University of California, Davis

Final Project

Building a Model using Multiple Linear Regression

Ana Boeriu
aiboeriu@ucdavis.edu
STA 206
Dr. Jie Peng
December 13, 2020

Abstract

Using multiple linear regression, the association of beta carotene with twelve predictor variables were studied in 315 patients who had an elective surgical procedure during a three-year period to biopsy, or removal of a lesion that was found to be noncancerous. Beta carotene plasma was negatively related to body mass index (BMI) and positively related to grams of fiber consumed per day. Furthermore, out of the 13 males who never smoked only six take supplemental vitamins compared to the 104 females who take vitamins out of 143 females who have never smoked.

Introduction

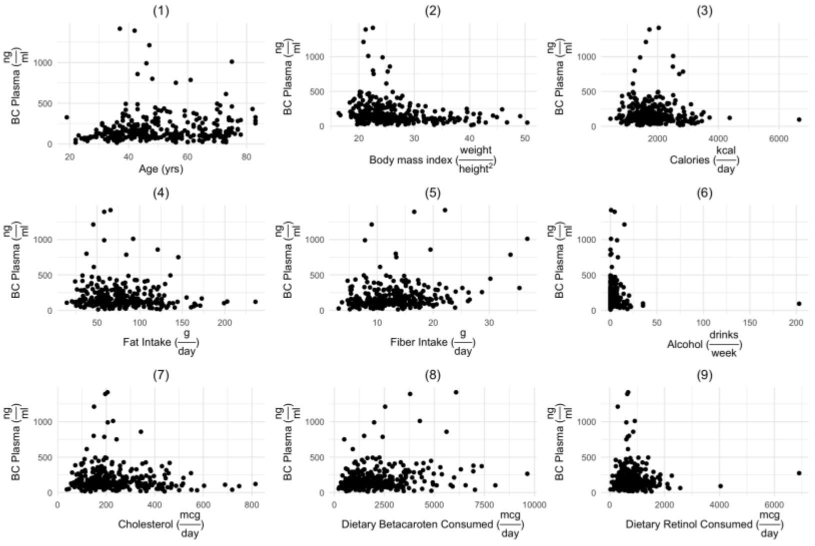
Observational studies have suggested that low dietary intake or low plasma concentrations of beta-carotene, or other carotenoids might be associated with increased risk of developing certain types of cancer. However, few studies have investigated the determinants of plasma concentrations of these micronutrients. The goal of this project is to build a model which can model the amount of beta carotene plasma based off of physiological characteristics from the subjects. We are using the approach of multiple linear regression to build a “correct” model which is focused on accuracy as there will only be variables that have a significant relationship with beta carotene plasma count, which is our Y. A full description of the data can be seen in the appendix. We will then perform diagnostics that assesses how well the model we choose meets the assumptions for multiple linear regression and if necessary, transform the data. Because there are so many different variables and models we could choose from, we will perform a series of model selection techniques to find the set of variables that will give us the best model that meets our goal of correctness. We will be using BIC as our model criteria and Forwards Backwards Selection as the subset selection and verify the robustness of our model using model validation. We will use the model to calculate predictions that provide insight about the beta carotene plasma levels. Doctors or nutritionists may be interested in this model to see how people can improve their overall health and also reduce the risk of cancer. Anybody may be interested in using this model, especially those who must carefully monitor their beta plasma carotene levels.

Methods and Results

In this section we will be discussing in detail our analysis supporting our conclusions with statistical tests and plots.

A. Plots Y vs X_i

The scatterplots below show the relationships between the variables and beta carotene plasma cell count.



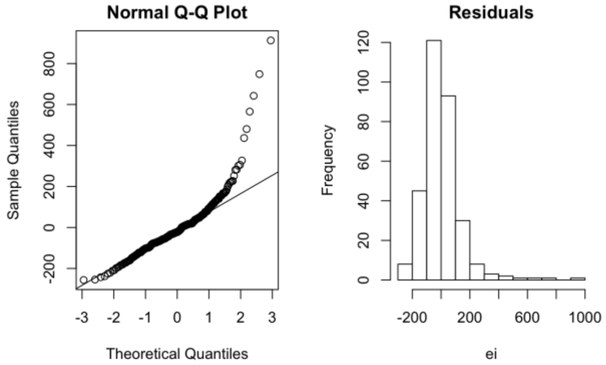
Graph 6 seems to have an outlier of about 200 alcoholic drinks per week. Upon further investigation, the individual (65-year-old male) consumed 203 alcoholic beverages per week. Given that an individual may sleep between six to nine hours each day this suggests that an alcoholic beverage is consumed every 30 to 40 minutes each day. Furthermore, there are two data point in graph 9 that could be potential outliers. Notice that one of the data points is above 6000 micrograms. The tolerable upper intake level for adults is 3,000 micrograms. Any amount higher than that is considered toxic and can be fatal. We will not remove these data entries yet as we do not know if they are truly outliers of just highly influential points.

B. Diagnostics

Diagnostics are performed to evaluate how well the data meets the assumptions needed for multiple regression, which are that observations are independent, errors have constant variance, and they are normally distributed. We will look at different tests for these assumptions and note any possible outliers. Typically, we would transform data to correct for non-normality, non-linearity, or non-constant variance if necessary.

1. Testing for Normality

i. QQ Plot and Histogram of errors



This QQ plot has quite a few data points not on the Y=X line which signifies non-normal errors. Furthermore, the histogram does not look approximately normal.

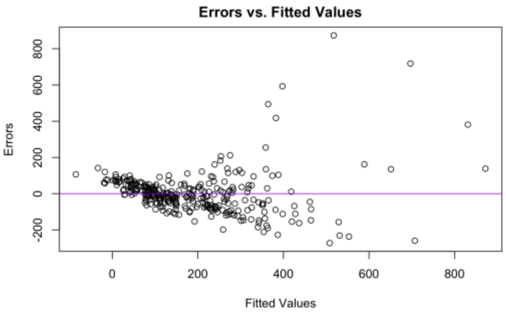
ii. **Shapiro Wilks Normality Test**

$H_o: Data\ is\ normal \quad H_A: Data\ is\ non-normal$

Since our p-value of 2.2×10^{-16} is less than alpha of 0.05, we reject the null hypothesis and conclude that the data is not normally distributed.

2. **Testing for Constant Variance**

i. **Errors vs Fitted Values**



ii. **Fligner Killeen Heteroscedasticity Test**

$H_o: Residuals\ have\ equal\ variance \quad H_a: Residuals\ do\ not\ have\ equal\ variance$

Since our p-value of 2.844×10^{-9} is less than alpha of 0.05, we reject the null hypothesis and conclude that the residuals do not have constant variance.

C. **Data Transformation**

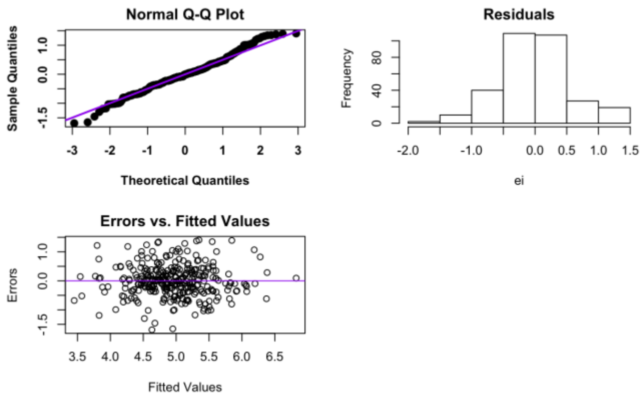
In this section we will transform the data to correct for non-normality and non-constant variance as the assumptions for multiple linear regression have not been met. We will use the Box-Cox transformation on our Y variable. We will then look for outliers as many times transforming the data first will also help adjust for outliers.

1. **Box Cox**

Box-Cox transformations use $\frac{X^{\lambda-1}}{\lambda}$ to find the lambda that maximizes log likelihood. The value of λ that maximizes the log likelihood is $\lambda = 0.02$ which we will round down to 0. Thus, we will perform a log transformation to our Y.

2. **Verifying the Assumptions of Multiple Linear Regression**

Here we will check to see if the log transformation meets the assumptions of multiple linear regression which are that observations are independent, errors have constant variance, and they are normally distributed.

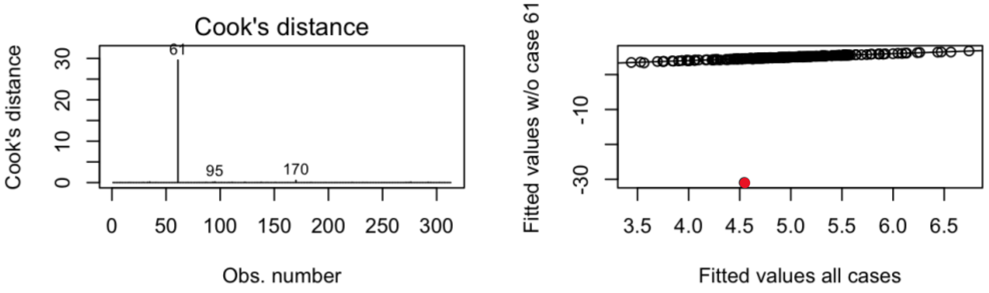


The plots shown above do seem to meet the assumptions of normality. The QQ plot does show the majority of data points on the Y=X line which signifies normal errors. Furthermore, our p-value for Shapiro-Wilks normality test is 0.01625. Even though at an alpha of 0.05 we conclude that data is not normally distributed, the shapiro's test can be overly conservative and reject the null unnecessarily. For Fligner Killeen's heteroscedasticity test the p-value is 0.2041. Thus, we fail to reject the null and conclude that errors have constant variance.

3. Outliers

Our next step was to look for and remove any outliers so that we could further meet the assumptions of linear regression. Using an alpha value of 0.1, we found an outlier with the following criteria: any 39-year-old male with a beta carotene plasma count of 417.8 (nanograms per milliliter) who currently smokes, takes vitamins often, has a body mass index of 22, and does not consume alcohol. Furthermore, any patient who meets this criterion and also has a daily nutrient intake of 1951.4 calories, 109.1 grams of fat, 4.7 grams of fiber, 461.1 milligrams of cholesterol, 998 micrograms of beta carotene, 588 micrograms of retinol. We will remove this outlier so that it does not skew the data. We also looked for leverage points as those have an especially large influence on the regression line and can signify an outlier. However, it is important to note that leverage points are not always outliers. Using a more conservative threshold of $D_i > 1$ we found one influential point.

i. Cooks distance Plot and Fitted Values Plot



Examining the cook's distance plot, we can see that case 61 is a highly influential data point. This data point corresponds to a 65-year-old male who has an alcohol intake of 203 alcoholic drinks per week among other physiological and nutritional characteristics. As we can see in the fitted values plot, the model is affected by case 61. As shown in the picture, there is some difference between the two fits and this data point should be removed. Thus, we removed less than 1% of our original data.

D. Model Selection

In this section we are using stepwise regression and a specific subset selection process to see which combination of explanatory variables will produce the best model based off of the criteria we have chosen to use which is BIC. There are many variables in our model, and we must make sure the ones in our model are significant. We looked at the model chosen when using all subsets and then Forwards Backwards Selection.

1. Model Criteria

Because our goal is to have a correct model, we are choosing only explanatory variables which have a significant relationship with beta carotene plasma count. This model may be smaller than ones which have the goal of prediction. In general, AIC or BIC are often used as model selection criteria for correctness as they penalize large models. AIC may overfit correct models and BIC penalizes large models even more, so we chose to focus on BIC as we imagine if somebody is trying to see a person's beta carotene plasma count to prescribe a medication for example, they will really need the result to be correct. We choose the model that lowers BIC the most.

2. Model Stepwise Selection

Subset selection does not evaluate all possible models; however, it is faster and does not cost as much because not all subsets are calculated. We chose to use Forwards Backwards Selection because we want a correct model which means having a smaller one. Of course, underfitting a model is not ideal however between underfitting and overfitting, underfitting the model is more likely to achieve the smaller and often more correct model which Forwards Selection or Forwards Backwards Selection is more likely to do. Forwards Backwards Selection does not underfit as much as Forwards Selection though, so we used Forwards Backwards as our subset selection. In order to be able to show the robustness of the model we will split our data into a training dataset and validation dataset. While in many cases deciding how to divide the data can be quite subjective, as models can change based on the division of data, we will focus on choosing a model that has the least difference in values between the training dataset and testing dataset. The model below is the one that lowers BIC the most when using Forwards Backwards selection using 55% of the original data in the training dataset and 45% in the validation dataset.

Model	P (# of β)	BIC
$Y \sim X_{BMI} + X_{vituse}$	4	-125.71

3. Model Validation

The goal is to have similar estimated regression coefficients, similar standard errors, and a small difference between MSPE.

	Train Estimates	Validation Estimates	Train standard errors	Validation standard errors
Intercept	5.403	5.834	0.233	0.313
BMI	-0.026	-0.041	0.008	0.011
Vitamin Use (occasionally)	0.305	0.265	0.128	0.167
Vitamin Use (often)	0.459	0.2946	0.117	0.149

Since all of the estimated coefficients as well as their standard errors agree quite closely on the two datasets, our model is not sensitive to different datasets. Regardless of which dataset we use, either the training or validation, we will obtain similar estimates of regression coefficients, implying that there is consistency in parameter estimation. SSE and $R^2_{adj}$are also quite close (see appendix for details). Lastly, we obtained a mean square prediction error (MSPE) value of 0.469 and an (SSE/n) value of 0.237. Since these two values are quite similar, we can conclude that there is no severe overfitting. Ideally, we would want our mean square prediction error to be close to zero since that would mean that our prediction is close to the original value. However, since our model is quite small, and it is not a predictive model (only includes important X's) its MSPE will tend to be a bit larger.

4. Final Model

$$log(Y) = 5.587935 - 0.032767X_{BMI} + \ 0.294263X_{V,not\ often} + 0.388381X_{V,often}$$

$\widehat{\beta_0}$: There is no reasonable interpretation since body mass index cannot be zero.
$\widehat{\beta_1}$: When a patient's BMI increases by 1 unit, their beta plasma count will decrease by about 0.968 (ng/mL) on average, holding all other variables constant
$\widehat{\beta_2}$: Patients who occasionally take vitamins, had a higher beta carotene plasma count on average compared to those who do not take vitamins by 1.34 (ng/ml) holding all other variables constant.
$\widehat{\beta_3}$ Patients who often take vitamins, had a higher beta carotene plasma count on average compared to those who do not take vitamins by 1.6 (ng/ml) holding all other variables constant.

E. Prediction Intervals

To practice using our model we calculated and interpreted intervals for a patient with a body mass index of 27 who does not take vitamins, a patient with a body mass index of 19 who takes vitamins fairly often, and a patient with a body mass index of 31 who takes vitamins occasionally. We will use the Bonferroni multiplier to adjust our intervals.

BMI	Vitamin Use	estimate	95% lower	95% Upper
27	no	4.703231	3.010061	6.396400
19	Fairly often	5.353746	3.658081	7.049411
31	Not often	4.866426	3.169241	6.563611

We are 95% overall confident that for a patient who doesn't take vitamins and has a BMI of 27, his beta carotene plasma count is between 3.01 and 6.4 nanograms/liter. The actual estimated value is 4.7 nanograms per milliliter which falls in that range.

We are 95% simultaneously confident that a patient who has a BMI of 19 and taker her vitamins fairly often, her plasma beta carotene cell count is between 3.658 and 7.049 nanograms per milliliters. The actual estimated value of 5.35 confirms the interval is in the correct range.

We are familywise 95% confident that for a patient who occasionally takes vitamins and that has a BMI of 31 their beta carotene plasma count is between 3.16 and 6.56. The actual estimated value of 4.86 fall well within the specified range.

Ideally, we would want to have an independent dataset to perform analysis and inference. However, since we do not, we performed inference on the same dataset. While this is commonly used in practice, it is not entirely correct. When we performed our prediction intervals, although we used a 95% corrected interval, the interval may actually cover less than 95%.

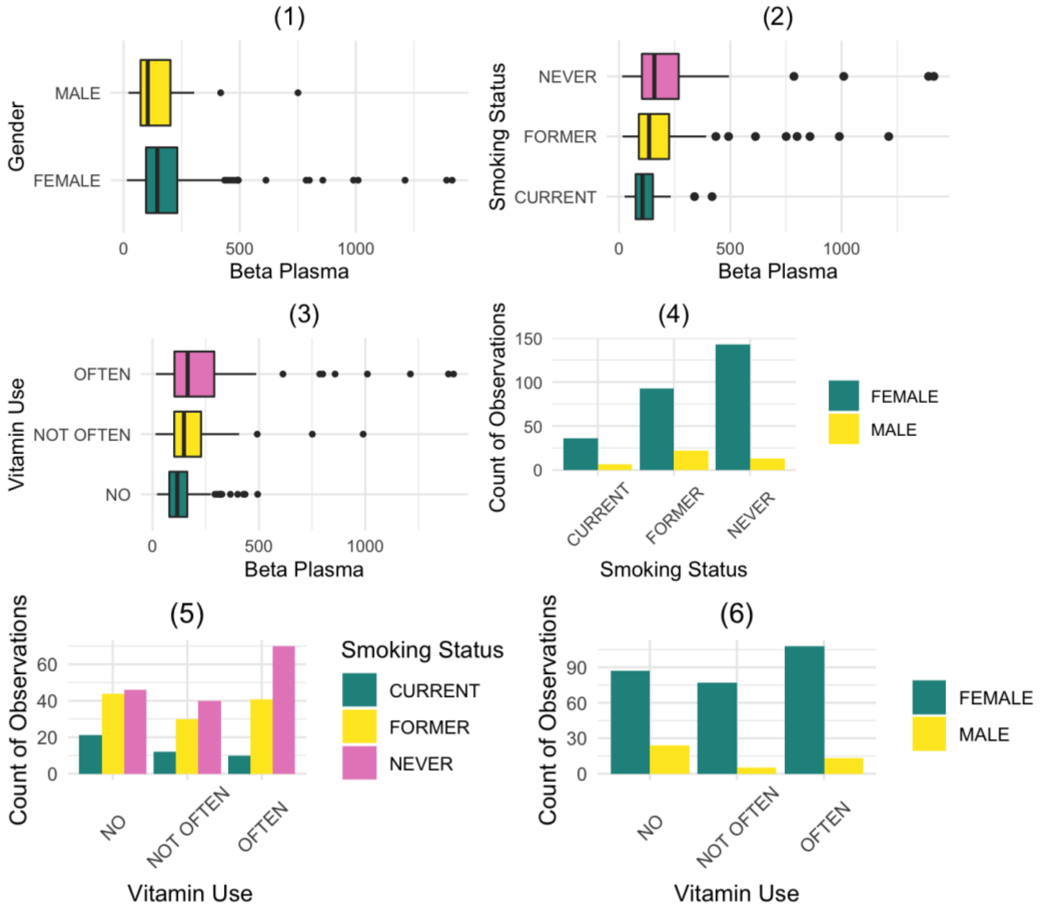
Conclusion and Discussion

In this project we started with a dataset from a medical study and decided to build a model geared towards correctness rather than prediction which often will produce a smaller model. Through scatterplots we were able to identify any potential outliers and the relationship between the response variable and the predictor variable. We then verified the assumptions of multiple linear regression and transformed our data to meet those assumptions of normality and constant variance. With this transformed data we removed an outlier and checked for leverage points. Next, we chose a best model using model validation to check the robustness of our best model. Since data is often messy and variables may be present when they really aren't significant, we proceeded with a model selection technique of stepwise regression to find which variables did not need to be in our model. We decided to use BIC as the model criteria and Forwards Backwards Selection subset selection as these helps achieve our goal of building a model that is more correct in predicting beta carotene plasma count. This showed us that the significant variables are "bmi" and "vitamin use" so these were the two variables we used in our revised model. With this new model, we predicted beta carotene plasma count using various scenarios. Doctors, nutritionists, or anyone who want to monitor their beta carotene plasma count could hopefully find this information useful.

Appendix

A. Boxplots

The box plots provide information on how the different variables interact with red blood cell count, such as the mean, amount of red blood cell count at different quartiles, and outliers.



Because gender is a categorical variable, this is a grouped box plot which shows information of red blood cell count for males and females. One can see that females on average have higher beta plasma cell counts than females. Furthermore, patients who never smoked have a slightly higher beta carotene plasma count on average. As seen in graph 4, this study consisted of predominantly females that never smoked. Similarly, most of the males in this study were former smokers. Patients who often took vitamins never smoked, while those who do not take supplemental vitamins had a similar number of nonsmokers (46) and former smokers (44).

D. Model Selection

3. Model Validation

Recall that the goal was to choose a data division that had a good number of data points in each dataset and also produced a model that had the least difference in parameter estimation, SSE, R^2_{adj} and MSPE vs SSE/n. Investigating on how to split the data, we concluded that the best data division was 55% of observations in the training and 45% in the validation.

	SSE	R_{adj}^2
Train	71.678	0.119
Validation	76.293	0.117

We can see that SSE and R_{adj}^2 are quite similar. Notice that the SSE in the validation dataset is larger than in the training dataset. The reason is that there is more error of fitting a model on new data. Note that we used 55% of our data to create a best model and then tested that model on the other 45% of the data.

R appendix

Read in the Data

```
plasma =
read.table("/Users/anab/Documents/MS_UCDavis/STA206/final_
proj/Plasma.txt", header = TRUE)
head(plasma)
str(plasma)
plasma$SEX = as.factor(plasma$SEX)
plasma$SMOKSTAT = as.factor(plasma$SMOKSTAT)
plasma$VITUSE = as.factor(plasma$VITUSE)
str(plasma)
names(plasma)[4] = "BMI"
beta_c_plasma = plasma[,1:13]
#change the order of variables, betaplasma 1st and categorical
next then numeric
beta_c_plasma = beta_c_plasma[c("BETAPLASMA", "SEX",
"SMOKSTAT", "VITUSE", "AGE", "BMI","CALORIES","FAT",
"FIBER","ALCOHOL","CHOLESTEROL" , "BETADIET" ,"RETDIET" )]
head(beta_c_plasma)
str(beta_c_plasma)
summary(beta_c_plasma[1:4])
summary(beta_c_plasma[5:10])
summary(beta_c_plasma[11:13])
names(beta_c_plasma) =tolower(names(beta_c_plasma))
head(beta_c_plasma)
bplasma = subset(beta_c_plasma, betaplasma!=0)
```

Summary Stat

```
Summary(bplasma)
```

Plots Y vs X_i

Scatterplots

```
library(latex2exp)
library(ggplot2)
library(gridExtra)
g1 =ggplot(bplasma)+
geom_point(aes(x = age , y= betaplasma))+xlab(TeX("Age (yrs)"))+
  ylab(TeX("BC Plasma ($\\frac{ng}{ml}$)"))+ggtitle("(1)")+
  theme_minimal()+theme(plot.title = element_text(hjust = 0.5))
 g2 = ggplot(bplasma)+
  geom_point(aes(x = bmi , y= betaplasma))+
  xlab(TeX("Body mass index ($\\frac{weight}{height^{2}}$)"))+
  ylab(TeX("BC Plasma ($\\frac{ng}{ml}$)"))+
   theme_minimal()+ggtitle("(2)")+
   theme(plot.title = element_text(hjust = 0.5))
 g3 = ggplot(bplasma)+
    geom_point(aes(x = calories , y= betaplasma))+
    xlab(TeX("Calories ($\\frac{kcal}{day}$)"))+
   ylab(TeX("BC Plasma ($\\frac{ng}{ml}$)"))+ggtitle("(3)")+
    theme_minimal()+theme(plot.title = element_text(hjust = 0.5))
   g4 = ggplot(bplasma)+
    geom_point(aes(x = fat , y= betaplasma))+
    xlab(TeX("Fat Intake ($\\frac{g}{day}$)"))+
    ylab(TeX("BC Plasma ($\\frac{ng}{ml}$)"))+ ggtitle("(4)")+
    theme_minimal()+theme(plot.title = element_text(hjust = 0.5))
   g5 = ggplot(bplasma)+
    geom_point(aes(x = fiber , y= betaplasma))+
    xlab(TeX("Fiber Intake ($\\frac{g}{day}$)"))+
    ylab(TeX("BC Plasma ($\\frac{ng}{ml}$)"))+ggtitle("(5)")+
    theme_minimal()+theme(plot.title = element_text(hjust = 0.5))
   g6 = ggplot(bplasma)+
    geom_point(aes(x = alcohol , y= betaplasma))+
    xlab(TeX("Alcohol ($\\frac{drinks}{week}$)"))+
    ylab(TeX("BC Plasma ($\\frac{ng}{ml}$)"))+ggtitle("(6)")+
    theme_minimal()+theme(plot.title = element_text(hjust = 0.5))
      g7 = ggplot(bplasma)+
    geom_point(aes(x = cholesterol , y= betaplasma))+
    xlab(TeX("Cholesterol ($\\frac{mcg}{day}$)"))+
    ylab(TeX("BC Plasma ($\\frac{ng}{ml}$)"))+ggtitle("(7)")+
    theme_minimal()+theme(plot.title = element_text(hjust =
  0.5))
  g8 = ggplot(bplasma)+
    geom_point(aes(x = betadiet , y= betaplasma))+
    xlab(TeX("Dietary Betacaroten Consumed
  ($\\frac{mcg}{day}$)"))+
    ylab(TeX("BC Plasma ($\\frac{ng}{ml}$)"))+ggtitle("(8)")+
    theme_minimal()+theme(plot.title = element_text(hjust =
  0.5))
   g9 = ggplot(bplasma)+
    geom_point(aes(x = retdiet , y= betaplasma)) +
    xlab(TeX("Dietary Retinol Consumed
  ($\\frac{mcg}{day}$)"))+
    ylab(TeX("BC Plasma ($\\frac{ng}{ml}$)"))+ggtitle("(9)")+
    theme_minimal()+theme(plot.title = element_text(hjust =
  0.5))
  grid.arrange(g1,g2,g3,g4,g5,g6,g7,g8,g9, nrow = 3, ncol = 3)
```

Boxplots

```
library(grid.text)
# "#21908CFF" aqua
# "#FDE725FF" yellow
# "#E78AC3" pink
# "#66C2A5 light green
gender_bc =  ggplot(bplasma) +
  geom_boxplot(aes(x = sex, y = betaplasma,
fill=sex),outlier.size = 1) +
   labs(title = "(1)")+ xlab("Gender") + ylab("Beta Plasma")+
  coord_flip() +
  scale_fill_manual(values=c("#21908CFF", "#FDE725FF",
"#E78AC3"))+
theme_minimal()+theme(legend.position =
'none')+theme(plot.title = element_text(hjust = 0.5))
 smoke = ggplot(data = bplasma) +
  geom_boxplot(aes(x = smokstat, y = betaplasma, fill=smokstat))
+
  labs(title = "(2)")+ ylab("Beta Plasma") + xlab("Smoking Status")+
  coord_flip() +
  scale_fill_manual(values=c("#21908CFF", "#FDE725FF",
"#E78AC3"))+theme_minimal()+
   theme(legend.position = 'none') +theme(plot.title =
element_text(hjust = 0.5))
 vitaminUse  = ggplot(data = bplasma, aes(x = vituse, y =
betaplasma)) +
  geom_boxplot(outlier.size = 1, aes(fill=vituse)) +
  labs(title = "(3)")+ xlab("Vitamin Use") + ylab("Beta Plasma")+
  coord_flip() +
  scale_fill_manual(values=c("#21908CFF", "#FDE725FF",
"#E78AC3"))+ theme_minimal()+
  theme(legend.position="none")+theme(plot.title =
element_text(hjust = 0.5))
 bar = ggplot(bplasma)+
  geom_bar(aes(x = smokstat, fill = sex), position = "dodge")+
  scale_fill_manual(values=c("#21908CFF", "#FDE725FF",
"#E78AC3"))+
  labs(title = "(4)", y = "Count of Observations", x = "Smoking
Status")+theme_minimal()+
  theme(legend.title = element_blank()) +theme(plot.title =
element_text(hjust = 0.5))+
  theme(axis.text.x = element_text(angle = 45, vjust = 0.6))
 bar2 = ggplot(bplasma)+
  geom_bar(aes(x = vituse , fill = smokstat ), position = "dodge")+
  scale_fill_manual(values=c("#21908CFF", "#FDE725FF", "#E78AC3"))+
  labs(title = "(5)", y = "Count of Observations", x = " Vitamin Use")+
  theme_minimal()+labs(fill="Smoking Status")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme(axis.text.x = element_text(angle = 45, vjust = 0.6))
 bar3 = ggplot(bplasma)+
  geom_bar(aes(x = vituse, fill = sex), position = "dodge")+
  scale_fill_manual(values=c("#21908CFF", "#FDE725FF", "#E78AC3"))+
  labs(title = "(6)", y = "Count of Observations", x = "Vitamin Use")+theme_minimal()+
  theme(legend.title = element_blank()) +theme(plot.title = element_text(hjust = 0.5))+
  theme(axis.text.x = element_text(angle = 45, vjust = 0.6))
 #BoxPlots Graph Display
 grid.arrange(gender_bc, smoke, vitaminUse,bar,bar2,bar3, ncol=3, nrow = 2)
```

Diagnostics

Testing for Normality

QQ Plot and Histogram of errors

```
initial_model = lm(betaplasma~.^2, data = bplasma)
summary(model)
ength(model$residuals)
par(mfrow = c(1,2))
#check to see if data is normally distributed
qqnorm(initial_model$residuals)
qqline(initial_model$residuals)
hist(initial_model$residuals, main = "Residuals", xlab = "ei",pch = 16,cex = 1.25, col = "white")
```

Shapiro Wilks Normality Test

```
ei= initial_model$residuals
the.SWtest = shapiro.test(ei)
the.SWtest
```

Testing for Constant Variance

Errors vs Fitted Values

```
plot(initial_model$fitted.values, initial_model$residuals, main = "Errors vs. Fitted Values",xlab = "Fitted Values",  ylab = "Errors")
abline(h = 0,col = "purple")
```

Fligner Killeen Heteroscedasticity Test

```
Group = rep("Lower",nrow(bplasma)) #Creates a vector that repeats "Lower" n times
Group[bplasma$betaplasma > median(bplasma$betaplasma)] = "Upper" #Changing the appropriate values to "Upper"
Group = as.factor(Group) #Changes it to a factor, which R recognizes as a grouping variable.
bplasma$Group = Group
the.FKtest= fligner.test(initial_model$residuals, bplasma$Group)
the.FKtest
bplasma$Group = NULL
```

Data Transformation

Box Cox

```
boxCox1 = boxcox(initial_model, plotit = TRUE)
lambdaBC = boxCox1$x[which.max(boxCox1$y)]
lambdaBC
```

Verifying the Assumptions of Multiple Linear Regression

```
newY1 = data.frame(log(bplasma$betaplasma))
names(newY1) = "Y" # this is really just lnY
dataTrans = cbind(newY1,bplasma)
head(dataTrans)
dataTrans$betaplasma = NULL
model_after_transformation = lm(Y~.^2 , data = dataTrans)
#sw and fk test
ei= model_after_transformation$residuals
the.SWtest = shapiro.test(ei)
the.SWtest
Group = rep("Lower",nrow(dataTrans)) #Creates a vector that repeats "Lower" n times
Group[dataTrans$Y > median(dataTrans$Y)] = "Upper" #Changing the appropriate values to "Upper"
Group = as.factor(Group) #Changes it to a factor, which R recognizes as a grouping variable.
dataTrans$Group = Group
the.FKtest= fligner.test(model_after_transformation$residuals, dataTrans$Group)
the.FKtest
dataTrans$Group = NULL
par(mfrow=  c(2,2))
qqnorm(model_after_transformation$residuals, pch = 19,cex = 1.25)
qqline(model_after_transformation$residuals,lwd = 2, col = "purple")
hist(model_after_transformation$residuals, main = "Residuals", xlab = "ei",pch = 16,cex = 1.25,col="white")
plot(model_after_transformation$fitted.values, model_after_transformation$residuals,
main = "Errors vs. Fitted Values", xlab = "Fitted Values",ylab = "Errors")
abline(h = 0,col = "purple")
```

Outliers

```
library(leaps)
library(MPV)
ei.s=model_after_transformation$residuals/sqrt(sum(model_after_transformation$residuals^2)/(nrow(dataTrans) - length(model_after_transformation$coefficients)))
ri = rstandard(model_after_transformation)
ti = rstudent(model_after_transformation)
alpha = 0.1
n= nrow(dataTrans)
p = length(model_after_transformation$coefficients)
cutoff = qt(1-alpha/(2*n), n-p )
cutoff.deleted = qt(1-alpha/(2*n), n -p -1 )
outliers = which(abs(ei.s) > cutoff | abs(ri) > cutoff | abs(ti) > cutoff.deleted)
dataTrans[outliers,] #remove 39
dataTrans = dataTrans[!(row.names(dataTrans) %in% c('39')), ]
rownames(dataTrans) = seq(length=nrow(dataTrans))
model_after_transformation = lm(Y~.^2 , data = dataTrans)
```

Cooks Distance and its Plot

```
h = influence(model_after_transformation)$hat
sort(h[which(h>2*p/n)], decreasing = TRUE)
lev.hat = which(all.values[,"hat"] >2*p/n)
dataTrans[lev.hat,]
res = model_after_transformation$residuals
mse = anova(model_after_transformation)["Residuals", 3]
cook.d = res^2*h/(p*mse*(1-h)^2)
sort(cook.d[which(cook.d>1)], decreasing = T)
all.values =
influence.measures(model_after_transformation)$infmat
lev.DI = which(all.values[,"cook.d"] >1 )
dataTrans[lev.DI,]
par(mfrow = c(1,1))
plot(model_after_transformation, which=4)
```

Fitted Values Plot

```
which(rownames(dataTrans)=="61")
fit=lm(Y ~.^2, data=dataTrans,
subset=setdiff(rownames(dataTrans), "62"))
rbind(model_after_transformation$coefficients,fit$coefficients)
```

```
plot(model_after_transformation$fitted.value, predict(fit,
dataTrans), xlab="Fitted values all cases", ylab="Fitted values w/o
case 61") ## compare fitted values
abline(0,1)
dataTrans = dataTrans[!(row.names(dataTrans) %in% c('61')), ]
rownames(dataTrans) = seq(length=nrow(dataTrans))
```

Model Selection

Stepwise Selection and Model Validation

```
set.seed(100)
n = nrow(dataTrans) #312
ind = sample(1:n, n*0.55, replace=FALSE)
train = dataTrans[ind, ] #training set
valid = dataTrans[-ind, ] #validation/test set
full.model = lm(Y~.^2, data = train)
empty.model = lm(Y ~ 1, data = train)
FB.model.BIC = stepAIC(empty.model,  scope = list(lower =
empty.model, upper= full.model), k = log(n),trace=FALSE,direction
= "both")
FB.model.BIC
train1 = lm(formula = Y ~ bmi + vituse, data = train)
valid1 = lm(formula = Y ~ bmi + vituse, data = valid)
train_est = summary(train1)$coefficients[,1]
valid_estim = summary(valid1)$coefficients[,1]
train_se = summary(train1)$coefficients[,2]
valid_se = summary(valid1)$coefficients[,2]
mod_sum = cbind(train_est, valid_estim, train_se, valid_se )
colnames(mod_sum) = c("Train Est","Valid Est","Train s.e.","Valid
s.e.")
mod_sum
mod_sum[,1] - mod_sum[,2] #diff train est - test est
mod_sum[,3] - mod_sum[,4] #train se - test se
sse_t = sum(train1$residuals^2)
sse_v = sum(valid1$residuals^2)
Radj_t = summary(train1)$adj.r.squared
Radj_v = summary(valid1)$adj.r.squared
train_sum = c(sse_t,Radj_t)
valid_sum = c(sse_v,Radj_v)
criteria = rbind(train_sum,valid_sum)
colnames(criteria) = c("SSE","R2_adj")
criteria
newdata = valid[, -1]
y.hat = predict(train1, newdata)
MSPE = mean((valid$Y - y.hat)^2)
MSPE
MSPE - sse_t/n
```

Final Model

```
final_model = lm(formula = Y ~ bmi + vituse, data = dataTrans)
summary(final_model)
```

Prediction intervals

```
mult.fun = function(n,p,g,alpha){
  bon = qt(1-alpha/(2*g), n-p)
  WH = sqrt(p*qf(1-alpha,p,n-p))
  Sch = sqrt(g*qf(1-alpha,g,n-p))
  all.mul = c(bon,WH,Sch)
  all.mul = round(all.mul,3)
  names(all.mul) = c("Bon","WH","Sch")
  return(all.mul)}
mult.CI = function(C.star,x.stars,the.model,alpha,the.type =
"confidence"){
  all.preds = predict(the.model,x.stars)
  if(the.type == "confidence"){
    all.se = predict(the.model,x.stars,interval = the.type,se.fit =
TRUE)$se.fit
  } else if(the.type == "prediction"){
    all.se = predict(the.model,x.stars,interval = the.type,se.fit =
TRUE)$se.fit
    MSE =
sum(the.model$residuals^2)/(length(the.model$residuals) -
length(the.model$coefficients))
    all.se = sqrt(all.se^2 + MSE) }
  LB = all.preds - C.star*all.se
  UB = all.preds + C.star*all.se
  all.CIs = cbind(LB,UB)
  colnames(all.CIs) = paste((1-alpha)*100, "%",c(" Lower","
Upper"), sep = "")
  results = cbind(all.preds,all.CIs)
  colnames(results)[1] = "Estimate"
  return(results)}
all.of.them = mult.fun(nrow(dataTrans),
length(final_model$coefficients), 3, 0.05)
all.of.them
xs = data.frame(bmi=c(27,19,31),vituse = c("NO","OFTEN","NOT
OFTEN"))
all.the.CIs = mult.CI(all.of.them[1], xs,
final_model,0.05,"prediction")
all.the.CIs
cbind(xs,all.the.CIs
```