WESTERN SYDNEY
UNIVERSITY

Assignment for Programming for Data Science

Subject Code: COMP7024
Name: Pengda Yang
Student ID: 22154543
Pages: 10

Declaration:
Name: Pengda Yang

Student ID: 22154543

Subject Code: COMP7024

By including this statement, I, the author of this work, verify that:

☐ You hold a copy of this submission if the original is lost or damaged.
☐ No part of this submission has been copied from any other student's work or from any other third party (including generative AI) except where due acknowledgment is made in the submission.
☐ No part of this submission has been submitted by you in another (previous or current) assessment, except where appropriately referenced, and with prior permission from the teacher/tutor/supervisor/Subject Coordinator for this subject.
☐ No part of this submission has been written/produced for you by any other person or technology except where collaboration has been authorised by the teacher/tutor/supervisor/Subject Coordinator either in the assessment resources section of the Learning Guide for this assessment task, in the instructions for this assessment task, or through vUWS.
☐ You are aware that this submission will be reproduced and submitted to detection software programs for the purpose of investigating possible breaches of the Student Misconduct Rule, for example, plagiarism, contract cheating, or unauthorised use of generative AI. Turnitin or other tools of investigation may retain a copy of the submission for the purposes of future investigation.
☐ You will not make this submission available to any other person unless required by the University.
☐ You hereby certify that you have read and understand what the School of Computing, Engineering and Mathematics defines as minor and substantial breaches of misconduct as outlined in the subject outline for this subject.

Data Processing

Input data sets

```
library(data.table)
books <- fread("books_new.csv", sep = ";", header = TRUE, fill = TRUE)
```

Since the data in the file is separated by semicolons, it is necessary to process the semicolons as well as set the header and automatically fill the null value.

For the original data, I have filled in any missing double quotes around the data content to ensure that R can read it correctly, while keeping the content and quantity of the data unchanged.

```
RatingPGA = read.csv('RatingPGA.csv')
RatingPGB = read.csv('RatingPGB.csv')
users = read.csv('users.csv')
```

Q1

```
library(tidyverse)

head(RatingPGA)

##    X   User       ISBN Book.Rating
## 1  1 276725 034545104X           0
## 2  6 276733 2080674722           0
## 3 11 276746 0425115801           0
## 4 12 276746 0449006522           0
## 5 17 276747 0060517794           9
## 6 20 276747 0671537458           9

head(books)

##          ISBN
##        <char>
## 1: 0195153448
## 2: 0002005018
## 3: 0060973129
## 4: 0374157065
## 5: 0393045218
## 6: 0399135782
##
                       Book-Title
##
                           <char>
## 1:
              Classical Mythology
## 2:
                     Clara Callan
## 3:
             Decision in Normandy
```

```
## 4: Flu: The Story of the Great Influenza Pandemic of 1918 and the Se
arch for the Virus That Caused It
## 5:
           The Mummies of Urumchi
## 6:
           The Kitchen God's Wife
##             Book-Author Year-Of-Publication                  Publish
er
##                  <char>              <char>                     <cha
r>
## 1:   Mark P. O. Morford                2002    Oxford University Pre
ss
## 2: Richard Bruce Wright                2001      HarperFlamingo Cana
da
## 3:         Carlo D'Este                1991            HarperPerenni
al
## 4:     Gina Bari Kolata                1999       Farrar Straus Giro
ux
## 5:      E. J. W. Barber                1999 W. W. Norton &amp; Compa
ny
## 6:              Amy Tan                1991           Putnam Pub Gro
up
##                                                      Image-URL-S
##                                                           <char>
## 1: http://images.amazon.com/images/P/0195153448.01.THUMBZZZ.jpg
## 2: http://images.amazon.com/images/P/0002005018.01.THUMBZZZ.jpg
## 3: http://images.amazon.com/images/P/0060973129.01.THUMBZZZ.jpg
## 4: http://images.amazon.com/images/P/0374157065.01.THUMBZZZ.jpg
## 5: http://images.amazon.com/images/P/0393045218.01.THUMBZZZ.jpg
## 6: http://images.amazon.com/images/P/0399135782.01.THUMBZZZ.jpg
##                                                      Image-URL-M
##                                                           <char>
## 1: http://images.amazon.com/images/P/0195153448.01.MZZZZZZZ.jpg
## 2: http://images.amazon.com/images/P/0002005018.01.MZZZZZZZ.jpg
## 3: http://images.amazon.com/images/P/0060973129.01.MZZZZZZZ.jpg
## 4: http://images.amazon.com/images/P/0374157065.01.MZZZZZZZ.jpg
## 5: http://images.amazon.com/images/P/0393045218.01.MZZZZZZZ.jpg
## 6: http://images.amazon.com/images/P/0399135782.01.MZZZZZZZ.jpg
##                                                      Image-URL-L
##                                                           <char>
## 1: http://images.amazon.com/images/P/0195153448.01.LZZZZZZZ.jpg
## 2: http://images.amazon.com/images/P/0002005018.01.LZZZZZZZ.jpg
## 3: http://images.amazon.com/images/P/0060973129.01.LZZZZZZZ.jpg
## 4: http://images.amazon.com/images/P/0374157065.01.LZZZZZZZ.jpg
## 5: http://images.amazon.com/images/P/0393045218.01.LZZZZZZZ.jpg
## 6: http://images.amazon.com/images/P/0399135782.01.LZZZZZZZ.jpg
```

