Data Prep Assignment

1. The Sales Dataset contains sales numbers for 811 products across 52 weeks. In addition to raw data, the file also contains min-max normalized version of the data. Explore **dimension reduction** solutions for the Sales dataset (3 points)

a. Discuss the appropriateness of the technique used for the given dataset (Why is the technique you have chosen appropriate in this context?)

I use Principal Component Analysis to reduce dimensions based on the dataset, which contains 811 observations and 52 attributes. PCA is a statistical procedure that transforms and converts a data set into a new data set containing linearly uncorrelated variables, known as principal components. The basic idea is that the data set is transformed into a set of components where each one attempts to capture as much of the variance (information) in data as possible. Furthermore, it is an easy and fast way to get the overall view of data reduction. It also satisfied many types of dataset.

I choose normalized data to do PCA in R because the input data are normalized, so that each attribute falls within the same range. In our data, Min-Max function converts the numeric data into values between 0 and 1. This normalization step helps ensure that differences in scale won't affect the significance of the predictors. In order to explain this, I use raw data and normalized data to do PCA in R respectively and find that there is a large standard deviation in Comp1. of raw data. So, Doing PCA should use scaled data.

b. Discuss the quality of the solution in terms of the original data and your recommendation (How well did the technique perform? Should we use it? If yes, what would be the reduced dimensions?)

We obtain 52 principal components. Each component contains standard deviation, proportion of variance, and cumulative proportion. Each of these explains a percentage of the total variation in the dataset. That is to say: Comp.1 explains 33% of the total variance, which means that nearly one-thirds of the information in the dataset (52 variables) can be captured by just that one Principal Component. PC2 explains 4% of the variance. So, by knowing the position of a sample in relation to PC1 and PC2, you can get a view on where it stands in relation to other samples, as just PC1 and PC2 can explain 37% of the variance (see cumulative proportion). From Comp.1 to comp.40, they can explain 90% of the variance. Through analysis for output of PCA, I think PCA approach is feasible for dimension reduction, but we may find more suitable method for this type data.

c. Include code used to perform the above analysis and outputs (summaries, relevant metrics, plots)

Code:

data=read.csv("C:\\Users\\ludai\\Desktop\\BAN620\\Assignment1\\Sales_Transactions_Dataset_Weekly.csv",header=T)

pcadata=princomp(data[-1,2:53])

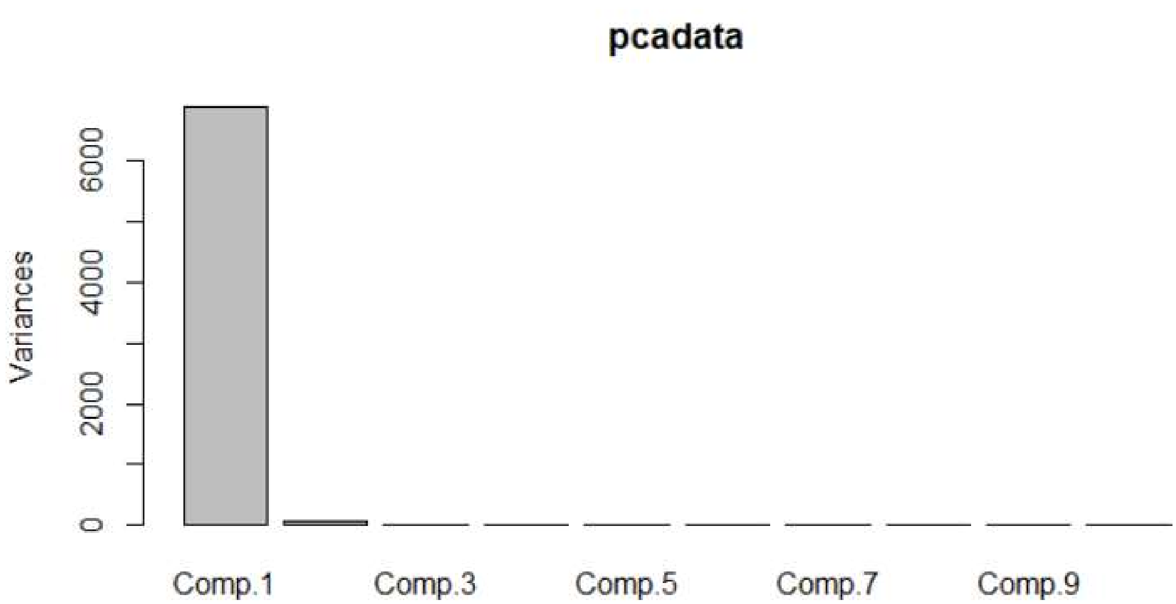
summary(pcadata)
plot(pcadata)

pcadata_Normalized=princomp(data[c(56:107)])

