Data cleaning

Missing Values

Example 1*

```
# Lets create a data set dt
# ---
# OUR CODE GOES BELOW
#
Name <- c("John", "Tim", NA)
Sex <- c("men", "men", "women")
Age <- c(45, 53, NA)
dt <- data.frame(Name, Sex, Age)
# Then print out this data set below
dt
```

```
##   Name   Sex Age
## 1 John   men  45
## 2  Tim   men  53
## 3 <NA> women  NA
```

Finding the null values

```
# Lets Identify missing data in your dataset
# by using the function is.na()
# ---
#
is.na(dt)
```

```
##       Name   Sex   Age
## [1,] FALSE FALSE FALSE
## [2,] FALSE FALSE FALSE
## [3,]  TRUE FALSE  TRUE
```

```
colSums(is.na(dt))
```

```
## Name  Sex  Age
##    1    0    1
```

Dealing with missing values

```
# Omitting the null values
na.omit(dt)
```

```
##   Name Sex Age
## 1 John men  45
## 2  Tim men  53
```

Dealing with null values

```
# Recode/fill the missing value in a column with a number
# ---
#
dt$Age[is.na(dt$Age)] <- 99
dt
```

```
##   Name   Sex Age
## 1 John   men  45
## 2  Tim   men  53
## 3 <NA> women  99
```

Filling the null value with mean

```
# Question: Recode or fill the missing value in a column with the mean
#value of the column-#-
#
dt$Age[is.na(dt$Age)] <- mean(dt$Age, na.rm = TRUE)
# print the dt table below
dt
```

```
##   Name   Sex Age
## 1 John   men  45
## 2  Tim   men  53
## 3 <NA> women  99
```

Challenge 1

```
# ---
# Question: Using the given bus data set below, re-code the missing
# and travel_to columns with a then appropriate value
#
# Lets first of all import our data table
# ---
#
library("data.table")
bus_dataset <- read.csv('http://bit.ly/BusNairobiWesternTransport')
# First check have a look at the data set
# --
#
head(bus_dataset)
```

```
##   ride_id seat_number payment_method payment_receipt travel_date
travel_time
## 1    1442         15A          Mpesa      UZUEHCBUSO    17-10-17
7:15
## 2    5437         14A          Mpesa      TIHLBUSGTE    19-11-17
7:12
## 3    5710          8B          Mpesa      EQX8Q5G19O    26-11-17
7:05
## 4    5777         19A          Mpesa      SGP18CL0ME    27-11-17
7:10
```

```
## 5    5778         11A          Mpesa      BM97HFRGL9    27-11-17
7:12
## 6    5777         18B          Mpesa      B6PBDU30IZ    27-11-17
7:10
##   travel_from travel_to car_type max_capacity
## 1      Migori   Nairobi      Bus           49
## 2      Migori   Nairobi      Bus           49
## 3      Keroka   Nairobi      Bus           49
## 4    Homa Bay   Nairobi      Bus           49
## 5      Migori   Nairobi      Bus           49
## 6    Homa Bay   Nairobi      Bus           49
```

Checking for missing values

```
colSums(is.na(bus_dataset))
```

```
##         ride_id     seat_number  payment_method payment_receipt
travel_date
##               0               0               0               0
0
##     travel_time     travel_from       travel_to        car_type
max_capacity
##               0               0               0               0
0
```

Challenge 2

```
# Question: Clean the given dataset
# ---
# Dataset url = http://bit.ly/MS-PropertyDataset
# ---
property_dataset <- read.csv('http://bit.ly/MS-PropertyDataset')
# printing the data
head(property_dataset)
```

```
##         PID ST_NUM   ST_NAME OWN_OCCUPIED NUM_BEDROOMS NUM_BATH SQ_FT
## 1 100001000    104    PUTNAM            Y            3        1  1000
## 2 100002000    197 LEXINGTON            N            3      1.5    --
## 3 100003000     NA LEXINGTON            N          n/a        1   850
## 4 100004000    201  BERKELEY           12            1      NaN   700
## 5        NA    203  BERKELEY            Y            3        2  1600
## 6 100006000    207  BERKELEY            Y         <NA>        1   800
```

