Assignment 1 IDA

Lim Xiang Ling 2709194

2023-10-23

1. Describe mtcars Dataset

This dataset includes information about various car models and their performance characteristics. The mtcars dataset is also derived from the 1974 Motor Trend US magazine and comprises 32 observations on 11 variables.

The variables include:

Mpg - The mpg variable represents miles per gallon (mpg) for the car model.

Cyl - The cyl variable represents the number of cylinders in the engine of the car model.

Disp - The disp variable represents the displacement of the car model, which is the volume of air and fuel mixture that the engine can compress and burn in one cycle, in cubic inches.

Hp - The hp variable represents the gross horsepower of the car model.

Drat - The drat variable represents the rear axle ratio of the car model.

Wt - The wt variable represents the weight of the car model, in thousands of pounds.

Qsec - The qsec variable represents the time taken to cover a quarter-mile distance from a standing start.

Vs - The vs variable represents the type of engine (0 = V-shaped, 1 = straight).

Am - The am variable represents the type of transmission (0 = automatic, 1 = manual).

Gear - The gear represents the number of forward gears in the car model.

Carb - The carb variable represents the number of carburettors in the engine of the car model.

Since I am using R for this assignment, the mtcars dataset can be loaded directly by typing data(mtcars). You can also access the dataset by clicking on this link:

https://r-data.pmagunia.com/system/files/datasets/dataset-10551.csv?ref=hackernoon.com

```r
data(mtcars)

head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

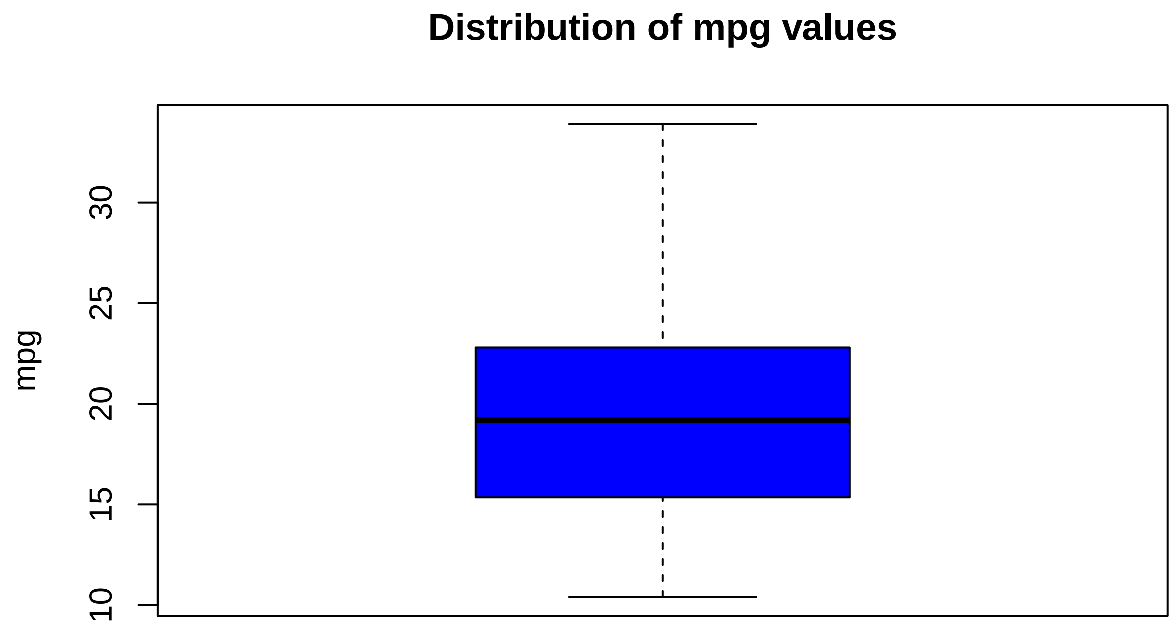
```r
summary(mtcars)
```

