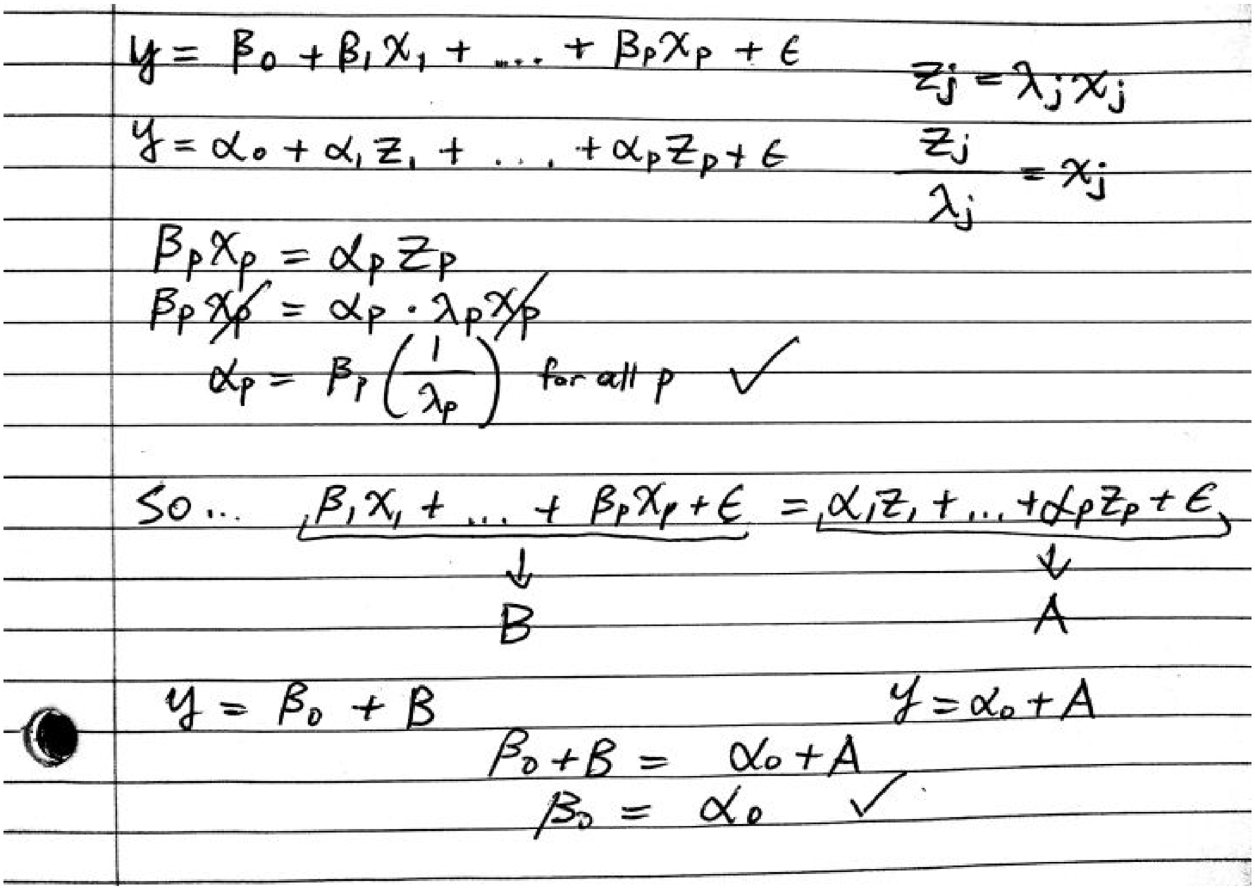
Wilson Wang

Problem 1: James Chapter 3, Exercise 4.

a. It is given that the true relationship between X and Y is linear. So, I would expect the least squares line to be closer to the actual true regression line. So, the training RSS for the linear regression should be less than the RSS for the cubic regression.
b. Since we now depend on the test data, we need more information, but the cubic regression will suffer from overfit from training and hence may have a higher test RSS.
c. The cubic regression has a higher flexibility and than the linear fit and should therefore have a lower RSS.
d. Still not enough information for the test set since we don't know how close it is to linear. So it really depends on whether the true relationship is cubic or linear.

