ADVANCED DATA MANAGEMENT PROJECT

Data Cleansing - Solutions to R-Programming Exercises

Before you begin, ensure that the required packages are installed and that the data is in R Studio.

```
library(readr)
library(dplyr)
library(lubridate)
patients<- read.csv("patients.csv", header=T, na.strings=c("","NA"))
flights<- read.csv("flights.csv", header=T, na.strings=c("","NA"))
consfile<- read.csv("consfile.csv", header=T, na.strings=c("","NA"))
patfile<- read.csv("patfile.csv", header=T, na.strings=c("","NA"))
View(patients)
View(consfile)
View(flights)
View(patfile)
```

Exercise 1.1

Use **counts** and **bar charts and pie charts** (as in Examples above) to produce frequency counts, horizontal bar charts and pie charts indicating missing and invalid values of the variables Crew and Dest from the **flights** data set. Use the space below to write down your findings in relation to the **flights** data set.

Exercise 1.1 - Solution & Output

1.1aFrequency Counts

```
# Produce a Simple Frequency Count for Crew Variable
count(flights, 'Crew')
```

```
`"Crew"`     n
  <chr>    <int>
1 Crew        25
```

```
# Produce a Simple Frequency Count for Dest Variable
count(flights, 'Dest')
```

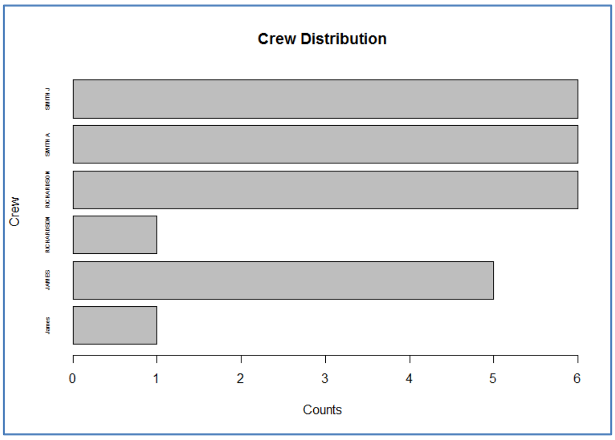
```
  `"Dest"`     n
  <chr>    <int>
1 Dest        25
```

1.1b Horizontal Bar Charts

```
# Simple Horizontal Bar Plot with Added Labels - Crew
#read the missing cells into the counts object along with missing
any values
counts <- table(flights$Crew, useNA ="ifany")

#Assign name "NA" to the missing values within the counts object
names(counts)[is.na(names(counts))] <- "NA"

#Display barplot
barplot(counts, main="Crew Distribution", xlab='Counts', ylab='Crew', hori
z=TRUE, cex.names=0.45)
```

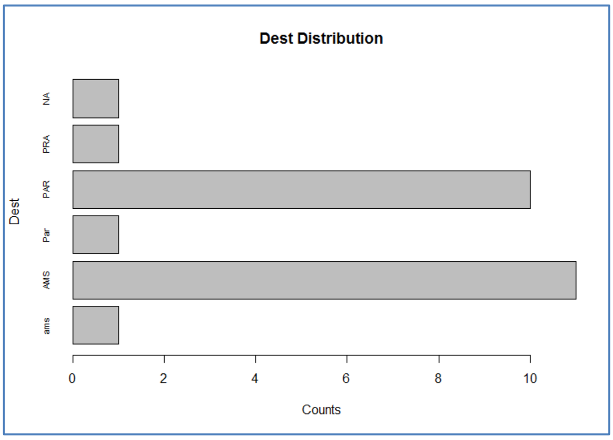


Bar Plot of Crew Variable

```
# Simple Horizontal Bar Plot with Added Labels - Dest
#read the missing cells into the counts object along with missing
any values
counts <- table(flights$Dest, useNA ="ifany")

#Assign name "NA" to the missing values within the counts object
names(counts)[is.na(names(counts))] <- "NA"

#Display barplot
barplot(counts, main="Dest Distribution", xlab='Counts', ylab='Dest', hori
z=TRUE, cex.names=0.70)
```



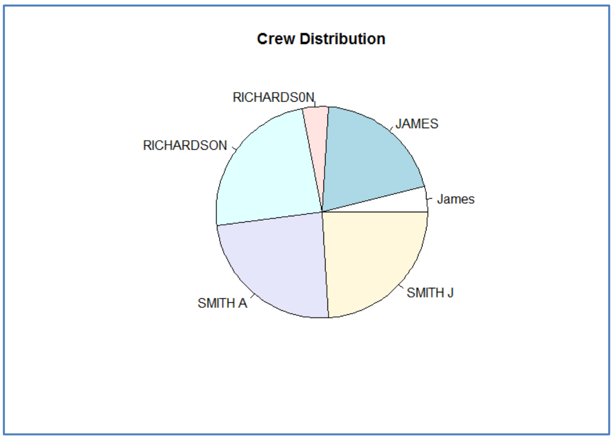
Bar Plot of Dest Variable (showing missing values as NA)

1.1c Horizontal Bar Charts

```
# Simple Pie Chart with Added Labels - Crew
#read the missing cells into the counts object along with missing
any values
counts <- table(flights$Crew, useNA ="ifany")

#Assign name "NA" to the missing values within the counts object
names(counts)[is.na(names(counts))] <- "NA"

#Display pie chart
pie(counts, main="Crew Distribution")
```

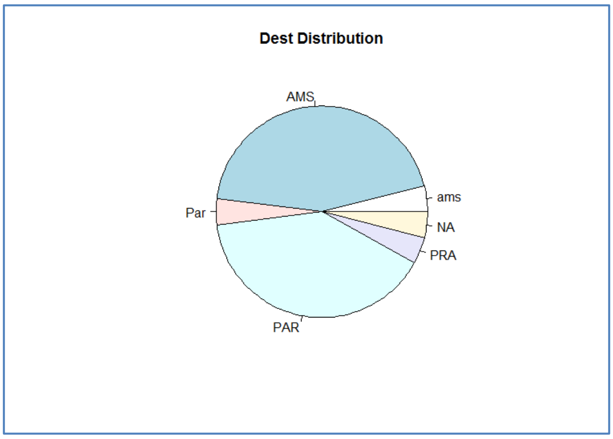


Pie Chart Plot of Crew Variable

```
# Simple Pie Chart with Added Labels - Dest
#read the missing cells into the counts object along with missing
any values
counts1 <- table(flights$Dest, useNA ="ifany")

#Assign name "NA" to the missing values within the counts object
names(counts1)[is.na(names(counts1))] <- "NA"

#Display pie chart
pie(counts1, main="Dest Distribution")
```



Pie Chart of Dest Variable (showing missing values as NA)

Exercise 1.2

Use the techniques above to locate and identify all missing and invalid character variable values in the **flights** data set.

Exercise 1.2 - Solution & Output

1.2a Missing Values

```
#Store indexes of missing values in an integer-valued vector
MissingValues= which(is.na(flights), arr.ind=TRUE)
#Get rownames of missing values and store in object x
x = rownames(flights)[MissingValues[,1]]

#Get column names of missing values and store in object y
y = colnames(flights)[MissingValues[,2]]

#Merge objects x and y with equal dimensions
LocatedMissingValues = paste(x, y, sep=" ")
LocatedMissingValues

[1] "11 Id"      "24 Dest"     "15 Date"     "19 Date"     "14 Freight"
```

1.2b Invalid Values

```
#Check if the columns contain any non-numeric values
NonNum <- unlist(lapply(flights, is.numeric))
NonNum

     Id     Dest     Date     Crew  Boarded  Freight     Mail  Revenue
  FALSE    FALSE    FALSE    FALSE     TRUE     TRUE     TRUE     TRUE

#List all values in non-numeric columns
flights[ , NonNum]

#Check if the columns contain any non-character values
NonChar <- unlist(lapply(flights, is.character))
NonChar

     Id     Dest     Date     Crew  Boarded  Freight     Mail  Revenue
  FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE

#List all values in non-character columns
flights[ , NonChar]

data frame with 0 columns and 25 rows
```

Exercise 2.1

Use **summary** and **hist** (as in Example 2.1) to produce summary statistics and histograms to identify missing and potential outlying values of all the numeric variables in the flights data set.

Exercise 2.1 - Solution & Output

```
#Generate Summary Stat Measures – Min,Max,AVG,MED,STD,Quartiles
summary(flights)
```

```
      Id       Dest         Date                    Crew      Boarded
    Freight
271105 : 2   ams : 1   Min.   :2002-03-04   James     :1   Min.   :129.0
 Min.   :205.0
271112 : 2   AMS :11   1st Qu.:2002-03-07   JAMES     :5   1st Qu.:345.0
 1st Qu.:281.2
271101 : 1   Par : 1   Median :2002-03-10   RICHARDSON:1   Median :366.0
 Median :309.0
271103 : 1   PAR :10   Mean   :2002-12-21   RICHARDSON:6   Mean   :355.3
 Mean   :343.8
271104 : 1   PRA : 1   3rd Qu.:2002-03-14   SMITH A   :6   3rd Qu.:401.0
 3rd Qu.:430.0
(Other):17   NA's: 1   Max.   :2020-03-16   SMITH J   :6   Max.   :497.0
 Max.   :498.0
NA's   : 1             NA's   :2
 NA's   :1
     Mail          Revenue
Min.   :146.0   Min.   :31403
1st Qu.:167.0   1st Qu.:42086
Median :177.0   Median :44614
Mean   :178.2   Mean   :45965
3rd Qu.:186.0   3rd Qu.:48529
Max.   :215.0   Max.   :84659
```

Exercise 2.2

In the flights data set, suppose that it is known in advance that

- Boarded should lie between 200 and 500
- Freight should lie between 150 and 550
- Mail should lie between 100 and 250
- Revenue should lie between 25000 and 65000

Use the data frame of Example 2.2 to locate and identify all missing and invalid (out of range) numeric variable values in the flights data set.

Exercise 2.2 - Solution & Output

2.2a Filter the Data

```
#From the dplyr package use the %>% and 'filter' function
#Select & display missing Boarded values
flights %>% filter(is.na(Boarded))
```

```
[1] Id      Dest    Date    Crew    Boarded Freight Mail    Revenue
<0 rows> (or 0-length row.names)
```

```
#Select & display missing Freight values
flights %>% filter(is.na(Freight))
```

```
      Id Dest     Date    Crew Boarded Freight Mail Revenue
1 271113  AMS 11.03.02 SMITH A     401      NA  174   47986
```

```
#Select & display missing Mail values
flights %>% filter(is.na(Mail))
```

```
[1] Id      Dest    Date    Crew    Boarded Freight Mail    Revenue
<0 rows> (or 0-length row.names)
```

```
#Select & display missing Revenue values
flights %>% filter(is.na(Revenue))
```

```
[1] Id      Dest    Date    Crew    Boarded Freight Mail    Revenue
<0 rows> (or 0-length row.names)
```

2.2b Subset the Data

```
#Boarded should lie between 200 and 500
outliers1 <- subset(flights, Boarded < 200 | Boarded > 500)
```

```
#display the out-of-range Boarded values
outliers1
```

```
       Id Dest     Date       Crew Boarded Freight Mail Revenue
10 271109  AMS 09.03.02 RICHARDSON     129     368  203   31403
```

```
#Freight should lie between 150 and 550
outliers2 <- subset(flights, Freight < 150 | Freight > 550)

#display the out-of-range Freight values
outliers2

[1] Id      Dest    Date    Crew    Boarded Freight Mail    Revenue
<0 rows> (or 0-length row.names)

#Mail should lie between 100 and 250
outliers3 <- subset(flights, Mail < 100 | Mail > 250)

#display the out-of-range Mail values
outliers3

[1] Id      Dest    Date    Crew    Boarded Freight Mail    Revenue
<0 rows> (or 0-length row.names)

#Revenue should lie between 25000 and 65000
outliers4 <- subset(flights, Revenue < 25000 | Revenue > 65000)

#display the out-of-range Revenue values
outliers4

       Id Dest     Date    Crew Boarded Freight Mail Revenue
12 271111  AMS 10.03.02 SMITH A     389     479  188   84659
```

Exercise 2.3

Use normal plots to identify potential outliers and to suggest reasonable ranges for each of the numeric variables in the flights data set. Compare your findings with the vertical bar charts for these variables from Exercise 2.1.

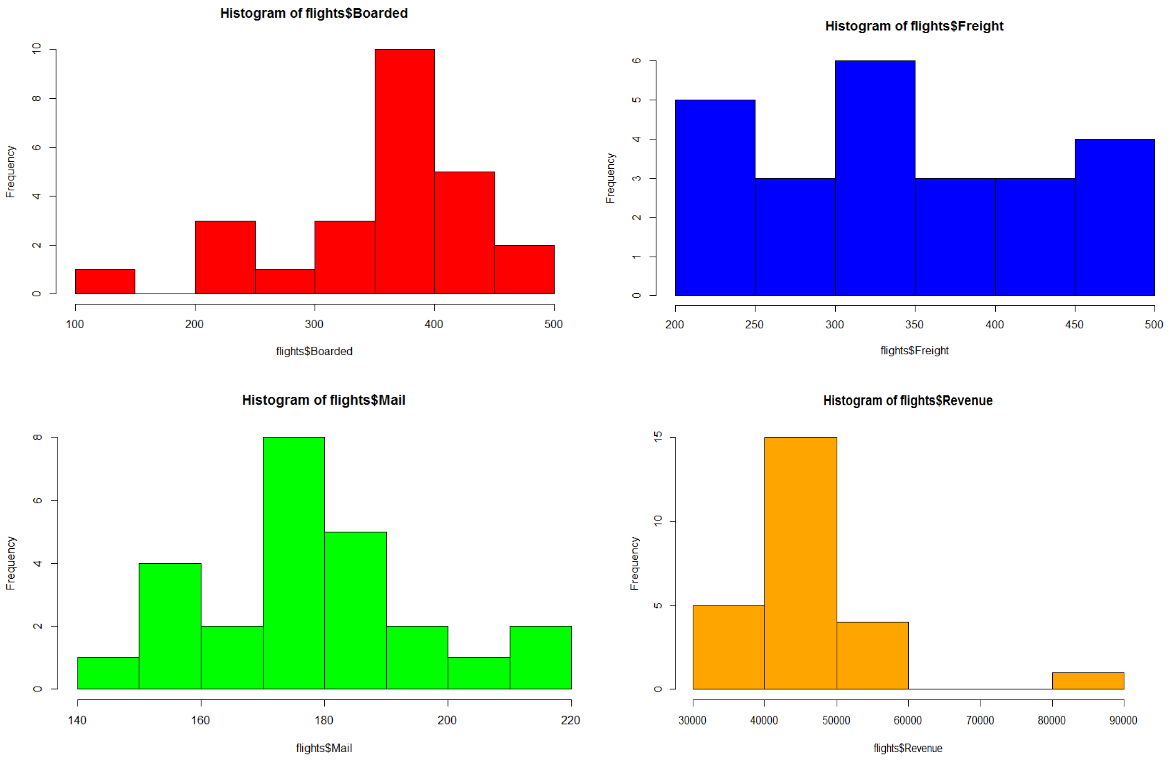
If necessary, re-plot without any very extreme outliers to identify further outliers and to refine your estimate of a reasonable range.

Of course, once reasonable ranges have been established for each numeric variable using these plots, the techniques of sections 3.4.3 or 3.4.4 above can be employed to identify and locate the outliers – these are simply the invalid (out of range) observations.

Exercise 2.3 - Solution & Output

```
#Generate Histograms for Boarded, Freight, Mail and Revenue.
hist(flights$Boarded,col="red")
hist(flights$Freight,col="blue")
hist(flights$Mail,col="green")
hist(flights$Revenue,col="Orange")
```

The outputs from the above code are as follows:



Histograms of Boarded, Freight, Mail and Revenue Variables

- Boarded this has a clear outlier to the LHS side of the plot, therefore, adjust the range to 200-500.
- Freight is relatively constant/flat but has three defined peaks, therefore, no adjustment necessary.

- Mail displays has an almost normal distribution but has three clear please, no adjustment necessary
- Revenue this has a clear outlier at the RHS of the plot, therefore, adjust the range to 30,000-60,000.

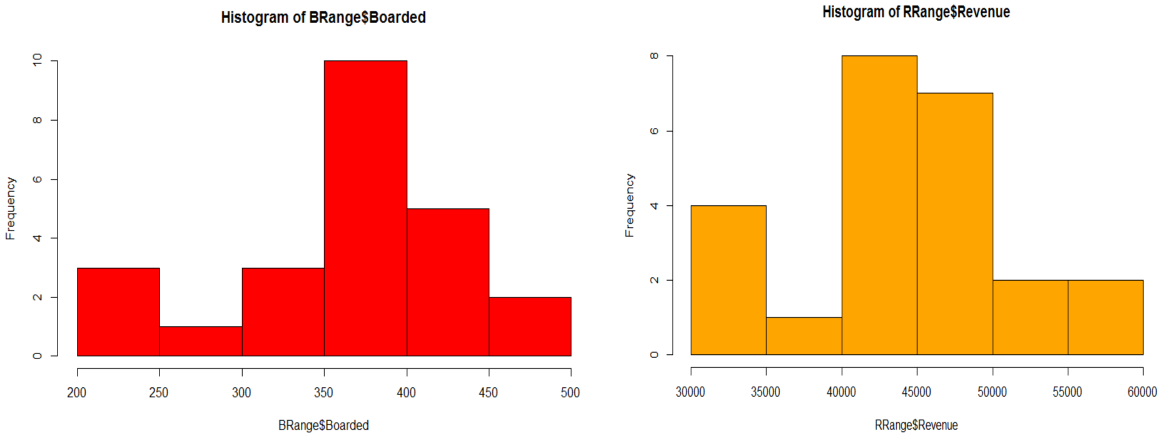
The following code can be used to make the required adjustments:

```
#Boarded should lie between 200 and 500 (using the subset function)
BRange <- subset(flights, Boarded > 200 & Boarded <= 500)
BRange

#Generate Histograms for Boarded.
hist(BRange$Boarded,col="red")

#Revenue should lie between 30,000 and 60,000 (using the subset
function)
RRange <- subset(flights, Revenue >= 30000 & Revenue <= 60000)
RRange

#Generate Histograms for Boarded.
hist(RRange$Revenue,col="orange")
```



Histograms of the new Boarded (BRange) and Revenue (RRange) Variables

All future data cleansing for Boarded and Revenue would take place on the new data sets BRange and RRange respectively.

Exercise 3.1

Suppose that all the individual flights in the data set **flight** should have taken place between the 4'th and 17'th of March 2002 inclusive.

Use the techniques outlined above in Example 3.1 to locate and identify all missing and invalid (out of range) dates in the **flights** data set.

Exercise 3.1 - Solution & Output

3.1a Identify All Invalid (out of range) Data Values

```
#Suppose that all visits should have taken place between 04/03/2002
and 17/03/2002 inclusive
#Use lubridate for easy manipulation of date values
#Check that the Date variable is a “Date”
class(flights$Date)

[1] "factor"

#format Date variable as Date (if not already).
#We use the 'dmy' function as Date variable is in the d-m-y format
#Note the format change as opposed to the solution in Example 3
flights$Date = dmy(flights$Date)

#Now Check again that the Date variable is a “Date”
class(flights$Date)

[1] "Date"

#Select & display invalid visits outside specified dates
flights %>% filter(!(Date >= "2002-03-04" & Date <= "2002-03-17"))

      Id Dest       Date  Crew Boarded Freight Mail Revenue
1 271122  PAR 2020-03-16 JAMES     332     292  179   41344
```

3.1b Identify All Missing Date Values

```
#Select & display missing values for Date
flights %>% filter(is.na(Date))

      Id Dest Date  Crew Boarded Freight Mail Revenue
1 271114  PAR <NA> JAMES     497     308  160   56950
2 271118  PAR <NA> James     348     235  171   42086
```

Exercise 4.1

On the **flights** data set, the variable Id is an ID variable.

Using the techniques of Example 4.1, identify all observations in this data set with duplicate values of Id.

4.1a Identify Duplicate Values of the ID Variable

```
#create data frame to hold duplicate values from defined column
duplicate <- data.frame(table(flights$Id))

#count the frequency of each duplicate(s)
duplicate[duplicate$Freq > 1,]
```

```
     Var1 Freq
4  271105    2
10 271112    2
```

```
#show observations with user-defined duplicates by Id
flights[flights$Id %in% duplicate$Var1[duplicate$Freq > 1],]
```

```
       Id Dest       Date       Crew Boarded Freight Mail Revenue
5  271105  AMS 2002-03-07 RICHARDSON     248     307  215   34655
6  271105  AMS 2002-03-07 RICHARDSON     248     307  215   34655
13 271112  PAR 2002-03-10      JAMES     415     463  182   50889
22 271112  AMS 2002-03-16    SMITH A     226     379  185   32059
```

4.1b Identifying Unique Duplicate Values of the ID Variable

```
DuplicateFlightsByID <- flights[flights$Id %in% duplicate$Var1[dupli
cate$Freq > 1],]
UniqueDuplicateFlightsByID <- DuplicateFlightsByID[!duplicated(Dupli
cateFlightsByID),]
UniqueDuplicateFlightsByID
```

```
       Id Dest       Date       Crew Boarded Freight Mail Revenue
5  271105  AMS 2002-03-07 RICHARDSON     248     307  215   34655
13 271112  PAR 2002-03-10      JAMES     415     463  182   50889
22 271112  AMS 2002-03-16    SMITH A     226     379  185   32059
```