PS #3 Question 3

David Zynda

February 5, 2019

Previous Work

```
## Problem 14

# (a) Generate Data

x1 = rnorm(1000, 0, 1)
x2 = rnorm(1000, 0, 1)
e = rnorm(1000, 0, 1)

y = 1 + x1 + x2 + e

x1 = as.matrix(x1)
x2 = as.matrix(x2)
e = as.matrix(e)

y = as.matrix(y)


my_lm = lm(y ~ x1 + x2)
summary(my_lm)


# (b) compute residuals


t(my_lm$residuals) %*% x1
t(my_lm$residuals) %*% x2

## (c) verify true residuals are not orthoganol

t(e) %*% x1
t(e) %*% x2

## (d) Auxiliary Regression

aux = lm(x1 ~ x2)

solve(t(aux$residuals) %*% aux$residuals) %*% (t(aux$residuals) %*% y)


## What is E(x'x)

t(x1) %*% x1
```

T-test

Write a simulation program showing the t-test of the coefficient on x_{1i} being 0 performs as it is derived.

The simulation below will give an actual parameter value of 0 for x_1 and 1 for x_2.

```
x0 = rep(1, 1000)
x1 = rnorm(1000, 0, 1)
x2 = rnorm(1000, 0, 1)
e = rnorm(1000, 0, 1)

y = x0 + 0*x1 + x2 + e

X = as.matrix(cbind(x0, x1, x2))

beta = solve((t(X) %*% X)) %*% (t(X) %*% y)

y_hat = X %*% beta

res = y - y_hat
res_sq = res ** 2
stdev = sum(res_sq) / 998


## Display beta estimates: first row slope, second row estimate for beta1

beta
```

```
##           [,1]
## x0 1.011004854
## x1 0.008140332
## x2 0.956342656
```

Compare this with the `lm()` package in `R`:

```
summary(lm(y ~ x1 + x2))
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4558 -0.6965  0.0460  0.6763  4.0196
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.01100    0.03242  31.180   <2e-16 ***
## x1           0.00814    0.03229   0.252    0.801
## x2           0.95634    0.03252  29.409   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.025 on 997 degrees of freedom
## Multiple R-squared:  0.4649, Adjusted R-squared:  0.4639
```

```
## F-statistic: 433.1 on 2 and 997 DF,  p-value: < 2.2e-16
```

Lastly, compute t-stat of x_1. The null hypothesis, of course, is that $\beta_1 = 0$.

```
std_error_beta = stdev * sqrt(1000 / (1000 * t(x1) %*% x1 - sum(x1)^2)) # page 12 http://www.stat.cmu.e

t = (beta[2] - 0) / std_error_beta
t
```

```
##            [,1]
## [1,] 0.2463716
```

The t statistic is `0.2463716`

F-test

Generate Simulation

```
x1 = rnorm(100, 0, 1)
x2 = rnorm(100, 0, 1)
e = rnorm(100, 0, 1)

y =  0 * x1 + 0 * x2 + e
```

```
X = as.matrix(cbind(x1, x2))
beta = solve(t(X) %*% X) %*% t(X) %*% y

y_hat = X %*% beta
res = y - y_hat
res2 = y - 0

sse_r = t(res2)%*%res2
sse_u = t(res)%*%res

Fstat = ( ( (sse_r - sse_u) / (100 - 98) ) /  (sse_u / 100))
```

The F-stat is `0.4475703`. Hence, it is not signficant.

Significance at the 95% level:

```
qf(.95, df1=100, df2=98)
```

```
## [1] 1.394535
```