Exercises for Data Visualisation

September 4, 2017

Emanuela Furfaro
emanuela.furfaro@quantide.com[1]

[1] mailto:emanuela.furfaro@quantide.com

Contents

Chapter 1

Introduction

In this document you will find some exercises about data visualisation. In the first part you will find some exercises on basic data visulisation, while in the second part you will find exercises on advanced topics of data visulisation.

Chapter 2

Data Visualization with ggplot2

Load ggplot2 package, supposing it is already installed.

```
require(tidyverse)
```

2.1 Data

2.1.1 iris

Some of the following exercises are based on the iris dataset, taken from the datasets package. It is a base package so it is already installed and loaded.

```
data("iris")
```

This dataset gives the measurements in centimeters of length and width of sepal and petal, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica.

iris dataset contains the following variables:

- Sepal.Length: length of iris sepal
- Sepal.Width: width of iris sepal
- Petal.Length: length of iris petal
- Petal.Width: width of iris petal
- Species: species of iris

```
dim(iris)
```

```
## [1] 150   5
```

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

2.1.2 Comic characters data

Other exercises are based on `marvel_wikia_data` dataset, that you may find in the folder `exercises/data`.

```
marvel_wikia_data <- read_csv("marvel-wikia-data.csv")
```

```
## Parsed with column specification:
## cols(
##   page_id = col_integer(),
##   name = col_character(),
##   urlslug = col_character(),
##   ID = col_character(),
##   ALIGN = col_character(),
##   EYE = col_character(),
##   HAIR = col_character(),
##   SEX = col_character(),
##   GSM = col_character(),
##   ALIVE = col_character(),
##   APPEARANCES = col_integer(),
##   `FIRST APPEARANCE` = col_character(),
##   Year = col_integer()
## )
```

```
getwd()
```

```
## [1] "/home/emanuela/dev/qtraining/060-ggplot/data"
```

The data comes from Marvel Wikia. The file was scraped in August 2014 and contains the following variables:

- `page_id`: The unique identifier for that characters page within the wikia
- `name`: The name of the character
- `urlslug`: The unique url within the wikia that takes you to the character
- `ID`: The identity status of the character (Secret Identity, Public identity, [on marvel only: No Dual Identity])
- `ALIGN`: If the character is Good, Bad or Neutral
- `EYE`: Eye color of the character
- `HAIR`: Hair color of the character
- `SEX`: Sex of the character (e.g. Male, Female, etc.)
- `GSM`: If the character is a gender or sexual minority (e.g. Homosexual characters, bisexual characters)
- `ALIVE`: If the character is alive or deceased
- `APPEARANCES`: The number of appareances of the character in comic books (as of Sep. 2, 2014. Number will become increasingly out of date as time goes on.)
- `FIRST APPEARANCE` The month and year of the character's first appearance in a comic book, if available
- `YEAR`: The year of the character's first appearance in a comic book, if available

```
dim(marvel_wikia_data)
```

```
## [1] 16376    13
```

```
head(marvel_wikia_data)
```

```
## # A tibble: 6 x 13
##   page_id                                    name
##     <int>                                   <chr>
## 1    1678                 Spider-Man (Peter Parker)
## 2    7139           Captain America (Steven Rogers)
## 3   64786 "Wolverine (James \\\"Logan\\\" Howlett)"
## 4    1868   "Iron Man (Anthony \\\"Tony\\\" Stark)"
## 5    2460                        Thor (Thor Odinson)
## 6    2458                 Benjamin Grimm (Earth-616)
## # ... with 11 more variables: urlslug <chr>, ID <chr>, ALIGN <chr>, EYE <chr>,
## #   HAIR <chr>, SEX <chr>, GSM <chr>, ALIVE <chr>, APPEARANCES <int>, `FIRST
## #   APPEARANCE` <chr>, Year <int>
```

```
head(str(marvel_wikia_data))
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    16376 obs. of  13 variables:
##  $ page_id         : int  1678 7139 64786 1868 2460 2458 2166 1833 29481 1837 ...
##  $ name            : chr  "Spider-Man (Peter Parker)" "Captain America (Steven Rogers)" "Wolverine
##  $ urlslug         : chr  "\\/Spider-Man_(Peter_Parker)" "\\/Captain_America_(Steven_Rogers)" "
##  $ ID              : chr  "Secret Identity" "Public Identity" "Public Identity" "Public Identity"
##  $ ALIGN           : chr  "Good Characters" "Good Characters" "Neutral Characters" "Good Characte
##  $ EYE             : chr  "Hazel Eyes" "Blue Eyes" "Blue Eyes" "Blue Eyes" ...
##  $ HAIR            : chr  "Brown Hair" "White Hair" "Black Hair" "Black Hair" ...
##  $ SEX             : chr  "Male Characters" "Male Characters" "Male Characters" "Male Characters"
##  $ GSM             : chr  NA NA NA NA ...
##  $ ALIVE           : chr  "Living Characters" "Living Characters" "Living Characters" "Living Cha
##  $ APPEARANCES     : int  4043 3360 3061 2961 2258 2255 2072 2017 1955 1934 ...
##  $ FIRST APPEARANCE: chr  "Aug-62" "Mar-41" "Oct-74" "Mar-63" ...
##  $ Year            : int  1962 1941 1974 1963 1950 1961 1961 1962 1963 1961 ...
##  - attr(*, "spec")=List of 2
##   ..$ cols   :List of 13
##   .. ..$ page_id         : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ name            : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ urlslug         : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ ID              : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ ALIGN           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ EYE             : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ HAIR            : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ SEX             : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ GSM             : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ ALIVE           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ APPEARANCES     : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ FIRST APPEARANCE: list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ Year             : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   ..$ default: list()
##   .. ..- attr(*, "class")= chr  "collector_guess" "collector"
##   ..- attr(*, "class")= chr "col_spec"
```

```
## NULL
```

2.2 Scatterplot

Let us consider `iris` dataset.

2.2.1 Exercise 1

a. Generate a scatterplot to analyze the relationship between `Sepal.Width` and `Sepal.Length` variables.

b. Set the size of the point as 3 and their colour (`colour` and `fill` arguments) as "green". *advanced* c. Add "Sepal Characteristics" as a red italic title and change axis title to "Sepal length" and "Sepal width".

