Data 609 HW 5

Avery Davidowitz

2023-04-23

Import

```
library(optimr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

EX 1

Carry out the logistic regression (example 22 on page 94) in R using the data:

```
df <- data.frame(
  x = c(.1, .5, 1, 1.5, 2, 2.5),
  y = c(0, 0, 1, 1, 1, 0)
)

logit <- function(a, b) {
  result <- 1 / (1 + exp(-1*(a + b*df$x)))
  return(result)
}

log_likelihood <- function(par) {
  a <- par[1]
  b <- par[2]
  sum_l <- sum((df$y * log(logit(a, b))) + ((1 - df$y) * log(1 - logit(a, b))))
  return(-1 * sum_l)
}

optimr(par=c(1,1), log_likelihood, control = list(fnscale = -1))
```

```
## $par
```

```
## [1] -0.8979871  0.7097970
##
## $value
## [1] 3.916239
##
## $counts
## function gradient
##       75       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

Checking results against base general linear model function

```
fit <- glm(y ~ x, data = df, family = "binomial")
summary(fit)
```

```
##
## Call:
## glm(formula = y ~ x, family = "binomial", data = df)
##
## 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
```

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.

```
summary(mtcars)
```

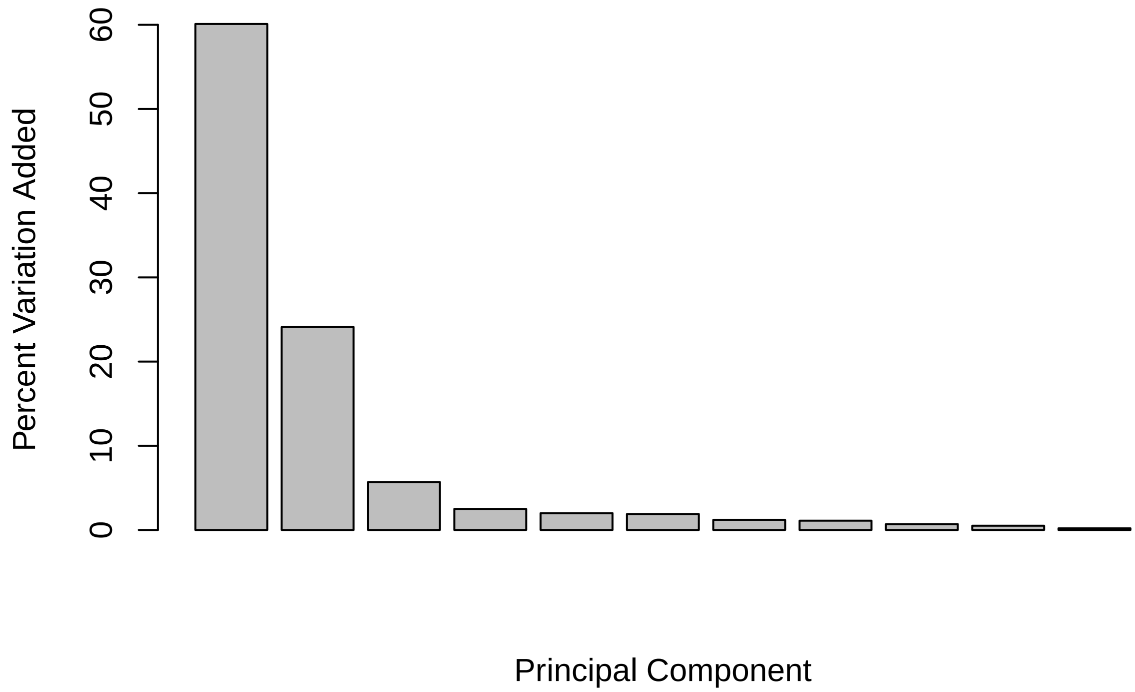
```
##       mpg             cyl             disp             hp
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0
```

```
##  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
```

