Matrices and vectors for regression with R

Peter Tempfli

2/13/2019

1a

```
m1 <- cbind(c(1,1,1,1), c(3,4,5,6), c(5,5,3,1))
m1
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    1    4    5
## [3,]    1    5    3
## [4,]    1    6    1
```

```
m2 <- rbind(c(1,3,5), c(1,4,5), c(1,5,3), c(1,6,1))
m2
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    1    4    5
## [3,]    1    5    3
## [4,]    1    6    1
```

```
m3 <- matrix(c(1,1,1,1,3,4,5,6,5,5,3,1), ncol=3)
m3
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    1    4    5
## [3,]    1    5    3
## [4,]    1    6    1
```

1b

```
X = m1
XX <- t(X) %*% X
XX
```

```
##      [,1] [,2] [,3]
## [1,]    4   18   14
## [2,]   18   86   56
## [3,]   14   56   60
```

1c

```
XX_i <- solve(XX)
XX_i
```

```
##            [,1]       [,2]       [,3]
## [1,]  84.333333 -12.333333 -8.1666667
## [2,] -12.333333   1.833333  1.1666667
## [3,]  -8.166667   1.166667  0.8333333
```

```
round(XX %*% XX_i)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

2

```
Y <- matrix(c(1,2,3,5))
```

2a What is the sample size, n?

There are 4 samples (4 rows)

2b How many are the regressors, K?

K = 2. (3 columns, but 1st is the intercept)

2c Compute the least squares estimator of β

```
B_matrix <- solve(t(X) %*% X) %*% t(X) %*% Y
B_matrix
```

```
##            [,1]
## [1,]  0.1666667
## [2,]  0.8333333
## [3,] -0.3333333
```

Check the betas with the lm function:

```
df <- data.frame(X)
names(df) <- c('intercept1', 'first', 'second')
control_mod <- lm(Y ~ df$first + df$second)

control_mod
```

```
##
## Call:
## lm(formula = Y ~ df$first + df$second)
##
## Coefficients:
## (Intercept)     df$first    df$second
##      0.1667       0.8333      -0.3333
```

```
B_matrix
```

```
##             [,1]
## [1,]  0.1666667
## [2,]  0.8333333
## [3,] -0.3333333
```

2d T-test

```
beta_h = B_matrix

y_h <- X%*%beta_h

U_h <- Y - y_h

n <- (length(Y))

K <- ncol(X) - 1

s_2 <- sum(U_h^2)/(n-K-1)

SE <- sqrt(s_2*diag(solve(t(X)%*%X)))

t_0 <- beta_h/SE
t_0
```

```
##             [,1]
## [1,]  0.04445542
## [2,]  1.50755672
## [3,] -0.89442719
```

2e P-value

```
2*pt(abs(t_0),df=(n-K-1),lower.tail=F)
```

```
##           [,1]
## [1,] 0.9717174
## [2,] 0.3728590
## [3,] 0.5354409
```

This is a huge P-value, bc there are not enough datapoints.

2f

```
summary(control_mod)
```

```
##
## Call:
## lm(formula = Y ~ df$first + df$second)
##
## Residuals:
##          1          2          3          4
## -1.388e-17  1.667e-01 -3.333e-01  1.667e-01
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1667     3.7491   0.044    0.972
## df$first      0.8333     0.5528   1.508    0.373
## df$second    -0.3333     0.3727  -0.894    0.535
##
## Residual standard error: 0.4082 on 1 degrees of freedom
## Multiple R-squared:  0.981,  Adjusted R-squared:  0.9429
## F-statistic: 25.75 on 2 and 1 DF,  p-value: 0.138
```