Multivariate Analysis of House prices

### Introduction

**Objective and Data Set Description:**

The dataset has 79 explanatory variables and 1460 observations, describing (almost) every aspect of residential homes (dimensions, neighborhoods, sale prices etc.) in Ames, Iowa. The data set is multivariate where the final dimensions will be selected afterwards by reducing. Predicting the final price of each home is also possible with this data set using regression.

For details of variable the link to the data set is below: <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data>

**Motivation: To address following Business Questions:**

1. Dimensionality reduction using PCA and Factor analysis.
2. Clustering on the data set to find the clusters of different types houses. For improved price assessment and marketing based on different classes of houses identified.
3. Predicting the house prices in Ames, Iowa using regression model and principle component regression.

**Variable of our analysis and their descriptions (16 variables):**

1. LotArea :Lot size in square feet
2. MasVnrArea :Masonry veneer area in square feet
3. BsmtFinSF1 :Basement type 1 finished square feet
4. BsmtUnfSF :Unfinished square feet of basement area
5. TotalBsmtSF :Total square feet of basement area
6. X1stFlrSF :First Floor square feet
7. X2ndFlrSF :Second floor square feet
8. GrLivArea :Above grade (ground) living area square feet
9. BsmtFullBath :Basement full bathrooms
10. FullBath :Full bathrooms above grade
11. BedroomAbvGr :Number of bedrooms above basement level
12. KitchenAbvGr :Number of kitchens
13. TotRmsAbvGrd :Total rooms above grade (does not include bathrooms)
14. GarageArea :Size of garage in square feet
15. WoodDeckSF :Wood deck area in square feet
16. OpenPorchSF :Open porch area in square feet

### Data Preprocessing and Cleaning

We have narrowed our data set to 16 variables. For further dimension reduction and missing value analysis we are going to do correlation analysis and visualization as below:

**a) Import data set into R environment**

#data cleaning  
Housing<-read.csv("C:/Education/Multivariate Analysis/Project/Housing/Data/Full\_Housing\_New.csv")  
Housing<-Housing[,-c(1,18)] #Excluding ID and SalePrice  
str(Housing)

## 'data.frame': 2919 obs. of 16 variables:  
## $ LotArea : int 8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...  
## $ MasVnrArea : int 196 0 162 0 350 0 186 240 0 0 ...  
## $ BsmtFinSF1 : int 706 978 486 216 655 732 1369 859 0 851 ...  
## $ BsmtUnfSF : int 150 284 434 540 490 64 317 216 952 140 ...  
## $ TotalBsmtSF : int 856 1262 920 756 1145 796 1686 1107 952 991 ...  
## $ X1stFlrSF : int 856 1262 920 961 1145 796 1694 1107 1022 1077 ...  
## $ X2ndFlrSF : int 854 0 866 756 1053 566 0 983 752 0 ...  
## $ GrLivArea : int 1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...  
## $ BsmtFullBath: int 1 0 1 1 1 1 1 1 0 1 ...  
## $ FullBath : int 2 2 2 1 2 1 2 2 2 1 ...  
## $ BedroomAbvGr: int 3 3 3 3 4 1 3 3 2 2 ...  
## $ KitchenAbvGr: int 1 1 1 1 1 1 1 1 2 2 ...  
## $ TotRmsAbvGrd: int 8 6 6 7 9 5 7 7 8 5 ...  
## $ GarageArea : int 548 460 608 642 836 480 636 484 468 205 ...  
## $ WoodDeckSF : int 0 298 0 0 192 40 255 235 90 0 ...  
## $ OpenPorchSF : int 61 0 42 35 84 30 57 204 0 4 ...

We have imported the complete data set (training and test) and excluded the ID and SalePrice variables, for the purpose of further multivariate analysis.

**b) Dealing with Missing Values in Original data set.**

Let us observe our data set for missing values as follows:

sum(is.na(Housing)) #Count NA values across variables

## [1] 29

sapply(Housing, function(x) sum(is.na(x)))# number of nas

## LotArea MasVnrArea BsmtFinSF1 BsmtUnfSF TotalBsmtSF   
## 0 23 1 1 1   
## X1stFlrSF X2ndFlrSF GrLivArea BsmtFullBath FullBath   
## 0 0 0 2 0   
## BedroomAbvGr KitchenAbvGr TotRmsAbvGrd GarageArea WoodDeckSF   
## 0 0 0 1 0   
## OpenPorchSF   
## 0

We observe 29 NA values in our dataset. Considering these missing values to be completely random, we replace/impute these values with their respective column mean values.

imp<-apply(Housing,2,mean,na.rm= T)  
house<- Housing  
for(i in 1:ncol(Housing)){  
 house[is.na(house[,i]),i]<-imp[i]  
}  
sapply(house, function(x) sum(is.na(x)))#imputing nas with col means

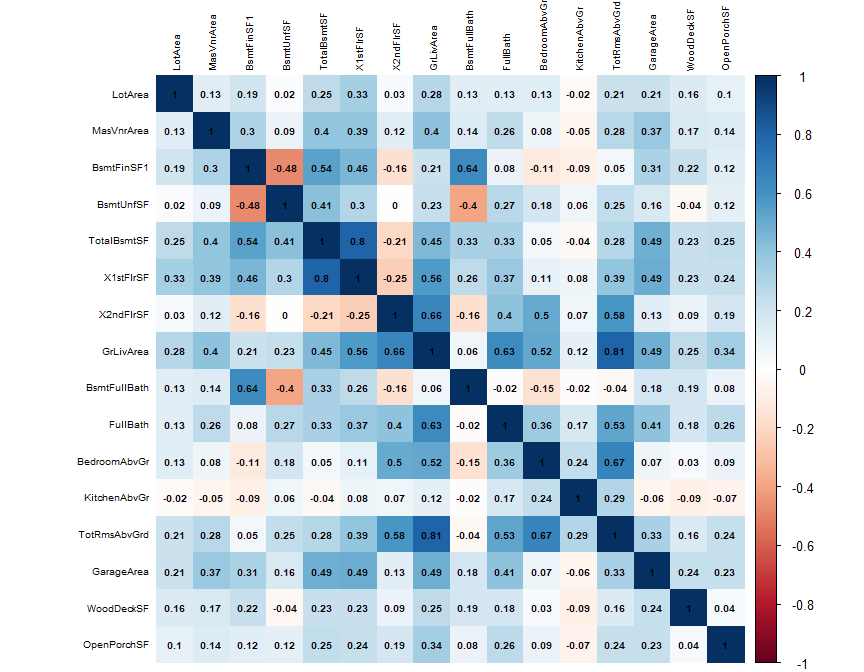
## LotArea MasVnrArea BsmtFinSF1 BsmtUnfSF TotalBsmtSF   
## 0 0 0 0 0   
## X1stFlrSF X2ndFlrSF GrLivArea BsmtFullBath FullBath   
## 0 0 0 0 0   
## BedroomAbvGr KitchenAbvGr TotRmsAbvGrd GarageArea WoodDeckSF   
## 0 0 0 0 0   
## OpenPorchSF   
## 0

#str(house)

**c) Correlation analysis**

Let us observe the correlations between the variables of our dataset, as we understand that there needs to be correlation between the variables for proceeding with the multivariate analysis.

library(corrplot)  
corrplot(cor(house), method="color", addCoef.col = "black",   
 tl.col="black", tl.cex=0.6, number.cex=0.6)