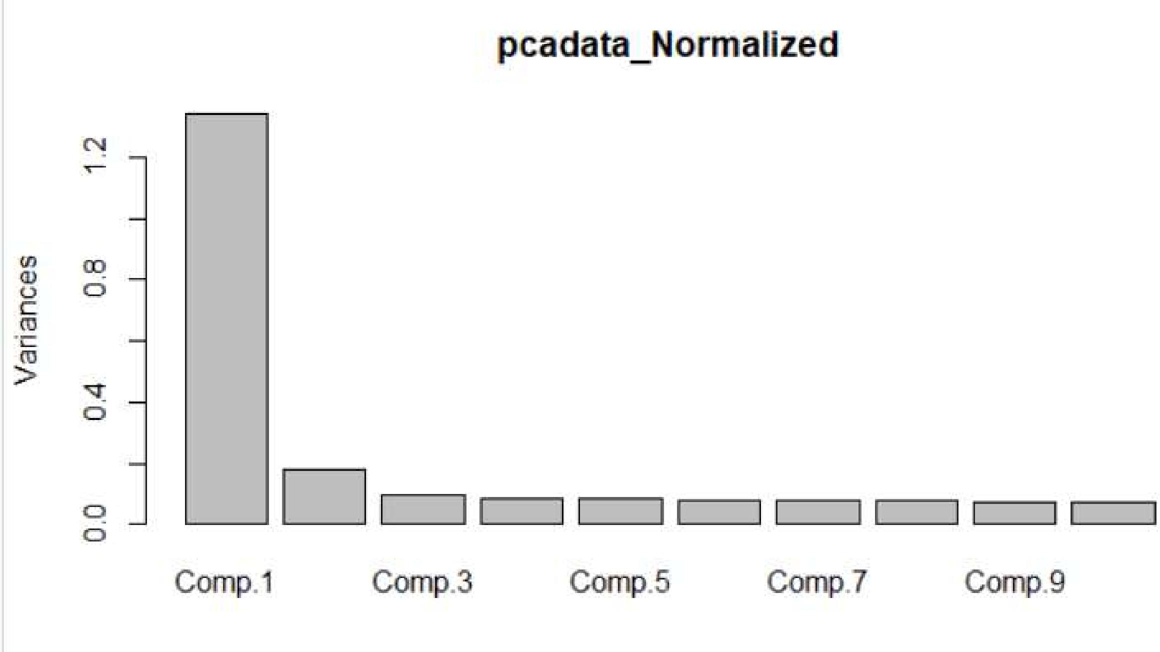
summary(pcadata_Normalized)

plot(pcadata_Normolization)

```
> pcadata=princomp(data[-1,2:53])
> summary(pcadata)
Importance of components:
                          Comp.1      Comp.2      Comp.3      Comp.4      Comp.5      Comp.6      Comp.7      Comp.8      Comp.9
Standard deviation     82.9392674 7.878361506 4.594213975 4.365127062 4.332973428 4.244587979 4.202417941 4.142200913 4.055680142
Proportion of Variance  0.9228515 0.008326898 0.002831613 0.002556261 0.002518741 0.002417033 0.002369245 0.002301833 0.002206677
Cumulative Proportion   0.9228515 0.931178446 0.934010059 0.936566320 0.939085061 0.941502093 0.943871338 0.946173171 0.948379849
                          Comp.10     Comp.11     Comp.12     Comp.13     Comp.14    Comp.15     Comp.16     Comp.17     Comp.18
Standard deviation     3.985767468 3.870856133 3.849872809 3.702777041 3.693661019 3.62735435 3.604029901 3.571435771 3.517539189
Proportion of Variance 0.002131255 0.002010136 0.001988402 0.001839359 0.001830314 0.00176519 0.001742562 0.001711186 0.001659928
Cumulative Proportion  0.950511103 0.952521240 0.954509642 0.956349001 0.958179315 0.95994450 0.961687067 0.963398252 0.965058181
                         Comp.19     Comp.20     Comp.21     Comp.22     Comp.23     Comp.24     Comp.25    Comp.26     Comp.27
Standard deviation     3.44675030 3.387250363 3.345773075 3.314068368 3.282824674 3.211874528 3.156463107 3.11447421 3.063608033
Proportion of Variance 0.00159379 0.001539239 0.001501773 0.001473447 0.001445795 0.001383976 0.001336635 0.00130131 0.001259151
Cumulative Proportion  0.96665197 0.968191210 0.969692983 0.971166430 0.972612225 0.973996201 0.975332836 0.97663415 0.977893298
                          Comp.28     Comp.29     Comp.30     Comp.31     Comp.32     Comp.33     Comp.34     Comp.35      Comp.36
Standard deviation     3.046753642 2.987995890 2.947926565 2.942524831 2.887818076 2.874363933 2.787442588 2.750204317 2.7109152875
Proportion of Variance 0.001245335 0.001197765 0.001165856 0.001161587 0.001118797 0.001108396 0.001042373 0.001014709 0.0009859238
Cumulative Proportion  0.979138633 0.980336397 0.981502253 0.982663840 0.983782636 0.984891032 0.985933406 0.986948114 0.9879340382
                           Comp.37      Comp.38      Comp.39     Comp.40      Comp.41      Comp.42      Comp.43      Comp.44
Standard deviation     2.6658207732 2.6331507347 2.5903436802 2.581704371 2.5324499413 2.4887424026 2.4673917170 2.3684781208
Proportion of Variance 0.0009533961 0.0009301712 0.0009001735 0.000894179 0.0008603857 0.0008309432 0.0008167472 0.0007525757
Cumulative Proportion  0.9888874343 0.9898176055 0.9907177790 0.991611958 0.9924723438 0.9933032870 0.9941200342 0.9948726099
                           Comp.45      Comp.46      Comp.47      Comp.48      Comp.49   Comp.50      Comp.51      Comp.52
Standard deviation     2.3590674045 2.3559553541 2.2827843525 2.1866407706 2.1265906864 2.1169166 2.0747335287 1.9501337112
Proportion of Variance 0.0007466072 0.0007446386 0.0006991031 0.0006414552 0.0006067074 0.0006012 0.0005774789 0.0005101997
Cumulative Proportion  0.9956192171 0.9963638557 0.9970629588 0.9977044141 0.9983111214 0.9989123 0.9994898003 1.0000000000
```