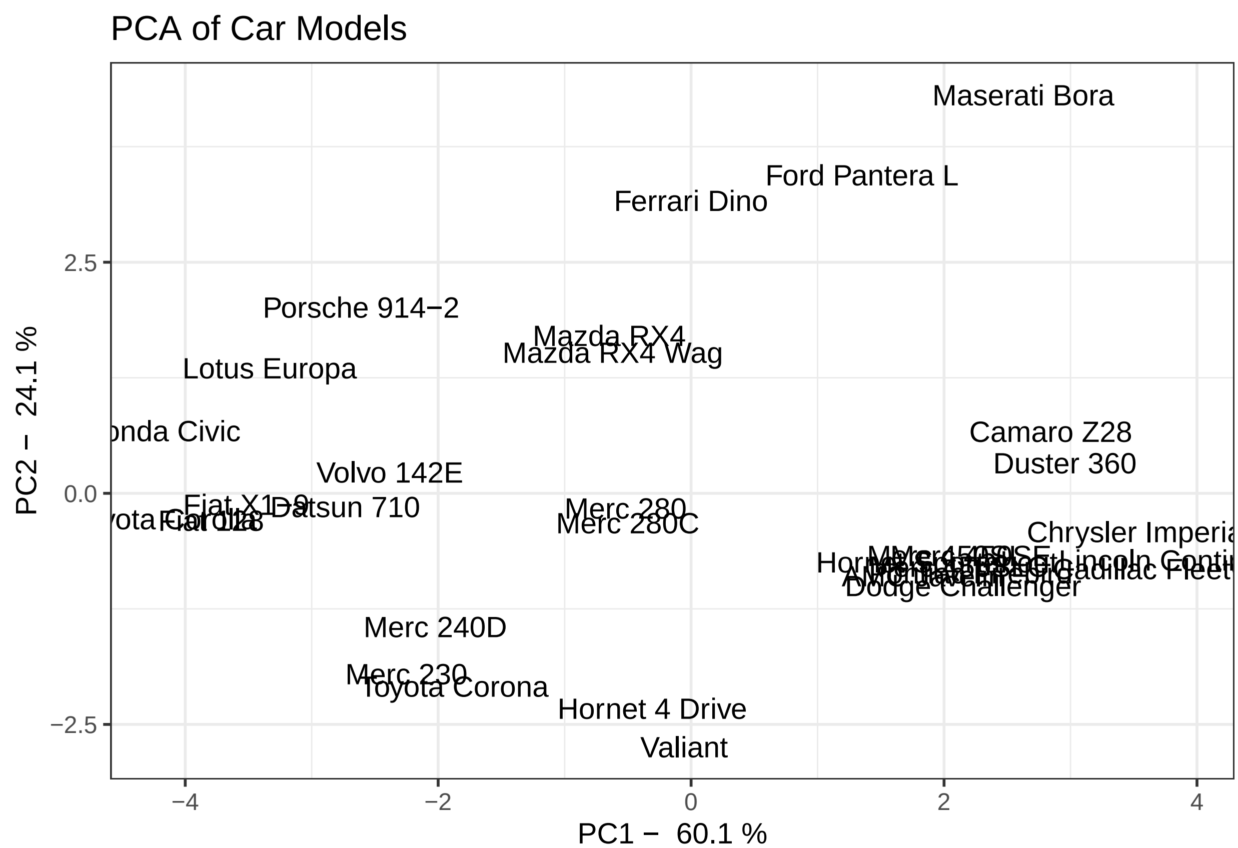
```r
pca <- prcomp(mtcars, scale=TRUE)
pca_var <- pca$sdev^2
pca_car_percent <- round(pca_var/sum(pca_var)*100, 1)
summary(pca)
```

```
## 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
```

```r
barplot(pca_car_percent, xlab="Principal Component", ylab="Percent Variation Added")
```




```
pca_data <- data.frame(car=rownames(pca$x), x=pca$x[,1], y=pca$x[,2])
ggplot(data=pca_data, aes(x=x, y=y, label=car)) +
  geom_text() +
  xlab(paste("PC1 - ", pca_car_percent[1],"%")) +
  ylab(paste("PC2 - ", pca_car_percent[2],"%")) +
  theme_bw() +
  ggtitle("PCA of Car Models")
```



Reducing the dimensionality down to 2 principal components for the mtcars data set still captures 84 % of the total variability in the data set. After the second each component adds significantly less information. Plotting PCA1 and PCA2 shows that we have one very tight cluster of car models that must be very similar to each other.

