Diagnostics and Remedial Measures Homework Solutions

1. (a)

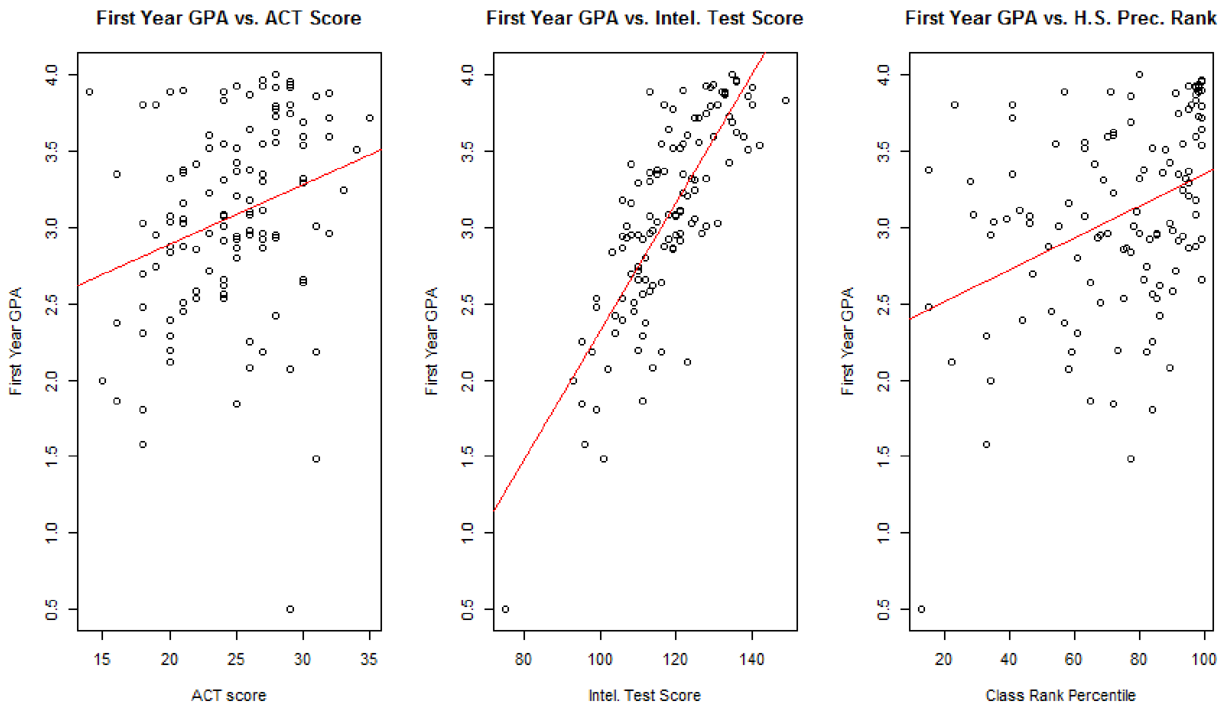
```
> data <- read.table("CH03PR03.txt")
> head(data)
    V1 V2  V3 V4
1 3.897 21 122 99
2 3.885 14 132 71
3 3.778 28 119 95
4 2.540 22  99 75
5 3.028 21 131 46
6 3.865 31 139 77
> names(data) <- c("GPA", "ACT", "IntelligenceScore", "ClassRankPerc")
> head(data)
   GPA ACT IntelligenceScore ClassRankPerc
1 3.897  21               122            99
2 3.885  14               132            71
3 3.778  28               119            95
4 2.540  22                99            75
5 3.028  21               131            46
6 3.865  31               139            77
> lmACT <- lm(data$GPA ~ data$ACT)
> lmIS <- lm(data$GPA ~ data$IntelligenceScore)
> lmCRP <- lm(data$GPA ~ data$ClassRankPerc)
> summary(lmACT)

Call:
lm(formula = data$GPA ~ data$ACT)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.74004 -0.33827  0.04062  0.44064  1.22737 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.11405    0.32089   6.588  1.3e-09 ***
data$ACT     0.03883    0.01277   3.040  0.00292 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6231 on 118 degrees of freedom
Multiple R-squared:  0.07262,    Adjusted R-squared:  0.06476 
F-statistic:  9.24 on 1 and 118 DF,  p-value: 0.002917

> summary(lmIS)

Call:
```

```
lm(formula = data$GPA ~ data$IntelligenceScore)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.1672 -0.2402 -0.0225  0.2977  1.0193 

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept)            -1.873921   0.345709  -5.421  3.2e-07 ***
data$IntelligenceScore  0.041944   0.002915  14.389  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3899 on 118 degrees of freedom
Multiple R-squared:  0.637,	Adjusted R-squared:  0.6339 
F-statistic:   207 on 1 and 118 DF,  p-value: < 2.2e-16

> summary(lmCRP)

Call:
lm(formula = data$GPA ~ data$ClassRankPerc)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.94233 -0.40879  0.05516  0.48679  1.25950 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)        2.306901   0.185497  12.436  < 2e-16 ***
data$ClassRankPerc 0.010417   0.002406   4.329 3.15e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6011 on 118 degrees of freedom
Multiple R-squared:  0.1371,	Adjusted R-squared:  0.1298 
F-statistic: 18.74 on 1 and 118 DF,  p-value: 3.153e-05
```

The r^2 value using the intelligence test score as predictor is highest at .637.

(b) Here are plots with best fit lines:

```
> par(mfrow=c(1, 3))
> plot(data[,1] ~ data[,2], main= "First Year GPA vs. ACT Score", xlab="ACT score", ylab="First Year GPA")
> abline(lmACT, col="red")
> plot(data[,1] ~ data[,3], main= "First Year GPA vs. Intel. Test Score", xlab="Intel. Test Score", 
ylab="First Year GPA")
> abline(lmIS, col="red")
> plot(data[,1] ~ data[,4], main= "First Year GPA vs. H.S. Prec. Rank", xlab="Class Rank Percentile",
ylab="First Year GPA")
> abline(lmCRP, col="red")
```



At this point is seems that the intelligence test score is the best predictor of the three. Also, it seems there is an outlier in the first plot at about (29, .5), which is the same student represented in the second graph above at about (75, .5). We should probably throw that point out. I will proceed without throwing that point out for now, and we can always come back to redo all our analyses after removing that point if need be.

