HW2

Adhrit Srivastav; Ams22362

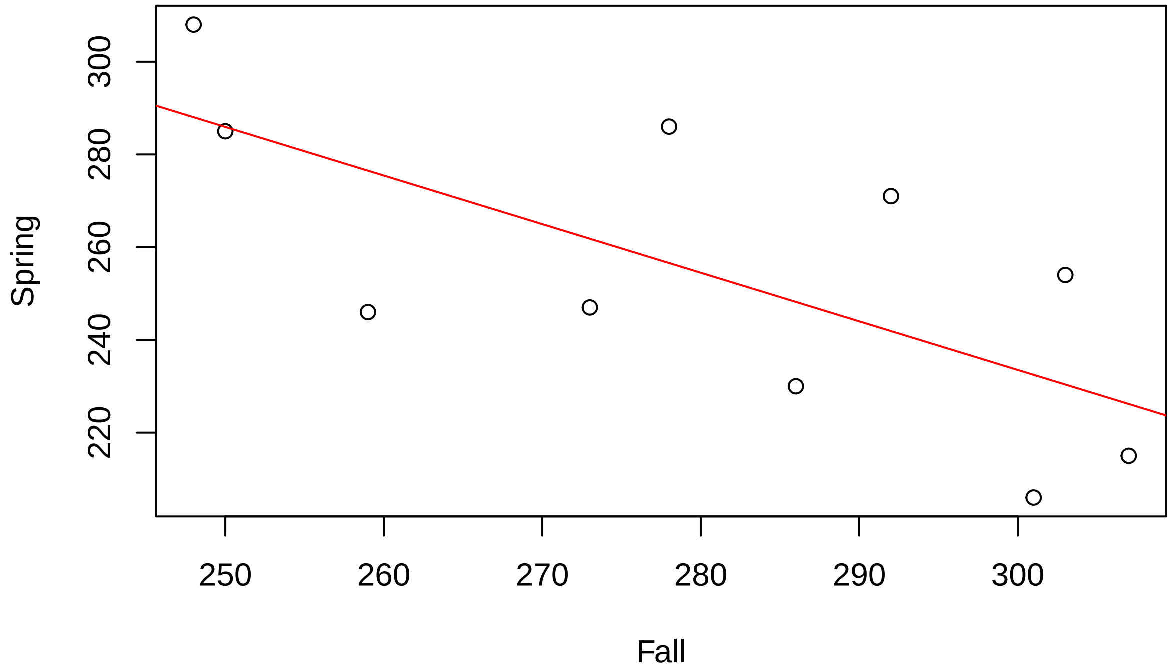
9/10/21

2.28 Enrollment in mathematics courses

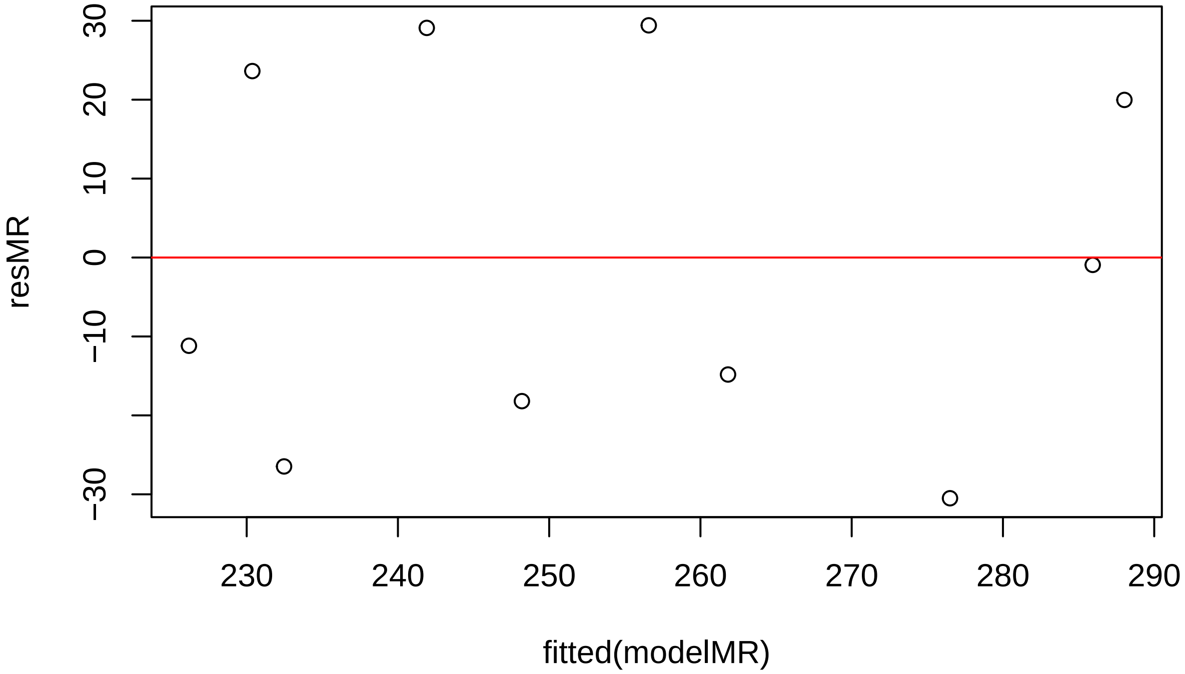
```
mathEnrollment = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/chpt2/datasets/Ma
attach(mathEnrollment)

# remove row 2 (AYear 2003)
MathEnrollmentR = mathEnrollment[-3,]
attach(MathEnrollmentR)
```

```
## The following objects are masked from mathEnrollment:
##
##     AYear, Fall, Spring
```

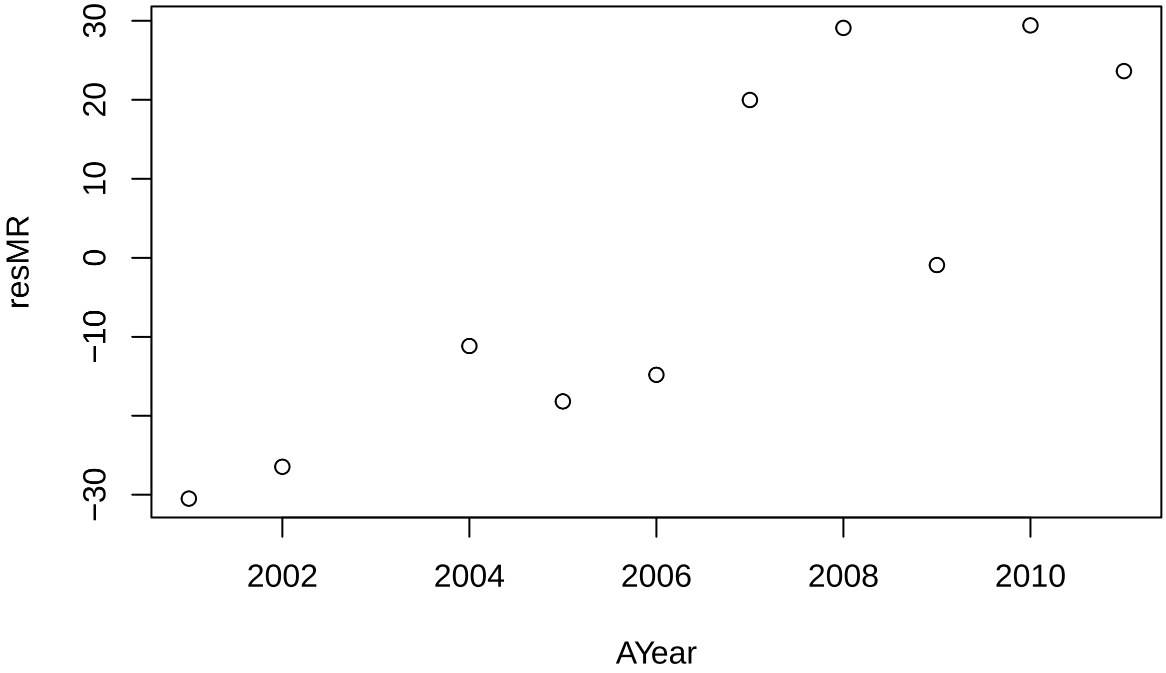
```
plot(Spring ~ Fall)
modelMR = lm(Spring ~ Fall)
abline(modelMR, col = 'red')
```



```
# residual plot
resMR = resid(modelMR)
plot(fitted(modelMR), resMR)
abline(h = 0, col = 'red')
```



```
plot(resMR ~ AYear)
```



```
summary(modelMR)
```

```
##
## Call:
## lm(formula = Spring ~ Fall)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.500 -17.353  -6.058  22.711  29.418
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 548.0094   106.7236   5.135 0.000891 ***
## Fall         -1.0483     0.3805  -2.755 0.024870 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.94 on 8 degrees of freedom
## Multiple R-squared:  0.4868, Adjusted R-squared:  0.4227
## F-statistic: 7.589 on 1 and 8 DF,  p-value: 0.02487
```

```
anova(modelMR)
```

```
## Analysis of Variance Table
##
```

```
## Response: Spring
##           Df Sum Sq Mean Sq F value  Pr(>F)
## Fall       1 4721.1  4721.1  7.5895 0.02487 *
## Residuals  8 4976.5   622.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Confidence Interval for a Prediction
new = data.frame(Fall=-1.0483)
predict(modelMR, new, se.fit=T, interval='confidence', level=0.95)
```

```
## $fit
##        fit      lwr      upr
## 1 549.1083 302.0859 796.1308
##
## $se.fit
## [1] 107.1214
##
## $df
## [1] 8
##
## $residual.scale
## [1] 24.94113
```

```
-1.0483 - qt(.975, 8)*0.3805
```

```
## [1] -1.925735
```

```
-1.0483 + qt(.975, 8)*0.3805
```

```
## [1] -0.1708654
```

a) There is a positive trend. The fact that there is a trend indicates that the independence test is not passed.

b) 48.68% of variability in spring enrollment is explained by using a simple linear model with fall enrollment as the predictor.

c) Provided in the code output.

d) From the ANOVA table output, we can see the F test statistic is greater than 1 and the p=value is very close to 0.02487 (and less than .05), which gives us evidence that there is significant linear association between spring and fall enrollments.

e) The 95% confidence interval for the slope of this model is =1.925735 to -0.1708654 This interval does not contain zero which means there is a statistically significant difference between the fall and spring enrollments.

