Exam 2 - Question 1

Adam McQuistan

Tuesday, April 05, 2016

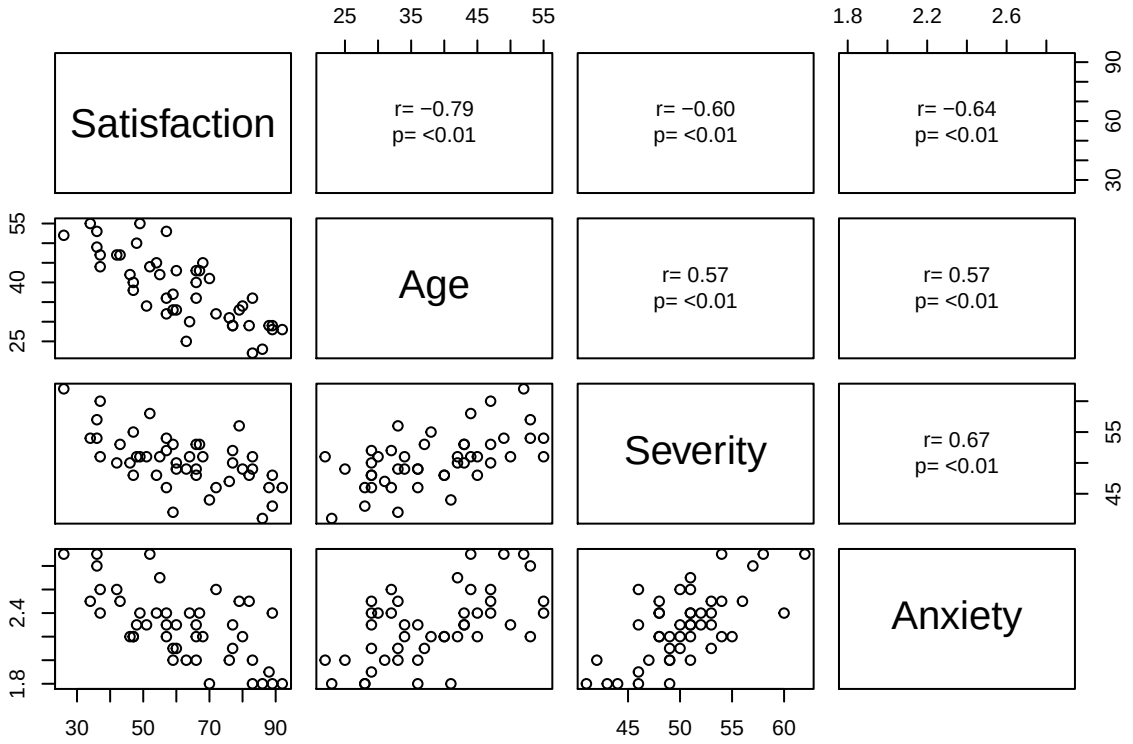
Problem 1 - Do problem 6.15 on page 250.

- Do not do part (a)
- Do parts (b-g)
- Extra - Conduct Brown-Forsythe test or Levene test. Group them based on median of predicted value of Y.

B. Scatter plot matrix and correlation matrix with interpretation.

```
panel.cor <- function(x, y, digits = 2, cex.cor, ...){
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  # correlation coefficient
  r <- cor(x, y)
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste("r= ", txt, sep = "")
  text(0.5, 0.6, txt)

  # p-value calculation
  p <- cor.test(x, y)$p.value
  txt2 <- format(c(p, 0.123456789), digits = digits)[1]
  txt2 <- paste("p= ", txt2, sep = "")
  if(p<0.01) txt2 <- paste("p= ", "<0.01", sep = "")
  text(0.5, 0.4, txt2)
}
df <- read.csv("data/6.15-6.16.csv")
names(df) = c("Satisfaction","Age", "Severity", "Anxiety")
pairs(df, upper.panel = panel.cor)
```



The matrix plot shows that all three predictor varaibles are all at least moderately correlated and linear to the outcome varaible of satisfaction.

