# System Analysis & Design Assignment 2

Rhichard Koh

2023-03-11

```
library(tinytex)
library(plotly)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.0     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ lubridate 1.9.2     ✓ tibble    3.2.0
## ✓ purrr     1.0.1     ✓ tidyr     1.3.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks plotly::filter(), stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the  ]8;;http://conflicted.r-lib.org/ conflicted package ]8;;  to force all conflicts t
o become errors
```

```
library(ggplot2)
```

```
my.data <- read.csv('./netflix_titles.csv')
my.data
```

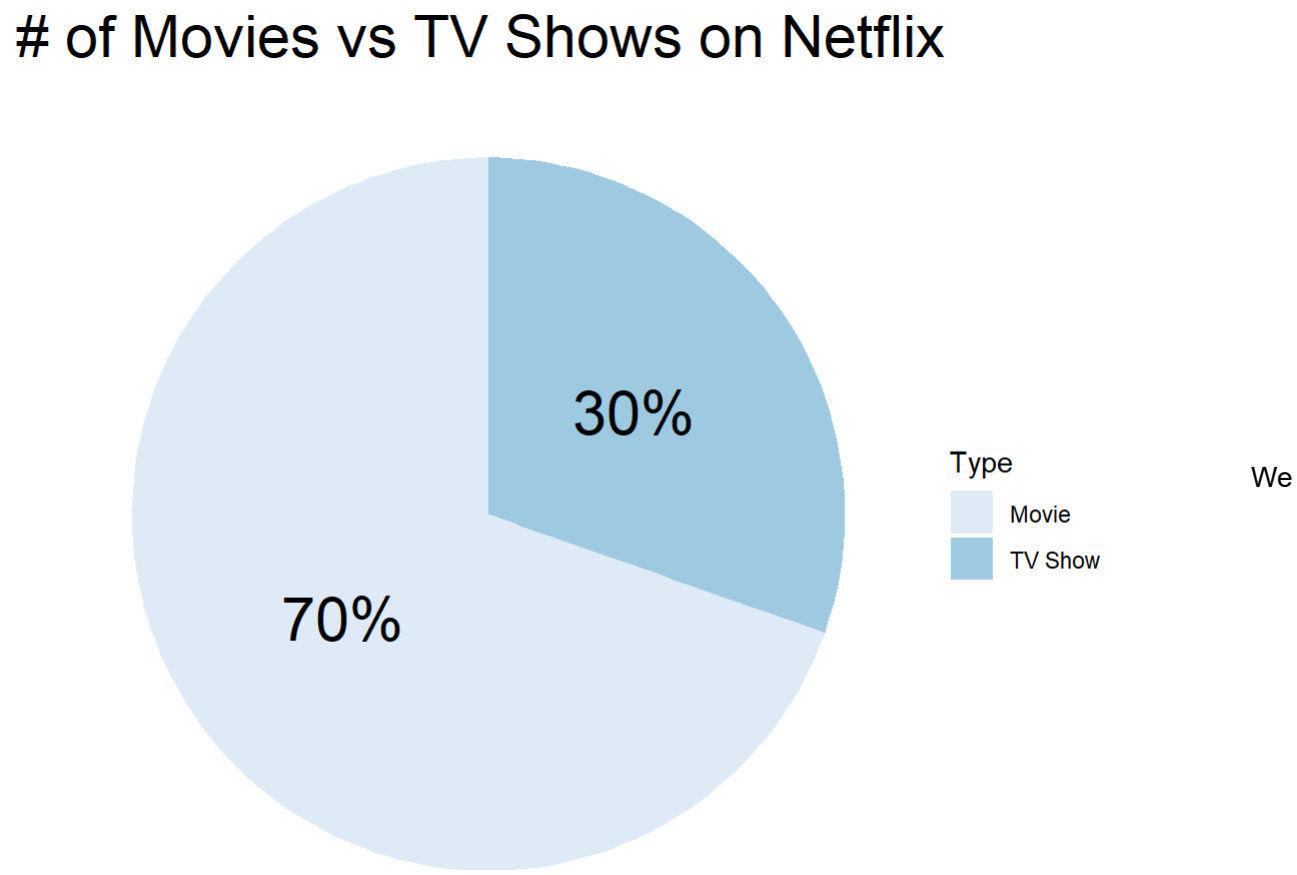
| show… <chr> | type <chr> | title <chr> |
|---|---|---|
| s1 | Movie | Dick Johnson Is Dead |
| s2 | TV Show | Blood & Water |
| s3 | TV Show | Ganglands |
| s4 | TV Show | Jailbirds New Orleans |
| s5 | TV Show | Kota Factory |
| s6 | TV Show | Midnight Mass |
| s7 | Movie | My Little Pony: A New Generation |
| s8 | Movie | Sankofa |
| s9 | TV Show | The Great British Baking Show |
| s10 | Movie | The Starling |