2.34 Moth eggs

```
moth = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/chpt2/datasets/MothEggs.cs
attach(moth)

# correlation coefficient between the variables
cor(BodyMass, Eggs)
```

```
## [1] 0.4413396
```

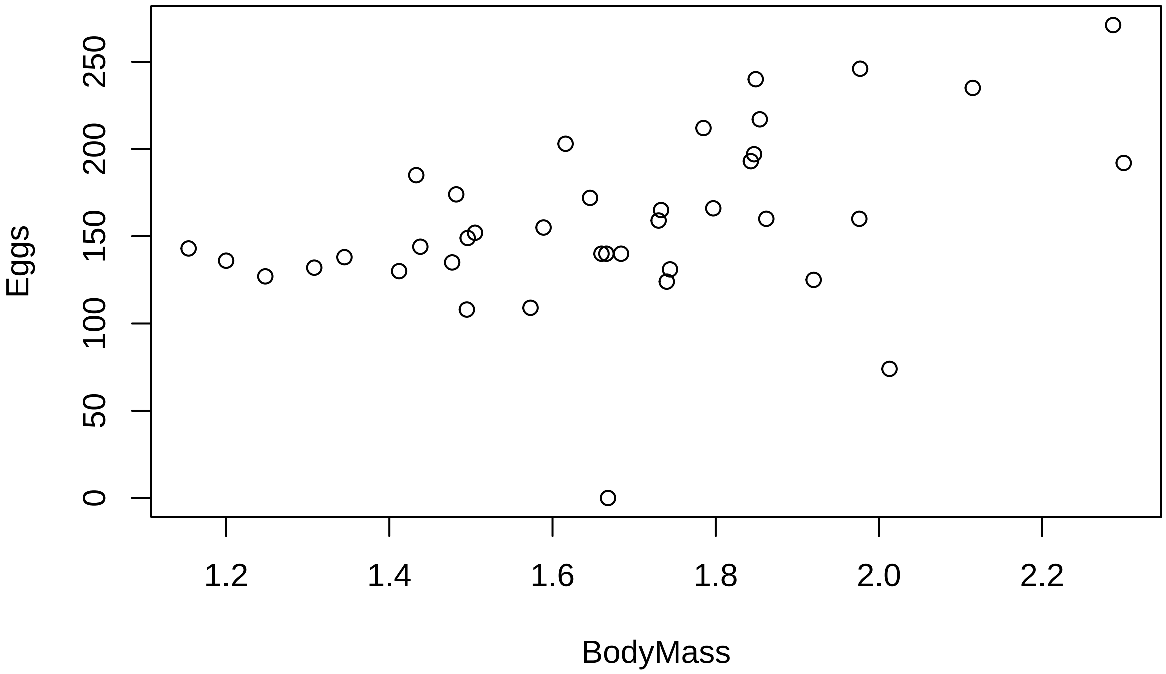
```
# get the p-value to determine statistical significance of correlation coefficient
modelM = lm(Eggs ~ BodyMass)
anova(modelM)
```

```
## Analysis of Variance Table
##
## Response: Eggs
##           Df Sum Sq Mean Sq F value   Pr(>F)
## BodyMass   1  17926 17926.4  8.9502 0.004916 **
## Residuals 37  74107  2002.9
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
summary(modelM)
```

```
##
## Call:
## lm(formula = Eggs ~ BodyMass)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -157.586  -17.187    3.162   25.790   67.960
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    24.38      45.38   0.537  0.59423
## BodyMass       79.86      26.69   2.992  0.00492 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 44.75 on 37 degrees of freedom
## Multiple R-squared:  0.1948, Adjusted R-squared:  0.173
## F-statistic:  8.95 on 1 and 37 DF,  p-value: 0.004916
```

```
plot(Eggs ~ BodyMass)
```



a) I would expect there to be a positive association between body mass and number of eggs because a moth with more mass is likely to be healthier. Healthier organisms tend to have increased fertility and in this case more numbers of eggs.

b) The correlation coefficient for measuring the strength of linear association between BodyMass and Eggs is .441.

c) The p-value is .004916 which is less than the significance value (.05) and thus indicates that the association between the two variables is statistically significant.

d) $E\hat{g}gs = 24.38 + 79.86$*BodyMass

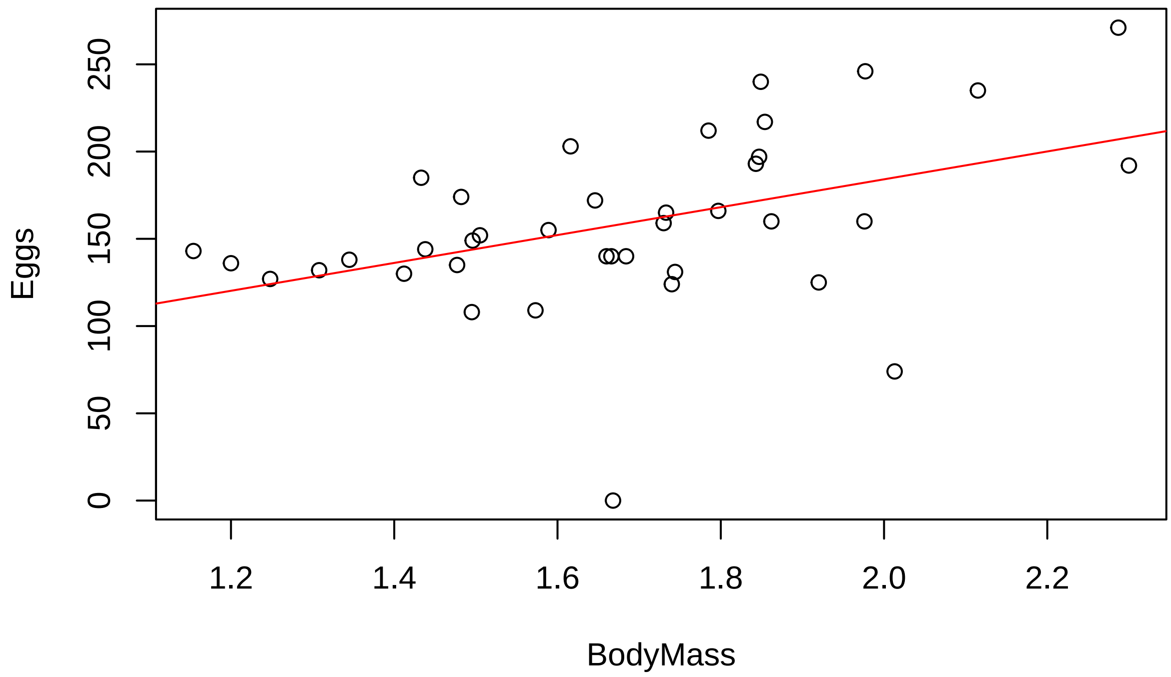
e) There is one outlier where the BodyMass was 1.668 and the number of Eggs were 0.

2.35 Moth eggs: subset

```
moth = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/chpt2/datasets/MothEggs.cs
attach(moth)
```

```
## The following objects are masked from moth (pos = 3):
##
##     BodyMass, Eggs
```

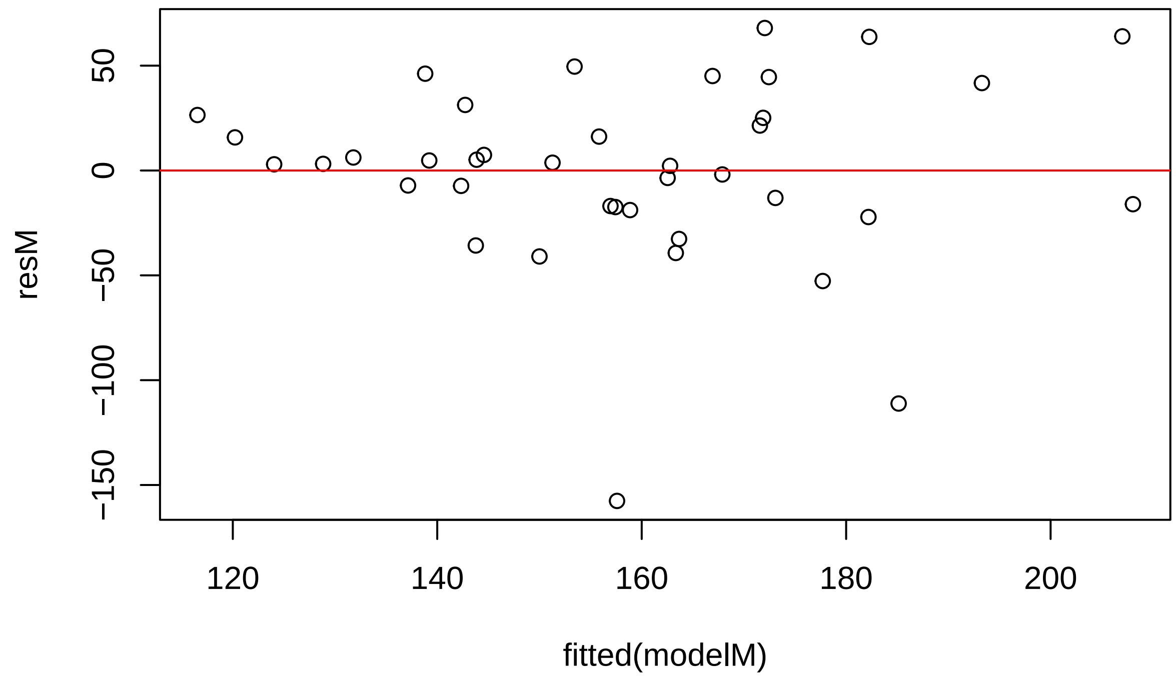
```
plot(Eggs ~ BodyMass)
modelM = lm(Eggs ~ BodyMass)
abline(modelM, col = 'red')
```



```
summary(modelM)
```

```
##
## Call:
## lm(formula = Eggs ~ BodyMass)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -157.586  -17.187    3.162   25.790   67.960
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    24.38      45.38   0.537  0.59423
## BodyMass       79.86      26.69   2.992  0.00492 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.75 on 37 degrees of freedom
## Multiple R-squared:  0.1948, Adjusted R-squared:  0.173
## F-statistic:  8.95 on 1 and 37 DF,  p-value: 0.004916
```

```
# residual plot
resM = resid(modelM)
plot(fitted(modelM), resM)
abline(h = 0, col = 'red')
```



