Data 609 - Module5

Amit Kapoor

10/22/2021

Contents

Ex. 1

Carry out the logistic regression (Example 22 on Page 94) in R using the data

x	0.1	0.5	1.0	1.5	2.0	2.5
y	0	0	1	1	1	0

The formula is $y(x) = \frac{1}{1+exp[-(a+bx)]}$

Solution

We will use here `glm` function with family as binomial to perform logistic regression for given values of x and y.

```
x <- c(0.1, 0.5, 1.0, 1.5, 2.0, 2.5)
y <- c(0, 0, 1, 1, 1, 0)

# Logistics Regression
lr <- glm(y~x, family = binomial)
summary(lr)
```

```
##
## Call:
## glm(formula = y ~ x, family = binomial)
##
## Deviance Residuals:
##       1        2        3        4        5        6
## -0.8518  -0.9570   1.2583   1.1075   0.9653  -1.5650
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.8982     1.5811  -0.568    0.570
```

```
## x              0.7099     1.0557   0.672    0.501
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8.3178  on 5  degrees of freedom
## Residual deviance: 7.8325  on 4  degrees of freedom
## AIC: 11.832
##
## Number of Fisher Scoring iterations: 4
```

The null deviance shows how well our model can predict by only using intercept. Residual deviance shows how well our model can predict using intercepts and inputs. AIC is Akaike information criterian. It is used to compare models. All these values if smaller than better.

Ex. 2

Using the motor car database (mtcars) of the built-in data sets in R to carry out the basic principal component analysis and explain your results

Solution

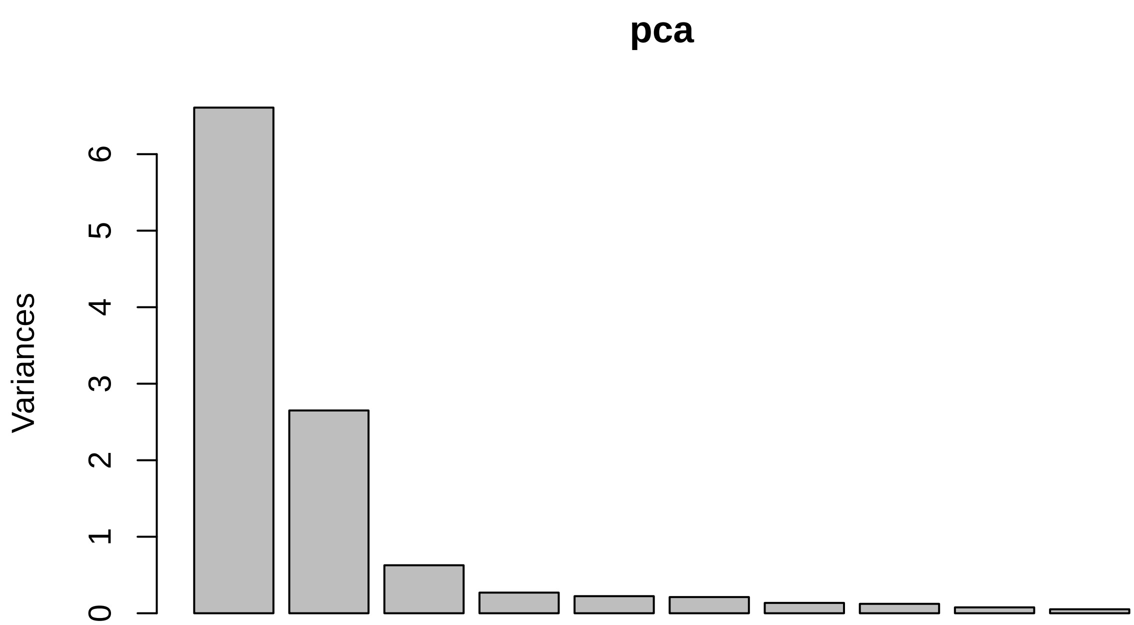
We will use prcomp() to perform principal component analysis on mtcars dataset. The scale argument True implies variables will be scaled to have unit variance.

```
pca <- prcomp(mtcars, scale=TRUE)
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6    PC7
## Standard deviation     2.5707 1.6280 0.79196 0.51923 0.47271 0.46000 0.3678
## Proportion of Variance 0.6008 0.2409 0.05702 0.02451 0.02031 0.01924 0.0123
## Cumulative Proportion  0.6008 0.8417 0.89873 0.92324 0.94356 0.96279 0.9751
##                            PC8    PC9    PC10   PC11
## Standard deviation     0.35057 0.2776 0.22811 0.1485
## Proportion of Variance 0.01117 0.0070 0.00473 0.0020
## Cumulative Proportion  0.98626 0.9933 0.99800 1.0000
```

