STAT 513 fa 2020 hw 7

Do NOT use fancy R functions like `lm()` or `t.test()` to do any part of this homework.

1. Bring into the workspace the built-in R data set called `Puromycin` using the command `data(Puromycin)`. You may type `?Puromycin` to read more about the data. It contains the variables `rate`, `conc`, and `state`. Let $Y_1, \ldots, Y_n$ denote the values in the column `rate`, $x_{11}, \ldots, x_{n1}$ the values in the column `conc`, and $x_{12}, \ldots, x_{n2}$ the values defined by

$$x_{i2} = \begin{cases} 1 & \text{if } \texttt{state}_i = \texttt{treated} \\ 0 & \text{if } \texttt{state}_i = \texttt{untreated} \end{cases} \qquad \text{for } i = 1, \ldots, n,$$

where $\texttt{state}_i$ is the ith value of the column `state`. Suppose we wish to fit the model

$$Y_i = \beta_0 + \beta_1 \log(x_{i1}) + \beta_2 x_{i2} + \beta_3 x_{i2} \log(x_{i1}) + \varepsilon_i, \quad i = 1, \ldots, n,$$

where we believe that $\varepsilon_1, \ldots, \varepsilon_n$ are independent Normal$(0, \sigma^2)$ random variables.

   (a) Define new covariate values u_{i1}, u_{i2}, and u_{i3}, $i = 1, \ldots, n$, such that the above model can be expressed as

$$Y_i = \beta_0 + \beta_1 u_{i1} + \beta_2 u_{i2} + \beta_3 u_{i3} + \varepsilon_i, \quad i = 1, \ldots, n.$$

   Your answer should be like $u_{i1} =$, $u_{i2} =$, and $u_{i3} =$.

   Define $u_{1i} = \log(x_{i1})$, $u_{i2} = x_{i2}$, and $u_{i3} = \log(x_{i1})x_{i2}$, for $i = 1, \ldots, n$.

   (b) State the regression model for the treated and the untreated cases; that is, give an expression for Y_i when $x_{i2} = 1$ and when $x_{i2} = 0$.

   Treated:

$$Y_i = (\beta_0 + \beta_2) + (\beta_1 + \beta_3) \log(x_{i1}) + \varepsilon_i$$

   Untreated:

$$Y_i = \beta_0 + \beta_1 \log(x_{i1}) + \varepsilon_i$$

   (c) Use R to construct the design matrix

$$\mathbf{X} = \begin{bmatrix} 1 & u_{11} & u_{12} & u_{13} \\ \vdots & \vdots & \vdots & \\ 1 & u_{n1} & u_{n2} & u_{n3} \end{bmatrix}$$

   and then compute the least-squares estimators of β_0, β_1, β_2, and β_3.

   The R code

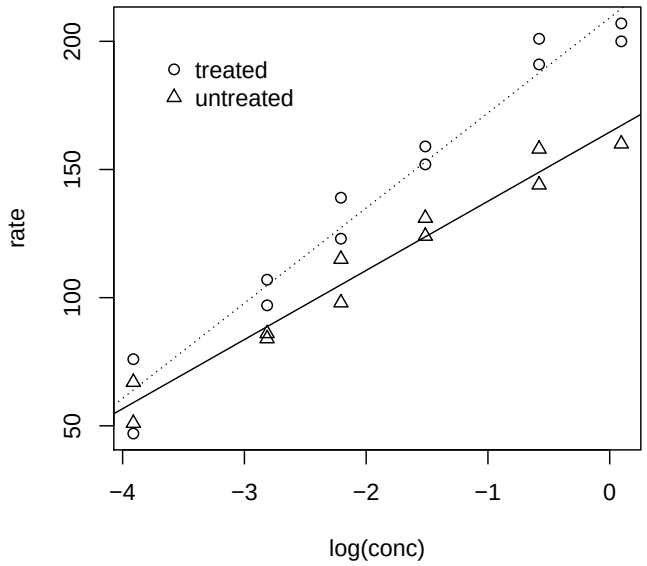
```
n <- nrow(Puromycin)

Y <- Puromycin$rate
u1 <- log(Puromycin$conc)
u2 <- ifelse(Puromycin$state=="untreated",0,1)
u3 <- u1*u2

X <- cbind(rep(1,n),u1,u2,u3)
beta.hat <- solve( t(X) %*% X) %*% t(X) %*% Y
```