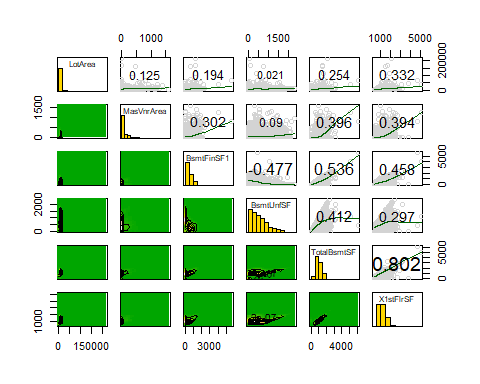


From the above corellelogram, we observe weak correlation of following variables with rest of the variables:

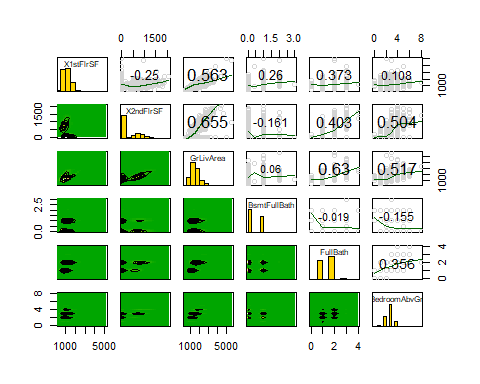
1. KitchenAbvGrd
2. WoodDeckSF
3. OpenPorchSF
4. LotArea

Let us confirm the same from further correlation visualizations.

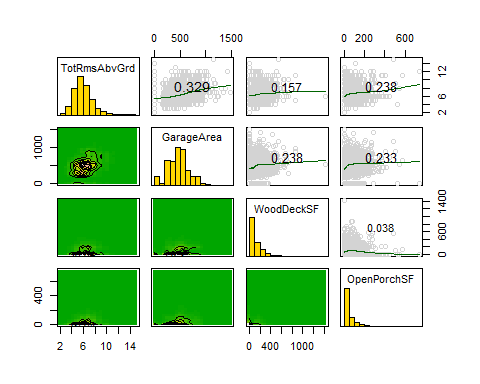
library(ResourceSelection)  
#str(house)  
#dim(house)  
#cor(house)#correltion matrix  
kdepairs(house[,1:6])



kdepairs(house[,6:11])



kdepairs(house[,13:16])##12 is the kitchen above the garage . most of the values 1 or 2 , so it acts as binary variable



#multivariate normality check

From the above visualizations, we confirm that the 3 variables above mentioned express weak correlation with the rest of the variables and hence we exclude them from further multivariate analysis.

#New dataset  
house.new<-house[,-c(1,12,15,16)]  
str(house.new)

## 'data.frame': 2919 obs. of 12 variables:  
## $ MasVnrArea : num 196 0 162 0 350 0 186 240 0 0 ...  
## $ BsmtFinSF1 : num 706 978 486 216 655 ...  
## $ BsmtUnfSF : num 150 284 434 540 490 64 317 216 952 140 ...  
## $ TotalBsmtSF : num 856 1262 920 756 1145 ...  
## $ X1stFlrSF : num 856 1262 920 961 1145 ...  
## $ X2ndFlrSF : num 854 0 866 756 1053 ...  
## $ GrLivArea : num 1710 1262 1786 1717 2198 ...  
## $ BsmtFullBath: num 1 0 1 1 1 1 1 1 0 1 ...  
## $ FullBath : num 2 2 2 1 2 1 2 2 2 1 ...  
## $ BedroomAbvGr: num 3 3 3 3 4 1 3 3 2 2 ...  
## $ TotRmsAbvGrd: num 8 6 6 7 9 5 7 7 8 5 ...  
## $ GarageArea : num 548 460 608 642 836 480 636 484 468 205 ...

**d) Outlier Analysis**

In the previous visualizations we observe outliers in the dataset. Here we attempt to identify and exclude the outliers using the Mahalanobis distances.

We perform our outlier analysis, considering only the training dataset. Also, we perform scaling to standardize the range of independent variables.

house.new <- house.new[1:1460,] #Train data  
house.scale<- scale(house.new)  
#head(house.scale)  
summary(house.scale)

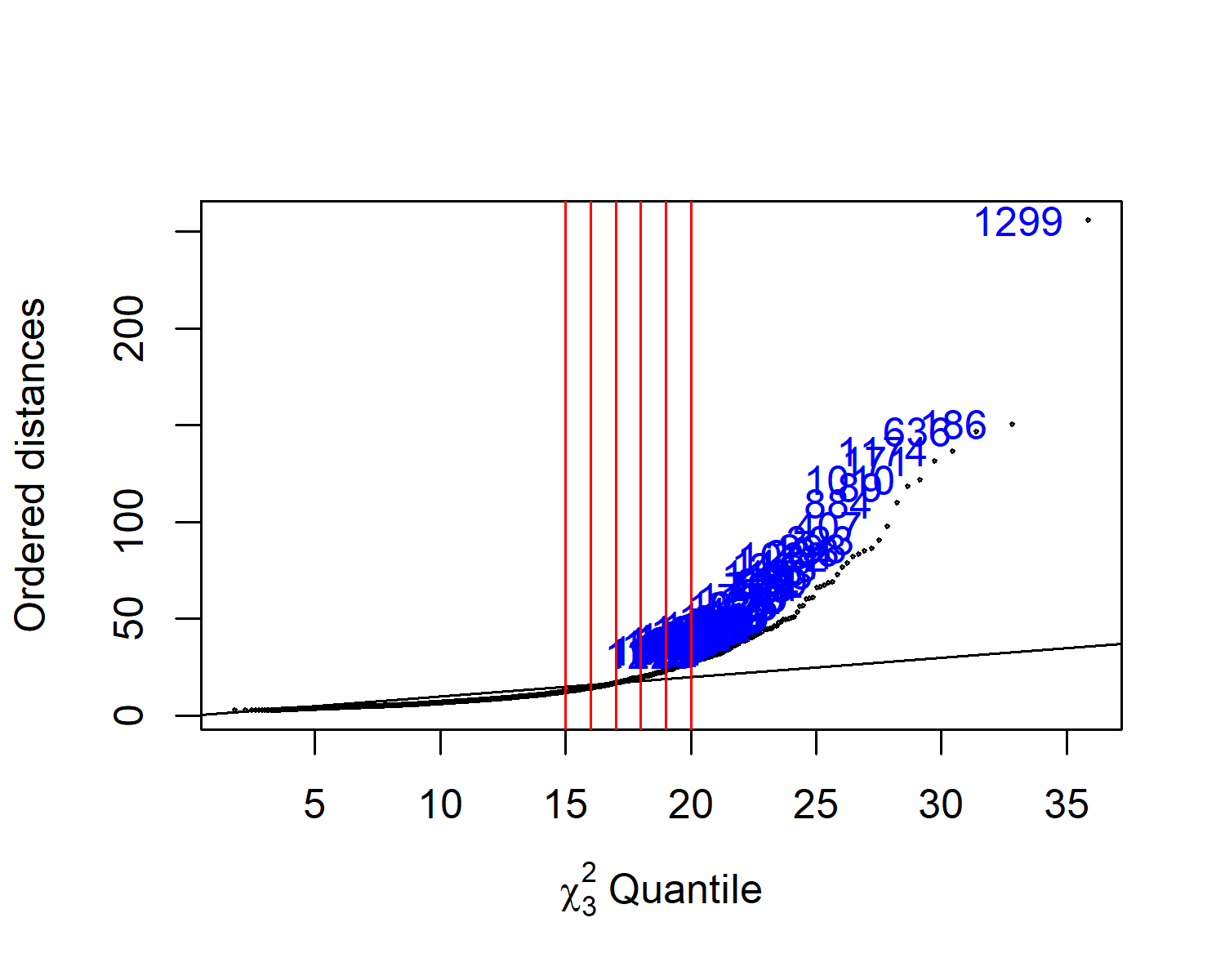
## MasVnrArea BsmtFinSF1 BsmtUnfSF TotalBsmtSF   
## Min. :-0.5742 Min. :-0.9727 Min. :-1.2837 Min. :-2.4103   
## 1st Qu.:-0.5742 1st Qu.:-0.9727 1st Qu.:-0.7791 1st Qu.:-0.5965   
## Median :-0.5742 Median :-0.1319 Median :-0.2031 Median :-0.1503   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.3355 3rd Qu.: 0.5889 3rd Qu.: 0.5449 3rd Qu.: 0.5489   
## Max. : 8.2867 Max. :11.4018 Max. : 4.0029 Max. :11.5170   
## X1stFlrSF X2ndFlrSF GrLivArea BsmtFullBath   
## Min. :-2.1434 Min. :-0.7949 Min. :-2.24835 Min. :-0.8197   
## 1st Qu.:-0.7259 1st Qu.:-0.7949 1st Qu.:-0.73450 1st Qu.:-0.8197   
## Median :-0.1956 Median :-0.7949 Median :-0.09794 Median :-0.8197   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.00000 Mean : 0.0000   
## 3rd Qu.: 0.5914 3rd Qu.: 0.8728 3rd Qu.: 0.49723 3rd Qu.: 1.1074   
## Max. : 9.1296 Max. : 3.9356 Max. : 7.85288 Max. : 4.9617   
## FullBath BedroomAbvGr TotRmsAbvGrd GarageArea   
## Min. :-2.8408 Min. :-3.5137 Min. :-2.7795 Min. :-2.21220   
## 1st Qu.:-1.0257 1st Qu.:-1.0621 1st Qu.:-0.9338 1st Qu.:-0.64769   
## Median : 0.7895 Median : 0.1637 Median :-0.3186 Median : 0.03283   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.00000   
## 3rd Qu.: 0.7895 3rd Qu.: 0.1637 3rd Qu.: 0.2967 3rd Qu.: 0.48184   
## Max. : 2.6046 Max. : 6.2928 Max. : 4.6033 Max. : 4.42001