```
##       mpg             cyl             disp             hp       
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0  
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
##       drat             wt             qsec             vs        
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000  
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000  
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000  
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375  
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000  
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000  
##        am              gear            carb      
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000  
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000  
##  Median :0.0000   Median :4.000   Median :2.000  
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812  
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000  
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000  
```

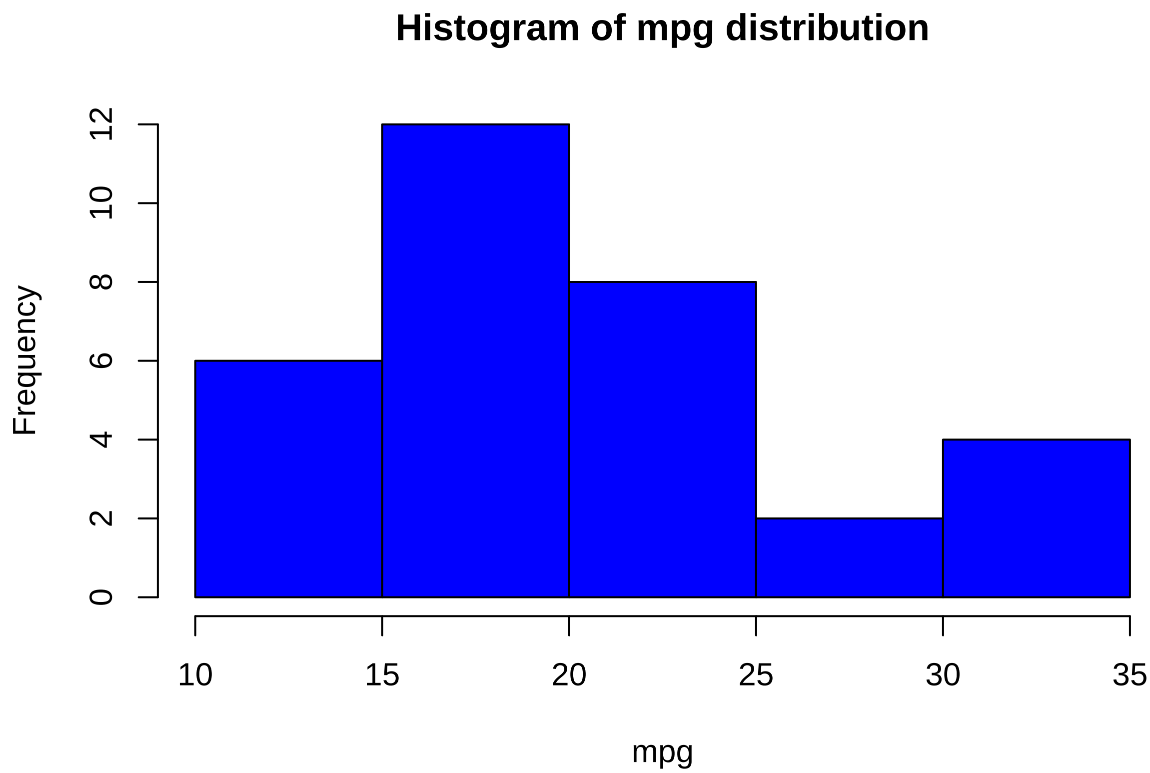
Exploratory Data Analysis

To visualise how the mpg distribution looks like:

```r
# boxplot of mpg values
boxplot(mtcars$mpg,
        main='Distribution of mpg values',
        ylab='mpg',
        col='blue',
        border='black')
```




```r
# histogram of mpg distribution
hist(mtcars$mpg,
     col='blue',
     main='Histogram of mpg distribution',
     xlab='mpg',
     ylab='Frequency')
```



I will discretise the mpg into:

"Very Low": 1

"Low": 2

"Medium": 3

"High": 4

"Very High":5

Based on the histogram, there are the most cars with mpg of 15-20 miles per gallon (Low), and least number of cars with 25-30 miles per gallon (High). From the boxplot, it seems like there are no anomalies as the majority of the data lies within the 2 whiskers.

2. How I preprocessed the data and why

First, I check for missing data:

```
which(is.na(mtcars))
```

```
## integer(0)
```

There is no missing data, hence no need for any form of data imputations.

From the original dataset, I will create 2 datasets:

a. dataframe consisting of only column 1: mpg only

b. dataframe consisting of the remaining columns, excluding categorical variables 'am' and 'vs' since PCA only works on numeric variables.

Then I will proceed to standardise all the variables using Z score standardisation as PCA will perform better when all the features are on the same scale. This is because according to the slides, we are interested in the overall inherent dependency structure of the data, regardless of (arbitrary) measurement units/scales in individual dimensions.

```
mtcars_new <- mtcars[ -c(8,9)]

mtcars_new$discretempg <- cut(as.numeric(unlist(mtcars_new["mpg"])),
              breaks=c(10,15, 20, 25, 30, 35),
              labels=c('Very Low', 'Low', 'Medium', 'High', 'Very High'))

# preprocessed data a
mtcars_mpg_labelled <- mtcars_new[c(10)]

# preprocessed data b
mtcars_new_pca <- mtcars_new[-c(1)]
mtcars_new_pca <- mtcars_new_pca[c(1:8)]

# zscore standardisation
mtcars_new_pca_zscore <- mtcars_new_pca
mtcars_new_pca_zscore <- dplyr::mutate_at(mtcars_new_pca_zscore, c(1:8), list(Z=scale))
mtcars_new_pca_zscore <- mtcars_new_pca_zscore[-c(1:8)]
```

3. What features (coordinates) did you use for labelling the projected points with different markers? What questions on the data did ask/investigate.

I used the 'mpg' feature for labelling the projected points with different markers.

"Very Low": orange

"Low": yellow

"Medium": green

"High": blue

"Very High": pink

I want to find out how many principle component is needed to capture at least 80% variance of the data.

I also want to find out the characteristics of cars with "Very Low" and "Very High" mpg (eg high 'dist' + high 'wt').

I will answer the next 2 questions together:

4. What interesting aspects of the data did you detect based on the data visualisations? 5. What interesting aspects of the data did you detect based on eigenvector and eigenvalue analysis of the data covariance matrix?

First I will create the data covariance matrix:

```
#covariance matrix
cov(mtcars_new_pca_zscore)
```

```
##              cyl_Z     disp_Z       hp_Z      drat_Z       wt_Z      qsec_Z
```

```
## cyl_Z   1.0000000  0.9020329  0.8324475 -0.69993811  0.7824958 -0.59124207
## disp_Z  0.9020329  1.0000000  0.7909486 -0.71021393  0.8879799 -0.43369788
## hp_Z    0.8324475  0.7909486  1.0000000 -0.44875912  0.6587479 -0.70822339
## drat_Z -0.6999381 -0.7102139 -0.4487591  1.00000000 -0.7124406  0.09120476
## wt_Z    0.7824958  0.8879799  0.6587479 -0.71244065  1.0000000 -0.17471588
## qsec_Z -0.5912421 -0.4336979 -0.7082234  0.09120476 -0.1747159  1.00000000
## gear_Z -0.4926866 -0.5555692 -0.1257043  0.69961013 -0.5832870 -0.21268223
## carb_Z  0.5269883  0.3949769  0.7498125 -0.09078980  0.4276059 -0.65624923
##            gear_Z     carb_Z
## cyl_Z  -0.4926866  0.5269883
## disp_Z -0.5555692  0.3949769
## hp_Z   -0.1257043  0.7498125
## drat_Z  0.6996101 -0.0907898
## wt_Z   -0.5832870  0.4276059
## qsec_Z -0.2126822 -0.6562492
## gear_Z  1.0000000  0.2740728
## carb_Z  0.2740728  1.0000000
```

Then I will perform eigendecomposition as the eigenvectors represent the principal components of the covariance matrix and the eigenvalues is the magnitude of the eigenvectors. The eigenvector that has the largest value is the feature with maximum variance.

```r
eigen(cov(mtcars_new_pca_zscore))
```

```
## eigen() decomposition
## $values
## [1] 4.80521984 2.08017995 0.48210218 0.26502098 0.17479228 0.10906851 0.05981502
## [8] 0.02380124
##
## $vectors
##            [,1]         [,2]        [,3]        [,4]        [,5]        [,6]
## [1,] -0.4397107  0.006493485  0.21389399  0.04117715  0.18531236  0.02585384
## [2,] -0.4329689 -0.097947929  0.02325701  0.34154106 -0.44542861  0.14459253
## [3,] -0.4018186  0.261415110 -0.02755330  0.06992909 -0.23581312 -0.78493300
## [4,]  0.3374188  0.348884539 -0.11277805  0.84480258  0.13928091 -0.03059658
## [5,] -0.3998604 -0.179105944 -0.50702904  0.19356866 -0.12225944  0.42501264
## [6,]  0.2521760 -0.478355318 -0.63254001 -0.02871245 -0.08864198 -0.37604432
## [7,]  0.2235402  0.555702724 -0.19470252 -0.28156232 -0.62706739  0.19478214
## [8,] -0.2653893  0.480205676 -0.49511212 -0.21331221  0.52557339  0.03719490
##             [,7]        [,8]
## [1,]  0.83363458  0.17091062
## [2,] -0.01593520 -0.68277769
## [3,] -0.18347022  0.24324000
## [4,]  0.12126786  0.05455856
## [5,] -0.18423726  0.53287799
## [6,]  0.36411288 -0.15750318
## [7,]  0.29552606  0.07448889
## [8,] -0.05074471 -0.35797632
```

Next I will calculate and plot the results of the PCA:

```r
#z score normalisation applied here where scale = true
pca_res <- prcomp(mtcars_new_pca,data = mtcars_new, scale. = TRUE, center = TRUE)
```

```
## Warning: In prcomp.default(mtcars_new_pca, data = mtcars_new, scale. = TRUE,
##     center = TRUE) :
##  extra argument 'data' will be disregarded
```