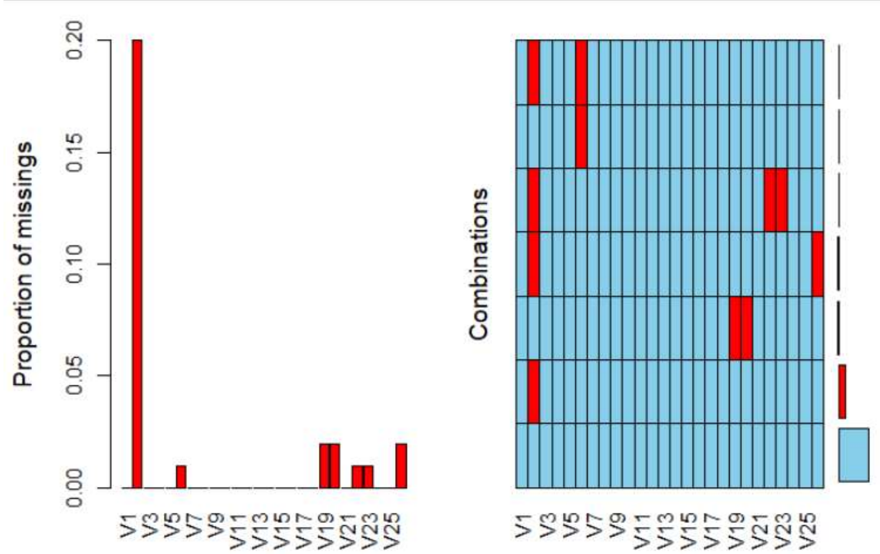
```
> pcadata_Normalized=princomp(data[c(56:107)])
> summary(pcadata_Normalized)
Importance of components:
                          Comp.1     Comp.2     Comp.3    Comp.4     Comp.5     Comp.6     Comp.7     Comp.8     Comp.9    Comp.10
Standard deviation     1.1575278 0.42657823 0.31383118 0.2926315 0.29044628 0.28263815 0.27995075 0.27575645 0.26582955 0.26477084
Proportion of Variance 0.3313972 0.04500734 0.02436005 0.0211801 0.02086496 0.01975821 0.01938426 0.01880777 0.01747803 0.01733909
Cumulative Proportion  0.3313972 0.37640452 0.40076457 0.4219447 0.44280963 0.46256783 0.48195209 0.50075986 0.51823789 0.53557698
                          Comp.11    Comp.12    Comp.13    Comp.14    Comp.15    Comp.16    Comp.17    Comp.18    Comp.19    Comp.20
Standard deviation     0.26274002 0.25463701 0.25061407 0.24829645 0.24532392 0.24146938 0.23863971 0.23767559 0.23596770 0.23361507
Proportion of Variance 0.01707412 0.01603722 0.01553448 0.01524849 0.01488558 0.01442149 0.01408547 0.01397189 0.01377181 0.01349857
Cumulative Proportion  0.55265110 0.56868832 0.58422280 0.59947130 0.61435688 0.62877836 0.64286383 0.65683572 0.67060753 0.68410610
                          Comp.21    Comp.22    Comp.23    Comp.24    Comp.25    Comp.26   Comp.27    Comp.28    Comp.29    Comp.30
Standard deviation     0.23037013 0.22858089 0.22679403 0.22506129 0.22284857 0.22104460 0.2198610 0.21647916 0.21424919 0.21319570
Proportion of Variance 0.01312618 0.01292307 0.01272182 0.01252817 0.01228303 0.01208498 0.0119559 0.01159093 0.01135336 0.01124198
Cumulative Proportion  0.69723228 0.71015535 0.72287717 0.73540534 0.74768837 0.75977335 0.7717292 0.78332017 0.79467353 0.80591551
                          Comp.31   Comp.32    Comp.33    Comp.34    Comp.35   Comp.36     Comp.37     Comp.38     Comp.39
Standard deviation     0.21017680 0.2092383 0.20732418 0.20571280 0.20465328 0.2033130 0.198063865 0.195843100 0.194610089
Proportion of Variance 0.01092586 0.0108285 0.01063129 0.01046667 0.01035913 0.0102239 0.009702786 0.009486423 0.009367348
Cumulative Proportion  0.81684137 0.8276699 0.83830116 0.84876783 0.85912696 0.8693509 0.879053641 0.888540064 0.897907412
                          Comp.40     Comp.41     Comp.42     Comp.43     Comp.44     Comp.45     Comp.46     Comp.47     Comp.48
Standard deviation     0.19244816 0.189986538 0.186807674 0.184509478 0.181432915 0.179120872 0.178247810 0.176387547 0.175187785
Proportion of Variance 0.00916038 0.008927536 0.008631283 0.008420217 0.008141756 0.007935574 0.007858404 0.007695233 0.007590906
Cumulative Proportion  0.90706779 0.915995328 0.924626611 0.933046828 0.941188585 0.949124158 0.956982562 0.964677796 0.972268701
                           Comp.49     Comp.50     Comp.51     Comp.52
Standard deviation     0.171848239 0.168721390 0.167379543 0.161572485
Proportion of Variance 0.007304258 0.007040869 0.006929321 0.006456851
Cumulative Proportion  0.979572959 0.986613828 0.993543149 1.000000000
```