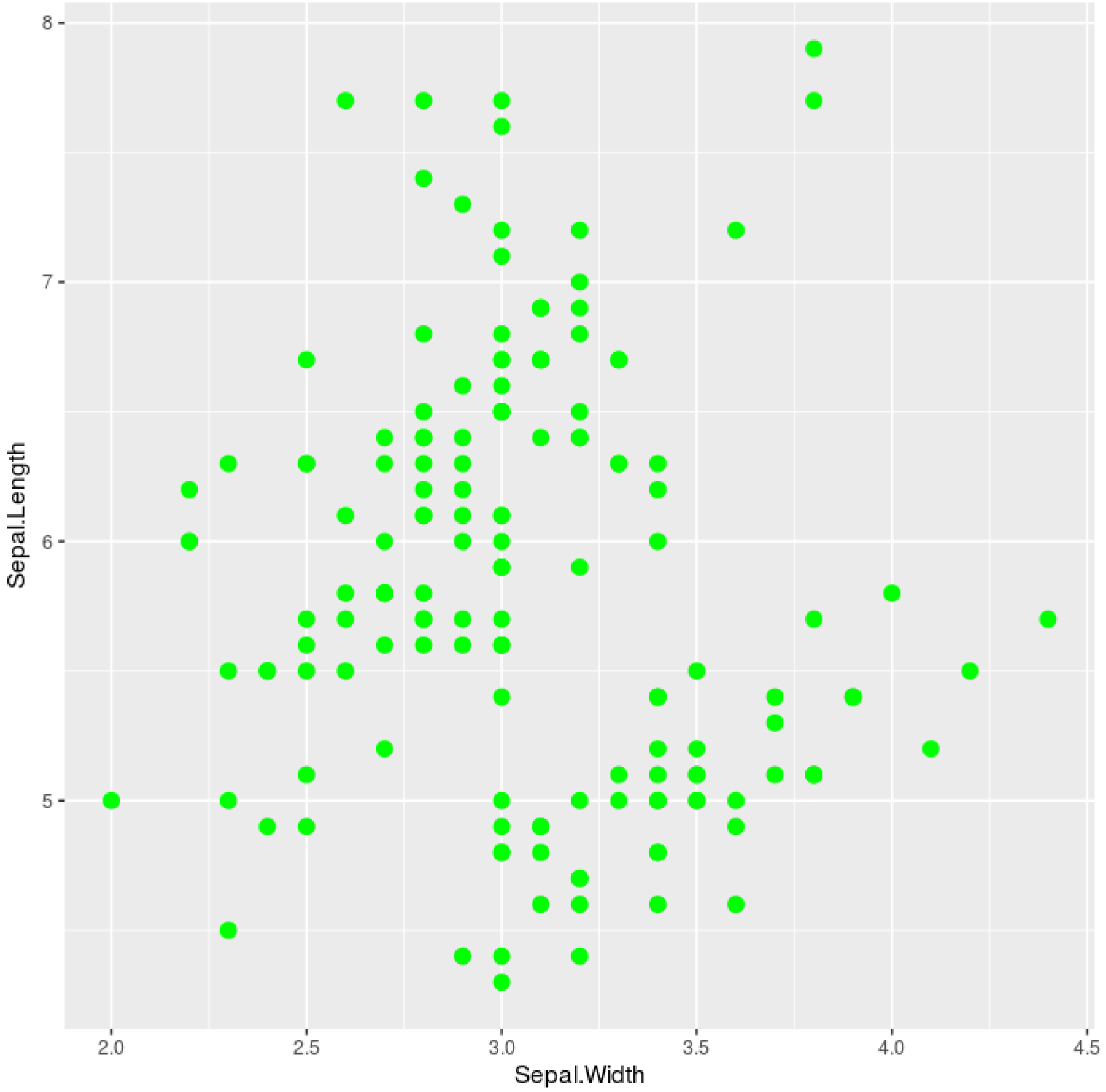


Figure 2.1:

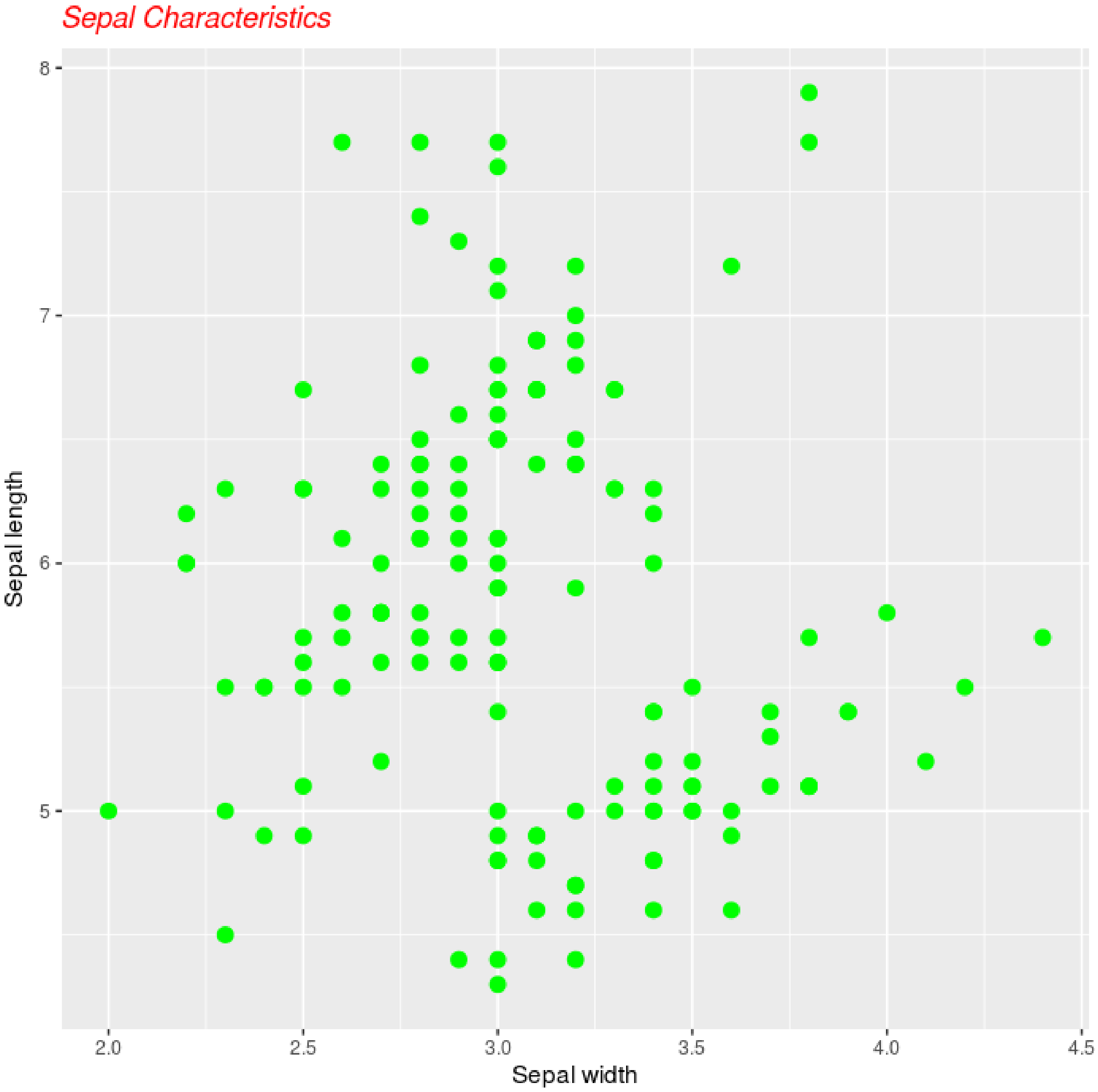

Sepal Characteristics
Sepal length
8
7
6
5
Sepal width
2.0
2.5
3.0
3.5
4.0
4.5

Figure 2.2:

2.2.2 Exercise 2

a. Generate a scatterplot to analyze the relationship between `Petal.Width` and `Petal.Length` variables according to iris species, mapped as `colour` aes. *advanced* b. Change axis title to “Sepal length” and “Sepal width”. *advanced* c. Move the legend to the bottom.