Part C. Create a regression model for all three predictors and state the predicted regression function. How is β_2 interpreted?

```
result <- lm(Satisfaction ~ Age + Severity + Anxiety, data=df)
result_smry <- summary(result)
F_stat <- round(as.numeric(result_smry$fstatistic["value"]),1)
F_crit <- round(qf(0.95, df1=3, df2=result_smry$df[2]),1)
result_smry
```

```
##
## Call:
## lm(formula = Satisfaction ~ Age + Severity + Anxiety, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.3524  -6.4230   0.5196   8.3715  17.1601
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 158.4913    18.1259   8.744 5.26e-11 ***
## Age          -1.1416     0.2148  -5.315 3.81e-06 ***
```

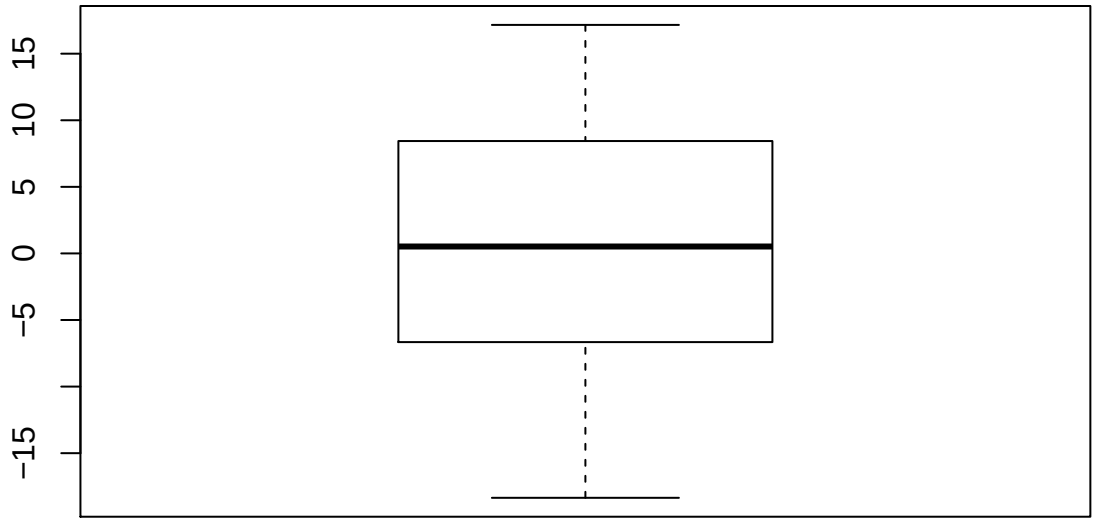
```
## Severity     -0.4420     0.4920  -0.898   0.3741
## Anxiety     -13.4702     7.0997  -1.897   0.0647 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.06 on 42 degrees of freedom
## Multiple R-squared:  0.6822, Adjusted R-squared:  0.6595
## F-statistic: 30.05 on 3 and 42 DF,  p-value: 1.542e-10
```

Regression Model: Satisfaction = 158.49 - 1.14 x Age - 0.44 x Severity - 13.47 x Anxiety

β_2 (Severity) has a coefficient of -0.44 which means that as severity increases 1 unit satisfaction drops 0.44 units.

