Homework 2 - Stat 230 SLR II - Transformations, tests and intervals: Baseball, metabolic rate and more about Pines

Dhyey Mavani

PROBLEMS TO TURN IN: #1.45, #1.46 (slightly modified), #1.48, #2.30, Additional (see below)

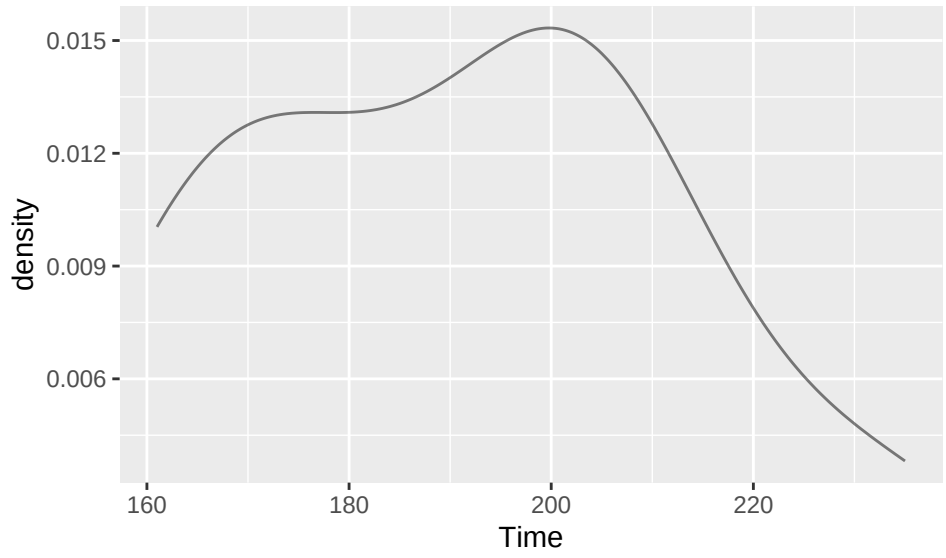
```
data(BaseballTimes2017)
```

Exercise 1.45

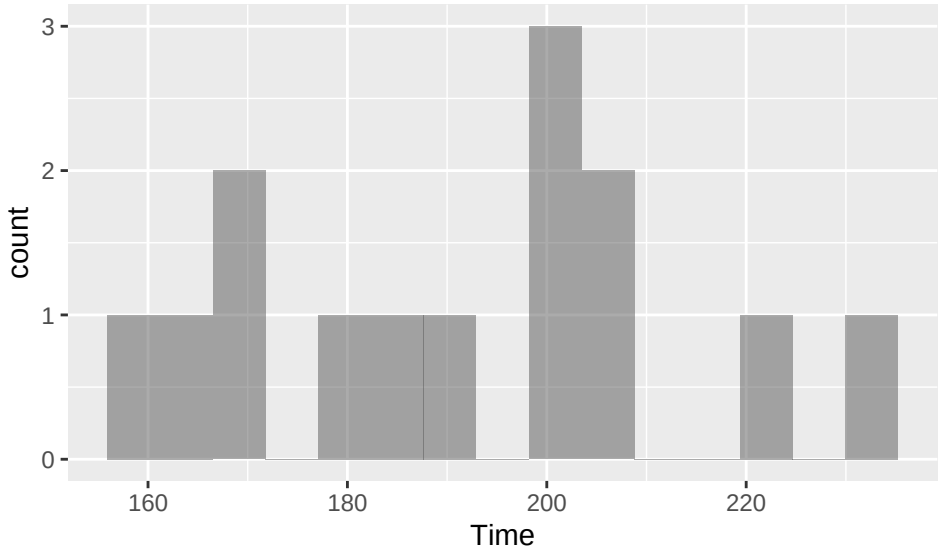
part a: Describe distribution of Time

SOLUTION: The distribution is skewed towards the right and we can see that the median is 194.5 with interquartile range of 171.5 to 203.75. It is relatively bimodal.

```
gf_dens(~ Time, data = BaseballTimes2017)
```




```
gf_histogram(~ Time, bins = 15, data = BaseballTimes2017)
```



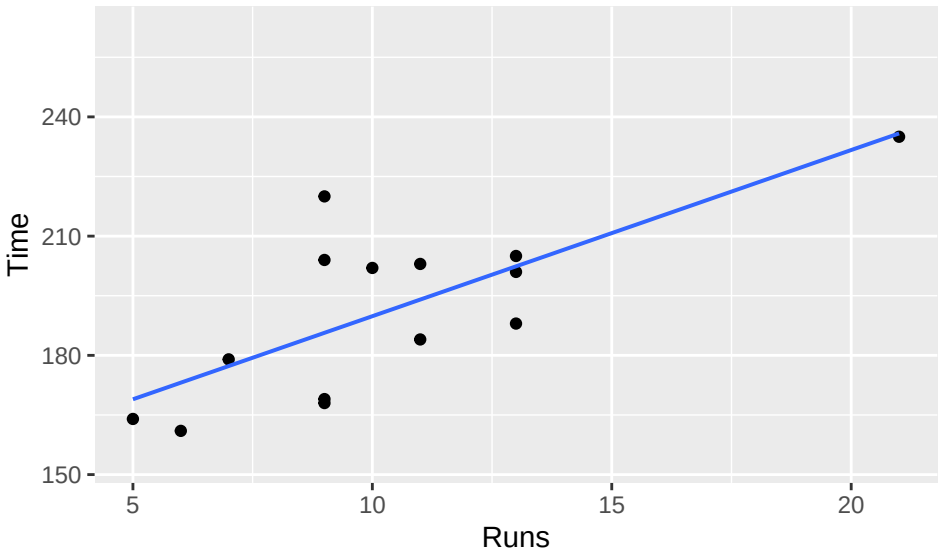

```
favstats(~ Time, data = BaseballTimes2017)
```

```
##  min    Q1 median     Q3 max    mean      sd  n missing
##  161 171.5  194.5 203.75 235 191.643 22.0928 14       0
```

