the countries in decreasing order of the number of used cars
sorted_countries <- names(used_cars_by_country)[order(used_cars_by_country, decreasing = TRU
E)]

# Filter out the "Not Known" country from the top countries
top_countries <- sorted_countries[!sorted_countries %in% "Not Known"][1:3]

# Display the resulting table for the top countries (excluding "Not Known") with all import s
tatuses
top_countries_table <- vehicles_country_status[top_countries, ]
top_countries_table
```

```
##
##                   new re-reg scratch used
##   Japan          1318     39       0 1172
##   Germany         175      5       0  137
##   United Kingdom   79     19       0   34
```

Question 3

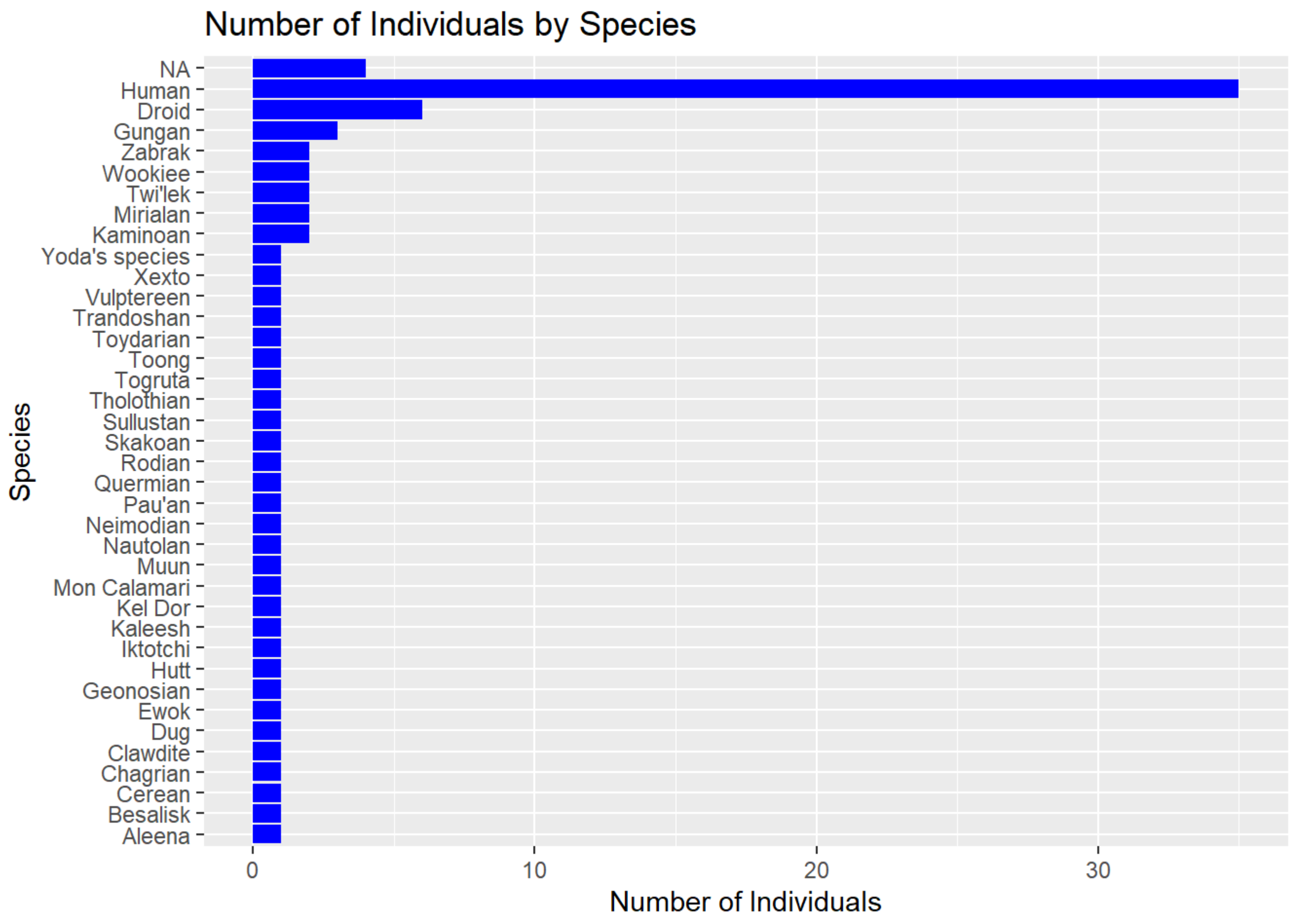
```
library(dplyr)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
# Load the Star Wars dataset
data(starwars)

