HW_1_Kaushik

KK

September 29, 2015

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

2

Installing all required packages

```
#local({r <- getOption("repos");
#       r["CRAN"] <- "http://cran.r-project.org"; options(repos=r)})
#install.packages("dplyr")
library("dplyr")
```

```
## Warning: package 'dplyr' was built under R version 3.2.3
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#install.packages('Hmisc')
library("Hmisc")
```

```
## Loading required package: grid
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.2.3
```

```
## Warning: replacing previous import by 'ggplot2::unit' when loading 'Hmisc'
```

```
## Warning: replacing previous import by 'ggplot2::arrow' when loading 'Hmisc'
```

```
## Warning: replacing previous import by 'scales::alpha' when loading 'Hmisc'

##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:dplyr':
##
##     combine, src, summarize
##
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
#install.packages("VIM")
library(VIM)
```

```
## Loading required package: colorspace
## Loading required package: data.table
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:dplyr':
##
##     between, last
##
## VIM is ready to use.
##  Since version 4.0.0 the GUI is in its own package VIMGUI.
##
##           Please use the package to use the new (and old) GUI.
##
## Suggestions and bug-reports can be submitted at: https://github.com/alexkowa/VIM/issues
##
## Attaching package: 'VIM'
##
## The following object is masked from 'package:datasets':
##
##     sleep
```

```
#install.packages("mice")
library(mice)
```

```
## Loading required package: Rcpp

## Warning: package 'Rcpp' was built under R version 3.2.3

## mice 2.22 2014-06-10
```

```
#install.packages("corrplot")
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.2.3
```

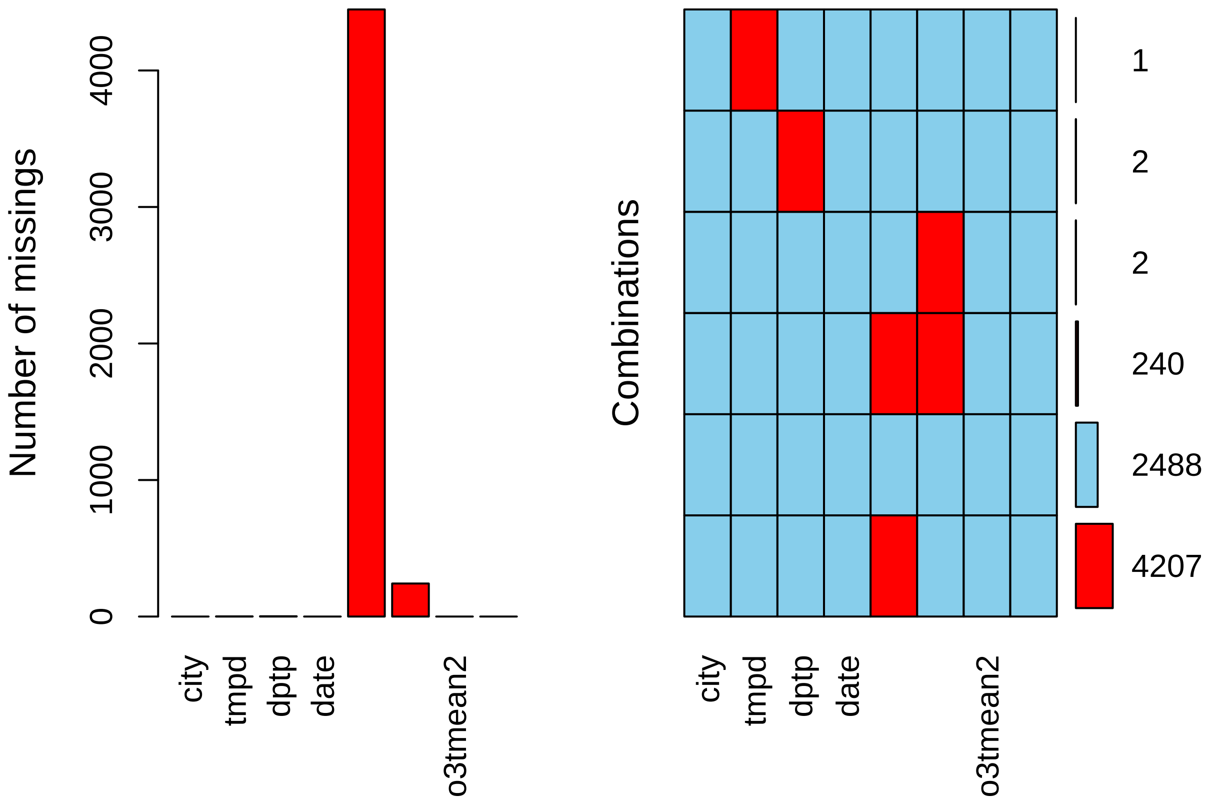
```r
#install.packages("PerformanceAnalytics")
library(PerformanceAnalytics)
```

```
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
##
## Attaching package: 'xts'
##
## The following object is masked from 'package:data.table':
##
##     last
##
## The following objects are masked from 'package:dplyr':
##
##     first, last
##
##
## Attaching package: 'PerformanceAnalytics'
##
## The following object is masked from 'package:graphics':
##
##     legend
```

```r
chicago =tbl_df(chicago)
#gives the structure of the data along with the datatypes
str(chicago)
```