1-10 of 8,807 rows | 1-3 of 12 columns Previous **1** 2 3 4 5 6 … 881 Next

First Data Visualization Technique – Pie Chart

```
data <- my.data %>%
        group_by(type) %>%
        summarize(counts = n(),
                  percentage = n()/nrow(my.data))

ggplot(data, aes(x = "", y=percentage, fill = type)) +
  geom_bar(width = 1, stat = "identity") +coord_polar(theta = "y", start=0)+
scale_fill_brewer(palette="Blues")+
  labs(fill="Type",
       x=NULL,
       y=NULL,
       title="# of Movies vs TV Shows on Netflix ") +
geom_text(aes(label = paste0(round(percentage*100),'%')),size=8, position = position_stack(vjust
= 0.5))+
theme_void()+theme(plot.title = element_text(hjust=0.5,size=22))
```

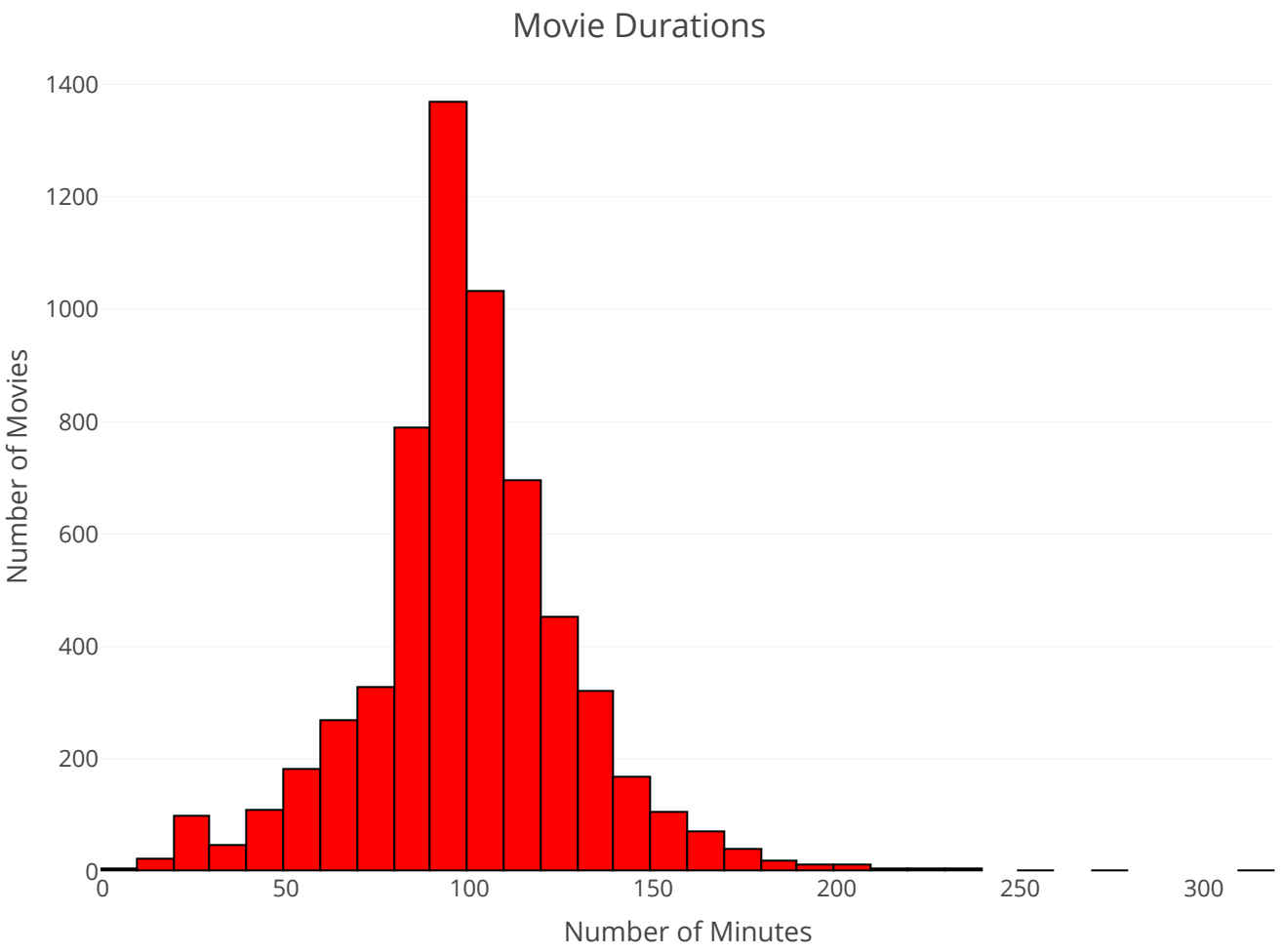



can see that 70% of listings on Netflix are Movies and only 30% are TV Shows.which makes sense because every movie gets its own title however every tv show has many episodes and seasons under the same title.

Second Data Visualization Technique – Histogram

```
movies <- my.data %>% select(type, duration) %>%
  filter(type == "Movie") %>%
  drop_na() %>%
  mutate(mins = parse_number(duration))
movies %>%
  plot_ly(
    x = ~ mins,
    type = "histogram",
    nbinsx = 40,
    marker = list(
      color = "red",
      line = list(color = "black",
                  width = 1))
  ) %>%
  layout(
    title = "Movie Durations",
    yaxis = list(title = "Number of Movies",
                 zeroline = FALSE),
    xaxis = list(title = "Number of Minutes",
                 zeroline = FALSE))
```

```
## Warning: Ignoring 3 observations
```