### (a) draw a horizontal bar chart  ###
# Count the number of individuals by species
species_count <- starwars %>%
  group_by(species) %>%
  summarize(count = n())

# Plot the horizontal bar chart
ggplot(species_count, aes(x = count, y = reorder(species, count))) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(x = "Number of Individuals", y = "Species") +
  ggtitle("Number of Individuals by Species")
```

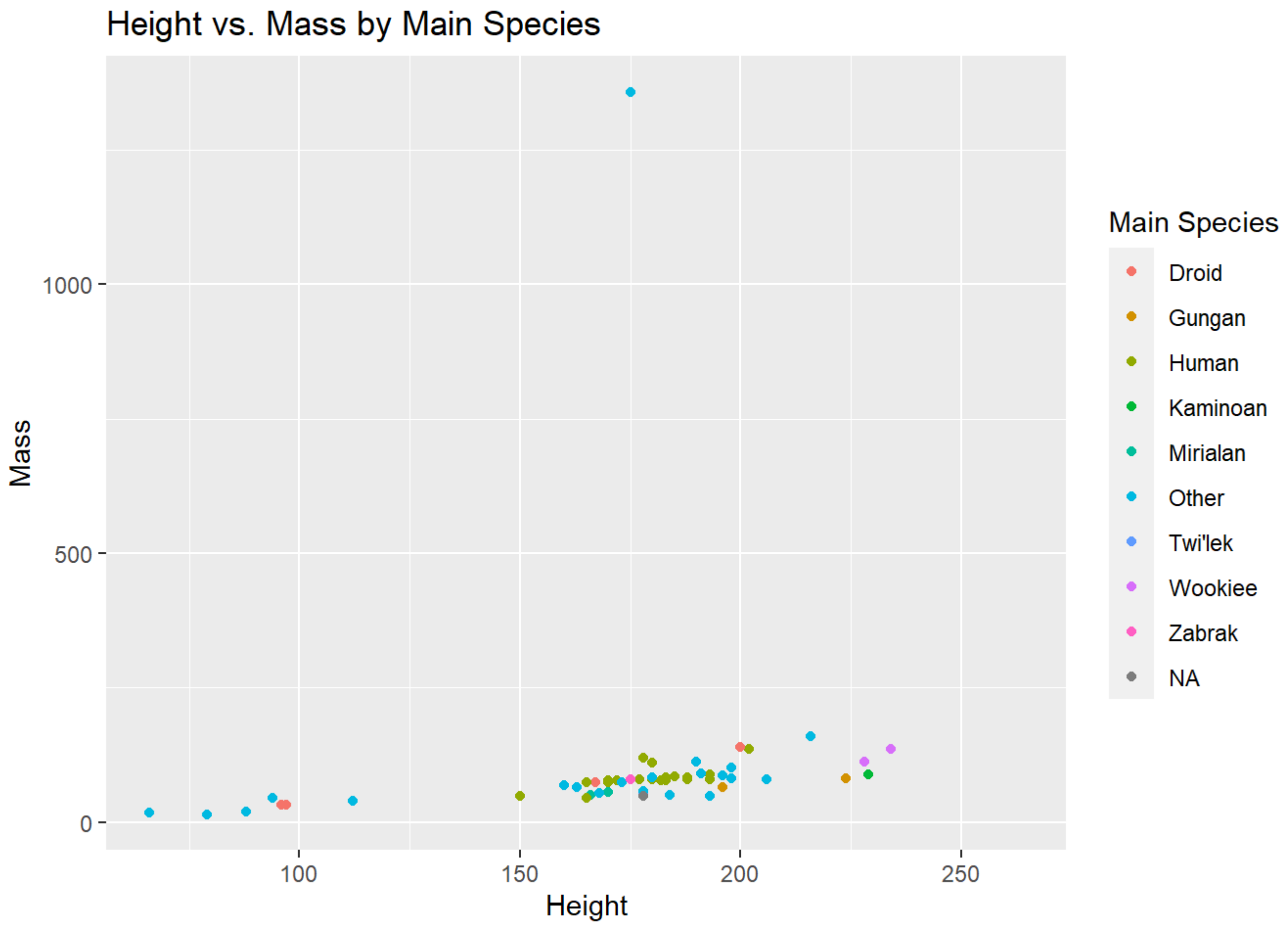


