Capstone Project CKME 136

Student: Walter Cios; Student ID 500791945

# Introduction

...

# Step 1: Data Cleaning and Preparation

### Load the package to read the raw data from Excel; load also ggplot2 for graphs; load dplyr for applying transformations on the dataframes; load package GGally to be used for plotting

library(readxl)  
library(ggplot2)  
library(dplyr)  
library(GGally)

### Loading the data

After examining the original excel file, I noticed that only the first sheet had data, so I will only read that from the file

raw\_data <- read\_excel("./Data/ENB2012\_data.xlsx", sheet = 1)

### Display structure of "raw\_data"

I reviewed that all 768 observations are loaded

str(raw\_data)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 768 obs. of 10 variables:  
## $ X1: num 0.98 0.98 0.98 0.98 0.9 0.9 0.9 0.9 0.86 0.86 ...  
## $ X2: num 514 514 514 514 564 ...  
## $ X3: num 294 294 294 294 318 ...  
## $ X4: num 110 110 110 110 122 ...  
## $ X5: num 7 7 7 7 7 7 7 7 7 7 ...  
## $ X6: num 2 3 4 5 2 3 4 5 2 3 ...  
## $ X7: num 0 0 0 0 0 0 0 0 0 0 ...  
## $ X8: num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Y1: num 15.6 15.6 15.6 15.6 20.8 ...  
## $ Y2: num 21.3 21.3 21.3 21.3 28.3 ...

### Check for NA in the whole dataset and found none

....

sum(sapply(raw\_data, is.na))

## [1] 0

As an alternative we could also have done an NA check per columns and as expected found none

sapply(raw\_data, function(x) { sum(is.na(x)) })

## X1 X2 X3 X4 X5 X6 X7 X8 Y1 Y2   
## 0 0 0 0 0 0 0 0 0 0

### Change column name to real variable names

....

data <- raw\_data  
names(data) <- c("Relative Compactness", "Surface Area", "Wall Area", "Roof Area", "Overall Height",  
 "Orientation", "Glazing Area", "Glazing Area Distribution", "Heating Load", "Cooling Load")

## Converting categorical variables to factors

As per the dataset description the variable "Orientation" has only four values N,W,S,E The variable "Glazing Area Distribution"" has also the above and in addition "0" and "1"

toCategorical <- Vectorize(function(column) {  
 switch(as.character(column),  
 "0" = "No Glazing Area",  
 "1" = "Uniform",  
 "2" = "North",  
 "3" = "East",  
 "4" = "South",  
 "5" = "West",  
 NA)  
})  
  
data$`Glazing Area Distribution` <- as.factor(toCategorical(data$`Glazing Area Distribution`))  
data$Orientation <- as.factor(toCategorical(data$Orientation))

## Review again structure, this time for "data"

summary(data)

## Relative Compactness Surface Area Wall Area Roof Area   
## Min. :0.6200 Min. :514.5 Min. :245.0 Min. :110.2   
## 1st Qu.:0.6825 1st Qu.:606.4 1st Qu.:294.0 1st Qu.:140.9   
## Median :0.7500 Median :673.8 Median :318.5 Median :183.8   
## Mean :0.7642 Mean :671.7 Mean :318.5 Mean :176.6   
## 3rd Qu.:0.8300 3rd Qu.:741.1 3rd Qu.:343.0 3rd Qu.:220.5   
## Max. :0.9800 Max. :808.5 Max. :416.5 Max. :220.5   
## Overall Height Orientation Glazing Area Glazing Area Distribution  
## Min. :3.50 East :192 Min. :0.0000 East :144   
## 1st Qu.:3.50 North:192 1st Qu.:0.1000 No Glazing Area: 48   
## Median :5.25 South:192 Median :0.2500 North :144   
## Mean :5.25 West :192 Mean :0.2344 South :144   
## 3rd Qu.:7.00 3rd Qu.:0.4000 Uniform :144   
## Max. :7.00 Max. :0.4000 West :144   
## Heating Load Cooling Load   
## Min. : 6.01 Min. :10.90   
## 1st Qu.:12.99 1st Qu.:15.62   
## Median :18.95 Median :22.08   
## Mean :22.31 Mean :24.59   
## 3rd Qu.:31.67 3rd Qu.:33.13   
## Max. :43.10 Max. :48.03

From the summmary we can see that Orientation and Glazing Area Distribution, are qualitative-categorical variables, the other variables are quantitaive-continuous

# Step 2: Exploratory Data Analysis

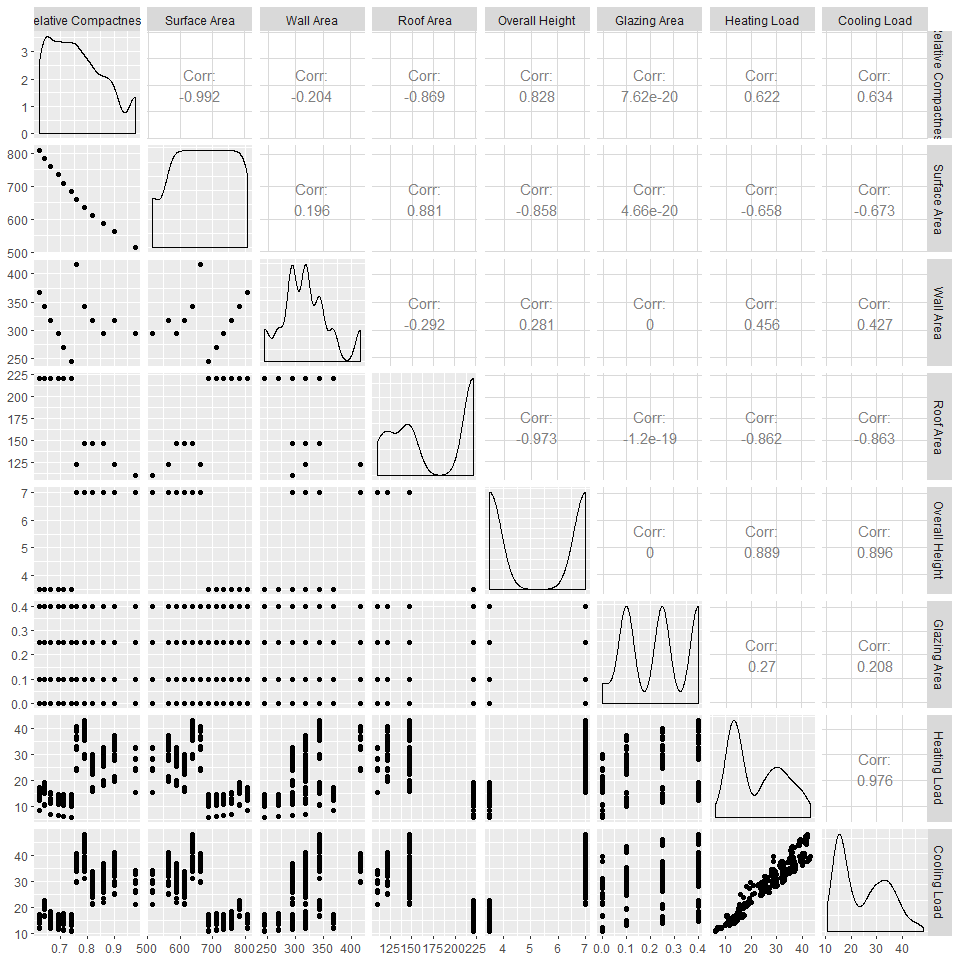
Split up categorical variables from the continuous ones, into a new dataframe "data\_numeric"

data\_numeric <- select(data, -Orientation, -`Glazing Area Distribution`)

## Paired Scatterplots

I looked for a better display of plot and found the package "GGally" that is being used for scatterplots between all pairs of variables

ggpairs(data\_numeric)



# Spep 2 to be further detailed in the future anlaysis process