D. Obtain the residuals and prepare a boxplot. Are there any outliers.

```
boxplot(result_smry$residuals)
```




```
hist(result_smry$residuals, breaks=12)
```

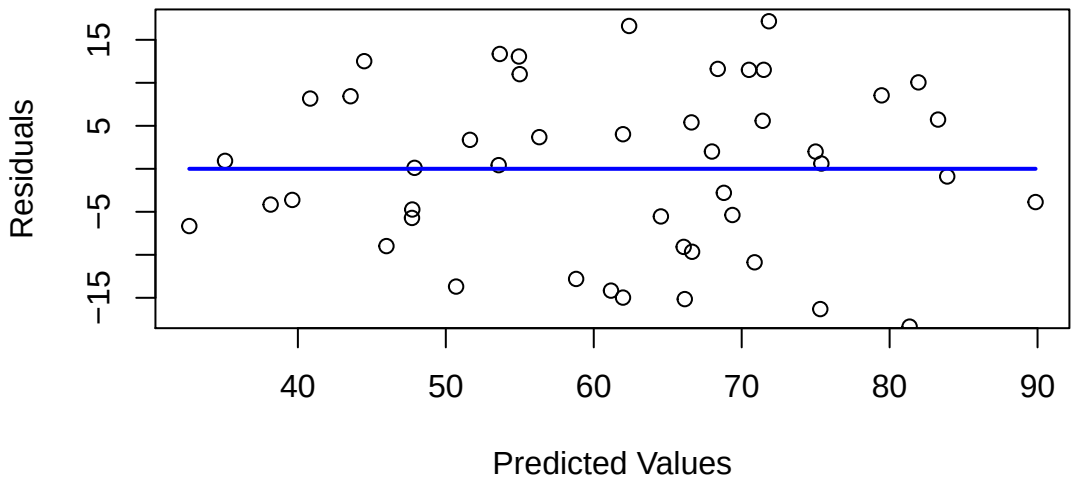
There does not appear to be outliers in the box plot but the histogram shows a non normal distribution of residuals.

E. Plot the residuals against predicted values and two factor interactions. Prepare a normal probability plot. Interpret.

```
df_model <- result$model[, 1:4]
df_model$Residuals <- result_smry$residuals
df_model$PredictedVals <- result$fitted.values
df_model$AgeSeverity<- df_model$Age * df_model$Severity
df_model$AgeAnxiety <- df_model$Age * df_model$Anxiety
df_model$SeverityAnxiety <- df_model$Severity * df_model$Anxiety
```