#str(house.scale)

From the above summary we observe that the max values of most variables even after scaling are above value 3, i.e. we notice significant outliers in the dataset. Following we visualize the distances of observations using the Chi-plot.

**Chi-Plot**

cm <-colMeans(house.scale)  
S <-cov(house.scale)  
d<-mahalanobis(house.scale,(cm),S)  
plot(qc<-qchisq((1:nrow(house.scale) -1/2) /nrow(house.scale), df =ncol(house.scale)), sd<-sort(d),xlab =expression(paste(chi[3]^2, " Quantile")), ylab ="Ordered distances", cex = .2)  
oups <- which(rank(abs(qc - sd), ties = "random") > nrow(house.scale)-70)  
text(qc[oups], sd[oups] - 1.5, names(oups),pos = 2, col = "blue")  
abline(a =0, b =1)  
abline(v=c(15:20), col="red")



We observe the distance do not lie normal on the chi-plot and hence the observations are not multivariate normally distributed. We exclude the observations having distance greater than 16, as we observe a deviation of distances in the chi-plot from quantile point 16. Hence, we proceed further excluding observations having distances greater than 16.

library(plyr)  
library(dplyr)  
m\_dist <- mahalanobis(house.new, colMeans(house.new), cov(house.new))# getting m-dist  
house.new$MD <- round(m\_dist, 2)# rounding off and md col   
  
house.new$outlier <- "No" #adding column for outlier  
house.new$outlier[house.new$MD > 16] <- "Yes"  
house.out <- house.new %>% filter(house.new$outlier=="No") # filter out outliers  
dim(house.out)#dimension

## [1] 1224 14

house.reqd <- house.out[,1:12]#final data set for further analysis

We utilize the knowledge gained from this course to implement the PCA, Factor analysis for dimension reductions.

### Principle Component Analysis

In this section, we perform the PCA to reduce the number of variables in the data set while accounting for as much original variation in the data set as possible.

# principal component analysis  
house.pca<-princomp(house.reqd, cor = T)  
summary(house.pca, loadings = T)

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5  
## Standard deviation 2.0641119 1.6855247 1.3693981 0.89410597 0.84324487  
## Proportion of Variance 0.3550465 0.2367495 0.1562709 0.06661879 0.05925516  
## Cumulative Proportion 0.3550465 0.5917960 0.7480669 0.81468568 0.87394084  
## Comp.6 Comp.7 Comp.8 Comp.9  
## Standard deviation 0.68624990 0.63458412 0.56462230 0.42238153  
## Proportion of Variance 0.03924491 0.03355808 0.02656653 0.01486718  
## Cumulative Proportion 0.91318575 0.94674383 0.97331036 0.98817754  
## Comp.10 Comp.11 Comp.12  
## Standard deviation 0.35627176 0.121771691 1.056499e-02  
## Proportion of Variance 0.01057746 0.001235695 9.301588e-06  
## Cumulative Proportion 0.99875500 0.999990698 1.000000e+00  
##   
## Loadings:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8  
## MasVnrArea -0.246 0.132 0.711 0.616 -0.165  
## BsmtFinSF1 0.456 0.382 0.173 0.320 0.191  
## BsmtUnfSF -0.175 -0.159 -0.622 -0.352 -0.104  
## TotalBsmtSF -0.311 0.355 -0.274 -0.170   
## X1stFlrSF -0.323 0.333 -0.255 -0.229 0.114 0.112 0.178  
## X2ndFlrSF -0.206 -0.420 0.343 0.154 -0.130 -0.222 0.245  
## GrLivArea -0.439 -0.146 0.131 0.118 -0.167 0.366  
## BsmtFullBath 0.402 0.366 -0.139 -0.108 -0.116 -0.711 -0.383  
## FullBath -0.357 0.184 -0.517 0.522 0.189 -0.493  
## BedroomAbvGr -0.272 -0.272 0.181 -0.444 0.358 -0.245 0.306 -0.519  
## TotRmsAbvGrd -0.390 -0.221 0.140 -0.219 0.136 -0.110 0.185  
## GarageArea -0.328 0.158 0.282 -0.399 -0.762 0.211   
## Comp.9 Comp.10 Comp.11 Comp.12  
## MasVnrArea   
## BsmtFinSF1 0.166 -0.290 -0.589   
## BsmtUnfSF 0.167 -0.197 -0.592   
## TotalBsmtSF 0.374 -0.463 0.548   
## X1stFlrSF -0.186 0.582 0.481   
## X2ndFlrSF 0.379 0.603   
## GrLivArea 0.219 0.372 -0.637   
## BsmtFullBath   
## FullBath -0.115   
## BedroomAbvGr 0.247   
## TotRmsAbvGrd -0.707 -0.408   
## GarageArea

We observe that the first 3 principle components captures 75 percent of variation in the dataset. Also, the first 3 PCs have standard deviaiton of greater than 1. Hence we consider the first 3 PCs.

