Exercises: Data Visualisation

September 5, 2017

Quantide[1]

[1] mailto:

Contents

1 Introduction 5

2 Data Visualization with `ggplot2` 7

2.1 Data . . . 7

2.1.1 iris . . . 7

2.1.2 Comic characters data . . . 8

2.2 Scatterplot . . . 10

2.2.1 Exercise 1 . . . 10

2.2.2 Exercise 2 . . . 12

2.3 Line PLot . . . 14

2.3.1 Exercise 1 . . . 14

2.4 Barplot . . . 18

2.4.1 Exercise 1 . . . 18

2.4.2 Exercise 2 . . . 20

2.5 Histogram . . . 23

2.5.1 Exercise 1 . . . 23

2.6 Boxplot . . . 27

2.6.1 Exercise 1 . . . 27

Chapter 1

Introduction

In this document you will find some exercises about data visualisation. Most of the exercises are composed by some basic data visualisation questions and some questions on advanced topics of data visulisation, labeled as *advanced*.

Chapter 2

Data Visualization with ggplot2

Load ggplot2 package, supposing it is already installed.

```
require(tidyverse)
require(grid)
```

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()
## )
```

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>
```

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 “orchid3”.

```
pl <- ggplot(data = iris, mapping = aes(x=Sepal.Width, y=Sepal.Length)) +
        geom_point(size=3, colour="orchid3", fill="orchid3")
pl
```

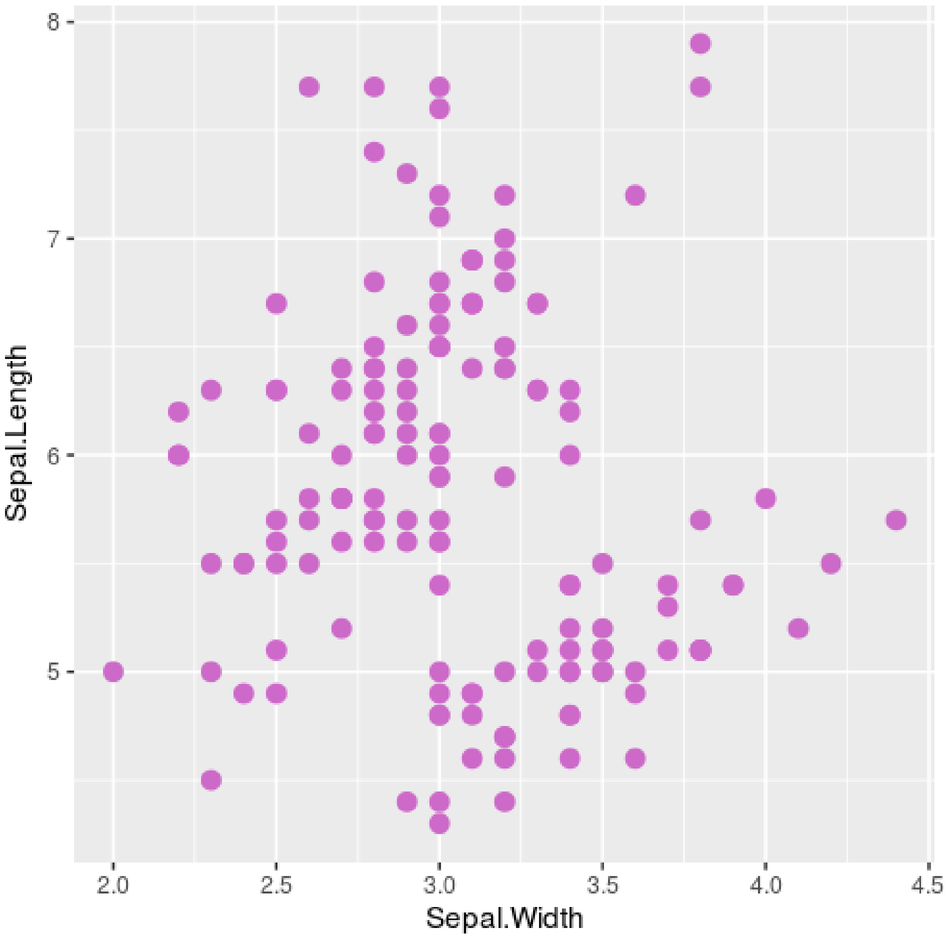


Figure 2.1:

c. *advanced*: Add "Sepal Characteristics" as a red italic title and change axis title to "Sepal length" and "Sepal width".

```
pl + ggtitle("Sepal Characteristics") +
  labs(x = "Sepal width", y = "Sepal length") +
  theme(plot.title=element_text(face="italic", colour="red"))
```

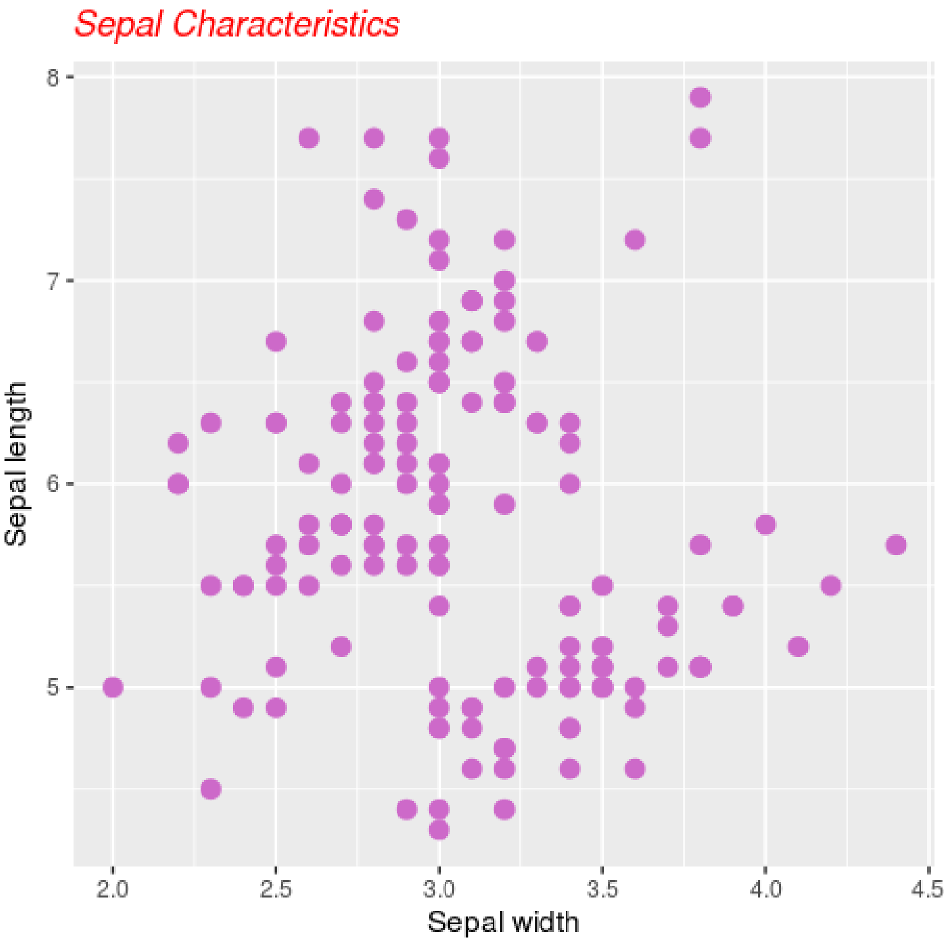


Figure 2.2:

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.

```
pl <- ggplot(data = iris, mapping = aes(x=Sepal.Width, y=Sepal.Length, colour=Species)) +
        geom_point()
pl
```

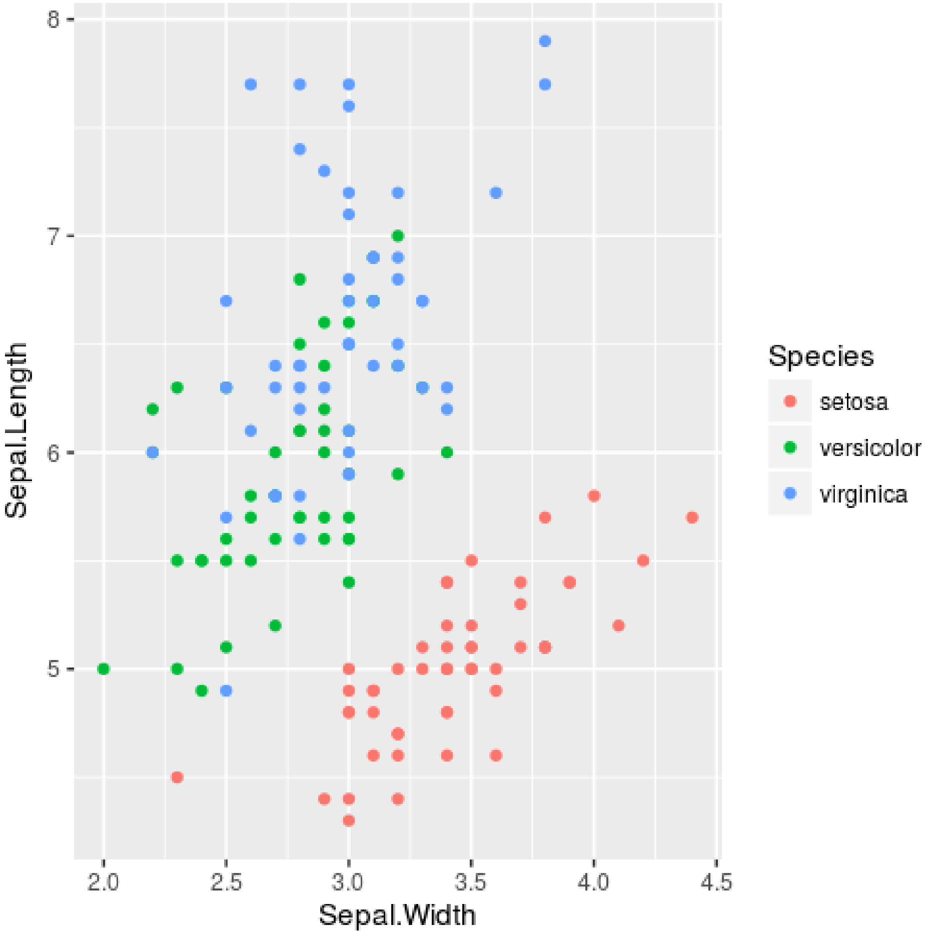


Figure 2.3:

b. *advanced*: Change axis title to “Sepal length” and “Sepal width”.

c. *advanced*: Move the legend to the bottom.

```
pl +
  labs(x = "Sepal width", y = "Sepal length") +
  theme(legend.position="bottom")
```

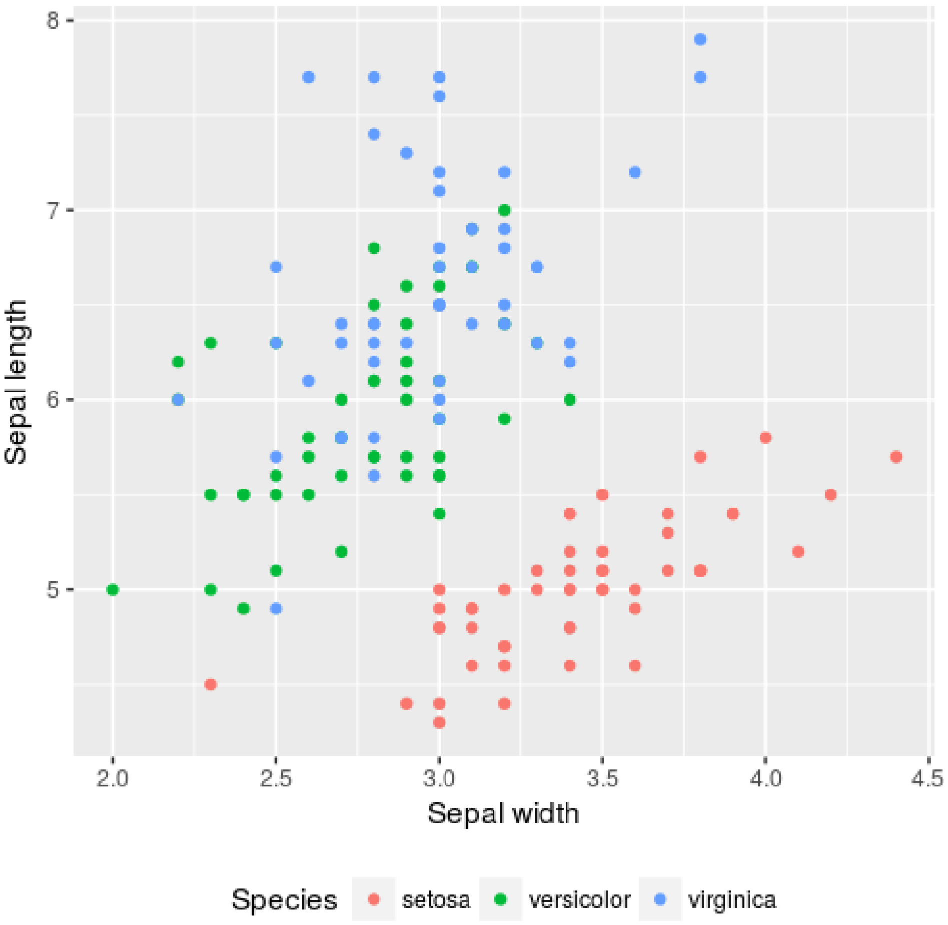

Sepal length
8
7
6
5
2.0
2.5
3.0
3.5
4.0
4.5
Sepal width
Species
setosa
versicolor
virginica

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.

```
number_characters <- marvel_wikia_data %>%
  group_by(Year, SEX) %>%
  summarise(new_char = n()) %>%
  ungroup()

ggplot(data=number_characters, mapping=aes(x=Year, y=new_char, colour= SEX)) +
  geom_line()
```