gives $\hat{\beta}_0 = 164.58839$, $\hat{\beta}_1 = 26.98207$, $\hat{\beta}_2 = 44.60611$, and $\hat{\beta}_3 = 10.12837$.

(d) The following R code produces a scatterplot of $Y_1, \ldots, Y_n$ against the values $\log(x_{11}), \ldots, \log(x_{n1})$, where circles are used for the treatment cases and triangles for the untreated cases:

```
plot(Y~u1,pch=ifelse(u2==1,1,2),xlab="log(conc)",ylab="rate")
legend(x = -3.75, y = 200,legend=c("treated","untreated"),pch=c(1,2),bty="n")
```

Add some commands to this R code in order to produce a plot like the one below, with least-squares lines fitted to the treated and untreated cases. *Hint: Think about how to use the least-squares estimates $\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{\beta}_2$ and $\hat{\beta}_3$ to produce these lines with the* `abline()` *function.*



Use these commands:

```
abline(beta.hat[1],beta.hat[2])
abline(beta.hat[1]+beta.hat[3],beta.hat[2]+beta.hat[4],lty=3)
```

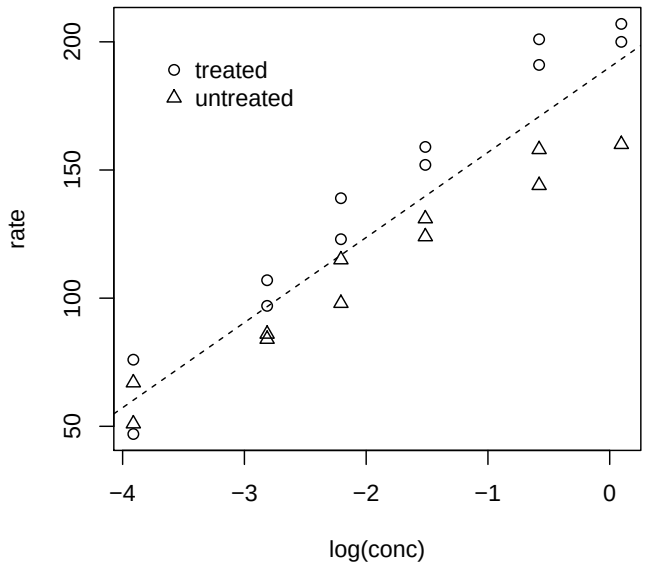
(e) Suppose we wish to investigate whether the treatment has any effect; more precisely, suppose we wish to test whether the regression functions for treated and untreated cases are the same. Give a set of hypotheses we could test in order to decide whether the two regression functions are equal.

We wish to test the hypotheses

$$H_0\text{: } \beta_3 = \beta_4 = 0 \text{ versus } H_1\text{: Either } \beta_3 \text{ or } \beta_4 \text{ or both are not equal to zero.}$$

If $\beta_3 = \beta_4 = 0$, then the regression function for the treated cases is the same as for the untreated cases.

(f) Consider the plot below, in which a least-squares line has been fitted to the points without regard for the treatment variable. Give the intercept and slope of this line.



This fitted line corresponds to a reduced model in which the treatment variable is ignored. So we omit the variables u_{i2} and u_{i3} from the model. In R, we do the following:

```
X.red <- cbind(rep(1,n),u1)
beta.hat.red <- solve( t(X.red) %*% X.red) %*% t(X.red) %*% Y

plot(Y~u1,pch=ifelse(u2==1,1,2),xlab="log(conc)",ylab="rate")
legend(x = -3.75, y = 200,legend=c("treated","untreated"),pch=c(1,2),bty="n")

abline(beta.hat.red[1],beta.hat.red[2],lty=2)
```

This gives the intercept 190.0854 and the slope 33.20268.