(c)

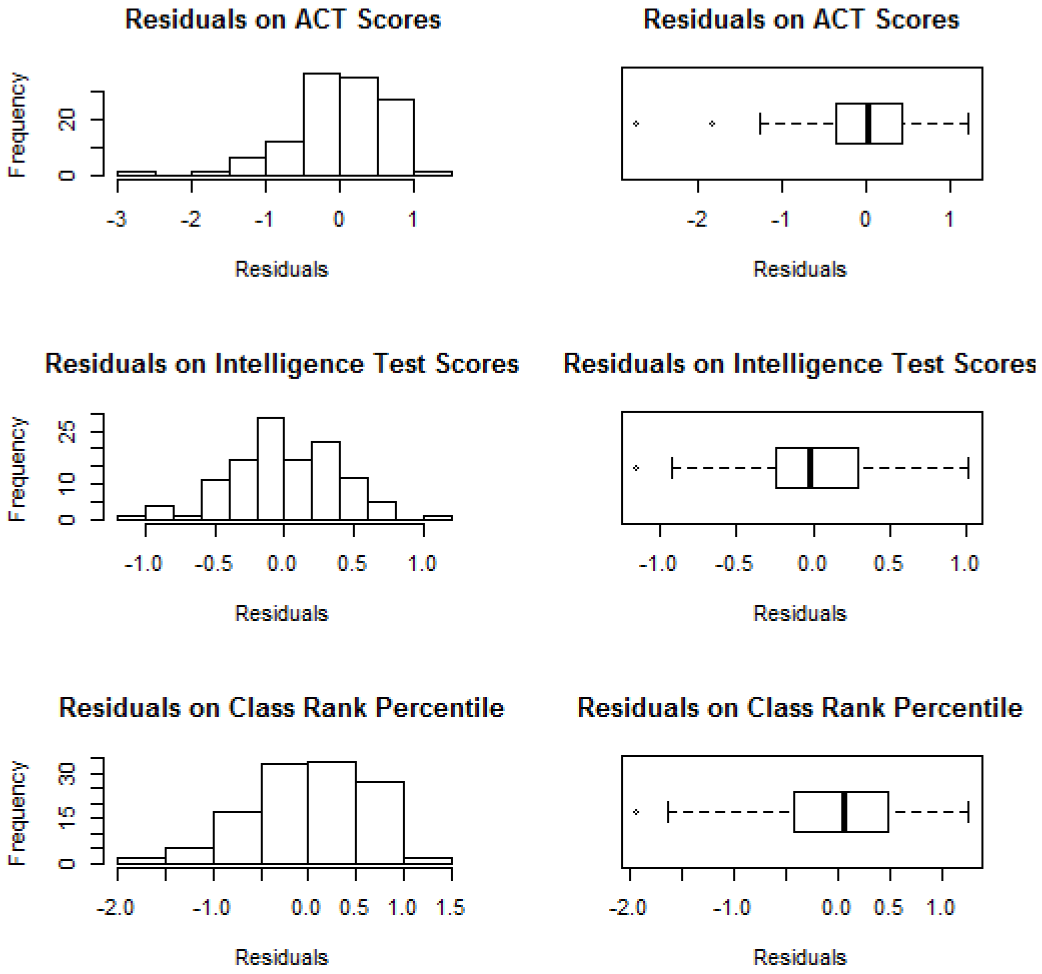
```
> shapiro.test(lmACT$residual)

        Shapiro-Wilk normality test

data:  lmACT$residual
W = 0.9525, p-value = 0.0003304

> shapiro.test(lmIS$residual)

        Shapiro-Wilk normality test

data:  lmIS$residual
```

```
W = 0.9913, p-value = 0.6566

> shapiro.test(lmCRP$residual)

        Shapiro-Wilk normality test

data:  lmCRP$residual
W = 0.9743, p-value = 0.02137
```

The Shapiro-Wilk p-value for residual normality test using intelligence test as predictor is the only one of the three that really does not indicate a departure from normality. Thus that model seems to be the only one that is not inconsistent with the model assumption of normally distributed errors.

(d) Code:

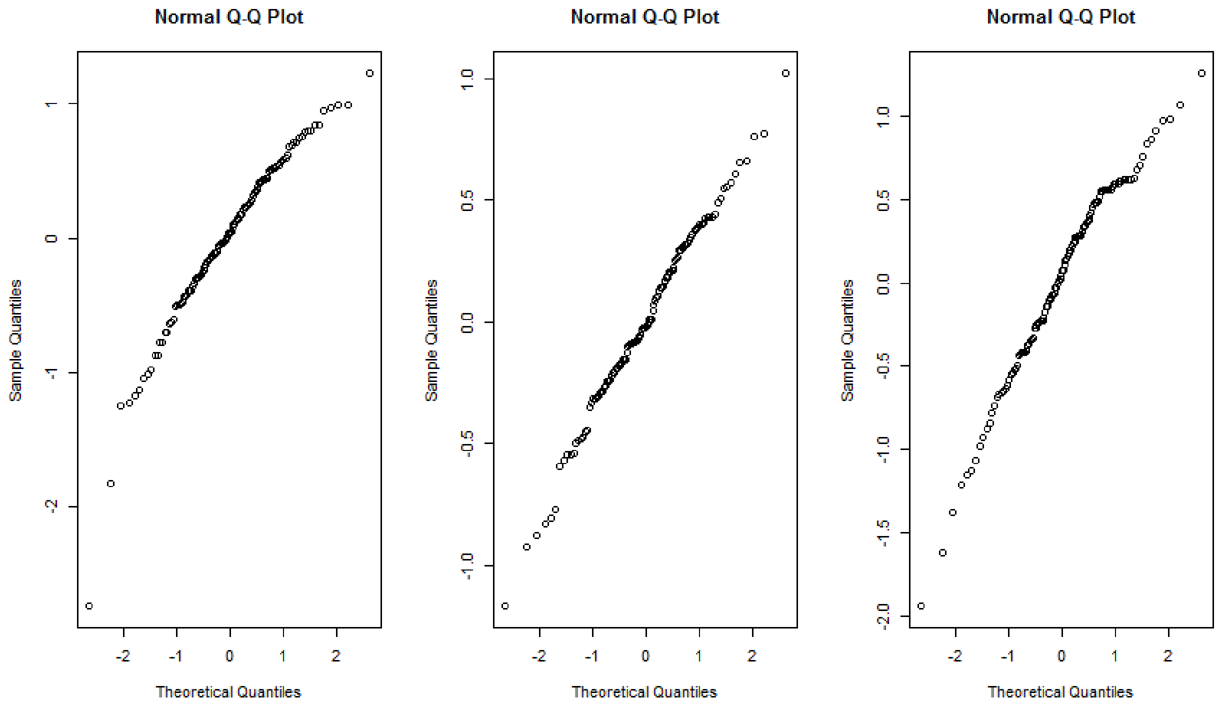
```
> windows()
> par(mfrow=c(3,2))
> hist(lmACT$residual, main="Residuals on ACT Scores", xlab="Residuals")
> boxplot(lmACT$residual, horizontal=TRUE, main="Residuals on ACT Scores", xlab="Residuals")
> hist(lmIS$residual, main="Residuals on Intelligence Test Scores", xlab="Residuals")
> boxplot(lmIS$residual, horizontal=TRUE, main="Residuals on Intelligence Test Scores",
xlab="Residuals")
> hist(lmCRP$residual, main="Residuals on Class Rank Percentile", xlab="Residuals")
> boxplot(lmCRP$residual, horizontal=TRUE, main="Residuals on Class Rank Percentile",
xlab="Residuals")
```



The residuals on ACT scores as well as those on class rank percentile look a little left-skewed, which is consistent with the normal probability plots in part (e) below.

