HomeWork-2 of IsYe 6414, Submitted by Sudhanshu Raj Singh

Homework-2

Chapter 3 Exercise 1

a) The next chunk contains the code for calling the lm function to fit a linear model with lpsa as response and the other variables as predictors and summarising the model object returned by the lm function. As can be seen in the summary, the standard error for the age parameter is 0.011173, the confidence interval for the parameter associated with age is: -0.01967+c(-1,1)*qt(100-(100-Confidence Level)/2,df=97-9)*0.01173. Using this formula, I go the confidence intervals as: 95% confidence interval= (-0.04183,0.0026) 90% confidence interval= (-0.03820,-0.0010)

Analyzing the 90% confidence interval, it does not contain 0 in it and hence based on this, the hypothesis that age has no impact on the response can be rejected.

Analyzing the 95% confidence interval, it does contain 0 in it and hence based on this, the hypothesis that age has no impact on the responce cannot be rejected. We will conclude that Age indeed has no impact on our response.

It in also evident in the p-values in summary of the lmod object. For age, the p-value is 0.08229. We will reject the null hypothesis if the significance level is 10% but not reject it if the significance level is 5%.

```
rm(list=ls())

library("faraway")
```

```
## Warning: package 'faraway' was built under R version 3.4.4
```

```
data(prostate,package="faraway")
lmod<-lm(lpsa ~ .,data=(prostate))
summary(lmod)
```

```
##
## Call:
## lm(formula = lpsa ~ ., data = (prostate))
##
## 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
```

```
conf_95<--0.019627+c(-1,1)*qt(0.975,97-9)*0.011173
conf_90<--0.019627+c(-1,1)*qt(0.95,97-9)*0.011173
conf_95
```

```
## [1] -0.04183099  0.00257699
```

```
conf_90
```

```
## [1] -0.038200482 -0.001053518
```

b) The next chunk contains the code for generating an ellipse based on joint hypothesis testing for age and lbph. Origin is also plotted. As is clearly visible, origin lies insise the ellipse which means that the null hypothesis that Beta(age) = Beta(lbph) = 0 is not rejected because the origin does lie inside the ellipse. Thus the hypothesis that age and lbph do not affect the response jointly stands.

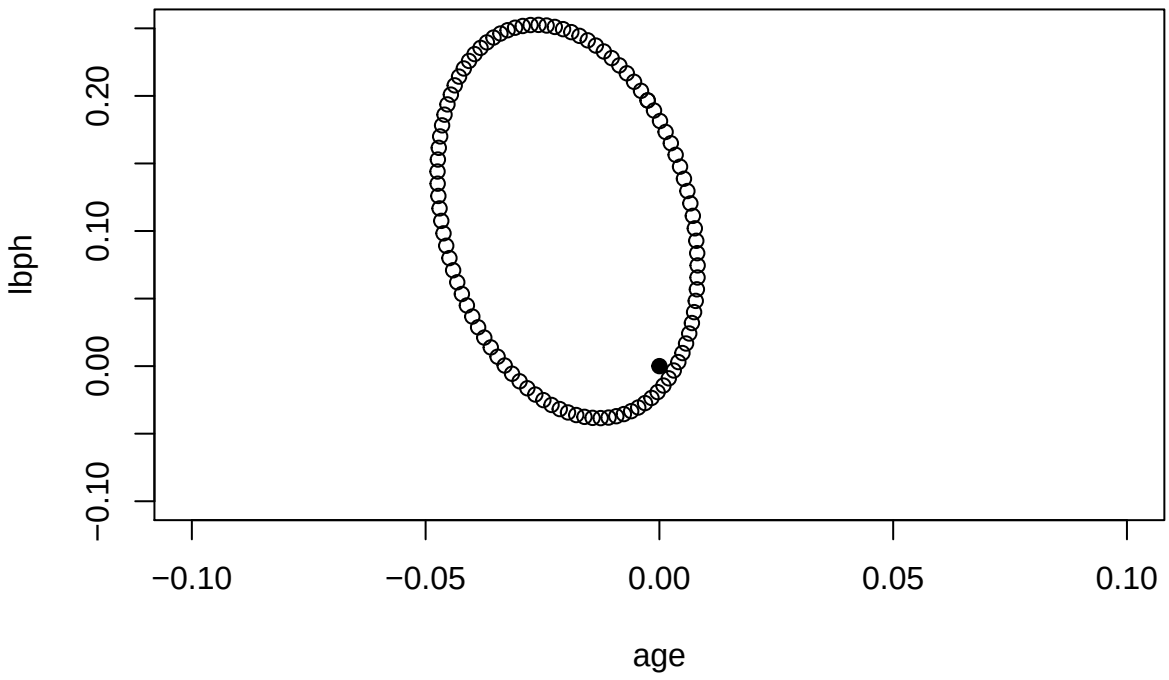
```
library("ellipse")
```

```
## Warning: package 'ellipse' was built under R version 3.4.4
```

```
##
## Attaching package: 'ellipse'
```

```
## The following object is masked from 'package:graphics':
##
##     pairs
```

```
plot(ellipse(lmod,c(4,5)),ylim=c(-0.1,0.25),xlim=c(-0.1,0.1))
points(x=0,y=0,pch=19)
```



