HW 2

STAT 5211

Emma Johnson

January 28, 2025

4

a)

The training data would have a lower RSS for the cubic model than the linear model because it would fit the data points better (i.e. more **flexible**), thus, minimizing the errors between the observed and fitted values when the model can curve to fit the training set.

b)

On the other hand, the RSS would be greater for the cubic model than the linear model when fitting the testing set, as it would display **overfitting** from the training set in which the model was created under. The linear model would have a lower RSS here, because if the model was truly linear, you wouldn't risk overfitting the data when transitioning from the training to the testing set.

c)

The cubic model would, still, fit the training data tighter than the linear model when just assessing the training set. As a rule of thumb, the higher the degree of the model, the more flexible the model will be. In turn, the observed data will fit within the expected values of the model better and better as you increase the degree. The issue, however, is that this decreasing generalizeability and extrapolation. Typically, a model with a low degree has a greater interpretation than one with a higher degree (unless, of course, the true distribution follows that higher degree, which is seemingly unlikely).

d)

There is not enough information to verify which model would display a lower RSS for the training data. This is because, without assessing key assumptions to a regression model (e.g. pattern of residuals), we cannot determine the nature of the distribution of data to draw conclusions/prediction value. We do not know how far it is from linear, and similarly, how far it is from cubic. The answer to part (c) does not influence this one.

8: Auto Data

a)

```r
attach(Auto)
model <- lm(mpg ~ horsepower)
summary(model)
```

```
##
```

```
## Call:
## lm(formula = mpg ~ horsepower)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

i)

There is a relationship between the predictor and the response because the p-value associated with the coefficient of the predictor variable (i.e. horsepower) is much extremely small (i.e. p = 2e-16). This signifies a significant relationship between mpg and horsepower.

ii)

Continuing on from the point made in part (i), this low p-value signifies a significant relationship between mpg and horsepower. In other words, a strong relationship between the two variables. Further, the R^2 term suggests that 60.59% of the variability in mpg is explained by horsepower under this model.

iii)

Since the estimated value of the coefficient attached to horsepower is negative, this signifies a negative relationship between horsepower and mpg.

iv)

```r
# Predicting the mpg based on a 98 horsepower value:
predict(model, newdata = data.frame(horsepower = 98))
```

```
##        1
## 24.46708
```

```r
# 95% Confidence Interval:
predict(model, data.frame(horsepower = 98),
    interval = "confidence")
```