part b: Check scatterplots for relationships

SOLUTION: The relationship between Time and Runs shows the highest R-squared value and highest correlation coefficient, hence I selected Runs as the best predictor for the Time.

```
#example plot, explore different options for predictor
gf_point(Time ~ Runs, data = BaseballTimes2017) %>%
  gf_lm()
```



```
fm <- lm(Time ~ Runs, data = BaseballTimes2017)
msummary(fm)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   148.04      12.00   12.34  3.5e-08 ***
## Runs            4.18       1.08    3.87   0.0022 **
##
## Residual standard error: 15.3 on 12 degrees of freedom
## Multiple R-squared:  0.555,  Adjusted R-squared:  0.518
## F-statistic:   15 on 1 and 12 DF,  p-value: 0.00224
```

```
cor(Time ~ Runs, data = BaseballTimes2017, use = "pairwise.complete.obs")
```

```
## [1] 0.744907
```

```
# commented the others out
# gf_point(Time ~ Margin, data = BaseballTimes2017) %>%
#   gf_lm()
# fm <- lm(Time ~ Margin, data = BaseballTimes2017)
# msummary(fm)
# cor(Time ~ Margin, data = BaseballTimes2017, use = "pairwise.complete.obs")
#
# gf_point(Time ~ Pitchers, data = BaseballTimes2017) %>%
#   gf_lm()
# fm <- lm(Time ~ Pitchers, data = BaseballTimes2017)
# msummary(fm)
# cor(Time ~ Pitchers, data = BaseballTimes2017, use = "pairwise.complete.obs")
#
#
# gf_point(Time ~ Attendance, data = BaseballTimes2017) %>%
#   gf_lm()
# fm <- lm(Time ~ Attendance, data = BaseballTimes2017)
# msummary(fm)
# cor(Time ~ Attendance, data = BaseballTimes2017, use = "pairwise.complete.obs")

#For your solution, just show the "best" scatterplot!
```

part c: Fit most appropriate model, report equation, and interpret slope

SOLUTION: Time = 148.04 + 4.18 * Runs is the equation. The slope of 4.18 means for every additional unit increase in Runs, the Time increases by 4.18

```
fm <- lm(Time ~ Runs, data = BaseballTimes2017) #example model fit
msummary(fm)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   148.04      12.00   12.34  3.5e-08 ***
## Runs            4.18       1.08    3.87   0.0022 **
##
## Residual standard error: 15.3 on 12 degrees of freedom
## Multiple R-squared:  0.555,  Adjusted R-squared:  0.518
## F-statistic:   15 on 1 and 12 DF,  p-value: 0.00224
```

part d: Check conditions and comment

SOLUTION:

Yes, at first pass from a visual standpoint we can proceed. We will need to use the LINE pnuemonic to consider during the ASSESS phase.

Linearity - Assess via a scatterplot of X and Y.

We can see the linearity using the scatterplot above.

Independence - Read the research design/methods to see if it observations were independent (randomly selected)

We cannot say anything about this using R, so we need to read the experiment and documentation.

Normality - Assess the distribution of errors with histograms/qqplots of residuals errors should be centered at zero (ZERO MEAN), no skew or pattern (RANDOM)- necessary for inference.

The histogram is not relatively normal. There are a few high residuals > 20 that many need to be considered. All other residuals fall between +/- 20.

In the QQPlot, we can see that most points are on the line, there is no real shape and only a few are off at the ends, hence we are good there!

Equal (Constant) Variance - Assess with a residual vs. fitted plot. Looking for no pattern

Next, we can check the condition for constant variance of errors by looking at the residual vs fitted plots.

A few outliers are noted in red but nothing exerting a great deal of leverage (unduly influencing the overall relationship). There is no pattern of errors (looks like a cloud) and is mostly linear. IF the variance in Y was not equal across X we could see a fan shape indicating heteroscedastcity which would need to be resolved with a transformation.

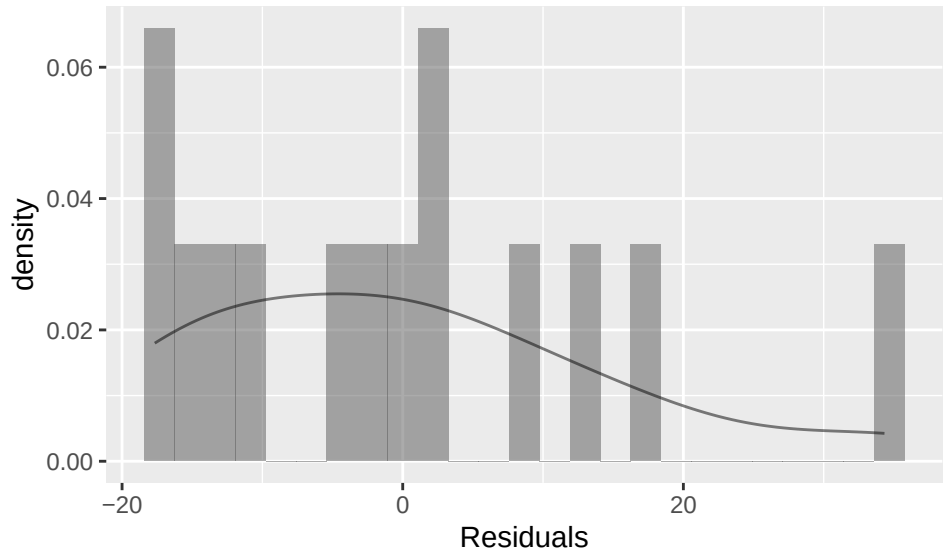
```
# Fit
fm <- lm(Time ~ Runs, data = BaseballTimes2017)
msummary(fm)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   148.04      12.00   12.34  3.5e-08 ***
## Runs            4.18       1.08    3.87   0.0022 **
##
## Residual standard error: 15.3 on 12 degrees of freedom
## Multiple R-squared:  0.555,  Adjusted R-squared:  0.518
## F-statistic:   15 on 1 and 12 DF,  p-value: 0.00224
```