(g) Compute the sum of the squared residuals for the model in part (f).

We do

```
e.hat.red <- Y - X.red %*% beta.hat.red
SSE.red <- sum(e.hat.red^2)
```

which gives 6210.028.

(h) Compute the sum of the squared residuals for the model in part (a).

We do

```
e.hat <- Y - X %*% beta.hat
SSE.full <- sum(e.hat^2)
```

which gives 1591.245.

(i) Compute the F-statistic for testing the hypotheses in part (e) with the full-reduced model F-test.

This is the full-reduced model F-test with $r = 1$. Note that $p = 3$. Therefore the test statistic is

$$\frac{(\text{SSE}_{\text{Red}} - \text{SSE}_{\text{Full}})/(3-1)}{\text{SSE}_{\text{Full}}/(23-3-1)},$$

which we compute in R with

```
F.stat <- ((SSE.red - SSE.full)/(3-1))/(SSE.full/(n-3-1))
```

which gives 27.5749.

(j) Give the critical value for the full-reduced model F-test at the $\alpha = 0.01$ significance level.

The critical value is the upper α quantile of the $F_{3-1,23-3-1}$ distribution, which is

`qf(.99,3-1,n-3-1)` $= 5.925879$.

(k) State your conclusion about the hypotheses in part (e) at the $\alpha = 0.01$ significance level.

Since $27.5749 > 5.925879$ we reject $H_0 : \beta_3 = \beta_4 = 0$ at the $\alpha = 0.05$ significance level.

(l) Give the p-value of these data for testing the hypotheses in part (e).

The p-value is the smallest significance level at which the test statistic value 27.5749 would lead us to reject the null hypothesis. This is equal to the area under the pdf of the $F_{3-1,23-3-1}$ distribution to the right of 27.5749, which is given by

$$\texttt{1-pf(F.stat,2,n-3-1)} = 2.410538 \times 10^{-6}.$$

(m) Using the full model in part (a), construct a 95% confidence interval for the height of the true regression function for treated cases when the natural log of the concentration is equal to -1.

The height of the regression function for treated cases when the natural log of the concentration is equal to -1 is given by

$$\beta_0 + \beta_1(-1) + \beta_2(1) + \beta_3(-1) = \beta_0 - \beta_1 + \beta_2 - \beta_3,$$

which can be written as $\mathbf{a}^T\boldsymbol{\beta}$, with $\mathbf{a} = (1, -1, 1, -1)^T$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \beta_3)^T$. From the formula in the notes, we get a 95% confidence interval for $\mathbf{a}^T\boldsymbol{\beta}$ with the following:

```
sigma.hat <- sqrt( sum(e.hat) / (n - 3 - 1) )

a = c(1,-1,1,-1)

loco <- t(a)%*%beta.hat-qt(.975,n-3-1)*sigma.hat*sqrt(t(a)%*%solve(t(X)%*%X)%*%a)
upco <- t(a)%*%beta.hat+qt(.975,n-3-1)*sigma.hat*sqrt(t(a)%*%solve(t(X)%*%X)%*%a)
```

which gives the interval $(165.6007, 178.5674)$.

(n) Using the full model in part (a), construct a 95% confidence interval for the difference in the heights of the true regression functions for treated and untreated cases when the natural log of the concentration is equal to -1.