2. Analyze the Imports85 dataset (https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.names) (3 Points)

a. Present a summary of missing values

```
> a

 Missings in variables:
 Variable Count
       V2    41
       V6     2
      V19     4
      V20     4
      V22     2
      V23     2
      V26     4
> |
```



b. Discuss the overall effect of missing values on building a predictive model if the goal is to predict normalized losses.

Missing values in data is a common phenomenon in real world problems. Knowing how to handle missing values effectively is an important step to reduce bias and to produce powerful models.

According to the plot, normalized losses variable (V2) has 20% missing values less than 50%. So, creating predictive modeling is a feasible option, even though prediction data might exist bias.

c. Discuss recommended action for each feature w.r.t missing values.

For missing value, sometimes we can Ignore the tuple when class label is missing (assuming the mining task involves classification). However, this method is not very effective, unless the tuple contains several attributes with missing values.

If the dataset has enough large number of observations and variables, where all the classes to be predicted are sufficiently represented, we can delete the observations or variables that contain the missing value. In this Imports85 dataset, deleting the observations or variables is not an appropriate method for missing values because the Imports85 only contain 205 observations and 26 attributes, however the missing values have 59. If we delete the variables and observations, the method may cause the predictive model's bias.

We can replace missing value with mean, median or mode of attributes. If the variation of dataset is low or the variables has low leverage over the response, the imputing missing values with mean, median, or mode is acceptable and could possibly give satisfactory results. For example, in this Imports85 data, we can use V26 attribute mean or median to handle the missing values.

We also can use inference-based such as Bayesian formula or decision tree to impute the probable value for missing value.

d. Discuss the best possible technique to impute missing values for feature nos 6, 19, 20, 22, 23, 26 (list 2 or more techniques and identify the best among them along with justification).

For columns 6,19,20,22,23,26, and based on the above summary, the missing values are less than 4. We can use mean. median or mode matching method to handle missing value.

We can also use prediction model to impute missing values. Prediction is most advanced method to impute your missing values and includes different approaches such as: kNN Imputation, rpart, and mice in R.