c) The next chunk contains the code for conducting a permutation test corresponding to the t-test for age in the model.It is found that the value is very close to the theoretically calculated value.

```
nreps<-4000
set.seed(123)
tstats<-numeric(nreps)
for (i in 1:nreps)
{
  lmods<-lm(lpsa~lcavol+lweight+sample(age)+lbph+svi+lcp+gleason+pgg45,data=prostate)
  tstats[i]<-summary(lmods)$coef[4,3]
}
p=mean(abs(tstats)>1.758)#abs(summary(lmod)$coef[4,3]))
sprintf("The value of t calculated on basis of permutation test is %s",p)
```

```
## [1] "The value of t calculated on basis of permutation test is 0.08275"
```

d) only three variables will remain as can be seen from summary: lcavol,lweight and svi which have p-values<0.05. The next chunk contains the code for testing the two models according to F-test. One, the smaller one, containing only significant variables: lcavol, lweight and svi while the second one contains all the variable. As can be seen by calling the anova function, the p-value is 0.21,which is way bigger than the significance level of 0.05 meaning that our NULL hypothesis that smaller model is better cannot be rejected. And we will conclude that the smaller model is better than the larger one.

```
lmodel<-lm(lpsa~lcavol+lweight+svi,data=prostate)
anova(lmodel,lmod)
```

```
## Analysis of Variance Table
##
```

```
## Model 1: lpsa ~ lcavol + lweight + svi
## Model 2: lpsa ~ lcavol + lweight + age + lbph + svi + lcp + gleason +
##     pgg45
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     93 47.785
## 2     88 44.163  5    3.6218 1.4434 0.2167
```

Chapter3, Exercise 2

a) The next chunk contains the code for loading the cheddar data and then modelling the response as taste and threee chemical components as predictors. From the p-values, only H2S and Lactic are statistically significant as they have p-values<0.05.

```
data(cheddar,package="faraway")
head(cheddar)
```

```
##   taste Acetic   H2S Lactic
## 1  12.3  4.543 3.135   0.86
## 2  20.9  5.159 5.043   1.53
## 3  39.0  5.366 5.438   1.57
## 4  47.9  5.759 7.496   1.81
## 5   5.6  4.663 3.807   0.99
## 6  25.9  5.697 7.601   1.09
```

```
lmod3<-lm(taste~Acetic+H2S+Lactic,data=cheddar)
summary(lmod3)
```

```
##
## Call:
## lm(formula = taste ~ Acetic + H2S + Lactic, data = cheddar)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.390  -6.612  -1.009   4.908  25.449
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -28.8768    19.7354  -1.463  0.15540
## Acetic        0.3277     4.4598   0.073  0.94198
## H2S           3.9118     1.2484   3.133  0.00425 **
## Lactic       19.6705     8.6291   2.280  0.03108 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.13 on 26 degrees of freedom
## Multiple R-squared:  0.6518, Adjusted R-squared:  0.6116
## F-statistic: 16.22 on 3 and 26 DF,  p-value: 3.81e-06
```