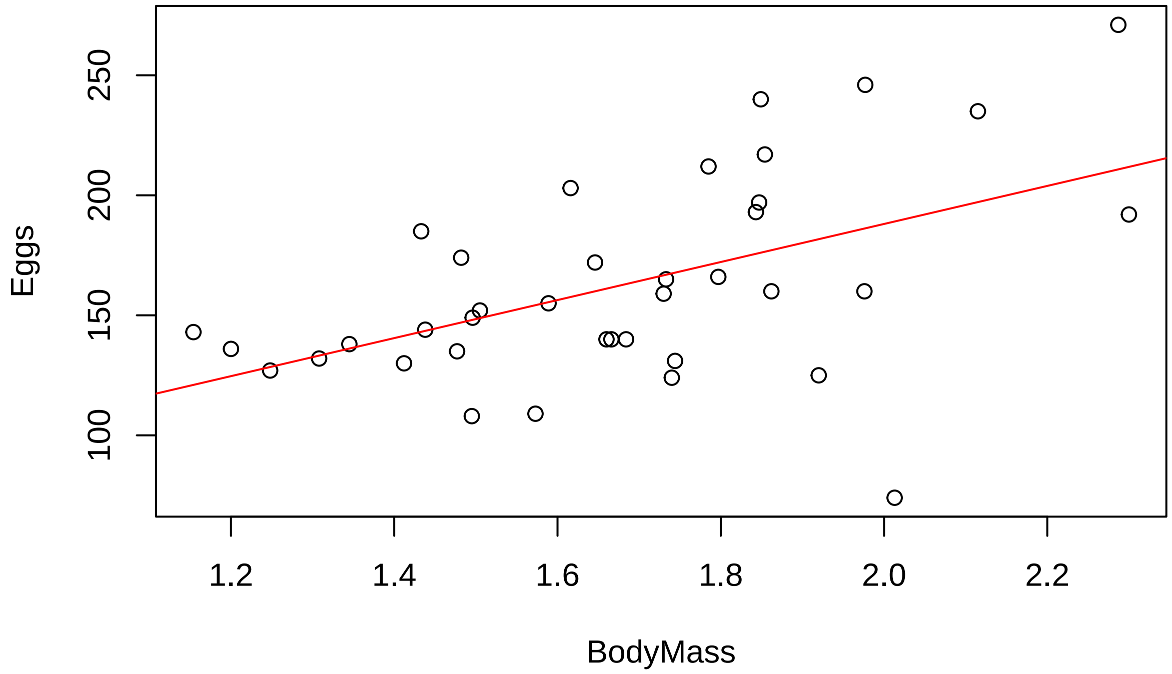
```
nrow(moth)
```

```
## [1] 39
```

```
mothR = moth[-39,]
attach(mothR)
```

```
## The following objects are masked from moth (pos = 3):
##
##     BodyMass, Eggs

## The following objects are masked from moth (pos = 4):
##
##     BodyMass, Eggs
```

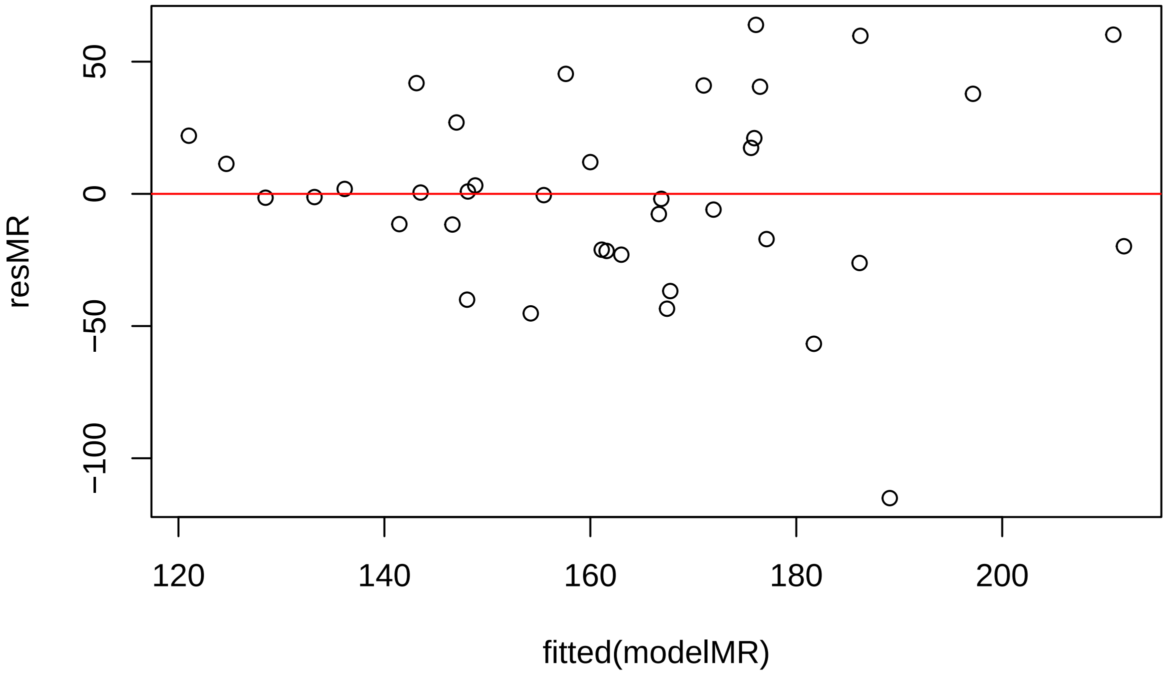
```
plot(Eggs ~ BodyMass)
modelMR = lm(Eggs ~ BodyMass)
abline(modelMR, col = 'red')
```



```
summary(modelMR)
```

```
##
## Call:
## lm(formula = Eggs ~ BodyMass)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -115.079  -20.785   -0.846   21.763   63.917
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    29.56      37.28   0.793 0.433043
## BodyMass       79.24      21.92   3.615 0.000911 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 36.75 on 36 degrees of freedom
## Multiple R-squared:  0.2664, Adjusted R-squared:  0.246
## F-statistic: 13.07 on 1 and 36 DF,  p-value: 0.0009108
```

```
# residual plot
resMR = resid(modelMR)
plot(fitted(modelMR), resMR)
abline(h = 0, col = 'red')
```



a) $\hat{Eggs} = 29.56 + 79.24$*BodyMass

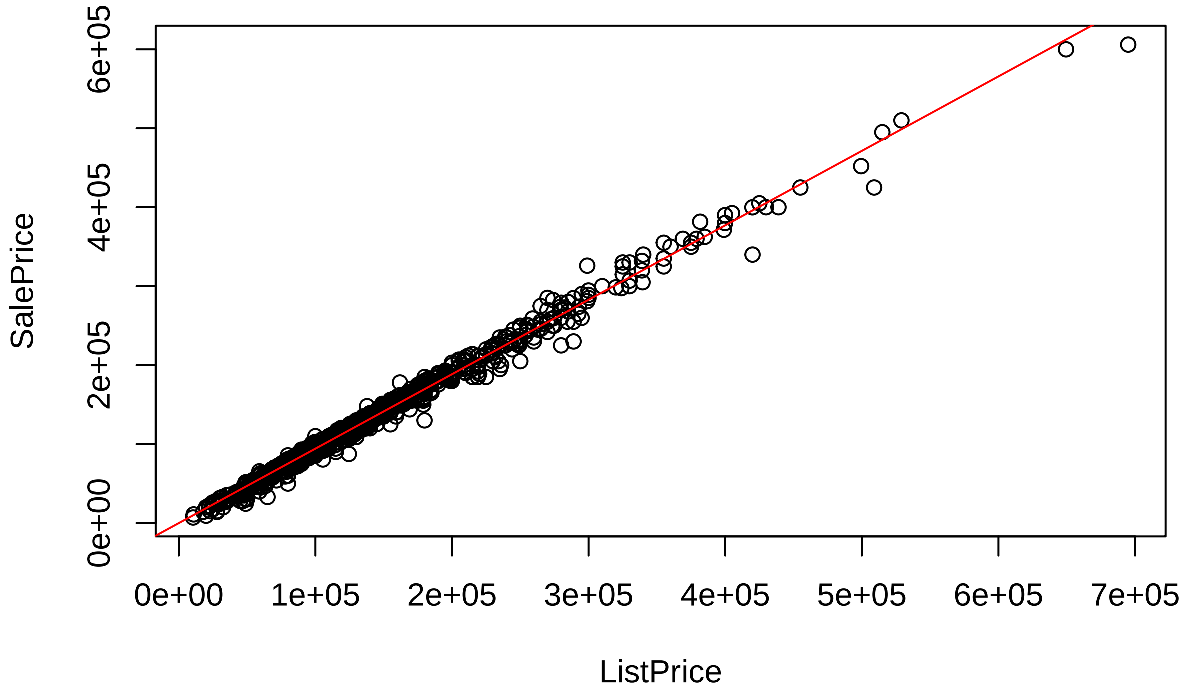
b) The conditions are not met because the residual plot's data points fanned out, indicating it did not pass the equal variance condition.

c) The estimated slope with the moth that had no eggs was 79.86 while the estimated slope without the moth with no eggs was 79.24.

d) The percent of variability with the moth that had no eggs is 19.48% and the percent of variability without the moth that had no eggs is 26.64%.