In this data, we use kNN function to impute missing values. kNN Imputation uses k-Nearest Neighbours approach to impute missing values. The kNN advantage is that you could impute all the missing values in all variables with one call to the function. It takes the whole data frame as the argument and you don't even have to specify which variable you want to impute.

e. Include code used to perform the above analysis and outputs (summaries, relevant metrics, plots)

Code:

library(VIM)

data <-imports.85

data[data=="?"]<-NA

a<- aggr(data)

plot(a)

a

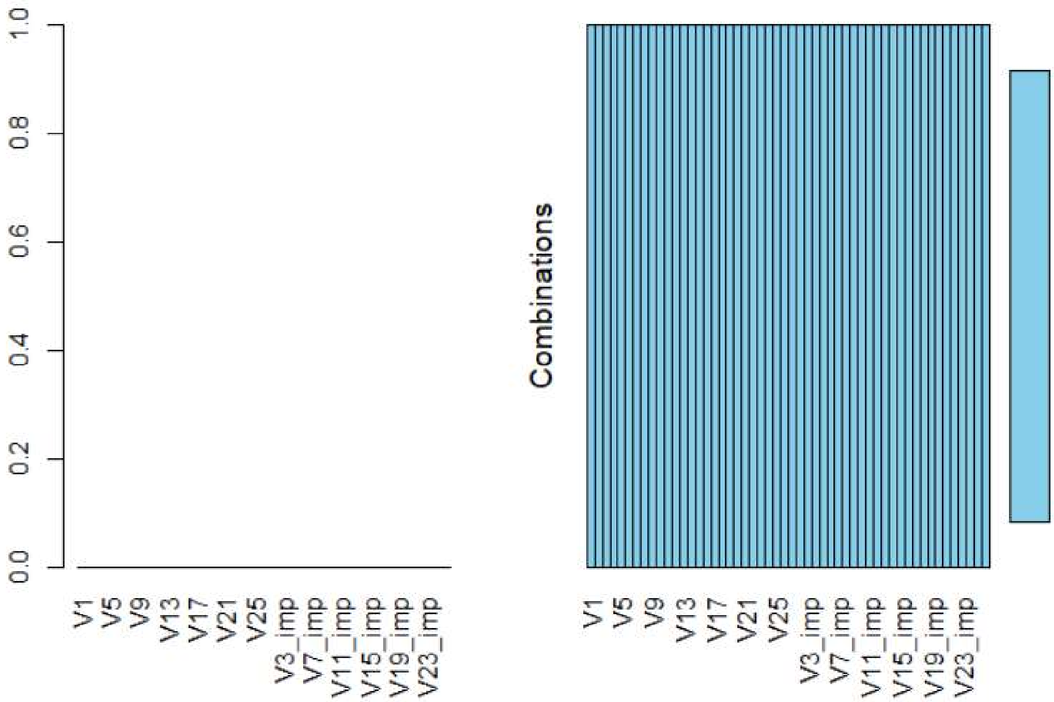
summary(a)

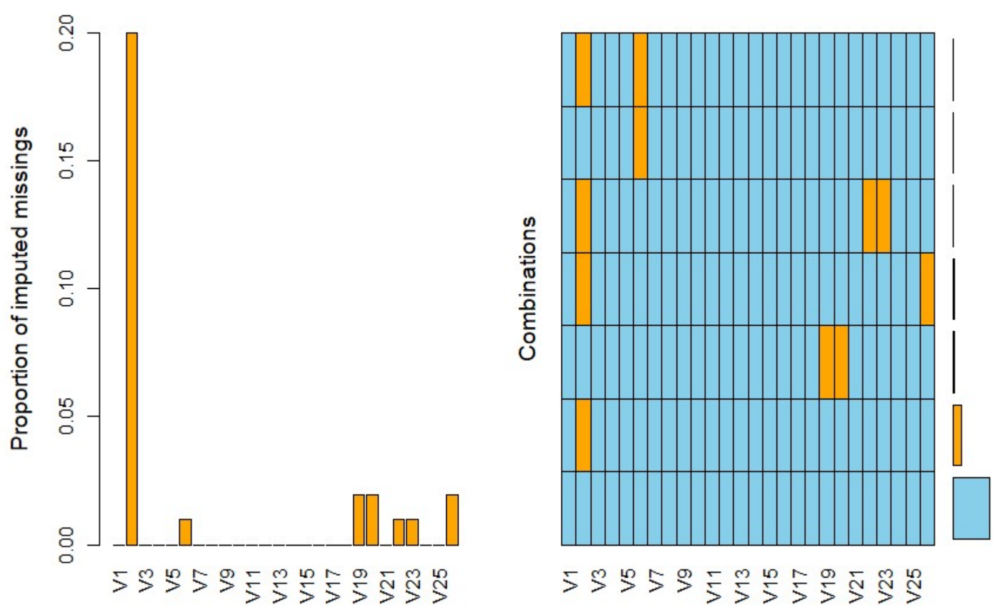
output:

```
28   3.03 3.39  7.60  102 5500  24  30  8558
29   3.34 3.46  8.50   88 5000  24  30  8921
30   3.60 3.90  7.00  145 5000  19  24 12964
31   2.91 3.41  9.60   58 4800  49  54  6479
32   2.91 3.41  9.20   76 6000  31  38  6855
33   2.91 3.07 10.10   60 5500  38  42  5399
34   2.91 3.41  9.20   76 6000  30  34  6529
35   2.91 3.41  9.20   76 6000  30  34  7129
36   2.91 3.41  9.20   76 6000  30  34  7295
37   2.92 3.41  9.20   76 6000  30  34  7295
38   3.15 3.58  9.00   86 5800  27  33  7895
 [ reached getOption("max.print") -- omitted 167 rows ]
> a<- aggr(data)
> a

 Missings in variables:
 Variable Count
       V2    41
       V6     2
      V19     4
      V20     4
      V22     2
      V23     2
      V26     4
> |

      V25     0
      V26     4

 Missings in combinations of variables:
                                             Combinations Count    Percent
 0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0   159 77.5609756
 0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:1:1:0:0:0:0:0:0     4  1.9512195
 0:0:0:0:0:1:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0     1  0.4878049
 0:1:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0    34 16.5853659
 0:1:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:1     4  1.9512195
 0:1:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:1:1:0:0:0     2  0.9756098
 0:1:0:0:0:1:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0     1  0.4878049
> |
```