b) The next chunk contains the code for modelling taste as response and all the responses on their original scale which means taking the exponenet of Acetic and H2S data. From the p-values it is clear that only lactic acid is statistically significant.

```
lmod4<-lm(taste~exp(Acetic)+exp(H2S)+Lactic,data=cheddar)
summary(lmod4)
```

```
##
## Call:
## lm(formula = taste ~ exp(Acetic) + exp(H2S) + Lactic, data = cheddar)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.209  -7.266  -1.651   7.385  26.335
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.897e+01  1.127e+01  -1.684   0.1042
## exp(Acetic)  1.891e-02  1.562e-02   1.210   0.2371
## exp(H2S)     7.668e-04  4.188e-04   1.831   0.0786 .
## Lactic       2.501e+01  9.062e+00   2.760   0.0105 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.19 on 26 degrees of freedom
## Multiple R-squared:  0.5754, Adjusted R-squared:  0.5264
## F-statistic: 11.75 on 3 and 26 DF,  p-value: 4.746e-05
```

c) Cannot use F-test as there is non linear transformation of variables meaning models are not nested and also models have the same degrees of freedom.From adjusted R-Square and p-value it looks like first model is better at prediction. As can be seen, anova function returns no result for F-statistic and p-values.

```
anova(lmod3,lmod4)
```

```
## Analysis of Variance Table
##
## Model 1: taste ~ Acetic + H2S + Lactic
## Model 2: taste ~ exp(Acetic) + exp(H2S) + Lactic
##   Res.Df    RSS Df Sum of Sq F Pr(>F)
## 1     26 2668.4
## 2     26 3253.6  0    -585.2
```

```
summary(lmod3)
```

```
##
## Call:
## lm(formula = taste ~ Acetic + H2S + Lactic, data = cheddar)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.390  -6.612  -1.009   4.908  25.449
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -28.8768    19.7354  -1.463  0.15540
## Acetic        0.3277     4.4598   0.073  0.94198
## H2S           3.9118     1.2484   3.133  0.00425 **
## Lactic       19.6705     8.6291   2.280  0.03108 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.13 on 26 degrees of freedom
```

```
## Multiple R-squared:  0.6518, Adjusted R-squared:  0.6116
## F-statistic: 16.22 on 3 and 26 DF,  p-value: 3.81e-06
```

```
summary(lmod4)
```

```
##
## Call:
## lm(formula = taste ~ exp(Acetic) + exp(H2S) + Lactic, data = cheddar)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.209  -7.266  -1.651   7.385  26.335
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.897e+01  1.127e+01  -1.684   0.1042
## exp(Acetic)  1.891e-02  1.562e-02   1.210   0.2371
## exp(H2S)     7.668e-04  4.188e-04   1.831   0.0786 .
## Lactic       2.501e+01  9.062e+00   2.760   0.0105 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.19 on 26 degrees of freedom
## Multiple R-squared:  0.5754, Adjusted R-squared:  0.5264
## F-statistic: 11.75 on 3 and 26 DF,  p-value: 4.746e-05
```

d)From (a), the estimate of coefficient for H2S is 3.9118. If H2S is increased 0.01, it is expected that taste will increases by $0.01 \times 3.9118 = 0.039$ on average, holding everything else constant

e) The next chunk contains the code for calculating answer to this part

```
k=0
kn=0.01
percent_increase=(exp(kn)-exp(k))*100/exp(k)
percent_increase
```

```
## [1] 1.005017
```

Chapter4, Exercise 1

a) The next chunk contains the code for loading the prostate data and fitting the model with lpsa as response and other variables as predictors. Also contained is the prediction for a new data point given in the problem. As the problem asks for a prediction for a new value, Prediction interval is more appropriate. The fit value is 2.38 and interval is 2.85

```
data(prostate,package="faraway")
lmod5<-lm(lpsa ~ .,data=(prostate))
summary(lmod5)
```

```
##
## Call:
## lm(formula = lpsa ~ ., data = (prostate))
##
## 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
```

```
x0=data.frame(lcavol=1.44692,lweight=3.62301,age=65.000,lbph=0.30010,svi=0.0,lcp=-0.79851,gleason=7,pgg
lpsa_pred1<-predict(lmod5,x0,interval="prediction")
lpsa_pred1
```

```
##        fit       lwr      upr
## 1 2.389053 0.9646584 3.813447
```

b) The next chunk contains the code for making the prediction about the new data point which has different value for age that is 20. The predicted value is 3.27 and the interval is 3.46. The interval is wider in that case as the age value is more distant from the mean value for age that is 63.86 which is nearer to 65 than to 20.

```
x01=data.frame(lcavol=1.44692,lweight=3.62301,age=20.000,lbph=0.30010,svi=0.0,lcp=-0.79851,gleason=7,pg
lpsa_pred2<-predict(lmod5,x01,interval="prediction")
lpsa_pred2
```