2.40 Predicting Grinnell house prices

```
gHouses = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/chpt2/datasets/Grinnell
attach(gHouses)

plot(SalePrice ~ ListPrice)
modelG = lm(SalePrice ~ ListPrice)
abline(modelG, col = 'red')
```



```
summary(modelG)
```

```
##
## Call:
## lm(formula = SalePrice ~ ListPrice)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -55942  -3275    846   4141  44168
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.448e+02  5.236e+02  -0.277    0.782
## ListPrice    9.431e-01  3.201e-03 294.578   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 8019 on 927 degrees of freedom
## Multiple R-squared:  0.9894, Adjusted R-squared:  0.9894
## F-statistic: 8.678e+04 on 1 and 927 DF,  p-value: < 2.2e-16
```

```
# predict mean SalePrice for a home listed at $300,000
#predSPrice = predict(modelG, data.frame(ListPrice = 300000))
```

```r
#Confidence Interval for a Prediction
#newG = data.frame(ListPrice = predSPrice)
#predict(modelG, newG, se.fit=T, interval='confidence', level=0.90)

#Prediction Interval for a Prediction
#predict(modelG, newG, se.fit=T, interval='prediction', level=0.90)




# predict mean Price for a textbook with 450 pages
newS = data.frame(ListPrice = 300000)

#Confidence Interval for a Prediction
predict(modelG, newS, se.fit=T, interval = "confidence", level = .95)
```

```
## $fit
##        fit      lwr      upr
## 1 282774.6 281652.3 283896.9
##
## $se.fit
## [1] 571.8653
##
## $df
## [1] 927
##
## $residual.scale
## [1] 8019.288
```

```r
#Prediction Interval for a Prediction
predict(modelG, newS, se.fit=T, interval = "prediction", level = .95)
```

```
## $fit
##        fit      lwr      upr
## 1 282774.6 266996.6 298552.6
##
## $se.fit
## [1] 571.8653
##
## $df
## [1] 927
##
## $residual.scale
## [1] 8019.288
```

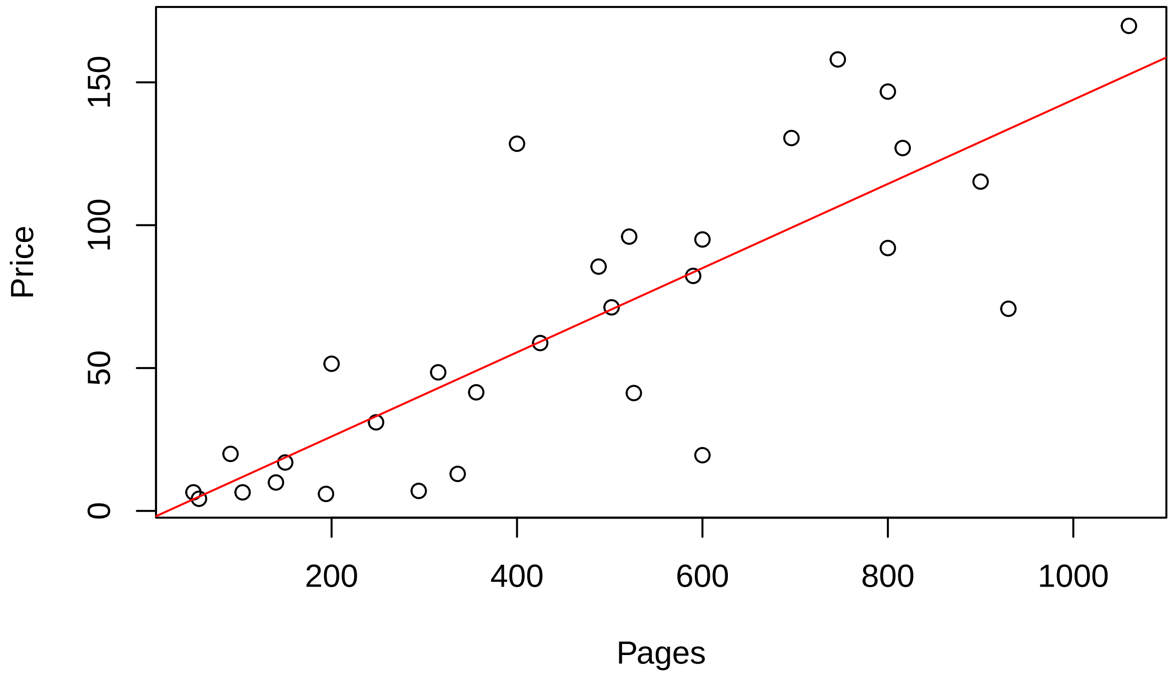
a) The 90% Confidence Interval for the mean SalePrice of a home that is listed at $300,000 is 281652.3 to 283896.9.

b) The 90% Prediction Interval for the mean SalePrice of a home that is listed at $300,000 is 266996.6 to 298552.6.

c) The Prediction Interval indicates that all future individual observations of houses listed at $300,000 will be between 281652.3 and 283896.9 dollars. The Confidence Interval indicates that the likely range of values associated with a home listed at $300,000 will be between 266996.6 and 298552.6 dollars. We are 90% confident the mean μ_Y SalePrice is between 281652.3 and 283896.9 dollars for the predictor $x^* = 300000$. 90% of the singular observed SalePrices y is between 266996.6 and 298552.6 dollars for the predictor $x^* = 300000$. This makes sense because in the prediction interval, we are predicting a specific value which is much more difficult and will result in a wider interval. Whereas in the confidence interval we are predicting a mean value which is much easier and thus has a narrower interval.

2.44 Tetbook prices: regression intervals

```
txtPrice = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/chpt2/datasets/TextPri
attach(txtPrice)

plot(Price ~ Pages)
modelT = lm(Price ~ Pages)
abline(modelT, col = 'red')
```



```
summary(modelT)
```

```
##
## Call:
## lm(formula = Price ~ Pages)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -65.475 -12.324  -0.584  15.304  72.991
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.42231   10.46374  -0.327    0.746
## Pages        0.14733    0.01925   7.653 2.45e-08 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 29.76 on 28 degrees of freedom
## Multiple R-squared:  0.6766, Adjusted R-squared:  0.665
## F-statistic: 58.57 on 1 and 28 DF,  p-value: 2.452e-08
```

```
# predict mean Price for a textbook with 450 pages
newT = data.frame(Pages=450)

#Confidence Interval for a Prediction
predict(modelT,newT, se.fit=T, interval = "confidence", level = .95)
```

```
## $fit
##        fit      lwr      upr
## 1 62.87549 51.73074 74.02024
##
## $se.fit
## [1] 5.440691
##
## $df
## [1] 28
##
## $residual.scale
## [1] 29.76046
```

```
#Prediction Interval for a Prediction
predict(modelT,newT, se.fit=T,interval = "prediction", level = .95)
```

```
## $fit
##        fit       lwr      upr
## 1 62.87549 0.9035981 124.8474
##
## $se.fit
## [1] 5.440691
##
## $df
## [1] 28
##
## $residual.scale
## [1] 29.76046
```

```
# midpoint of confidence interval
sum1 = (51.73074 + 74.02024)/2

# midpoint of prediction interval
```

```
sum2 = (0.9035981 + 124.8474)/2

# narrowest prediction interval for price
mean(Pages)
```

```
## [1] 464.5333
```

```
# predict mean Price for a textbook with 450 pages
newTM = data.frame(Pages=1500)

#Prediction Interval for a Prediction
predict(modelT, newTM, se.fit=T, interval = "prediction", level = .95)
```

```
## $fit
##        fit      lwr     upr
## 1 217.5704 143.3587 291.782
##
## $se.fit
## [1] 20.66038
##
## $df
## [1] 28
##
## $residual.scale
## [1] 29.76046
```

a) The 95% confidence interval for the mean price of a 450 page textbook is 51.73074 to 74.02024 dollars.

b) The 95% prediction interval for the mean price of a 450 page textbook is 0.9035981 to 124.8474 dollars.

c) The midpoint of the two intervals are exactly the same: 62.87549. This makes sense because the intervals are essentially the same except that the prediction interval is naturally wider than the confidence interval. The intervals are also centered on the same linear model, which further explains their same midpoint.