Checking for missing values

```
property_dataset[ property_dataset == "na"] <- NA
property_dataset[ property_dataset == "NaN"] <- NA
property_dataset[ property_dataset == "--"] <- NA
property_dataset[ property_dataset == " "] <- NA
colSums(is.na(property_dataset))
```

```
##          PID       ST_NUM      ST_NAME OWN_OCCUPIED NUM_BEDROOMS
NUM_BATH
##            1            2            0            0            2
1
##        SQ_FT
##            1
```

```
property_dataset[ complete.cases(property_dataset),]
```

```
##         PID ST_NUM ST_NAME OWN_OCCUPIED NUM_BEDROOMS NUM_BATH SQ_FT
## 1 100001000    104  PUTNAM            Y            3        1  1000
## 8 100008000    213 TREMONT            Y            1        1
```

```
new_property_dataset <- na.omit(property_dataset)
head(new_property_dataset)
```

```
##         PID ST_NUM ST_NAME OWN_OCCUPIED NUM_BEDROOMS NUM_BATH SQ_FT
## 1 100001000    104  PUTNAM            Y            3        1  1000
## 8 100008000    213 TREMONT            Y            1        1
```

Challenge 3

```
# Dataset url = http://bit.ly/AirQualityDataset

air_quality <- read.csv('http://bit.ly/AirQualityDataset')
head(air_quality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

```
colSums(is.na(air_quality))
```

```
##   Ozone Solar.R    Wind    Temp   Month     Day
##      37       7       0       0       0       0
```

Refilling the null values with mode

```
air_quality$Ozone[is.na(air_quality$Ozone)] <- mean(air_quality$Ozone, na.rm
= TRUE)
air_quality$Solar.R[is.na(air_quality$Solar.R)] <- mean(air_quality$Solar.R,
na.rm = TRUE)

colSums(is.na(air_quality))
```

```
##   Ozone Solar.R    Wind    Temp   Month     Day
##       0       0       0       0       0       0
```

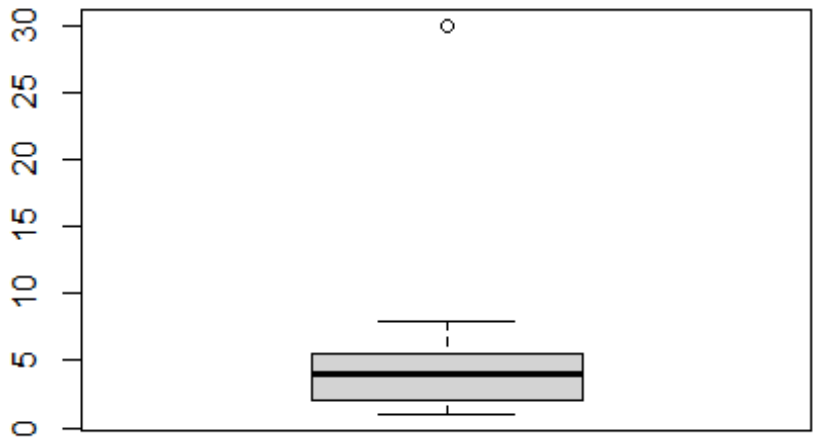
Outliears

Screening for outliers

```
# Let's create the vector A
# ---
#
A <- c(3, 2, 5, 6, 4, 8, 1, 2, 30, 2, 4)
# then print it out
A

##  [1]  3  2  5  6  4  8  1  2 30  2  4

# We then plot a boxplot to help us visualise any existing outliers
# ---
#
boxplot(A)
```




```
# Then use the function boxplot.stats which lists the outliers in the vectors
# ---
#
boxplot.stats(A)$out

## [1] 30
```

Checking for inconsistency

```
# Say from our vector x above, values above 20 are obvious inconsistencies
# then we using logical indices to check for
# ---
#
non_greater_than_20 <- A > 20
# printing out non_greater_than_20
non_greater_than_20
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
```

Challenge

```
# Question: Use the given bus dataset below, determine whether there are any
obvious inconsistencies
# ---
# Dataset url = http://bit.ly/BusNairobiWesternTransport
# ---
#
# Importing our database
# ---
# install.package("data.table") # install package data.table to work with
data tables
library(data.table) # load package install.package("tidyverse") # install
packages to work with data frame - extends into visualization
#library(tidyverse)
bus_dataset <- read.csv('http://bit.ly/BusNairobiWesternTransport')
# Previewing the dataset
# ---
#
head(bus_dataset)
```

```
##   ride_id seat_number payment_method payment_receipt travel_date
travel_time
## 1    1442         15A          Mpesa      UZUEHCBUSO    17-10-17
7:15
## 2    5437         14A          Mpesa      TIHLBUSGTE    19-11-17
7:12
## 3    5710          8B          Mpesa      EQX8Q5G19O    26-11-17
7:05
## 4    5777         19A          Mpesa      SGP18CL0ME    27-11-17
7:10
## 5    5778         11A          Mpesa      BM97HFRGL9    27-11-17
7:12
## 6    5777         18B          Mpesa      B6PBDU30IZ    27-11-17
7:10
##   travel_from travel_to car_type max_capacity
## 1      Migori   Nairobi      Bus           49
## 2      Migori   Nairobi      Bus           49
## 3      Keroka   Nairobi      Bus           49
## 4    Homa Bay   Nairobi      Bus           49
```

```
## 5      Migori   Nairobi      Bus           49
## 6    Homa Bay   Nairobi      Bus           49
```

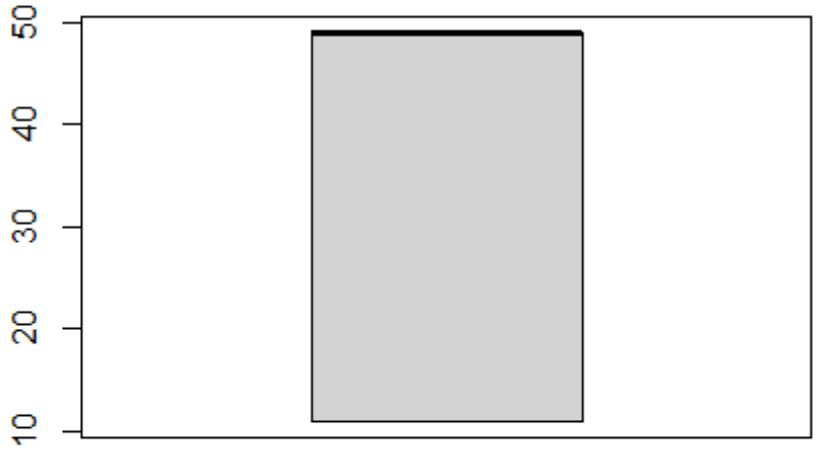
```
dim(bus_dataset)
```

```
## [1] 51645    10
```

```
class(bus_dataset)
```

```
## [1] "data.frame"
```

```
boxplot(bus_dataset$max_capacity)
```




```
boxplot.stats(bus_dataset$max_capacity)$out
```

```
## integer(0)
```

Duplicated Data

Identifying Duplicated Data

```
# Identify duplicate data in the given data frame
# ---
# Creating our vectors
# ---
#
x1 <- c(2, 4, 5, 6)
```

```r
x2 <- c(2, 3, 5, 6)
x3 <- c(2, 4, 5, 6)
x4 <- c(2, 4, 5, 6)
# Create a data frame df from the above vectors
# ---
#
df <- data.frame(rbind(x1, x2, x3, x4))
# Then printing out this dataset
df
```

```
##    X1 X2 X3 X4
## x1  2  4  5  6
## x2  2  3  5  6
## x3  2  4  5  6
## x4  2  4  5  6
```

```r
# Now lets find the duplicated rows in the dataset df
# and assign to a variable duplicated_rows below
# ---
#
duplicated_rows <- df[duplicated(df),]
# Lets print out the variable duplicated_rows and see these duplicated rows
duplicated_rows
```

```
##    X1 X2 X3 X4
## x3  2  4  5  6
## x4  2  4  5  6
```

Removing duplicates

```r
# Removing these duplicated rows in the dataset or
# showing these unique items and assigning to a variable unique_items below
# ---
#
unique_items <- df[!duplicated(df), ]
# What about seeing what these unique items are?
# ---
#
unique_items
```

```
##    X1 X2 X3 X4
## x1  2  4  5  6
## x2  2  3  5  6
```

```r
# using the unique method
# Now there is another way we can also remove duplicated rows
# in the dataset or show the unique items;
# We simply use the unique() function
# ---
#
unique_items2 <- unique(df)
```

```
# After having assigned the unique items to the variable unique_items2,
# we will now print out this variable and have a look at these unique items
unique_items2
```

```
##    X1 X2 X3 X4
## x1  2  4  5  6
## x2  2  3  5  6
```

Challenge 1

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
duplicated_iris <- iris[duplicated(iris),]
duplicated_iris
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 143          5.8         2.7          5.1         1.9 virginica
```

```
unique_iris <- iris[!duplicated(iris),]
head(unique_iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
unique_iris2 <- unique(iris)
head(unique_iris2)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Challenge 2

```
# Reading our dataset
# ---
```

```r
#
video_games <- read.csv('http://bit.ly/VideoGamesDataset')
# Previewing the first 6 records of the video games data set
# ---
#
head(video_games)
```

```
##   X151603712 The.Elder.Scrolls.V.Skyrim purchase  X1.0 X0
## 1  151603712 The Elder Scrolls V Skyrim     play 273.0  0
## 2  151603712                 Fallout 4 purchase   1.0  0
## 3  151603712                 Fallout 4     play  87.0  0
## 4  151603712                     Spore purchase   1.0  0
## 5  151603712                     Spore     play  14.9  0
## 6  151603712          Fallout New Vegas purchase   1.0  0
```

```r
# checking for duplicates
duplicated_videos <- video_games[duplicated(video_games),]
head(duplicated_videos)
```

```
##      X151603712                  The.Elder.Scrolls.V.Skyrim purchase X1.0
X0
## 1968   11373749                 Sid Meier's Civilization IV purchase    1
0
## 1970   11373749 Sid Meier's Civilization IV Beyond the Sword purchase    1
0
## 1972   11373749         Sid Meier's Civilization IV Warlords purchase    1
0
## 2724   56038151                Grand Theft Auto San Andreas purchase    1
0
## 2726   56038151                  Grand Theft Auto Vice City purchase    1
0
## 2728   56038151                       Grand Theft Auto III purchase    1
0
```

```r
# removing the duplicates
videos_games <- video_games[!duplicated(video_games),]
head(videos_games)
```

```
##   X151603712 The.Elder.Scrolls.V.Skyrim purchase  X1.0 X0
## 1  151603712 The Elder Scrolls V Skyrim     play 273.0  0
## 2  151603712                 Fallout 4 purchase   1.0  0
## 3  151603712                 Fallout 4     play  87.0  0
## 4  151603712                     Spore purchase   1.0  0
## 5  151603712                     Spore     play  14.9  0
## 6  151603712          Fallout New Vegas purchase   1.0  0
```

```r
videos_games1 <- unique(video_games)
head(videos_games1)
```

```
##   X151603712 The.Elder.Scrolls.V.Skyrim purchase  X1.0 X0
## 1  151603712 The Elder Scrolls V Skyrim     play 273.0  0
```

```
## 2  151603712                  Fallout 4 purchase   1.0  0
## 3  151603712                  Fallout 4     play  87.0  0
## 4  151603712                      Spore purchase   1.0  0
## 5  151603712                      Spore     play  14.9  0
## 6  151603712          Fallout New Vegas purchase   1.0  0
```