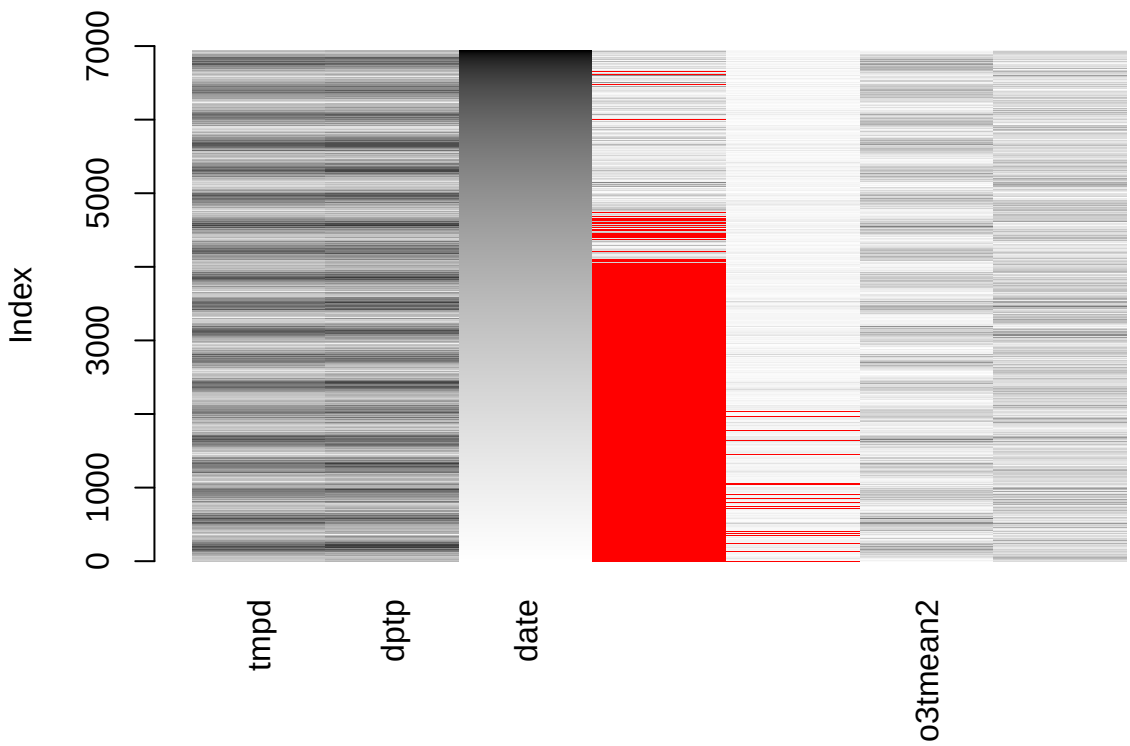
```
## Classes 'tbl_df', 'tbl' and 'data.frame':    6940 obs. of  8 variables:
##  $ city      : chr  "chic" "chic" "chic" "chic" ...
##  $ tmpd      : num  31.5 33 33 29 32 40 34.5 29 26.5 32.5 ...
##  $ dptp      : num  31.5 29.9 27.4 28.6 28.9 ...
##  $ date      : Date, format: "1987-01-01" "1987-01-02" ...
##  $ pm25tmean2: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ pm10tmean2: num  34 NA 34.2 47 NA ...
##  $ o3tmean2  : num  4.25 3.3 3.33 4.38 4.75 ...
##  $ no2tmean2 : num  20 23.2 23.8 30.4 30.3 ...
```

```r
#colnames(chicago)
# tmpd  dptp    date    pm25tmean2  pm10tmean2  o3tmean2    no2tmean2

#visually detecting missing values
#aggr function displays the number of missing values and also the number of collectively missing values
aggr(chicago, prop=FALSE, numbers=TRUE)
```




```r
#Plots a color matrix, representing the values stored in the chicago dataset
matrixplot(chicago[,c(2:8)])
```

```
## Warning in hex(RGB(r, g, b), gamma = gamma, fixup = fixup, ...): 'gamma' is
## deprecated and has no effect

## Warning in hex(RGB(r, g, b), gamma = gamma, fixup = fixup, ...): 'gamma' is
## deprecated and has no effect

## Warning in hex(RGB(r, g, b), gamma = gamma, fixup = fixup, ...): 'gamma' is
## deprecated and has no effect

## Warning in hex(RGB(r, g, b), gamma = gamma, fixup = fixup, ...): 'gamma' is
## deprecated and has no effect

## Warning in hex(RGB(r, g, b), gamma = gamma, fixup = fixup, ...): 'gamma' is
## deprecated and has no effect

## Warning in hex(RGB(r, g, b), gamma = gamma, fixup = fixup, ...): 'gamma' is
## deprecated and has no effect

## Warning in hex(RGB(r, g, b), gamma = gamma, fixup = fixup, ...): 'gamma' is
## deprecated and has no effect
```




```
# We see that there are many missing values. let us get a count of them first
sum(is.na(chicago$tmpd)) # 4% of the values are missing
```

```
## [1] 1
```

```
sum(is.na(chicago$dptp))
```

```
## [1] 2
```

```
sum(is.na(chicago$date))
```

```
## [1] 0
```

```
sum(is.na(chicago$pm25tmean2)) # 60-70% of the values are missing. Which is huge and makes it more inte
```

```
## [1] 4447
```

```
sum(is.na(chicago$pm10tmean2)) # 4% of the values are missing
```

```
## [1] 242
```

```r
sum(is.na(chicago$o3tmean2))
```

```
## [1] 0
```

```r
sum(is.na(chicago$no2tmean2))
```

```
## [1] 0
```

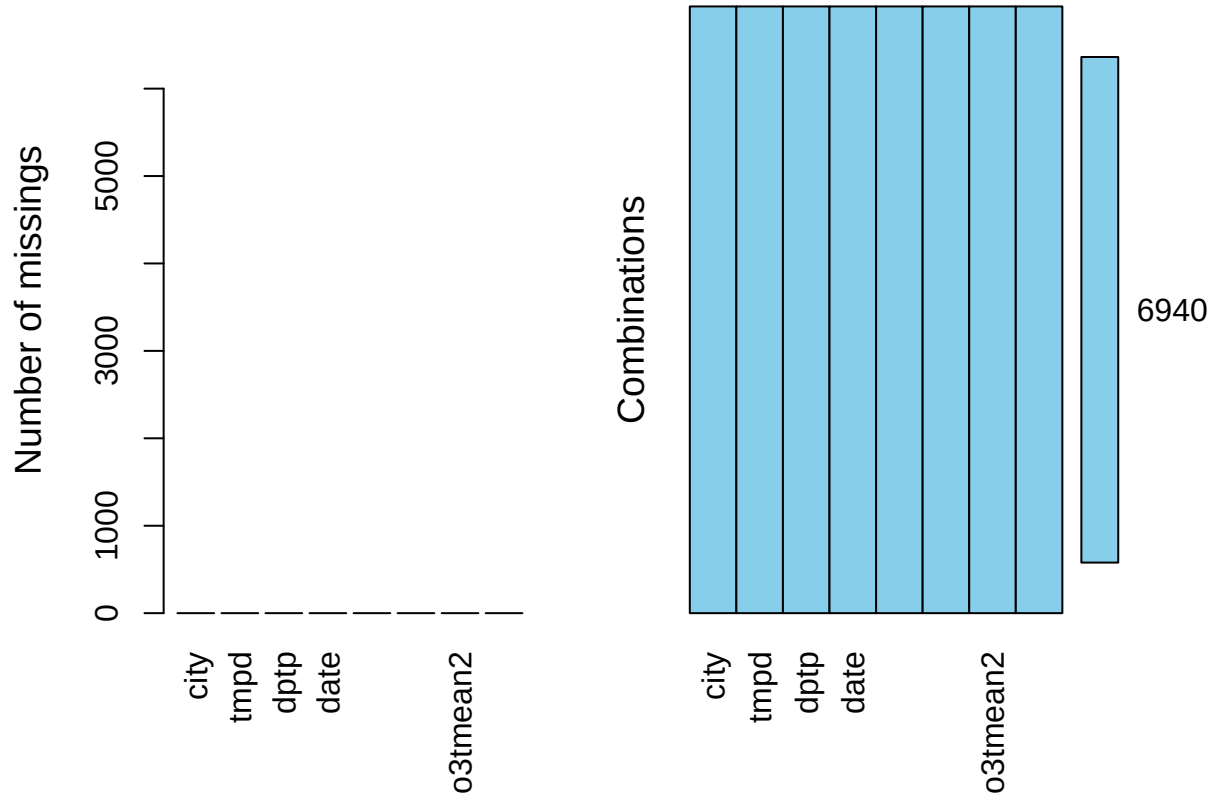
```r
#Imputing missing values. Let us follow three procedures and choose the best one.

#Procedure_1: Imputing median into the missing values
#Defining a fucntion to impute median into the missing values

median_imputed_data = chicago

impute_median = function (a){
  missing = is.na(a)
  imputed = a
  imputed[missing] = median(a,  na.rm=TRUE) # Replace median where the value is missing
  return (imputed)
}
# Testing the median function
k=as.numeric (c("1", "2", "5","three", "4"))
```

```
## Warning: NAs introduced by coercion
```