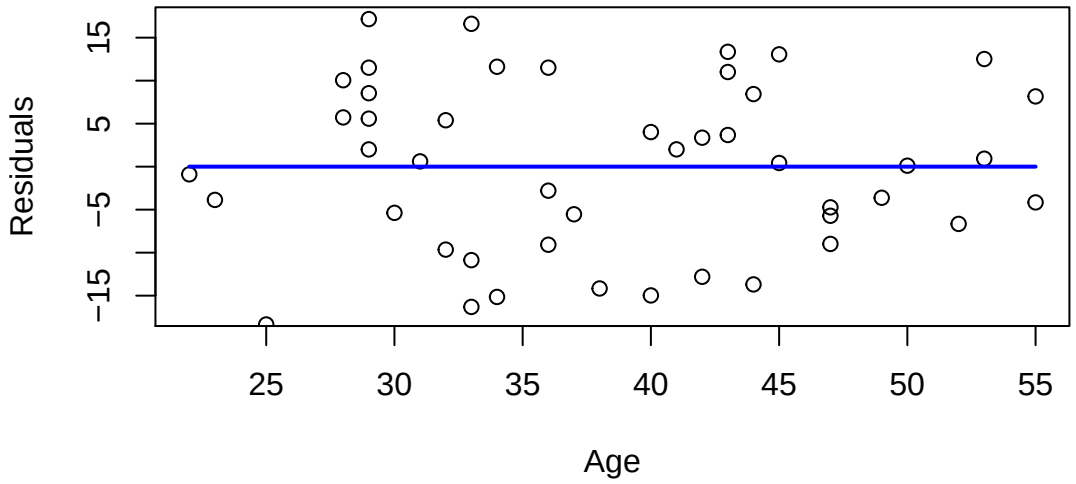
```
with(df_model, {
     plot(x=PredictedVals, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Predicted Values", ylab="Residuals", main="")

     points(c(min(PredictedVals), max(PredictedVals)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



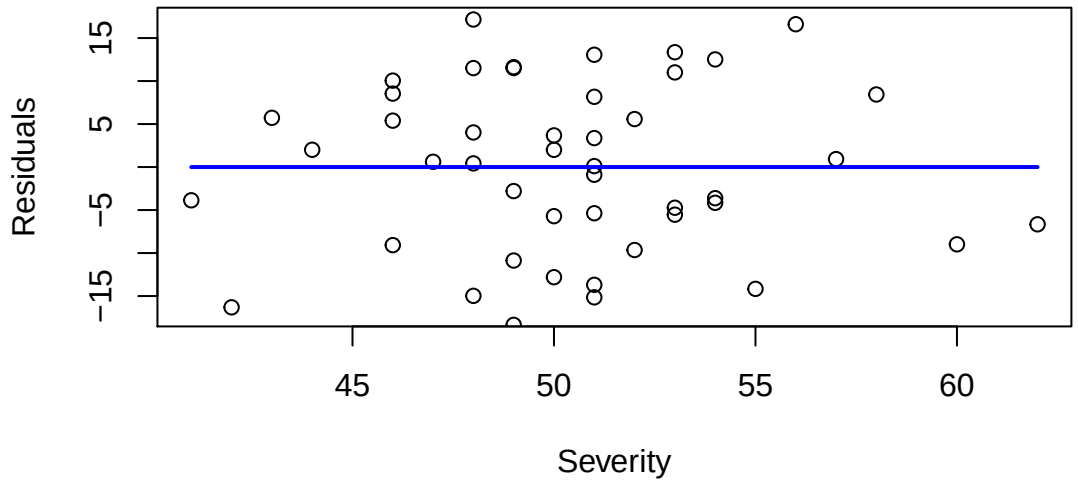

```
with(df_model, {
     plot(x=Age, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Age", ylab="Residuals", main="")

