AS1: The Nuts and Bolts of Data Analysis Using R

By: Kevin Karnadi Kirmansjah 紀維鑫 - 109006241

(a) First, find out the path to the directory containing the data and load it onto R. The dataset has 541,909 rows and 8 columns. After importing the data, make sure you have imported it properly by using the head() and the str() functions.

```
data = read.csv("https://bit.ly/3jenBeO")
head(data); str(data)
```

```
##   InvoiceNo StockCode                         Description Quantity  InvoiceDate
## 1    536365    85123A  WHITE HANGING HEART T-LIGHT HOLDER        6 12/1/10 8:26
## 2    536365     71053                 WHITE METAL LANTERN        6 12/1/10 8:26
## 3    536365    84406B      CREAM CUPID HEARTS COAT HANGER        8 12/1/10 8:26
## 4    536365    84029G KNITTED UNION FLAG HOT WATER BOTTLE        6 12/1/10 8:26
## 5    536365    84029E      RED WOOLLY HOTTIE WHITE HEART.        6 12/1/10 8:26
## 6    536365     22752        SET 7 BABUSHKA NESTING BOXES        2 12/1/10 8:26
##   UnitPrice CustomerID        Country
## 1      2.55      17850 United Kingdom
## 2      3.39      17850 United Kingdom
## 3      2.75      17850 United Kingdom
## 4      3.39      17850 United Kingdom
## 5      3.39      17850 United Kingdom
## 6      7.65      17850 United Kingdom
```

```
## 'data.frame':    541909 obs. of  8 variables:
##  $ InvoiceNo  : chr  "536365" "536365" "536365" "536365" ...
##  $ StockCode  : chr  "85123A" "71053" "84406B" "84029G" ...
##  $ Description: chr  "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" "CREAM CUPID HEARTS COAT HANGER" "KNITTED UNION FLAG HOT WATER BOTTLE" ...
##  $ Quantity   : int  6 6 8 6 6 2 6 6 6 32 ...
##  $ InvoiceDate: chr  "12/1/10 8:26" "12/1/10 8:26" "12/1/10 8:26" "12/1/10 8:26" ...
##  $ UnitPrice  : num  2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...
##  $ CustomerID : int  17850 17850 17850 17850 17850 17850 17850 17850 17850 13047 ...
##  $ Country    : chr  "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom" ...
```

(b) We will be using data from 2011/07 to 2011/08. Convert InvoiceDate to date class and subset the data. You should see 3,664 unique transactions (i.e., InvoiceNo) left in the data. One alternative is to read the data from just those dates rather than reading in the entire dataset and subsetting to those dates.

- You can use length() and unique() functions on InvoiceNo to find out the number of unique transactions.
- You may find it useful to convert the Date and Time variables to Date/Time classes in R using the strptime() and as.Date() functions. lubridate package can also be used which will be helpful for more complex date operations.

All data after this code (until further subsets) will be using the subsetted data from the code below

```
data$InvoiceDate = as.Date(data$InvoiceDate, "%m/%d/%y")
data = subset(data, InvoiceDate >= as.Date("2011/07/01") & InvoiceDate <= as.Date("2011/08/31"
))
```

```
cat("Unique transactions:", length(unique(data$InvoiceNo)), "\n")
```

```
## Unique transactions: 3664
```

(c) Use for-loops to

(1) compute the mean of Quantity and UnitPrice.

```
sum_quantity = 0
for(i in data$Quantity) {
  sum_quantity = sum_quantity + i
}
mean_quantity = sum_quantity / length(data$Quantity)
sum_unitprice = 0
for(i in data$UnitPrice) {
  sum_unitprice = sum_unitprice + i
}
mean_unitprice = sum_unitprice / length(data$UnitPrice)
cat("Quantity mean:", mean_quantity, "\n")
```

```
## Quantity mean: 10.65901
```

```
cat("UnitPrice mean:", mean_unitprice, "\n")
```

```
## UnitPrice mean: 4.308608
```

(2) determine the types of each column.

```
for(name in colnames(data)) {
  cat("Type of", name, "is", class(data[, name]), "\n")
}
```

```
## Type of InvoiceNo is character
## Type of StockCode is character
## Type of Description is character
## Type of Quantity is integer
## Type of InvoiceDate is Date
## Type of UnitPrice is numeric
## Type of CustomerID is integer
## Type of Country is character
```

(3) compute the number of unique values in each column.

```
for(name in colnames(data)) {
  cat("Unique values in", name, ":", length(unique(data[, name])), "\n")
}
```

```
## Unique values in InvoiceNo : 3664
## Unique values in StockCode : 2982
## Unique values in Description : 2953
## Unique values in Quantity : 287
## Unique values in InvoiceDate : 52
## Unique values in UnitPrice : 447
## Unique values in CustomerID : 1541
## Unique values in Country : 28
```

(d) Subset the data for which the transactions took place in the U.K., Netherlands, and Australia.

All data after this code (until further subsets) will be using the subsetted data from the code below

```
data = subset(data, Country == "United Kingdom" | Country == "Netherlands" | Country == "Austra
lia")
```

Using the subset of data,

(4) Report the average and standard deviation (round them up to 3 decimal points) of the UnitPrice.

```
cat("Mean of UnitPrice:", round(mean(data$UnitPrice), 3), "\n")
```

```
## Mean of UnitPrice: 4.344
```

```
cat("Standard deviation of UnitPrice:", round(sd(data$UnitPrice), 3), "\n")
```

```
## Standard deviation of UnitPrice: 98.961
```

(5) the number of unique transactions made in these countries.

```
cat("Unique transactions:", length(unique(data$InvoiceNo)), "\n")
```

```
## Unique transactions: 3332
```

(6) How many customers residing in these countries made transactions in July and August of 2011?

```
cat("Customers residing in these countries made transactions in July and August of 2011:", leng
th(unique(data$CustomerID)), "\n")
```

```
## Customers residing in these countries made transactions in July and August of 2011: 1380
```

(e) Do we see any customers who made a refund?

Yes, there are customers who made a refund.

(7) If we do, how many customers made a refund (make sure to exclude the observations without the CustomerID)?

```
temp_data = subset(data, !is.na(CustomerID))
temp_data = subset(temp_data, grepl("c", temp_data$InvoiceNo) | grepl("C", temp_data$InvoiceN
o))
cat(length(unique(temp_data$CustomerID)), "customers made a refund", "\n")
```

```
## 337 customers made a refund
```

Assign the IDs of the customers who made at least one refund during the period into a vector called cust_refund.

```
cust_refund = c(unique(temp_data$CustomerID))
head(cust_refund)
```

```
## [1] 16746 17338 17888 12901 16422 16498
```

(f) Some customers made purchases without logging into the e-commerce site. This would create records of transactions for which the CustomerID is missing (i.e., NA). These transactions cannot be traced since we do not know who ordered the products. Create a variable called Sales by multiplying the Quantity and the UnitPrice.

(8) Then, calculate the total sales amount for those that are missing the CustomerID.

```
temp_data = subset(data, is.na(CustomerID))
Sales = sum(temp_data$Quantity * temp_data$UnitPrice)
cat("Total sales amount without Customer ID:", Sales, "\n")
```

```
## Total sales amount without Customer ID: 166454.3
```

(9) How many transactions were made without the customers logging into the e-commerce site?

```
cat("Transactions without Customer ID:", nrow(temp_data[,]), "\n")
```

```
## Transactions without Customer ID: 19306
```

(g) Extra Credit:

(EC1) Create a variable containing the monthly aggregate spending for each customer.

```
data$Sales = data$Quantity * data$UnitPrice
monthly_aggregate = aggregate(Sales ~ CustomerID+month(InvoiceDate), data = data, FUN = sum, n
a.rm = T)
monthly_aggregate = setNames(monthly_aggregate, c("CustomerID", "Month", "Sales"))
head(monthly_aggregate); tail(monthly_aggregate)
```

```
##   CustomerID Month   Sales
## 1      12388     7  902.09
## 2      12415     7 2796.36
## 3      12431     7 1069.12
## 4      12748     7 1113.27
## 5      12830     7 2915.92
## 6      12833     7  417.38
```

```
##      CustomerID Month  Sales
## 1768      18237     8 243.21
## 1769      18241     8 537.95
## 1770      18248     8 286.56
## 1771      18257     8 627.27
## 1772      18272     8 372.25
## 1773      18282     8  98.76
```

(EC2) Then, report the IDs and the monthly purchase amount of the five customers who have spent the most money in July 2011.

```
monthly_aggregate_july = subset(monthly_aggregate, Month == 7)
top5 = head(monthly_aggregate_july[order(monthly_aggregate_july$Sales, decreasing = T), ], n =
5)
row.names(top5) = c(1, 2, 3, 4, 5)
print(top5)
```

```
##   CustomerID Month    Sales
## 1      18102     7 19889.16
## 2      17949     7 11590.58
## 3      14088     7  9038.69
## 4      16684     7  8807.56
## 5      17450     7  8485.14
```