Code:

```
akknn <- kNN(data)
x=aggr(aknn)
plot(x)
impute_missing<-aggr(aknn,delimiter="_imp")
```





```
> x=aggr(aknn)
> plot(x)
> impute_missing<-aggr(aknn,delimiter="_imp")
> summary(impute_missing)

 Imputed missings per variables:
 Variable Count
       V1     0
       V2    41
       V3     0
       V4     0
       V5     0
       V6     2
       V7     0
       V8     0
       V9     0
      V10     0
      V11     0
      V12     0
      V13     0
      V14     0
      V15     0
      V16     0
      V17     0
      V18     0
      V19     4
      V20     4
      V21     0
      V22     2
      V23     2
      V24     0
      V25     0
      V26     4

 Imputed missings in combinations of variables:
                                              Combinations Count    Percent
 0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0   159 77.5609756
 0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:2:2:0:0:0:0:0:0     4  1.9512195
 0:0:0:0:0:2:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0     1  0.4878049
 0:2:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0    34 16.5853659
 0:2:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:2     4  1.9512195
 0:2:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:2:2:0:0:0     2  0.9756098
 0:2:0:0:0:2:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0:0     1  0.4878049
> 
```

3. Analyze the Imports85Imp dataset (Same data as before but with no missing values) (4 points)

a. The dataset has several continuous valued variables that appear to be closely related to each other (engine size, bore etc.). Perform PCA to see if the 14 continuous valued variables (nos 10 -14,17,19-26) can be reduced to fewer principal components.

First, the 14 continuous valued variable can be reduced to fewer principal components. The proportion of variance in Comp.1 is 0.995 that can explain 99.5% of variance, however, the standard deviation is 7922. Such high standard deviation indicates the Component may be dominated by a variable with high value (variable V26). This is undesirable. So, we should normalize the raw data before performing PCA. See the following question analysis.