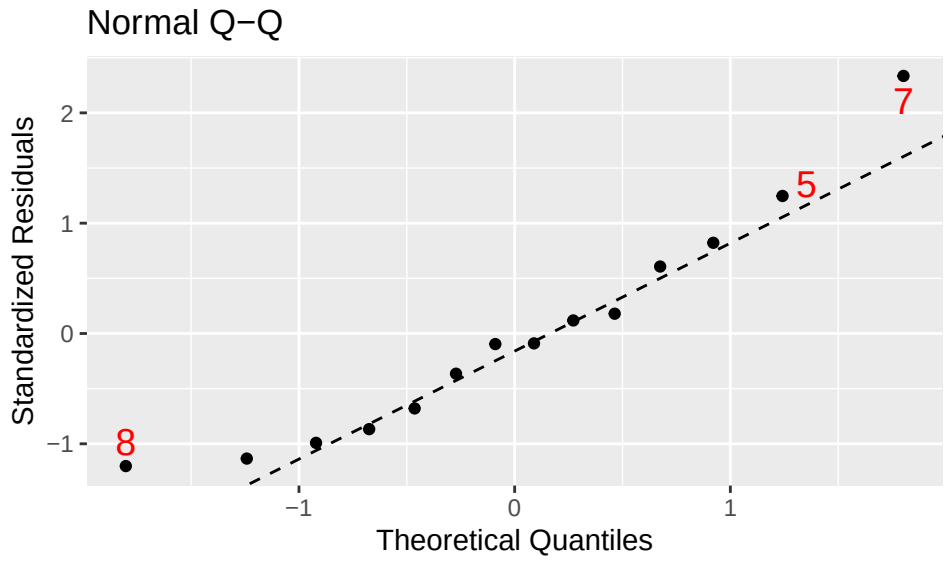
```
confint(fm, level = 0.95)
```

```
##                 2.5 %    97.5 %
## (Intercept) 121.90699 174.17812
## Runs          1.82565   6.53605
```

```
# Assess
# generates a histogram with fitted density curve
gf_dhistogram(~ residuals(fm), xlab = "Residuals") %>%
  gf_dens()
```




```
# easiest way to get QQplot for residuals
mplot(fm, which = 2)
```




```
# residual vs fitted plot
mplot(fm, which = 1)
```

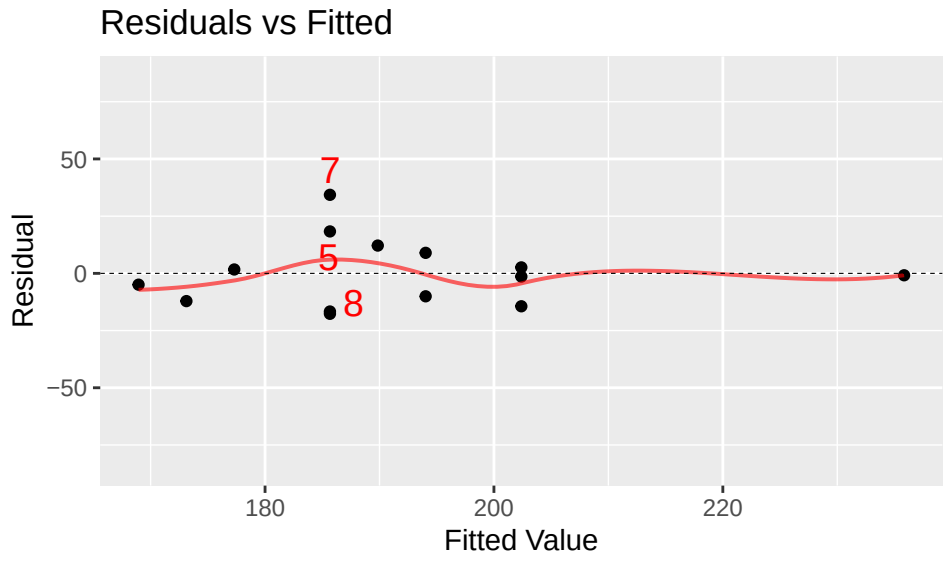
```
## `geom_smooth()` using formula 'y ~ x'

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 194.03

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 8.3617
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1.0205e-16
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at
## 194.03
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius
## 8.3617
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition
## number 1.0205e-16
```




```
mplot(fm, which = 1) #residuals vs fitted
```

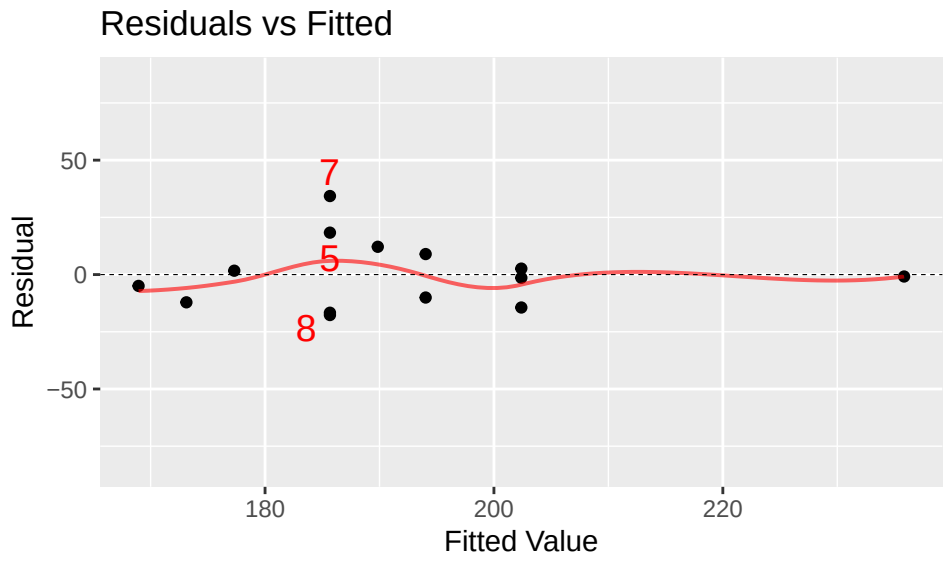
```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 194.03
```