(e) Code:

```
> windows()
> par(mfrow=c(1,3))
> qqnorm(lmACT$residual, main="Normal QQ Plot for Residuals on ACT")
> qqnorm(lmIS$residual, main="Normal QQ Plot for Residuals on Intel. Test")
> qqnorm(lmCRP$residual, main="Normal QQ Plot for Residuals on Class Rank")
```



(f) To run Levene's test, you'll need two or more categories. So we can split the observations by ACT score like this:

```
> ACTcode <- ifelse(data$ACT<26, "l", "h")
```

This will code the observations as "l" for low and "h" for high according to whether or not the ACT score was below 26. Now we can run Levene's test. Note this test requires the car package.

```
> library(car)
> leveneTest(lmACT$residual ~ ACTcode)
Levene's Test for Homogeneity of Variance (center = median)
       Df F value Pr(>F)
group   1  0.8042 0.3717
      118
Warning message:
In leveneTest.default(y = y, group = group, ...) : group coerced to factor.
```

(g) Now I'll run Levene's test on the residuals on the intelligence predictor, splitting according to (-infty, 120), [120, infty):

```
> IScode <- ifelse(data[,3]<120, "l", "h")
> head(IScode)
[1] "h" "h" "l" "l" "h" "h"
> head(data[,3])
[1] 122 132 119  99 131 139
> leveneTest(lmIS$residual ~ IScode)
```

```
Levene's Test for Homogeneity of Variance (center = median)
       Df F value Pr(>F)
group   1  4.4177 0.0377 *
      118
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Warning message:
In leveneTest.default(y = y, group = group, ...) : group coerced to factor.
```

According to Levene’s test, there may be a difference in variance between the residuals for the so-called low scores and the so-called high scores. This difference might be alleviate, btw, upon removal of that outlier:

```
> datano <- data[-9,]
> lmISno <- lm(datano[,1] ~ datano[,3])
> ISnocode <- ifelse(datano[,3]<120, "l", "h")
> leveneTest(lmISno$residual ~ ISnocode)
Levene's Test for Homogeneity of Variance (center = median)
       Df F value  Pr(>F)
group   1  4.1622 0.04359 *
      117
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Warning message:
In leveneTest.default(y = y, group = group, ...) : group coerced to factor.
```

Well, this didn’t improve the p-value as much as I’d hoped. Actually, since we are probably safe in assuming the residuals in this case are normally distributed, we can use a better test for equality of variance: Bartlett’s test is better if you can “satisfy” normality (otherwise use Levene’s test):

```
> bartlett.test(lmISno$residual ~ ISnocode)

        Bartlett test of homogeneity of variances

data:  lmISno$residual by ISnocode
Bartlett's K-squared = 3.6377, df = 1, p-value = 0.05649
```

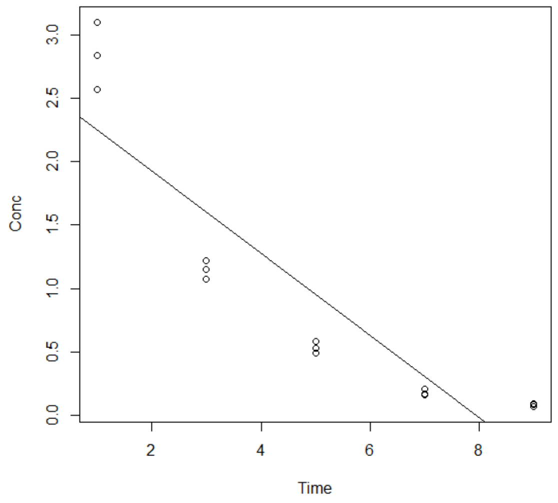
So it seems the residuals on the intelligence test score predictor might not have equal variance across the two groups (low and high). Perhaps some variance-stabilizing data transformation would be needed here…

(h) After all of this, it really seems the intelligence test score is the “best” predictor of first-year college GPA. The residual analysis seems to show the assumptions are more-or-less met, although it would be better to get the variance of the errors under control. Another method (besides applying appropriate data transformations) to controlling the error variance would be to apply weighted-least squares, which we will take up in a later section of notes. The r^2 value is good when using the intelligence test score, and so the sum of squared errors (SSE) is small (which is what you want). The graph looks good, too, and r^2 would probably improve if the outlier was removed.

2.

(a)

```
> solconc <- read.table("CH03PR15.txt")
> head(solconc)
    V1 V2
1 0.07  9
2 0.09  9
3 0.08  9
4 0.16  7
5 0.17  7
6 0.21  7
> names(solconc) <- c("Conc", "Time")
> head(solconc)
  Conc Time
1 0.07    9
2 0.09    9
3 0.08    9
4 0.16    7
5 0.17    7
6 0.21    7
> attach(solconc)
> out <- lm(Conc ~ Time)
> summary(out)

Call:
lm(formula = Conc ~ Time)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.5333 -0.4043 -0.1373  0.4157  0.8487 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.5753     0.2487  10.354 1.20e-07 ***
Time         -0.3240     0.0433  -7.483 4.61e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4743 on 13 degrees of freedom
Multiple R-squared:  0.8116,    Adjusted R-squared:  0.7971 
F-statistic: 55.99 on 1 and 13 DF,  p-value: 4.611e-06

> plot(Conc ~ Time)
> abline(out)
```



So it seems the linear model isn't too bad. The coefficient of determination is 81.16%. The p-values for the regression coefficients are quite small, indicating sound statistical evidence that any linear coefficients (beta0 and beta1) would truly differ from 0.

(b) The result of the ANOVA F-test is at the end of the output in part (a) and is 4.611 parts in a million. Note, of course, that this p-value is exactly the same as the p-value for the t-test for the slope coefficient beta1. This is because in the t-test we're testing

H0: beta1 = 0 vs. Ha: beta1 does not equal 0,

and with the ANOVA F-test we're testing to see if an association exists or not between the two variables… This is an F test, and it's equivalent to the t-test because an F random variable with 1 numerator degree of freedom and d denominator degrees of freedom has the same distribution as the square of a t- random variable with d degrees of freedom!! Definitely reject the hypothesis that beta0 = 0 in favor of the claim that it differs from 0. This conclusion also seems well supported by the plot.