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

```
sprintf("The predicted value for this case is %s",lpsa_pred2[1])
```

```
## [1] "The predicted value for this case is 3.27272565327705"
```

```
sprintf("The mean value for age is %s",mean(prostate$age))
```

```
## [1] "The mean value for age is 63.8659793814433"
```

c) From summary of the model object, it is clear that lcavol, lweight and svi are significant predictors at 5% level. Since age is not a predictor anymore, the predictions for both (a) and (b) will be same. The predicted response is 2.37 and the interval is 2.87 which is slightly wider than when predicted from the model with all predictors. Also seen from the object returned by the anova function, since the p-value is 0.21 which is greater than 0.05, we fail to reject that the smaller model is better.Hence I will use the smaller model.

```
lmod6<-lm(lpsa ~ lcavol+lweight+svi,data=(prostate))
predict(lmod6,x0,interval="prediction")
```

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

```
predict(lmod6,x01,interval="prediction")
```

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

```
anova(lmod6,lmod5)
```

```
## Analysis of Variance Table
##
## Model 1: lpsa ~ lcavol + lweight + svi
## Model 2: lpsa ~ lcavol + lweight + age + lbph + svi + lcp + gleason +
##     pgg45
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     93 47.785
## 2     88 44.163  5    3.6218 1.4434 0.2167
```

Chapter 4, Exercise 2

a) The next chunk contains the code for loading the teengamb data and modelling gamble as response and all other variables as predictors. Also it contains the code for extracting the averages of data for male sex and predicting the response for such a male. The predicted value is 29.775 and CI is 93.1 units wide.

```
data(teengamb,package="faraway")
head(teengamb)
```

```
##   sex status income verbal gamble
## 1   1     51   2.00      8    0.0
## 2   1     28   2.50      8    0.0
## 3   1     37   2.00      6    0.0
## 4   1     28   7.00      4    7.3
## 5   1     65   2.00      8   19.6
## 6   1     61   3.47      6    0.1
```

```
lmod7<-lm(gamble~.,data=(teengamb))
male_data <- teengamb[teengamb$sex %in% 0,]
new_x <- sapply(male_data[,1:4], mean)
gam_predict1<-predict(lmod7,newdata=as.data.frame(t(new_x)),interval="prediction",level=0.95)
gam_predict1
```

```
##      fit       lwr      upr
## 1 29.775 -16.82649 76.37649
```

b) The next chunk contains the code for making the prediction for a male having maximum value for every attribute.The predicted value is 71.3 and the CI is 108.49 units wide. This result is expected as maximum values are far away from teh mean value which is tha case in (a)

```
new_x1 <- sapply(male_data[,1:4], max)
gam_predict<-predict(lmod7,newdata=as.data.frame(t(new_x1)),interval="prediction",level=0.95)
gam_predict
```

```
##        fit      lwr    upr
## 1 71.30794 17.06588 125.55
```

c) The next chunk contains the code for making the linear model with square root of the response.also contained is the prediction for an average male.The predicted value is 4.39 and the CI is (0.11,8.67). Squaring the prediction we arrive at our answer that is fit is 19.27 and CI is (0.0122,75.18)

```
lmod8<-lm(sqrt(gamble)~.,data=(teengamb))

gam_predict2<-predict(lmod8,as.data.frame(t(new_x)),interval="prediction")
gam_predict2
```

```
##        fit       lwr      upr
## 1 4.390678 0.1104836 8.670872
```

```
gam_predict2^2
```

```
##        fit        lwr      upr
## 1 19.27805 0.01220663 75.18403
```

d)Below is the code to make a prediction for female with given values of predictors. We get a negative value of predicted value which does not make any practical sense and hence is not credible since the response is a square root that cannot be negative.

```
gam_predict2<-predict(lmod8,data.frame(sex=1,status=20,income=1,verbal=10),interval="prediction")
gam_predict2
```

```
##        fit       lwr      upr
## 1 -2.08648 -6.908863 2.735903
```