Problem 2:

a.



$y = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p + \epsilon$ $\qquad z_j = \lambda_j x_j$

$y = \alpha_0 + \alpha_1 z_1 + \dots + \alpha_p z_p + \epsilon$ $\qquad \frac{z_j}{\lambda_j} = x_j$

$\beta_p x_p = \alpha_p z_p$

$\beta_p \cancel{x_p} = \alpha_p \cdot \lambda_p \cancel{x_p}$

$\alpha_p = \beta_p \left(\frac{1}{\lambda_p}\right)$ for all p ✓

So... $\underbrace{\beta_1 x_1 + \dots + \beta_p x_p + \epsilon}_{\downarrow \atop B} = \underbrace{\alpha_1 z_1 + \dots + \alpha_p z_p + \epsilon}_{\downarrow \atop A}$

$y = \beta_0 + B$ $\qquad y = \alpha_0 + A$

$\beta_0 + B = \alpha_0 + A$

$\beta_0 = \alpha_0$ ✓

b. No. In linear regression, we already have beta values that influence how significant a data point in a feature will influence our regression calculation. Hence, we won’t actually need to scale the relative numerical sizes of the features.

Problem 3:

a. In the beginning I had a linear regression equation with all the values in question. With all these values, the linear model only had the query variable as one with optimal p-value. Noticing that the largest p-value resided in CPI.Energy and Unemployment, I removed those from my equation to get a Linear regression equation: lm(formula = ElantraSales ~ ElantraQueries + CPI.All, data = elantra.train). I observed the p-values for each independent variable and looked for the ones with low p-values to keep. On the other hand I sought to get rid of the variables with higher p-values. In the end, we got very optimal p-values for ElantraQueries and CPI.All (all very close to 0). CPI.all and CPI.energy were very closely related, which brought out a concern for multicollinearity between the two variables. After running the linear model and finding optimal p-values, I can safely conclude that ElantraQueries and CPI.All are significant and my equation results to: Y = -65668.40 + 135.10 * ElantraQueries + 324.42 * CPI.All. Let's analyze the coefficients. The intercept (-65668.40) is negative and makes sense because if the queries and CPI.all were 0, then no one would be interested/buying Elantra's so Hyundai would lose money after making all those Elantra cars. The coefficients for our variables are positive because the more positive the queries/CPI.all are, the higher the resulting sales will be. These factors have a positive impact on Elantra sales. We can look at our r-squared to see how well the model fits our data set. Our r-squared value is 0.4667 which is not very good. This means that the model doesn't predict training set observations very well.
b. The new regression model has some changes, but keeps the same variables we used previously. However, it is noticeable that the p-value of ElantraQueries has risen significantly to 0.06. However, CPI.All is still pretty good. Here is the regression equation:

Y = -49905.66 + MonthFactorAugust * -187.20 + MonthFactorDecember * -2919.18 + MonthFactorFebruary * -5183.31 * MonthFactorJanuary * -7004.52 + MonthFactorJuly * 256.01 + MonthFactorJune * 119.68 + MonthFactorMarch * 1332.26 + MonthFactorMay * 467.55 + MonthFactorNovember * -5403.60 + MonthFactorOctober * -6195.88 + MonthFactorSeptember * -2715.19 + ElantraQueries * 45.41+ CPI.All * 281.41+ CPI.Energy * 25.91+ Unemployment * -548.44.

Each month variable has a different coefficient to signify how likely a customer will buy a Elantra in that month. For example January has a very negative coefficient value because customers tend to not buy new cars in January. On the other hand, customers seem to like buying cars in March so March has a high coefficient value. The training set R-squared is 0.7084. The most significant variables are the month factors from September to February. These p-values were below 0.1. Adding the independent variable MonthFactor certainly increased the R-squared value pretty significantly, so it did improve the quality of the model. All significant variables were also month related, which indicated how important the monthfactors are in comparison to other variables. An alternative way to modelling seasonality is through a time series. This is a better visualization of the upward and downward trends of car sales per month. Maybe an alternative way would be grouping together months of a given season and analyzing trends based on each season's months.