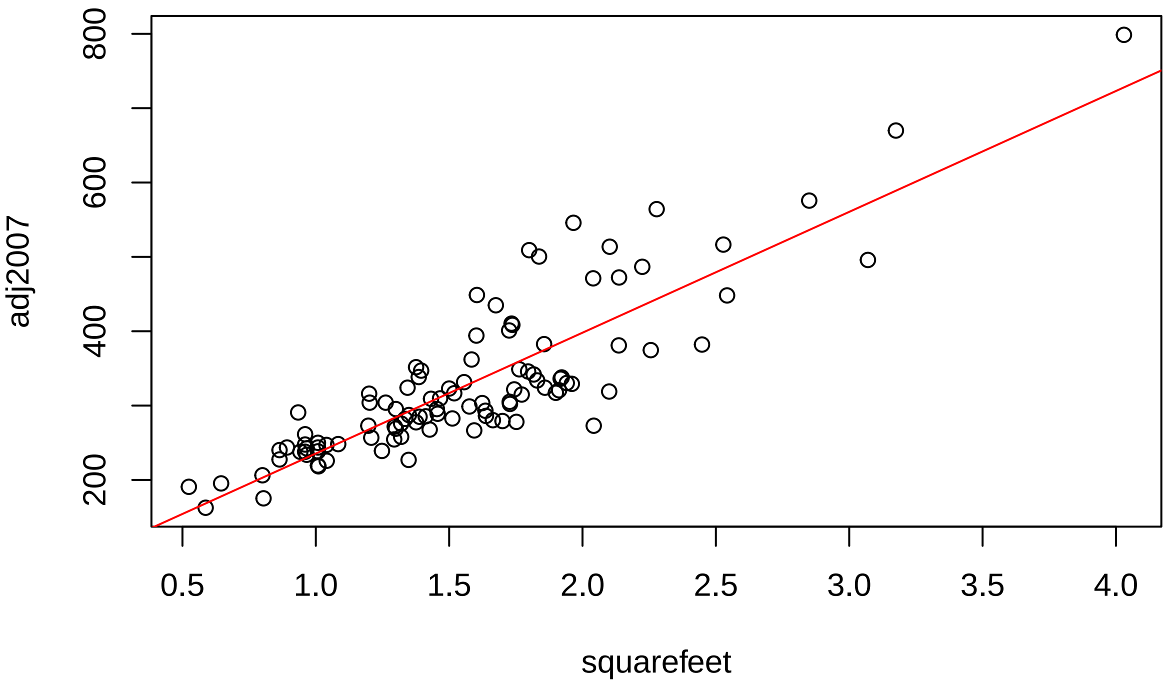
d) The width of the prediction interval is wider because the standard error is larger because it is much more difficult to predict a specific value versus a mean value.

e) 464 pages would produce the narrowest possible prediction interval for its price because it's the mean of all textbook pages which will minimize the standard error and thus reduce the prediction interval.

f) The 95% prediction interval for the price of a particular 1500 page textbook is 143.3587 to 291.782 dollars. I am not fully confident in this interval because 1500 pages is very far out of the range of the data. The data only goes until 1060 and we are making the model predict something far outside of its range and not with substantial data to allow us to be 95% confident in its interval.

2.46 Real estate near Rails to Trails: home size transformation

```
rTrails = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/chpt2/datasets/RailsTra
attach(rTrails)

plot(adj2007 ~ squarefeet)
modelRT = lm(adj2007 ~ squarefeet)
abline(modelRT, col = 'red')
```



```
summary(modelRT)
```

```
##
## Call:
## lm(formula = adj2007 ~ squarefeet)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -131.940  -36.635   -4.109   32.470  153.323
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   72.973     15.541   4.695 8.32e-06 ***
## squarefeet   162.526      9.351  17.381  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 53 on 102 degrees of freedom
## Multiple R-squared:  0.7476, Adjusted R-squared:  0.7451
## F-statistic: 302.1 on 1 and 102 DF,  p-value: < 2.2e-16
```

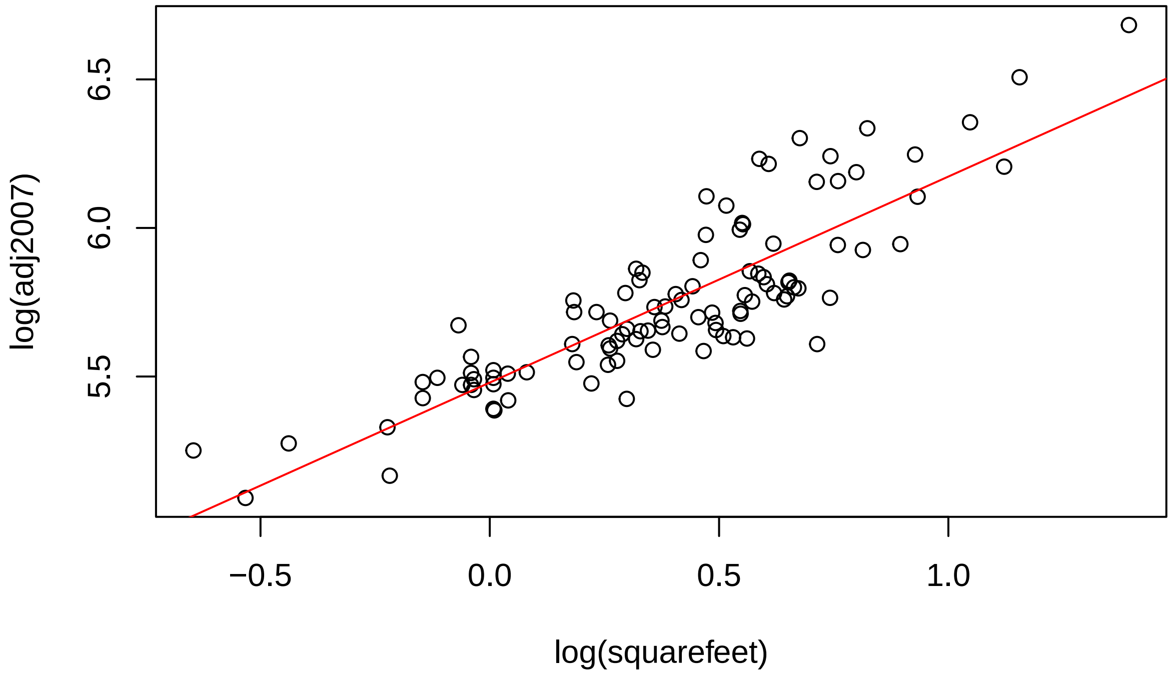
```
predict(modelRT, data.frame(squarefeet = 1.5))
```

```
##        1
## 316.7619
```

```
newRT = data.frame(squarefeet = 1.5)
predict(modelRT,newRT, se.fit=T,interval = "prediction", level = .95)
```

```
## $fit
##        fit      lwr      upr
## 1 316.7619 211.1232 422.4006
##
## $se.fit
## [1] 5.234139
##
## $df
## [1] 102
##
## $residual.scale
## [1] 53.00104
```

```
# log transform data
plot(log(adj2007) ~ log(squarefeet))
modelRTL = lm(log(adj2007) ~ log(squarefeet))
abline(modelRTL, col = 'red')
```



```
summary(modelRTL)
```

```
##
```

```
## Call:
## lm(formula = log(adj2007) ~ log(squarefeet))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36543 -0.11094 -0.01864  0.10474  0.35402
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.47958    0.02137  256.41   <2e-16 ***
## log(squarefeet)  0.69334    0.04082   16.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1461 on 102 degrees of freedom
## Multiple R-squared:  0.7388, Adjusted R-squared:  0.7362
## F-statistic: 288.5 on 1 and 102 DF,  p-value: < 2.2e-16
```

```
newRT = data.frame(squarefeet = 1.5)
predict(modelRTL,newRT, se.fit=T,interval = "prediction", level = .95)
```

```
## $fit
##        fit      lwr      upr
## 1 5.760702 5.469522 6.051882
##
## $se.fit
## [1] 0.01434299
##
## $df
## [1] 102
##
## $residual.scale
## [1] 0.1460991
```

```
# prediction interval back to dollars
exp(5.469522)
```

```
## [1] 237.3467
```

```
exp(6.051882)
```

```
## [1] 424.912
```

a) The selling price of a home that is 1500 square feet in size is 316.7619 thousand dollars.

b) The 95% prediction interval is [211.1232, 422.4006]. This means the model is 95% confidence predicting a singular adj2007 value from the range of 211.1232 to 422.4006. 95% of the singular observed prices y are between 211.1232 and 422.4006 thousand dollars for the predictor $x^* = 1.5$.

c) The model does adhere to conditions except that there is one outlier to the top right of the scatter plot. This is likely having some degree of effect to the model and the prediction interval in part b. The data also fans out the prediction intervals don't account for the increasing variance as the squarefootages increases.

d) The data has become less spread out with the logarithmic transformation. Numerically, the r-squared values are still similar, but visually the data appears to be better fit and less sparse. The outlier in the untransformed data does not appear to be as much of an outlier in the transformed data.