```
### (b) Add a column  ###
# Add a column 'num' showing the number of individuals of each species
starwars <- starwars %>%
  mutate(num = table(species)[as.character(species)])
```

```
### (c) Add a column  ###
# Add a column 'mainspecies' based on 'num' values
starwars <- starwars %>%
  mutate(mainspecies = ifelse(num > 1, as.character(species), "Other"))
```

```
### (d) draw a scatter plot of the height and mass  ###
# Draw a scatter plot of height and mass, colored by 'mainspecies'
ggplot(starwars, aes(x = height, y = mass, color = mainspecies)) +
  geom_point() +
  labs(x = "Height", y = "Mass", color = "Main Species") +
  ggtitle("Height vs. Mass by Main Species")
```

```
## Warning: Removed 28 rows containing missing values (`geom_point()`).
```



```
### (e) Identify outlier ###
# Identify and remove the outlier
outlier <- starwars %>%
  filter(height > 150 & mass > 500) %>%
  select(name, height, mass, species)
outlier
```

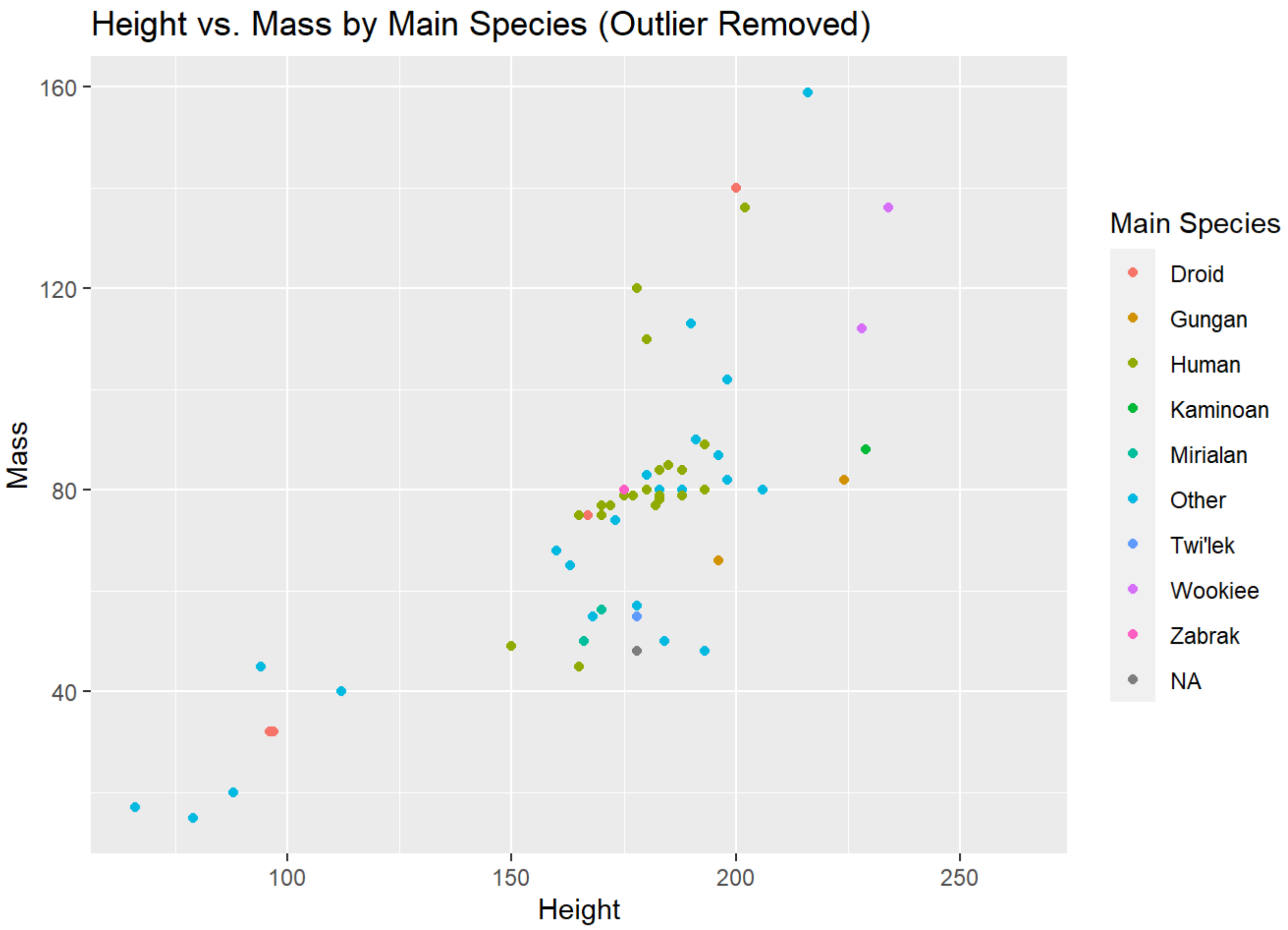
```
## # A tibble: 1 × 4
##   name                  height  mass species
##   <chr>                  <int> <dbl> <chr>  
## 1 Jabba Desilijic Tiure    175  1358 Hutt
```