```
##        fit      lwr      upr
## 1 24.46708 23.97308 24.96108
```

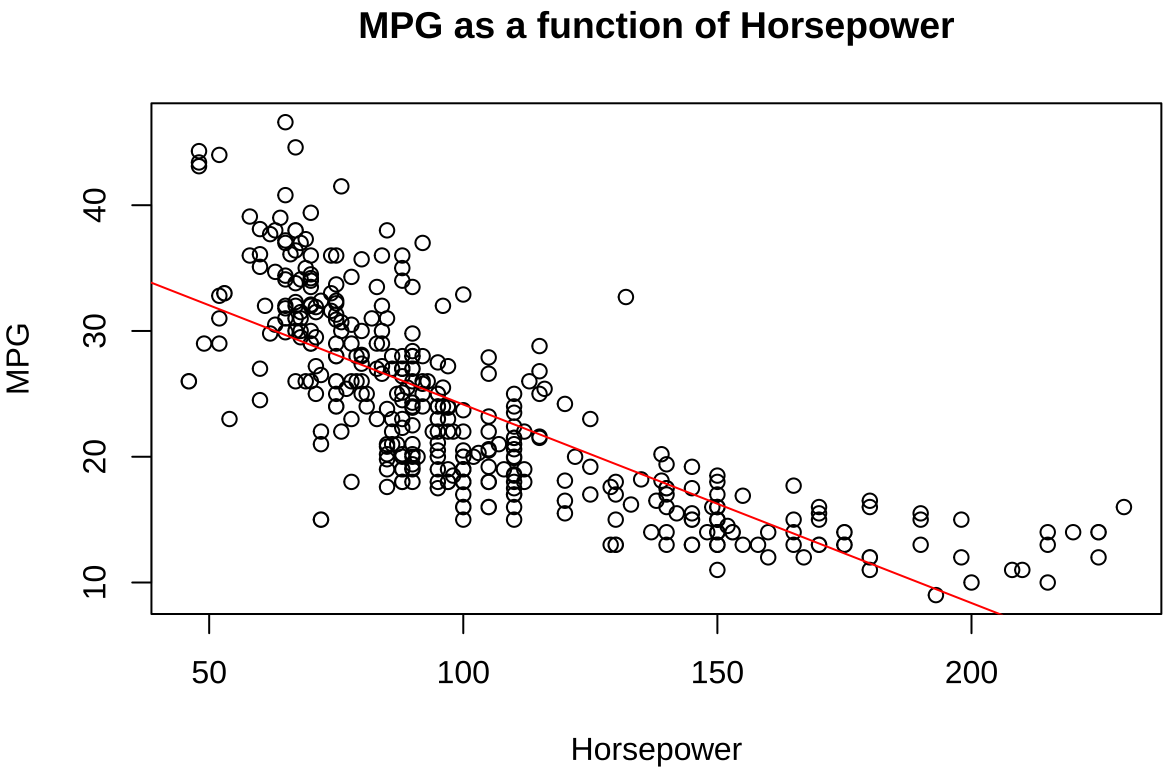
```r
# CI.95 = [23.97308, 24.96108]

# 95% Prediction Interval:
predict(model, data.frame(horsepower = 98),
    interval = "prediction")
```

```
##        fit     lwr      upr
## 1 24.46708 14.8094 34.12476
```

```
# PI.95 = [14.8094, 34.12476]
```

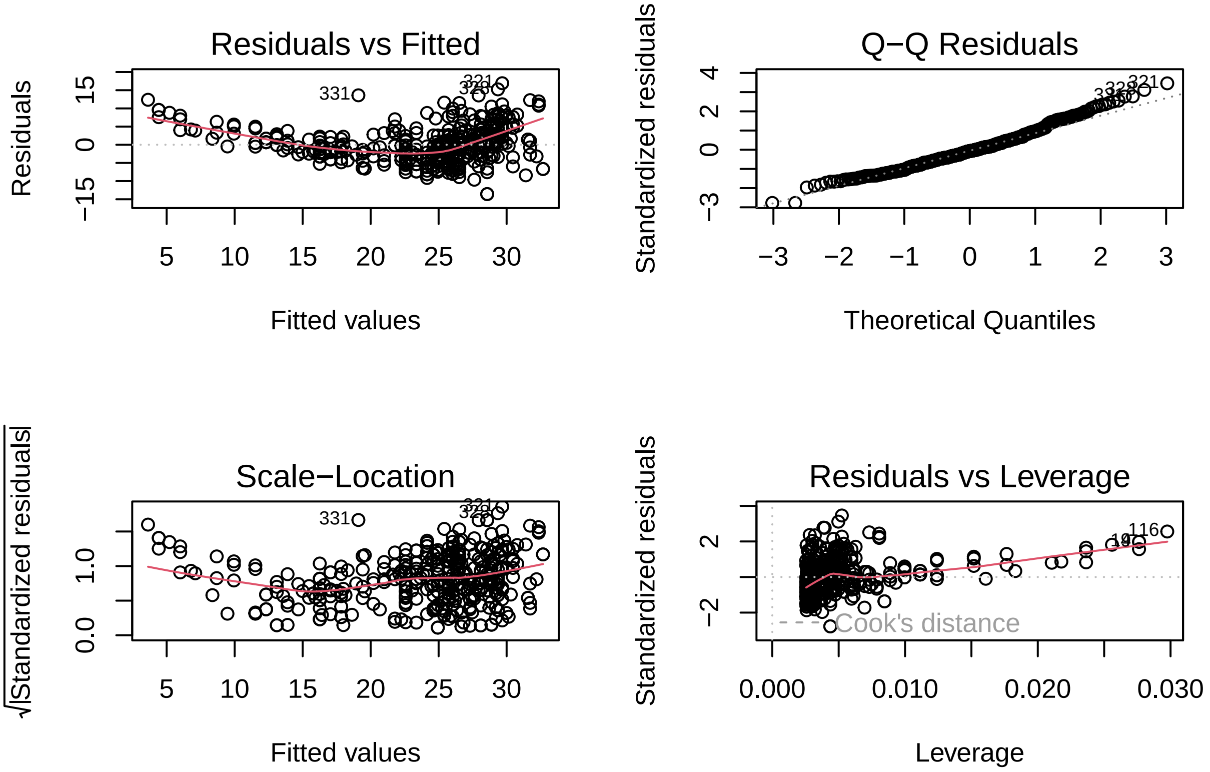
b)