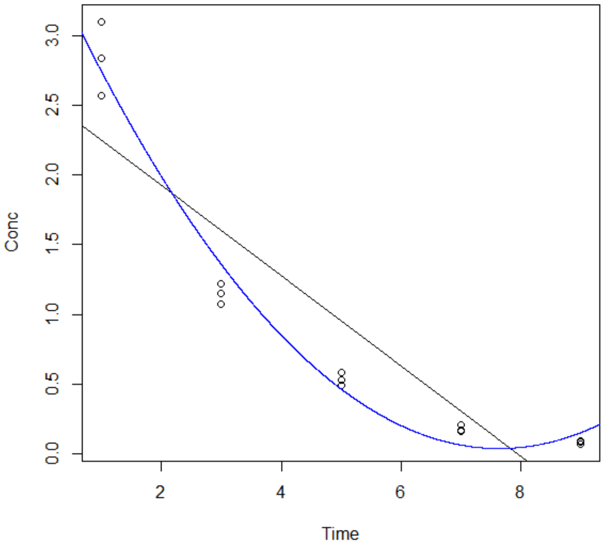
(c) I'm not sure I understand the question… the wording seems a little strange… If I understand it correctly, my answer is "no: the test does not indicate what kind of function to use (linear vs. quadratic, etc.). Even if we did a lack-of-linear-fit test which indicated a lack of linear fit, we would still not necessarily know what kind of model to use. However, if we look at the plot, it certainly seems reasonable that a polynomial model would be a good choice. Let's in fact run a lack of fit test:

```
> library(alr3)
Loading required package: car
Warning messages:
1: package 'alr3' was built under R version 3.0.3
2: package 'car' was built under R version 3.0.3
> pureErrorAnova(out)
Analysis of Variance Table
```

```
Response: Conc
            Df  Sum Sq Mean Sq F value    Pr(>F)
Time         1 12.5971 12.5971 800.325 7.080e-11 ***
Residuals    13  2.9247  0.2250
 Lack of fit  3  2.7673  0.9224  58.603 1.194e-06 ***
 Pure Error  10  0.1574  0.0157
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

So the lack of linear fit test indeed indicates a lack of linear fit!!! The p-value is about 1.19 parts in a million!!! Let’s try a polynomial model (not that we’ve learned this yet, but let’s do it and check out the multiple r^2 value and plot:

```
> polyout <- lm(Conc ~ Time + I(Time^2))
> summary(polyout)

Call:
lm(formula = Conc ~ Time + I(Time^2))

Residuals:
    Min      1Q  Median      3Q     Max
-0.2876 -0.1092  0.0261  0.1034  0.3572

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.619619   0.151158  23.946 1.69e-11 ***
Time        -0.938286   0.071362 -13.148 1.74e-08 ***
I(Time^2)    0.061429   0.006944   8.846 1.32e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.18 on 12 degrees of freedom
Multiple R-squared:  0.9749,    Adjusted R-squared:  0.9708
F-statistic: 233.5 on 2 and 12 DF,  p-value: 2.473e-10

> xcurve <- seq(0, 10, .001)
> ycurve <- 3.619619 -0.938286*xcurve + 0.061429*xcurve^2
> lines(xcurve, ycurve, col='blue', lty=1)
```



The polynomial fit looks pretty good. The p-values for the coefficients are very low, and the r^2 value is great: 0.9749.

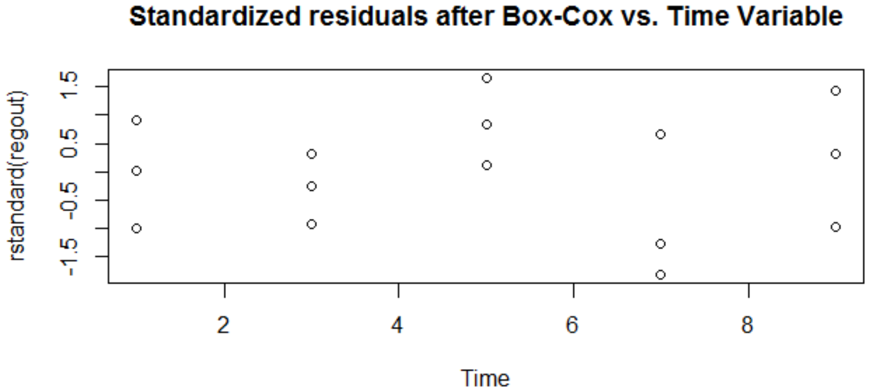
#3. Problem 3.16 from the book…

(a) We’ve already done this in the previous problem.

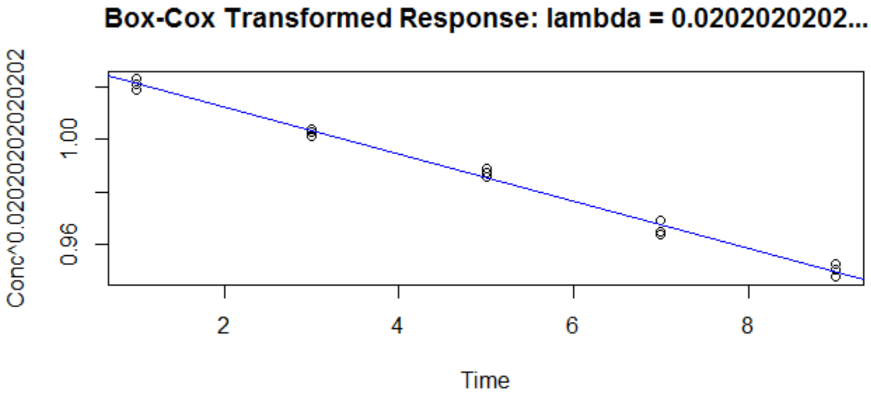
(b) Box-Cox! Here’s my code:

```
> xcurve <- seq(0, 10, .001)
> ycurve <- 3.619619 -0.938286*xcurve + 0.061429*xcurve^2
> lines(xcurve, ycurve, col='blue', lty=1)
> library(MASS)

Attaching package: ‘MASS’

The following object is masked from ‘package:alr3’:

    forbes

> trans <- boxcox(Conc ~ Time)
> trans
> lambda <- trans$x
> loglh <- trans$y
> boxcox <- cbind(lambda, loglh)
> boxcox[order(-loglh), ]
           lambda      loglh
  [1,]  0.02020202  13.4557358
  [2,]  0.06060606  13.2665123
  [3,] -0.02020202  12.7302103
  [4,]  0.10101010  12.2741506
  [5,] -0.06060606  11.2245048
  [6,]  0.14141414  10.6350115
  [7,] -0.10101010   9.3193673
  [8,]  0.18181818   8.6318079
  [9,] -0.14141414   7.3452472
 [10,]  0.22222222   6.5680968
.
.
.
> regout <- lm(Conc^0.0202020202 ~ Time)
> summary(regout)

Call:
lm(formula = Conc^0.0202020202 ~ Time)

Residuals:
       Min         1Q     Median        3Q       Max
-0.0038939 -0.0019707  0.0002368  0.0016077  0.0036004