We can interpret the three principle components based on the direction and magnitude of loadings as follows:

**First PC**

house.pca$loadings[,1]

## MasVnrArea BsmtFinSF1 BsmtUnfSF TotalBsmtSF X1stFlrSF   
## -0.24632739 -0.09883743 -0.17534231 -0.31104675 -0.32300770   
## X2ndFlrSF GrLivArea BsmtFullBath FullBath BedroomAbvGr   
## -0.20624749 -0.43853878 -0.04329910 -0.35675277 -0.27218333   
## TotRmsAbvGrd GarageArea   
## -0.38978554 -0.32818719

We observe that all the variables are in the same direction for PC1 and has more weightage for TotalBsmtSF, X1stFlrSF, GrLivArea, FullBath, TotRmsAbvGrd and GarageArea. Hence we can consider the PC1 to represent overall characteristics of the house.

PC1 - “Overall Feature”

**Second PC**

house.pca$loadings[,2]

## MasVnrArea BsmtFinSF1 BsmtUnfSF TotalBsmtSF X1stFlrSF   
## 0.1318088 0.4558169 -0.1592894 0.3552447 0.3330759   
## X2ndFlrSF GrLivArea BsmtFullBath FullBath BedroomAbvGr   
## -0.4200702 -0.1458787 0.4024352 -0.0737999 -0.2715232   
## TotRmsAbvGrd GarageArea   
## -0.2205623 0.1575859

We observe that PC2 gives more weightage to X2ndFlrSF in the same direction of PC1 and in opposite direction to BsmtFinSF1, TotalBsmtSF, X1stFlrSF and BsmtFullBath. Hence, we can consider this component to be describing the Second Floor area in contrast to the Basement and First Floor areas.

PC2 - “Second Floor Feature”

**Third PC**

house.pca$loadings[,3]

## MasVnrArea BsmtFinSF1 BsmtUnfSF TotalBsmtSF X1stFlrSF   
## 0.06240068 0.38151464 -0.62188303 -0.27435805 -0.25544703   
## X2ndFlrSF GrLivArea BsmtFullBath FullBath BedroomAbvGr   
## 0.34291252 0.13082062 0.36572903 -0.03603942 0.18144872   
## TotRmsAbvGrd GarageArea   
## 0.13954785 -0.03364352

We observe that PC3 gives more weightage to BsmtUndSF in the same direction of PC1 and in opposite direction to BsmtFinSF1, X2stFlrSF and BsmtFullBath. Hence, we can consider this component to be describing the Basement unfinished area in contrast to the finished basement area and Second Floor areas.

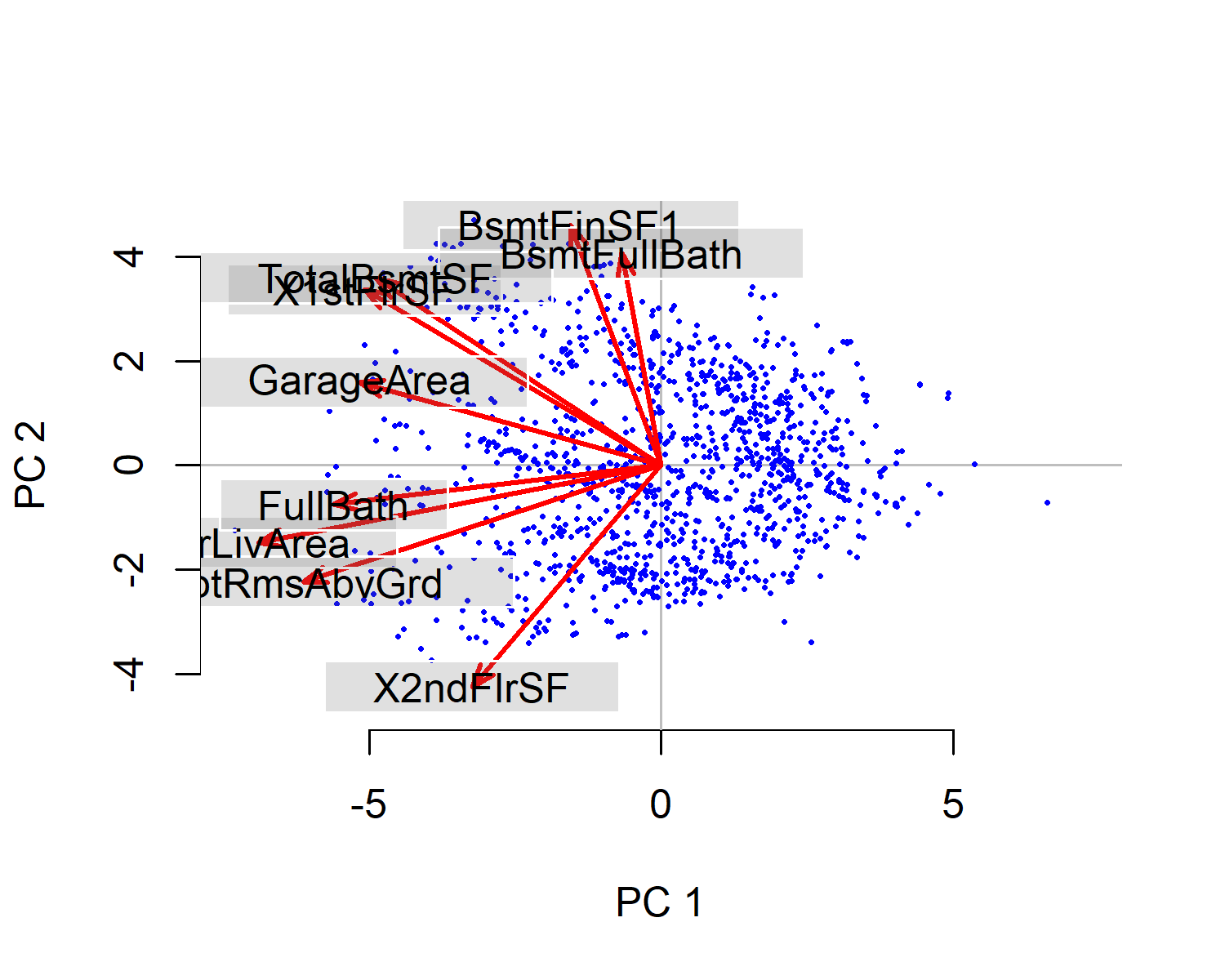
PC3 - “Unfinished Basement area”

**Bi-plot**

Here we attempt to plot the bi-plot which represents the variables and observations on to a single plot of PC components.

