Data 609 Module 5 HW

Bryan Persaud

4/18/2021

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:

```
x <- c(0.1, 0.5, 1.0, 1.5, 2.0, 2.5)
y <- c(0, 0, 1, 1, 1, 0)
glm(y ~ x, family = "binomial")
```

```
##
## Call:  glm(formula = y ~ x, family = "binomial")
##
## Coefficients:
## (Intercept)            x
##     -0.8982       0.7099
##
## Degrees of Freedom: 5 Total (i.e. Null);  4 Residual
## Null Deviance:       8.318
## Residual Deviance: 7.832     AIC: 11.83
```

The logistic regression for the x and y data is seen above.

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:

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

```
str(mtcars_pca)
```

```
## List of 5
##  $ sdev    : num [1:11] 2.571 1.628 0.792 0.519 0.473 ...
##  $ rotation: num [1:11, 1:11] -0.363 0.374 0.368 0.33 -0.294 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:11] "mpg" "cyl" "disp" "hp" ...
##   .. ..$ : chr [1:11] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:11] 20.09 6.19 230.72 146.69 3.6 ...
##   ..- attr(*, "names")= chr [1:11] "mpg" "cyl" "disp" "hp" ...
##  $ scale   : Named num [1:11] 6.027 1.786 123.939 68.563 0.535 ...
##   ..- attr(*, "names")= chr [1:11] "mpg" "cyl" "disp" "hp" ...
##  $ x       : num [1:32, 1:11] -0.647 -0.619 -2.736 -0.307 1.943 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:32] "Mazda RX4" "Mazda RX4 Wag" "Datsun 710" "Hornet 4 Drive" ...
##   .. ..$ : chr [1:11] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

Using R we can see the principal component analysis above. There are 11 different objects shown, one for each column in the mtcars dataset.

Ex.3

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

Solution:

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

```
## $d
## [1] 1.8886669 1.7223506 1.5475680 0.5732437
##
## $u
##              [,1]       [,2]       [,3]        [,4]
## [1,] -0.005504608  0.9442970 -0.1463882  0.29469201
## [2,]  0.359393204 -0.1143761 -0.9222497 -0.08491277
## [3,]  0.537218686  0.2775975  0.2447059 -0.75792789
## [4,]  0.763021826 -0.1347623  0.2610461  0.57575320
##
## $v
```

```
##              [,1]       [,2]       [,3]        [,4]
## [1,] -0.93858237 -0.1993172 -0.1576295  0.23143768
## [2,]  0.31960476 -0.2164943 -0.5057100  0.73403430
## [3,] -0.03292639  0.4674197 -0.6129037 -0.06524236
## [4,]  0.04427639 -0.1207704  0.5284918  0.52186976
## [5,]  0.11777686 -0.8248250 -0.2538815 -0.36197463
```

The single value decomposition can be seen above.

Ex.4

First try to simulate 100 data points for y using $y = 5x_1 + 2x_2 + 2x_3 + x_4$. where x, 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:

```
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
data_frame <- as.data.frame(cbind(y, x1, x2, x3, x4))
data_pca <- prcomp(data_frame, scale = TRUE)
summary(data_pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4       PC5
## Standard deviation     1.4173 1.0559 1.0045 0.9314 4.014e-16
## Proportion of Variance 0.4017 0.2230 0.2018 0.1735 0.000e+00
## Cumulative Proportion  0.4017 0.6247 0.8265 1.0000 1.000e+00
```

```
str(data_pca)
```

```
## List of 5
##  $ sdev    : num [1:5] 1.42 1.06 1.00 9.31e-01 4.01e-16
##  $ rotation: num [1:5, 1:5] -0.701 -0.367 -0.243 -0.552 -0.106 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:5] "y" "x1" "x2" "x3" ...
##   .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:5] 10.5038 1.5206 1.4672 0.0422 -0.1181
##   ..- attr(*, "names")= chr [1:5] "y" "x1" "x2" "x3" ...
##  $ scale   : Named num [1:5] 2.86 0.289 0.293 1.114 0.97
##   ..- attr(*, "names")= chr [1:5] "y" "x1" "x2" "x3" ...
##  $ x       : num [1:100, 1:5] -1.16 -2.24 1.93 1.54 2.09 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

The PCA for the data can be seen above. We see that the results are what you would expect from the formula.