Chapter 4 Exercise 3

a) The next chunk contains the code for loading the data and tabulating it as required by the problem. We can cautiously make a prediction for temperature of 25 C and Humidity of 60% assuming a linear relationship. As 25= (20+30)/2 and 60=(45+75)/2..we can assume the water content to be (72.5+81.5+69.5+78.25)/4 = 75.4375. But these predictions are made assuming that the relationship is linear.

```
data(snail,package="faraway")
snail
```

```
##    water temp humid
## 1     76   20    45
## 2     64   20    45
## 3     79   20    45
## 4     71   20    45
## 5     72   20    75
## 6     82   20    75
## 7     86   20    75
## 8     86   20    75
## 9    100   20   100
## 10    96   20   100
## 11    92   20   100
## 12   100   20   100
## 13    72   30    45
## 14    72   30    45
## 15    64   30    45
## 16    70   30    45
## 17    72   30    75
## 18    75   30    75
## 19    82   30    75
## 20    84   30    75
```

```
## 21   100   30   100
## 22    94   30   100
## 23    98   30   100
## 24    99   30   100
```

```
head(snail)
```

```
##   water temp humid
## 1    76   20    45
## 2    64   20    45
## 3    79   20    45
## 4    71   20    45
## 5    72   20    75
## 6    82   20    75
```

```
xtabs(water ~ temp + humid, snail)/4
```

```
##     humid
## temp    45    75   100
##   20 72.50 81.50 97.00
##   30 69.50 78.25 97.75
```

b) Below is code from modelling water content as repsonse and temperature and humidity as predictors. Also contained is the prediction for temperature and humidty as 25 and 60 respectively. The predicted value is 76.44 and CI is (64.58, 88.29)

```
lmod9<-lm(water~temp+humid,data=snail)
summary(lmod9)
```

```
##
## Call:
## lm(formula = water ~ temp + humid, data = snail)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.456  -2.915   1.461   3.613   8.749
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 52.61081    6.85346   7.677 1.59e-07 ***
## temp        -0.18333    0.22645  -0.810    0.427
## humid        0.47349    0.05036   9.403 5.63e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.547 on 21 degrees of freedom
## Multiple R-squared:  0.8092, Adjusted R-squared:  0.791
## F-statistic: 44.53 on 2 and 21 DF,  p-value: 2.793e-08
```

```
pred_w<-predict(lmod9,data.frame(temp=25,humid=60),interval = "prediction")
pred_w
```

```
##        fit      lwr      upr
## 1 76.43681 64.58094 88.29269
```

c) Predicting for Temp=30 and Humidity=75, the predicted value is 82.62 and CI is (70.61, 94.63). While in case (a) it is 78.25. The first case is just the arithmetic mean for all the observations at the temperature and humidity values. While in this case we are using regression. Regression is trying to

minimize variation for both values of temperature (20 and 30) while in case (a) when we are taking mean we are minimizing the variation only at that particular temperature if we assume the humidity to be constant at 75.

```
pred_w1<-predict(lmod9,data.frame(temp=30,humid=75),interval = "prediction")
pred_w1
```

```
##        fit     lwr      upr
## 1 82.62248 70.6147 94.63027
```

d) Below is the code for part d. As can be seen we can simply solve the equation (beta1$temp$+$beta2$Hum=0) by assuming two values of humidity and arriving at two values of humidity. There is one equation and two variables hence there is no unique solution.

```
Hum=c(60,0)
Temp=summary(lmod9)$coef[3,1]*Hum*(-1)/summary(lmod9)$coef[2,1]
Temp
```

```
## [1] 154.96   0.00
```

```
pred_w2<-predict(lmod9,data.frame(temp=Temp,humid=Hum),interval = "prediction")
pred_w2
```

```
##        fit       lwr       upr
## 1 52.61081 -9.729599 114.95121
## 2 52.61081 34.274977  70.94663
```

e) Below is the code for calculating value of humidity would give a predicted response of 80% water content at temperature of 25 C The value of Humidity is 67.525318

```
Temp=25
Hum=(80-summary(lmod9)$coef[1,1]-summary(lmod9)$coef[2,1]*Temp)/summary(lmod9)$coef[3,1]
Hum
```

```
## [1] 67.52538
```