ISBN is a unique identifier for the book, so it can be used as a keyword to join the two tables, where Tidyverse will be used to match the RatingPGA table to the publisher.

```
rating_books = RatingPGA %>% left_join(books, by= "ISBN")

Pub_Rating = aggregate(rating_books$Book.Rating~rating_books$Publisher,
 rating_books, mean)

Rating_by_order = Pub_Rating[order(Pub_Rating$`rating_books$Book.Rating
`, decreasing = TRUE), ]
```

By ordering the rank of publishers rating, top 20 would be chosen to show in the table by kable packages.

```
Top_pub = Rating_by_order[1:20, ]
library(knitr)
table = kable(Top_pub)
table
```

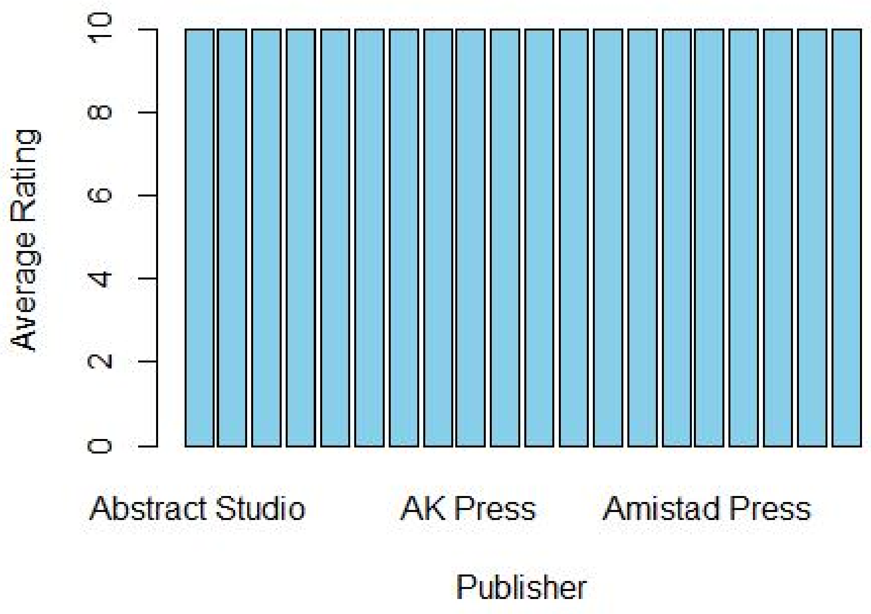
	rating_books$Publisher \vert rating_{b}ooks$Book.Rating	
35	Abstract Studio	10
43	Access Pub Network	10
48	Accord Publishing	10
55	ACS Publications	10
57	ACTA Publications	10
75	Adler's Foreign Books Inc	10
87	AG Press Publishing	10
94	Airlife Publishing	10
98	AK Press	10
124	Algora Publishing	10
129	All About Kids Pub	10
130	All about Kids Publishing	10
150	Altitude Publishing Canada	10
160	Amber Lotus	10
176	American Psychiatric Association	10
184	Amistad Press	10
221	Appalachian Trail Conference	10
232	Aqua Quest Pubn	10
233	Aquarian Press	10