     points(c(min(Age), max(Age)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



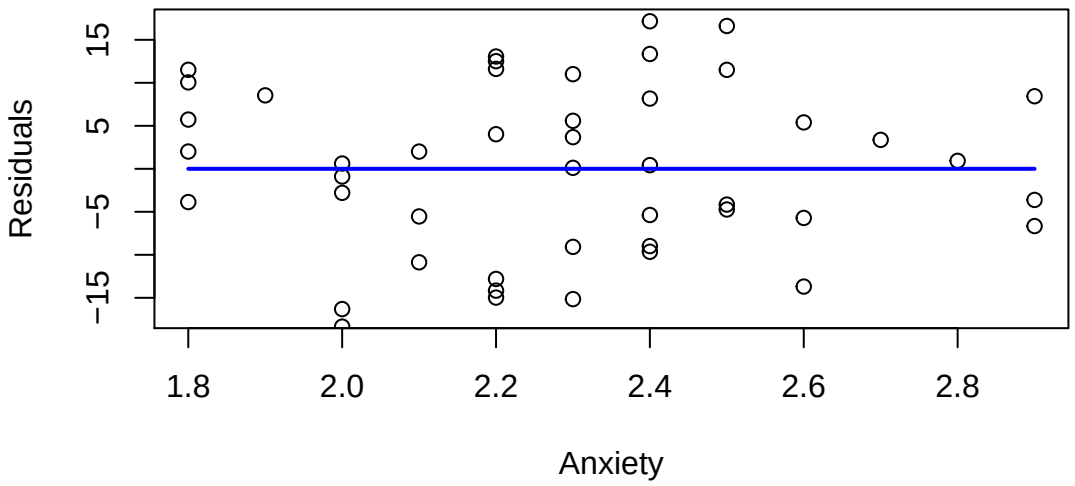
```r
with(df_model, {
     plot(x=Severity, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Severity", ylab="Residuals", main="")

     points(c(min(Severity), max(Severity)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



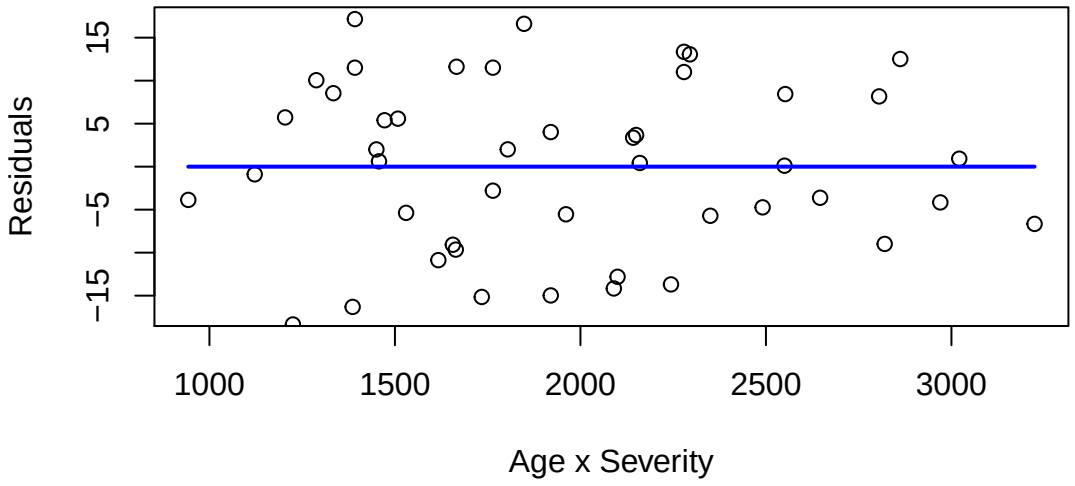