Coefficients:
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.0302214  0.0011764  875.73  < 2e-16 ***
Time        -0.0089532  0.0002048  -43.72  1.7e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.002243 on 13 degrees of freedom
Multiple R-squared:  0.9932,    Adjusted R-squared:  0.9927
F-statistic:  1911 on 1 and 13 DF,  p-value: 1.701e-15
> windows()
> plot(regout)
Waiting to confirm page change...
Waiting to confirm page change...
Waiting to confirm page change...
Waiting to confirm page change...
> windows()
> par(mfrow = c(2,1))
> hist(regout$resid)
> qqnorm(regout2$resid)
Error in qqnorm(regout2$resid) : object 'regout2' not found
> qqnorm(regout$resid)
> qqline(regout$resid)
> skewness(regout$resid)
Error: could not find function "skewness"
> library(moments)
Warning message:
package ‘moments’ was built under R version 3.0.3
> skewness(regout$resid)
[1] -0.1218911
> windows()
> par(mfrow=c(2,1))
> plot(Time, rstandard(regout), main="Standardized residuals after Box-Cox vs. Time Variable")
> plot(Conc^0.02020202020202 ~ Time, main = "Box-Cox Transformed Response: lambda =
0.0202020202...")
> abline(regout, col="blue")
```

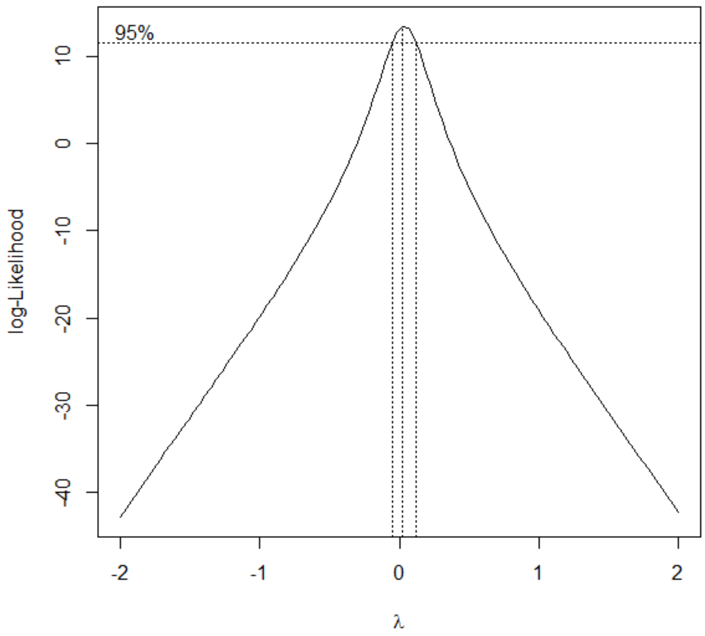


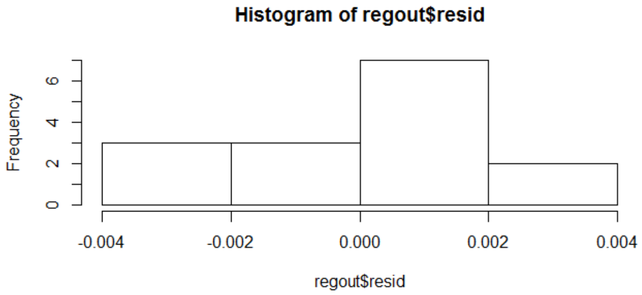
Standardized residuals after Box-Cox vs. Time Variable

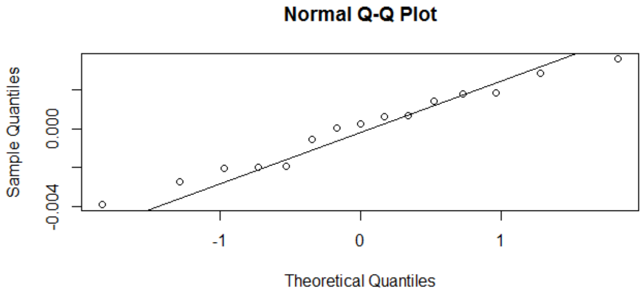


Box-Cox Transformed Response: lambda = 0.0202020202...

Great transformation!!!!!!!!!!!







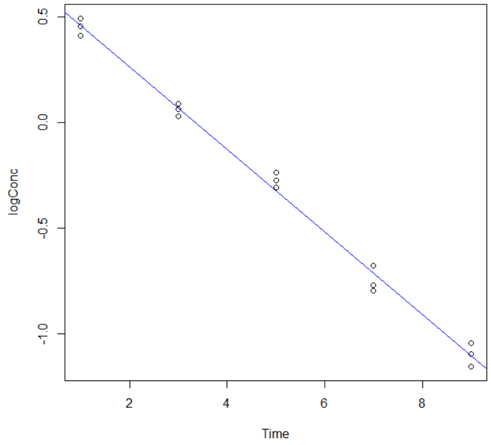
(c) Using the log_10 transformation:

```
> logConc <- log10(Conc)
> logout <- lm(logConc ~ Time)
> summary(logout)
```

```
Call:
lm(formula = logConc ~ Time)
```

```
Residuals:
      Min        1Q    Median        3Q       Max
-0.082958 -0.044421  0.006813  0.033512  0.085550

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.654880   0.026181   25.01 2.22e-12 ***
Time        -0.195400   0.004557  -42.88 2.19e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04992 on 13 degrees of freedom
Multiple R-squared:  0.993,     Adjusted R-squared:  0.9924
F-statistic:  1838 on 1 and 13 DF,  p-value: 2.188e-15
```

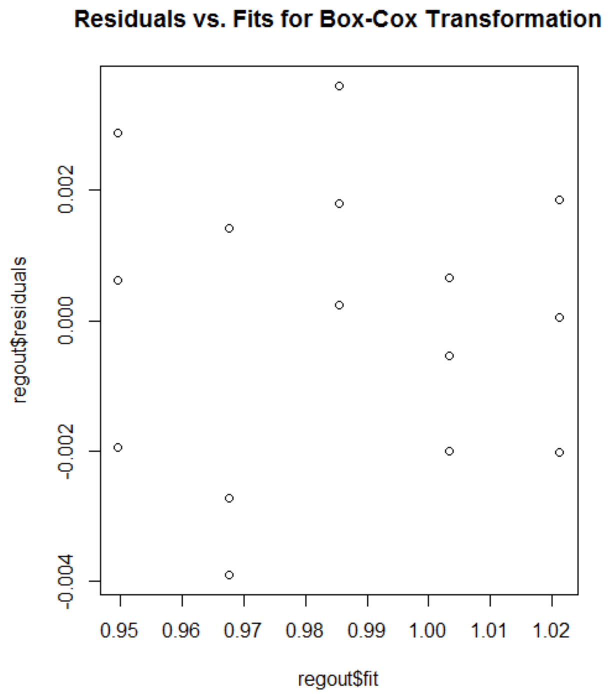
```
> windows()
> plot(logConc ~ Time)
> abline(logout, col="blue")
```

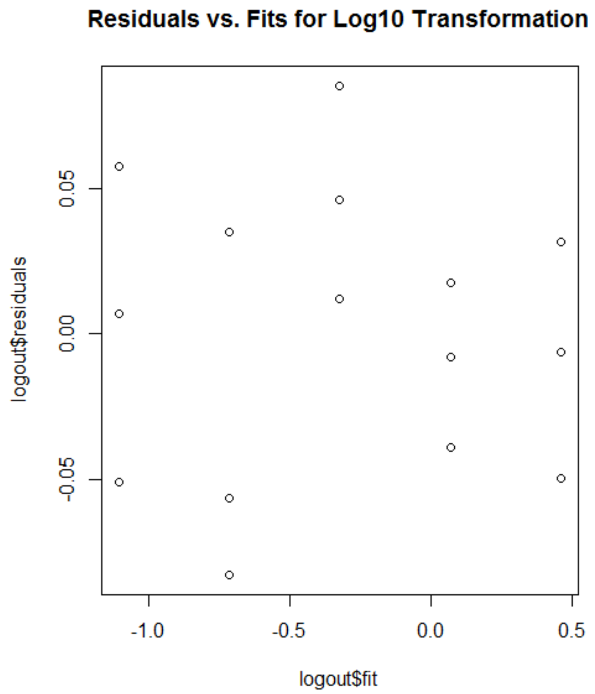


Very nice! So I think both transformations- the Box-Cox and the log10- are pretty good. The r^2 values were about the same for each.

(e) Let's plot the residuals vs. the fits and make a normal probability plot for both the Box-Cox and the log10 transformation scenarios:

```
> par(mfrow=c(1, 2))
> plot(regout$residuals ~ regout$fit, main="Residuals vs. Fits for Box-Cox Transformation")
> plot(logout$residuals ~ logout$fit, main="Residuals vs. Fits for Log10 Transformation")
```





These plots are VERY similar. The scales are different because the transformations are different. However, the point patters look the same!

(f) This questions is pretty tricky because THEY NEVER TOLD US THE ORIGINAL UNITS!!! All we know is we're to predict concentration with time. So let's make up our own units! Let's say the concentration unit is molarity, and the time unit is hours (who really knows?). For the log10 transformation, the regression line is

Estimate(Log10(Concentration)) = 0.654880 -0.195400 * Time

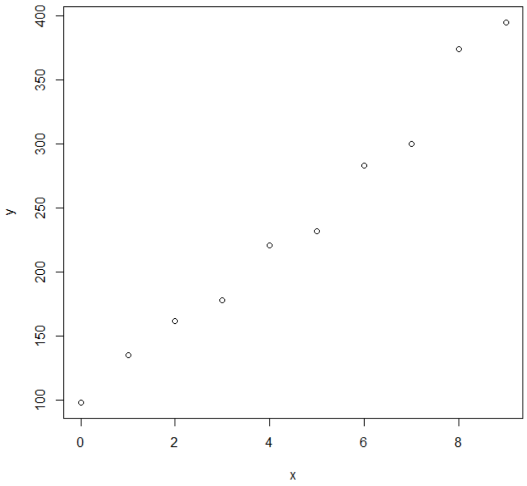
which means

Estimate (Concentration in molarity) = 10 ^ (0.654880 - 0.195400 * Time in hours)

4. ALSM 3.17. The data set is given in the problem.

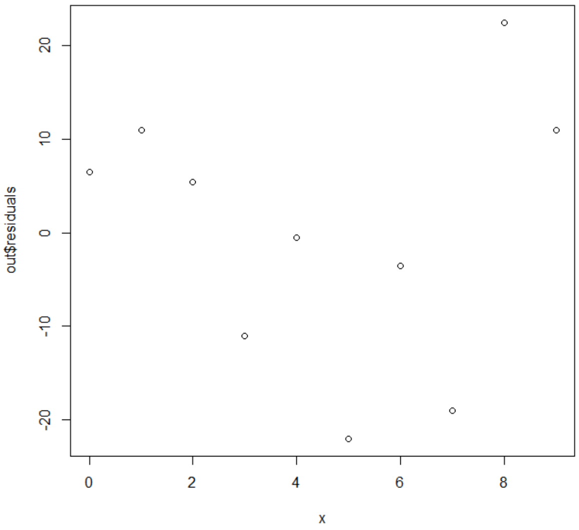
(a)

```
> x <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
> y <- c(98, 135, 162, 178, 221, 232, 283, 300, 374, 395)
> plot(y ~ x)
```



A linear model (like y = mb + b) certainly seems reasonable in this case. However, let's plot the residuals vs. the explanatory variable (which remember is coded time… years I believe):

```
> out <- lm(y~x)
> windows()
> plot(out$residuals ~ x)
```

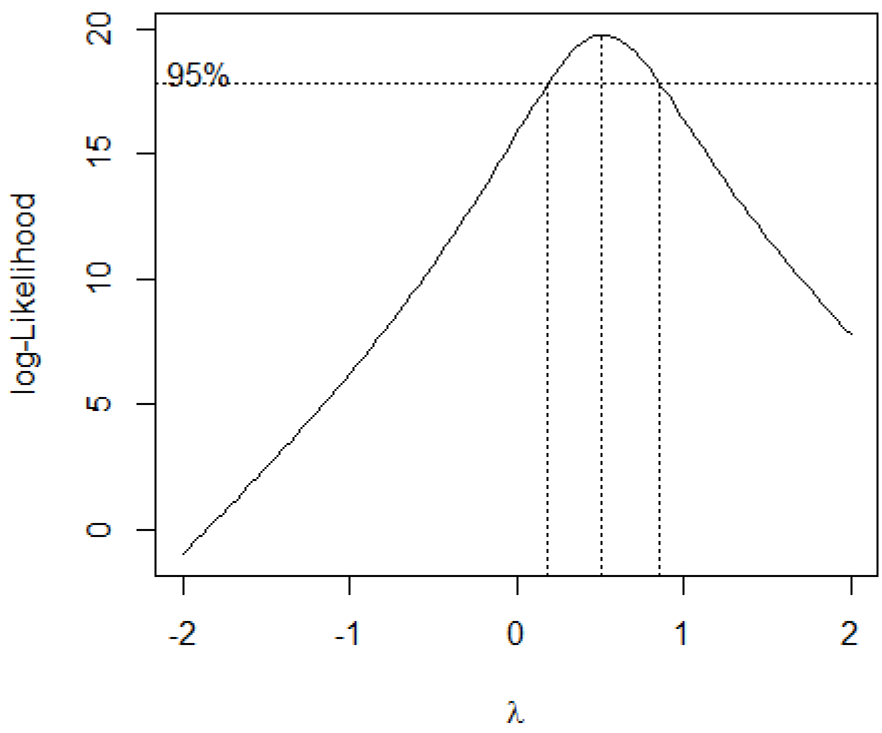


There could be something going on here between the residuals and the explanatory variable. Hmm…

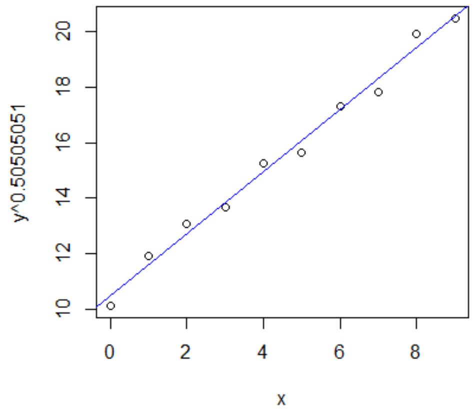
b) Box-Cox!! From the plot below, it looks like the optimal lambda value is about 0.05.

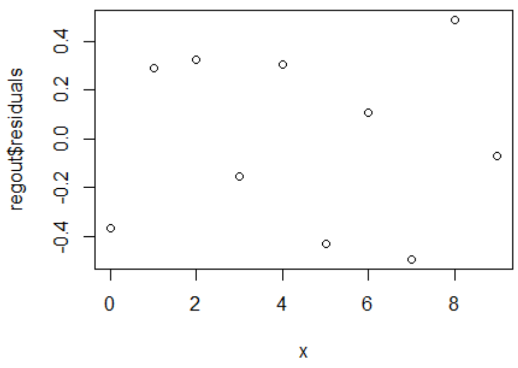
```
> library(MASS)
> trans <- boxcox(y ~ x)
> lambda <- trans$x
> loglh <- trans$y
> boxcox <- cbind(lambda, loglh)
> boxcox[order(-loglh),]
          lambda      loglh
 [1,] 0.50505051 19.7574566
 [2,] 0.54545455 19.7414832
 [3,] 0.46464646 19.7042926
```

Etc.
So it looks like the optimal lambda is about 0.50505051.



