# Pre-processing of Dataset

Noel C. Sieras

2022-12-16

## Preliminaries

This will prevent some errors in loading some of the chunks and laoding of the dataset.

## Loading of packages This package `tidyverse` will help the loading of the packeges needed for the pre=processing of data. The package `bestNormalize` will be used for normalizing the given dataset.

```
pacman::p_load(tidyverse)
pacman::p_load(bestNormalize)
```

## Loading of the radiomics dataset using the `readr` package The radiomics dataset is loaded and assigned to a variable name **RDat**.

```
library(readr)
RDat=read_csv("radiomics_completedata.csv", show_col_types = FALSE)
```

## Checking of null and missing values

```
sum(is.na(RDat))
```

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

Based from the result 0, the dataset RDat has *no* null and missing values.
## Normality Test Kolmogorov-Smirnov (`ks.test()`) test is used to check for normality of the dataset.

```
RD1=RDat%>%select_if(is.numeric)
RD1=RD1[,-c(1:2)]
RD2=apply(RD1,2,function(x){ks.test(x,"pnorm")})
```

## Unlist the dataset RD2 The unlist produce a vector which containd all the atomic components which occur in RD2 dataset

```
KS_list=unlist(lapply(RD2, function(x) x$p.value))
```

## Checking the number of variables that are not normally distributed Counting the number of variables that are not normally distributed.

```
sum(KS_list<0.05)
```

```
## [1] 428
```

From the result, there are 428 variables that are not normally distributed.
## Checking the number of variables that are normally distributed

```
sum(KS_list>0.05)
```

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

The result of 0 means that there is no nromally distributed variable.
## Checking the variable with the maximum value p-value in the list

```
which.max(KS_list)
```

```
## Kurtosis_hist.PET
##                 9
```

From the result, the variable Kurtosis_hist.PET has the maximum p.value.
## Normalization of the dataset and checking of normality The `orderNorm` is used for normalization. The Kolmogorov-Smirnov test is used for checking the normality of the dataset.

```
tempDFR=RDat[,c(3,5:length(names(RDat)))]
tempDFR=apply(tempDFR,2,orderNorm)
tempDFR=lapply(tempDFR, function(x) x$x.t)
tempDFR=tempDFR%>%as.data.frame()
testRD=apply(tempDFR,2,function(x){ks.test(x,"pnorm")})
testRD=unlist(lapply(testRD, function(x) x$p.value))
```

## Checking the number of variables which are normally distributed

```
sum(testRD>0.05)
```

```
## [1] 428
```

From the result, there are 428 variables which are normally normally distributed.
## Checking the number of variables of which are not normally distributed

```
sum(testRD<0.05)
```

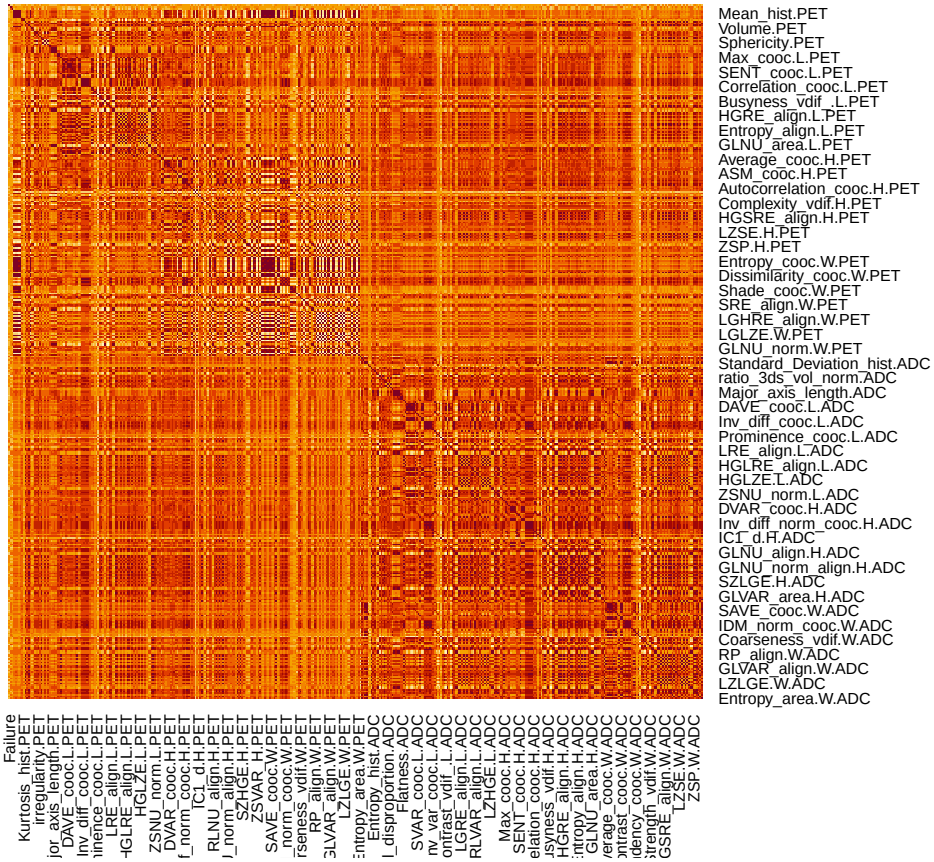
```
## [1] 0
```

From the result of the chunk, there is no more variable that is not normally distributed.
## Collecting all the variables into one dataset

```
RDat[,c(3,5:length(names(RDat)))]=tempDFR
```

## Checking for correlation

```
CorMatrix=cor(RDat[,-c(1,2)])
heatmap(CorMatrix,Rowv=NA,Colv=NA,scale="none",revC = T)
```




## Transforming a some variables as categorical

```
RDat$Institution=as.factor(RDat$Institution)
RDat$Failure.binary=as.factor(RDat$Failure.binary)
```

## Saving a normalize dataset as `normalRad`

```
write.csv(RDat, "D:/FilesWorkOn&Saved/PhDStat/@MSUIIT/SY20222023/01FirstSemester/STT225_StatisticalComput
```

## Splitting of dataset into a training data and testing data

```
splitter <- sample(1:nrow(RDat), round(nrow(RDat) * 0.8))
trainRDat <- RDat[splitter, ]
testRDat  <- RDat[-splitter, ]
```