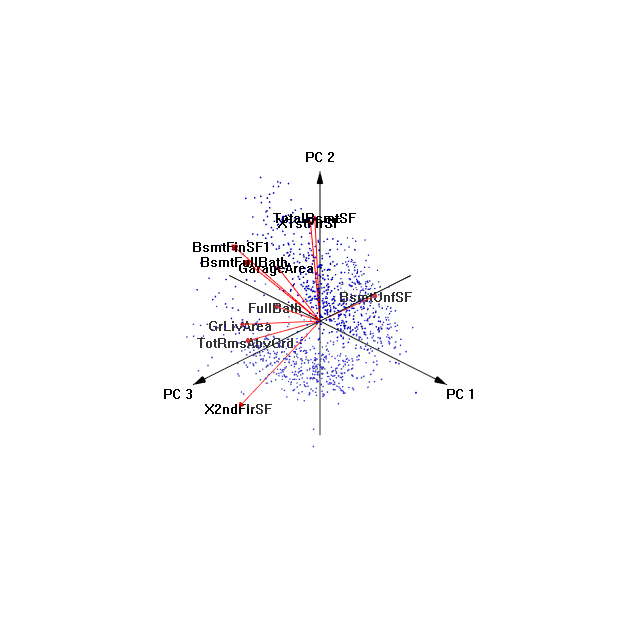
*(We make use of the pca3d package, which essentially is a shortcut to RGL graphics library and get the pca visualization done.)*

#2d representation  
library(pca3d)  
pca2d(house.pca, biplot=T, col="blue", radius = 0.3, title = "2D bi-plot")



As we have considered 3 PCs, we attempt for a 3 dimensional plot and interpret accordingly. (We here do not label the observartions as it would turn complete mess)

pca3d(house.pca, biplot = T, radius= 0.3, col="blue", axes.color = "black")  
snapshotPCA3d(file="3d.png")



From the above bi-plot we observe:

1. BsmtFinSF1 and BsmtFullBath are closely correlated and hence have similar profiles.
2. TotalBsmtSF and X1stFlrSF are closely and hence have similar profiles.
3. FullBath, GrLivArea and TotRmsAbvGrd are correlated to each other in the same direction.
4. We observe the BsmtUnfSF is oppositely correlated to BsmtFinSF, which was captured by the PC3 and we conclude that they have opposite profiles which is reasonable to understand as they represent unfinished and finished areas of the house.
5. Further we observe nearly uncorrelated group of variables which are at 90 degrees to each other.

*Note: We here are not interpreting anything about the observations and conclude only saying that points close to each other have similar scores on the PCs.*

### Exploratory Factor Analysis

We perform the Exploratory factor analysis, as a dimension reduction technique and observe the latent variables of the housing dataset.

# Scalling the data  
house.reqd.scale<-scale(house.reqd)  
# Factor analysis  
(house.fa <- factanal(house.scale, factors = 3))

##   
## Call:  
## factanal(x = house.scale, factors = 3)  
##   
## Uniquenesses:  
## MasVnrArea BsmtFinSF1 BsmtUnfSF TotalBsmtSF X1stFlrSF   
## 0.804 0.005 0.385 0.294 0.005   
## X2ndFlrSF GrLivArea BsmtFullBath FullBath BedroomAbvGr   
## 0.005 0.005 0.565 0.587 0.662   
## TotRmsAbvGrd GarageArea   
## 0.300 0.693   
##   
## Loadings:  
## Factor1 Factor2 Factor3  
## MasVnrArea 0.285 0.308 0.141   
## BsmtFinSF1 0.447 0.892   
## BsmtUnfSF 0.118 0.311 -0.710   
## TotalBsmtSF 0.817 0.173   
## X1stFlrSF 0.126 0.990   
## X2ndFlrSF 0.943 -0.324   
## GrLivArea 0.886 0.459   
## BsmtFullBath 0.257 0.600   
## FullBath 0.553 0.314   
## BedroomAbvGr 0.558 -0.150   
## TotRmsAbvGrd 0.766 0.318 -0.108   
## GarageArea 0.298 0.455 0.105   
##   
## Factor1 Factor2 Factor3  
## SS loadings 3.097 2.830 1.763  
## Proportion Var 0.258 0.236 0.147  
## Cumulative Var 0.258 0.494 0.641  
##   
## Test of the hypothesis that 3 factors are sufficient.  
## The chi square statistic is 2446.04 on 33 degrees of freedom.  
## The p-value is 0

print(house.fa$loadings, cut= 0.5)

##   
## Loadings:  
## Factor1 Factor2 Factor3  
## MasVnrArea   
## BsmtFinSF1 0.892   
## BsmtUnfSF -0.710   
## TotalBsmtSF 0.817   
## X1stFlrSF 0.990   
## X2ndFlrSF 0.943   
## GrLivArea 0.886   
## BsmtFullBath 0.600   
## FullBath 0.553   
## BedroomAbvGr 0.558   
## TotRmsAbvGrd 0.766   
## GarageArea   
##   
## Factor1 Factor2 Factor3  
## SS loadings 3.097 2.830 1.763  
## Proportion Var 0.258 0.236 0.147  
## Cumulative Var 0.258 0.494 0.641

We observe from the cumulative variance of factors, we observe that the 3 factors captures 64 percent variance in the dataset.

Further we observe that 3 factors were sufficient to explain the variability in the dataset. As we observe the factor loadings, we make notive of the following:

1. Factor1 gives more weightage to the X2ndFlrSF, GrLivArea and TotRmsAbvGrd and may be considered to represent the latent variable “Above Grade rooms and 2nd floor area”.
2. Factor2 gives more weightage to the TotalBsmtSF and X1stFlrSF and may be considered to represent the latent variable “Basement and 1st floor area”.
3. Factor3 gives more weightage to the BsmtFinSF1 and BsmtFullBath in direction of PC1 and BsmtUnfSF in opposite direction and hence may be considered to represent the latent variable “Impact of finished area and Basement full bathrooms”.

### Confirmatory factor analysis

We perform the confirmatory analysis to confirm the similarity between restricted covariance matrix (obtained through factor analysis) and non-restricted covariance matrix (obtained from raw data).

#chouse<-(house.new[,-c(1,3,9,10,12)])  
chouse<-house.reqd  
library(sem)  
real\_model <- specifyModel(file = "realestate\_model1.txt")  
real\_model

## Path Parameter StartValue  
## 1 Factor1 -> X2ndFlrSF lambda1   
## 2 Factor1 -> GrLivArea lambda2   
## 3 Factor1 -> TotRmsAbvGrd lambda3   
## 4 Factor2 -> TotalBsmtSF lambda4   
## 5 Factor2 -> X1stFlrSF lambda5   
## 6 Factor1 <-> Factor2 corr   
## 7 X2ndFlrSF <-> X2ndFlrSF theta1   
## 8 GrLivArea <-> GrLivArea theta2   
## 9 TotRmsAbvGrd <-> TotRmsAbvGrd theta3   
## 10 TotalBsmtSF <-> TotalBsmtSF theta4   
## 11 X1stFlrSF <-> X1stFlrSF theta5   
## 12 Factor1 <-> Factor1 <fixed> 1   
## 13 Factor2 <-> Factor2 <fixed> 1

opt <- options(fit.indices = c("GFI", "AGFI", "SRMR"))  
real\_sem <- sem(real\_model, cor(chouse), nrow(chouse)) #cor or cov same result.  
real\_sem

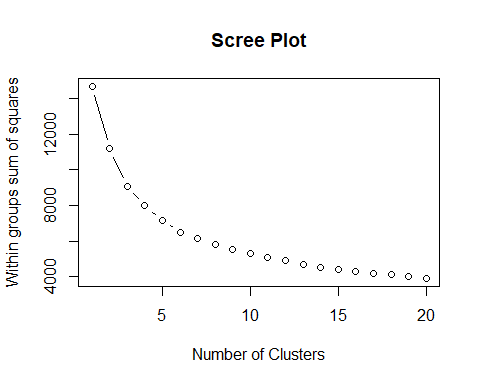
##   
## Model Chisquare = -2013151107 Df = 4   
##   
## lambda1 lambda2 lambda3 lambda4 lambda5   
## 0.144174251 3.569625104 -0.020333439 -0.266278082 -0.183941670   
## corr theta1 theta2 theta3 theta4   
## -0.250321413 1.127602682 -11.925361089 -0.011854047 0.163091876   
## theta5   
## -0.006216737   
##   
## Iterations = 670

#summary(real\_sem)  
#MasVnrArea, BsmtFinSF1, BsmtUnfSF, BsmtFullBath, FullBath, BedroomAbvGr, GarageArea  
#MasVnrArea, BsmtUnfSF, FullBath, BedroomAbvGr, GarageArea

We attempted confirmatory factor analysis using both 2 and 3 factors. However, in both the cases the result is “coefficient covariances cannot be computed”. Here, R is unable to calculate the factor loadings to show the summary. Also we found lambda values more than one and theta values negative. So, it a bad model too. Hence, we can conclude that it is not possible to get the latent variables with every confirmatory factor analysis.