The difference in these heights is given by

$$\beta_0 + \beta_1(-1) + \beta_2(1) + \beta_3(-1) - [\beta_0 + \beta_1(-1)] = \beta_2 - \beta_3,$$

which can be written as $\mathbf{a}^T\boldsymbol{\beta}$, with $\mathbf{a} = (0, 0, 1, -1)^T$. In R we use the code

```
a = c(0,0,1,-1)

loco <- t(a)%*%beta.hat-qt(.975,n-3-1)*sigma.hat*sqrt(t(a)%*%solve(t(X)%*%X)%*%a)
upco <- t(a)%*%beta.hat+qt(.975,n-3-1)*sigma.hat*sqrt(t(a)%*%solve(t(X)%*%X)%*%a)
```

which gives the interval $(24.67704, 44.27844)$.

```
a = c(0,0,1,-1)

loco <- t(a)%*%beta.hat-qt(.975,n-3-1)*sigma.hat*sqrt(t(a)%*%solve(t(X)%*%X)%*%a)
upco <- t(a)%*%beta.hat+qt(.975,n-3-1)*sigma.hat*sqrt(t(a)%*%solve(t(X)%*%X)%*%a)
```

(o) Give the ANOVA table resulting from fitting the full model in part (a).

In R we make the following calculations:

```
Y.hat <- X %*% beta.hat

SST <- sum( (Y - mean(Y))^2)
SSE <- sum( (Y - Y.hat)^2 )
SSM <- SST - SSE

MSE <- SSE / (n-3-1)
MSM <- SSM / 3

Fn <- MSM/MSE

1-pf(Fn,3,n-3-1)
```

Then the ANOVA table is

	df	SS	MS	F_n	p-value
Model	3	48074.06	16024.69	191.3401	2.264855×10^{-14}
Error	19	1591.245	83.74976		
Total	22	49665.3			

2. Show that in the $p = 1$ case (simple linear regression), the matrix formula $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ gives

$$\hat{\beta}_0 = \bar{Y}_n - \hat{\beta}_1\bar{x}_n \quad \text{and} \quad \hat{\beta}_1 = S_{xY}/S_{xx}.$$

Hint: for any 2×2 invertible matrix we have

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

We have

$$\mathbf{X}^T\mathbf{Y} = \begin{bmatrix} \sum_{i=1}^n Y_i \\ \sum_{i=1}^n x_i Y_i \end{bmatrix} \quad \text{and} \quad \mathbf{X}^T\mathbf{X} = \begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2. \end{bmatrix}$$

Applying the inversion formula for a 2×2 matrix, we have

$$(\mathbf{X}^T\mathbf{X})^{-1} = \frac{1}{n\sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}\begin{bmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{bmatrix} = \frac{1}{nS_{xx}}\begin{bmatrix} \sum_{i=1}^n x_i^2 & -n\bar{x}_n \\ -n\bar{x}_n & n \end{bmatrix}.$$

Finally

$$
\begin{aligned}
(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} &= \frac{1}{nS_{xx}}\begin{bmatrix} \sum_{i=1}^n x_i^2 & -n\bar{x}_n \\ -n\bar{x}_n & n \end{bmatrix}\begin{bmatrix} \sum_{i=1}^n Y_i \\ \sum_{i=1}^n x_i Y_i \end{bmatrix} \\
&= \frac{1}{S_{xx}}\begin{bmatrix} \bar{Y}_n \sum_{i=1}^n x_i^2 - \bar{x}_n \sum_{i=1}^n x_i Y_i \\ -n\bar{x}_n\bar{Y}_n + \sum_{i=1}^n x_i Y_i \end{bmatrix} \\
&= \frac{1}{S_{xx}}\begin{bmatrix} \bar{Y}_n(\sum_{i=1}^n x_i^2 - n\bar{x}_n^2) - \bar{x}_n(\sum_{i=1}^n x_i Y_i - n\bar{x}_n\bar{Y}_n) \\ \sum_{i=1}^n (x_i - \bar{x}_n)(Y_i - \bar{Y}_n) \end{bmatrix} \\
&= \begin{bmatrix} \bar{Y}_n - \bar{x}_n S_{xY}/S_{xx} \\ S_{xY}/S_{xx} \end{bmatrix},
\end{aligned}
$$

which verifies the result.