```
starwars <- starwars %>%
  anti_join(outlier)
```

```
## Joining with `by = join_by(name, height, mass, species)`
```

```
# Redraw the scatter plot without the outlier
ggplot(starwars, aes(x = height, y = mass, color = mainspecies)) +
  geom_point() +
  labs(x = "Height", y = "Mass", color = "Main Species") +
  ggtitle("Height vs. Mass by Main Species (Outlier Removed)")
```

```
## Warning: Removed 28 rows containing missing values (`geom_point()`).
```



```r
### (f) side-by-side scatter plots ###
# Filter the data for humans and droids
humans <- starwars %>% filter(mainspecies == "Human")
droids <- starwars %>% filter(mainspecies == "Droid")

# Create scatter plots for humans and droids using facet_wrap
ggplot() +
  geom_point(data = humans, aes(x = height, y = mass, color = mainspecies)) +
  geom_smooth(data = humans, aes(x = height, y = mass, color = mainspecies), method = "lm", s
e = FALSE) +
  geom_point(data = droids, aes(x = height, y = mass, color = mainspecies)) +
  geom_smooth(data = droids, aes(x = height, y = mass, color = mainspecies), method = "lm", s
e = FALSE) +
  facet_wrap(~ mainspecies, ncol = 2) +
  labs(x = "Height", y = "Mass", color = "Main Species") +
  ggtitle("Scatter Plots of Height vs. Mass for Humans and Droids")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