```
plot(horsepower, mpg,
     main = "MPG as a function of Horsepower",
     xlab = "Horsepower",
     ylab = "MPG")
abline(model, col = "red")
```



c)

```
par(mfrow = c(2, 2))
plot(model)
```



The residuals seem to follow a nonlinear pattern and do not appear with a constant variance under the linear model. This is a cause for concern and suggests that there should be a transformation of variables in order to create a more reliable model fit.

13

```
set.seed(1)
```

a)

```
x <- rnorm(n = 100, mean = 0, sd = 1)
```

b)

```
eps <- rnorm(n = 100, mean = 0, sd = sqrt(0.25))
```

c)

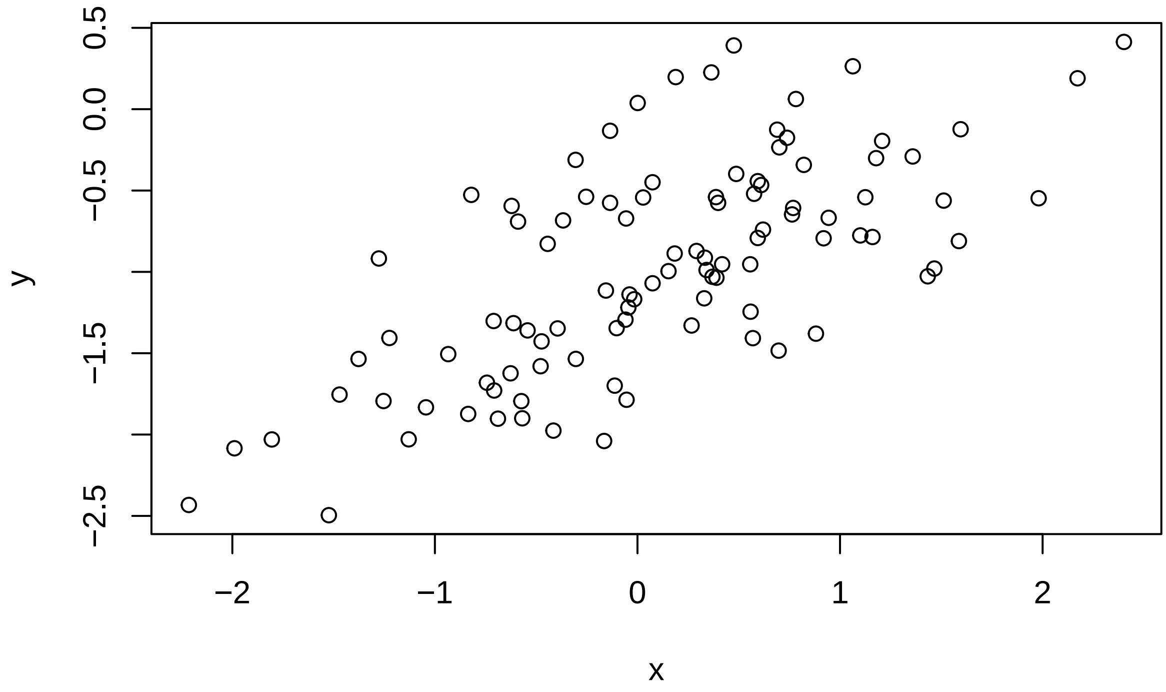
```
y <- -1 + 0.5*x + eps
length(y) # verifying that the length of y = 100, matching x and eps
```