We can see here the first PCA components shows 60% variation in the data which gets reduced in further components. Drwaing screeplot below confirms the variance decrease.

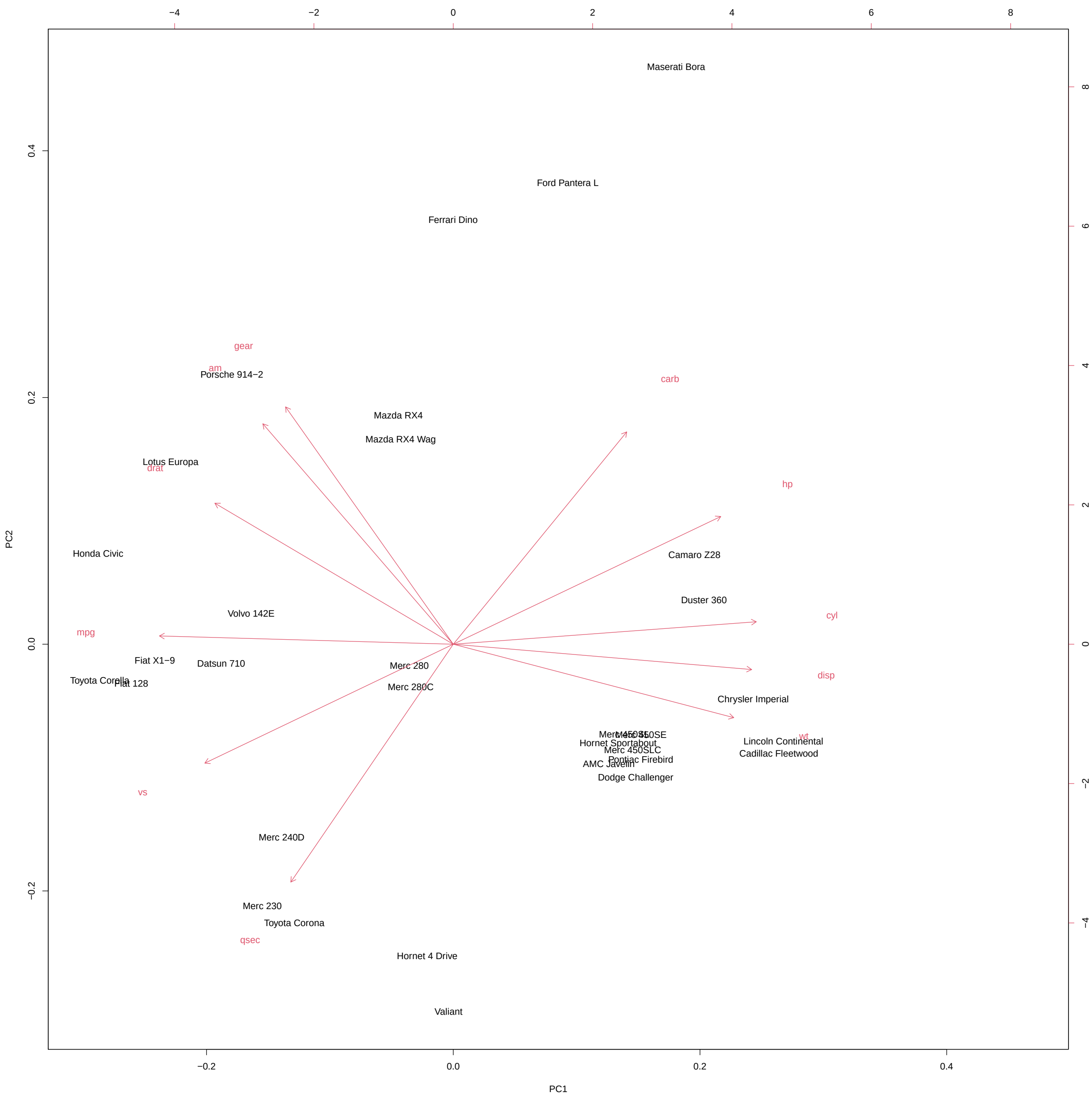
```
screeplot(pca)
```



pca biplot shows the scores of each case and the loading of each variable on the first two principal components. The left and bottom axes shows principal axes scores and top and right axes shows the loadings.

Seeing the biplot below hp, cycl, disp and wt are similar and gear, am, mpg, drat, qsec, and vs are similar based on their pca1 components.

```
biplot(pca)
```



Ex. 3

Generate a random 4 X 5 matrix, and find its singular value decomposition using R.

Solution

```
matrix <- matrix(rnorm(20), nrow = 4)
matrix
```

```
##            [,1]        [,2]        [,3]        [,4]       [,5]
## [1,]  1.5262656 -0.48099627 -0.15703212 -0.23222521 -0.1883491
## [2,] -0.9658266  0.92637533  0.56489534 -1.09648089  0.7375336
## [3,] -1.7231930  0.08176142  0.06757138  0.24192562 -0.8780220
## [4,]  1.1122914 -1.34403045  0.77352112 -0.05935782  0.4402459
```

```
svd(matrix)
```

```
## $d
## [1] 3.0483179 1.7553755 1.2235964 0.5566322
##
## $u
##            [,1]         [,2]       [,3]       [,4]
## [1,] -0.5015625  0.005159429 -0.2978513 -0.8122149
## [2,]  0.4089326 -0.855964590  0.1077413 -0.2974736
## [3,]  0.5350497  0.489171818  0.4729605 -0.5007406
## [4,] -0.5430755 -0.167358697  0.8221829  0.0327925
##
## $v
##             [,1]       [,2]        [,3]        [,4]
## [1,] -0.88131481 -0.1108031 -0.37525031 -0.09521592
## [2,]  0.45721321 -0.3022120 -0.67284819  0.05404782
## [3,] -0.02432842 -0.3308364  0.63384398 -0.08797127
## [4,] -0.05584502  0.6070655  0.01360789  0.70370053
## [5,] -0.10261455 -0.6468457  0.06722461  0.69647640
```

