Data Cleaning Process

Data context

- Case company: an online retailer based in the UK. The business is fully e-commerce business without any physical store.
- Products: all-occasions gifts
- Customers: mainly wholesalers
- Data period: **1st of Dec, 2010 - 9nd of Dec, 2011**

Data import

The first step of data cleaning tasks is to import the data. 'readr' package was used for the ease of importing data and changing the data type of 'Invoice' column to character, 'Quantity' column to integer, and 'InvoiceDate' column to date-time.

```
#Import CSV file using 'readr' package

library(readr)
online_retail_II <- read_csv("C:/Users/Tan/Desktop/kitty/R/R Projects/E-commerce Data/online_retail_II.
                             col_types = cols(Invoice = col_character(),
                                              Quantity = col_integer(), InvoiceDate = col_datetime(forma
head(online_retail_II)
```

```
## # A tibble: 6 x 8
##   Invoice StockCode Description Quantity InvoiceDate         Price `Customer ID`
##   <chr>   <chr>     <chr>          <int> <dttm>              <dbl>         <dbl>
## 1 536365  85123A    WHITE HANG~        6 2010-12-01 08:26:00  2.55         17850
## 2 536365  71053     WHITE META~        6 2010-12-01 08:26:00  3.39         17850
## 3 536365  84406B    CREAM CUPI~        8 2010-12-01 08:26:00  2.75         17850
## 4 536365  84029G    KNITTED UN~        6 2010-12-01 08:26:00  3.39         17850
## 5 536365  84029E    RED WOOLLY~        6 2010-12-01 08:26:00  3.39         17850
## 6 536365  22752     SET 7 BABU~        2 2010-12-01 08:26:00  7.65         17850
## # ... with 1 more variable: Country <chr>
```

Change the name of 'Customer ID' column and Data summary

Using *summary()* function to have an overview at the dataset

```
online_retail_II<- rename(online_retail_II, "CustomerID" = "Customer ID")
summary(online_retail_II)
```

```
##    Invoice            StockCode         Description           Quantity
##  Length:541910      Length:541910      Length:541910      Min.   :-80995.00
##  Class :character   Class :character   Class :character   1st Qu.:     1.00
```

```
##  Mode  :character   Mode  :character   Mode  :character   Median :     3.00
##                                                           Mean   :     9.55
##                                                           3rd Qu.:    10.00
##                                                           Max.   : 80995.00
##
##   InvoiceDate                      Price             CustomerID
##  Min.   :2010-12-01 08:26:00   Min.   :-11062.06   Min.   :12346
##  1st Qu.:2011-03-28 11:34:00   1st Qu.:     1.25   1st Qu.:13953
##  Median :2011-07-19 17:17:00   Median :     2.08   Median :15152
##  Mean   :2011-07-04 13:35:22   Mean   :     4.61   Mean   :15288
##  3rd Qu.:2011-10-19 11:27:00   3rd Qu.:     4.13   3rd Qu.:16791
##  Max.   :2011-12-09 12:50:00   Max.   : 38970.00   Max.   :18287
##                                                    NA's   :135080
##    Country
##  Length:541910
##  Class :character
##  Mode  :character
##
##
##
##
```

Notice that there are **negative values** in 'Quantity' column, which is invalid. There are also **missing values** in 'CustomerID'.

Deal with missing values and 'Quantity' negative values

Strategy to deal with negative values and missing values: Deleting the rows with negative values and missing values.

```
#Remove rows with negative quantity
online_retail_II <- online_retail_II[online_retail_II$Quantity > 0,]
#Remove rows with missing values
online_retail_II <- na.omit(online_retail_II)
#Review the dataset
summary(online_retail_II)
```

```
##    Invoice            StockCode          Description           Quantity
##  Length:397925      Length:397925      Length:397925      Min.   :    1.00
##  Class :character   Class :character   Class :character   1st Qu.:    2.00
##  Mode  :character   Mode  :character   Mode  :character   Median :    6.00
##                                                           Mean   :   13.02
##                                                           3rd Qu.:   12.00
##                                                           Max.   :80995.00
##   InvoiceDate                      Price            CustomerID
##  Min.   :2010-12-01 08:26:00   Min.   :   0.000   Min.   :12346
##  1st Qu.:2011-04-07 11:12:00   1st Qu.:   1.250   1st Qu.:13969
##  Median :2011-07-31 14:39:00   Median :   1.950   Median :15159
##  Mean   :2011-07-10 23:44:09   Mean   :   3.116   Mean   :15294
##  3rd Qu.:2011-10-20 14:33:00   3rd Qu.:   3.750   3rd Qu.:16795
##  Max.   :2011-12-09 12:50:00   Max.   :8142.750   Max.   :18287
##    Country
```

```
##  Length:397925
##  Class :character
##  Mode  :character
##
##
##
```

Redefine data types of variables

Change data type of 'Country', 'CustomerID', 'Invoice', 'StockCode', 'Description' variables to **Factor** using *as.factor()* function

```
online_retail_II$Country <- as.factor(online_retail_II$Country)
online_retail_II$CustomerID <- as.factor(online_retail_II$CustomerID)
online_retail_II$Invoice <- as.factor(online_retail_II$Invoice)
online_retail_II$StockCode <- as.factor(online_retail_II$StockCode)
online_retail_II$Description <- as.factor(online_retail_II$Description)
```

Add customize variables to the dataset

- Add 'Amount_Spent' column by multiply 'Quantity' with 'Price'
- Add 'Day_of_the_week', 'Month_YR', 'Hour' columns and change their data types to **Factor**

```
#Add 'Amount_Spent' column
online_retail_II$Amount_spent <- online_retail_II$Quantity * online_retail_II$Price
#Add Day_of_the_week, Month_YR, Hour columns and change them to factor data type
online_retail_II$Day_of_the_week <- as.factor(weekdays(online_retail_II$InvoiceDate)) #there is no orde
online_retail_II$Month_Yr <- as.factor(format(online_retail_II$InvoiceDate, "%Y-%m"))
online_retail_II$Hour <- as.factor(format(online_retail_II$InvoiceDate, "%H"))
summary(online_retail_II)
```

```
##     Invoice         StockCode                                Description
##  576339 :   542   85123A :  2035   WHITE HANGING HEART T-LIGHT HOLDER:  2028
##  579196 :   533   22423  :  1724   REGENCY CAKESTAND 3 TIER          :  1724
##  580727 :   529   85099B :  1618   JUMBO BAG RED RETROSPOT           :  1618
##  578270 :   442   84879  :  1408   ASSORTED COLOUR BIRD ORNAMENT     :  1408
##  573576 :   435   47566  :  1397   PARTY BUNTING                     :  1397
##  567656 :   421   20725  :  1317   LUNCH BAG RED RETROSPOT           :  1316
##  (Other):395023   (Other):388426   (Other)                           :388434
##     Quantity         InvoiceDate                      Price
##  Min.   :    1.00   Min.   :2010-12-01 08:26:00   Min.   :   0.000
##  1st Qu.:    2.00   1st Qu.:2011-04-07 11:12:00   1st Qu.:   1.250
##  Median :    6.00   Median :2011-07-31 14:39:00   Median :   1.950
##  Mean   :   13.02   Mean   :2011-07-10 23:44:09   Mean   :   3.116
##  3rd Qu.:   12.00   3rd Qu.:2011-10-20 14:33:00   3rd Qu.:   3.750
##  Max.   :80995.00   Max.   :2011-12-09 12:50:00   Max.   :8142.750
##
##    CustomerID              Country        Amount_spent        Day_of_the_week
##  17841  :  7847   United Kingdom:354345   Min.   :     0.00   Friday   :54835
##  14911  :  5677   Germany       :  9042   1st Qu.:     4.68   Monday   :64899
##  14096  :  5111   France        :  8343   Median :    11.80   Sunday   :62775
```

```
##  12748  :  4596   EIRE          :  7238   Mean   :    22.39   Thursday :80052
##  14606  :  2700   Spain         :  2485   3rd Qu.:    19.80   Tuesday  :66476
##  15311  :  2379   Netherlands   :  2363   Max.   :168469.60   Wednesday:68888
##  (Other):369615   (Other)       : 14109
##     Month_Yr           Hour
##  2011-11: 64545   12     :72070
##  2011-10: 49557   13     :64031
##  2011-09: 40030   14     :54127
##  2011-05: 28322   11     :49092
##  2011-06: 27185   15     :45372
##  2011-03: 27177   10     :37999
##  (Other):161109   (Other):75234
```

Redefine data types of variables

Change data type of 'Country', 'CustomerID', 'Invoice', 'StockCode', 'Description' variables to **Factor** using *as.factor()* function

```
online_retail_II$Country <- as.factor(online_retail_II$Country)
online_retail_II$CustomerID <- as.factor(online_retail_II$CustomerID)
online_retail_II$Invoice <- as.factor(online_retail_II$Invoice)
online_retail_II$StockCode <- as.factor(online_retail_II$StockCode)
online_retail_II$Description <- as.factor(online_retail_II$Description)
```

Managing duplicate data

The main goal of this step is to detect possible identical data and consider if they are duplicate or just normal identical data by chance.

```
online_retail_II[duplicated(online_retail_II),]
```

```
## # A tibble: 5,192 x 12
##    Invoice StockCode Description Quantity InvoiceDate         Price CustomerID
##    <fct>   <fct>     <fct>          <int> <dttm>              <dbl> <fct>
##  1 536409  21866     UNION JACK~        1 2010-12-01 11:45:00  1.25 17908
##  2 536409  22866     HAND WARME~        1 2010-12-01 11:45:00  2.1  17908
##  3 536409  22900     SET 2 TEA ~        1 2010-12-01 11:45:00  2.95 17908
##  4 536409  22111     SCOTTIE DO~        1 2010-12-01 11:45:00  4.95 17908
##  5 536412  22327     ROUND SNAC~        1 2010-12-01 11:49:00  2.95 17920
##  6 536412  22273     FELTCRAFT ~        1 2010-12-01 11:49:00  2.95 17920
##  7 536412  22749     FELTCRAFT ~        1 2010-12-01 11:49:00  3.75 17920
##  8 536412  22141     CHRISTMAS ~        1 2010-12-01 11:49:00  2.1  17920
##  9 536412  21448     12 DAISY P~        1 2010-12-01 11:49:00  1.65 17920
## 10 536412  22569     FELTCRAFT ~        2 2010-12-01 11:49:00  3.75 17920
## # ... with 5,182 more rows, and 5 more variables: Country <fct>,
## #   Amount_spent <dbl>, Day_of_the_week <fct>, Month_Yr <fct>, Hour <fct>
```

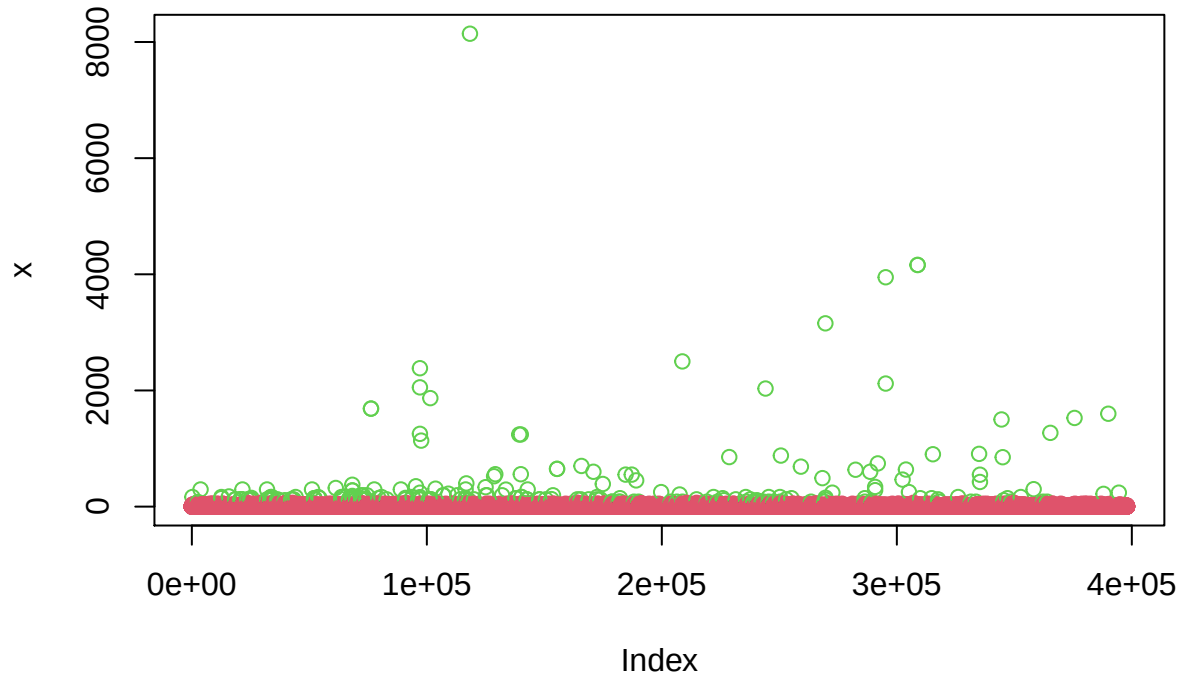
It seems like there are 5,192 rows of duplicate data, however, considering the character of e-commerce dataset, it's possible and normal to have customer add the same product to the cart at the same time. Therefore, the identical data wasn't deleted out of the dataset.

Managing outliers in the data

The goal of this step is to detect of outliers in the numeric or integer variables and consider if the outliers are relevant to the dataset or just typos. This report uses the **Six Sigma Method** for outliers detection.

'Price' variable outliers detection:

```
x = online_retail_II$Price
t = 3
m = mean(x, na.rm = F)
s = sd(x, na.rm = F)
b1 = m-s*t
b2 = m+s*t
y = ifelse(x>= b1 & x<= b2, 0, 1)
#Plot for outliers detection of 'Price' variable
plot(x, col= y+2)
```



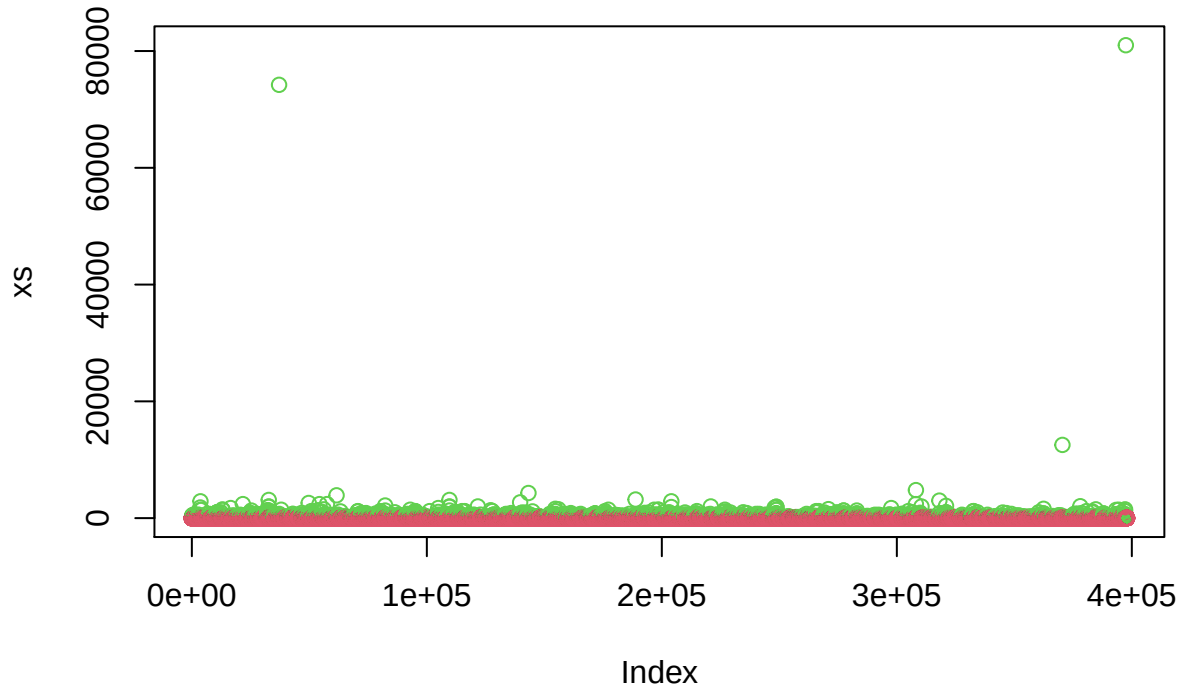

```
#Possible outliers
outl = which(y==1)
online_retail_II[outl,]
```

```
## # A tibble: 221 x 12
##    Invoice StockCode Description Quantity InvoiceDate         Price CustomerID
##    <fct>   <fct>     <fct>          <int> <dttm>              <dbl> <fct>
##  1 536392  22827     RUSTIC  SE~        1 2010-12-01 10:29:00 165   13705
```

```
##  2 536676  21769     VINTAGE PO~        1 2010-12-02 12:18:00  80.0 16752
##  3 536835  22655     VINTAGE RE~        1 2010-12-02 18:06:00 295   13145
##  4 537859  22828     REGENCY MI~        1 2010-12-08 16:11:00 165   14030
##  5 537859  22827     RUSTIC  SE~        2 2010-12-08 16:11:00 145   14030
##  6 538354  22826     LOVE SEAT ~        2 2010-12-10 15:45:00 175   16873
##  7 538662  22655     VINTAGE RE~        2 2010-12-13 15:44:00 125   15159
##  8 538662  22656     VINTAGE BL~        2 2010-12-13 15:44:00 125   15159
##  9 538999  22655     VINTAGE RE~        2 2010-12-15 12:09:00 125   16003
## 10 538999  22656     VINTAGE BL~        2 2010-12-15 12:09:00 125   16003
## # ... with 211 more rows, and 5 more variables: Country <fct>,
## #   Amount_spent <dbl>, Day_of_the_week <fct>, Month_Yr <fct>, Hour <fct>
```

'Quantity' variable outliers detection:

```
xs = online_retail_II$Quantity
ts = 3
ms = mean(x,na.rm = F)
ss = sd(x, na.rm = F)
b1s = ms - ss*ts
b2s = ms + ss*ts
ys = ifelse(xs>= b1s & xs<= b2s, 0,1)
#Plot for outliers detection of 'Quantity' variable
plot(xs,col= ys+2)
```



```r
#Possible outliers
outls = which(ys==1)
online_retail_II[outls,]
```

```
## # A tibble: 10,594 x 12
##    Invoice StockCode Description Quantity InvoiceDate         Price CustomerID
##    <fct>   <fct>     <fct>          <int> <dttm>              <dbl> <fct>
##  1 536371  22086     PAPER CHAI~       80 2010-12-01 09:00:00  2.55 13748
##  2 536378  21212     PACK OF 72~      120 2010-12-01 09:37:00  0.42 14688
##  3 536378  85071B    RED CHARLI~       96 2010-12-01 09:37:00  0.38 14688
##  4 536386  85099C    JUMBO  BAG~      100 2010-12-01 09:57:00  1.65 16029
##  5 536386  85099B    JUMBO BAG ~      100 2010-12-01 09:57:00  1.65 16029
##  6 536387  79321     CHILLI LIG~      192 2010-12-01 09:58:00  3.82 16029
##  7 536387  22780     LIGHT GARL~      192 2010-12-01 09:58:00  3.37 16029
##  8 536387  22779     WOODEN OWL~      192 2010-12-01 09:58:00  3.37 16029
##  9 536387  22466     FAIRY TALE~      432 2010-12-01 09:58:00  1.45 16029
## 10 536387  21731     RED TOADST~      432 2010-12-01 09:58:00  1.25 16029
## # ... with 10,584 more rows, and 5 more variables: Country <fct>,
## #   Amount_spent <dbl>, Day_of_the_week <fct>, Month_Yr <fct>, Hour <fct>
```

There are 221 possible outliers for 'Price' variable and 10,594 possible outliers for 'Quantity' in the dataset. However, considering the characters of the e-commerce dataset, it's possible that there were customers ordered a huge quantity of products and there were also possible expensive products. Therefore, this report didn't exclude the possible outliers

Plausibility check for Factor and Date-time variables

This step is to detect any invalid values of Factor and Date-time variables using *summary()* function.

```r
summary(online_retail_II)
```

```
##     Invoice         StockCode                                  Description
##  576339 :   542   85123A :  2035   WHITE HANGING HEART T-LIGHT HOLDER:  2028
##  579196 :   533   22423  :  1724   REGENCY CAKESTAND 3 TIER          :  1724
##  580727 :   529   85099B :  1618   JUMBO BAG RED RETROSPOT           :  1618
##  578270 :   442   84879  :  1408   ASSORTED COLOUR BIRD ORNAMENT     :  1408
##  573576 :   435   47566  :  1397   PARTY BUNTING                     :  1397
##  567656 :   421   20725  :  1317   LUNCH BAG RED RETROSPOT           :  1316
##  (Other):395023   (Other):388426   (Other)                           :388434
##     Quantity         InvoiceDate                      Price
##  Min.   :    1.00   Min.   :2010-12-01 08:26:00   Min.   :   0.000
##  1st Qu.:    2.00   1st Qu.:2011-04-07 11:12:00   1st Qu.:   1.250
##  Median :    6.00   Median :2011-07-31 14:39:00   Median :   1.950
##  Mean   :   13.02   Mean   :2011-07-10 23:44:09   Mean   :   3.116
##  3rd Qu.:   12.00   3rd Qu.:2011-10-20 14:33:00   3rd Qu.:   3.750
##  Max.   :80995.00   Max.   :2011-12-09 12:50:00   Max.   :8142.750
##
##    CustomerID              Country        Amount_spent        Day_of_the_week
##  17841  :  7847   United Kingdom:354345   Min.   :     0.00   Friday   :54835
##  14911  :  5677   Germany       :  9042   1st Qu.:     4.68   Monday   :64899
##  14096  :  5111   France        :  8343   Median :    11.80   Sunday   :62775
```

```
##  12748  :  4596   EIRE          :  7238   Mean   :    22.39   Thursday :80052
##  14606  :  2700   Spain         :  2485   3rd Qu.:    19.80   Tuesday  :66476
##  15311  :  2379   Netherlands   :  2363   Max.   :168469.60   Wednesday:68888
##  (Other):369615   (Other)       : 14109
##     Month_Yr           Hour
##  2011-11: 64545   12     :72070
##  2011-10: 49557   13     :64031
##  2011-09: 40030   14     :54127
##  2011-05: 28322   11     :49092
##  2011-06: 27185   15     :45372
##  2011-03: 27177   10     :37999
##  (Other):161109   (Other):75234
```

There is no invalid factor or date-time values detected from the dataset.