```r
impute_median(k)
```

```
## [1] 1 2 5 3 4
```

```r
#Imputing the median by calling the fucntion
median_imputed_data$pm25tmean2 = impute_median(median_imputed_data$pm25tmean2)
median_imputed_data$pm10tmean2 = impute_median(median_imputed_data$pm10tmean2)
median_imputed_data$tmpd = impute_median(median_imputed_data$tmpd)
median_imputed_data$dptp = impute_median(median_imputed_data$dptp)

# Now see if there are missing values. Visually plotting
aggr(median_imputed_data, prop=FALSE, numbers=TRUE)
```




```
#Procedure_2: Imputing missing values using mice package

chicago1 = chicago[,c(2,3,5,6,7,8)]

imp=mice(chicago1, seed=1234)
```

```
##
##  iter imp variable
##    1   1  tmpd  dptp  pm25tmean2  pm10tmean2
##    1   2  tmpd  dptp  pm25tmean2  pm10tmean2
##    1   3  tmpd  dptp  pm25tmean2  pm10tmean2
##    1   4  tmpd  dptp  pm25tmean2  pm10tmean2
##    1   5  tmpd  dptp  pm25tmean2  pm10tmean2
##    2   1  tmpd  dptp  pm25tmean2  pm10tmean2
##    2   2  tmpd  dptp  pm25tmean2  pm10tmean2
##    2   3  tmpd  dptp  pm25tmean2  pm10tmean2
##    2   4  tmpd  dptp  pm25tmean2  pm10tmean2
##    2   5  tmpd  dptp  pm25tmean2  pm10tmean2
##    3   1  tmpd  dptp  pm25tmean2  pm10tmean2
##    3   2  tmpd  dptp  pm25tmean2  pm10tmean2
##    3   3  tmpd  dptp  pm25tmean2  pm10tmean2
##    3   4  tmpd  dptp  pm25tmean2  pm10tmean2
##    3   5  tmpd  dptp  pm25tmean2  pm10tmean2
##    4   1  tmpd  dptp  pm25tmean2  pm10tmean2
##    4   2  tmpd  dptp  pm25tmean2  pm10tmean2
##    4   3  tmpd  dptp  pm25tmean2  pm10tmean2
```

```
##   4   4  tmpd  dptp  pm25tmean2  pm10tmean2
##   4   5  tmpd  dptp  pm25tmean2  pm10tmean2
##   5   1  tmpd  dptp  pm25tmean2  pm10tmean2
##   5   2  tmpd  dptp  pm25tmean2  pm10tmean2
##   5   3  tmpd  dptp  pm25tmean2  pm10tmean2
##   5   4  tmpd  dptp  pm25tmean2  pm10tmean2
##   5   5  tmpd  dptp  pm25tmean2  pm10tmean2
```

```
fit= with(imp,lm(chicago1$pm25tmean2 ~chicago1$pm10tmean2 + chicago1$no2tmean2 ))
pooled = pool(fit)
summary(pooled)
```

```
##                            est         se         t       df  Pr(>|t|)
## (Intercept)          0.5061427 0.44121264  1.147163 2485.738 0.2514248
## chicago1$pm10tmean2 0.2784793 0.01022208 27.242919 2485.738 0.0000000
## chicago1$no2tmean2  0.3067993 0.02046048 14.994722 2485.738 0.0000000
##                             lo 95     hi 95 nmis        fmi lambda
## (Intercept)          -0.3590395 1.3713249   NA 0.00080362      0
## chicago1$pm10tmean2  0.2584346 0.2985239   NA 0.00080362      0
## chicago1$no2tmean2   0.2666779 0.3469206   NA 0.00080362      0
```

```
mice_imputed_data = complete(imp, action=3)

# Imputing values retaining the date
chicago_date=chicago
chicago_date$date <- as.numeric(chicago_date$date)
chicago_date=chicago_date[,c(2,3,4,5,6,7,8)]
imp=mice(chicago_date, seed=1234)
```

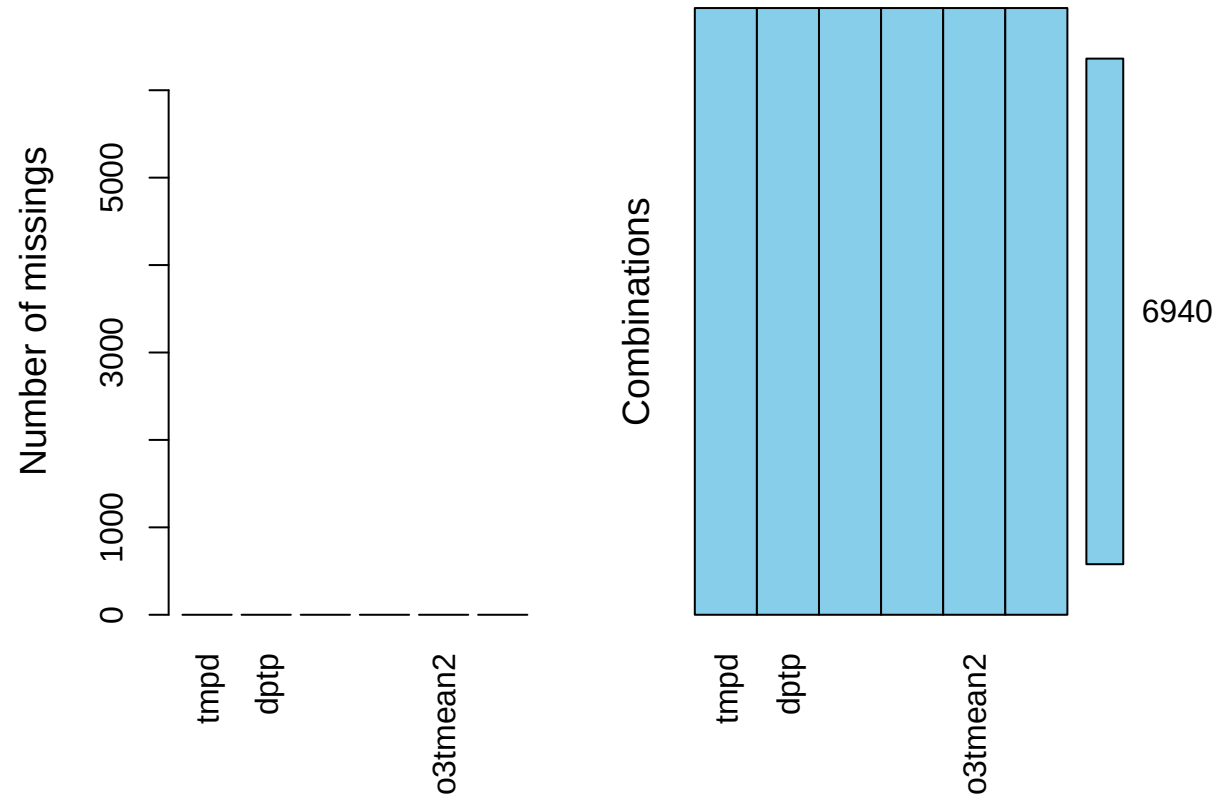
```