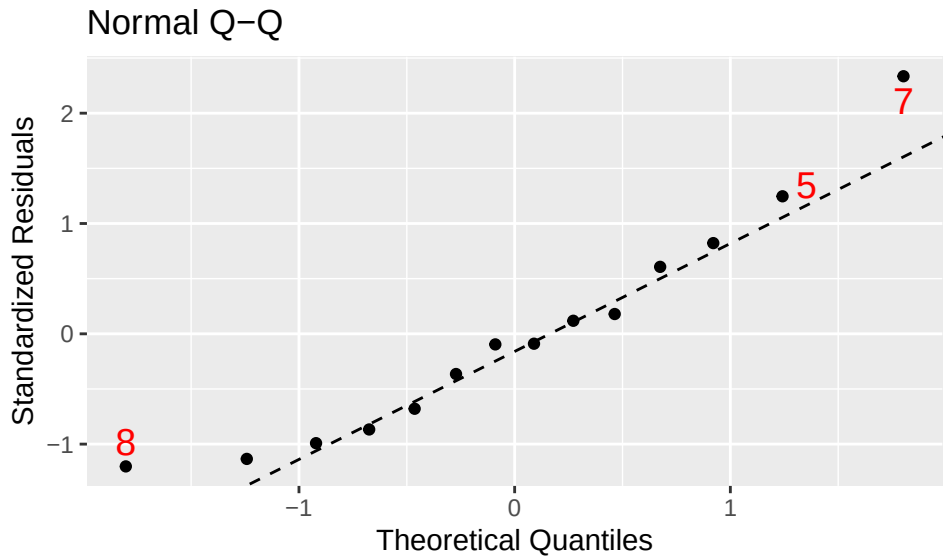
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 8.3617
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1.0205e-16
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at
## 194.03

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius
## 8.3617

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition
## number 1.0205e-16
```




```
mplot(fm, which = 2) #QQ plot of residuals
```

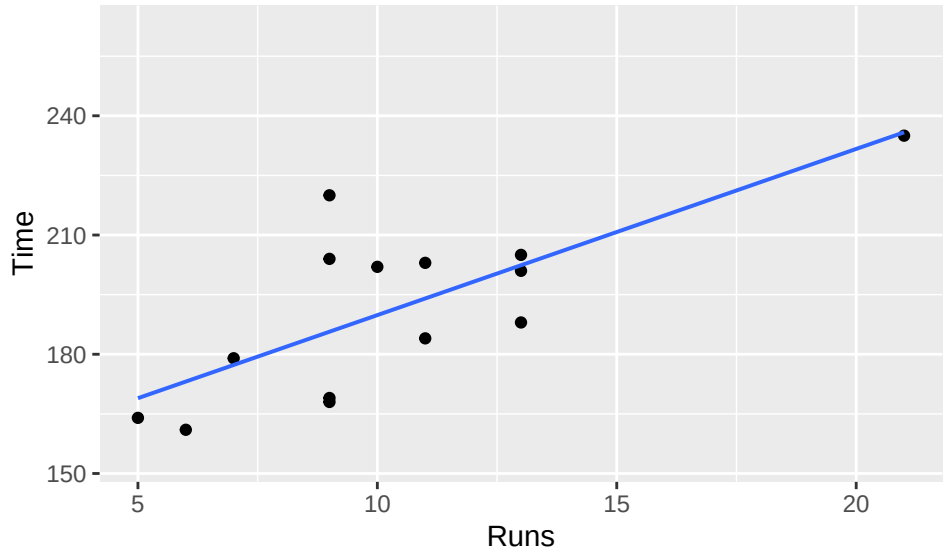


Exercise 1.46 - Slightly modified

part a: You should start writing your own short descriptions

SOLUTION: It is the point which is at the top right hand corner of the graph below.

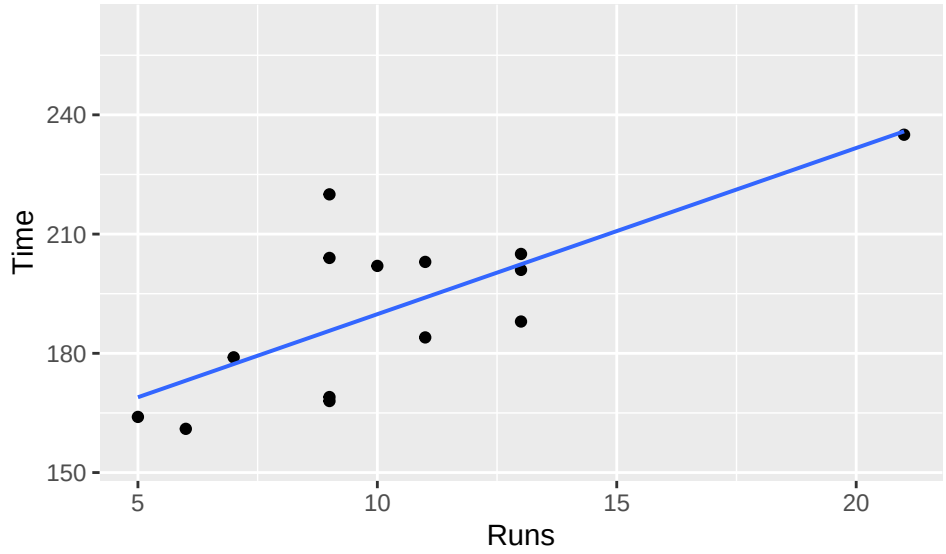
```
gf_point(Time ~ Runs, data = BaseballTimes2017) %>%
  gf_lm()
```



part INSERT: Fit the model using Runs as a predictor for Time in order to have it for comparison in part b. This is because folks may not chose Runs as the best predictor in the previous exercise (which is completely fine). If you did pick Runs, you already have the model fit, and can just reprint the summary to have here.

SOLUTION:

```
gf_point(Time ~ Runs, data = BaseballTimes2017) %>%
  gf_lm()
```




```
# Fit
fm <- lm(Time ~ Runs, data = BaseballTimes2017)
msummary(fm)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   148.04      12.00   12.34  3.5e-08 ***
## Runs            4.18       1.08    3.87   0.0022 **
##
## Residual standard error: 15.3 on 12 degrees of freedom
## Multiple R-squared:  0.555,  Adjusted R-squared:  0.518
## F-statistic:   15 on 1 and 12 DF,  p-value: 0.00224
```

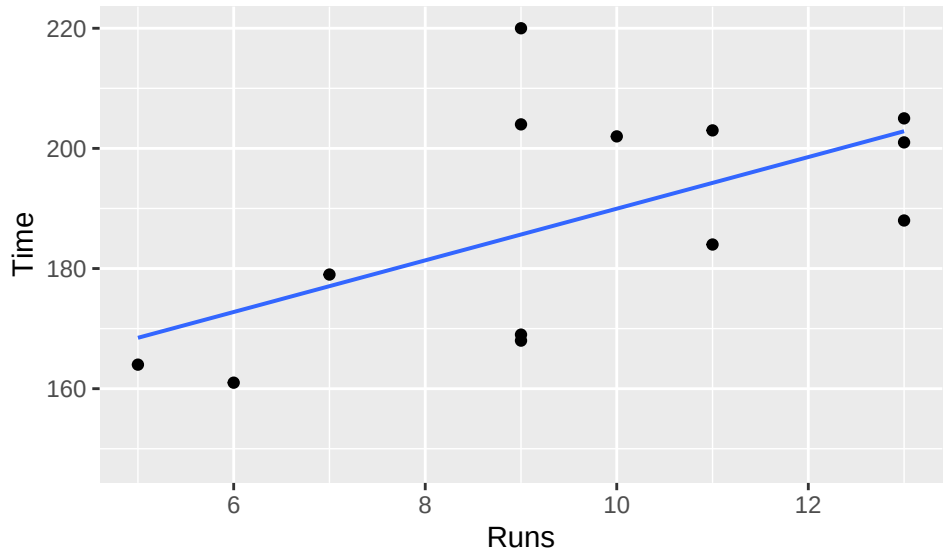
```
confint(fm, level = 0.95)
```

```
##                 2.5 %    97.5 %
## (Intercept) 121.90699 174.17812
## Runs          1.82565   6.53605
```

part b: change in equation and effect of this point

SOLUTION: The new equation is Time = 146.97 + 4.30 * Runs. The slope is increased when we remove that datapoint, which means the data point was leading to a decreased slope.

```
# create a new data set to repeat the analysis with using..... FILTER (or by removing the outlier in an
# copy over your model commands, change the name of the model, and change the data set name
# When it says repeat the analysis - just focus on parts c and d from 1.45, but with the new data set
BaseballTimes2017 <- BaseballTimes2017[!(BaseballTimes2017$Game=="CIN-MIL"),]
gf_point(Time ~ Runs, data = BaseballTimes2017) %>%
  gf_lm()
```




```
# Fit
fm <- lm(Time ~ Runs, data = BaseballTimes2017)
msummary(fm)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   146.97      17.67    8.32  4.5e-06 ***
## Runs            4.30       1.78    2.42    0.034 *
##
## Residual standard error: 16 on 11 degrees of freedom
## Multiple R-squared:  0.347,  Adjusted R-squared:  0.287
## F-statistic: 5.84 on 1 and 11 DF,  p-value: 0.0342
```

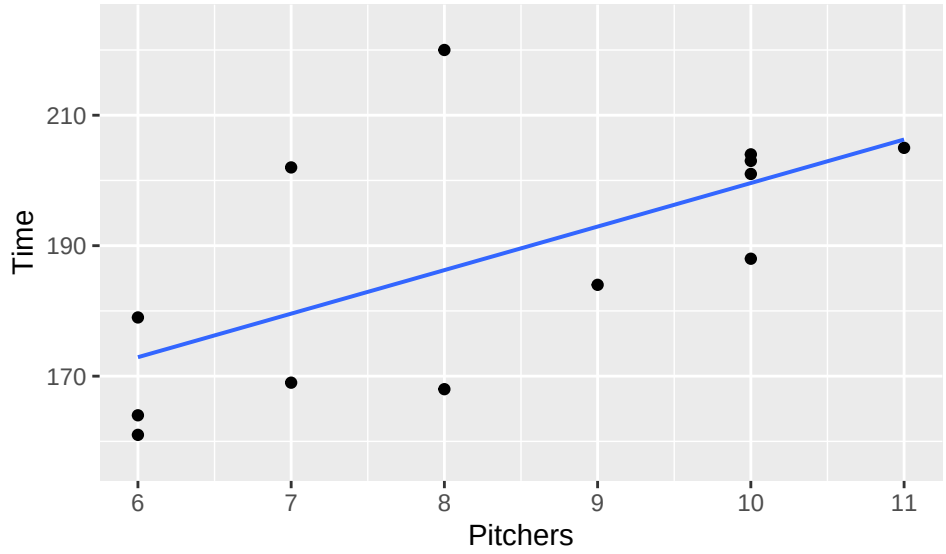
```
confint(fm, level = 0.95)
```

```
##                  2.5 %    97.5 %
## (Intercept) 108.074512 185.87046
## Runs          0.383379   8.21434
```

part c: Examine the scatterplots as instructed. Does it look like Runs is the best predictor for Time with the outlier removed?

SOLUTION: With the highest R-squared value of 40% and highest correlation with Time, now, Pitchers seems to be the best predictor for Time with the outlier removed.