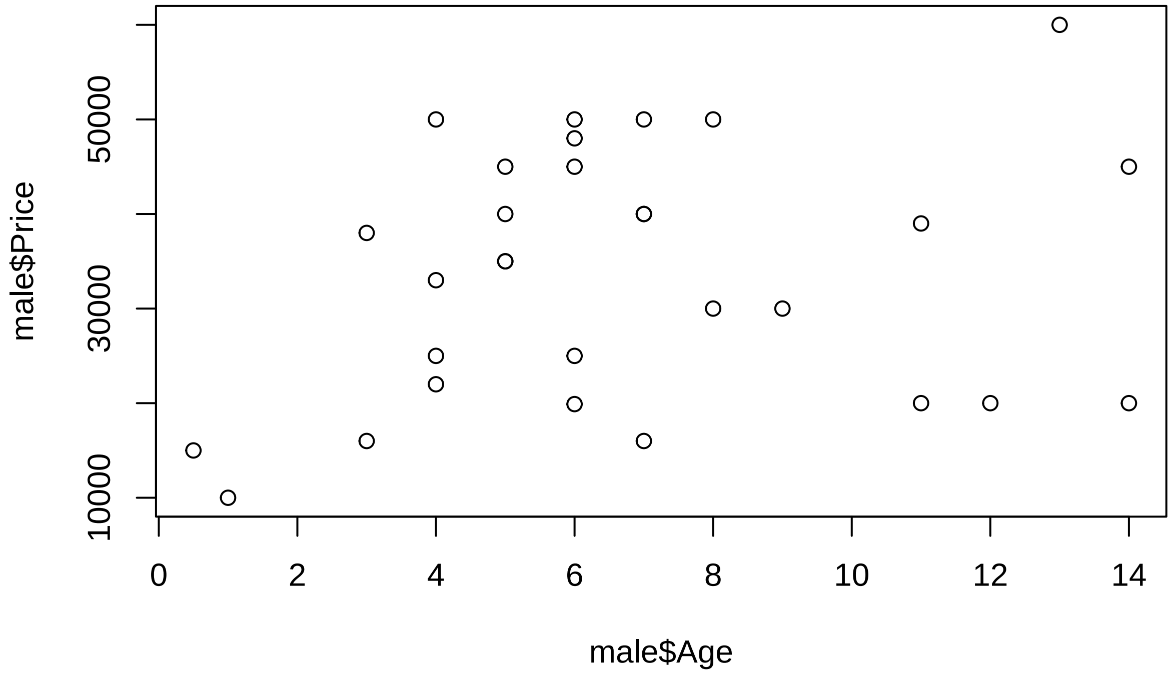
e) The prediction interval for the transformed data is [237.3467, 424.912]. This is a narrower prediction interval to the untransformed data indicating that the model will predict a singular value with 95% confidence with a more accurate degree because its range of predicable values is smaller.

2.57 Horses for sale

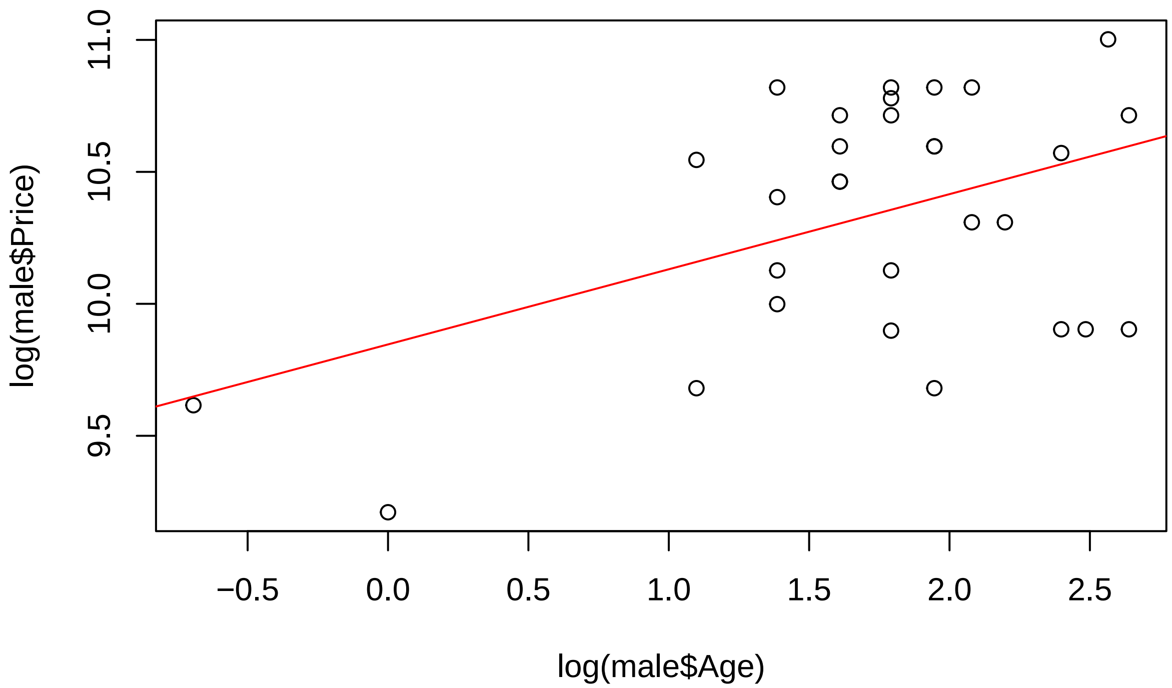
```
hPrices = read.csv("C:/Users/adhri/OneDrive/Documents/R/App_Reg_and_Time_Series/chpt2/datasets/HorsePri
attach(hPrices)
```

```
## The following object is masked from txtPrice:
##
##     Price
```

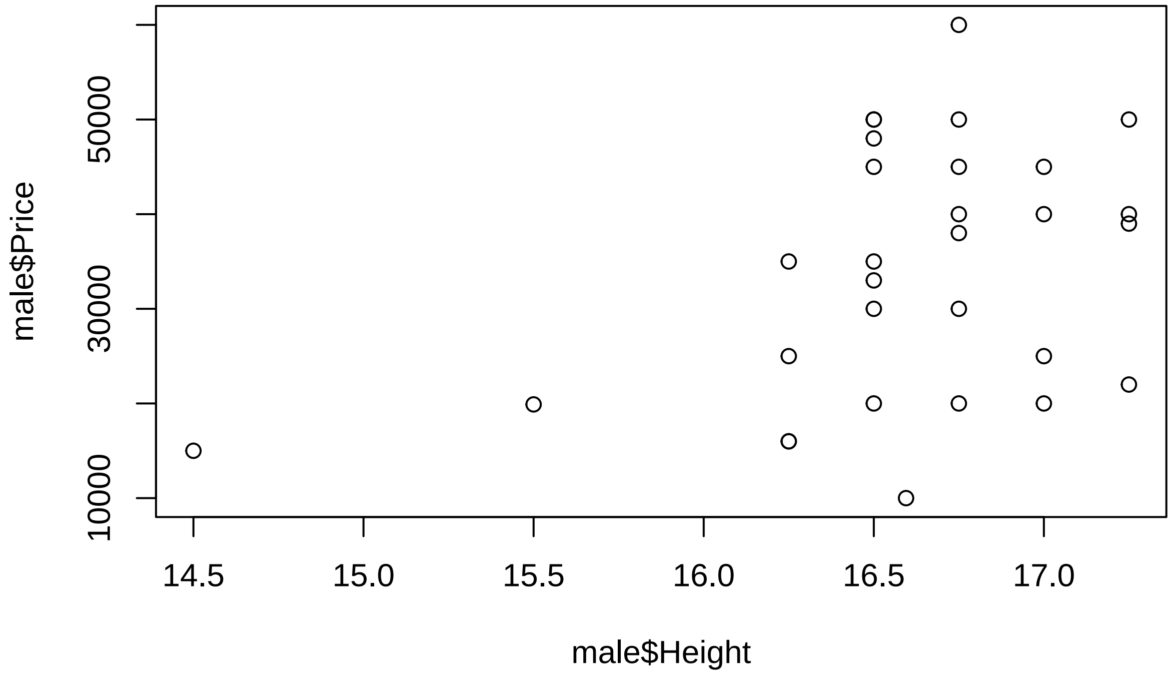
```
# split data so one df is just males and one is just females
male = subset(hPrices,Sex %in% "m")
male$Height[is.na(male$Height)] = mean(male$Height, na.rm = TRUE)  # Replace NA with mean height
female = subset(hPrices,Sex %in% "f")
female$Height[is.na(female$Height)] = mean(female$Height, na.rm = TRUE)  # Replace NA with mean height


# plot all the different variables against price
#male
#plot(male$Price ~ male$Age)
#plot(male$Price ~ male$Height)

#female
#plot(female$Price ~ female$Age)
#plot(female$Price ~ female$Height)

#transform the data
#male
#age
plot(male$Price ~ male$Age)    # benchmark, untransformed
```



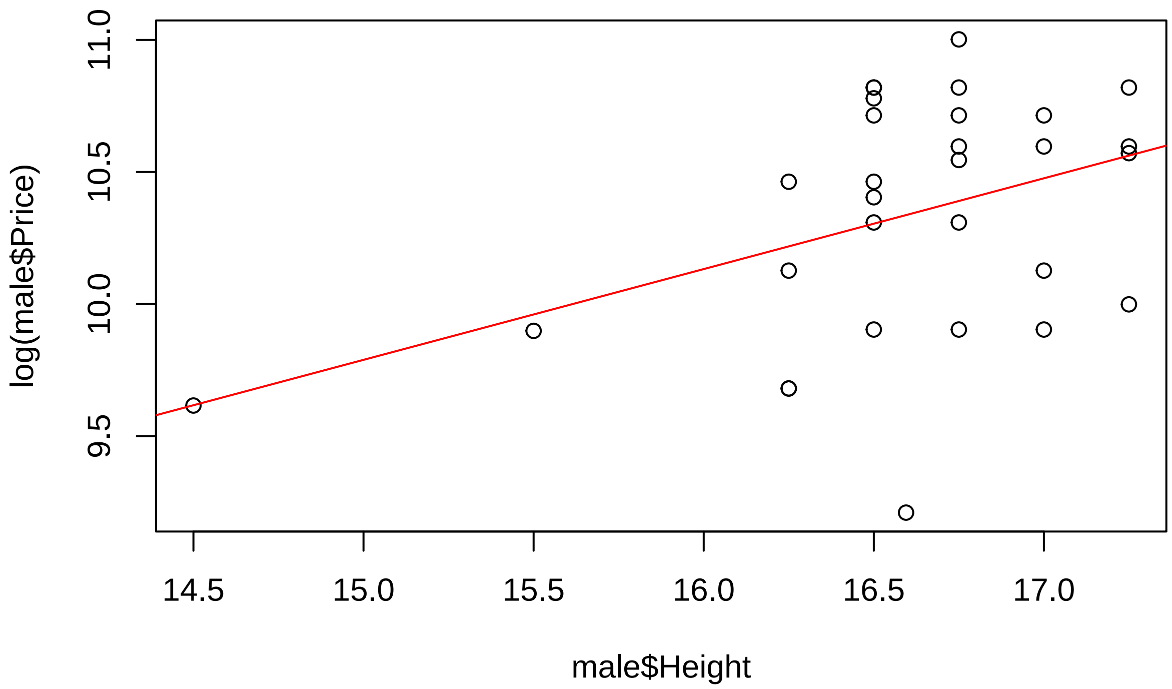
```
#plot(log(male$Price) ~ male$Age)
#plot(male$Price ~ log(male$Age))
plot(log(male$Price) ~ log(male$Age))     # best scatterplot
modelMA = lm(log(male$Price) ~ log(male$Age))
abline(modelMA, col = 'red')
```



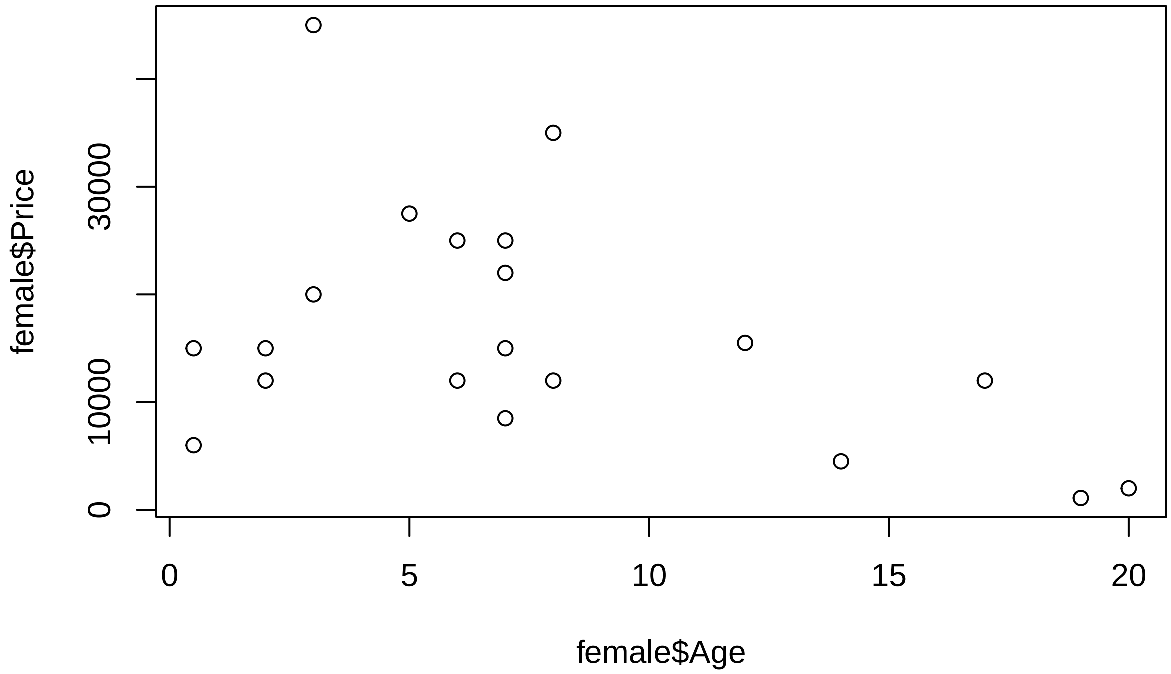
```
#height
plot(male$Price ~ male$Height)    # benchmark, untransformed
```



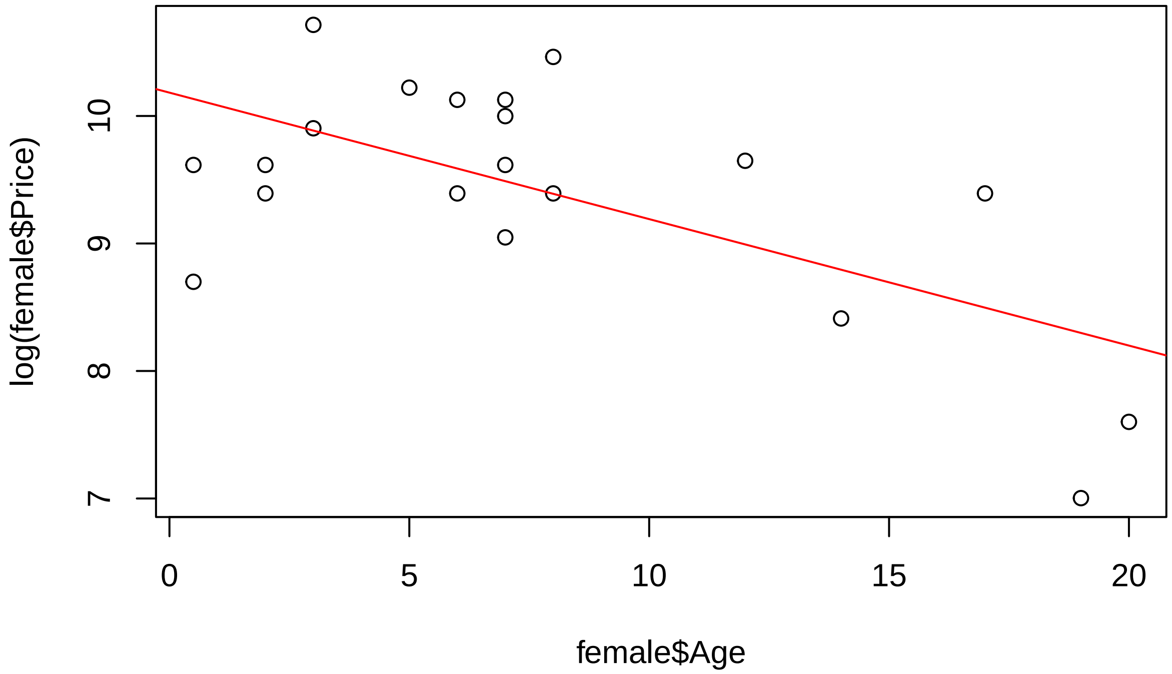
```
plot(log(male$Price) ~ male$Height)     # best scatterplot
#plot(male$Price ~ log(male$Height))
#plot(log(male$Price) ~ log(male$Height))
modelMH = lm(log(male$Price) ~ male$Height)
abline(modelMH, col = 'red')
```



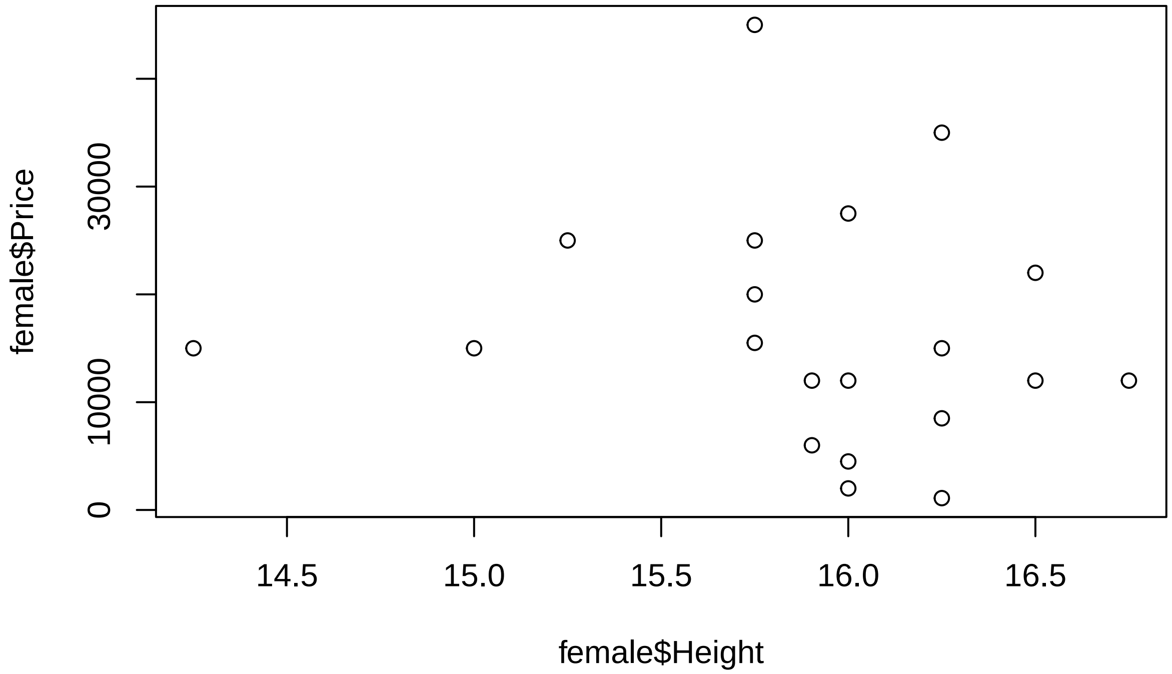
```
#female
#age
plot(female$Price ~ female$Age)     # benchmark, untransformed
```