rating_books$Publisher|rating_{b}ooks$Book.Ratin
g

243 Arcadia Publishing 10

Visualise the top 10 publishers.

```
barplot(Top_pub$`rating_books$Book.Rating`~Top_pub$`rating_books$Publis
her`, col='skyblue', main='', xlab='Publisher', ylab='Average Rating')
```



Finding: For the result, we can see many publishers show high rating in RatingPGA

Q2

Merge two sheet by User ID to match the age with user who made rating.

```
rating_age = merge(RatingPGA, users, by.x = 'User', by.y = 'User.ID', a
ll.x = TRUE)
```

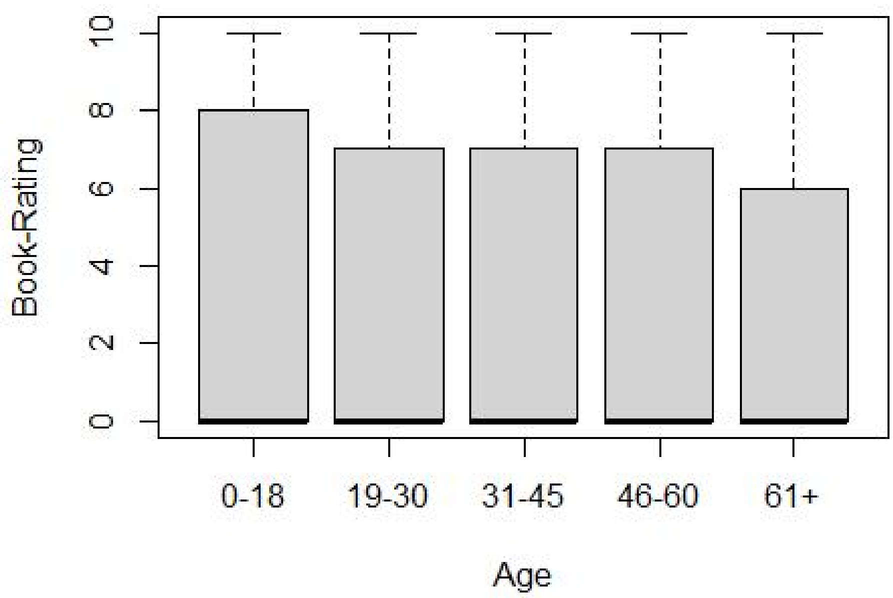
Delete null value.

```
rating_age_clean = rating_age[!is.na(rating_age$Age), ]
```

Cut function will be used to clustering the user by different age groups.

```
rating_age_clean$age_group = cut(rating_age_clean$Age,
                     breaks = c(0, 18, 30, 45, 60, Inf),
                     labels = c("0-18", "19-30", "31-45", "46-60", "61+
"),
                     right = TRUE)

boxplot(rating_age_clean$Book.Rating~rating_age_clean$age_group, data=r
ating_age_clean, xlab='Age', ylab='Book-Rating')
```



Finding: For the result, we can see that teens below 18 usually make higher rating than adults, the older people make lowest grade among all groups.