```r
with(df_model, {
     plot(x=Anxiety, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Anxiety", ylab="Residuals", main="")

     points(c(min(Anxiety), max(Anxiety)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



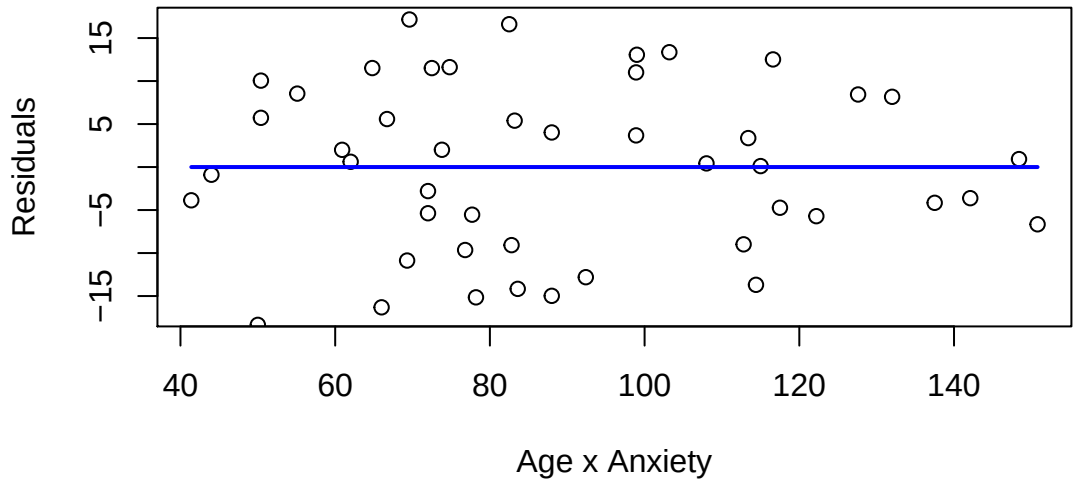

```
with(df_model, {
     plot(x=AgeSeverity, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Age x Severity", ylab="Residuals", main="")

     points(c(min(AgeSeverity), max(AgeSeverity)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```



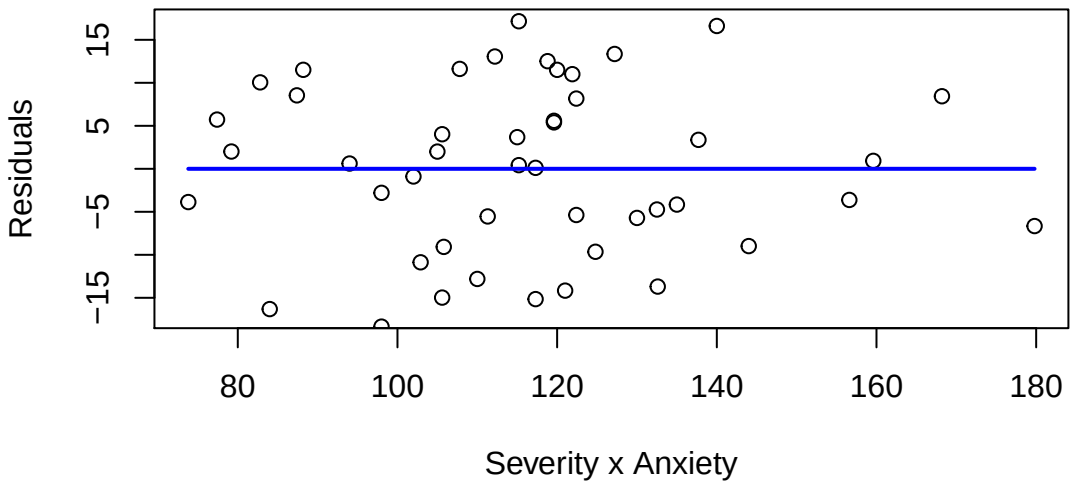
```r
with(df_model, {
     plot(x=AgeAnxiety, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Age x Anxiety", ylab="Residuals", main="")

     points(c(min(AgeAnxiety), max(AgeAnxiety)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```

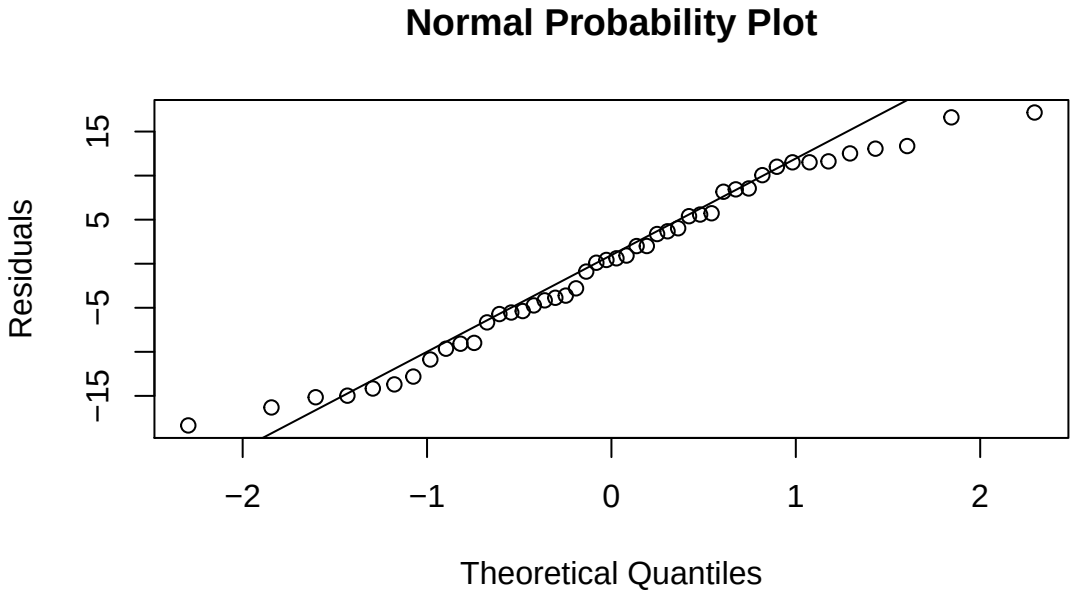



```r
with(df_model, {
     plot(x=SeverityAnxiety, y=Residuals,
          ylim=c(-max(Residuals), max(Residuals)),
          xlab="Severity x Anxiety", ylab="Residuals", main="")

     points(c(min(SeverityAnxiety), max(SeverityAnxiety)),
            c(0,0), type="l", lwd="2", col="blue")
     })
```




```
qqnorm(result$residuals, ylab="Residuals", main="Normal Probability Plot")
qqline(result$residuals)
```