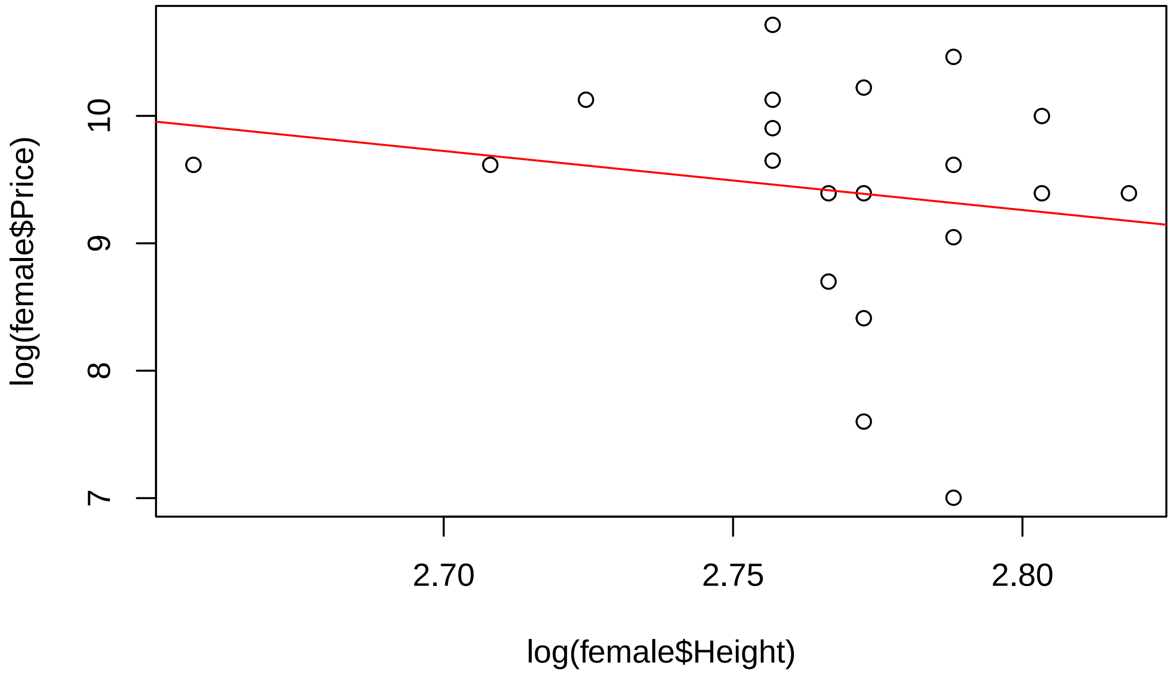
```
plot(log(female$Price) ~ female$Age)    # best scatterplot
#(female$Price ~ log(female$Age))
#plot(log(female$Price) ~ log(female$Age))
modelFA = lm(log(female$Price) ~ female$Age)
abline(modelFA, col = 'red')
```



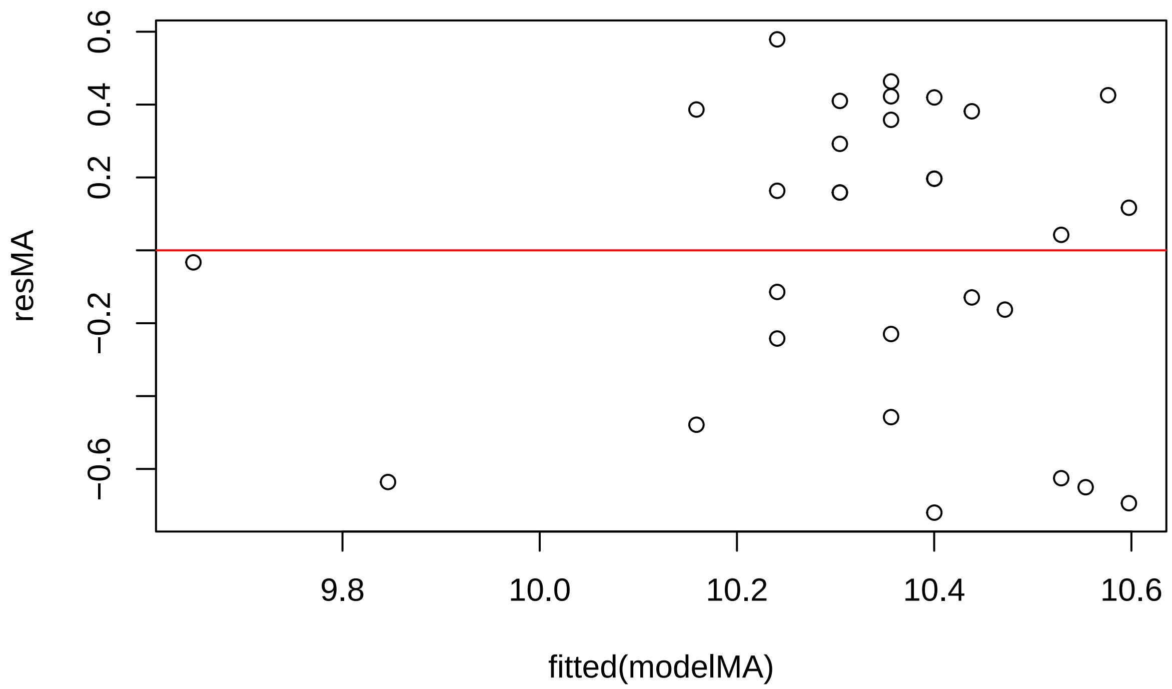
```
#height
plot(female$Price ~ female$Height)    # benchmark, untransformed
```



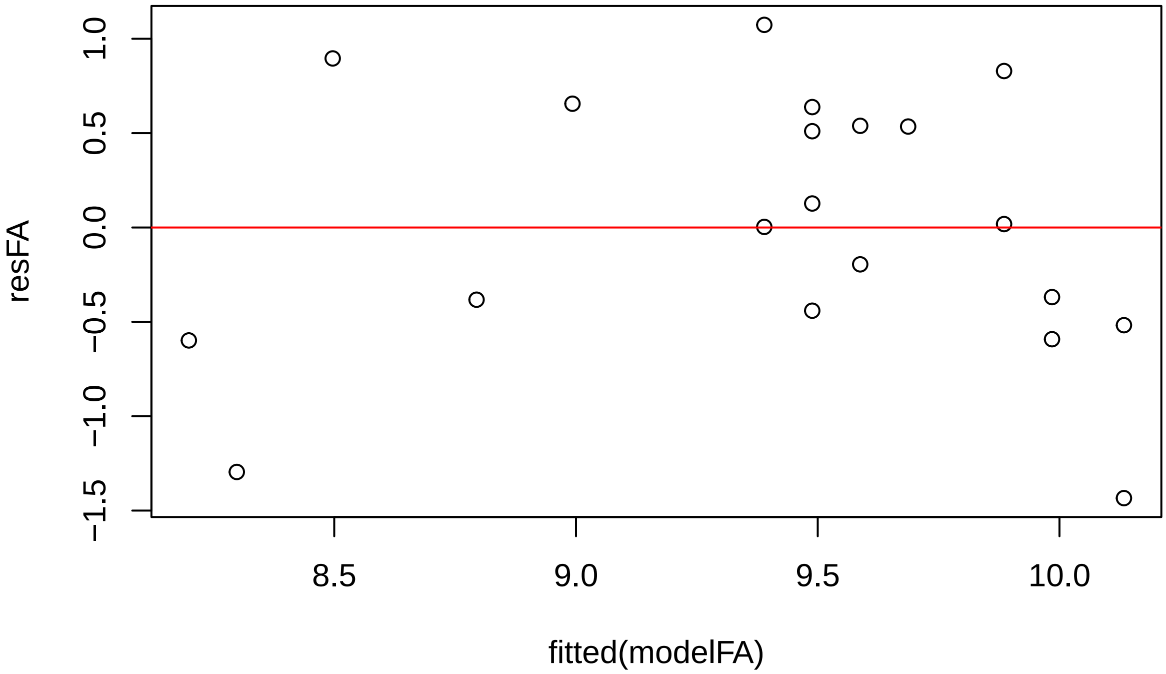
```
#(log(female$Price) ~ female$Height)
#plot(female$Price ~ log(female$Height))
plot(log(female$Price) ~ log(female$Height))    # best scatterplot
modelFH = lm(log(female$Price) ~ log(female$Height))
abline(modelFH, col = 'red')
```



```
# age is best predictor for male and female horse price


# plot residual plots

# male age
resMA = resid(modelMA)
plot(fitted(modelMA), resMA)
abline(h = 0, col = 'red')
```



```
# female age
resFA = resid(modelFA)
plot(fitted(modelFA), resFA)
abline(h = 0, col = 'red')
```



My first step was to look at the data. I noticed there were both male and female horses in the data, so I split the datasets into one male and one female dataset. I did this because there is a natural difference in size for males and females, and thus potentially a difference in price, so I wanted to keep the data disparate. I also noticed NA values in the Height column of both datasets, and filled them with the column average. I then began plotting the respective datasets and transforming them in efforts to achieve the best model. This felt like a slightly subjective process because some scatterplots ended up being clearly not useful, and some ended up being extremely similar. I felt that Age was the best predictor for both Male and Female horses through the scatterplots and applied linear models. I also created residual plots and the residual plot for Male Age had one outlier but other than that the data was nicely scattered around zero. The Female Age residual plot was also scattered randomly around zero. Thus I determined the models that were best for predicting horse prices were when I separated the data by sex. Next, I logarithmically transformed the Age and Price data for the male to get my final model. For the females, I logarithmically transformed the Price to get my final model.

2.61 Gate count: computing the least squares line from descriptive statistics

a) $\widehat{y}$ = 22576.6 + 111.826x
$\widehat{\beta}_1$ = 111.826, .701*(104807/657); $\bar{y}$ = 247235; $\bar{x}$ = 2009;
247235 = $\widehat{\beta}_0$ + 111.826*2009 -> $\widehat{\beta}_0$ = 22576.6

b) R^2 = .701ˆ2 = .4919; Thus 49.19% of variation in gate counts is explained by enrollments.

c) We just plug 1445 into the least squares line equation and the output is 184165.136 gate counts.

d) We plug 2200 into the least squares line equation and get the output of 268594. The actual gate count is 130000, so we subtract the values and get a residual of -138594.