Q3

ratingPGA

```
rating_locationA = rating_age
rating_location_cleanA = rating_locationA[!is.na(rating_locationA$Locat
ion), ]
str(rating_location_cleanA)

## 'data.frame':    119697 obs. of  6 variables:
##  $ User       : int  2 8 8 8 8 8 8 8 8 8 ...
##  $ X          : int  9562 9566 9567 9572 9574 9575 9576 9577 9578 95
79 ...
##  $ ISBN       : chr  "0195153448" "0374157065" "0393045218" "0743226
78X" ...
##  $ Book.Rating: int  0 0 0 5 0 0 5 5 0 6 ...
##  $ Location   : chr  "stockton, california, usa" "timmins, ontario, 
canada" "timmins, ontario, canada" "timmins, ontario, canada" ...
##  $ Age        : int  18 NA NA NA NA NA NA NA NA NA ...
```

Tidyverse function will be adopted to get the country name from location data by extracting the last word.

```
library(tidyverse)
rating_location_cleanA$country <- sapply(strsplit(rating_location_clean
A$Location, " "), tail, n = 1)
```

Grouping by countries and calculate the average rating for every group.

```
country_ratingPGA = rating_location_cleanA %>%
  group_by(country) %>%
  summarise(mean_rating = mean(Book.Rating)) %>%
  arrange(desc(mean_rating))
country_ratingPGA = country_ratingPGA[-c(1,3,5,8,9), ] # Delete invalid
 rows
```

Visulaise the result.

```
top5_countryA = country_ratingPGA[1:5, ]
```

ratingPGB

Same way for RatingPGB data set.

```
rating_locationB = merge(RatingPGB, users, by.x = 'User', by.y = 'User.
ID', all.x = TRUE)

rating_location_cleanB = rating_locationB[!is.na(rating_locationB$Age),
 ]

rating_location_cleanB$country <- sapply(strsplit(rating_location_clean
B$Location, " "), tail, n = 1)
```

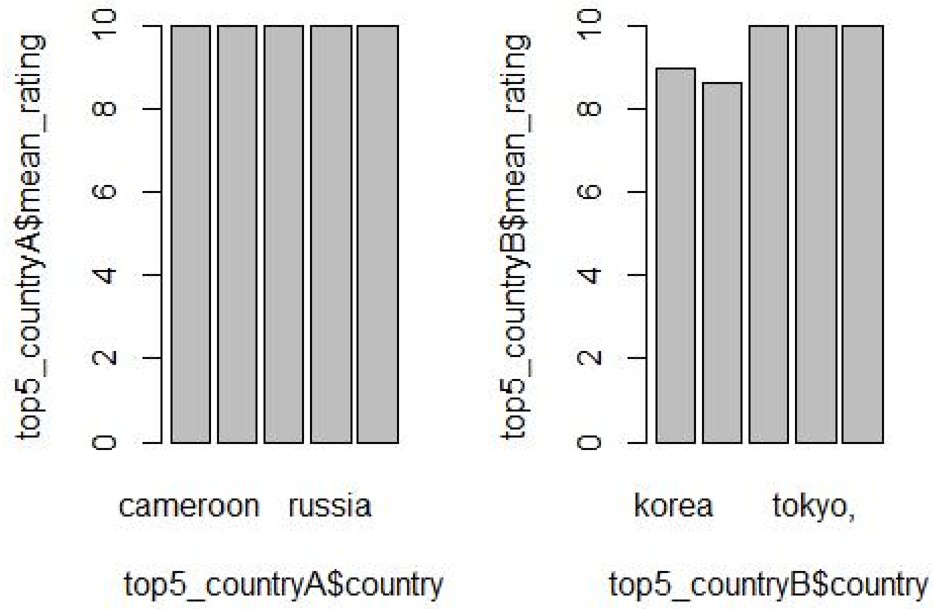
```
country_ratingPGB = rating_location_cleanB %>%
  group_by(country) %>%
  summarise(mean_rating = mean(Book.Rating)) %>%
  arrange(desc(mean_rating))

country_ratingPGB = country_ratingPGB[-c(5,6,7), ] # Delete invalid row
s

top5_countryB = country_ratingPGB[1:5, ]
```

Comparison 2 data sets

```
par(mfrow = c(1, 2))
barplot(top5_countryA$mean_rating~top5_countryA$country)
barplot(top5_countryB$mean_rating~top5_countryB$country)
```