Ex. 4

First try to simulate 100 data points for y using $y = 5x_1 + 2x_2 + 2x_3 + x_4$. where x_1, x_2 are uniformly distributed in [1,2], while x_3, x_4 are normally distributed with zero mean and unit variance. Then, use the principal component analysis (PCA) to analyze the data to find its principal components. Are the results expected from the formula?

Solution

In the first step, we will simulate x1, x2, x3 and x4 and create a dataframe following the given formula.

```
set.seed(609)
x1 <- runif(100, min=1, max=2)
x2 <- runif(100, min=1, max=2)
x3 <- rnorm(100, mean=0, sd=1)
x4 <- rnorm(100, mean=0, sd=1)

y = 5*x1 + 2*x2 + 2*x3 + x4

df <- as.data.frame(cbind(y,x1,x2,x3,x4))
summary(df)
```

```
##        y                x1              x2              x3         
##  Min.   : 1.548   Min.   :1.010   Min.   :1.009   Min.   :-2.0969  
##  1st Qu.: 8.461   1st Qu.:1.271   1st Qu.:1.198   1st Qu.:-0.7122  
##  Median : 9.880   Median :1.482   Median :1.450   Median :-0.1631  
##  Mean   :10.037   Mean   :1.479   Mean   :1.469   Mean   :-0.1081  
##  3rd Qu.:11.914   3rd Qu.:1.699   3rd Qu.:1.757   3rd Qu.: 0.4325  
##  Max.   :15.064   Max.   :1.989   Max.   :1.979   Max.   : 2.1450  
##        x4          
##  Min.   :-2.08701  
##  1st Qu.:-0.85024  
##  Median : 0.01832  
##  Mean   :-0.08108  
##  3rd Qu.: 0.61931  
```

```
##  Max.   : 2.14581
```