```
gf_point(Time ~ Pitchers, data = BaseballTimes2017) %>%
  gf_lm()
```




```
fm <- lm(Time ~ Pitchers, data = BaseballTimes2017)
msummary(fm)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   132.86      20.93    6.35  5.5e-05 ***
## Pitchers        6.67       2.47    2.71     0.02 *
##
## Residual standard error: 15.4 on 11 degrees of freedom
## Multiple R-squared:   0.4,   Adjusted R-squared:  0.345
## F-statistic: 7.33 on 1 and 11 DF,  p-value: 0.0204
```

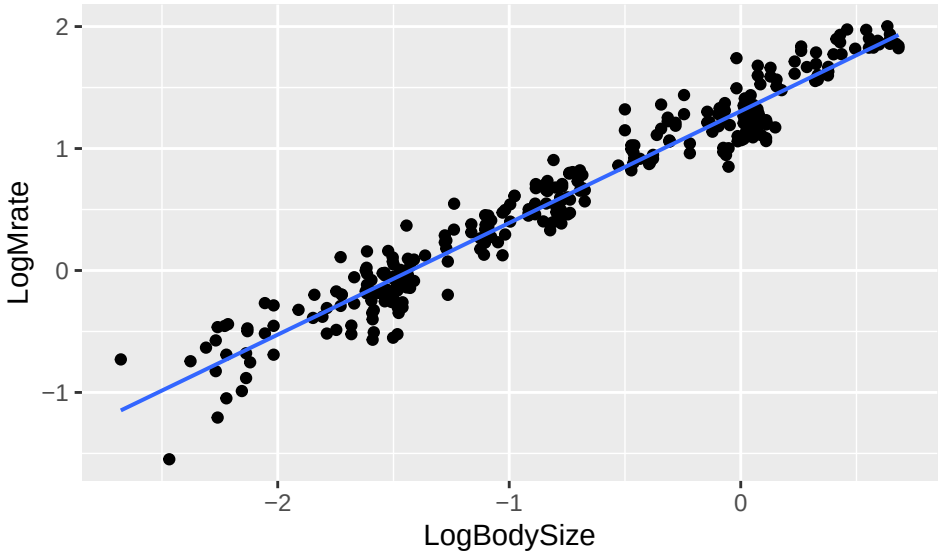
```
data(MetabolicRate)
```

Exercise 1.48

part a:

SOLUTION: we should use LogMrate as the response variable and LogBodySize as the predictor variable for the model because of their highest R-squared values and best fit among other possible combinations. We can see that the equation that this model gives us is as follows: LogMrate = 1.3066 + 0.9164 * LogBodySize.

```
#example plot
# gf_point(Mrate ~ BodySize, data = MetabolicRate) %>%
#   gf_lm()
# fm <- lm(Mrate ~ BodySize, data = MetabolicRate)
# msummary(fm)

gf_point(LogMrate ~ LogBodySize, data = MetabolicRate) %>%
  gf_lm()
```




```
fm <- lm(LogMrate ~ LogBodySize, data = MetabolicRate)
msummary(fm)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.3066     0.0136    96.3   <2e-16 ***
## LogBodySize   0.9164     0.0124    74.2   <2e-16 ***
##
## Residual standard error: 0.175 on 303 degrees of freedom
## Multiple R-squared:  0.948,  Adjusted R-squared:  0.948
## F-statistic: 5.51e+03 on 1 and 303 DF,  p-value: <2e-16
```

```
# Note the logged variables are already in the data set to try out!
# log in this context is log base 10, contrary to our normal natural log
```

Again, just show the plot you settled on as the one with the most appropriate linear relationship.

part b:

SOLUTION: LogMrate = 1.3066 + 0.9164 * LogBodySize is our equation and since BodySize is 1, this implies that the predicted Mrate value is 3.69341.

```
#fit the model from part a
fm3 <- lm(LogMrate ~ LogBodySize, data = MetabolicRate) #example code, change variables to what you lik
msummary(fm3)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.3066     0.0136    96.3   <2e-16 ***
## LogBodySize   0.9164     0.0124    74.2   <2e-16 ***
##
## Residual standard error: 0.175 on 303 degrees of freedom
## Multiple R-squared:  0.948,  Adjusted R-squared:  0.948
## F-statistic: 5.51e+03 on 1 and 303 DF,  p-value: <2e-16
```

```
fittedpred <- makeFun(fm3) #create function for prediction
predictedLogMrate <- fittedpred(LogBodySize = log(1, base = exp(1))) #example use of function
#note that in R, log means natural log, to get log base 10, use log10
predictedMrate <- exp(predictedLogMrate)

predictedMrate
```

```
##       1
## 3.69341
```

```
data(Goldenrod)
```

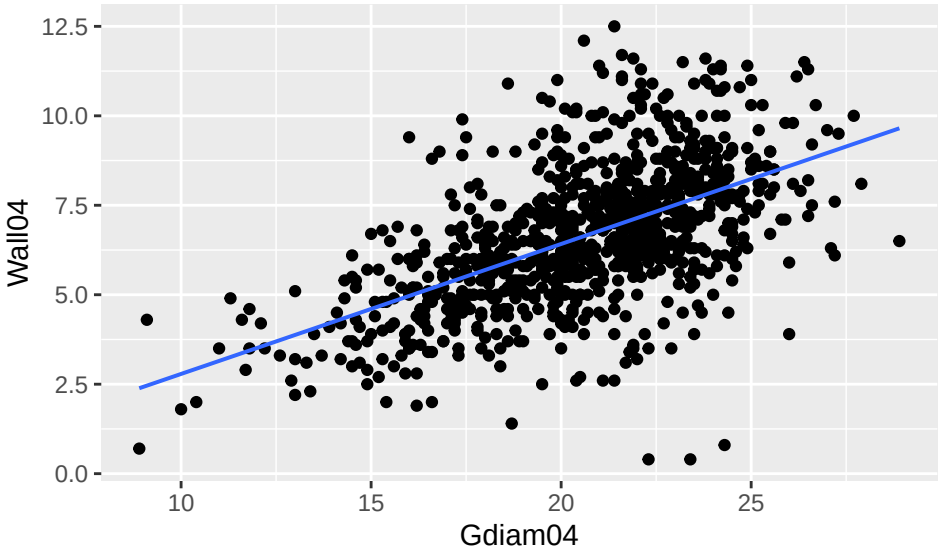
Exercise 2.30

part a:

SOLUTION: There is a positive linear correlation between Wall04 and Gdiam04 with correlation coefficient of about 0.57 and our results are justified by the cor.test. Also, since in the linear regression analysis, we can see that the t-test on coefficent of Gdiam04 is close to 0, we can say that there is a significant linear relationship.