Finding: For the result, there is true difference for two data set. It shows different country rank for book rating as well as RatingPGA has higher rating than RatingPGB

Q4

RatingPGA

Using tidyverse to select book published after 2000. then group by age for all users and calculate the mean of rating score for all groups.

```
rating_age_year = rating_age_clean %>%
  left_join(books %>% select(ISBN, 'Year-Of-Publication'), by = "ISBN")
 %>%
  filter('Year-Of-Publication' > 2000) %>%
  mutate(AgeGroup = case_when(
    Age <= 18 ~ "0-18",
    Age <= 30 ~ "19-30",
    Age <= 45 ~ "31-45",
    Age <= 60 ~ "46-60",
    TRUE ~ "61+"
  )) %>%
  group_by(AgeGroup) %>%
  summarise(AverageRating = mean(Book.Rating))
```

RatingPGB

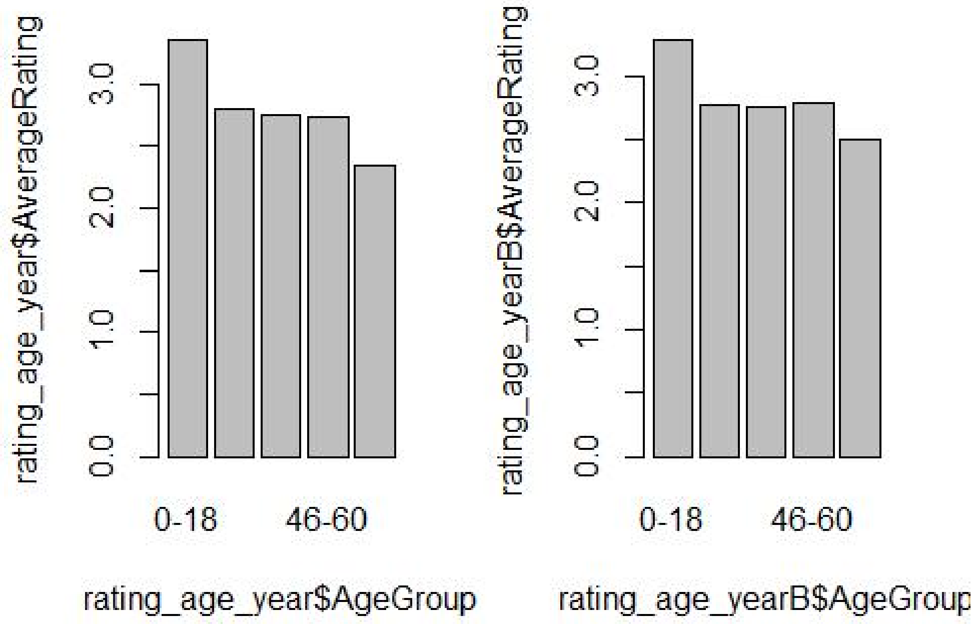
Same way for RatingPGB data set.

```
rating_ageB = merge(RatingPGB, users, by.x = 'User', by.y = 'User.ID',
all.x = TRUE)
rating_age_cleanB = rating_ageB[!is.na(rating_ageB$Age), ]
rating_age_yearB = rating_age_cleanB %>%
  left_join(books %>% select(ISBN, 'Year-Of-Publication'), by = "ISBN")
 %>%
  filter('Year-Of-Publication' > 2000) %>%
  mutate(AgeGroup = case_when(
    Age <= 18 ~ "0-18",
    Age <= 30 ~ "19-30",
    Age <= 45 ~ "31-45",
    Age <= 60 ~ "46-60",
    TRUE ~ "61+"
  )) %>%
  group_by(AgeGroup) %>%
  summarise(AverageRating = mean(Book.Rating))
```

Comparison 2 data sets

```
par(mfrow = c(1, 2))

barplot(rating_age_year$AverageRating~rating_age_year$AgeGroup)
barplot(rating_age_yearB$AverageRating~rating_age_yearB$AgeGroup)
```



Finding: For the result, there is almost same conclusion that younger people prefer to make higher rating for books published after 2000 than the elder people.