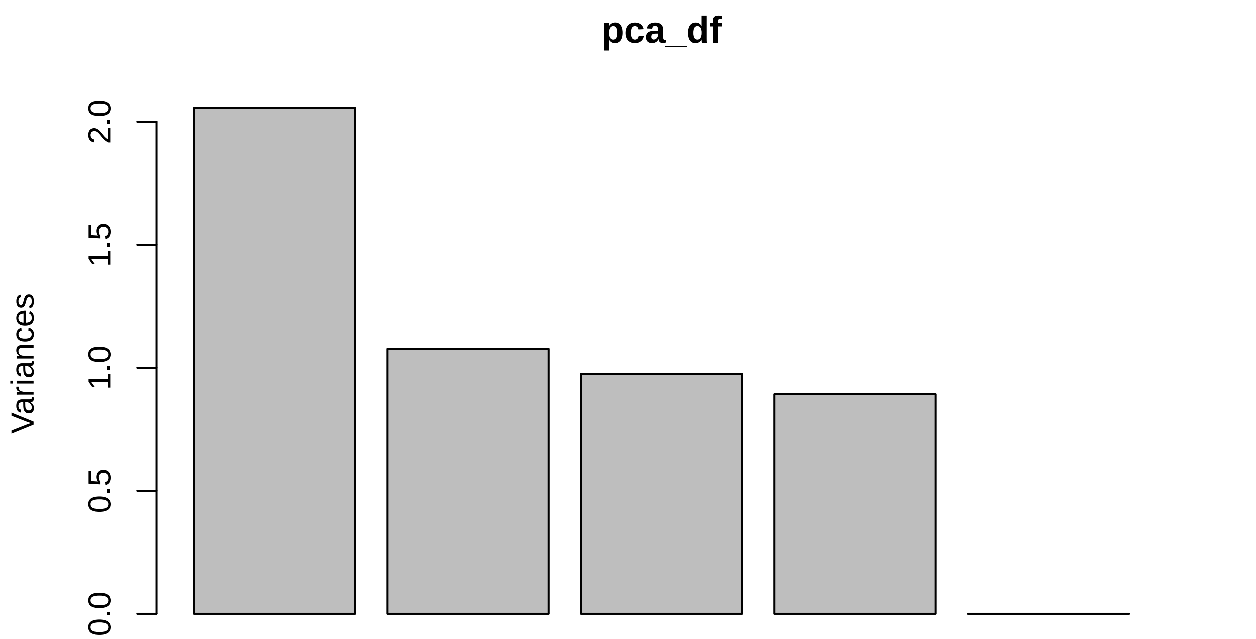
We will now use prcomp() to perform principal component analysis. The scale argument True implies variables will be scaled to have unit variance.

```
pca_df <- prcomp(df, scale=TRUE)
summary(pca_df)
```