EX 3

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

```
m <- matrix(sample.int(3, 20, replace=TRUE), nrow = 4, ncol = 5)
m
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    1    2    1    3
## [2,]    2    1    2    1    3
## [3,]    3    1    3    3    2
## [4,]    1    2    3    1    1
```

```
svd_m <- svd(m)
svd_m$d
```

```
## [1] 8.6446140 2.1214682 1.8685721 0.5276937
```

```r
svd_m$u
```

```
##            [,1]       [,2]       [,3]       [,4]
## [1,] -0.4373005  0.5758715 -0.1616656  0.6715687
## [2,] -0.4870838  0.4920929  0.1052575 -0.7138031
## [3,] -0.6306251 -0.5097370  0.5623935  0.1618450
## [4,] -0.4169288 -0.4079029 -0.8040514 -0.1152688
```

```r
svd_m$v
```

```
##            [,1]       [,2]        [,3]       [,4]
## [1,] -0.4303572 -0.1777338  0.49876494 -0.7310519
## [2,] -0.2763417 -0.1214151 -0.58981802 -0.2102113
## [3,] -0.5774035 -0.2908320 -0.44835848  0.1047189
## [4,] -0.3740119 -0.4096924  0.44243452  0.6216327
## [5,] -0.5149255  0.8373995  0.08108392  0.1548584
```

EX 4

First try to simulate 100 data points for y using:

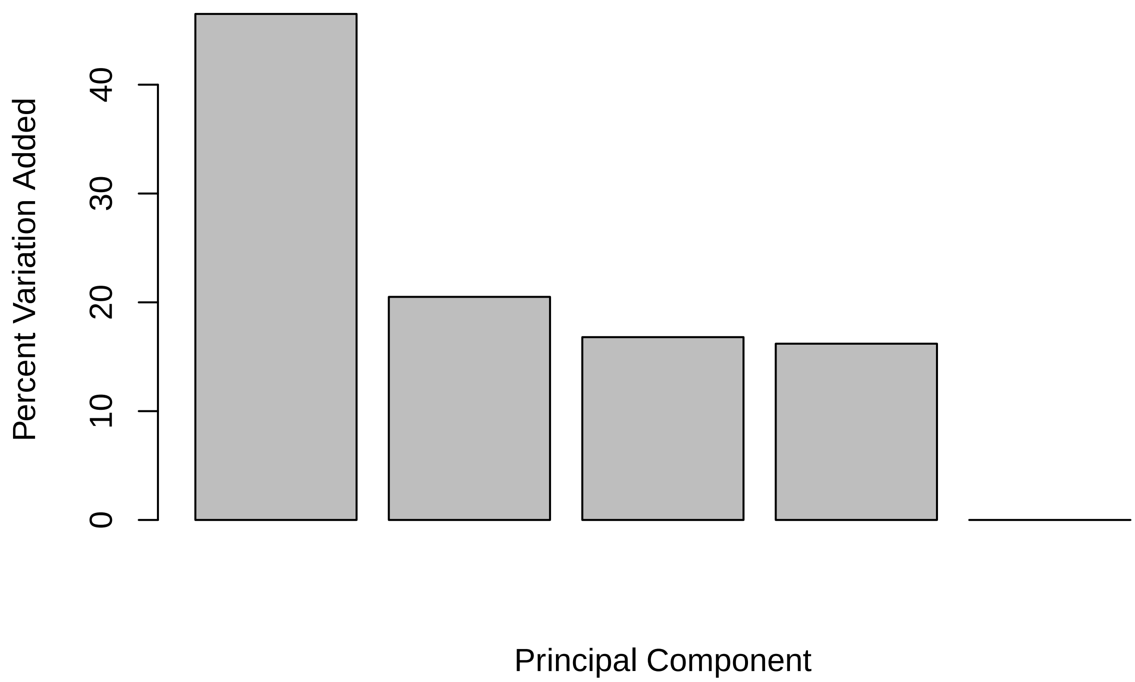
$$y = 5x_1 + 2x_2 + 2x_3 + x_4$$

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.

```r
x1 <- runif(n = 100, min = 1, max = 2)
x2 <- runif(n = 100, min = 1, max = 2)
x3 <- rnorm(n = 100, mean = 0, sd = 1)
x4 <- rnorm(n = 100, mean = 0, sd = 1)
y <- (5*x1 + 2*x2 + 2*x3 + x4)
df <- data.frame(y , x1, x2, x3, x4)
pca <- prcomp(df, scale=TRUE)
pca_var <- pca$sdev^2
pca_y_percent <- round(pca_var/sum(pca_var)*100, 1)
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4       PC5
## Standard deviation     1.5242 1.0129 0.9169 0.9001 5.232e-16
## Proportion of Variance 0.4646 0.2052 0.1682 0.1620 0.000e+00
## Cumulative Proportion  0.4646 0.6698 0.8380 1.0000 1.000e+00
```

```r
barplot(pca_y_percent, xlab="Principal Component", ylab="Percent Variation Added")
```



I'm not sure how to interpret these components. However, it does make sense that the 5th component for y adds no information because it is a linear combination of the Xs.