c. I ended up sticking with MonthFactors, ElantraQueries, and CPI.All as my variables since all ended up having p-values less than 0.1 after removing CPI.Energy and Unemployment (which had p-values that were very large). The training set's R-squared is 0.716 and the OSR-squared is -4.728011. Since OSR is significantly smaller, this is a sign of overfitting. In addition, the R-squared is still pretty low, so I don't think the model will be useful for Hyundai.

d. After experimenting with the variables, I ended up sticking with HondaQueries, CPI.All, and Unemployment because the model with these variables produced the best r-squared value. The r-squared value is 0.3331and the OSR-squared value is -2.574026. The performance of the Honda model is worse than the performance of the Elantra model.

e. The model predicts 29178.23 sales for Elantra and 15309.798 sales for Honda. This is probably a severe overestimation.

```
#
# Make sure your working directory was set as "Source file location".
# Make sure your data files are in the same folder as this R script.
#
install.packages(c("dplyr", "ggplot2", "GGally"))
library(dplyr)
library(ggplot2)
library(GGally)
install.packages("car")
library(car) # for VIF

# Load data:
elantra <- read.csv("Elantra142-Fall2018.csv")

str(elantra)
head(elantra)

# Plot scatter matrix
ggscatmat(elantra, columns = 0:8, alpha = 0.8)


# split into training and test sets

elantra.train <- filter(elantra, Year <= 2015)
head(elantra.train)
tail(elantra.train)
elantra.test <- filter(elantra, Year <= 2018 & Year >= 2016)

# train the model
#lm(y~x1+x2+...,data)
help(lm)
mod1 <- lm(ElantraSales ~ ElantraQueries + CPI.All, data =
elantra.train)
summary(mod1)

# A better model...
mod2 <- lm(ElantraSales ~ MonthFactor + ElantraQueries + CPI.All +
CPI.Energy + Unemployment, data = elantra.train)
summary(mod2)
vif(mod2)

mod3 <- lm(ElantraSales ~ MonthFactor + ElantraQueries + CPI.All, data =
elantra.train)
summary(mod3)
vif(mod3)

# compute OSR^2

# this builds a vector of predicted values on the test set
SSE = sum((elantra.test$ElantraSales - elantraPredictions)^2)
```

```
SST = sum((elantra.test$ElantraSales -
mean(elantra.train$ElantraSales))^2)
OSR2 = 1 - SSE/SST

# Load NEW HONDA data:
honda <- read.csv("HondaOdysseySales.csv")

str(honda)
head(honda)

# Plot scatter matrix
ggscatmat(honda, columns = 0:10, alpha = 0.8)


# split into training and test sets

honda.train <- filter(honda, Year <= 2015)
head(honda.train)
tail(honda.train)
honda.test <- filter(honda, Year <= 2018 & Year >= 2016)

# train the model
mod4 <- lm(HondaSales ~ MonthFactor + HondaQueries + CPI.All +
Unemployment, data = honda.train)
summary(mod4)

# compute OSR^2

# this builds a vector of predicted values on the test set
SSE = sum((honda.test$HondaSales - hondaPredictions)^2)
SST = sum((honda.test$HondaSales - mean(honda.train$HondaSales))^2)
OSR2 = 1 - SSE/SST
OSR2

AugustEst <- read.csv("AugustEst.csv")

AugustEst.train <- filter(AugustEst, Year <= 2015)
head(AugustEst.train)
tail(AugustEst.train)
AugustEst.test <- filter(AugustEst, Year <= 2018 & Year >= 2016)

# train the model
elantramod <- lm(ElantraSales ~ MonthFactor + ElantraQueries + CPI.All,
data = AugustEst.train)
summary(elantramod)
hondamod <- lm(HondaSales ~ MonthFactor + HondaQueries + CPI.All +
Unemployment, data = AugustEst.train)
summary(hondamod)


elantraNewPredictions <- predict(elantramod, newdata=AugustEst.test)
```

```
hondaNewPredictions <- predict(hondamod, newdata=AugustEst.test)
elantraNewPredictions
hondaNewPredictions
```