```r
head(pca_res)
```

```
## $sdev
## [1] 2.1920812 1.4422829 0.6943358 0.5148019 0.4180817 0.3302552 0.2445711
## [8] 0.1542765
##
## $rotation
##             PC1          PC2         PC3         PC4         PC5         PC6
## cyl   0.4397107 -0.006493485  0.21389399 -0.04117715  0.18531236 -0.02585384
## disp  0.4329689  0.097947929  0.02325701 -0.34154106 -0.44542861 -0.14459253
## hp    0.4018186 -0.261415110 -0.02755330 -0.06992909 -0.23581312  0.78493300
## drat -0.3374188 -0.348884539 -0.11277805 -0.84480258  0.13928091  0.03059658
## wt    0.3998604  0.179105944 -0.50702904 -0.19356866 -0.12225944 -0.42501264
## qsec -0.2521760  0.478355318 -0.63254001  0.02871245 -0.08864198  0.37604432
## gear -0.2235402 -0.555702724 -0.19470252  0.28156232 -0.62706739 -0.19478214
## carb  0.2653893 -0.480205676 -0.49511212  0.21331221  0.52557339 -0.03719490
##              PC7         PC8
## cyl   0.83363458 -0.17091062
## disp -0.01593520  0.68277769
## hp   -0.18347022 -0.24324000
## drat  0.12126786 -0.05455856
## wt   -0.18423726 -0.53287799
## qsec  0.36411288  0.15750318
## gear  0.29552606 -0.07448889
## carb -0.05074471  0.35797632
##
## $center
##        cyl       disp         hp       drat         wt       qsec       gear
##   6.187500 230.721875 146.687500   3.596563   3.217250  17.848750   3.687500
##       carb
##   2.812500
##
## $scale
##         cyl         disp           hp         drat           wt         qsec
##   1.7859216 123.9386938  68.5628685    0.5346787    0.9784574    1.7869432
##        gear         carb
##   0.7378041    1.6152000
##
## $x
##                             PC1        PC2         PC3         PC4          PC5
## Mazda RX4           -0.64738354 -1.1828309  0.26961666  0.12911933  0.704263314
## Mazda RX4 Wag       -0.62220221 -0.9862442 -0.06075048  0.08767061  0.644621749
## Datsun 710          -2.30846879  0.2933300  0.35170586  0.11256845 -0.316275460
## Hornet 4 Drive      -0.15488166  1.9814200  0.28127869  0.30702848 -0.209984621
## Hornet Sportabout    1.62839918  0.8573121  0.93256842 -0.14839587 -0.157042413
```

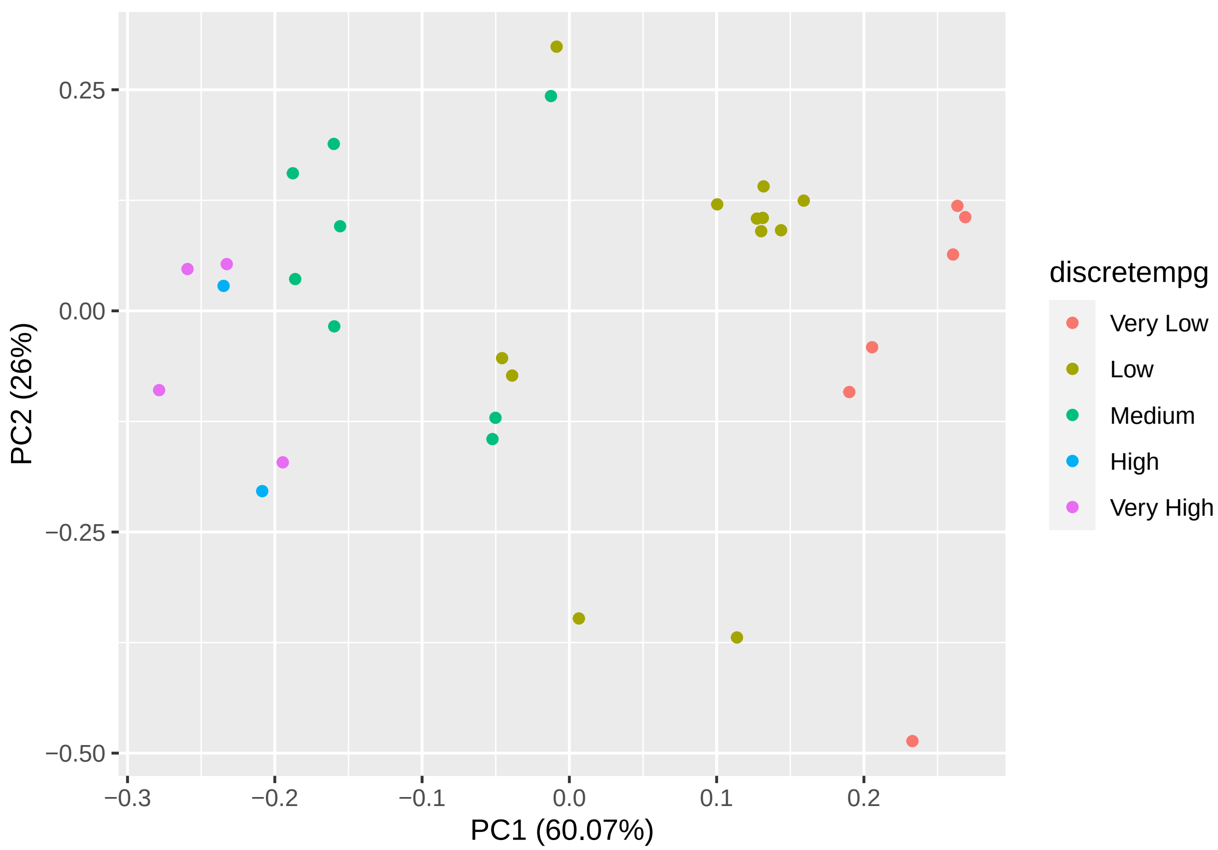
```
## Valiant             -0.10747735  2.4368571 -0.05846838  0.87273767 -0.226851141
## Duster 360           2.54904056 -0.3354249  0.62904594 -0.09513895  0.310906300
## Merc 240D           -1.93029468  0.7805570 -0.84421667  0.27263007 -0.242686562
## Merc 230            -2.32825091  1.2689874 -1.91290945 -0.05366293 -0.413925312
## Merc 280            -0.48182626 -0.5967823 -0.81463865 -0.06933960  0.443713547
## Merc 280C           -0.56649913 -0.4361654 -1.02702593 -0.05969885  0.413950324
## Merc 450SE           1.78218192  0.7436474  0.16412709  0.21847628  0.335354029
## Merc 450SL           1.61501184  0.7349496  0.26951669  0.28895221  0.367916364
## Merc 450SLC          1.57899647  0.8511800  0.10201556  0.28548786  0.341826655
## Cadillac Fleetwood   3.26713410  0.9686922 -0.90288066 -0.21854764 -0.343052500
## Lincoln Continental  3.33333123  