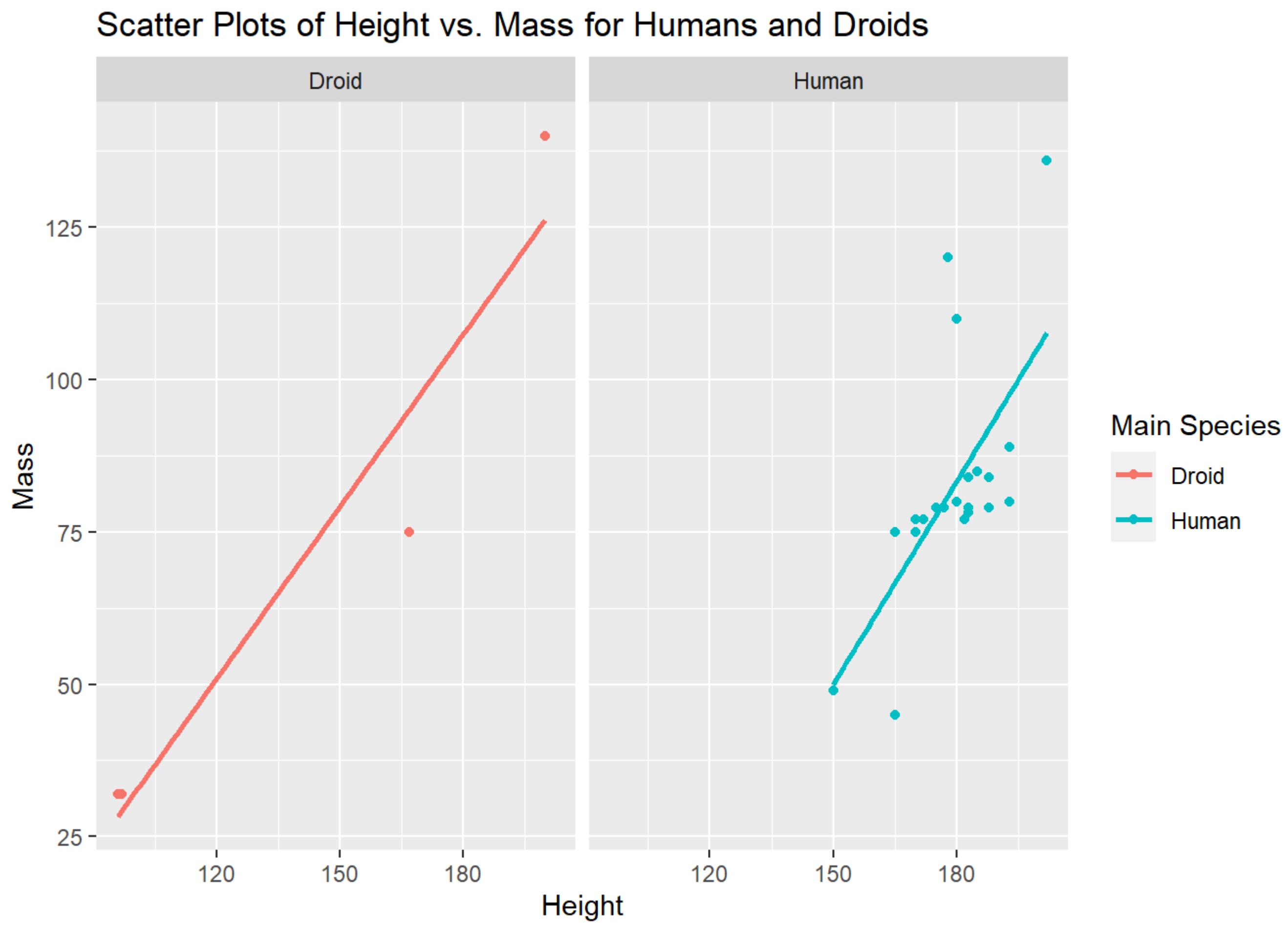
```
## Warning: Removed 13 rows containing non-finite values (`stat_smooth()`).
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 2 rows containing non-finite values (`stat_smooth()`).
```