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

```
ggplot(data=number_characters, mapping=aes(x=Year, y=new_char, colour= SEX)) +
  geom_line(mapping=aes(linetype = SEX)) +
  geom_point(mapping=aes(shape = SEX))
```

d. *advanced*: Choose a blue colour palette to represent the different lines (use the command `scale_colour_brewer( palette = "PuBu" )`)

e. *advanced*: Modify axis names and the key labels with `scale_colour_brewer` choosing options `name = "Characters gender"` and `labels = c("Agender", "Female", "Genderfluid", "Male", "Not available")`.

```
ggplot(data=number_characters, mapping=aes(x=Year, y=new_char, colour= SEX)) +
  geom_line(mapping=aes(colour = SEX)) + labs(x = "Year", y = "Number of characters") +
  scale_colour_brewer(palette="PuBu", name = "Characters gender",
                      labels=c("Agender", "Female", "Genderfluid",
                               "Male", "Not available"))
```

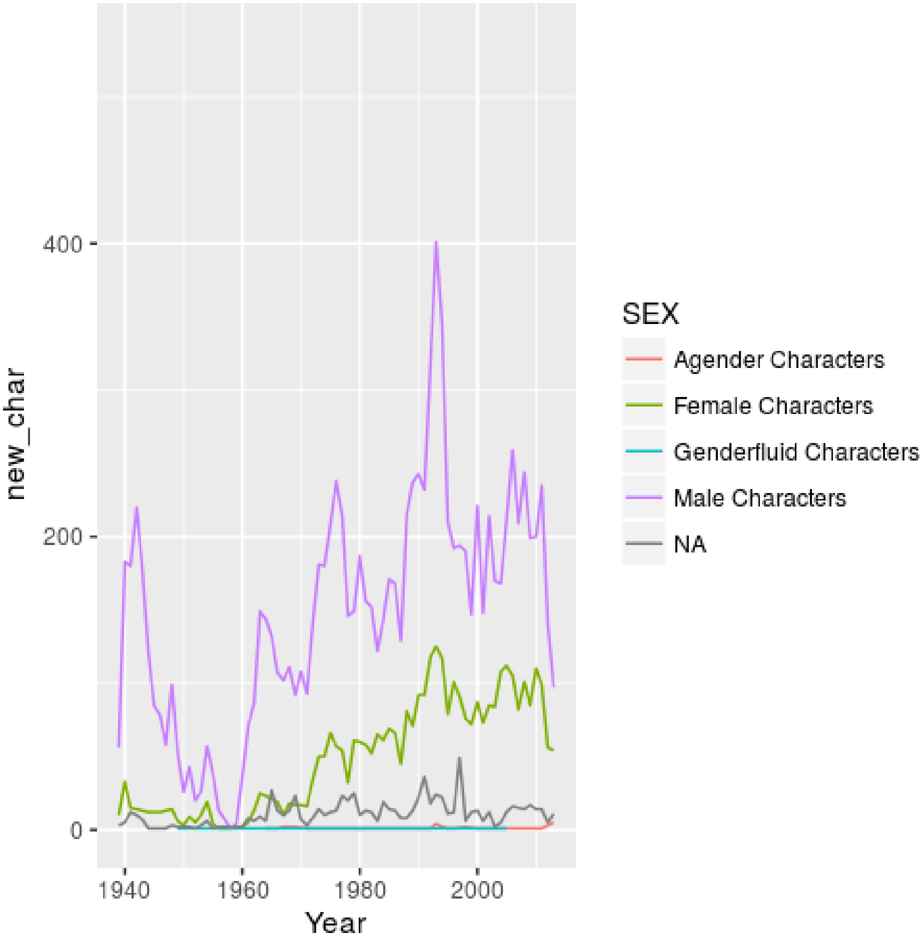
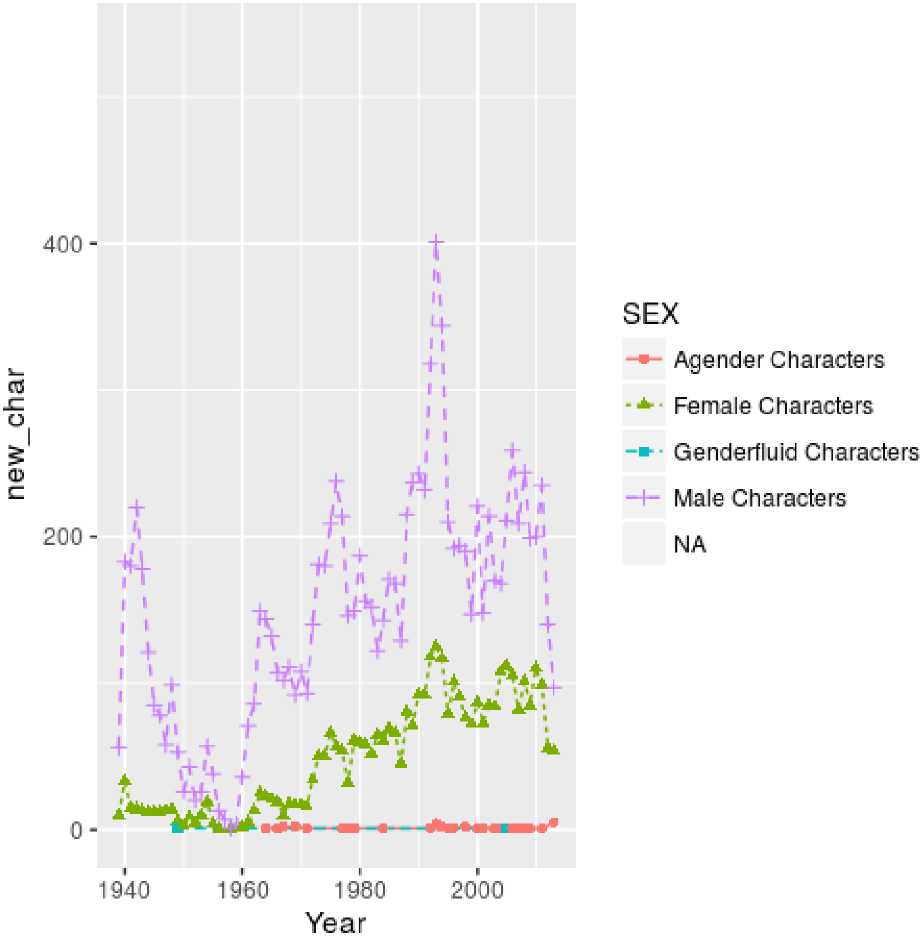

new_char
400
200
0
1940
1960
1980
2000
Year
SEX
Agender Characters
Female Characters
Genderfluid Characters
Male Characters
NA

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:

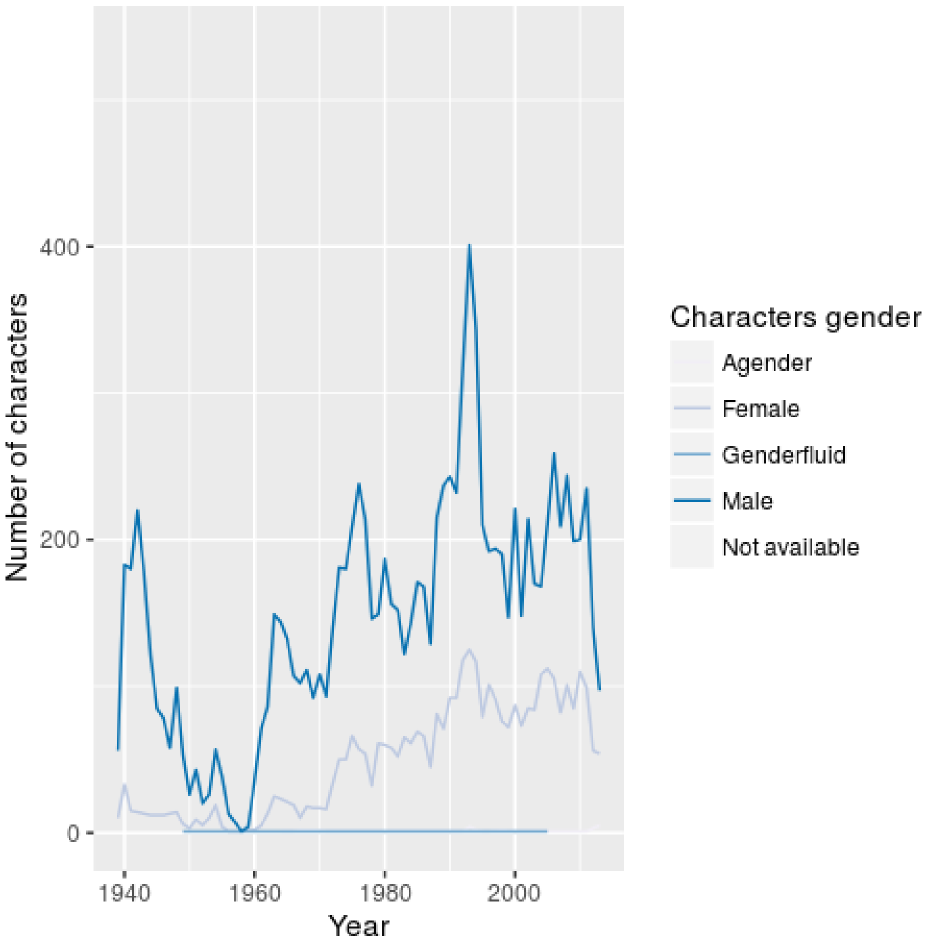

Number of characters
400
200
0
1940
1960
1980
2000
Year
Characters gender
Agender
Female
Genderfluid
Male
Not available

Figure 2.7:

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`. Set bars width as 0.7.

b. *advanced*: Rotate the x axis by 30° so that the axis text nomore overlaps.

```
ggplot(data=marvel_wikia_data, mapping=aes(x=ALIGN, fill=SEX)) +
    geom_bar(width=0.7) +
    theme(axis.text.x = element_text(angle=30, hjust=1, vjust=1))
```

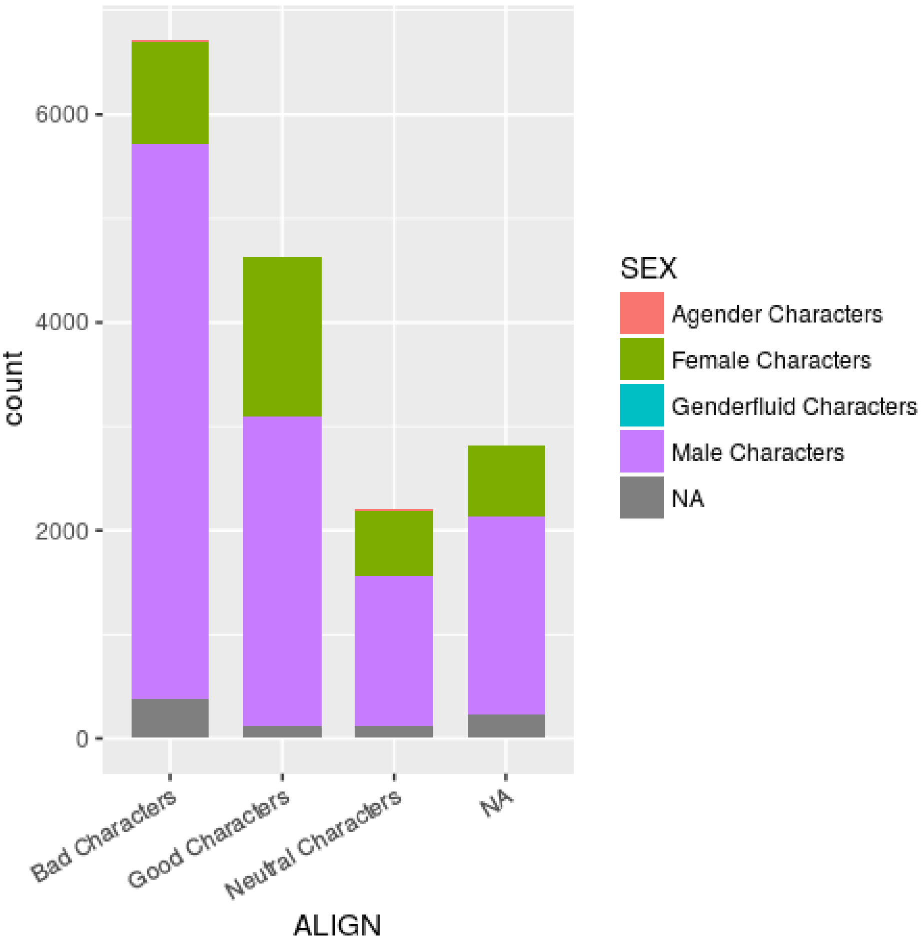


Figure 2.8:

c. 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`.

```
pl <- ggplot(data = marvel_wikia_data %>%
        filter(HAIR == "Black Hair" | HAIR == "Blond Hair"),
      mapping=aes(x=ALIGN, fill=HAIR)) +
   geom_bar(width=0.7) +
   theme(axis.text.x = element_text(angle=30, hjust=1, vjust=1))
pl
```