0.8644244 -0.95744485 -0.34327289 -0.329888919
## Chrysler Imperial    3.23039020  0.5198102 -0.83321032 -0.65766022 -0.219973572
## Fiat 128            -2.88461143  0.4312932  0.06630635 -0.10499708 -0.085862645
## Honda Civic         -3.45424366 -0.7310324  0.22497113 -1.19293531  0.640104463
## Toyota Corolla      -3.21521612  0.3860437  0.07268163 -0.22511898  0.005636812
## Toyota Corona       -1.98342917  1.5400178  0.07719438 -0.07566508  0.349751177
## Dodge Challenger     1.63513920  1.1484089  1.02863676  0.59081358 -0.024260707
## AMC Javelin          1.24469613  0.9824145  0.83558369  0.03691111  0.116937919
## Camaro Z28           2.35697246 -0.7483198  0.52978651 -0.94951501  0.469896390
## Pontiac Firebird     1.97358574  1.0167655  0.73435205 -0.22766286 -0.371128155
## Fiat X1-9           -2.91142017  0.2304362  0.40545150 -0.06255744 -0.025553697
## Porsche 914-2       -2.58593127 -1.6625157  0.43142664 -0.31709082 -0.609796204
## Lotus Europa        -2.41298394 -1.3968870  0.81117918  0.89998091 -0.698398075
## Ford Pantera L       1.41124819 -3.0123180  0.56050480 -0.86468994 -1.042365987
## Ferrari Dino         0.08026304 -2.8371350 -0.31512992  1.14721865  0.291427207
## Maserati Bora        2.88838873 -3.9680590 -0.80278700  0.72583117 -0.037372868
## Volvo 142E          -1.97965873 -0.1428323 -0.24848721 -0.30947692 -0.081887413
##                             PC6          PC7         PC8
## Mazda RX4           -0.46009396  0.005912376  0.16202803
## Mazda RX4 Wag       -0.45301193  0.072004783  0.07251144
## Datsun 710           0.08388222 -0.297983964 -0.17963196
## Hornet 4 Drive       0.08039856 -0.003786184  0.16027080
## Hornet Sportabout    0.05057134  0.191712384  0.17883872
## Valiant              0.10106679  0.054062009 -0.03581427
## Duster 360           0.50454721 -0.309747655  0.19283141
## Merc 240D           -0.43374070 -0.168296230  0.03020595
## Merc 230             0.59175103  0.394765850  0.13455350
## Merc 280            -0.28796106  0.195207235 -0.12866518
## Merc 280C           -0.16169707  0.317465024 -0.07578051
## Merc 450SE          -0.01524720  0.098402133 -0.38257509
## Merc 450SL           0.17452663  0.203174546 -0.17977937
## Merc 450SLC          0.23698412  0.275265060 -0.17175344
## Cadillac Fleetwood  -0.37947478 -0.160896060  0.25399449
## Lincoln Continental -0.35623644 -0.235601399  0.03640205
## Chrysler Imperial   -0.19787718 -0.287633884 -0.14269466
## Fiat 128             0.05522335  0.028030216 -0.12757239
## Honda Civic         -0.02175283  0.143823376  0.27532709
## Toyota Corolla       0.30968716  0.219781267  0.05650479
## Toyota Corona        0.60261050 -0.486846780  0.03353143
## Dodge Challenger    -0.27527182  0.129929074  0.01915881
## AMC Javelin         -0.10921068  0.323806123 -0.01357013
## Camaro Z28           0.33820115 -0.328980623 -0.10026475
## Pontiac Firebird    -0.16970686  0.100547087  0.18841840
## Fiat X1-9            0.05003063 -0.038255457 -0.03183841
```

