Name: Shiva Kumar Peddapuram

UnivID: 811235874

FINAL EXAM – Math 40015/50015

Fall 2022

SHOW ALL YOUR WORK and write complete and coherent answers. No partial credit will be given if no work is shown. Please write as clearly and neatly as possible. If I cannot read your answers, I cannot give you any credit. Feel free to ask for more paper if you need more space. GOOD LUCK !!!

**In the data set “landrent” in package alr4, the variables are average rent per acre Y planted to alfalfa,average rent paid $X_1$ for all tillable land, density of dairy cows $X_2$ (number per square mile), and proportion $X_3$ of farmland used as pasture. You need to answer the following questions based on your own code (not the lm function).**

**1. For the data, the full model is $E(Y) = \beta_0 + \beta_1X_1 + \beta_2X_2 + \beta_3X_3 + \beta_4X_2 * X_3$.**

**And the reduced model is $E(Y) = \beta_0 + \beta_1X_1 + \beta_2X_2 + \beta_3X_3$.**
**Use the F-test to test which model is more appropriate for the data. Compute the F-statistic in detail. Report the p-value and summarize your conclusion.**

Solution:

```r
##1
library(alr4)
data("landrent")
n=dim(landrent)[1]
p<-4
x1<- landrent$X1
x2<- landrent$X2
x3<- landrent$X3
x4<- x2*x3
y<- landrent$Y
X=matrix(c(rep(1,n),x1,x2,x3,x4),nrow=n,ncol=5,byrow=F)

beta.hat=solve(t(X)%*%X)%*%t(X)%*%y
sigma2.hat=t(y-X%*%beta.hat)%*%(y-X%*%beta.hat)/(n-5)
sigma.hat=sqrt(sigma2.hat)
sigma.hat
y.hat=X%*%beta.hat
t1=beta.hat[1]/sqrt(sigma2.hat*solve(t(X)%*%X)[1,1])
p.value1=2*(1-pt(abs(t1),n-5))
t1
p.value1
t2=beta.hat[2]/sqrt(sigma2.hat*solve(t(X)%*%X)[2,2])
p.value2=2*(1-pt(abs(t2),n-5))
t2
p.value2
t3=beta.hat[3]/sqrt(sigma2.hat*solve(t(X)%*%X)[3,3])
p.value3=2*(1-pt(abs(t3),n-5))
t3
p.value3
t4=beta.hat[4]/sqrt(sigma2.hat*solve(t(X)%*%X)[4,4])
p.value4=2*(1-pt(abs(t4),n-5))
t4
p.value4
t5=beta.hat[5]/sqrt(sigma2.hat*solve(t(X)%*%X)[5,5])
p.value5=2*(1-pt(abs(t4),n-5))
t5
p.value5

p.red=1
df.red=n-p.red-1
df.ful=n-p-1
X.red=cbind(rep(1,n),x1,x2,x3)
beta.hat.red=solve(t(X.red)%*%X.red)%*%t(X.red)%*%y
y.hat.red=X.red%*%beta.hat.red
Rss.red=as.numeric(t(y-y.hat.red)%*%(y-y.hat.red))
Rss.ful=as.numeric(t(y-y.hat)%*%(y-y.hat))
F.stat=((Rss.red-Rss.ful)/(df.red-df.ful))/(Rss.ful/df.ful)
pvalue.F=1-pf(F.stat,df.red-df.ful,df.ful)
pvalue.F
F.stat
```