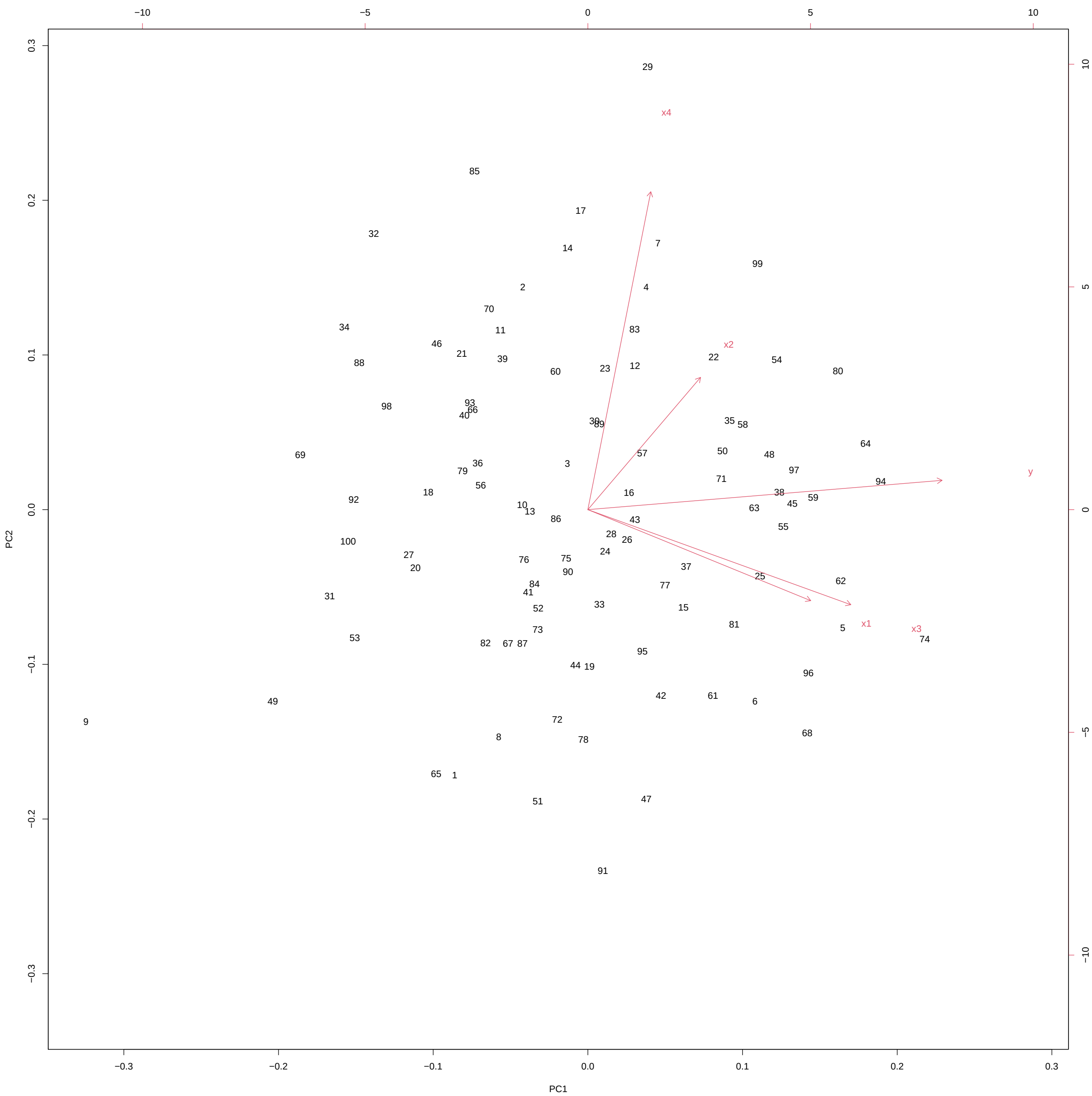
```
## Importance of components:
##                           PC1    PC2    PC3    PC4       PC5
## Standard deviation     1.4339 1.0377 0.9873 0.9447 4.513e-16
## Proportion of Variance 0.4112 0.2154 0.1949 0.1785 0.000e+00
## Cumulative Proportion  0.4112 0.6266 0.8215 1.0000 1.000e+00
```

The scree plot below shows that principal components variance values do not differ much and they cover similar variation in the data points.

```
screeplot(pca_df)
```




```
biplot(pca_df)
```




```
str(pca_df)
```

```
## List of 5
##  $ sdev    : num [1:5] 1.43 1.04 9.87e-01 9.45e-01 4.51e-16
##  $ rotation: num [1:5, 1:5] 0.693 0.436 0.221 0.515 0.123 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:5] "y" "x1" "x2" "x3" ...
##   .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:5] 10.0372 1.4795 1.4685 -0.1081 -0.0811
##   ..- attr(*, "names")= chr [1:5] "y" "x1" "x2" "x3" ...
##  $ scale   : Named num [1:5] 2.529 0.281 0.295 0.873 0.912
##   ..- attr(*, "names")= chr [1:5] "y" "x1" "x2" "x3" ...
##  $ x       : num [1:100, 1:5] -1.234 -0.603 -0.19 0.541 2.363 ...
##   ..- attr(*, "dimnames")=List of 2
```

```
##   .. ..$ : NULL
##   .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

The results above do show what we would expect from formula.