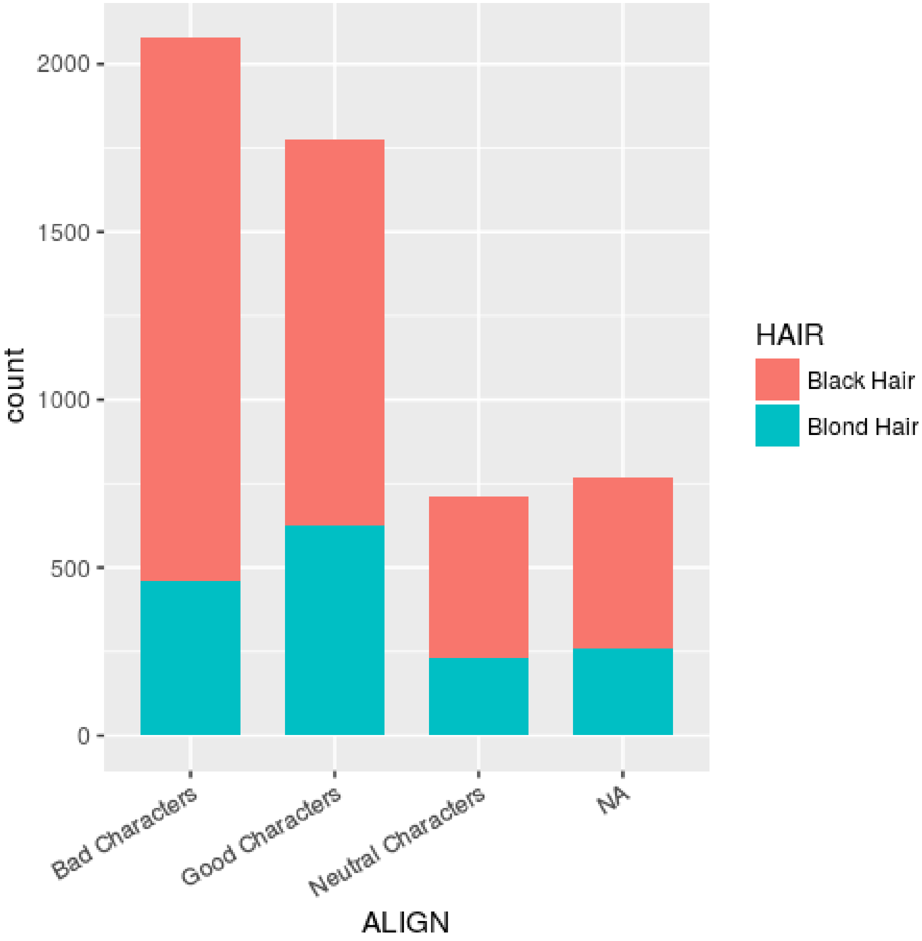


Figure 2.9:

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

e. *advanced*: Manually set colour `grey20` for black hair and `gold3` for blond hair and change the axis name from `ALIGN` to `Character Type`.

```
pl + scale_fill_manual(values=c("grey20", "gold3")) +
    labs(x = "Character type")
```

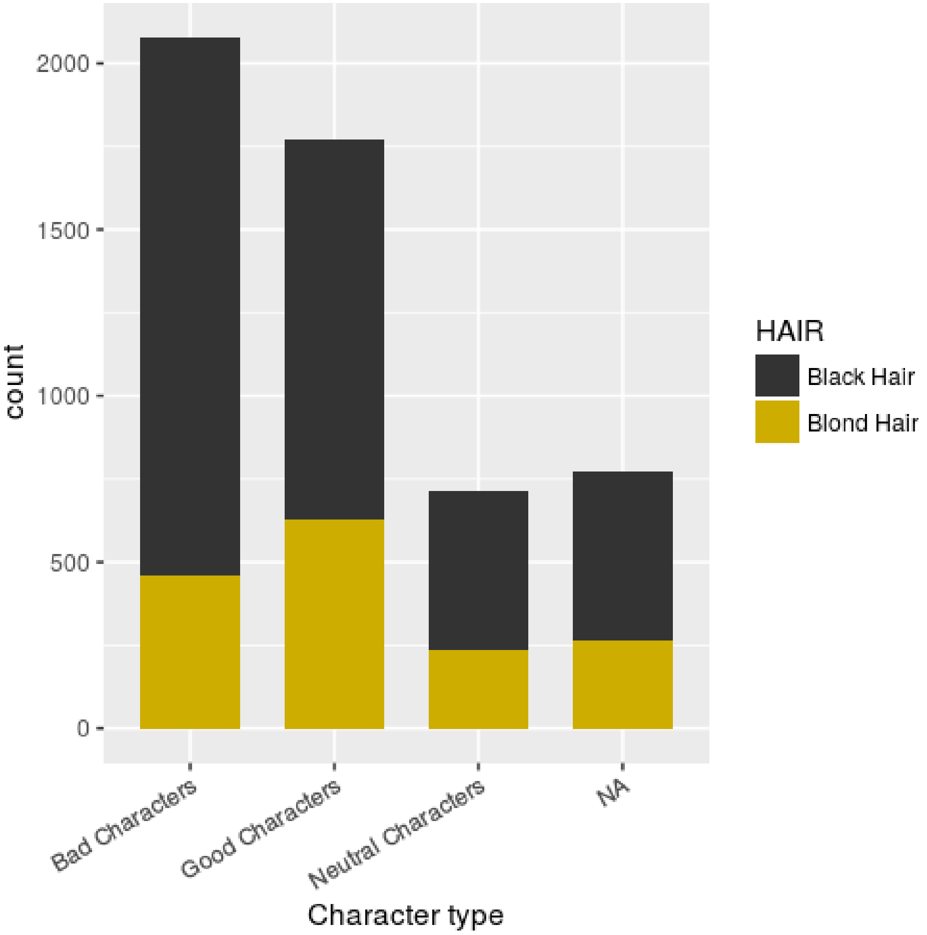


Figure 2.10:

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.

```
ggplot(data = marvel_wikia_data %>%
         filter(SEX == "Male Characters" | SEX == "Female Characters"),
       mapping=aes(x=ALIGN, fill=SEX)) +
  geom_bar(width=0.5, position="dodge")+
  coord_flip() + labs(x = "Character type")
```

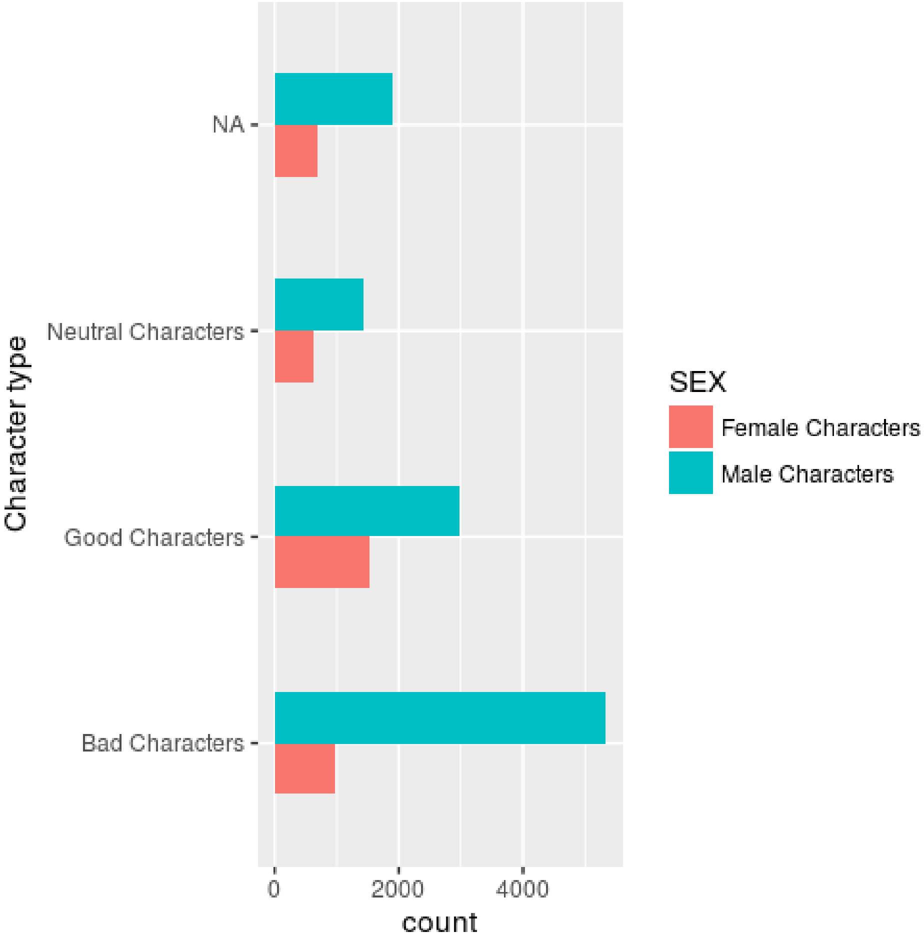


Figure 2.11:

b. *advanced*: Consider only comic characters with blue, black and brown eyes. Set `facet_grid(~ EYE)`

c. *advanced*: Customise legend, axis names and colours so that your plot is as clear as possible (for instance, you may choose colour blue for males and pink for females).

```
data_ex2_barplot <- marvel_wikia_data %>%
         filter(SEX == "Male Characters" | SEX == "Female Characters") %>%
         filter(EYE == "Blue Eyes" | EYE == "Brown Eyes" | EYE == "Black Eyes")

ggplot(data = , data_ex2_barplot, mapping=aes(x=ALIGN, fill=SEX)) +
  geom_bar(width=0.5, position="dodge") +
  labs(x = "Character type") + coord_flip() +
  facet_grid(~EYE) + theme(legend.position="bottom") +
  scale_fill_manual(values=c("lightpink1", "lightblue1"),
                    labels = c("Male", "Female"), name = c("Characters gender"))
```

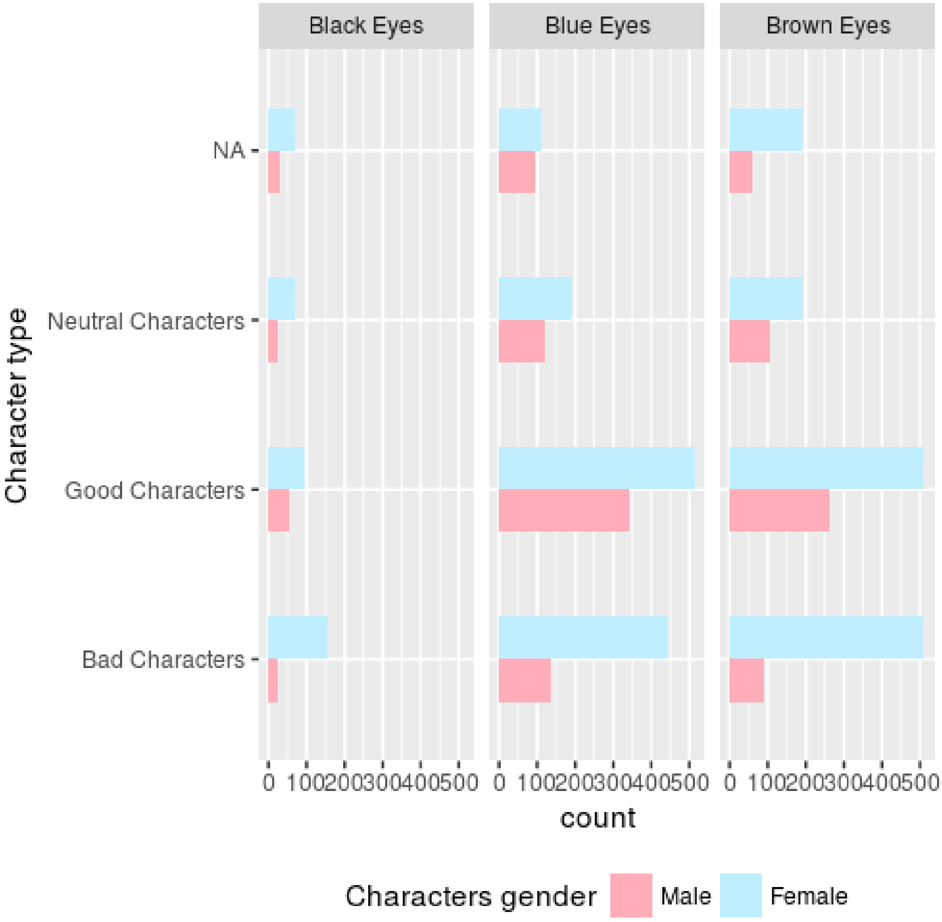

Black Eyes
Blue Eyes
Brown Eyes
NA
Neutral Characters
Good Characters
Bad Characters
Character type
0
100
200
300
400
500
count
Characters gender
Male
Female

Figure 2.12:

2.5 Histogram

Let us consider iris dataset.

2.5.1 Exercise 1

a. Represent the distribution of Sepal_Length variable with an histogram.

b. Set bins fill colour as "hotpink" and bins line colour as "deeppink".

c. Set the number of bins as 15.

```
pl <- ggplot(data=iris, aes(x=Sepal.Length)) +
    geom_histogram(fill="hotpink", colour="deeppink", bins=15)
pl
```

d. *advanced*: Map the grouping variable Species to fill and choose a pink colour palette (PuRd)

```
ggplot(data=iris, aes(x=Sepal.Length, fill = Species)) +
  geom_histogram(bins = 14, colour="deeppink") +
  scale_fill_brewer(palette="PuRd")
```

e. *advanced*: Using facet_grid() produce a different panel for each Species

```
pl +
  facet_grid(Species ~ .)
```

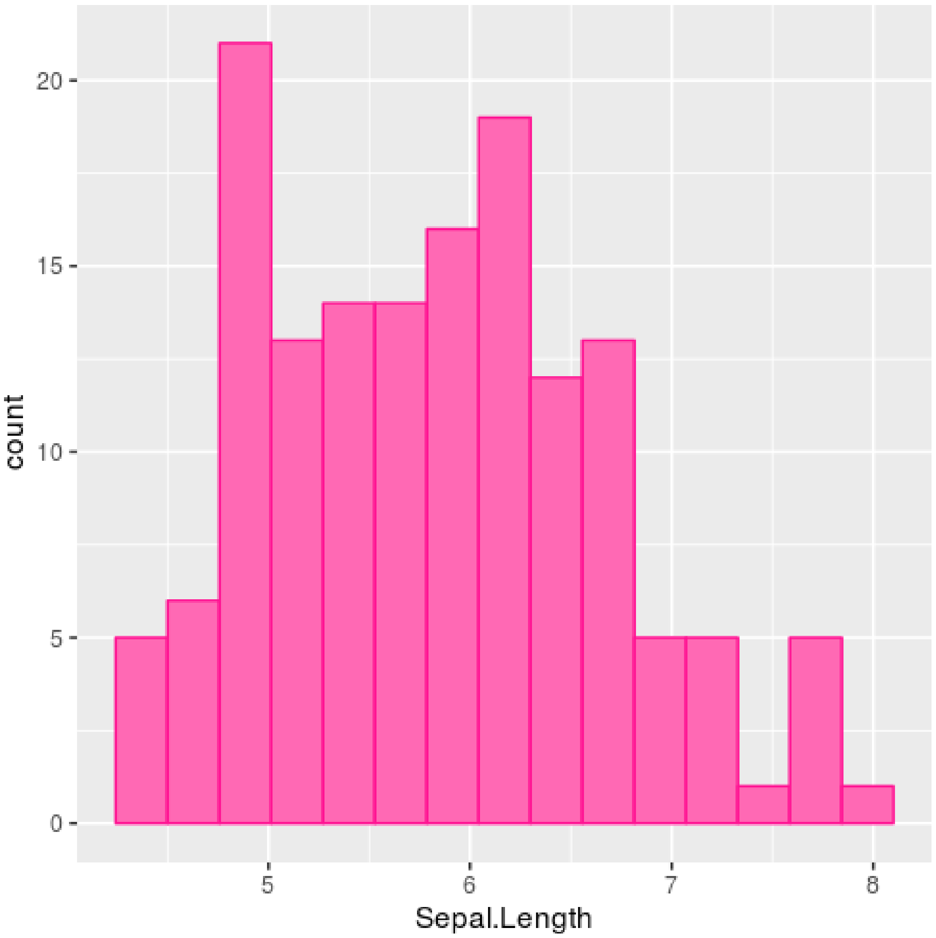


Figure 2.13:

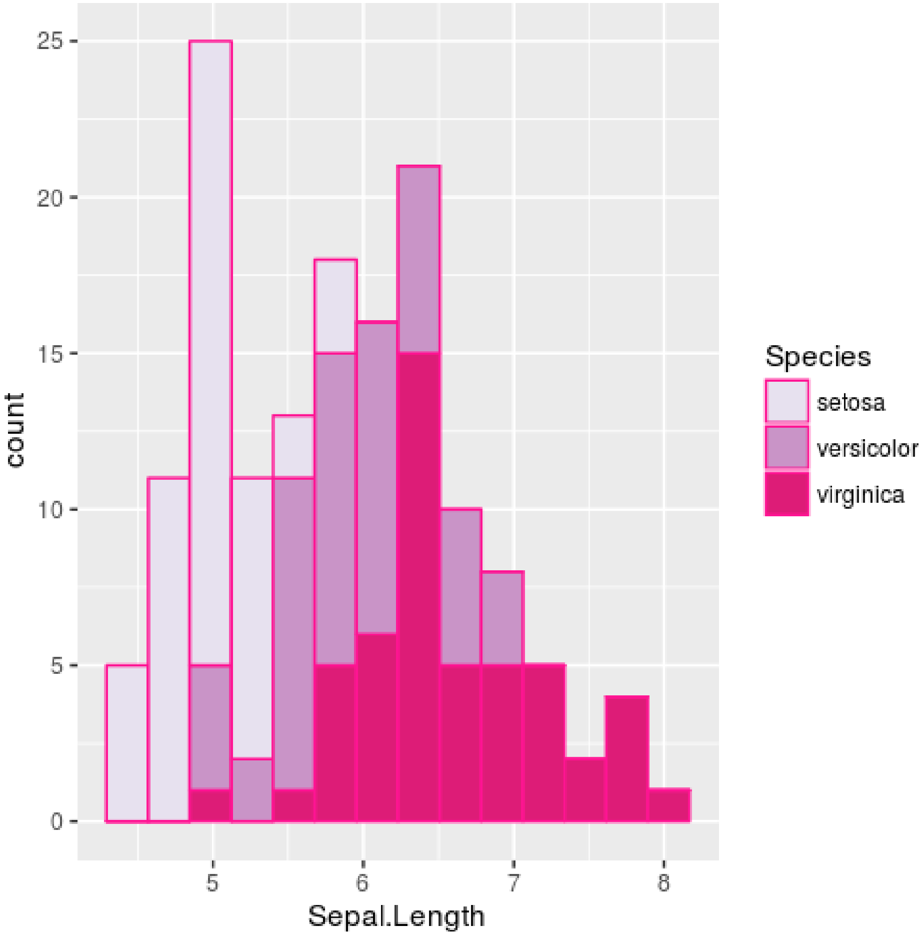

25
20
15
10
5
0
count
5
6
7
8
Sepal.Length
Species
setosa
versicolor
virginica

Figure 2.14:

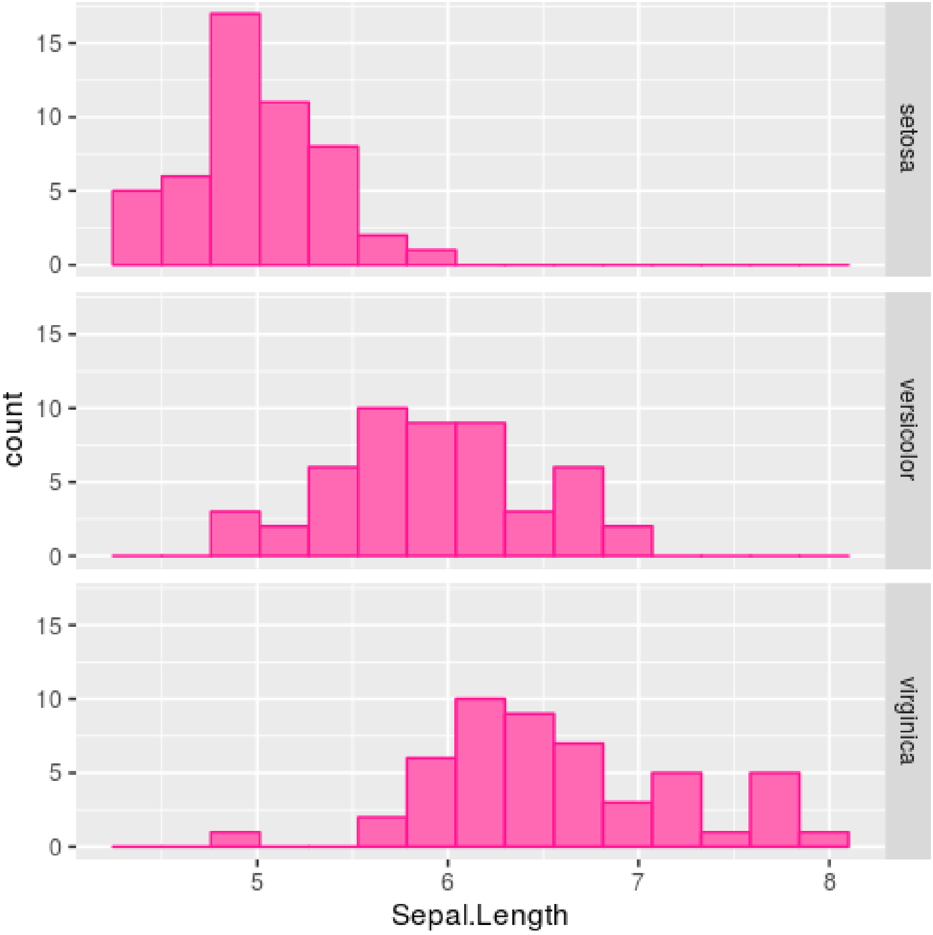

15
10
5
0
setosa
versicolor
virginica
count
5
6
7
8
Sepal.Length

Figure 2.15:

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 = "aquamarine2"` and `color = "aquamarine4"`

```
ggplot(data=marvel_wikia_data %>% filter(Year == 2012), aes(x = 0, y = APPEARANCES)) +
  xlab("") + geom_boxplot(colour = "aquamarine4", fill = "aquamarine2",
  outlier.colour="red", outlier.shape=10, outlier.size=2)
```

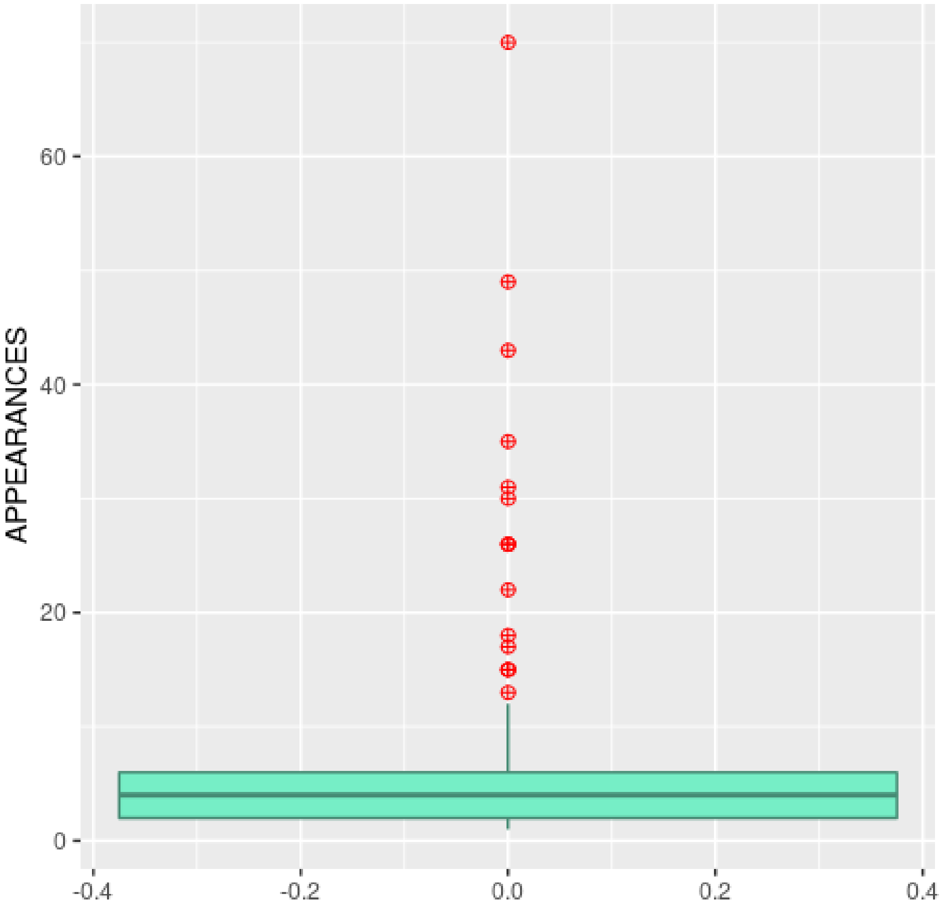


Figure 2.16:

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

```
data_boxplot <- marvel_wikia_data %>% filter(Year == 2012) %>%
         filter(ALIGN == "Bad Characters" | ALIGN == "Good Characters")

ggplot(data = data_boxplot, aes(x = ALIGN, y = APPEARANCES)) +
  labs(x = "Character type") +
  geom_boxplot(colour = "aquamarine4", fill = "aquamarine2",
  outlier.colour="red", outlier.shape=10, outlier.size=2)
```

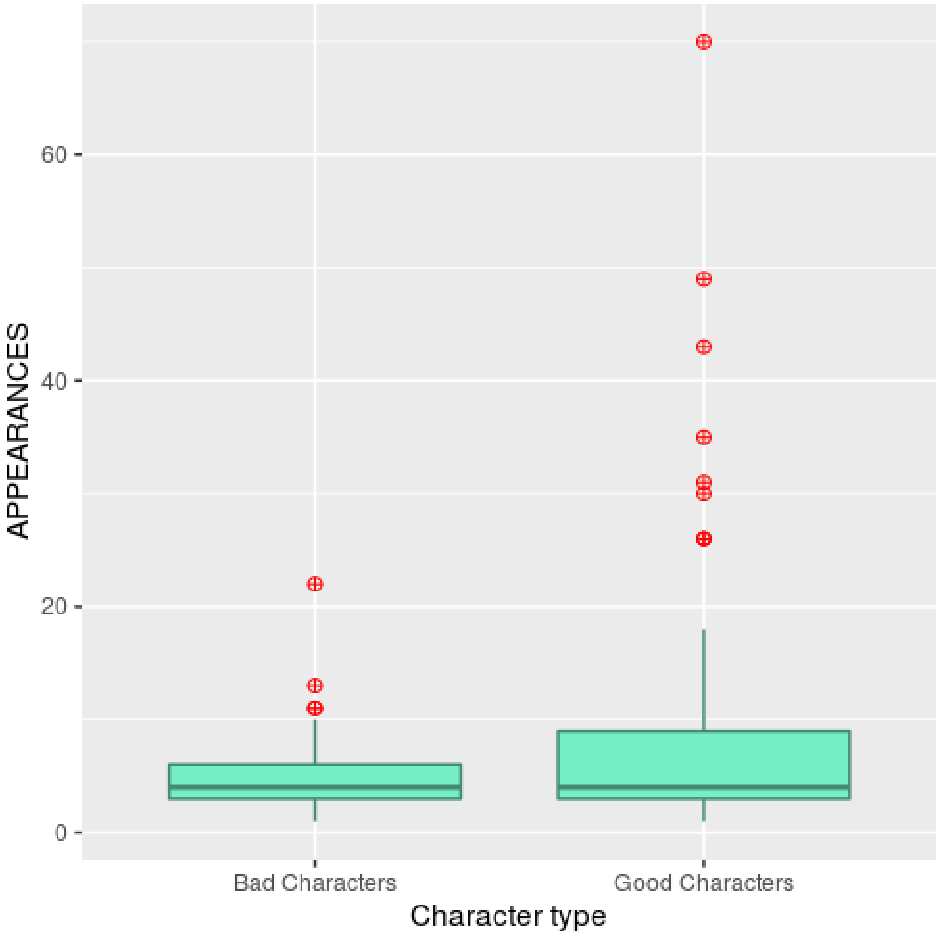


Figure 2.17: