STATS 305A H.W.# 4

Adhitya Venkatesh

October 23, 2017

Computational Problems:

1.)

a.), b.)

```
library('faraway')
data_prost = faraway::prostate

lin_mod = lm(lpsa ~ ., data = data_prost)
F_0 = as.numeric(summary(lin_mod)$fstatistic[1])
t_0 = summary(lin_mod)$coef[4,3]
x = data_prost[,1:8]


count_F = 0
count_t = 0

for(i in 1:1000){ #1000 was chosen as 97! is computationally infeasible

  perm_data = x[sample(nrow(x)),]
  perm_data$lpsa = data_prost$lpsa
  test_mod = lm(lpsa ~ ., data = perm_data)
  F_test = as.numeric(summary(test_mod)$fstatistic[1])
  t_test = summary(test_mod)$coef[4,3]

  if(F_test > F_0) {
      count_F = count_F + 1
  }
  if(t_test > t_0){
      count_t = count_t + 1
  }

}

print("Answer to part A:")
```

```
## [1] "Answer to part A:"
```

```
print(count_F/1000)
```

```
## [1] 0
```

```
print("Answer to part B:")
```

```
## [1] "Answer to part B:"
```

```
print(count_t/1000)
```



```
## [1] 0.956
```

c.)

```r
library(car)
```

```
## Warning: package 'car' was built under R version 3.3.2
```

```
##
## Attaching package: 'car'
```

```
## The following objects are masked from 'package:faraway':
##
##     logit, vif
```

```r
data_boot = Boot(lin_mod, R = 1000)
```

```
## Loading required namespace: boot
```

```r
conf_intervals = confint(data_boot, level = 0.95)
Age_lowerBound = conf_intervals[4,1]
Age_upperBound = conf_intervals[4,2]

print(Age_lowerBound)
```

```
## [1] -0.03992343
```

```r
print(Age_upperBound)
```

```
## [1] 0.001449159
```

```r
summary(lin_mod)
```

```
##
## Call:
## lm(formula = lpsa ~ ., data = data_prost)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7331 -0.3713 -0.0170  0.4141  1.6381
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.669337   1.296387   0.516  0.60693
## lcavol       0.587022   0.087920   6.677 2.11e-09 ***
## lweight      0.454467   0.170012   2.673  0.00896 **
## age         -0.019637   0.011173  -1.758  0.08229 .
## lbph         0.107054   0.058449   1.832  0.07040 .
## svi          0.766157   0.244309   3.136  0.00233 **
## lcp         -0.105474   0.091013  -1.159  0.24964
## gleason      0.045142   0.157465   0.287  0.77503
## pgg45        0.004525   0.004421   1.024  0.30886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7084 on 88 degrees of freedom
## Multiple R-squared:  0.6548, Adjusted R-squared:  0.6234
## F-statistic: 20.86 on 8 and 88 DF,  p-value: < 2.2e-16
```



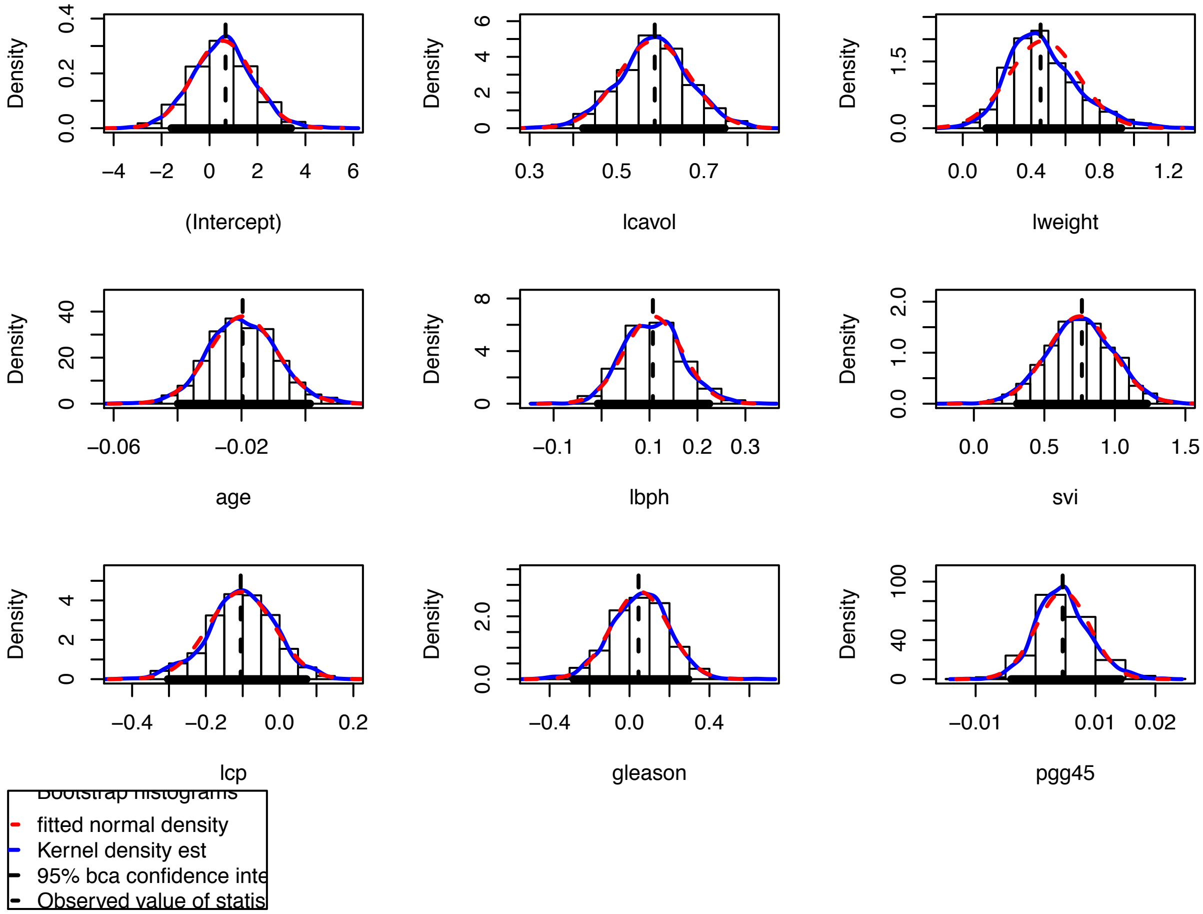
```r
print(t_0) #t statistic for age
```

```
## [1] -1.757599
```

We deduce from the model summary that "age" is not statistically significant at 95% confidence level, which is reinforced by the fact that 0 is contained in the bootstrapped 95% confidence interval.

d.)

```r
hist(data_boot, legend = "separate")
```



e.)

```r
sub_lin_mod = lm(lpsa ~ lcavol + lweight + svi, data = data_prost)
summary(sub_lin_mod)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + svi, data = data_prost)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.72964 -0.45764  0.02812  0.46403  1.57013
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.26809    0.54350  -0.493  0.62298
## lcavol       0.55164    0.07467   7.388  6.3e-11 ***
## lweight      0.50854    0.15017   3.386  0.00104 **
```

```
## svi          0.66616    0.20978   3.176  0.00203 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7168 on 93 degrees of freedom
## Multiple R-squared:  0.6264, Adjusted R-squared:  0.6144
## F-statistic: 51.99 on 3 and 93 DF,  p-value: < 2.2e-16
```

Comparing the summaries of the full model to the sub-model, it's apparent that the former is superior because the fit yields a higher $ R^2 $ value.

f.)

```r
test_data = data.frame(lcavol = 1.44692, lweight = 3.62301, age = 64, lbph = 0.30010, svi = 0, lcp = -0

predict(lin_mod, test_data, interval = "predict")
```

```
##       fit       lwr      upr
## 1 2.40869 0.9842826 3.833097
```

g.)

```r
test_data_2 = data.frame(lcavol = 1.44692, lweight = 3.62301, age = 20, lbph = 0.30010, svi = 0, lcp =

predict(lin_mod, test_data_2, interval = "predict")
```

```
##        fit      lwr      upr
## 1 3.272726 1.538744 5.006707
```

The interval is longer when age $= 20$ mathematically because the variance contribution of the age to the overall variance of the prediction is increased. This is likely due to the fact that compared to age 64 (all other covariates held constant), at age 20 there is a lot more uncertainty of lpsa levels given the human body undergoes far more physiological change than at age 64, when disease onset is far more predictable.

h.)

```r
test_data_3 = data.frame(lcavol = 1.44692, lweight = 3.62301, svi = 0)
predict(sub_lin_mod, test_data_3, interval = "predict")
```

```
##        fit       lwr      upr
## 1 2.372534 0.9383436 3.806724
```

The intervals are narrower for the prediction from the sub-model. However, I would prefer the prediction from the former model as despire age not being statistically significant at the 5% level, it's clear from the significant prediction gap between f.) and g.) that age noticeably alters the difference as it is still significant at the 10% level.

2.)

a.)

The p-value of each test k becomes $p_k/m = (1 - \alpha_k)/m$. Thus, FWER $= \delta =$ probability of getting at least one test incorrect

$$= 1 - P(E)$$

where E is the event that none of the tests are wrong. Proceeding in a similar fashion to the lecture notes section 6.5, we obtain

$$
\begin{aligned}
\delta &\geq 1 - \sum_{k=1}^{m} (1-\alpha_k)/m \\
&= 1 - 1 + \sum_{k=1}^{m} \alpha_k/m
\end{aligned}
$$

Finally as each $\alpha_k \leq \alpha$,

$$
\leq \sum_{k=1}^{m} \alpha/m = \alpha
$$

b.)

We start by defining a generating function for a given correlation:

```
correlatedGenerator = function(x, r){
  e  = rnorm(length(x), mean=0, sd=sqrt(1-r**2))
  y  = r*x + e
  return(y)
}
```

```
#We will proceed by fitting a linear model between variables w and z
#where z is generated by feeding in w and correlation vector into above function.

w = rnorm(1000)
y = correlatedGenerator(w, 0.5)

bonf_mod_0 = lm(y~w)
t_0 = summary(bonf_mod_0)$coef[,3][2]

r_vals = seq(0, 1, length.out = 20)
m_vals = seq(10, 1000, length.out = 50)

vec_m = 0  #for the case m varies
vec_r = 0  #for the case r varies

for (i in 1:50) {
  count_m = 0
  for (k in 1:m_vals[i]) {
    y_gen = correlatedGenerator(w, 0.5)
    bonf_mod_test = lm(y_gen ~ w)
    t_test = summary(bonf_mod_test)$coef[,3][2]
    if(t_test > t_0){
      count_m = count_m + 1
    }
  }
 vec_m = c(vec_m, count_m/50)
}

for (i in 1:20) {
  count_r = 0
  for (k in 1:100) {
    y_gen = correlatedGenerator(w, r_vals[i])
    bonf_mod_test = lm(y_gen ~ w)
```

```
    t_test = summary(bonf_mod_test)$coef[,3][2]
    if(t_test > t_0){
      count_r = count_r + 1
    }

  }
vec_r = c(vec_r, count_r/50)
}
```

We see that in general increasing both the correlation and the number of tests (with the other held constant) leads to increased conservativeness of test (both vec_r and vec_m decrease).

2. H.W#4 STATS 305A : Theoretical

$$\mathrm{Cov}\left(\sum_{i=1}^{I}\sum_{j=1}^{J}(\bar{\varepsilon}_{i\cdot}-\bar{\varepsilon}_{\cdot\cdot})^2, \sum_{i=1}^{I}\sum_{j=1}^{J}(\bar{\varepsilon}_{\cdot j}-\bar{\varepsilon}_{\cdot\cdot})^2\right)$$

$$= \mathrm{Cov}\left(J\sum_{i=1}^{I}(\bar{\varepsilon}_{i}-\bar{\varepsilon}_{\cdot\cdot})^2, I\sum_{j=1}^{J}(\bar{\varepsilon}_{\cdot j}-\bar{\varepsilon}_{\cdot\cdot})^2\right)$$

$$= IJ\sum_{i=1}^{I}\sum_{j=1}^{J}\underbrace{\mathrm{Cov}\left((\bar{\varepsilon}_{i\cdot}-\bar{\varepsilon}_{\cdot\cdot})^2, (\bar{\varepsilon}_{\cdot j}-\bar{\varepsilon}_{\cdot\cdot})^2\right)}$$

$$= \mathrm{Cov}\left((\bar{\varepsilon}_{i\cdot}^2-2\bar{\varepsilon}_{\cdot\cdot}\bar{\varepsilon}_{i\cdot}+\bar{\varepsilon}_{\cdot\cdot}^2), (\bar{\varepsilon}_{\cdot j}^2-2\bar{\varepsilon}_{\cdot\cdot}\bar{\varepsilon}_{\cdot j}+\bar{\varepsilon}_{\cdot\cdot}^2)\right)$$

$$= \mathrm{Cov}(\bar{\varepsilon}_{i\cdot}^2, \bar{\varepsilon}_{\cdot j}^2) + \mathrm{Cov}(\bar{\varepsilon}_{i\cdot}^2, \bar{\varepsilon}_{\cdot\cdot}^2) +$$

$$4\,\mathrm{Cov}(\bar{\varepsilon}_{\cdot\cdot}\bar{\varepsilon}_{i\cdot}, \bar{\varepsilon}_{\cdot j}\bar{\varepsilon}_{\cdot\cdot}) + \mathrm{Cov}(\bar{\varepsilon}_{\cdot\cdot}^2, \bar{\varepsilon}_{\cdot j}^2)$$

$$+ \mathrm{Cov}(\bar{\varepsilon}_{\cdot j}^2, \bar{\varepsilon}_{\cdot\cdot}^2) - 2\,\mathrm{cov}(\bar{\varepsilon}_{i\cdot}^2, \bar{\varepsilon}_{i\cdot}\bar{\varepsilon}_{\cdot j})$$

$$-2\,\mathrm{Cov}(\bar{\varepsilon}_{\cdot j}^2, \bar{\varepsilon}_{i}\bar{\varepsilon}_{\cdot j}) + \mathrm{Cov}(\bar{\varepsilon}_{\cdot\cdot}^2, \bar{\varepsilon}_{i\cdot}\bar{\varepsilon}_{\cdot j})$$

Using the fact that

$$E(\varepsilon_{ij}\varepsilon_{kl}) = 0 \quad \forall (i,j) \neq (k,l)$$

$E(\varepsilon_{0j}^2) = \sigma^2$, $E(\varepsilon_{ij}) = 0$, we have

$$= IJ \sum_{i=1}^{I} \sum_{j=1}^{J} (0) = 0$$

$\therefore$ $\sum_i \sum_j (\bar{\varepsilon}_{i\cdot} - \bar{\varepsilon}_{\cdot\cdot})^2$ statistically independent of $\sum_i \sum_j (\bar{\varepsilon}_{\cdot j} - \bar{\varepsilon}_{\cdot\cdot})^2$ $\because$ covariance between them is 0