```
# fit model and get summary
# then check conditions (don't re-express, just comment if there are issues)
# before running the appropriate test

gf_point(Wall04 ~ Gdiam04, data = Goldenrod) %>% gf_lm()
```

```
## Warning: Removed 113 rows containing non-finite values (stat_lm).

## Warning: Removed 113 rows containing missing values (geom_point).
```



```r
# cor(Wall04 ~ Gdiam04, data = Goldenrod, use = "pairwise.complete.obs") #necessary because not all are

fm <- lm(Wall04 ~ Gdiam04, data = Goldenrod)
msummary(fm)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.8450     0.3577   -2.36    0.018 *
## Gdiam04       0.3632     0.0172   21.12   <2e-16 ***
##
## Residual standard error: 1.61 on 940 degrees of freedom
##   (113 observations deleted due to missingness)
## Multiple R-squared:  0.322,  Adjusted R-squared:  0.321
## F-statistic:  446 on 1 and 940 DF,  p-value: <2e-16
```

```r
cor.test(Wall04 ~ Gdiam04, data = Goldenrod)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Wall04 and Gdiam04
## t = 21.12, df = 940, p-value <2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.522410 0.609152
## sample estimates:
##      cor
## 0.567353
```

part b:

SOLUTION: As we can see from the output of code chunk in part (a), the estimated slope of the same is 0.3632 with a standard error of 0.0172

```r
# Fit
fm <- lm(Wall04 ~ Gdiam04, data = Goldenrod)
msummary(fm)
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.8450     0.3577   -2.36    0.018 *
## Gdiam04       0.3632     0.0172   21.12   <2e-16 ***
##
## Residual standard error: 1.61 on 940 degrees of freedom
##   (113 observations deleted due to missingness)
## Multiple R-squared:  0.322,  Adjusted R-squared:  0.321
## F-statistic:  446 on 1 and 940 DF,  p-value: <2e-16
```

```r
confint(fm, level = 0.95)
```

```
##                 2.5 %    97.5 %
## (Intercept) -1.546894 -0.143116
## Gdiam04      0.329431  0.396912
```

part c:

SOLUTION: The size of the typical standard residual error for this simple linear regression model is 1.61 as we can see in the value of the Residual standard error as part of the output of the code chunk in part (a)

part d:

SOLUTION: No, because as we can see in the value of R-squared, this model only accounts for around 32% of the variability.

part e:

SOLUTION: We are 95% confident that Wall04 is between 6.31375 and 6.5231.

```
#use the predict function
new_data <- data.frame(Gdiam04 = c(20))
predict(fm, new_data, int = "confidence", level = 0.95)
```

```
##       fit     lwr    upr
## 1 6.41842 6.31375 6.5231
```

part f:

SOLUTION: Since the correlation coefficient is greater than 0.5 we can say that there is significant relationship between the variables at play. Also, because the p-value is close to 0 we can say that there is a significant linear relationship.

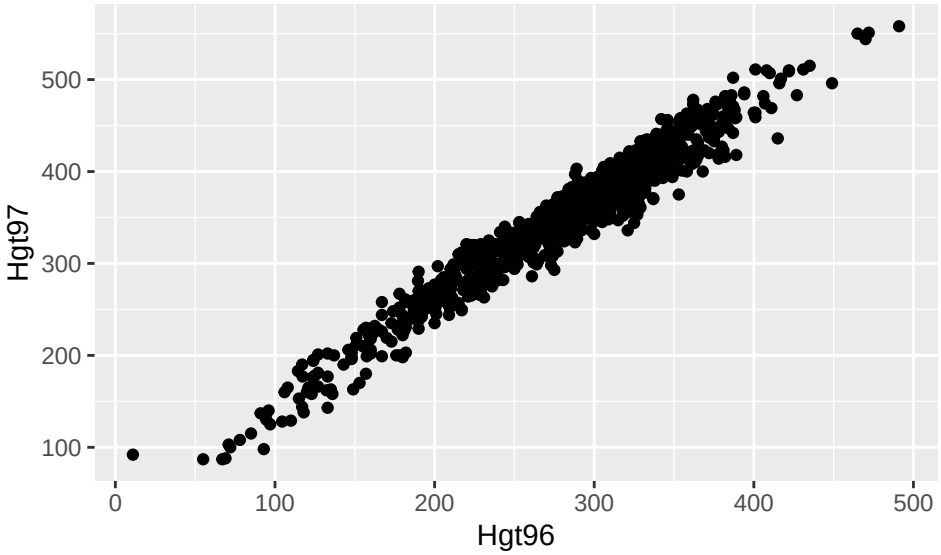
```
cor.test(Wall04 ~ Gdiam04, data=Goldenrod)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Wall04 and Gdiam04
## t = 21.12, df = 940, p-value <2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.522410 0.609152
## sample estimates:
##      cor
## 0.567353
```

Additional (based on 2.33 in the textbook with info from 2.32) The model info is provided for you. The model info is provided for you.