##
##  iter imp variable
##   1   1  tmpd  dptp  pm25tmean2  pm10tmean2
##   1   2  tmpd  dptp  pm25tmean2  pm10tmean2
##   1   3  tmpd  dptp  pm25tmean2  pm10tmean2
##   1   4  tmpd  dptp  pm25tmean2  pm10tmean2
##   1   5  tmpd  dptp  pm25tmean2  pm10tmean2
##   2   1  tmpd  dptp  pm25tmean2  pm10tmean2
##   2   2  tmpd  dptp  pm25tmean2  pm10tmean2
##   2   3  tmpd  dptp  pm25tmean2  pm10tmean2
##   2   4  tmpd  dptp  pm25tmean2  pm10tmean2
##   2   5  tmpd  dptp  pm25tmean2  pm10tmean2
##   3   1  tmpd  dptp  pm25tmean2  pm10tmean2
##   3   2  tmpd  dptp  pm25tmean2  pm10tmean2
##   3   3  tmpd  dptp  pm25tmean2  pm10tmean2
##   3   4  tmpd  dptp  pm25tmean2  pm10tmean2
##   3   5  tmpd  dptp  pm25tmean2  pm10tmean2
##   4   1  tmpd  dptp  pm25tmean2  pm10tmean2
##   4   2  tmpd  dptp  pm25tmean2  pm10tmean2
##   4   3  tmpd  dptp  pm25tmean2  pm10tmean2
##   4   4  tmpd  dptp  pm25tmean2  pm10tmean2
##   4   5  tmpd  dptp  pm25tmean2  pm10tmean2
##   5   1  tmpd  dptp  pm25tmean2  pm10tmean2
##   5   2  tmpd  dptp  pm25tmean2  pm10tmean2
```

```
##   5   3  tmpd  dptp  pm25tmean2  pm10tmean2
##   5   4  tmpd  dptp  pm25tmean2  pm10tmean2
##   5   5  tmpd  dptp  pm25tmean2  pm10tmean2
```

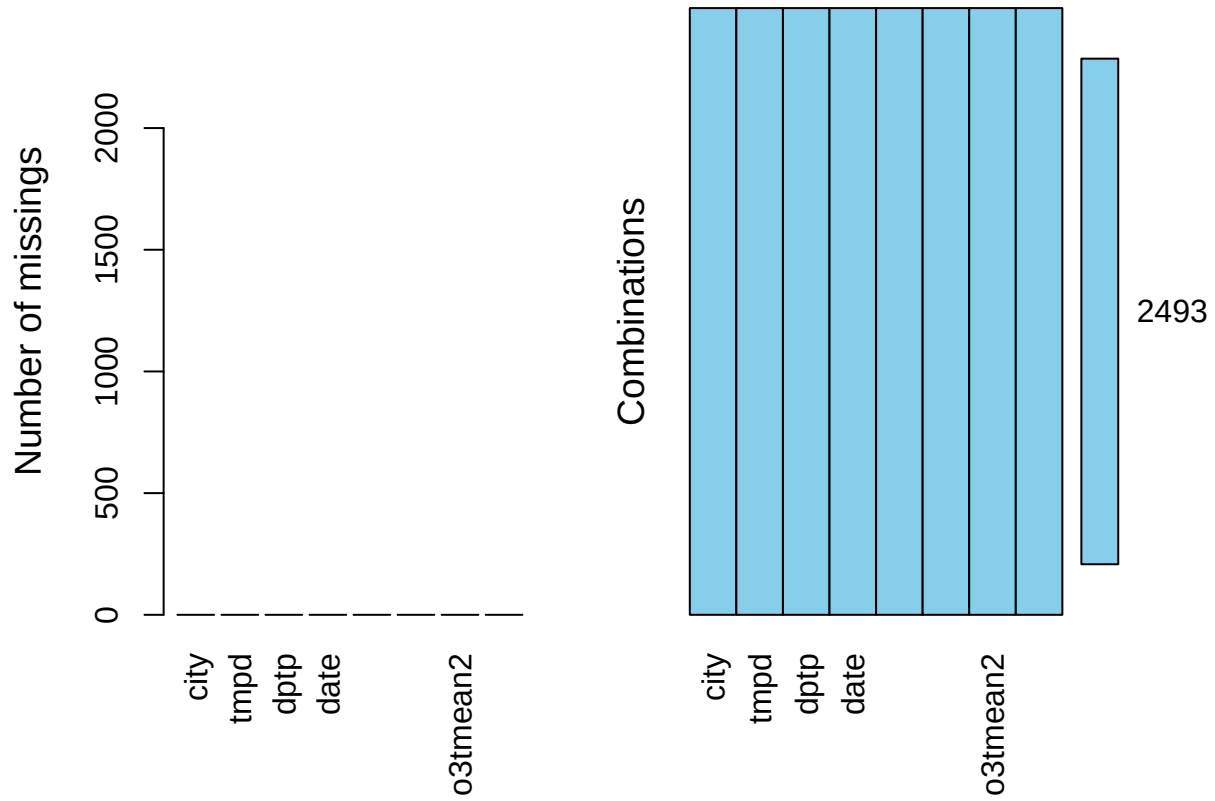
```
fit= with(imp,lm(chicago_date$pm25tmean2 ~chicago_date$pm10tmean2 + chicago_date$no2tmean2 ))
pooled = pool(fit)
summary(pooled)
```

```
##                               est         se         t       df  Pr(>|t|)
## (Intercept)             0.5061427 0.44121264  1.147163 2485.738 0.2514248
## chicago_date$pm10tmean2 0.2784793 0.01022208 27.242919 2485.738 0.0000000
## chicago_date$no2tmean2  0.3067993 0.02046048 14.994722 2485.738 0.0000000
##                               lo 95     hi 95 nmis        fmi lambda
## (Intercept)             -0.3590395 1.3713249   NA 0.00080362      0
## chicago_date$pm10tmean2  0.2584346 0.2985239   NA 0.00080362      0
## chicago_date$no2tmean2   0.2666779 0.3469206   NA 0.00080362      0
```

```
mice_imputed_data_date = complete(imp, action=3)


# Now see if there are missing values. Visually plotting
aggr(mice_imputed_data, prop=FALSE, numbers=TRUE)
```

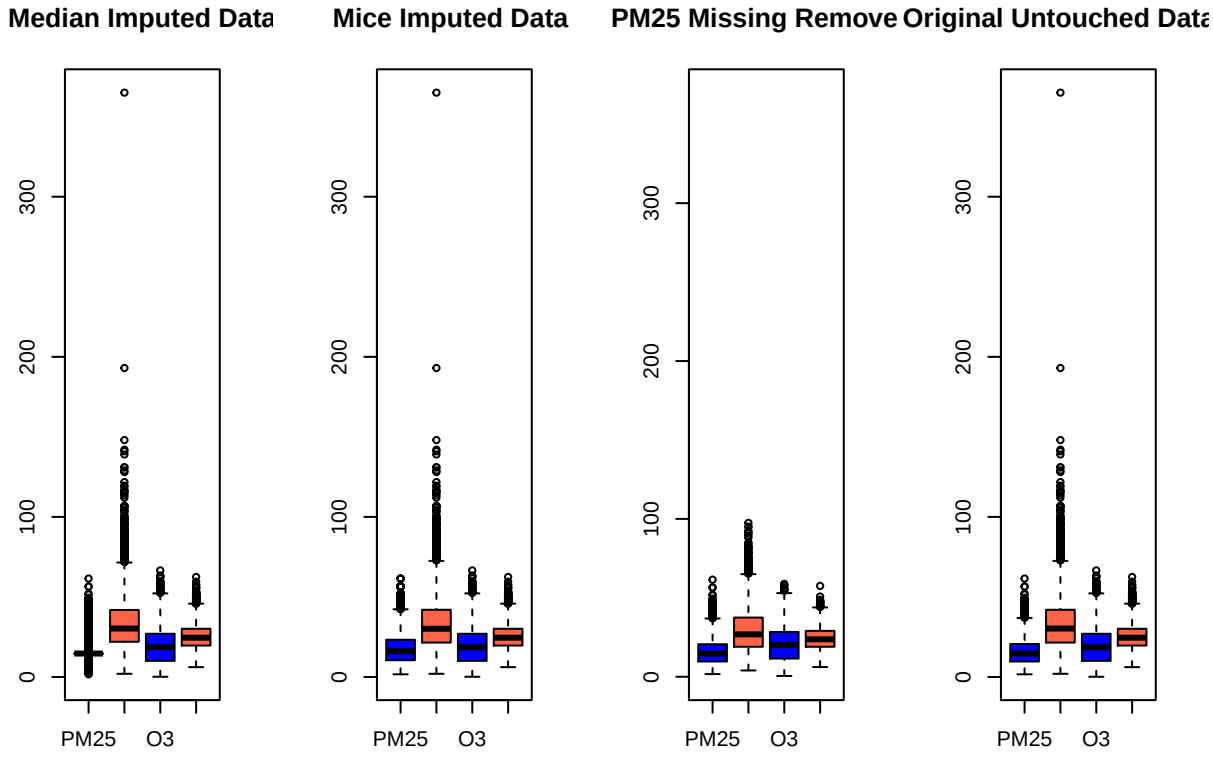


```r
#Procedure_3: Ignoring and removing the missing values

ignore_missing_data = chicago

# This fucntion removes rows based on Nas in columns
missing_fun = function(data, desiredCols) {
  completeVec = complete.cases(data[, desiredCols])
  return(data[completeVec, ])
}

#Removing rows
ignore_missing_data =  missing_fun(chicago, "pm25tmean2")

#Checking for missing values. We still see that there are 1 or 2 values missing. Replacing them median
ignore_missing_data$pm10tmean2 = impute_median(ignore_missing_data$pm10tmean2)
ignore_missing_data$tmpd = impute_median(ignore_missing_data$tmpd)
ignore_missing_data$dptp = impute_median(ignore_missing_data$dptp)

# Now see if there are missing values. Visually plotting
aggr(ignore_missing_data, prop=FALSE, numbers=TRUE)
```




```r
# Now let us comapre the variation in the variables imputed using the three methods.
#Let us check the box plots to see if any outlier treatment has to be done

old.par = par(mfrow=c(1, 4))
```

```r
boxplot(median_imputed_data$pm25tmean2,median_imputed_data$pm10tmean2,median_imputed_data$o3tmean2,media
        names=c("PM25","PM10","O3","NO2"),col=c("blue","tomato"),
        main="Median Imputed Data",outer = TRUE)

boxplot(mice_imputed_data$pm25tmean2,mice_imputed_data$pm10tmean2,mice_imputed_data$o3tmean2,mice_imput
        names=c("PM25","PM10","O3","NO2"),col=c("blue","tomato"),  main="Mice Imputed Data ",outer = TR

boxplot(ignore_missing_data$pm25tmean2,ignore_missing_data$pm10tmean2,ignore_missing_data$o3tmean2,igno
        names=c("PM25","PM10","O3","NO2"),col=c("blue","tomato"), ylim=c(0,370),  main="PM25 Missing Re

boxplot(chicago$pm25tmean2,chicago$pm10tmean2,chicago$o3tmean2,chicago$no2tmean2,
        names=c("PM25","PM10","O3","NO2"),col=c("blue","tomato"),  main="Original Untouched Dataset",ou
```




```r
par(old.par)

#Checking the how the correlation factors changes on these 3 datasets

mice_cor = cor(mice_imputed_data)
par(oma=c(0,0,2,0)) # Specifying the outer margins

med_imp = cor(median_imputed_data[,c(2,3,5,6,7,8)])
par(oma=c(0,0,2,0))

missing_rem =cor(ignore_missing_data[,c(2,3,5,6,7,8)])
par(oma=c(0,0,2,0))
```

```
#arranging the correlation plots side by side
old.par = par(mfrow=c(1, 3))
corrplot(mice_cor, method = "number",title = 'Mice Imputed Data                    Median Imputed Data
```

```
## Warning in text.default(pos.xlabel[, 1], pos.xlabel[, 2], newcolnames, srt
## = tl.srt, : "outer" is not a graphical parameter
```

```
## Warning in text.default(pos.ylabel[, 1], pos.ylabel[, 2], newrownames, col
## = tl.col, : "outer" is not a graphical parameter
```

```
corrplot(med_imp, method = "number",outer=TRUE)
```

```
## Warning in text.default(pos.xlabel[, 1], pos.xlabel[, 2], newcolnames, srt
## = tl.srt, : "outer" is not a graphical parameter
```

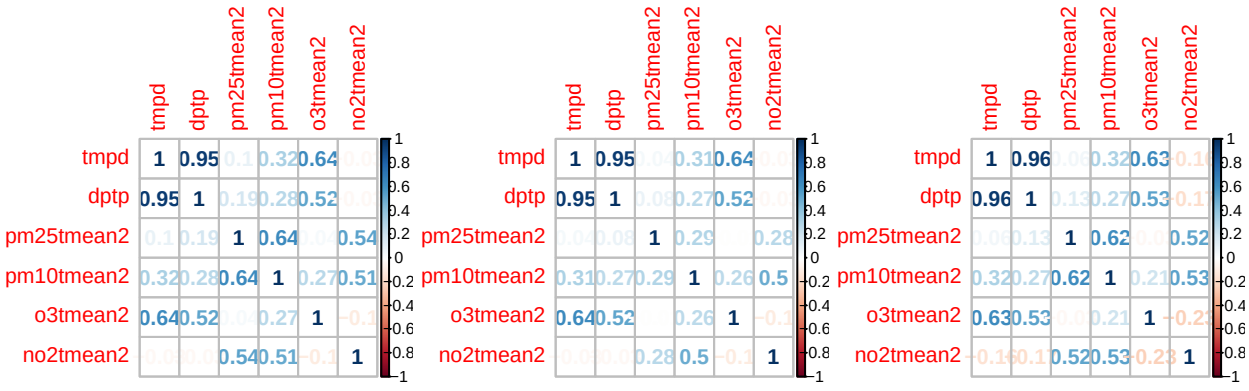
```
## Warning in text.default(pos.ylabel[, 1], pos.ylabel[, 2], newrownames, col
## = tl.col, : "outer" is not a graphical parameter
```

```
corrplot(missing_rem, method = "number",outer=TRUE)
```

```
## Warning in text.default(pos.xlabel[, 1], pos.xlabel[, 2], newcolnames, srt
## = tl.srt, : "outer" is not a graphical parameter
```

```
## Warning in text.default(pos.ylabel[, 1], pos.ylabel[, 2], newrownames, col
## = tl.col, : "outer" is not a graphical parameter
```

Mice Imputed Data **Median Imputed Data** **Ignore Missing Data**



```r
par(old.par)

# One important observation is that after imputing the values the correlation coeffcients are almost th
# Looking at the summary stats
summary(median_imputed_data)
```

```
##      city                tmpd             dptp             date
##  Length:6940        Min.   :-16.00   Min.   :-25.62   Min.   :1987-01-01
##  Class :character   1st Qu.: 35.00   1st Qu.: 27.00   1st Qu.:1991-10-01
##  Mode  :character   Median : 51.00   Median : 39.88   Median :1996-07-01
##                     Mean   : 50.31   Mean   : 40.34   Mean   :1996-07-01
##                     3rd Qu.: 67.00   3rd Qu.: 55.75   3rd Qu.:2001-04-01
##                     Max.   : 92.00   Max.   : 78.25   Max.   :2005-12-31
##    pm25tmean2       pm10tmean2        o3tmean2         no2tmean2
##  Min.   : 1.70   Min.   :  2.00   Min.   : 0.1528   Min.   : 6.158
##  1st Qu.:14.66   1st Qu.: 22.00   1st Qu.:10.0729   1st Qu.:19.654
##  Median :14.66   Median : 30.28   Median :18.5218   Median :24.556
##  Mean   :15.22   Mean   : 33.77   Mean   :19.4355   Mean   :25.232
##  3rd Qu.:14.66   3rd Qu.: 41.86   3rd Qu.:27.0010   3rd Qu.:30.139
##  Max.   :61.50   Max.   :365.00   Max.   :66.5875   Max.   :62.480
```

```r
summary(ignore_missing_data)
```

```
##      city                tmpd             dptp             date
##  Length:2493        Min.   :-3.00   Min.   :-12.70   Min.   :1998-01-05
##  Class :character   1st Qu.:36.00   1st Qu.: 27.30   1st Qu.:2000-09-17
##  Mode  :character   Median :53.00   Median : 41.90   Median :2002-06-04
##                     Mean   :51.43   Mean   : 41.12   Mean   :2002-05-13
##                     3rd Qu.:68.00   3rd Qu.: 56.60   3rd Qu.:2004-03-15
##                     Max.   :90.00   Max.   : 76.60   Max.   :2005-12-31
##    pm25tmean2       pm10tmean2       o3tmean2       no2tmean2
##  Min.   : 1.70   Min.   : 4.00   Min.   : 0.50   Min.   : 6.158
##  1st Qu.: 9.70   1st Qu.:19.00   1st Qu.:11.54   1st Qu.:19.006
##  Median :14.66   Median :26.86   Median :20.17   Median :23.719
##  Mean   :16.23   Mean   :29.74   Mean   :20.59   Mean   :24.256
##  3rd Qu.:20.60   3rd Qu.:37.50   3rd Qu.:28.39   3rd Qu.:29.062
##  Max.   :61.50   Max.   :97.50   Max.   :58.84   Max.   :57.564
```

```r
summary(mice_imputed_data)
```

```
##       tmpd             dptp          pm25tmean2      pm10tmean2
##  Min.   :-16.00   Min.   :-25.62   Min.   : 1.70   Min.   :  2.00
##  1st Qu.: 35.00   1st Qu.: 27.00   1st Qu.:10.50   1st Qu.: 21.50
##  Median : 51.00   Median : 39.88   Median :16.10   Median : 30.15
##  Mean   : 50.31   Mean   : 40.34   Mean   :18.10   Mean   : 33.89
##  3rd Qu.: 67.00   3rd Qu.: 55.75   3rd Qu.:23.25   3rd Qu.: 42.00
##  Max.   : 92.00   Max.   : 78.25   Max.   :61.50   Max.   :365.00
##     o3tmean2         no2tmean2
##  Min.   : 0.1528   Min.   : 6.158
##  1st Qu.:10.0729   1st Qu.:19.654
##  Median :18.5218   Median :24.556
```

```
##  Mean   :19.4355   Mean   :25.232
##  3rd Qu.:27.0010   3rd Qu.:30.139
##  Max.   :66.5875   Max.   :62.480
```

```
summary(chicago)
```

```
##      city                tmpd             dptp             date
##  Length:6940        Min.   :-16.00   Min.   :-25.62   Min.   :1987-01-01
##  Class :character   1st Qu.: 35.00   1st Qu.: 27.00   1st Qu.:1991-10-01
##  Mode  :character   Median : 51.00   Median : 39.88   Median :1996-07-01
##                     Mean   : 50.31   Mean   : 40.34   Mean   :1996-07-01
##                     3rd Qu.: 67.00   3rd Qu.: 55.75   3rd Qu.:2001-04-01
##                     Max.   : 92.00   Max.   : 78.25   Max.   :2005-12-31
##                     NA's   :1        NA's   :2
##    pm25tmean2      pm10tmean2        o3tmean2         no2tmean2
##  Min.   : 1.70   Min.   :  2.00   Min.   : 0.1528   Min.   : 6.158
##  1st Qu.: 9.70   1st Qu.: 21.50   1st Qu.:10.0729   1st Qu.:19.654
##  Median :14.66   Median : 30.28   Median :18.5218   Median :24.556
##  Mean   :16.23   Mean   : 33.90   Mean   :19.4355   Mean   :25.232
##  3rd Qu.:20.60   3rd Qu.: 42.00   3rd Qu.:27.0010   3rd Qu.:30.139
##  Max.   :61.50   Max.   :365.00   Max.   :66.5875   Max.   :62.480
##  NA's   :4447    NA's   :242
```

```
#Chossing the best data set in this case
#From the correlation plot and summary we can infer that mice dataset is closest to the actual data.
#However mice predicts the missing values using a linear relationship and multiple iterations.
#Assumption: The data is missing. Not that missing data is not captured.
#As we see the data b/w 1987 and 1998 for PM25 variable is missing. As PM25 captures 70% of data we can
#So using the Mice imputed data might be ideal. Imputing median might bias our analysis towards median

#Visualization of a Correlation Matrix.
#On top the (absolute) value of the correlation plus the result of the cor.test as stars. On bottom, th
#chart.Correlation(mice_cor, histogram=TRUE)

"Certain Outlier treatment has been done for 'pm25tmean2' values in the next question after interpretin
'seemingly high' value of PM2.5"
```

```
## [1] "Certain Outlier treatment has been done for 'pm25tmean2' values in the next question after inte
```

```
##3

sub_pm25=arrange(mice_imputed_data, desc(pm25tmean2))
head(sub_pm25,n = 10)
```

```
##    tmpd   dptp pm25tmean2 pm10tmean2  o3tmean2 no2tmean2
## 1  66.0 47.750       61.5       79.0 26.000000  47.05442
## 2  77.0 65.625       61.5       79.0 22.191498  32.66848
## 3  62.5 48.625       61.5       71.5 15.675084  48.28212
## 4  36.0 32.100       61.5       57.5  4.812500  41.52778
## 5  61.5 49.875       56.5      106.0 31.529762  55.41667
## 6  77.5 61.375       56.5       80.0 47.268914  51.17500
## 7  49.5 33.375       56.5      106.5 15.115609  40.27065
```

```
## 8   8.5  2.750       56.5       87.0  4.659420  42.88504
## 9  32.0 29.375       56.5       77.0  8.666667  49.01894
## 10 85.0 58.375       56.5      116.0 58.886425  46.27342
```

```r
summary(mice_imputed_data)
```

```
##       tmpd             dptp          pm25tmean2      pm10tmean2
##  Min.   :-16.00   Min.   :-25.62   Min.   : 1.70   Min.   :  2.00
##  1st Qu.: 35.00   1st Qu.: 27.00   1st Qu.:10.50   1st Qu.: 21.50
##  Median : 51.00   Median : 39.88   Median :16.10   Median : 30.15
##  Mean   : 50.31   Mean   : 40.34   Mean   :18.10   Mean   : 33.89
##  3rd Qu.: 67.00   3rd Qu.: 55.75   3rd Qu.:23.25   3rd Qu.: 42.00
##  Max.   : 92.00   Max.   : 78.25   Max.   :61.50   Max.   :365.00
##     o3tmean2         no2tmean2
##  Min.   : 0.1528   Min.   : 6.158
##  1st Qu.:10.0729   1st Qu.:19.654
##  Median :18.5218   Median :24.556
##  Mean   :19.4355   Mean   :25.232
##  3rd Qu.:27.0010   3rd Qu.:30.139
##  Max.   :66.5875   Max.   :62.480
```

```r
#Checking the quintile distribution
quantile(mice_imputed_data$pm25tmean2,probs=seq(0,1,by=.01)) # checking the quintile distribution
```

```
##        0%        1%        2%        3%        4%        5%        6%
##  1.700000  4.124339  4.800000  5.300000  5.700000  6.011875  6.400000
##        7%        8%        9%       10%       11%       12%       13%
##  6.600000  6.800000  7.050000  7.300000  7.537500  7.700000  7.885839
##       14%       15%       16%       17%       18%       19%       20%
##  8.100000  8.398125  8.600000  8.700000  9.000000  9.283143  9.471429
##       21%       22%       23%       24%       25%       26%       27%
##  9.700000  9.900000 10.100000 10.300000 10.500000 10.800000 10.936984
##       28%       29%       30%       31%       32%       33%       34%
## 11.239429 11.407750 11.600000 11.700000 11.900000 12.162500 12.400000
##       35%       36%       37%       38%       39%       40%       41%
## 12.700000 12.938000 13.200000 13.443250 13.700000 13.886667 14.150000
##       42%       43%       44%       45%       46%       47%       48%
## 14.400000 14.662500 14.828571 15.046786 15.250000 15.500000 15.764000
##       49%       50%       51%       52%       53%       54%       55%
## 15.900000 16.100000 16.350000 16.669500 16.800000 17.100000 17.400000
##       56%       57%       58%       59%       60%       61%       62%
## 17.700000 17.900000 18.100000 18.300000 18.550000 18.800000 19.100000
##       63%       64%       65%       66%       67%       68%       69%
## 19.400000 19.642643 19.800000 20.037500 20.400000 20.600000 21.000000
##       70%       71%       72%       73%       74%       75%       76%
## 21.400000 21.742857 22.162500 22.600000 22.900000 23.250000 23.800000
##       77%       78%       79%       80%       81%       82%       83%
## 24.300000 24.900000 25.400000 25.600000 26.242000 26.750000 27.300000
##       84%       85%       86%       87%       88%       89%       90%
## 27.628571 28.251250 28.900000 29.400000 30.175000 30.900000 31.710000
##       91%       92%       93%       94%       95%       96%       97%
## 32.549000 33.700000 34.740500 36.749000 38.100000 39.512500 40.900000
##       98%       99%      100%
## 44.957143 49.471429 61.500000
```

```
#Spotting the outliers in PM25 using a box plot
#boxplot(mice_imputed_data$pm25tmean2,mice_imputed_data$pm10tmean2,
#        names=c("PM25"),col=c("tomato","blue"),name="PM25 and PM10")

#mean_1 = mean(mice_imputed_data$pm25tmean2)#mean is 18.1
#sd_1 = sd(mice_imputed_data$pm25tmean2) #sd is 9.95

#There are certain observations which are 4 sds greater than mean, they can be replaced with median ins
# However since 61.5 is not very far off from 56.5, we can retain these values without making any chang
#mice_imputed_data$pm25tmean2[mice_imputed_data$pm25tmean2>(mean_1 + 4*sd_1)] = median(mice_imputed_dat

# Now look at the top 10 values of PM10
a=arrange(mice_imputed_data, desc(pm10tmean2))
head(a,n = 10)
```

```
##    tmpd   dptp pm25tmean2 pm10tmean2 o3tmean2 no2tmean2
## 1  67.5 53.250   49.47143        365 48.80556  12.24955
## 2  54.0 34.250   49.47143        193 14.18342  29.03333
## 3  76.0 52.750   40.90000        148 50.90476  39.35218
## 4  76.5 54.000   49.47143        142 49.20168  44.59664
## 5  64.5 52.500   40.90000        141 32.91735  25.35833
## 6  79.0 51.125   38.90000        139 54.17857  34.67977
## 7  83.5 65.750   56.50000        131 39.51852  55.48958
## 8  47.0 33.125   40.90000        131 20.07955  31.56944
## 9  67.0 56.875   38.90000        129 17.25928  35.67500
## 10 64.0 48.500   40.90000        128 33.61286  34.27273
```

```
#Let us also look at PM10
mean_2 = mean(mice_imputed_data$pm10tmean2)#mean is 33.89
sd_2 = sd(mice_imputed_data$pm10tmean2) #sd is 17.95
quantile(mice_imputed_data$pm10tmean2,probs=seq(0,1,by=.05)) # checking the quintile distribution
```

```
##        0%        5%       10%       15%       20%       25%       30%
##   2.00000  12.00000  15.49000  17.50000  19.50000  21.50000  23.50000
##       35%       40%       45%       50%       55%       60%       65%
##  25.00000  27.00000  28.50000  30.14583  32.25000  34.50000  36.86339
##       70%       75%       80%       85%       90%       95%      100%
##  39.00000  42.00000  46.00000  50.68667  57.01429  67.50000 365.00000
```

```
#Replacing the top two observations (treat them as outleirs) which are way away more than 10 standard d
mice_imputed_data_date$pm10tmean2[mice_imputed_data_date$pm10tmean2>(150)] = median(mice_imputed_data_da

#Cross Check PM10 now
a=arrange(mice_imputed_data_date, desc(pm10tmean2))
head(a,n = 10)
```

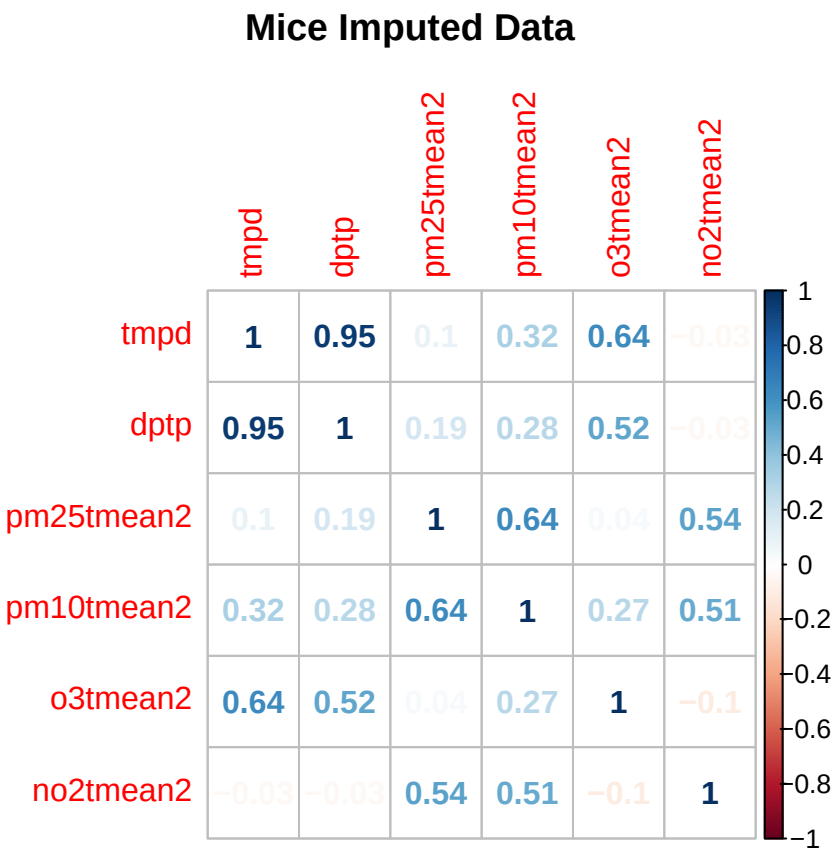
```
##    tmpd   dptp date pm25tmean2 pm10tmean2 o3tmean2 no2tmean2
## 1  76.0 52.750 6725   34.20000        148 50.90476  39.35218
## 2  76.5 54.000 6726   34.20000        142 49.20168  44.59664
## 3  61.5 48.250 6842   49.47143        142 23.78551  39.39881
## 4  64.5 52.500 8156   34.20000        141 32.91735  25.35833
## 5  79.0 51.125 6732   38.90000        139 54.17857  34.67977
```

```
## 6  83.5 65.750 7883   34.20000         131 39.51852  55.48958
## 7  47.0 33.125 8859   56.50000         131 20.07955  31.56944
## 8  67.0 56.875 8331   40.90000         129 17.25928  35.67500
## 9  64.0 53.000 6317   33.00000         128 32.83929  45.15972
## 10 64.0 48.500 9597   38.90000         128 33.61286  34.27273
```

```
#All other variables look fine without any outliers. Hence we can further proceed with our analysis.

##4
#Looking and at the correlation plot to find dependencies
corrplot(mice_cor, method = "number",title = 'Mice Imputed Data', outer=TRUE)
```

```
## Warning in text.default(pos.xlabel[, 1], pos.xlabel[, 2], newcolnames, srt
## = tl.srt, : "outer" is not a graphical parameter
```

```
## Warning in text.default(pos.ylabel[, 1], pos.ylabel[, 2], newrownames, col
## = tl.col, : "outer" is not a graphical parameter
```




```
#From the correlation plot we see that for high values of PM25, PM10 and NO2 values are also high (all
head(sub_pm25,n = 10)
```

```
##    tmpd   dptp pm25tmean2 pm10tmean2  o3tmean2 no2tmean2
## 1  66.0 47.750       61.5       79.0 26.000000  47.05442
## 2  77.0 65.625       61.5       79.0 22.191498  32.66848
```

```
## 3  62.5 48.625       61.5       71.5 15.675084  48.28212
## 4  36.0 32.100       61.5       57.5  4.812500  41.52778
## 5  61.5 49.875       56.5      106.0 31.529762  55.41667
## 6  77.5 61.375       56.5       80.0 47.268914  51.17500
## 7  49.5 33.375       56.5      106.5 15.115609  40.27065
## 8   8.5  2.750       56.5       87.0  4.659420  42.88504
## 9  32.0 29.375       56.5       77.0  8.666667  49.01894
## 10 85.0 58.375       56.5      116.0 58.886425  46.27342
```

```
#Top 10 values of No2
sub_no2=arrange(mice_imputed_data, desc(no2tmean2))
head(sub_no2,n = 10)
```

```
##    tmpd   dptp pm25tmean2 pm10tmean2  o3tmean2 no2tmean2
## 1  82.5 60.625   38.90000  116.00000 55.086119  62.47998
## 2  62.5 45.750   49.47143   91.87500 11.041667  59.51562
## 3  27.0 15.400   46.20000   62.50000  8.775362  57.56439
## 4  50.0 28.500   36.90000   78.28571 15.401042  57.52542
## 5  62.5 42.875   19.80000   66.50000 18.053571  56.23007
## 6  70.5 62.250   56.50000   96.00000 31.119188  55.90522
## 7  83.5 65.750   56.50000  131.00000 39.518519  55.48958
## 8  61.5 49.875   56.50000  106.00000 31.529762  55.41667
## 9  30.5 32.000   49.47143   68.75000  8.650815  53.89485
## 10 67.0 46.875   19.40000   87.50000 21.648148  53.13333
```

```
#Top 10 values of PM10
sub_pm10=arrange(mice_imputed_data, desc(pm10tmean2))
head(sub_pm10,n = 10)
```

```
##    tmpd   dptp pm25tmean2 pm10tmean2 o3tmean2 no2tmean2
## 1  67.5 53.250   49.47143        365 48.80556  12.24955
## 2  54.0 34.250   49.47143        193 14.18342  29.03333
## 3  76.0 52.750   40.90000        148 50.90476  39.35218
## 4  76.5 54.000   49.47143        142 49.20168  44.59664
## 5  64.5 52.500   40.90000        141 32.91735  25.35833
## 6  79.0 51.125   38.90000        139 54.17857  34.67977
## 7  83.5 65.750   56.50000        131 39.51852  55.48958
## 8  47.0 33.125   40.90000        131 20.07955  31.56944
## 9  67.0 56.875   38.90000        129 17.25928  35.67500
## 10 64.0 48.500   40.90000        128 33.61286  34.27273
```

```
quantile(mice_imputed_data$pm10tmean2,probs=seq(0,1,by=.05))
```

```
##        0%        5%       10%       15%       20%       25%       30%
##   2.00000  12.00000  15.49000  17.50000  19.50000  21.50000  23.50000
##       35%       40%       45%       50%       55%       60%       65%
##  25.00000  27.00000  28.50000  30.14583  32.25000  34.50000  36.86339
##       70%       75%       80%       85%       90%       95%      100%
##  39.00000  42.00000  46.00000  50.68667  57.01429  67.50000 365.00000
```

```
quantile(mice_imputed_data$no2tmean2,probs=seq(0,1,by=.05))
```

```
##        0%        5%       10%       15%       20%       25%       30%
##  6.158333 12.983125 15.423314 17.055015 18.515833 19.653819 20.756647
##       35%       40%       45%       50%       55%       60%       65%
## 21.779858 22.709074 23.604167 24.555556 25.556453 26.566350 27.666196
##       70%       75%       80%       85%       90%       95%      100%
## 28.868795 30.139040 31.540336 33.260938 35.563750 39.547222 62.479984
```

```
##5

#Z score normalization
y = mice_imputed_data_date$pm25tmean2
mice_imputed_data_N_PM25 = mice_imputed_data_date
#Normalized Data
mice_imputed_data_N_PM25$pm25_N = (y-mean(y))/(sd(y))

#min-max normalization
x = mice_imputed_data_date$pm10tmean2

#Normalized Data
mice_imputed_data_N_PM25$pm10_N = (x-min(x))/(max(x)-min(x))
mice_imputed_data_N =mice_imputed_data_N_PM25

##6
#year = as.POSIXlt(chicago_new_N$date)$year + 1900

mice_imputed_data_N$date <- as.Date(mice_imputed_data_N$date)

mice_imputed_data_N[c("year")] = as.POSIXlt(mice_imputed_data_N$date)$year + 1900

##7
aggregate(cbind(dptp,tmpd,pm10tmean2,o3tmean2,no2tmean2)~year,
          data = mice_imputed_data_N,
          FUN=function(x){summary(x)})
```

```
##    year dptp.Min. dptp.1st Qu. dptp.Median dptp.Mean dptp.3rd Qu.
## 1  1987   -12.250       27.750      39.000    41.740       58.000
## 2  1988   -15.880       25.560      36.190    37.320       51.840
## 3  1989   -23.000       24.380      37.750    38.200       54.880
## 4  1990    -7.875       28.880      41.620    42.100       56.880
## 5  1991    -6.250       28.000      40.620    41.130       57.000
## 6  1992    -9.625       29.410      38.500    40.110       52.090
## 7  1993    -5.750       27.500      38.750    40.860       56.120
## 8  1994   -25.620       27.120      38.880    39.380       54.880
## 9  1995   -13.250       25.380      39.000    39.500       55.250
## 10 1996   -21.000       25.500      40.000    39.920       57.120
## 11 1997   -10.380       27.250      38.750    39.930       54.380
## 12 1998    -6.000       31.100      46.200    44.270       58.000
## 13 1999    -6.900       27.400      41.000    40.990       55.200
## 14 2000    -7.100       25.450      41.150    40.250       56.270
## 15 2001    -1.700       28.000      42.200    41.310       55.400
```

```
## 16 2002    -0.600       26.500      37.800    41.410       58.100
## 17 2003    -9.300       23.800      39.800    38.340       53.300
## 18 2004   -12.700       26.830      43.150    40.010       54.080
## 19 2005    -7.800       26.100      38.800    39.730       56.100
##    dptp.Max. tmpd.Min. tmpd.1st Qu. tmpd.Median tmpd.Mean tmpd.3rd Qu.
## 1     75.880      2.00        36.50       51.00     52.13        69.00
## 2     76.620     -6.50        34.50       49.25     49.94        68.25
## 3     77.000     -6.50        32.00       50.50     48.37        66.00
## 4     75.880      3.50        38.00       52.00     51.67        67.00
## 5     71.750      6.50        34.50       50.50     51.40        70.00
## 6     77.250      5.50        35.00       48.75     48.81        63.50
## 7     78.120      5.00        34.50       48.50     48.72        64.50
## 8     74.000    -16.00        36.50       52.00     50.08        66.50
## 9     76.380      0.50        33.00       49.50     50.21        68.50
## 10    78.250    -12.00        32.00       47.50     47.39        65.00
## 11    75.750     -3.00        34.00       48.50     48.71        65.00
## 12    72.900      7.00        39.00       55.00     53.78        70.00
## 13    77.100     -2.00        37.00       52.00     51.46        66.00
## 14    70.700     -1.00        34.00       53.00     50.04        67.00
## 15    75.100      5.00        36.00       53.00     51.04        66.00
## 16    74.900      6.00        35.00       48.00     51.17        70.00
## 17    71.600      4.00        34.00       50.00     49.23        66.00
## 18    72.800     -3.00        36.25       53.00     50.13        66.00
## 19    71.400      5.00        34.00       52.00     51.53        71.00
##    tmpd.Max. pm10tmean2.Min. pm10tmean2.1st Qu. pm10tmean2.Median
## 1      87.00           3.000              27.140             37.000
## 2      89.50           6.000              27.510             37.000
## 3      86.50           9.375              27.000             37.500
## 4      86.50           4.000              24.000             31.000
## 5      88.00           2.000              25.000             33.500
## 6      80.00           2.000              25.500             34.000
## 7      84.00           3.000              24.000             34.000
## 8      86.00           3.000              24.860             35.500
## 9      92.00           7.500              22.500             33.000
## 10     83.00           8.000              21.000             28.670
## 11     85.50           6.000              20.500             28.000
## 12     85.00           4.500              21.500             29.000
## 13     90.00           4.500              19.000             28.570
## 14     82.00           4.500              20.000             27.500
## 15     85.00           5.000              20.500             29.000
## 16     87.00           4.000              17.500             25.000
## 17     84.00           6.500              17.500             24.000
## 18     83.00           5.000              17.200             25.000
## 19     90.00           5.000              18.500             25.500
##    pm10tmean2.Mean pm10tmean2.3rd Qu. pm10tmean2.Max. o3tmean2.Min.
## 1           39.920             49.000         128.000        2.1250
## 2           42.390             52.880         148.000        2.3750
## 3           40.550             50.000         115.000        2.0000
## 4           35.250             44.000         104.600        2.1250
## 5           36.540             43.710         131.000        2.0000
## 6           38.600             45.380         141.000        2.0000
## 7           36.630             45.000         107.000        0.4271
## 8           39.500             47.880         131.000        0.1528
## 9           35.620             44.000          92.500        0.5169
```

```
## 10          31.670             38.000         128.000        1.1670
## 11          30.790             37.500         113.500        0.9565
## 12          30.930             38.000          91.000        0.5556
## 13          31.260             39.500          91.000        0.8542
## 14          30.480             37.920          85.000        1.0420
## 15          32.400             39.500          97.500        0.5000
## 16          29.340             38.000          95.000        2.0960
## 17          27.510             34.800          75.000        1.0980
## 18          27.860             35.500          82.000        2.1970
## 19          28.650             37.000          79.000        1.7710
##    o3tmean2.1st Qu. o3tmean2.Median o3tmean2.Mean o3tmean2.3rd Qu.
## 1           10.3500         18.8300       20.5000          28.9700
## 2           10.5900         20.3900       22.2500          30.5100
## 3           11.3900         19.2900       20.8100          28.1100
## 4           11.2400         18.9800       19.7300          26.2800
## 5           10.7100         18.3800       20.0100          27.8400
## 6            7.5940         15.1600       16.0600          22.4300
## 7            7.7500         14.9900       15.8000          23.5200
## 8            9.2660         16.0100       17.3200          24.8100
## 9            8.1220         16.7800       18.1400          25.6900
## 10           7.0720         15.8200       16.7500          23.6800
## 11          10.0200         18.1800       18.6000          26.6200
## 12           8.4250         20.1600       19.2700          28.3400
## 13          10.7600         20.5300       20.4700          28.8800
## 14          10.1800         18.1500       18.5300          25.8500
## 15          10.4500         18.7500       19.3500          26.8700
## 16          11.7900         19.8900       20.9900          28.1800
## 17          12.3100         19.5400       20.9800          29.5500
## 18          12.9700         20.6600       20.5400          27.4800
## 19          12.4100         23.1200       23.1900          32.4100
##    o3tmean2.Max. no2tmean2.Min. no2tmean2.1st Qu. no2tmean2.Median
## 1        62.9700          7.867            19.260           23.490
## 2        61.6800          7.733            19.630           24.520
## 3        59.7300          9.581            21.180           26.140
## 4        52.2300          6.726            17.540           22.600
## 5        63.1000          6.385            16.790           21.380
## 6        50.8300          9.962            20.970           24.790
## 7        44.3000         10.260            21.230           25.770
## 8        52.1800         10.940            23.000           28.480
## 9        66.5900          8.925            22.050           27.260
## 10       58.4000         10.460            22.070           26.390
## 11       56.5400          7.729            19.910           25.480
## 12       50.6600          8.800            20.040           24.590
## 13       57.4900          7.642            19.790           24.670
## 14       55.7600          6.158            17.960           23.460
## 15       51.8200          8.650            20.570           25.070
## 16       54.8800          7.802            18.640           22.740
## 17       56.1700          7.698            19.540           24.620
## 18       44.4800          7.552            18.800           23.390
## 19       58.8400          8.833            18.060           22.620
##    no2tmean2.Mean no2tmean2.3rd Qu. no2tmean2.Max.
## 1          25.290            30.750         57.530
## 2          25.340            30.210         62.480
## 3          27.360            32.270         59.520
```

```
## 4          23.060           27.720         46.190
## 5          21.920           26.260         55.490
## 6          25.860           29.770         49.780
## 7          26.130           30.230         48.900
## 8          29.100           34.580         53.890
## 9          27.950           33.100         53.130
## 10         26.980           31.760         52.690
## 11         25.760           31.010         53.010
## 12         24.740           29.350         47.820
## 13         24.930           29.680         49.550
## 14         24.070           29.970         50.810
## 15         25.390           29.630         48.720
## 16         23.670           27.920         47.760
## 17         25.240           29.890         57.560
## 18         23.440           27.650         44.980
## 19         23.170           27.340         48.600
```

```
##8


#We see that as PM25 and avg_o3 are not following any pattern

pm25.quint <- cut(mice_imputed_data_N$pm25tmean2,breaks=c(quantile(mice_imputed_data_N$pm25tmean2,
                                                                   probs = seq(0, 1, by = 0.10))),
                  labels=c("0-10","10-20","20-30","30-40","40-50","50-60","60-70","70-80","80-90","90-1
                  include.lowest=TRUE)

mice_imputed_data_final <- cbind(mice_imputed_data_N,pm25.quint)
grp_by <- group_by(mice_imputed_data_final,pm25.quint)
summarise(grp_by,avg_o3tmean2 = mean(o3tmean2))
```

```
## Source: local data frame [10 x 2]
##
##    pm25.quint avg_o3tmean2
##        (fctr)        (dbl)
## 1        0-10     21.19265
## 2       10-20     18.91545
## 3       20-30     18.68522
## 4       30-40     19.21478
## 5       40-50     18.34046
## 6       50-60     18.64571
## 7       60-70     18.88857
## 8       70-80     19.31650
## 9       80-90     19.41127
## 10     90-100     21.68362
```