Execution part:

```
> ##1
> library(alr4)
> data("landrent")
> n=dim(landrent)[1]
> p<-4
> x1<- landrent$X1
> x2<- landrent$X2
> x3<- landrent$X3
> x4<- x2*x3
> y<- landrent$Y
> X=matrix(c(rep(1,n),x1,x2,x3,x4),nrow=n,ncol=5,byrow=F)
> beta.hat=solve(t(X)%*%X)%*%t(X)%*%y
> sigma2.hat=t(y-X%*%beta.hat)%*%(y-X%*%beta.hat)/(n-5)
> sigma.hat=sqrt(sigma2.hat)
> sigma.hat
         [,1]
[1,] 8.986852
> y.hat=X%*%beta.hat
> t1=beta.hat[1]/sqrt(sigma2.hat*solve(t(X)%*%X)[1,1])
> p.value1=2*(1-pt(abs(t1),n-5))
> t1
          [,1]
[1,] -2.018741
> p.value1
           [,1]
[1,] 0.04784446
> t2=beta.hat[2]/sqrt(sigma2.hat*solve(t(X)%*%X)[2,2])
> p.value2=2*(1-pt(abs(t2),n-5))
> t2
         [,1]
[1,] 13.40397
> p.value2
     [,1]
[1,]    0
> t3=beta.hat[3]/sqrt(sigma2.hat*solve(t(X)%*%X)[3,3])
> p.value3=2*(1-pt(abs(t3),n-5))
> t3
         [,1]
[1,] 4.785026
> p.value3
             [,1]
[1,] 1.097297e-05
> t4=beta.hat[4]/sqrt(sigma2.hat*solve(t(X)%*%X)[4,4])
> p.value4=2*(1-pt(abs(t4),n-5))
> t4
         [,1]
[1,] 1.367199
> p.value4
          [,1]
[1,] 0.1764994
> t5=beta.hat[5]/sqrt(sigma2.hat*solve(t(X)%*%X)[5,5])
> p.value5=2*(1-pt(abs(t4),n-5))
> t5
          [,1]
[1,] -2.164556
> p.value5
          [,1]
[1,] 0.1764994
> p.red=1
> df.red=n-p.red-1
> df.ful=n-p-1
> X.red=cbind(rep(1,n),x1,x2,x3)
> beta.hat.red=solve(t(X.red)%*%X.red)%*%t(X.red)%*%y
> y.hat.red=X.red%*%beta.hat.red
> Rss.red=as.numeric(t(y-y.hat.red)%*%(y-y.hat.red))
> Rss.ful=as.numeric(t(y-y.hat)%*%(y-y.hat))
> F.stat=((Rss.red-Rss.ful)/(df.red-df.ful))/(Rss.ful/df.ful)
> pvalue.F=1-pf(F.stat,df.red-df.ful,df.ful)
> pvalue.F
[1] 0.2076398
> F.stat
[1] 1.561768
>
```

***Explanation:***

***From the above we can conclude that the p value of f is 0.2076398 and F.stat value is 1.561768***

***Pvalue1= 0.04784446***

***Pvalue2= 0***

***Pvalue3= 1.097297e-05***

***Pvalue4= 0.1764994***

***Pvalue5= 0.1764994***

***Sigma hat value is 8.986852***

2.) **Suppose that the full model is chosen. Now you are asked to estimate all the parameters of the full model including the variance σ2**

**Solution:**

**Coding part:**

```
##2
X=matrix(c(rep(1,n),landrent$X1,landrent$X2,landrent$X3,landrent$X4),nrow=n,ncol=5,byrow=F)
beta.hat=solve(t(X)%*%X)%*%t(X)%*%landrent$Y
sigma2.hat=t(landrent$Y-X%*%beta.hat)%*%(landrent$Y-X%*%beta.hat)/(n-5)
sigma2.hat
beta.hat

sigma.hat=sqrt(sigma2.hat)
sigma.hat
```

***Explanation part:***

All the parameter values are as below

Execution part for the above code

```
> ##2
> X=matrix(c(rep(1,n),landrent$X1,landrent$X2,landrent$X3,landrent$X4),nrow=n,ncol=5,byrow=F)
> beta.hat=solve(t(X)%*%X)%*%t(X)%*%landrent$Y
> sigma2.hat=t(landrent$Y-X%*%beta.hat)%*%(landrent$Y-X%*%beta.hat)/(n-5)
> sigma2.hat
         [,1]
[1,] 86.69043
> beta.hat
            [,1]
[1,]  -2.8282148
[2,]   0.8832666
[3,]   0.4317553
[4,] -11.3804544
[5,]  -1.0117308
> 
```

So from the above code we gave initiated sigma2.hat values and beta hat value

So from the above the sigma2 havt value is 86.69043

Beta.hat values are

-2.8282148

0.8832666

0.4317553

-11.3804544

-1.0117308

**3.) Again for the full model, you need to construct a 99% confidence interval for each of the slopes**
**$\beta_1$,$\beta_2$,$\beta_3$ and $\beta_4$. Is 0 included by each confidence interval?**