```
> regout <- lm(y^0.50505051 ~ x)
> summary(regout)

Call:
lm(formula = y^0.50505051 ~ x)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.49396 -0.31557  0.01724  0.30425  0.48855 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 10.50006    0.22092   47.53 4.25e-11 ***
x            1.11723    0.04138   27.00 3.81e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3759 on 8 degrees of freedom
Multiple R-squared:  0.9891,	Adjusted R-squared:  0.9878 
F-statistic: 728.9 on 1 and 8 DF,  p-value: 3.815e-09

> plot(y^0.50505051 ~ x)
> abline(regout, col="blue")
> windows()
> plot(regout$residuals ~ x)
```

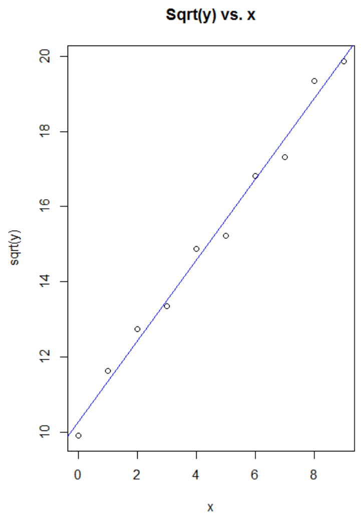


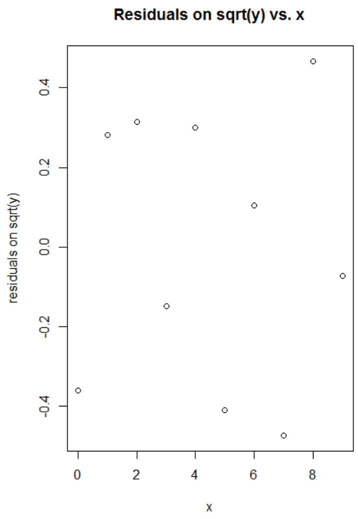


The residuals now look MUCH better… there doesn’t appear to be so much of a pattern.

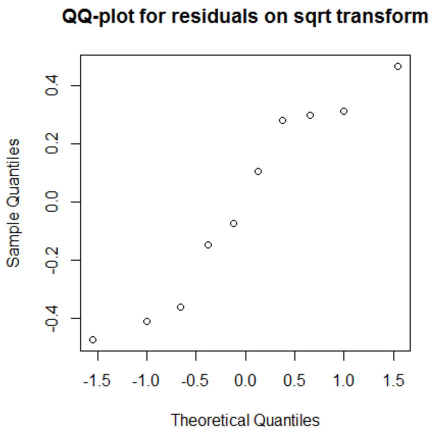
(c)– (f)

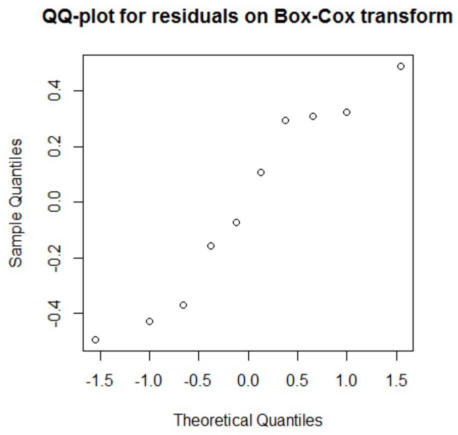
```
> sqrty <- sqrt(y)
> sqout <- lm(sqrty ~ x)
> windows()
> par(mfrow=c(1,2))
> plot(sqrty ~ x, xlab="x", ylab="sqrt(y)", main="Sqrt(y) vs. x")
> abline(sqout, col="blue")
> plot(sqout$residuals ~ x, xlab = "x", ylab="residuals on sqrt(y)", main="Residuals on sqrt(y) vs. x")
>
```





```
> par(mfrow=c(1,2))
> qqnorm(sqout$residuals, main = "QQ-plot for residuals on sqrt transform")
> qqnorm(regout$residuals, main = "QQ-plot for residuals on Box-Cox transform")
```





It seems both transforms have about the same effect on adjusting for the normality of the residuals. Actually, in terms of checking for normality, the original residuals, the residuals arising from the sqrt(y) transformation, and the residuals arising from the Box-Cox transformation all seem to not be inconsistent with the normality hypothesis:

```
> shapiro.test(out$resid)

        Shapiro-Wilk normality test

data:  out$resid
W = 0.9612, p-value = 0.7998

> shapiro.test(regout$resid)

        Shapiro-Wilk normality test

data:  regout$resid
W = 0.9191, p-value = 0.3496

> shapiro.test(sqout$resid)

        Shapiro-Wilk normality test

data:  sqout$resid
W = 0.9165, p-value = 0.3288
```

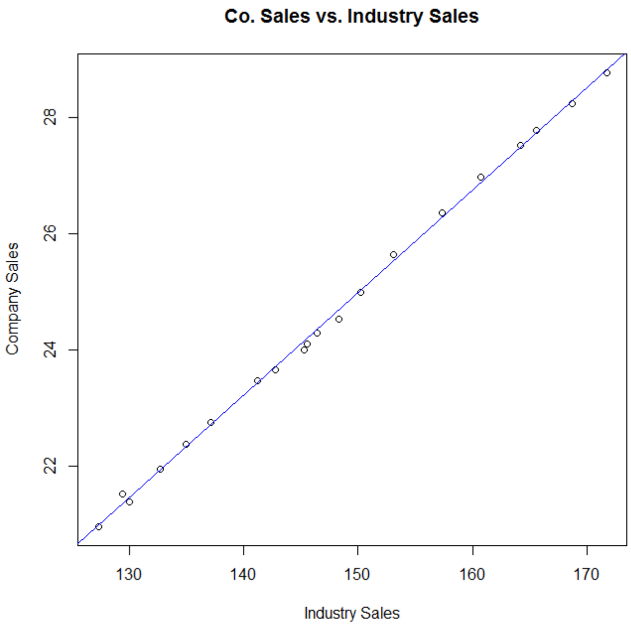
5.

(a)