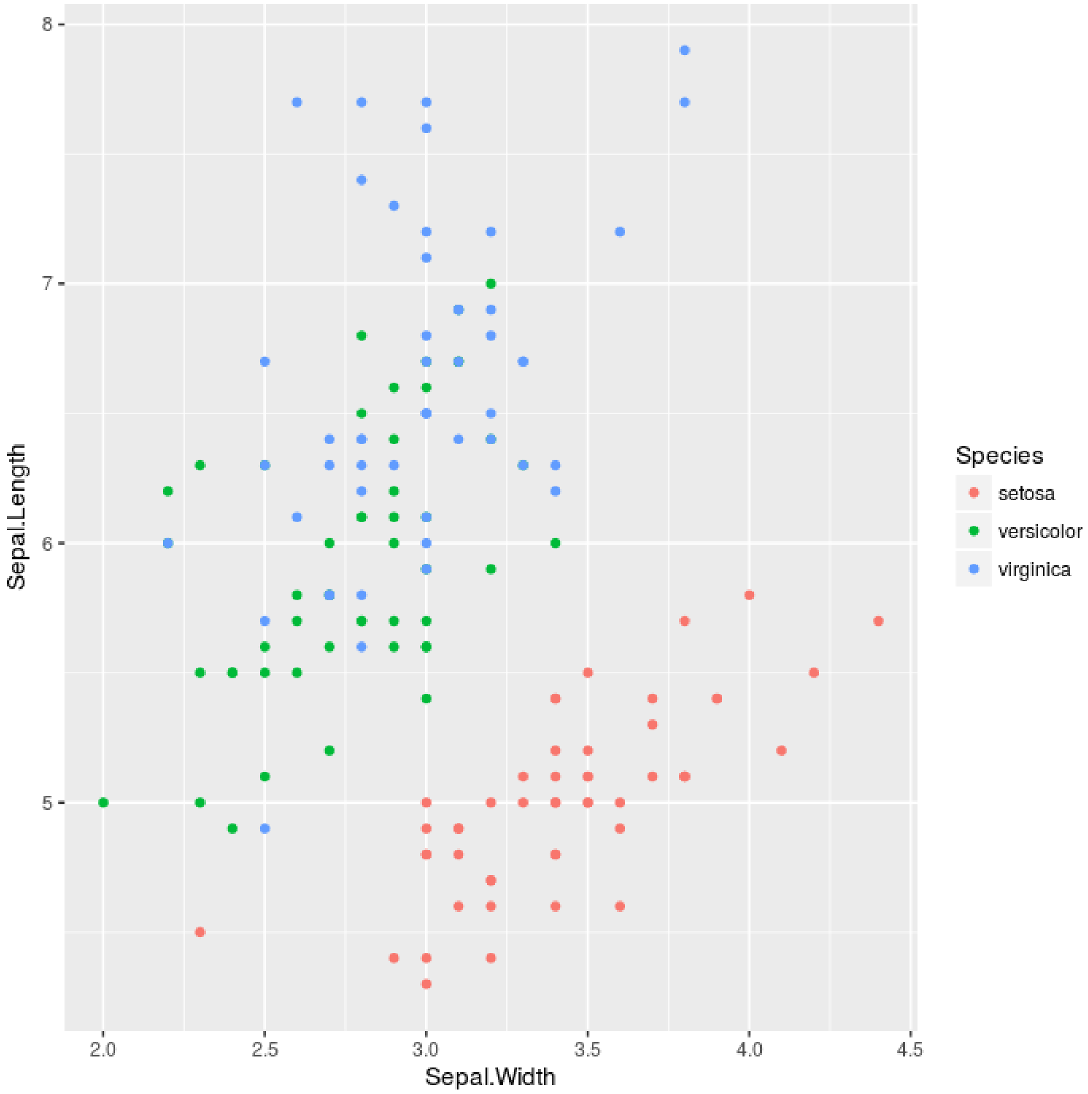


Figure 2.3:

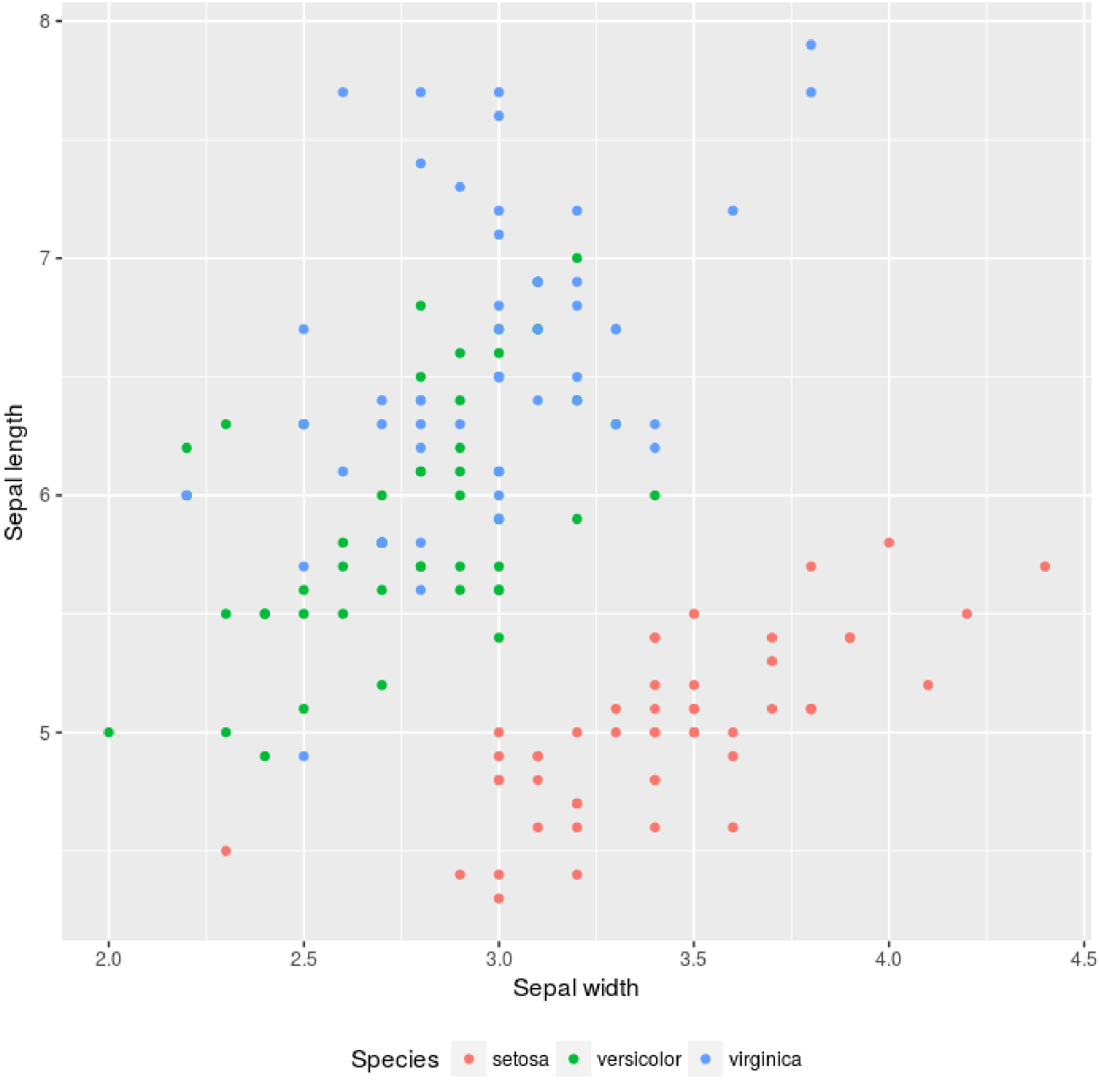

8
7
6
5
Sepal length
2.0
2.5
3.0
3.5
4.0
4.5
Sepal width
Species
setosa
versicolor
virginica

Figure 2.4:

2.3 Line PLot

Let us consider `marvel_wikia_data` dataset.

