SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTING

BIG DATA AND DATA ANALYTICS
LAB PROJECT 2
(DATASET DESCRIPTION & PREPARATION)

This lab project is based on a dataset about movie success in 2014 and 2015 by Ahmad et al. (2015) which is available on the online platform by Lichman et al (2013). Download the file movidata.csv from Blackboard and then practice the following topics in preparations for Lab Project 2.

PREPARATION IN WEEK 4

In preparation for Lab Project 2, load the moviedata.csv dataset and run the following command to create factor variable in the moviedata dataframe.

```
moviedata$sequelcat <- factor(moviedata$dummy_sequel, levels = c(0, 1),
  labels = c("ORIGINAL", "SEQUEL"))

moviedata$year <- factor(moviedata$year, levels = c(2014, 2015),
  labels = c("Y2014", "Y2015"))
```

Use the moviedata dataset to practice the following topics:

- Create box plots, violin plots, bar charts, and scatter plots
- Add summary statistics on top of existing plots
- Use to `install.packages("scales")` install the library "scales" and use it to display the gross movie budget in US dollars using different charts by adding the following to ggplot commands:
  - `+ scale_y_continuous(labels = dollar)`
- Remove NA value with !is.na(…)
- Conduct the following tests: *t*-test, ANOVA, Tukey HSD, Mann-Whitney U-Test
- Testing for variance homogeneity
- Find out more about the function "subset()"
- Study closely how the custom winsorising function works that we discussed in week 3. Then, use the custom winsorising function to winsorise data.
- Try different colour palettes in the "RColorBewer" and the "wesanderson" package
  - https://github.com/karthik/wesanderson
  - Use the display.brewer.all() to see the RColorBrewer palettes
- Create new variables in a dataset based on existing variables and explore the "as.factor()" function to convert a character variable into a factor. For example:

```
autodata$cylindercat[autodata$cylinders >= 3 & autodata$cylinders <= 5] <- "3 to 5 cylinders"
autodata$cylindercat[autodata$cylinders >= 6 & autodata$cylinders <= 8] <- "6 to 8 cylinders"
autodata$cylindercat <- as.factor(autodata$cylindercat)
```

- Search online for information on the function "ddply" of the package "plyr" (also have a look at ?ddply). This function is useful for data aggregation. Here are two examples:

```
ddply(autodata, c("origin"), summarise, N = length(weight), weight_avg=mean(weight),
  weight_sd=sd(weight), mpg_avg=mean(mpg), mpg_sd=sd(mpg))

ddply(autodata, c("origin", "cylindercat"), summarise, N=length(weight),
  weight_avg=mean(weight), weight_sd=sd(weight), mpg_avg=mean(mpg), mpg_sd=sd(mpg))
```

Note: You first need to install and then load the plyr package to use the ddply() function.

REFERENCES

Ahmed M, Jahangir M, Afzal H, Majeed A, Siddiqi I. Using Crowd-source based features from social media and Conventional features to predict the movies popularity. In Smart City/ SocialCom/S ustainCom (SmartCity), 2015 IEEE International Conference on 2015 Dec 19 (pp. 273-278). IEEE. https://ieeexplore.ieee.org/document/7463737

Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

DATASET

moviedata	*Conventional and Social Media Movies 2014 and 2015*

Description

A dataset about the success of movies in 2014 and 2015.

Usage

moviedata

Format

A data frame with 231 observations on the following 14 variables.

`movie`	Name of the movie
`year`	Year of movie release
`ratings`	Rating of the movie (0 – 10)
`genre`	Identifier for the genre of the movie (e.g., action, adventure, drama)
`gross`	Gross world-wide income from the movie (in US$)
`budget`	Budget for the movie
`screens`	Number of screens that the movie was initially launched in on the opening weekend in the US
`sequel`	A number indicating whether the movie is sequel or original (individual) movie, where higher numbers indicate later sequels in a series. For instance, for Mission Impossible a sequel value of 5 indicates that this is the fifth movie in the series.
`dummy_sequel`	0 – Original movie 1 – Sequel movie
`sentiment`	A sentiment score assessed through an analysis of tweets about the movie on Twitter. 0 represents a neutral sentiment, a positive value represents a positive sentiment, and a negative value indicates a negative sentiment. The sentiment score for each movie was calculated by retrieving all tweets related to each movie, assigning the sentiment score to each of them and then aggregating the score.
`views`	Number of times the movie trailer was viewed on YouTube
`likes`	Number of likes the movie trailer received on YouTube
`dislikes`	Number of dislikes the movie trailer received on YouTube
`comments`	Number of times the movie trailer received a comment on YouTube
`aggregate_followers`	The aggregate number of actor followers: Equal to sum of followers of top 3 cast from Twitter

Source

Ahmed M, Jahangir M, Afzal H, Majeed A, Siddiqi I. Using Crowd-source based features from social media and Conventional features to predict the movies popularity. In Smart City/ SocialCom/S ustainCom (SmartCity), 2015 IEEE International Conference on 2015 Dec 19 (pp. 273-278). IEEE. https://ieeexplore.ieee.org/document/7463737

Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.