## Coding part:

```
##3

#For slope beta1:
lower.b=beta.hat[2,1]-qt(1-0.01/2,length(landrent$Y)-2)*0.069
lower.b
upper.b=beta.hat[2,1]+qt(1-0.01/2,length(landrent$Y)-2)*0.069
upper.b

#For slope beta2:
lower.b=beta.hat[3,1]-qt(1-0.01/2,length(landrent$Y)-2)*0.108
lower.b
upper.b=beta.hat[3,1]+qt(1-0.01/2,length(landrent$Y)-2)*0.108
upper.b

#For slope beta3:
lower.b=beta.hat[4,1]-qt(1-0.01/2,length(landrent$Y)-2)*11.89
lower.b
upper.b=beta.hat[4,1]+qt(1-0.01/2,length(landrent$Y)-2)*11.89
upper.b

#For slope beta4:
lower.b=beta.hat[5,1]-qt(1-0.01/2,length(landrent$Y)-2)*2.84
lower.b
upper.b=beta.hat[5,1]+qt(1-0.01/2,length(landrent$Y)-2)*2.84
upper.b
```

```
##3

lower.b=beta.hat[2,1]-qt(1-0.05/2,length(landrent$Y)-2)*0.069
lower.b
upper.b=beta.hat[2,1]+qt(1-0.05/2,length(landrent$Y)-2)*0.069
upper.b
```

## Explanation:

```
>
> #For slope beta1:
> lower.b=beta.hat[2,1]-qt(1-0.01/2,length(landrent$Y)-2)*0.069
> lower.b
[1] 0.7001679
> upper.b=beta.hat[2,1]+qt(1-0.01/2,length(landrent$Y)-2)*0.069
> upper.b
[1] 1.066365
> #For slope beta2:
> lower.b=beta.hat[3,1]-qt(1-0.01/2,length(landrent$Y)-2)*0.108
> lower.b
[1] 0.145166
> upper.b=beta.hat[3,1]+qt(1-0.01/2,length(landrent$Y)-2)*0.108
> upper.b
[1] 0.7183446
> #For slope beta3:
> lower.b=beta.hat[4,1]-qt(1-0.01/2,length(landrent$Y)-2)*11.89
> lower.b
[1] -42.93181
> upper.b=beta.hat[4,1]+qt(1-0.01/2,length(landrent$Y)-2)*11.89
> upper.b
[1] 20.1709
> #For slope beta4:
> lower.b=beta.hat[5,1]-qt(1-0.01/2,length(landrent$Y)-2)*2.84
> lower.b
[1] -8.547967
> upper.b=beta.hat[5,1]+qt(1-0.01/2,length(landrent$Y)-2)*2.84
> upper.b
[1] 6.524506
> |
```

```
>
> lower.b=beta.hat[2,1]-qt(1-0.05/2,length(landrent$Y)-2)*0.069
> lower.b
[1] 0.7454641
> upper.b=beta.hat[2,1]+qt(1-0.05/2,length(landrent$Y)-2)*0.069
> upper.b
[1] 1.021069
> |
```

**From the above code the**

**Slope beta1 lower  value is 0.7001679 and the upper value is 1.066365**

**Slope beta2 lower  value is 0.145166and the upper value is 0.7183446**

**Slope beta3 lower  value is  -42.93181and the upper value is 20.1709**

**Slope beta4 lower  value is  -8.547967and the upper value is 6.524506**

4. Without actually conducting hypothesis tests, is it possible to tell whether the null hypothesis of $H_0 : \beta_i = 0$ vs $H_1 : \beta_i 6= 0$ for i = 1,2,3,4 is rejected or failed to be rejected based on the results from question 3 above? If yes, what should be the chosen significance level for each hypothesis test?

Yes

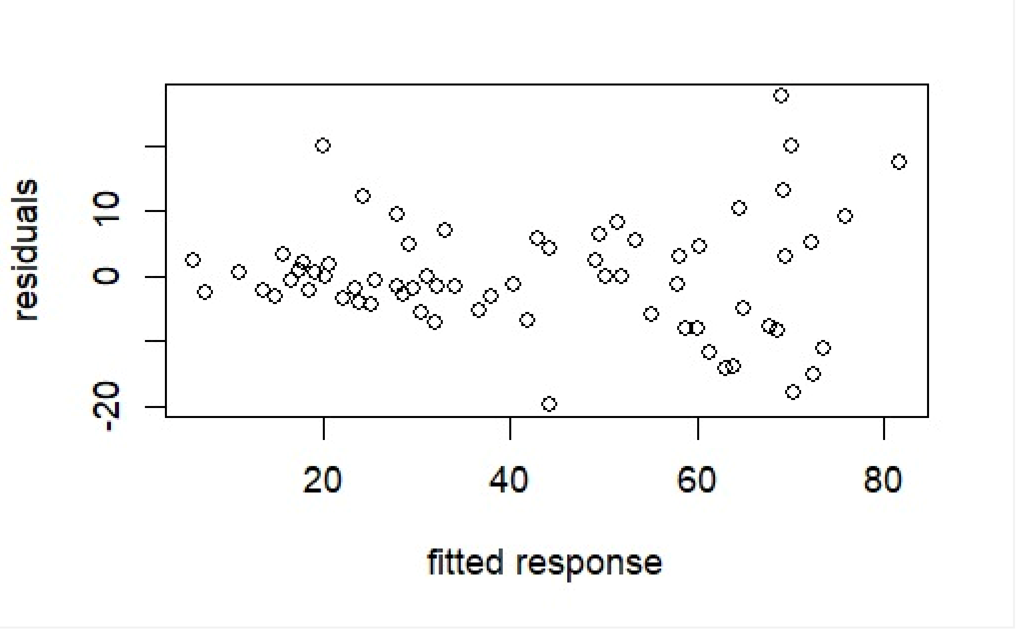
5. For the full model, obtain the residuals and make the residuals vs fitted response plot. Based on the