2.3.1 Exercise 1

a. Build a line plot to see the number of new characters that come out each year.

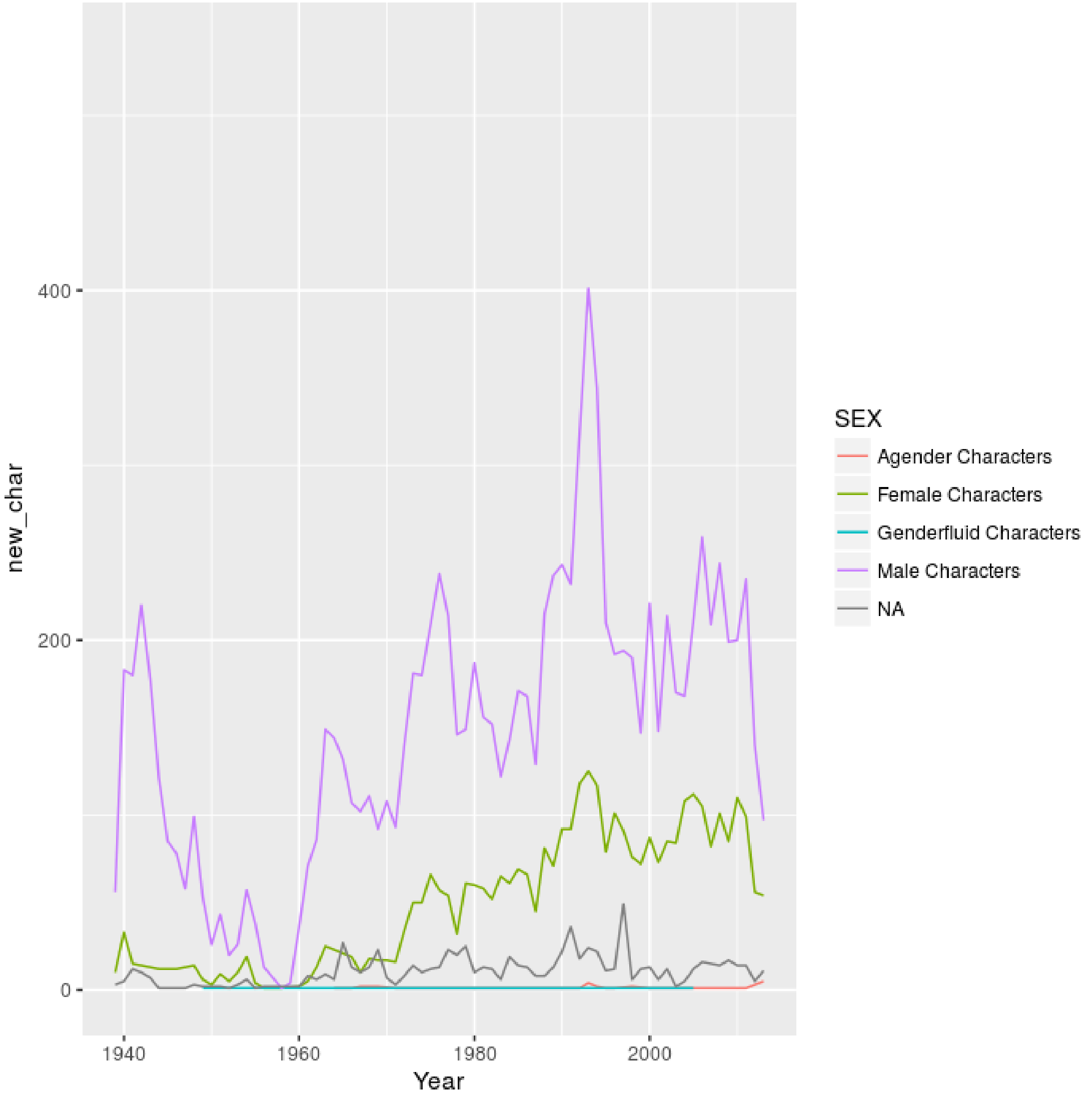
b. Build a lineplot to compare the differences in the number of female characters and male characters that come out each year.

c. Do as in b. but use different line types as well as different point types and different colours

```
## Warning: Removed 4 rows containing missing values (geom_path).
```

```
## Warning: Removed 4 rows containing missing values (geom_path).
```

```
## Warning: Removed 73 rows containing missing values (geom_point).
```



400
200
0
new_char
1940
1960
1980
2000
Year
SEX
Agender Characters
Female Characters
Genderfluid Characters
Male Characters
NA

Figure 2.5:

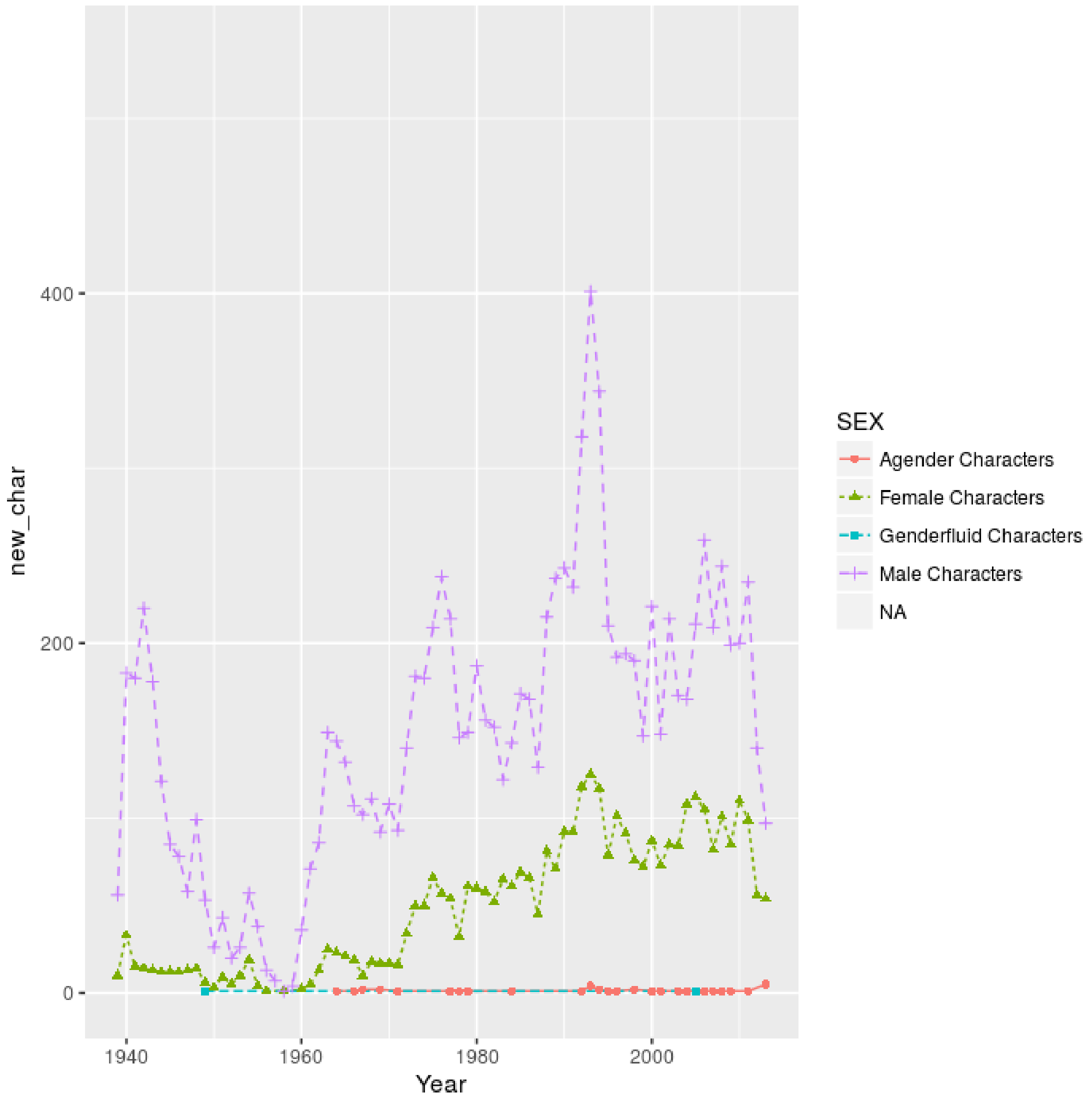

new_char
400
200
0
1940
1960
1980
2000
Year
SEX
Agender Characters
Female Characters
Genderfluid Characters
Male Characters
NA

Figure 2.6:

2.4 Barplot

Let us consider the `marvel_wikia_data` dataset.

2.4.1 Exercise 1

a. Build a stacked barplot for representing the number of new comic characters distinguishing them by `ALIGN` and map fill to `SEX`.

b. Consider only comic characters with blond hair and Black Hair (`filter(HAIR == "Black Hair" | HAIR == "Blond Hair")`). Build a stacked barplot for representing the number of new comic characters distinguishing them by `ALIGN` and map fill to `HAIR`.

c. Take the barplot in (b.) and represent the distribution on Blond Hair between the character type (Good, Bad, neutral).

2.4.2 Exercise 2

a. Consider only female and male comic characters (`filter(SEX == "Male Characters" | SEX == "Female Characters")`). Build a barplot with dodged barsfor representing the number comic characters distinguishing them by `ALIGN` and flip coordinates. Set bars width as 0.5.

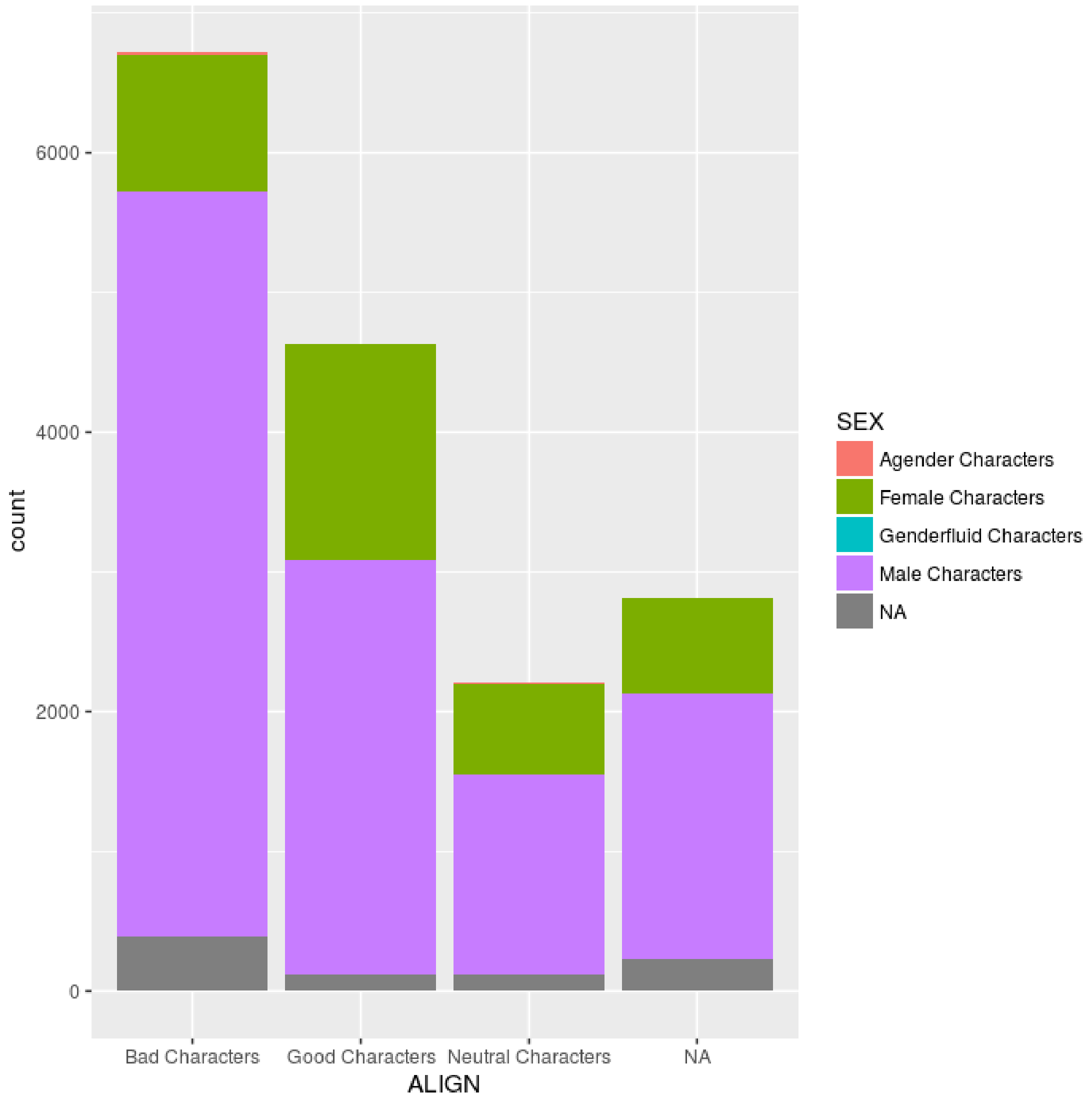

6000
4000
2000
0
count
Bad Characters
Good Characters
Neutral Characters
NA
ALIGN
SEX
Agender Characters
Female Characters
Genderfluid Characters
Male Characters
NA