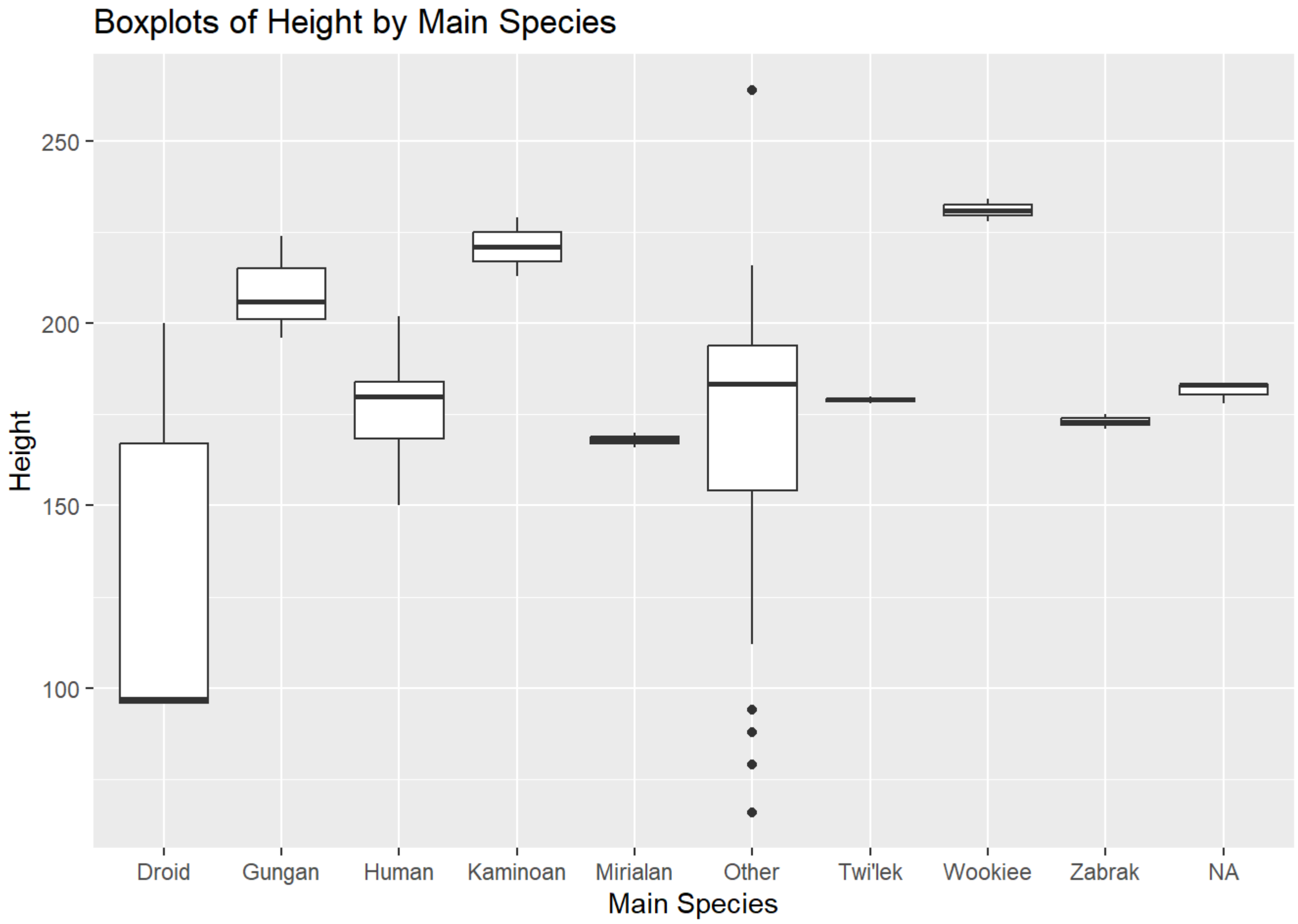
```
## Warning: Removed 13 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 2 rows containing missing values (`geom_point()`).
```