The residual scatter plots do not show any major or systematic pattern or non-normal variance of the error terms against any of the predictor variables or predicted varaibles as well as the two-factor interactions of them. The normal probability plot however does suggest there is a deviation from normality with values greater than the 3rd quartile of the predictors.

Part F. Conduct a formal test for lack of fit.

First check for repeating groups of predictor combinations to assess whether artificial repeating groups are needed.

```
df$CommonPredictors <- with(df,paste(
  as.character(Age),
  as.character(Severity),
  as.character(Anxiety),
  sep="-"))



library(dplyr)
library(knitr)
df_tbl <- tbl_df(df)
df_tbl_levels <- group_by(df_tbl, CommonPredictors) %>% summarize(LevelRepeats=n()) %>% arrange(LevelRe
df_levels <- as.data.frame(df_tbl_levels)
kable(df_levels)
```

CommonPredictors	LevelRepeats
22-51-2	1
23-41-1.8	1
25-49-2	1
28-43-1.8	1
28-46-1.8	1
29-46-1.9	1
29-48-2.4	1
29-48-2.5	1
29-50-2.1	1
29-52-2.3	1
30-51-2.4	1
31-47-2	1
32-46-2.6	1
32-52-2.4	1
33-42-2	1
33-49-2.1	1
33-56-2.5	1
34-49-2.2	1
34-51-2.3	1
36-46-2.3	1
36-49-1.8	1
36-49-2	1
37-53-2.1	1

CommonPredictors	LevelRepeats
38-55-2.2	1
41-44-1.8	1
42-50-2.2	1
42-51-2.7	1
43-50-2.3	1
43-53-2.3	1
43-53-2.4	1
44-51-2.6	1
44-58-2.9	1
45-48-2.4	1
45-51-2.2	1
47-50-2.6	1
47-53-2.5	1
47-60-2.4	1
49-54-2.9	1
50-51-2.3	1
52-62-2.9	1
53-54-2.2	1
53-57-2.8	1
55-51-2.4	1
55-54-2.5	1
40-48-2.2	2

Since there is only one repeating group of the three predictors we'll need to create artificial repeating groups that are close to each other. Predictor values will be grouped by equal fifths which will increase the number of similar groups the three predictors will fall into.

```
smoother <- function(x){
  groups <- quantile(x, probs=seq(0,1,0.2))
  groups <- as.numeric(groups)
  for(i in 1:length(x)){
    lower_idx <- max(which(as.numeric(groups) <= x[i]))
    upper_idx <- lower_idx + 1
    mp <- ifelse(lower_idx == length(groups),
              (groups[lower_idx-1] + groups[lower_idx]) / 2,
              (groups[lower_idx] + groups[upper_idx]) / 2)
    x[i] <- mp
  }

  return(x)
}
```

```
df$AgeSmooth <- smoother(df$Age)
df$SeveritySmooth <- smoother(df$Severity)
df$AnxietySmooth <- smoother(df$Anxiety)

df$CommonPredictors <- with(df,paste(
  as.character(AgeSmooth),
  as.character(SeveritySmooth),
  as.character(AnxietySmooth),
  sep="-"))

df_tbl <- tbl_df(df)
df_tbl_levels <- group_by(df_tbl, CommonPredictors) %>% summarize(LevelRepeats=n()) %>% arrange(LevelRe
df_levels <- as.data.frame(df_tbl_levels)
kable(df_levels)
```