Figure 2.7:

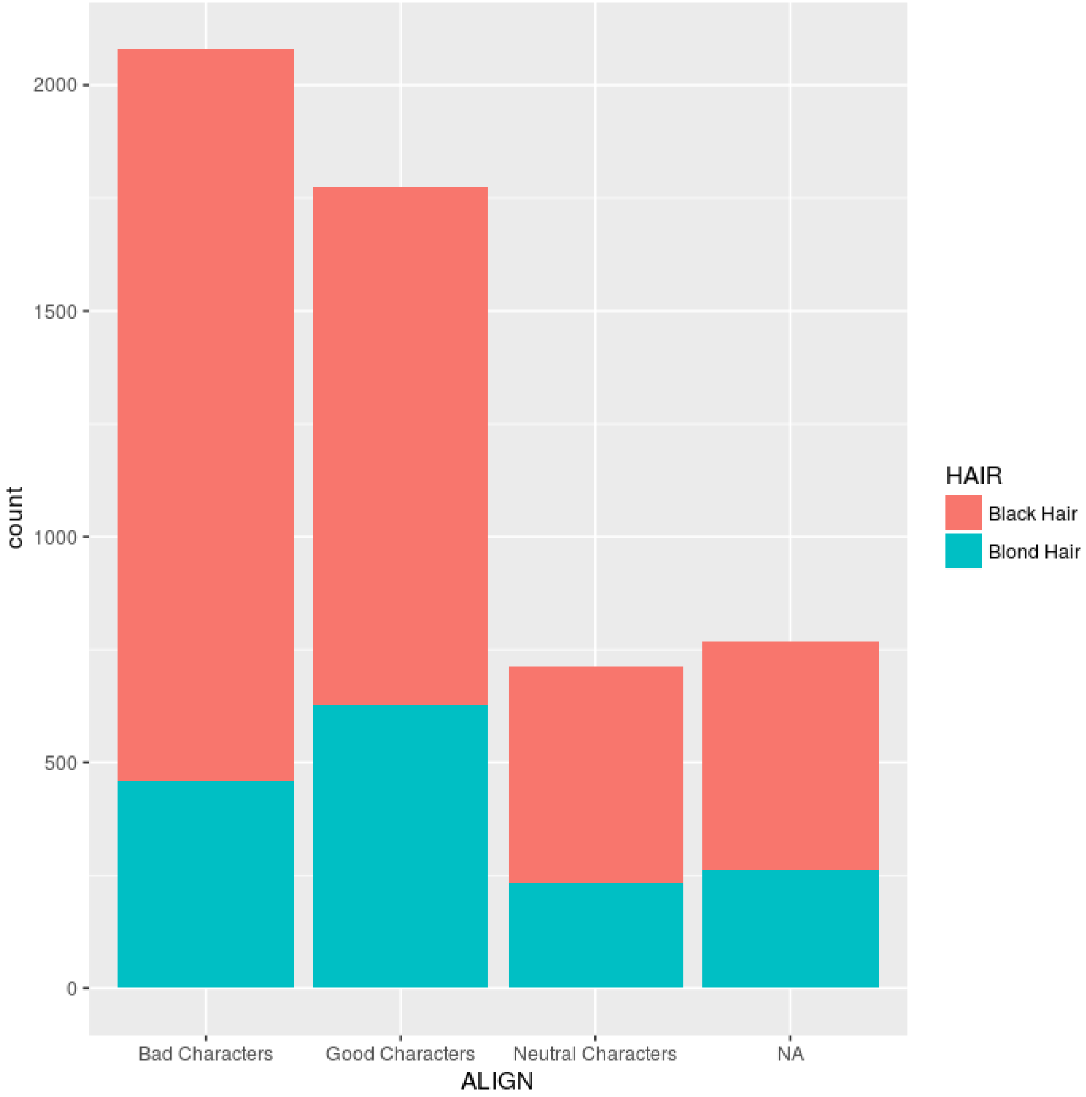

count
2000
1500
1000
500
0
Bad Characters
Good Characters
Neutral Characters
NA
ALIGN
HAIR
Black Hair
Blond Hair

Figure 2.8:

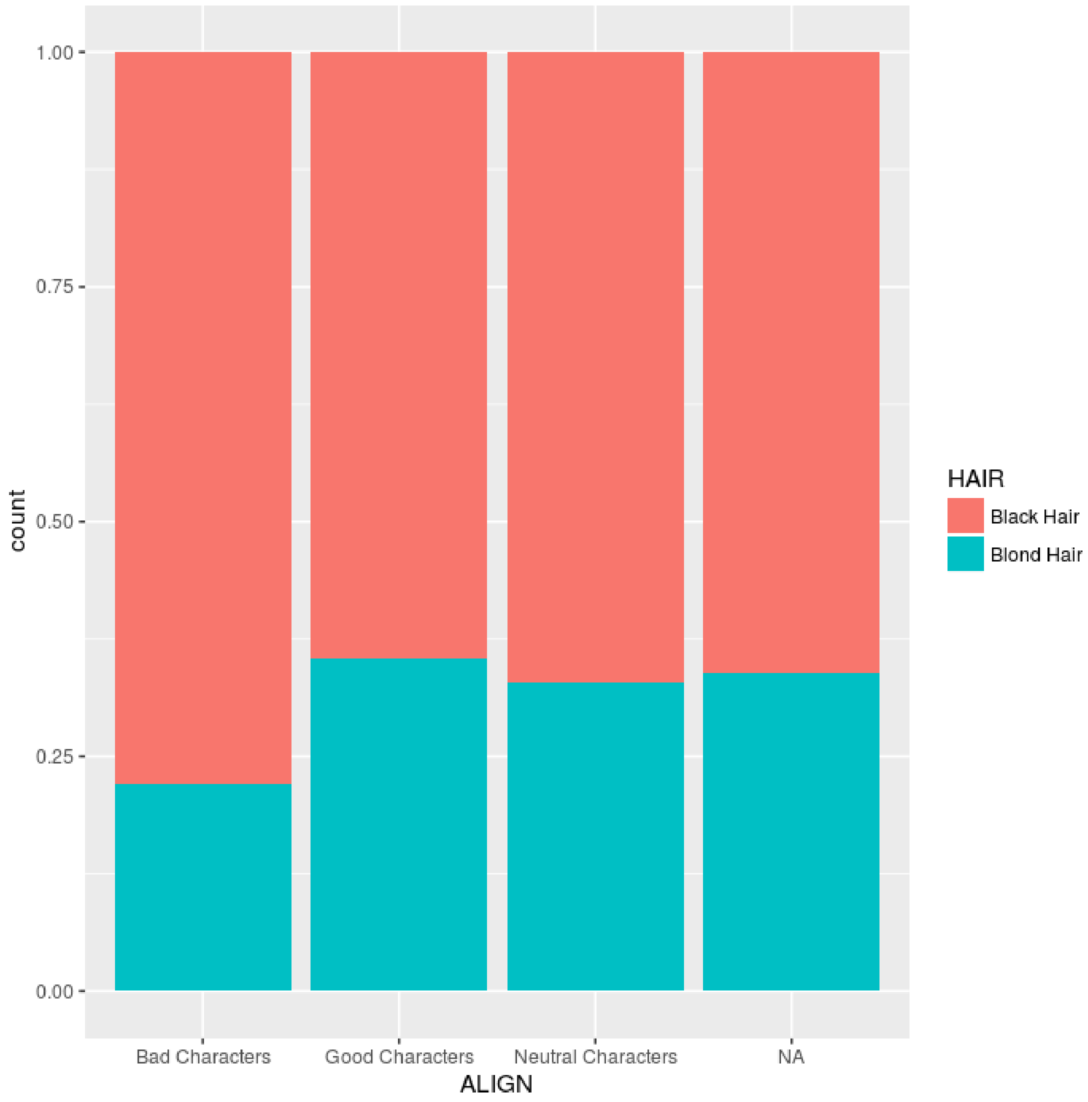

1.00
0.75
0.50
0.25
0.00
count
Bad Characters
Good Characters
Neutral Characters
NA
ALIGN
HAIR
Black Hair
Blond Hair

Figure 2.9:

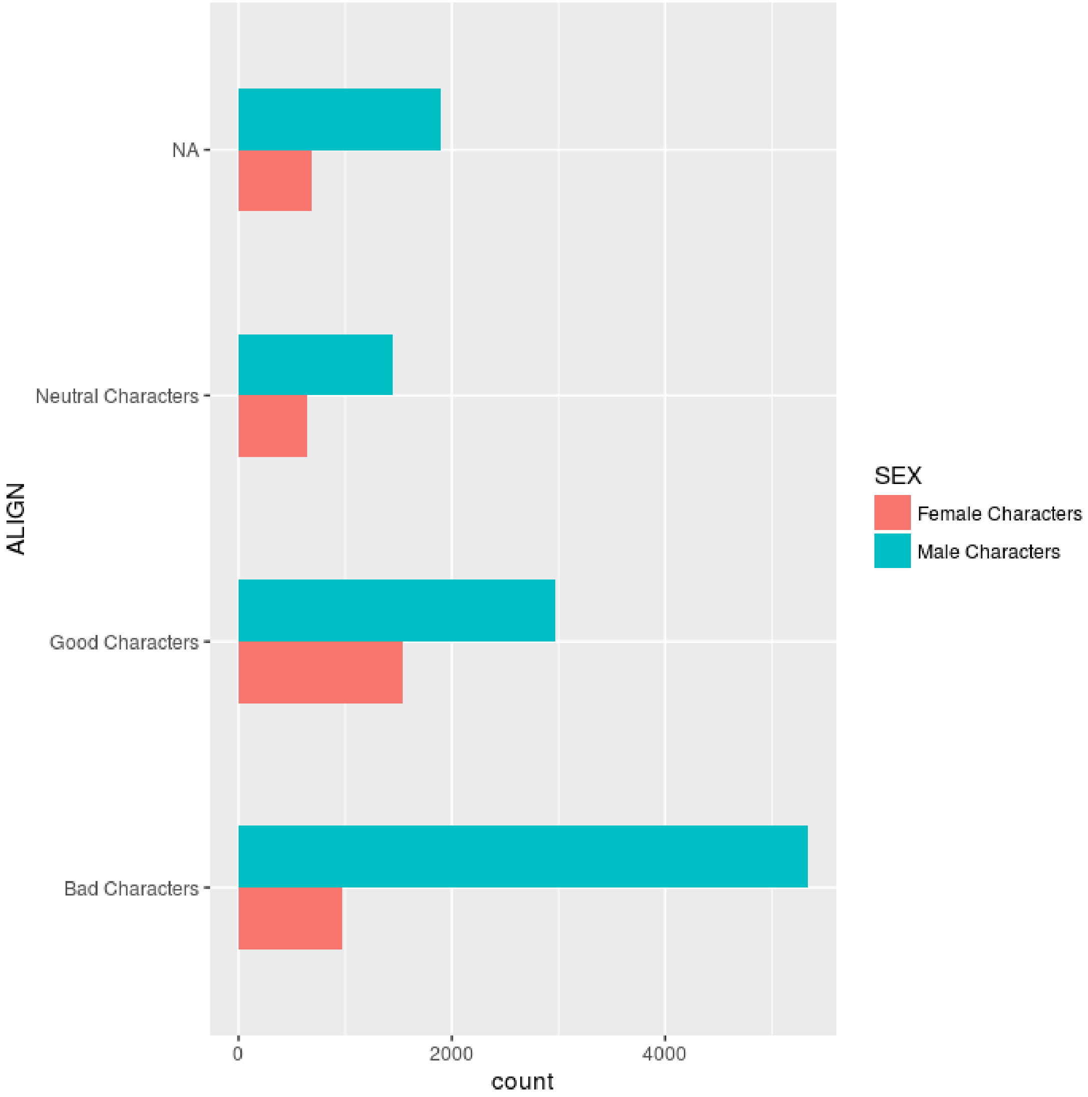

NA
Neutral Characters
Good Characters
Bad Characters
ALIGN
0
2000
4000
count
SEX
Female Characters
Male Characters

Figure 2.10:

2.5 Histogram

2.5.1 Exercise 1

2.5.2 Exercise 2

2.6 Boxplot

2.6.1 Exercise 1

a. Build a boxplot to represent the number of times that each comic character created in 2012 have appeared. Highlight outliers in red and set `outlier.shape=10` and `outlier.size=2`. Choose `fill = #00BFFF` and `color = #00008B`

b. Compare the number of times Bad comic characters and Good comic characters created in 2012 have appeared.

```
ggplot(data=marvel_wikia_data %>% filter(Year == 2012), aes(x = 0, y = APPEARANCES)) +
    geom_boxplot(colour = "#00008B", fill = "#00BFFF", outlier.colour="red", outlier.shape=1
    xlab("")
```

```
## Warning: Removed 27 rows containing non-finite values (stat_boxplot).
```

```
ggplot(data = marvel_wikia_data %>% filter(Year == 2012) %>%
         filter(ALIGN == "Bad Characters" | ALIGN == "Good Characters"),
       aes(x = ALIGN, y = APPEARANCES)) +
    geom_boxplot(colour = "#00008B", fill = "#00BFFF", outlier.colour="red", outlier.shape=1
```

```
## Warning: Removed 25 rows containing non-finite values (stat_boxplot).
```

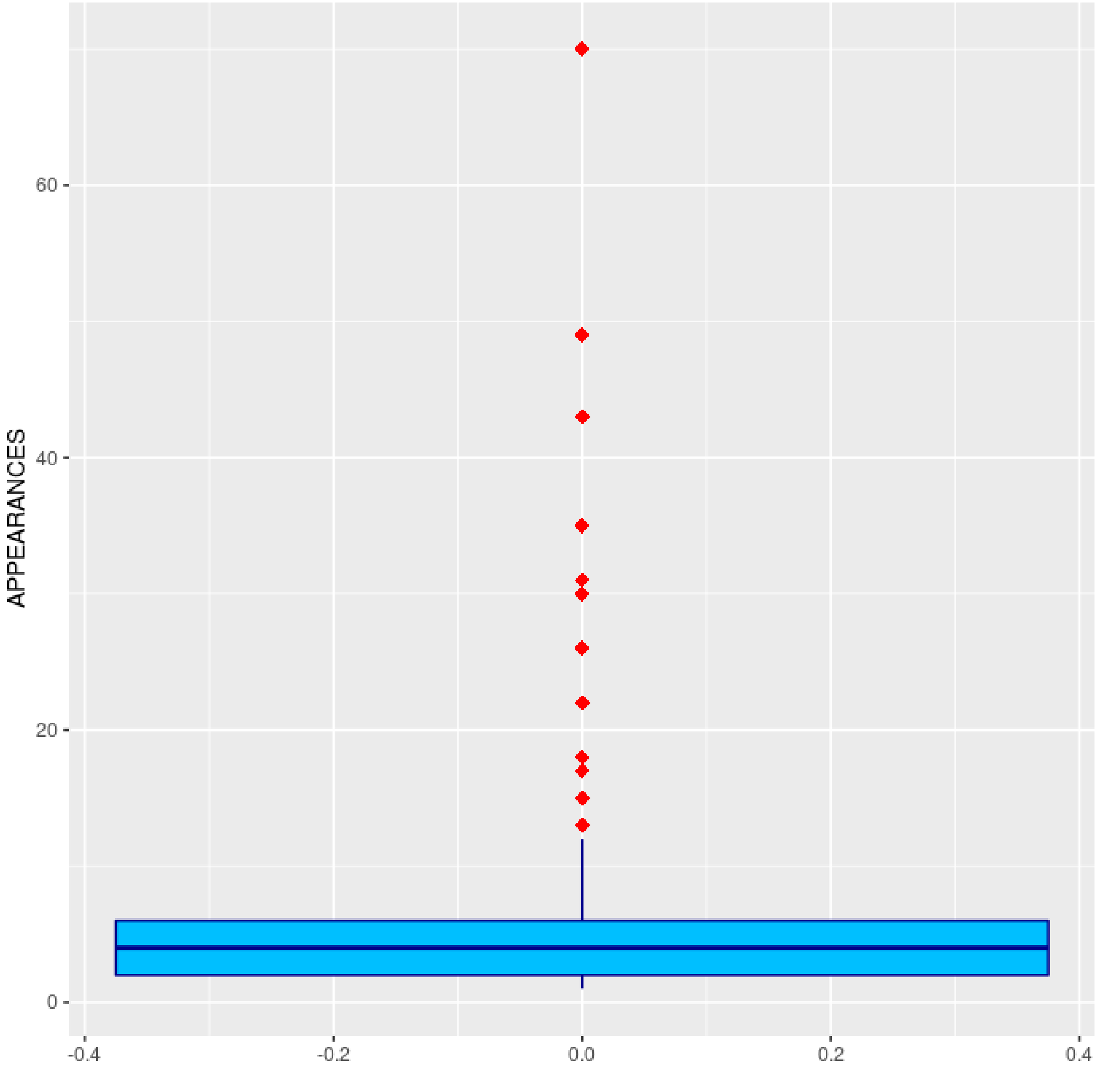


Figure 2.11:

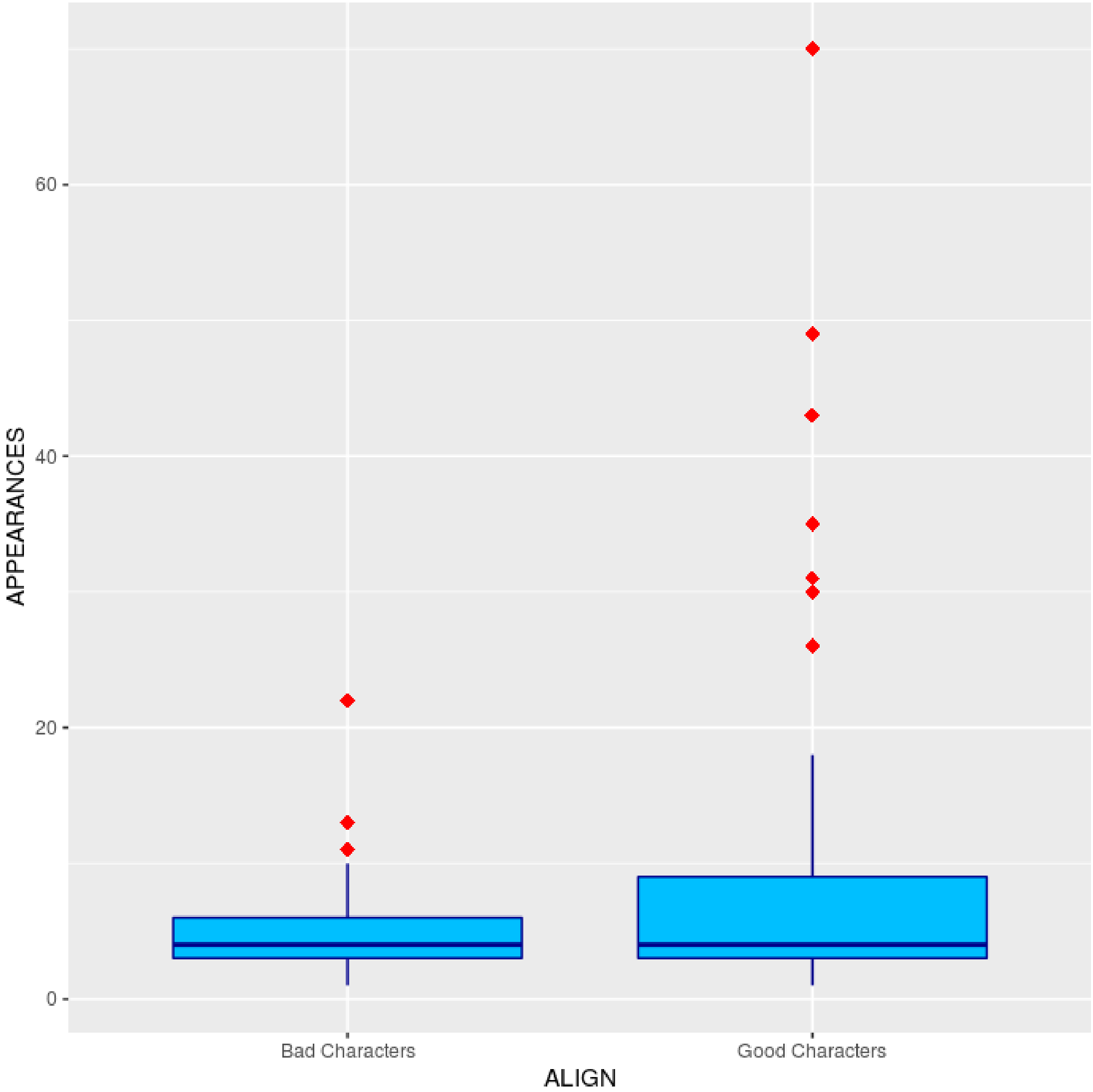

APPEARANCES
60
40
20
0
Bad Characters
Good Characters
ALIGN

Figure 2.12: