Quiz 6

a) Write code to compute SST, SS\_explained, and SS\_unexplained using their respective defining formulas, for a quadratic polynomial regression model of the brainhead data used in the previous lab. You may use lm() and predict(). Report the numerical answers.

 dat = read.table("http://sites.stat.washington.edu/people/marzban/390/spring21/brainhead\_dat.txt",header=T)

 x = dat[,3]

 y = dat[,4]

 lm.1 = lm(y ~ x + I(x^2)) #1pt for correct formula in lm(). No point for missing the I() or no square term

 SST = sum((y - mean(y))^2) # 3417710 #0.5pt for correct SST

 SS\_exp = sum((predict(lm.1) - mean(y))^2) # 2192907 #0.5pt for correct SSE

 SS\_unexp = sum( (y - predict(lm.1))^2 ) # 1224803 #0.5pt for correct SS\_unexp

#okay if used SST = SS\_exp + SSE to calculate one of the quantities

#No point for SS calculations if used anova, not calculated through formula

#-0.25 if numerical values not reported

2.5pts total

b) Write code to compute and report the cross-term for the anova decomposition in the previous part. Your answer may not be zero identically due to machine precision.

 sum( (predict(lm.1) - mean(y)) \* (y - predict(lm.1)) ) # -6.072682e-09

#1pt for correct cross-term calculation 0.5 partial

#give 0.5 if used anova() to get the cross-term value. In this question we give partial credits since not explicitly stated to use formula

#-0.25 if numerical value not reported

c) Write code to generate data (n=100) on x1, x2, and y, with some collinearity (correlation coeff about 0.5), following the equation y = 1 + 2\* x1 + 3\* x2, with NO error in y, and make the relevant scatterplots. YOU may choose the values of any other parameters for this simulation.

 library(MASS) # This library contains mvrnorm(); see

 set.seed(1) # Not necessary; only for ensuring reproducible output.

 n = 100

 r = 0.5

 dat = mvrnorm(n, rep(0, 2), matrix(c(1, r, r, 1), 2, 2)) #1pt for correct mvrnorm(), -0.5 if any argument wrong (n and r)

 x1 = dat[, 1]

 x2 = dat[, 2]

 y = 1 + 2\*x1 + 3\*x2 + rnorm(n, 0, 0) # Generate y, but with no error/noise. May skip rnorm() completely. #1pt for correctly computing y .leaving out rnorm is OK too #-0.5 if have error in y

 dat = data.frame(x1, x2, y) # Here is the whole data. #0.5pt for correctly organizing the data

 plot(dat) #0.5pt for making the relevant plot

3pts total

d) Write code to fit a model y = alpha + beta1\* x1 + beta2\* x2 to the resulting data, and report its R2 and s\_e.

 lm.1 = lm( y ~ x1 + x2) # Fit a plane through the data.

 summary(lm.1) # R2 = 1, s\_e = 0

#1pt for correct model. -0.5 if not reporting the values

e) Given the presence of unambiguous scatter in all of the scatterplots in the previous parts, why are the fits perfect (i.e., R^2 = 1, s\_e = 0)? Explain in words. Hint: visualize the data in 3d. No code needed.

Because the data was made to lie on a plain exactly;

That's what "No error" means in part c.

FYI: Note that the reason is NOT overfitting.

#1pt for correct explanation

f) Write code to generate data (n=1000) on x1, x2, and y, with little collinearity (correlation coeff about 0.1), following the equation y = 1 + 2 x1 + 3 x2 + 4 x1 x2, again with NO error in y, and make the relevant scatterplots. YOU may choose the values of any other parameters for this simulation.

 library(MASS) # This library contains mvrnorm(); see

 set.seed(1) # Again, not necessary.

 n = 1000

 r = 0.1

 dat = mvrnorm(n, rep(0, 2), matrix(c(1, r, r, 1), 2, 2)) #1pt for correct arguments. -0.5 point if any argument wrong (n and r)

 x1 = dat[, 1]

 x2 = dat[, 2]

 y = 1 + 2\*x1 + 3\*x2 + 4\*x1\*x2 + rnorm(n, 0, 0) # Generate y, and add NO error. #0.5pt for correct model #-0.5 if have error in y

 dat = data.frame(x1, x2, y) # Here is the whole data.

 plot(dat)

1.5pts total

g) Write code to fit a model y = alpha + beta1 x1 + beta2 x2 + beta3 x1 x2 (i.e., with interaction) to the resulting data, and report its R2 and s\_e.

 summary(lm( y ~ x1 + x2 + I(x1\*x2) )) # R2 = 1, s\_e = 0

#1pt for correct model. -0.5 if value not reported

h) Hopefully, you'll see a bowtie/butterfly pattern in the y-x1 and/or y-x2 plains in the previous part. Even if you don't, say something that would explain a bowtie/butterfly pattern in in the in y-x1 or y-x2 plains. Hint: visualize the data in 3d.

That pattern is a result of the interaction term.

Projecting a saddle surface in 3d onto the two plains gives rise to that pattern.

#1pt for correct explanation mentioning “interaction term” or “saddle surface”.