CommonPredictors	LevelRepeats
25.5-50-2.1	1
25.5-52-2.1	1
31.5-44.5-1.9	1
31.5-44.5-2.7	1
31.5-48.5-2.45	1
31.5-48.5-2.7	1
31.5-52-2.3	1
31.5-57.5-2.7	1
38-44.5-1.9	1
38-44.5-2.3	1
38-50-1.9	1
38-50-2.1	1
38-50-2.3	1
38-52-2.3	1
38-57.5-2.1	1
38-57.5-2.3	1
44.5-48.5-2.45	1
44.5-52-2.3	1
44.5-57.5-2.3	1
44.5-57.5-2.45	1
44.5-57.5-2.7	1
51-50-2.7	1
51-52-2.3	1
51-52-2.45	1
51-57.5-2.3	1

CommonPredictors	LevelRepeats
51-57.5-2.45	1
31.5-44.5-2.1	2
31.5-50-2.1	2
31.5-52-2.45	2
38-48.5-2.3	2
44.5-50-2.3	2
44.5-52-2.7	2
25.5-44.5-1.9	3
51-57.5-2.7	5

This method reduced the number of distinct levels from 46 to 34 which should be enough to allow for a decent lack of fit test.

```
reduced <- lm(Satisfaction ~ AgeSmooth + SeveritySmooth + AnxietySmooth, data=df)
full <- lm(Satisfaction ~ factor(AgeSmooth) + factor(SeveritySmooth) + factor(AnxietySmooth), data=df)

anova(reduced, full)
```

```
## Analysis of Variance Table
##
## Model 1: Satisfaction ~ AgeSmooth + SeveritySmooth + AnxietySmooth
## Model 2: Satisfaction ~ factor(AgeSmooth) + factor(SeveritySmooth) + factor(AnxietySmooth)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     42 4578.5
## 2     33 3409.0  9    1169.4 1.2578 0.2958
```

The anova of the reduced verse full model is used to assess the appropriateness (fittness) of the model:

$H_o : E\{Y\} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 x_3$ * Concludes that the regression function is linear where p-value $>$ 0.05

$H_a : E\{Y\} \neq \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 x_3$ * Concludes that there is a lack of linear fit where p-value $<=$ 0.05

Since p-value $>$ 0.05 we do not reject the null hypothesis, the model appears adequate

Part G. Conduct Breusch-Pagan test for constancy of error variance of the models.

```
library(lmtest)
bptest(result)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  result
## BP = 2.5583, df = 3, p-value = 0.4648
```

H_o: the error terms have constant variance

H_a: at least one parameter has errors with non-constanct variance

Since the p-value is > 0.05 we do not reject the H_o and conclude there is constant variance (no-heterskedacity).

Extra: Conduct Leven test with grouping by the median of predicted Y

```
library(lawstat)
median_yhat <- median(df_model$PredictedVals)
df_model$LeveneGrps <- ifelse(df_model$PredictedVals <= median_yhat, "A","B")
with(df_model, levene.test(Residuals, as.factor(LeveneGrps),location="mean"))
```

```
##
##  classical Levene's test based on the absolute deviations from the
##  mean ( none not applied because the location is not set to median
##  )
##
## data:  Residuals
## Test Statistic = 0.0213, p-value = 0.8847
```

The Leven test assumes the two population's variances are equal

H_o: no difference in population variances

H_o: there are differences in the populations variances

Since p-value > 0.05 we do not reject the null hypothsis and conclude equal variance of the error terms.