```
> data <- read.table("CH12TA02.txt")
> head(data)
     V1    V2
1 20.96 127.3
2 21.40 130.0
3 21.96 132.7
4 21.52 129.4
5 22.39 135.0
6 22.76 137.1
> time <- seq(1, nrow(data), 1)
> time
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
> data <- cbind(data, time)
> names(data) <- c("Sales", "IndSales", "Quarter")
> head(data)
  Sales IndSales Quarter
1 20.96    127.3       1
2 21.40    130.0       2
3 21.96    132.7       3
4 21.52    129.4       4
5 22.39    135.0       5
6 22.76    137.1       6
> attach(data)
> plot(Sales ~ IndSales, main="Co. Sales vs. Industry Sales", xlab="Industry Sales", ylab="Company
Sales")
> out <- lm(Sales ~ IndSales)
> abline(out, col="blue")
> summary(out)

Call:
lm(formula = Sales ~ IndSales)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.149142 -0.054399 -0.000454  0.046425  0.163754 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.454750   0.214146  -6.793 2.31e-06 ***
IndSales     0.176283   0.001445 122.017  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 0.08606 on 18 degrees of freedom
Multiple R-squared:  0.9988,    Adjusted R-squared:  0.9987
F-statistic: 1.489e+04 on 1 and 18 DF,  p-value: < 2.2e-16
```

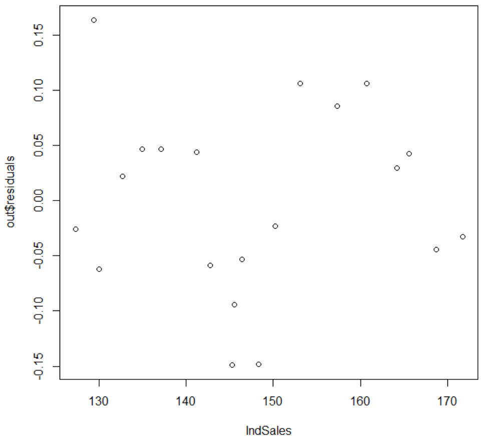
There seems to be a strong, positive linear association between the two variables.



However, if you look closely, you will observe a slight ripple across the regression line. We could investigate that by subtracting the line values (the fits) from the observe y-values. Of course, that's just the residuals! Let's graph the residuals versus the industry sales:

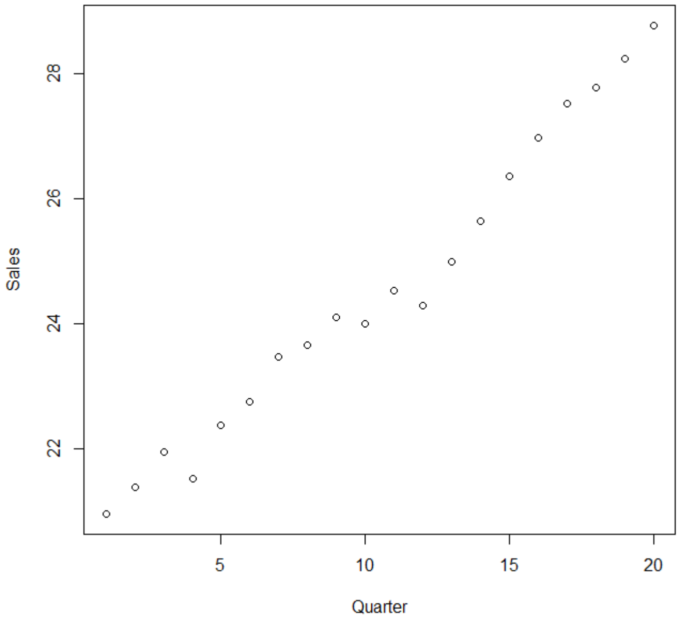
```
> plot(out$residuals ~ IndSales)
```

Not sure… there could be a pattern here.



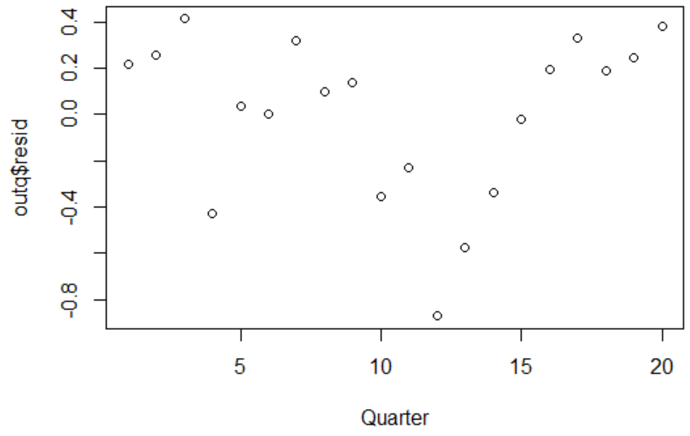
(b) Let’s look at the company’s sales vs. time (quarter):

```
> plot(Sales ~ Quarter)
> outq <- lm(Sales ~ Quarter)
> summary(outq)

Call:
lm(formula = Sales ~ Quarter)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.8735 -0.2578  0.1170  0.2482  0.4136 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 20.33742    0.16906  120.30   <2e-16 ***
Quarter      0.40301    0.01411   28.56   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3639 on 18 degrees of freedom
Multiple R-squared:  0.9784,	Adjusted R-squared:  0.9772 
F-statistic: 815.4 on 1 and 18 DF,  p-value: < 2.2e-16
```



Again, there might be a pattern to these residuals here… especially with respect to the quarter:

```
> plot(outq$resid ~ Quarter)
```



```
> library(car)
Warning message:
package ‘car’ was built under R version 3.0.3
> durbinWatsonTest(out)
 lag Autocorrelation D-W Statistic p-value
   1       0.6260046     0.7347256       0
 Alternative hypothesis: rho != 0
> durbinWatsonTest(outq)
 lag Autocorrelation D-W Statistic p-value
   1       0.5118392     0.8947571   0.002
 Alternative hypothesis: rho != 0
```

The test indicates serial correlation in both cases! This makes perfect sense if you think about it…

6.

Source	D.F.	SS	MS	F^*	p-value
Regression (Model)	1	34.783	34.783	98.4378	3.61×10^{-9}
Residual (Error)	20	7.067	0.35335	- - - -	
Lack of Fit	5	4.957	.9914	7.04787	0.001422
Pure Error	15	2.110	$.140\overline{6}$	∼ ∼	
Total	21	41.85			