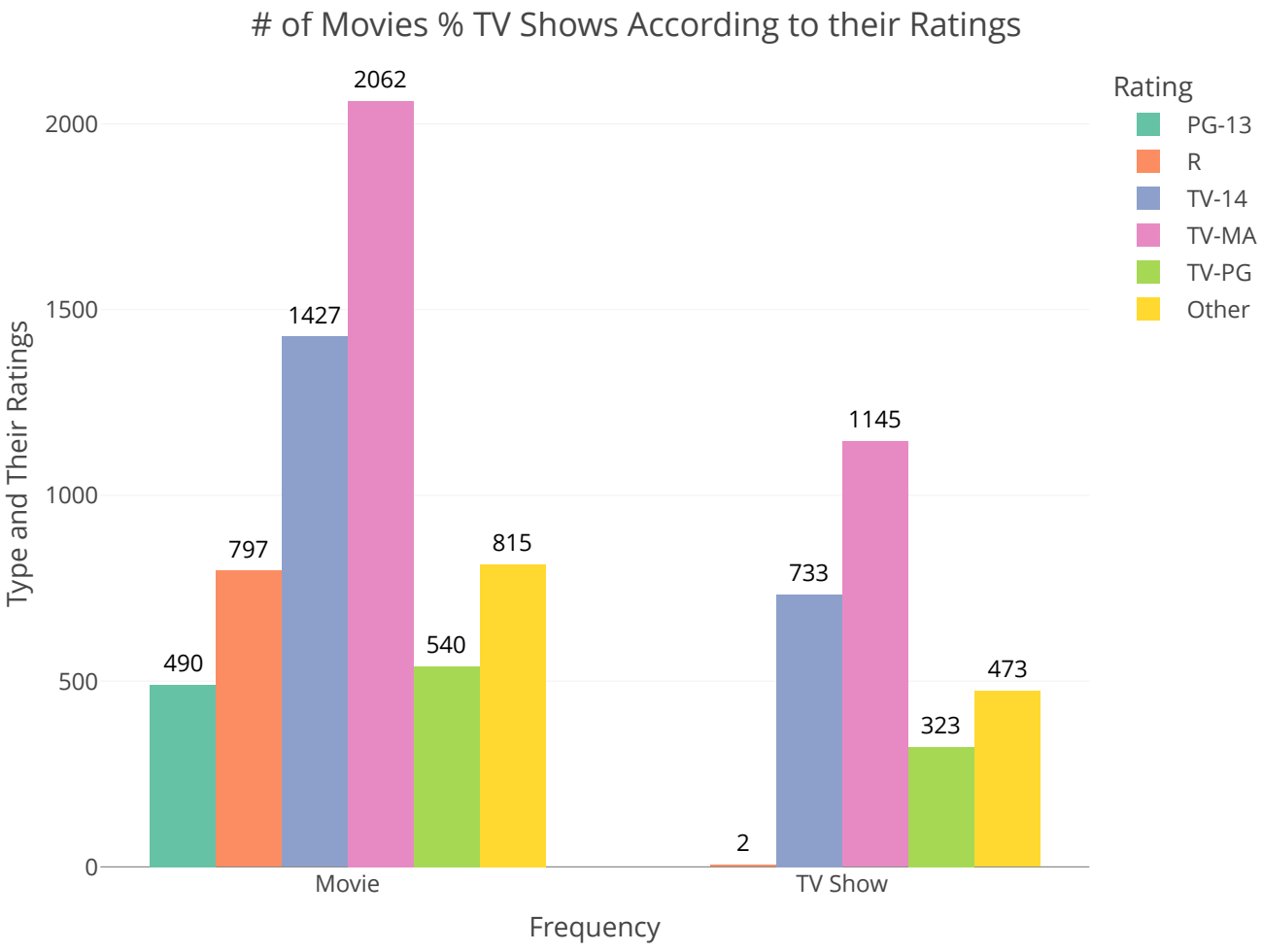



We can see the the the highest frequency of movies are about 90-99 mins long. The histogram is also left skewed.

Second Data Visualization Technique – Bar Graph

```
my.data %>% select(rating, type) %>%
  filter(!is.na(rating)) %>%
  mutate(rating = fct_lump(rating, 5)) %>%
  group_by(rating, type) %>%
  summarise(freq = n()) %>%
  arrange(freq) %>%
  plot_ly(x = ~ type ,
          y = ~ freq,
          type = "bar",
          color = ~ rating,
          text = ~ freq,
          textposition = 'outside',
          textfont = list(color = 'black', size = 12)) %>%
  layout(yaxis = list(categoryorder = "array",
                      categoryarray = ~ freq)) %>%
  layout(
    title = "# of Movies % TV Shows According to their Ratings",
    yaxis = list(title = "Type and Their Ratings"),
    xaxis = list(title = "Frequency"),
    legend = list(title = list(text = 'Rating')))
```

```
## `summarise()` has grouped output by 'rating'. You can override using the
## `.groups` argument.
```



We can see the TV-MA is the most popular between both Movies and TV Shows. The least Popular is PG-13 for Movies and R for TV Shows, Which makes sense because you are normally not allowed to show R things on television.