residual plot, can we say the linearity and constant variance assumptions hold for the model?

Code:

```
n <- dim(landrent)
X <- matrix(c(rep(1,n), landrent$X1, landrent$X2, landrent$X3, (landrent$X2 *landrent$X3)), nrow = n, ncol = 5, byrow=F)
Hat.matrix <- X%*%solve(t(X)%*%X)%*%t(X)
beta.hat <- solve(t(X)%*%X)%*%t(X)%*%Y
Y.hat <- X %*%beta.hat
residuals <- Y - Y.hat
plot(Y.hat, residuals, xlab = "fitted response")
```

Plot:




6. For the full model, compute the Cook's distances. Are there any influential outliers?

```
Y=T[,1]
n=length
p=dim(T)[2]-1
Design.mat=cbind(rep(1,n),T[,2:p])
Hat.mat=Design.mat%%solve(t(Design.mat)%%Design.mat)%*%t(Design.mat)
Beta.hat=solve(t(Design.mat)%%Design.mat)%%t(Design.mat)%*%Y
Y.hat=Design.mat%*%Beta.hat
Residuals=Y-Y.hat
Sigma2.hat=as.numeric(t(Residuals)%*%Residuals/(n-p-1))
Residuals.std=(sqrt(Sigma2.hat))^(-1)*Residuals/sqrt(1-diag(Hat.mat))
Cooks.dist=(p+1)^(-1)Residuals.std^2(diag(Hat.mat))/(1-diag(Hat.mat))
Cooks.dist
```

Cookes values are:

[1,] 0.0096767714

[2,] 0.0018195043

[3,] 0.0011164589

[4,] 0.0018918651

[5,] 0.0148028993

[6,] 0.0000000000

[7,] 0.0040994600

[8,] 0.0017958516

[9,] 0.1524981177

[10,] 0.0031335811

[11,] 0.0003579510

[12,] 0.0000000000

[13,] 0.0005165651

[14,] 0.0004618965

[15,] 0.0008748523

[16,] 0.0024739824

[17,] 0.0238670892

[18,] 0.0004078618

[19,] 0.0044626645

[20,] 0.0063360857

[21,] 2.3032168019

[22,] 0.0041461266

[23,] 0.0033917323

[24,] 0.0016862318

[25,] 0.0004451163

[26,] 0.0022295390

[27,] 0.0073755391

[28,] 0.0108104446

[29,] 0.0072242908

[30,] 0.0026610166

[31,] 0.0032014278

[32,] 0.0256406766

[33,] 0.1781418197

[34,] 0.0000000000

[35,] 0.0035344830

[36,] 0.0000000000

[37,] 0.0020024529

[38,] 0.0017217584

[39,] 0.0019575243

[40,] 0.0020871259

[41,] 0.0000000000

[42,] 0.8290480686

[43,] 0.0008620414

[44,] 0.0000000000

[45,] 0.0012575940

[46,] 0.0005721686

[47,] 0.0004889101

[48,] 0.0024873404

[49,] 0.0011713613

[50,] 0.0013437320

[51,] 0.0000000000

[52,] 0.0047446153

[53,] 0.0012186882

[54,] 0.0251603604

[55,] 0.0001844792

[56,] 0.0324588260

[57,] 0.0044145153

[58,] 0.0017752066

[59,] 0.0015641335

[60,] 0.0092061620

[61,] 0.0005836189

[62,] 0.0028163517

[63,] 0.0257507351

[64,] 0.0052395773

[65,] 0.0000000000

[66,] 0.0000000000

[67,] 0.0019049670

Influential values are:

[1]  9 21 33 42

**The data set "Challeng" records performance of O-rings for the 23 U.S. space shuttle missions prior to the Challenger disaster of January 20, 1986. For each of the previous missions, the temperature at takeoff and the pressure of a prelaunch test were recorded, along with the number of O-rings that failed out of 6. You need to answer the following questions based on the glm function of R.**

1. **Consider "temp" and "pres" as two predictors, "fail" as the number of "successes", and "n" as the total number of trials. Fit the binomial regression model y ~ temp + pres + temp : pres.**

```
##2nd part 1st question
library(alr4)
data(Challeng)
Challeng$fail <- Challeng$n - Challeng$fail
Challeng$fail <- as.factor(Challeng$fail)

model <- glm(fail ~ temp + pres + temp:pres, family = binomial(link = "logit"), data = Challeng)
summary(model)
```

**Execution part:**

```
> ##2nd part 1st question
> library(alr4)
> data(Challeng)
> Challeng$fail <- Challeng$n - Challeng$fail
> Challeng$fail <- as.factor(Challeng$fail)
> model <- glm(fail ~ temp + pres + temp:pres, family = binomial(link = "logit"), data = Challeng)
> summary(model)

Call:
glm(formula = fail ~ temp + pres + temp:pres, family = binomial(link = "logit"),
    data = Challeng)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3012   0.0000   0.2759   0.4422   0.9127

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  6.414e+01  1.537e+05       0        1
temp        -4.311e-01  2.239e+03       0        1
pres        -3.500e-01  7.687e+02       0        1
temp:pres    2.718e-03  1.119e+01       0        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 13.590  on 22  degrees of freedom
Residual deviance: 10.008  on 19  degrees of freedom
AIC: 18.008

Number of Fisher Scoring iterations: 19
```

```
> ##2nd question
> library(alr4)
> data(Challeng)
> Challeng$fail <- Challeng$n - Challeng$fail
> Challeng$fail <- as.factor(Challeng$fail)
> model <- glm(fail ~ temp + pres + temp:pres, family = binomial(link = "logit"), data = Challeng)
> summary(model)

Call:
glm(formula = fail ~ temp + pres + temp:pres, family = binomial(link = "logit"),
    data = Challeng)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3012   0.0000   0.2759   0.4422   0.9127

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  6.414e+01  1.537e+05       0        1
temp        -4.311e-01  2.239e+03       0        1
pres        -3.500e-01  7.687e+02       0        1
temp:pres    2.718e-03  1.119e+01       0        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 13.590  on 22  degrees of freedom
Residual deviance: 10.008  on 19  degrees of freedom
AIC: 18.008

Number of Fisher Scoring iterations: 19

> |
```

**2. Use your fitted model to estimate the probability of failure of an O-ring when the temperature was 31, and the pressure is 100**

**Code part:**

```
##2nd part 2nd question
predict(model, data.frame(temp = 31, pres = 100), type = "response")
```

**Execution part:**

```
> predict(model, data.frame(temp = 31, pres = 100), type = "response")
1
1
> |
> model2 <- glm(fail ~ temp + pres, family = binomial(link = "logit"), data = Challeng)
> summary(model2)

Call:
glm(formula = fail ~ temp + pres, family = binomial(link = "logit"),
    data = Challeng)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3012   0.0000   0.2759   0.4422   0.9127

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 29.07205 8937.49959   0.003    0.997
temp         0.11258    0.09677   1.163    0.245
pres        -0.17471   44.68749  -0.004    0.997

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 13.590  on 22  degrees of freedom
Residual deviance: 10.008  on 20  degrees of freedom
AIC: 16.008

Number of Fisher Scoring iterations: 19
```

3. **Consider another reduced binomial regression model y ~ temp + pres. Test which model (full model vs reduced model) is more appropriate? To answer it, you need to compute the test statistic ∆G2 in detail, report the p-value and summarize your conclusion**

**Solution:**

```
model2 <- glm(fail ~ temp + pres, family = binomial(link = "logit"), data = Challeng)
summary(model2)
```

**Execution part:**

```
> model2 <- glm(fail ~ temp + pres, family = binomial(link = "logit"), data = Challeng)
> summary(model2)

Call:
glm(formula = fail ~ temp + pres, family = binomial(link = "logit"),
    data = Challeng)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3012   0.0000   0.2759   0.4422   0.9127

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  29.07205 8937.49959   0.003    0.997
temp          0.11258    0.09677   1.163    0.245
pres         -0.17471   44.68749  -0.004    0.997

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 13.590  on 22  degrees of freedom
Residual deviance: 10.008  on 20  degrees of freedom
AIC: 16.008

Number of Fisher Scoring iterations: 19
```