```r
# Missing Data

## 1. Finding Missing Values

#### Finding Missing Values Example 1.1

## Example
# ---
# Lets create a dataset 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 dataset below
dt
```

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

```r
# 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
```

```r
# Example
# ---
# We can also find out total missing values in each column
# by using the function colSums()
# ---
# OUR CODE GOES BELOW
#
colSums(is.na(dt))
```

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

```r
## 2. Dealing with Missing Values

#### Dealing with Missing Values Code Example 2.1
```

```r
## Example
# ---
# Question: Show all rows from the dataset which don't contain any missing values
# ---
# OUR CODE GOES BELOW
#
na.omit(dt)
```

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

```r
#### Dealing with Missing Values Code Example 2.2

## Example
# ---
# Question: Recode/fill the missing value in a column with a number
# ---
# OUR CODE GOES BELOW
#
dt$Age[is.na(dt$Age)] <- 99

dt
```

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

```r
#### Dealing with Missing Values Code Example 2.3

## Example
# ---
# Question: Recode or fill the missing value in a column with the mean value of the column-#-
# ---
# 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)
#
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  49
```

```r
## Challenge 1
# ---
# Question: Using the given bus dataset below, recode the missing values of the payment_method
# and travel_to columns with athen appropriate values
# ---
# OUR CODE GOES BELOW
#

# Lets first of all import our data table
# ---
#
library("data.table")
bus_dataset <- fread('https://raw.githubusercontent.com/cimplival/datasets/master/buses-western-Nairobi

# First check have a look at the dataset
# --
#
head(bus_dataset)
```

```
##    ride_id seat_number payment_method payment_receipt travel_date travel_time
## 1:    1442         15A          Mpesa      UZUEHCBUSO  0017-10-17        7:15
## 2:    5437         14A          Mpesa      TIHLBUSGTE  0019-11-17        7:12
## 3:    5710          8B          Mpesa      EQX8Q5G19O  0026-11-17        7:05
## 4:    5777         19A          Mpesa      SGP18CL0ME  0027-11-17        7:10
## 5:    5778         11A          Mpesa      BM97HFRGL9  0027-11-17        7:12
## 6:    5777         18B          Mpesa      B6PBDU30IZ  0027-11-17        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
```

```r
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
```

```r
## Challenge 2
# ---
# Question: Clean the given dataset
# ---
# Dataset url = http://bit.ly/MS-PropertyDataset
# ---
# OUR CODE GOES BELOW
#
library("data.table")
da_ = fread('https://raw.githubusercontent.com/dataoptimal/posts/master/data%20cleaning%20with%20python
head(da_)
```

```
##          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
```

```
colSums(is.na(da_))
```

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

```
da_$PID[is.na(da_$PID)] <- mean(da_$PID, na.rm = TRUE)
da_$ST_NUM[is.na(da_$ST_NUM)] <- mean(da_$ST_NUM, na.rm = TRUE)
da_
```

```
##          PID   ST_NUM    ST_NAME OWN_OCCUPIED NUM_BEDROOMS NUM_BATH SQ_FT
## 1: 100001000 104.0000     PUTNAM            Y            3        1  1000
## 2: 100002000 197.0000  LEXINGTON            N            3      1.5    --
## 3: 100003000 191.4286  LEXINGTON            N          n/a        1   850
## 4: 100004000 201.0000   BERKELEY           12            1      NaN   700
## 5: 100005000 203.0000   BERKELEY            Y            3        2  1600
## 6: 100006000 207.0000   BERKELEY            Y         <NA>        1   800
## 7: 100007000 191.4286 WASHINGTON                         2   HURLEY   950
## 8: 100008000 213.0000    TREMONT            Y            1        1
## 9: 100009000 215.0000    TREMONT            Y           na        2  1800
```

```
colSums(is.na(da_))
```

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

```
na.omit(da_)
```

```
##          PID   ST_NUM    ST_NAME OWN_OCCUPIED NUM_BEDROOMS NUM_BATH SQ_FT
## 1: 100001000 104.0000     PUTNAM            Y            3        1  1000
## 2: 100002000 197.0000  LEXINGTON            N            3      1.5    --
## 3: 100003000 191.4286  LEXINGTON            N          n/a        1   850
## 4: 100004000 201.0000   BERKELEY           12            1      NaN   700
## 5: 100005000 203.0000   BERKELEY            Y            3        2  1600
## 6: 100007000 191.4286 WASHINGTON                         2   HURLEY   950
## 7: 100008000 213.0000    TREMONT            Y            1        1
## 8: 100009000 215.0000    TREMONT            Y           na        2  1800
```

```
## Challenge 3
# ---
# Question:
# ---
# Dataset url = http://bit.ly/AirQualityDataset
# ---
# OUR CODE GOES BELOW
#
url_data = fread('http://bit.ly/AirQualityDataset')
head(url_data)
```

```
##    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(url_data))
```

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

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

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