### Clustering Analysis: K-means

# For K-means clustering scalling is important as it calcualtes the distance from centeroids so we will use scalled data  
  
#Within groups sum of squares  
plot.wgss = function(mydata, maxc) {  
wss = numeric(maxc)  
for (i in 1:maxc) wss[i] = kmeans(mydata,centers=i, nstart = 20)$tot.withinss   
plot(1:maxc, wss, type="b", xlab="Number of Clusters",  
ylab="Within groups sum of squares", main="Scree Plot") }  
  
#Scree plot to decide the number of clusters  
plot.wgss(house.reqd.scale, 20)

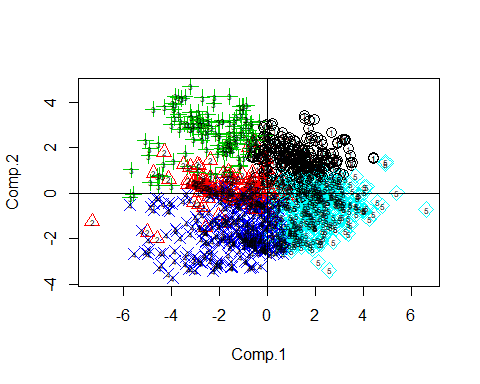


*The scree plot analysis shows we should pick 5 clusters as per elbow test.*

#building k-means clusters  
km2 <- kmeans(house.reqd.scale, 5)  
  
  
# NOTE: We cannot plot Multivarite data on 2 axis so better to plot clusters on Principle components, using non scaled data set and cor=true.  
pca <- princomp(house.reqd,cor=T)  
pca$loadings[,1:3] # how you name pc1, pc2, and pc3?

## Comp.1 Comp.2 Comp.3  
## MasVnrArea -0.24632739 0.1318088 0.06240068  
## BsmtFinSF1 -0.09883743 0.4558169 0.38151464  
## BsmtUnfSF -0.17534231 -0.1592894 -0.62188303  
## TotalBsmtSF -0.31104675 0.3552447 -0.27435805  
## X1stFlrSF -0.32300770 0.3330759 -0.25544703  
## X2ndFlrSF -0.20624749 -0.4200702 0.34291252  
## GrLivArea -0.43853878 -0.1458787 0.13082062  
## BsmtFullBath -0.04329910 0.4024352 0.36572903  
## FullBath -0.35675277 -0.0737999 -0.03603942  
## BedroomAbvGr -0.27218333 -0.2715232 0.18144872  
## TotRmsAbvGrd -0.38978554 -0.2205623 0.13954785  
## GarageArea -0.32818719 0.1575859 -0.03364352

#plotting the clusters against the PC1 and PC2  
plot(pca$scores[, c(1:2)], pch = km2$cluster, col=km2$cluster, cex=1.5)  
abline(h=0,v=0)  
text(pca$scores[, c(1:2)], labels = km2$cluster ,cex=0.5)

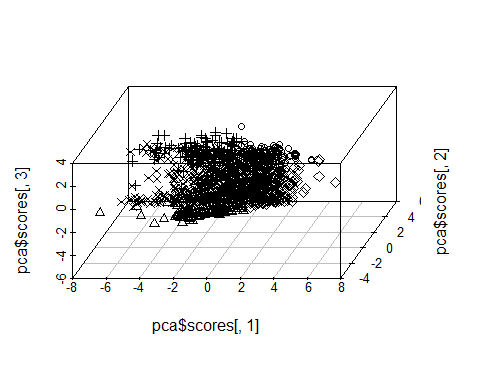


The 2D plot of PC1 scores and PC2 scores for our five clusters shows following:

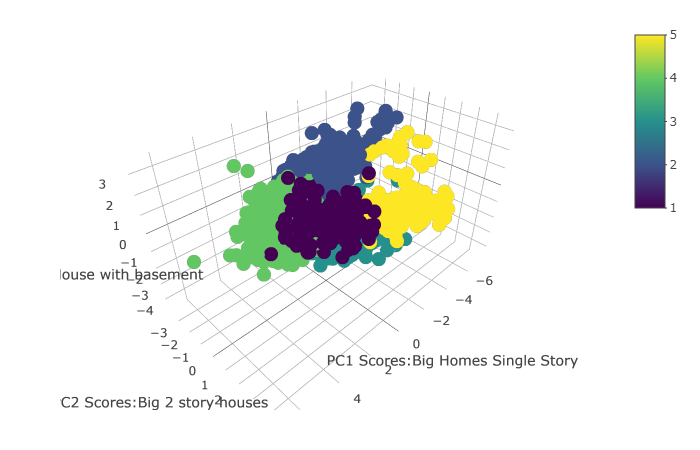
* Cluster 1:(black): Average houses having average surface area and basement features.
* Cluster 2(Red): Shows more weight towards positives of PC2, so they are the houses having big basements finished surface area and baement full baths.
* Cluster 3(Green): Shows big houses with high surface areas of both 1st and 2nd floors.
* Cluster 4(D.Blue): Houses having big 2nd floors areas.
* Cluster 5(L.Blue): Houses having big 1st floor,basement and garage areas.

\*\* 3D Representation plot of 3 Principal Components \*\*

#3D plot to show all clusters mapped on 3 Principal components we created in the earlier section  
library("scatterplot3d")  
scatterplot3d(x=pca$scores[, 1],y=pca$scores[, 2],z=pca$scores[, 3], pch = km2$cluster, angle=60)



# A better 3D Representation using 3D plot.  
#install.packages("plotly")  
library(plotly)  
scoresDF<-as.data.frame(pca$scores) #hoverlabel = km2$cluster,  
plot\_ly(scoresDF, x=~Comp.1, y=~Comp.2, z=~Comp.3, text = km2$cluster, color=km2$cluster ) %>% #(color range if required) colors = c('#BF382A', '#0C4B8E')   
 add\_markers() %>%  
 layout(scene = list(xaxis = list(title = 'PC1 Scores:Big Homes Single Story'),  
 yaxis = list(title = 'PC2 Scores:Big 2 story houses'),  
 zaxis = list(title = 'PC3 Scores:House with basement')))



The second 3D plot is an interactive plot, best visible when seen as html. Please check the attachment (html file) for this visualization). When hovering over an observation point on the 3d plot we can see the cluster it belongs to and what are the PC scores in three dimentions.The 5 clusters are color encoded as per legend on the plot. Further analysis of centeroids is explained in the next sections.