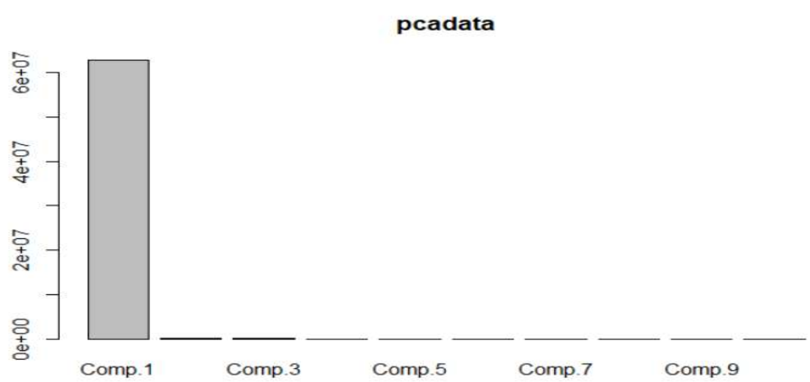
Code:

```
data=Imports85Imp

pcadata=princomp(data[,c(10：14，17，19：26)])

summary(pcadata)

plot(pcadata)

pcadata$loadings
```

```
> pcadata=princomp(data[,c(10:14, 17, 19:26)])
> summary(pcadata)
Importance of components:
                            Comp.1       Comp.2       Comp.3       Comp.4       Comp.5       Comp.6       Comp.7
Standard deviation     7922.941431 4.890997e+02 2.592576e+02 2.387760e+01 1.341187e+01 6.178406e+00 4.883842e+00
Proportion of Variance    0.995129 3.792288e-03 1.065539e-03 9.038320e-06 2.851578e-06 6.051448e-07 3.781200e-07
Cumulative Proportion     0.995129 9.989213e-01 9.999869e-01 9.999959e-01 9.999988e-01 9.999994e-01 9.999997e-01
                             Comp.8       Comp.9      Comp.10      Comp.11      Comp.12      Comp.13      Comp.14
Standard deviation     2.587413e+00 2.299480e+00 1.679505e+00 9.592733e-01 8.766715e-01 2.873999e-01 1.832342e-01
Proportion of Variance 1.061300e-07 8.382349e-08 4.471663e-08 1.458785e-08 1.218373e-08 1.309422e-09 5.322547e-10
Cumulative Proportion  9.999998e-01 9.999999e-01 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
> plot(pcadata)
>
```

```
Loadings:
    Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10 Comp.11 Comp.12 Comp.13 Comp.14
V10                                    -0.316 -0.220  0.577  0.648 -0.226           0.183
V11                                    -0.821 -0.372 -0.285 -0.300
V12                                                          0.105 -0.216  -0.172  -0.951
V13                                    -0.124 -0.104  0.214         0.940          -0.172
V14        -0.281 -0.957
V17                      -0.583  0.808
V19                                                                                        -0.236   0.972
V20                                                                                         0.971   0.236
V21                                     0.215 -0.420 -0.668  0.562
V22                      -0.798 -0.573        -0.169
V23         0.959 -0.281
V24                              0.101  0.285 -0.519  0.238 -0.155          0.735  -0.120
V25                                     0.276 -0.569  0.185 -0.362         -0.639   0.118
V26 -0.998

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10 Comp.11 Comp.12 Comp.13 Comp.14
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000   1.000   1.000   1.000   1.000   1.000
Proportion Var  0.071  0.071  0.071  0.071  0.071  0.071  0.071  0.071  0.071   0.071   0.071   0.071   0.071   0.071
Cumulative Var  0.071  0.143  0.214  0.286  0.357  0.429  0.500  0.571  0.643   0.714   0.786   0.857   0.929   1.000
>
```

```
> str(pcadata)
List of 7
 $ sdev    : Named num [1:14] 7922.9 489.1 259.3 23.9 13.4 ...
  ..- attr(*, "names")= chr [1:14] "Comp.1" "Comp.2" "Comp.3" "Comp.4" ...
 $ loadings: loadings [1:14, 1:14] -4.44e-04 -1.08e-03 -2.03e-04 -4.02e-05 -5.49e-02 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:14] "V10" "V11" "V12" "V13" ...
  .. ..$ : chr [1:14] "Comp.1" "Comp.2" "Comp.3" "Comp.4" ...
 $ center  : Named num [1:14] 98.8 174 65.9 53.7 2555.6 ...
  ..- attr(*, "names")= chr [1:14] "V10" "V11" "V12" "V13" ...
 $ scale   : Named num [1:14] 1 1 1 1 1 1 1 1 1 1 ...
  ..- attr(*, "names")= chr [1:14] "V10" "V11" "V12" "V13" ...
 $ n.obs   : int 205
 $ scores  : num [1:205, 1:14] -288 -3288 -3304 -728 -4249 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:205] "1" "2" "3" "4" ...
  .. ..$ : chr [1:14] "Comp.1" "Comp.2" "Comp.3" "Comp.4" ...
 $ call    : language princomp(x = data[, c(10:14, 17, 19:26)])
 - attr(*, "class")= chr "princomp"
> 
```



b. Compare the PCA results for (a) Raw data (b) Standardized data with mean 0 and sd 1 and (c) Min-Max normalized data. Describe the pros and cons of using principal components based on raw, standardized, and scaled data for data mining applications.

In question3.a, analyzing the PCA result, we saw the first principal component has a large standard deviation that shows this component is dominated by a variable with high values. So, doing PCA should scale raw data.

Standardization (also called z-score normalization) will transform variables so that they have zero mean and standard deviation of one. Normalization scales all numeric variables in the range [0,1].

Which method is better for transformation data is really dependent on application. In PCA we usually prefer standardization over Min-Max scaling, since we are interested in the components that maximize the variance.