```
## Porsche 914-2       -0.53095855 -0.148829855 -0.11360929
## Lotus Europa         0.02697508 -0.195339211  0.09595999
## Ford Pantera L       0.15411589  0.473011110 -0.12940150
## Ferrari Dino        -0.29170961 -0.115742950  0.05335640
## Maserati Bora        0.74155366 -0.045388590  0.09013855
## Volvo 142E           0.04182534 -0.403570812 -0.22108090
```

```
summary(pca_res)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.1921 1.4423 0.69434 0.51480 0.41808 0.33026 0.24457
## Proportion of Variance 0.6007 0.2600 0.06026 0.03313 0.02185 0.01363 0.00748
## Cumulative Proportion  0.6007 0.8607 0.92094 0.95407 0.97591 0.98955 0.99702
##                            PC8
## Standard deviation     0.15428
## Proportion of Variance 0.00298
## Cumulative Proportion  1.00000
```

```
autoplot(pca_res, data = mtcars_new, colour = 'discretempg', label.size = 3)
```



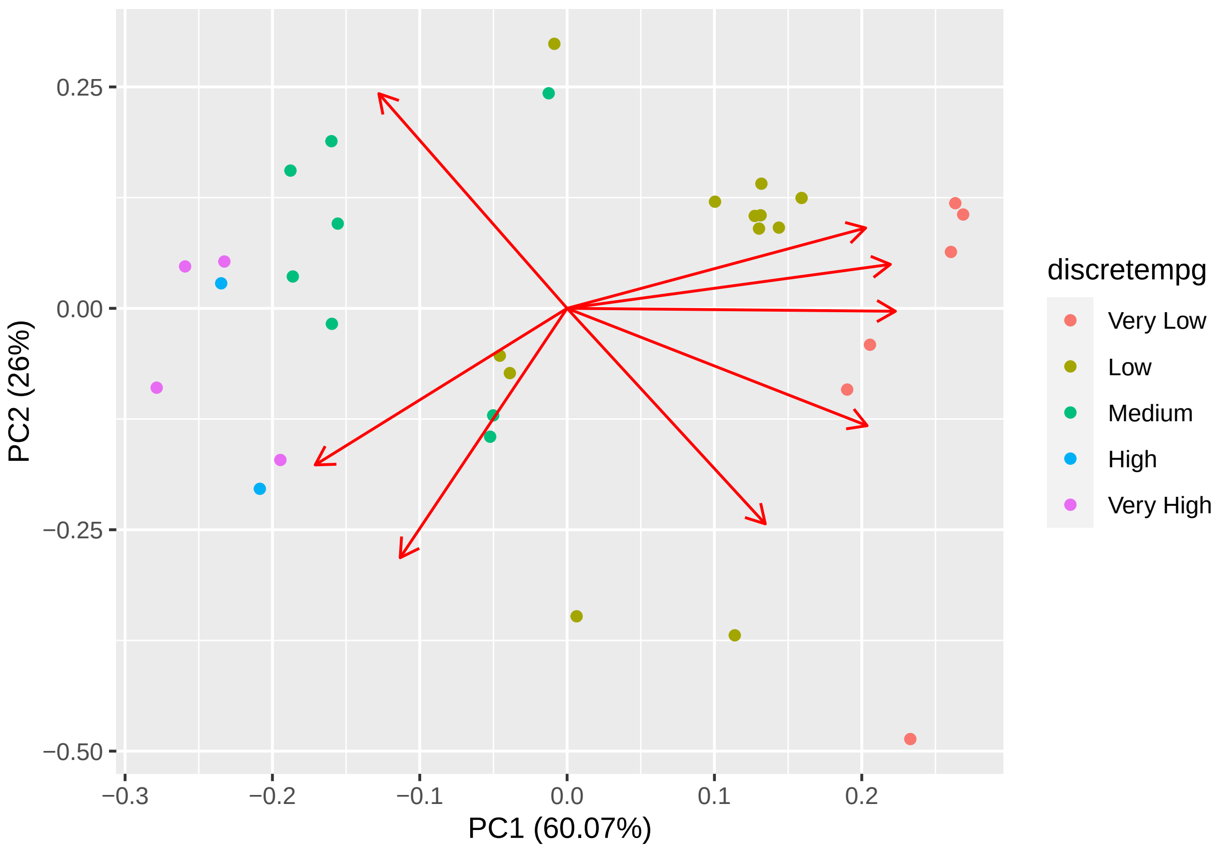
Observations:

The first principle component (PC1) has high values for ‘cyl’, ‘dist’ and ‘hp’, which indicates that this principle component describes the most variation in these variables.

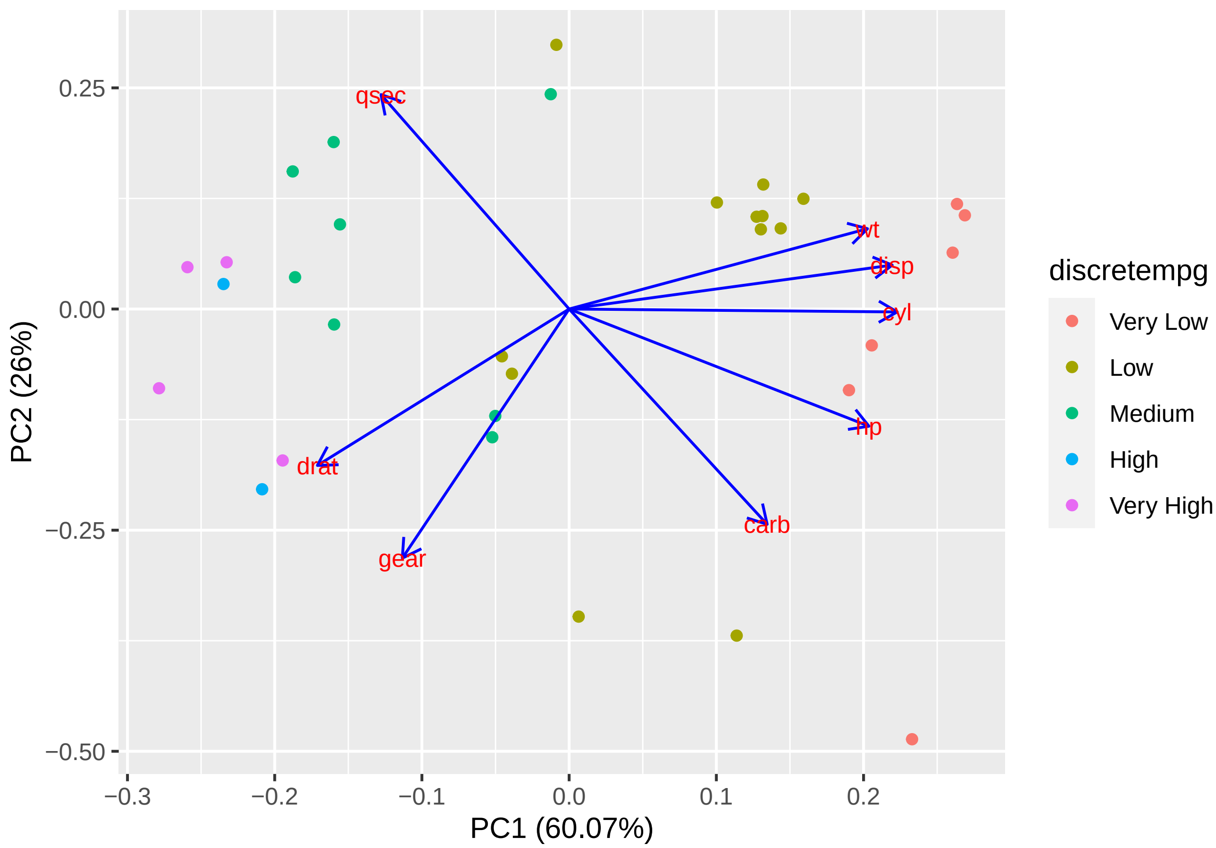
The second principle compoenent (PC2) has high values for 'carb' and 'qsec'. This indicates that this principle component describes the most variation in these variables.

Next I will plot the PCA with eigenvectors:

```r
#plot with no labelling of eigenvectors
autoplot(pca_res, data = mtcars_new, colour = 'discretempg',loadings = TRUE)
```




```r
#plot with labelling of eigenvectors
autoplot(pca_res, data = mtcars_new, colour = 'discretempg',
         loadings = TRUE, loadings.colour = 'blue',
         loadings.label = TRUE, loadings.label.size = 3)
```



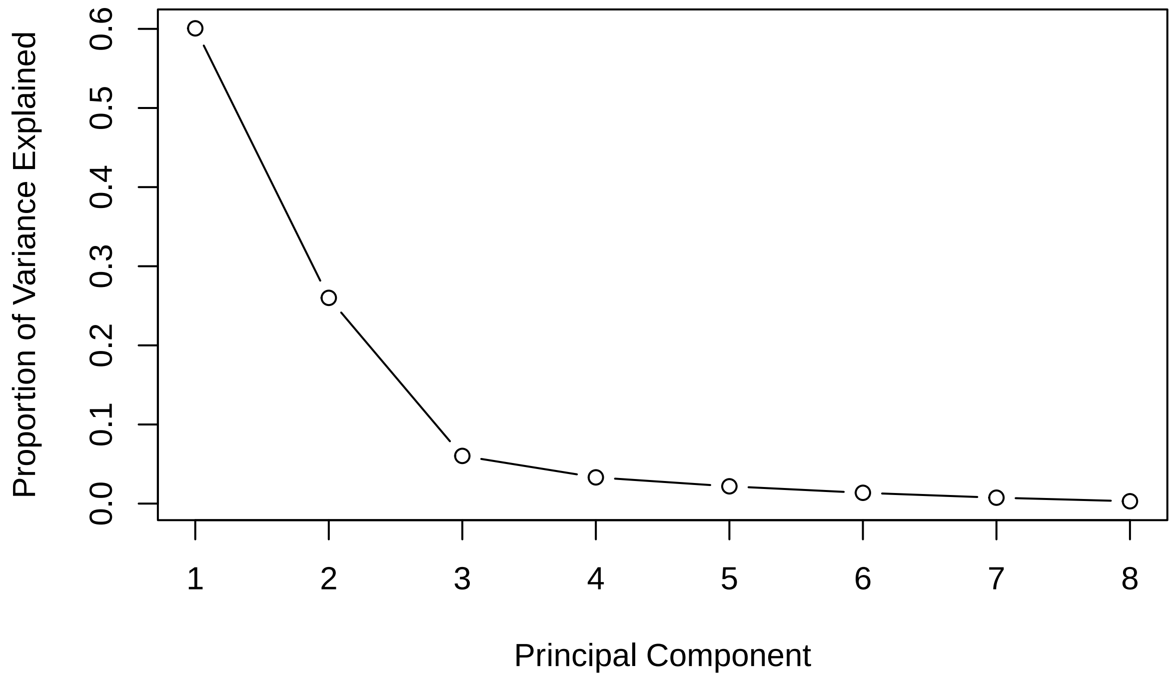
Observations:

From the plot, we can tell that cars that are from the same category of mpg are closely clustered together. For example, cars with "Very High" mpg tend to have high 'qsec', 'drat' and 'gear'. Cars with "Very low" tend to have higer 'wt', 'disp', 'cyl' and 'hp'.

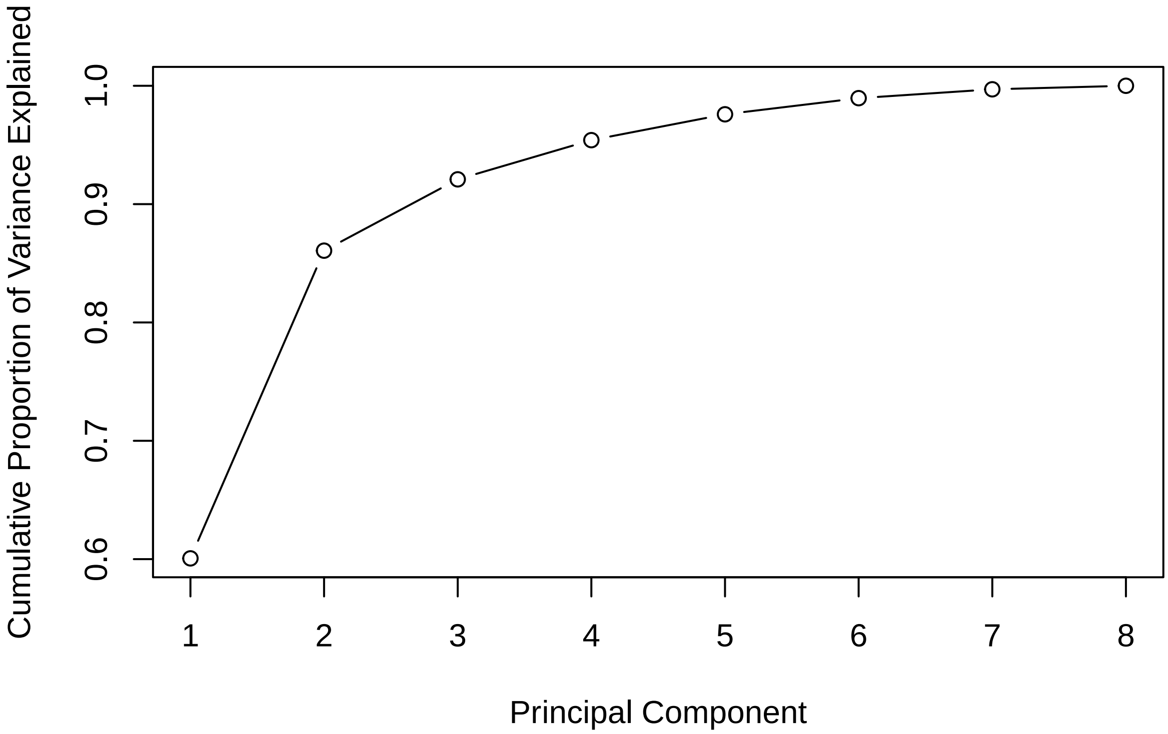
Arrows that are close together indicates high correlation (eg 'wt' and 'disp' are highly correlated).

Next I will plot the cumulative sum of the eigenvalues to determine how many principle components to select for dimensionality reduction:

```
#compute standard deviation of each principal component
 std_dev <- pca_res$sdev

#compute variance
 pr_var <- std_dev^2

 #proportion of variance explained
prop_varex <- pr_var/sum(pr_var)
plot(prop_varex, xlab = "Principal Component",

             ylab = "Proportion of Variance Explained",

             type = "b")
```




```
plot(cumsum(prop_varex), xlab = "Principal Component", ylab = "Cumulative Proportion of Variance Explai
```



From the plot, over 85% of the variance is captured within the 2 largest principle components.