#Centers of the clusters  
km2$centers

## MasVnrArea BsmtFinSF1 BsmtUnfSF TotalBsmtSF X1stFlrSF X2ndFlrSF  
## 1 -0.20830058 0.6461489 -0.72626585 -0.01696165 -0.05500583 -0.7278313  
## 2 0.18721480 -0.8670502 1.91440915 1.20115179 1.18790599 -0.7145118  
## 3 1.01301706 1.6480792 -0.26739414 1.56728943 1.57484276 -0.4996277  
## 4 0.08243948 -0.3130742 -0.01984283 -0.41768034 -0.38850755 1.2759631  
## 5 -0.44455857 -0.5470806 -0.08226440 -0.73766557 -0.73559589 -0.2642605  
## GrLivArea BsmtFullBath FullBath BedroomAbvGr TotRmsAbvGrd GarageArea  
## 1 -0.7304251 1.0083194 -0.7492175 -0.44730047 -0.6525119 -0.09904256  
## 2 0.2212630 -0.6783678 0.7186261 -0.05625971 0.1474209 0.62165997  
## 3 0.7166975 1.0545109 0.8047877 -0.09080379 0.3839153 0.90717882  
## 4 0.9122445 -0.2456174 0.7407799 0.80303540 0.9049115 0.22354653  
## 5 -0.8037111 -0.7421556 -0.8536456 -0.44372898 -0.6818884 -0.83105170

**Cluster Centroid Analysis**

Based on the centers above we reach to the following conclusions:

* Cluster 1: Houses having high, **basement** finished surface area.
* Cluster 2: Houses having high, **second flour + big house** surface area.
* Cluster 3: Houses having average, **basement surface areas garage and full baths**.
* Cluster 4: Houses having small areas in all variable repreresntative of **small houses**.
* Cluster 5: **Big houses have high quality** and higher percentage of finished area.

**Total houses in each cluster:**

table(km2$cluster) # number of observations in a cluster

##   
## 1 2 3 4 5   
## 268 139 142 353 322

\*\* Houses in a particular Cluster \*\* To check the details about the houses in each cluster we can find the cluster data points.

#One can check and trace back the house IDs which are in cluster 1. This may be helpfull to further study the cluster with attributes which were not accounted for but were available exernally i.e. in master data.  
h <- subset(house.reqd.scale, km2$cluster ==1)  
head(h)

## MasVnrArea BsmtFinSF1 BsmtUnfSF TotalBsmtSF X1stFlrSF X2ndFlrSF  
## [1,] -0.6436690 1.3333996 -0.7059280 0.65658120 0.46869059 -0.8114359  
## [2,] -0.6436690 0.7300189 -1.2388697 -0.67597198 -0.95597175 0.5719837  
## [3,] -0.6436690 1.0218982 -1.0547626 -0.11835853 -0.09689424 -0.8114359  
## [4,] -0.6436690 1.1568003 -1.0692973 0.02175973 -0.21001121 -0.8114359  
## [5,] -0.6436690 0.7422827 -0.9699764 -0.34426346 -0.60133477 -0.8114359  
## [6,] 0.9454894 0.7324716 -0.1342270 0.63084519 0.44117565 -0.8114359  
## GrLivArea BsmtFullBath FullBath BedroomAbvGr TotRmsAbvGrd  
## [1,] -0.4137501 -0.8237298 0.9008189 0.2470505 -0.2244694  
## [2,] -0.1827576 1.1968018 -1.0472417 -2.6605443 -0.9325892  
## [3,] -0.8410860 1.1968018 -1.0472417 -1.2067469 -0.9325892  
## [4,] -0.9265532 1.1968018 -1.0472417 0.2470505 -0.9325892  
## [5,] -1.2222235 1.1968018 -1.0472417 -1.2067469 -1.6407090  
## [6,] -0.4345394 1.1968018 -1.0472417 -1.2067469 -0.9325892  
## GarageArea  
## [1,] -0.01090519  
## [2,] 0.09333495  
## [3,] -1.33996695  
## [4,] -0.40701771  
## [5,] -0.57380193  
## [6,] -0.57380193

### Predicitve Anlaysis: Regression

Initially we consider all the variables of our considered dataset to perform linear regression to predict SalePrice of houses. Following we reconstruct our required dataset.

#data cleaning  
Housing<-read.csv("C:/Education/Multivariate Analysis/Project/Housing/Data/Full\_Housing\_New.csv")  
str(Housing)

## 'data.frame': 2919 obs. of 18 variables:  
## $ Id : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ LotArea : int 8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...  
## $ MasVnrArea : int 196 0 162 0 350 0 186 240 0 0 ...  
## $ BsmtFinSF1 : int 706 978 486 216 655 732 1369 859 0 851 ...  
## $ BsmtUnfSF : int 150 284 434 540 490 64 317 216 952 140 ...  
## $ TotalBsmtSF : int 856 1262 920 756 1145 796 1686 1107 952 991 ...  
## $ X1stFlrSF : int 856 1262 920 961 1145 796 1694 1107 1022 1077 ...  
## $ X2ndFlrSF : int 854 0 866 756 1053 566 0 983 752 0 ...  
## $ GrLivArea : int 1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...  
## $ BsmtFullBath: int 1 0 1 1 1 1 1 1 0 1 ...  
## $ FullBath : int 2 2 2 1 2 1 2 2 2 1 ...  
## $ BedroomAbvGr: int 3 3 3 3 4 1 3 3 2 2 ...  
## $ KitchenAbvGr: int 1 1 1 1 1 1 1 1 2 2 ...  
## $ TotRmsAbvGrd: int 8 6 6 7 9 5 7 7 8 5 ...  
## $ GarageArea : int 548 460 608 642 836 480 636 484 468 205 ...  
## $ WoodDeckSF : int 0 298 0 0 192 40 255 235 90 0 ...  
## $ OpenPorchSF : int 61 0 42 35 84 30 57 204 0 4 ...  
## $ SalePrice : int 208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...

Housing<-Housing[,-c(1,2,13,16,17)]  
#head(Housing)  
#str(Housing)  
#sum(is.na(Housing))  
#sapply(Housing, function(x) sum(is.na(x)))# number of nas  
house.train<-Housing[1:1460,]  
str(house.train)

## 'data.frame': 1460 obs. of 13 variables:  
## $ MasVnrArea : int 196 0 162 0 350 0 186 240 0 0 ...  
## $ BsmtFinSF1 : int 706 978 486 216 655 732 1369 859 0 851 ...  
## $ BsmtUnfSF : int 150 284 434 540 490 64 317 216 952 140 ...  
## $ TotalBsmtSF : int 856 1262 920 756 1145 796 1686 1107 952 991 ...  
## $ X1stFlrSF : int 856 1262 920 961 1145 796 1694 1107 1022 1077 ...  
## $ X2ndFlrSF : int 854 0 866 756 1053 566 0 983 752 0 ...  
## $ GrLivArea : int 1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...  
## $ BsmtFullBath: int 1 0 1 1 1 1 1 1 0 1 ...  
## $ FullBath : int 2 2 2 1 2 1 2 2 2 1 ...  
## $ BedroomAbvGr: int 3 3 3 3 4 1 3 3 2 2 ...  
## $ TotRmsAbvGrd: int 8 6 6 7 9 5 7 7 8 5 ...  
## $ GarageArea : int 548 460 608 642 836 480 636 484 468 205 ...  
## $ SalePrice : int 208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...

house.test<-Housing[1461:2919,1:12]  
str(house.test)

## 'data.frame': 1459 obs. of 12 variables:  
## $ MasVnrArea : int 0 108 0 20 0 0 0 0 0 0 ...  
## $ BsmtFinSF1 : int 468 923 791 602 263 0 935 0 637 804 ...  
## $ BsmtUnfSF : int 270 406 137 324 1017 763 233 789 663 0 ...  
## $ TotalBsmtSF : int 882 1329 928 926 1280 763 1168 789 1300 882 ...  
## $ X1stFlrSF : int 896 1329 928 926 1280 763 1187 789 1341 882 ...  
## $ X2ndFlrSF : int 0 0 701 678 0 892 0 676 0 0 ...  
## $ GrLivArea : int 896 1329 1629 1604 1280 1655 1187 1465 1341 882 ...  
## $ BsmtFullBath: int 0 0 0 0 0 0 1 0 1 1 ...  
## $ FullBath : int 1 1 2 2 2 2 2 2 1 1 ...  
## $ BedroomAbvGr: int 2 3 3 3 2 3 3 3 2 2 ...  
## $ TotRmsAbvGrd: int 5 6 6 7 5 7 6 7 5 4 ...  
## $ GarageArea : int 730 312 482 470 506 440 420 393 506 525 ...

imp<-apply(house.train,2,mean,na.rm= T)  
house.reg<- house.train  
for(i in 1:ncol(house.train))  
{house.reg[is.na(house.reg[,i]),i]<-imp[i]  
}  
sapply(house.reg, function(x) sum(is.na(x)))#imputing nas with col means

## MasVnrArea BsmtFinSF1 BsmtUnfSF TotalBsmtSF X1stFlrSF   
## 0 0 0 0 0   
## X2ndFlrSF GrLivArea BsmtFullBath FullBath BedroomAbvGr   
## 0 0 0 0 0   
## TotRmsAbvGrd GarageArea SalePrice   
## 0 0 0

house.train<-house.reg

Now that we have obtained our required train and test data sets, we perform linear regression as follows:

library(MASS)  
model1<- lm(SalePrice ~ .,data = house.train )  
#model2 <- stepAIC(model1)  
summary(model1)

##   
## Call:  
## lm(formula = SalePrice ~ ., data = house.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -632865 -17748 792 17766 270896   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -5719.269 5814.129 -0.984 0.32544   
## MasVnrArea 47.016 7.138 6.587 6.26e-11 \*\*\*  
## BsmtFinSF1 11.903 7.398 1.609 0.10786   
## BsmtUnfSF 6.275 7.397 0.848 0.39638   
## TotalBsmtSF 34.856 8.409 4.145 3.59e-05 \*\*\*  
## X1stFlrSF 57.582 24.239 2.376 0.01765 \*   
## X2ndFlrSF 69.282 23.829 2.908 0.00370 \*\*   
## GrLivArea -5.529 23.570 -0.235 0.81458   
## BsmtFullBath 12143.528 2986.478 4.066 5.04e-05 \*\*\*  
## FullBath 21510.271 2728.546 7.883 6.22e-15 \*\*\*  
## BedroomAbvGr -17210.216 1955.078 -8.803 < 2e-16 \*\*\*  
## TotRmsAbvGrd 4528.538 1443.424 3.137 0.00174 \*\*   
## GarageArea 72.826 6.663 10.929 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 42830 on 1447 degrees of freedom  
## Multiple R-squared: 0.7118, Adjusted R-squared: 0.7094   
## F-statistic: 297.8 on 12 and 1447 DF, p-value: < 2.2e-16

We use the above built model to predict the SalePrices in the test dataset.

x <- predict(model1, house.test)  
summary(x)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 38368 128170 169124 178159 214849 721882 18

We have submitted these results to the Kaggle Competition and obtained a score of 0.21 with a rank of 4525 on 04.24.2018.

Following we perform the principle component regression, which provides us with reduced dimensions keeping most of the variability, avoids multicollinearity between the predictor variables and reduces the risk of overfitting.

However, doing so the interpretation becomes difficult and also some deviation of accuracy from the complete model prediction.

# creating dataset for predictions using pca  
p.house.new <- house.new  
a <- row.names(house.new[house.new$outlier=="Yes",])  
p.house.train <- house.train  
p.house.train[-c(as.numeric(a)),] -> p.house.train.new

We build the training dataset using the 3 PC scores and append the SalePrice variable.

#Creating training dataset  
p.house.train.data <- data.frame(house.pca$scores[,1:3], SalePrice = p.house.train.new$SalePrice)

#Test dataset pca  
y <- predict(house.pca, newdata = house.test)

#PCR  
mf1 <- lm(SalePrice ~ ., data = p.house.train.data)  
results <- predict(mf1, as.data.frame(y[,1:3]))

We have submitted these results to the Kaggle competition as well and obtained a score of 0.23 and rank 4585, which is lower than the complete linear regression model as expected, since our 3 principle components captured only 75 percent of variation in the data set.

### Future Work and Analysis

* The clustering features show that we have distinct classes of houses in the market for sale. We can relate the prices of the houses sold with their classes to find further meaningful patterns. i.e., to answer questions like:
* Whether houses with big basement do get sold at above average price or lower price?
* What about the actual sale prices of houses having full baths in basements vs. without full baths?
* Acquiring further information about the customers demographics would be helpful in understanding what cluster(type) would they prefer to purchase.
* Extensive regression techniques may be utilized along with considering other variables in the original dataset to ensure accuracy in predictions of Houses Sale Price.
* Inclusion of categorical variables (like Neighbourhood) in analysis would justify or contradict the outliers excluded.

### 

### References

**Dataset**

* House prices advance regression techniques, Kaggle.com (<https://www.kaggle.com/c/house-prices-advanced-regression-techniques>)

**Textbook**

* <https://ttu.blackboard.com/bbcswebdav/pid-3383258-dt-content-rid-25030997_1/courses/201857-ISQS-6350-001/An%20Introduction%20to%20Applied%20Multivariate%20Analysis-Everet.pdf>

**Outlier Analysis**

* Outlier Detection with Mahalanobis Distance. (2016, December 08). Retrieved from <https://www.r-bloggers.com/outlier-detection-with-mahalanobis-distance/>

**PCA and Factor Analysis**

* Factor Analysis and Principal Components. (2002, May 07). Retrieved from <https://www.sciencedirect.com/science/article/pii/S0047259X8571069X>
* Use of Exploratory Factor Analysis in Published Research. (n.d.). Retrieved from <http://journals.sagepub.com/doi/abs/10.1177/0013164405282485>

**Multivariate Regression** +

* Manjula, R., Jain, S., Srivastava, S., & Kher, P. R. (2017). Real estate value prediction using multivariate regression models. IOP Conference Series: Materials Science and Engineering, 263, 042098. <doi:10.1088/1757-899x/263/4/042098>

**Principle component regression**

* Bansal, S., Dar, P., Jain, K., Jain, S., & Analytics Vidhya Content Team. (2016, July 27). Practical Guide to Principal Component Analysis (PCA) in R & Python. Retrieved from <https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/>
* Jolliffe, I. (1982). A Note on the Use of Principal Components in Regression. Journal of the Royal Statistical Society. Series C (Applied Statistics), 31(3), 300-303. <doi:10.2307/2348005>