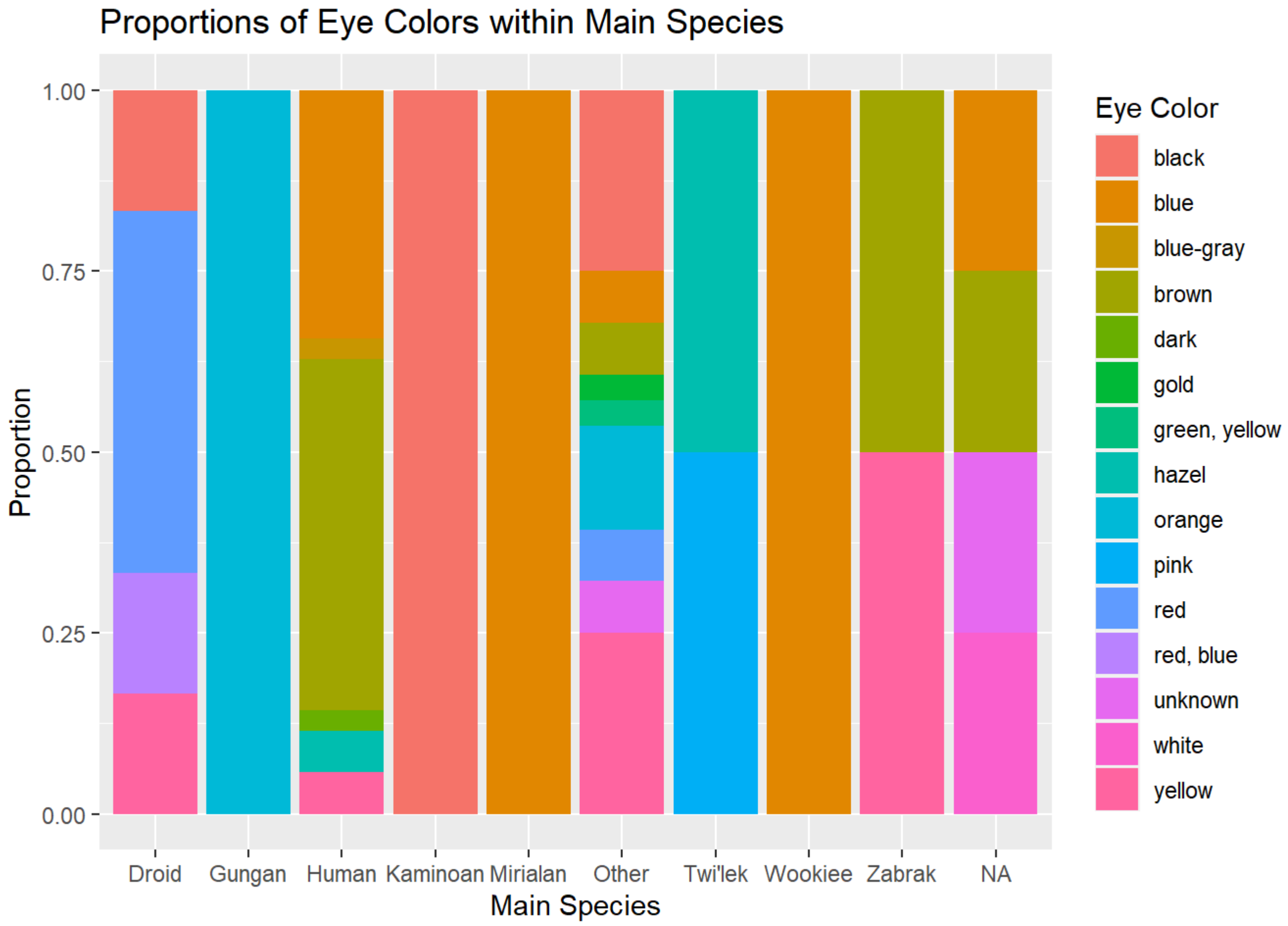


```
### (g) Draw boxplots ###
# Draw boxplots of height for each 'mainspecies'
ggplot(starwars, aes(x = mainspecies, y = height)) +
  geom_boxplot() +
  labs(x = "Main Species", y = "Height") +
  ggtitle("Boxplots of Height by Main Species")
```

```
## Warning: Removed 6 rows containing non-finite values (`stat_boxplot()`).
```



```
### (h) Draw a horizontal stacked bar chart ###
# Draw a horizontal stacked bar chart of eye color proportions within each 'mainspecies'
ggplot(starwars, aes(x = mainspecies, fill = eye_color)) +
  geom_bar(position = "fill") +
  labs(x = "Main Species", y = "Proportion", fill = "Eye Color") +
  ggtitle("Proportions of Eye Colors within Main Species")
```


Proportions of Eye Colors within Main Species
Proportion
1.00
0.75
0.50
0.25
0.00
Droid
Gungan
Human
Kaminoan
Mirialan
Other
Twi'lek
Wookiee
Zabrak
NA
Main Species
Eye Color
black
blue
blue-gray
brown
dark
gold
green, yellow
hazel
orange
pink
red
red, blue
unknown
white
yellow