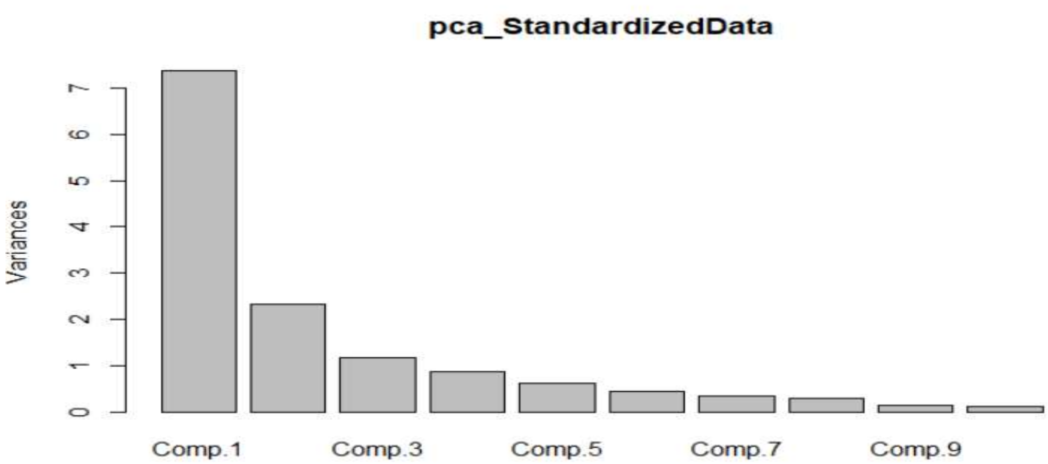
Sometimes Min-Max normalization may lose some information in the data, especially about outliers because it scales the “normal” data into a very small interval.

In this dataset, I think standardization and normalization are suitable for doing PCA. We read similar results of PCA for standardized data with mean 0 and SD 1 and Min-max normalized data. The proportion of variance of their Comp.1 is over 0.5 that explains 50% variance. For components 1 to 6, both results can get over 90% explanation. These two results are more accurate and desirable than PCA result from raw data.

Code:

```
StandardizedData <- scale(data[,c(10：14，17，19：26)])
StandardizedData
pca_StandardizedData=princomp(StandardizedData)
summary(pca_StandardizedData)
plot(pca_StandardizedData)
```

```
> pca_StandardizedData=princomp(StandardizedData)
> pca_StandardizedData=princomp(StandardizedData)
> summary(pca_StandardizedData)
Importance of components:
                          Comp.1    Comp.2     Comp.3     Comp.4     Comp.5     Comp.6     Comp.7     Comp.8
Standard deviation     2.7145495 1.5242199 1.08424429 0.93322138 0.79129213 0.66596307 0.58832743 0.53255681
Proportion of Variance 0.5289215 0.1667596 0.08438203 0.06251223 0.04494375 0.03183435 0.02484471 0.02035765
Cumulative Proportion  0.5289215 0.6956811 0.78006311 0.84257535 0.88751910 0.91935345 0.94419815 0.96455580
                           Comp.9     Comp.10     Comp.11     Comp.12    Comp.13     Comp.14
Standard deviation     0.38562267 0.345021195 0.296297906 0.262184582 0.22446230 0.138337378
Proportion of Variance 0.01067384 0.008544511 0.006301629 0.004934123 0.00361645 0.001373646
Cumulative Proportion  0.97522964 0.983774152 0.990075781 0.995009904 0.99862635 1.000000000
> plot(pca_StandardizedData)
```



Code:

```
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}
NormData <- as.data.frame(lapply(data[,c(10： 14， 17， 19： 26)], normalize))
pca_NormData=princomp(NormData)
summary(pca_NormData)
plot(pca_NormData)
```

```
> pca_NormData=princomp(NormData)
> summary(pca_NormData)
Importance of components:
                          Comp.1    Comp.2     Comp.3     Comp.4     Comp.5     Comp.6     Comp.7     Comp.8
Standard deviation     0.4966054 0.3152042 0.20479417 0.16871400 0.15080859 0.12517022 0.10910689 0.10005062
Proportion of Variance 0.5004373 0.2016094 0.08510636 0.05776025 0.04615077 0.03179281 0.02415634 0.02031264
Cumulative Proportion  0.5004373 0.7020468 0.78715314 0.84491339 0.89106417 0.92285697 0.94701331 0.96732595
                            Comp.9     Comp.10     Comp.11     Comp.12     Comp.13     Comp.14
Standard deviation     0.069383418 0.062807761 0.053853421 0.045082303 0.042231987 0.025036825
Proportion of Variance 0.009768732 0.008004855 0.005885094 0.004124193 0.003619177 0.001271995
Cumulative Proportion  0.977094686 0.985099540 0.990984635 0.995108828 0.998728005 1.000000000
> plot(pca_NormData)
>
```