```
data(Pines)
gf_point(Hgt97 ~ Hgt96, data=Pines)
```

```
## Warning: Removed 146 rows containing missing values (geom_point).
```

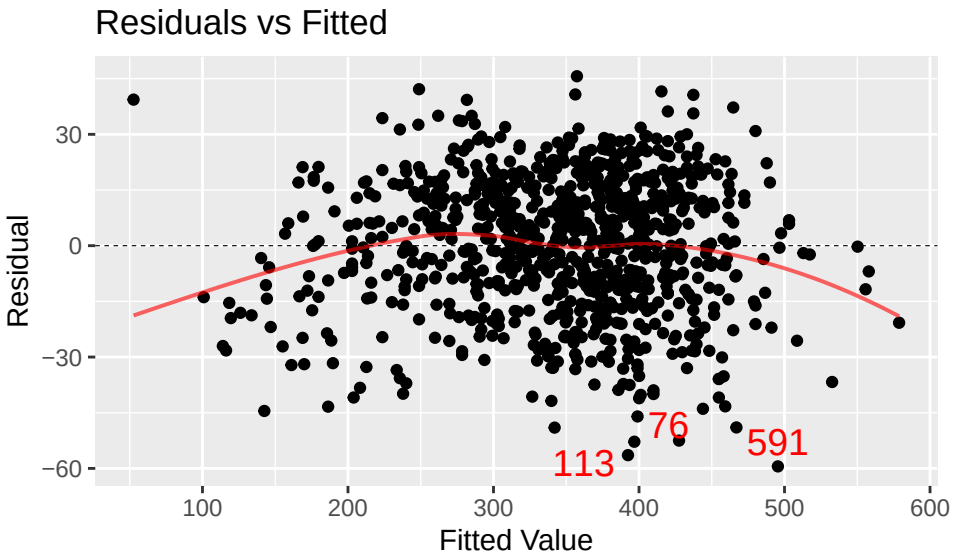



```
fm5 <- lm(Hgt97 ~Hgt96, data=Pines)
msummary(fm5)
```

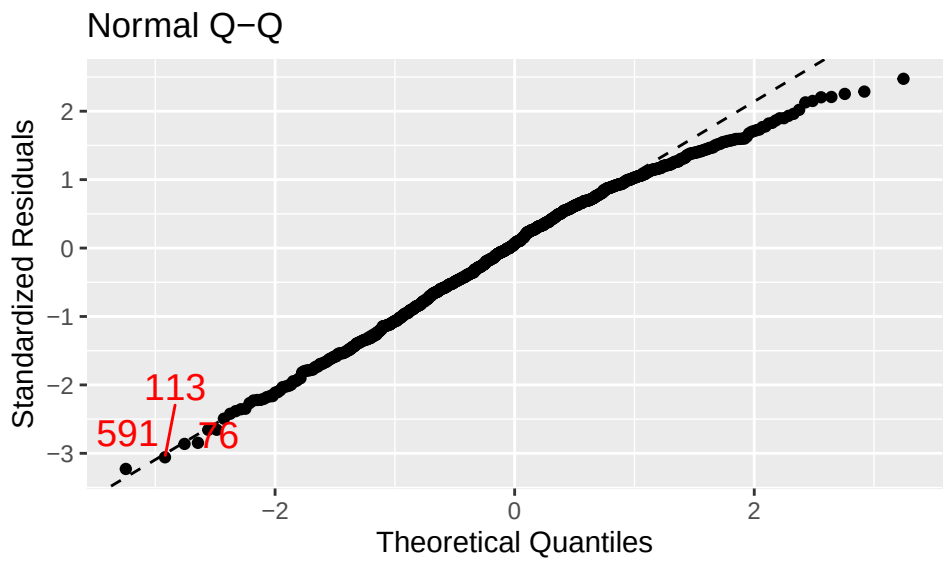
```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 40.59078    2.52448    16.1   <2e-16 ***
## Hgt96        1.09606    0.00873   125.5   <2e-16 ***
##
## Residual standard error: 18.5 on 852 degrees of freedom
##   (146 observations deleted due to missingness)
## Multiple R-squared:  0.949,  Adjusted R-squared:  0.949
## F-statistic: 1.57e+04 on 1 and 852 DF,  p-value: <2e-16
```

```
mplot(fm5, which=1)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
mplot(fm5, which=2)
```



This problem uses the regression using height in 1996 to predict height in 1997 that used in 2.32 for the white pine trees. Selected and slightly modified parts of 2.33 are outlined for you to complete below. You can assume that all conditions are satisfied for the regression.

part a: Find and interpret a 95% CI for the slope coefficient of Hgt96 in this regression model.

SOLUTION: The output of the code chunk below shows us the 95% CI for the model at hand. We can say that we are 95% confident that the slope coefficient of Hgt96 in the linear regression equation will lie between 1.07892 and 1.1132.

```
confint(fm5, level = 0.95)
```

```
##                2.5 %  97.5 %
## (Intercept) 35.63585 45.5457
## Hgt96        1.07892  1.1132
```

part b: Is the value of 1 included in your CI for the slope coefficient? What does this tell you about whether or not the trees are growing?

SOLUTION: The value of 1 is not included in the CI of my slope coefficient, this tells me that the trees are growing and Hgt97 would be greater than Hgt96. We are able to reject the null hypothesis since the value of 1 is not in the 95% confidence interval.

part c: Find and interpret a 99% CI for the mean height of trees in 1997 that were 200cm tall in 1996.

SOLUTION: 99% confidence interval for mean height of trees in 1997 that were 200 cm tall in 1996 is (257.373, 262.232), which means that we are 99% confident that on average a 200cm tall tree in 1996 will grow into a tree of height between 257.373 and 262.232 in year 1997.

```r
# use predict function
newData <- data.frame(Hgt96 = 200)
predict(fm5, newData, interval = 'confidence', level = 0.99)
```

```
##       fit     lwr     upr
## 1 259.803 257.373 262.232
```

part d: How would a 99% prediction interval for the height of trees in 1997 that were 200 cm tall in 1996 compare to the CI you obtained in part c.?

SOLUTION: 99% prediction interval for height of a tree in 1997 that was 200 cm tall in 1996 is (212.071, 307.535), which means that we are 99% confident that a 200cm tall tree in 1996 will grow into a tree of height between 212.071 and 307.535 in year 1997.

```r
# use predict function
newData <- data.frame(Hgt96 = 200)
predict(fm5, newData, interval = 'prediction', level = 0.99)
```

```
##       fit     lwr     upr
## 1 259.803 212.071 307.535
```

part e: If the confidence level were changed to 95% in part c., would the new interval be wider or narrower compared to your CI in part c.? Explain, without recalculating the interval.

SOLUTION: If the confidence level was changed from 99% to 95% for instance, it would make the interval narrower since it makes sense for us to miss the actual value 5% of the times instead of 1% of the times if we have a narrower interval in our prediction.