```
## [1] 100
```

Assuming that *eps* represents the vector of error terms, then β_0 is the intercept -1 and β_1 is 0.5.

d)

```
plot(x, y)
```



There appears to be a positive linear relationship between x and y according to the scatterplot.

e)

```
model <- lm(y ~ x)
summary(model)
```

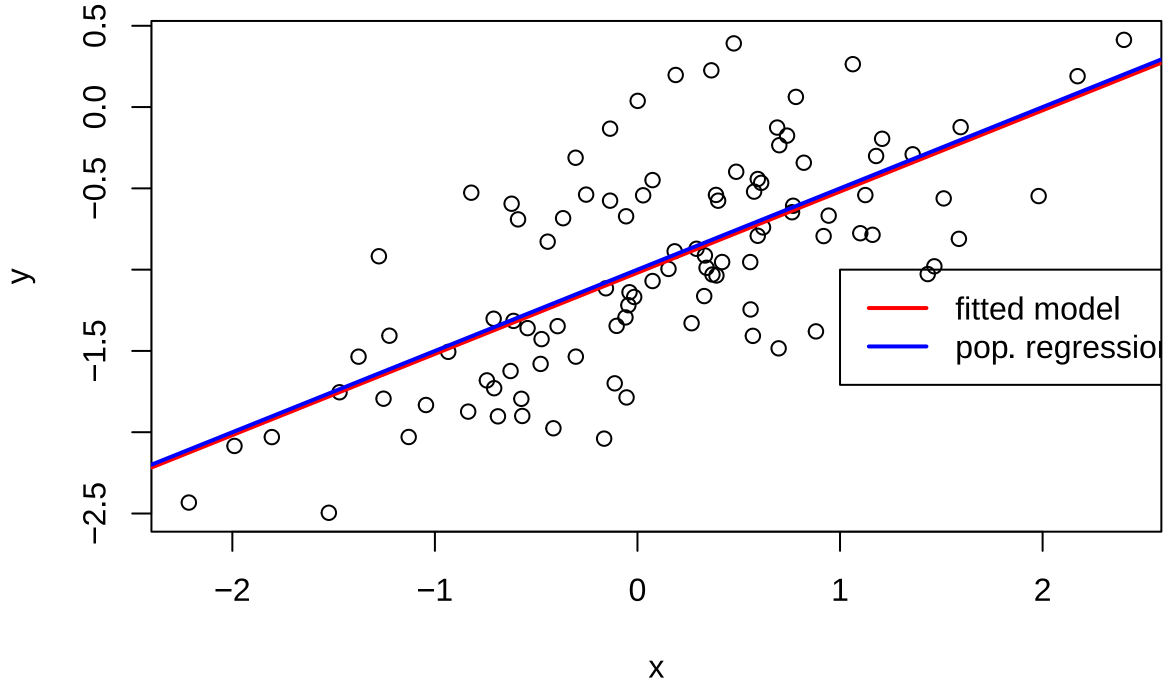
```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93842 -0.30688 -0.06975  0.26970  1.17309
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.01885    0.04849 -21.010  < 2e-16 ***
## x            0.49947    0.05386   9.273 4.58e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.4814 on 98 degrees of freedom
## Multiple R-squared:  0.4674, Adjusted R-squared:  0.4619
## F-statistic: 85.99 on 1 and 98 DF,  p-value: 4.583e-15
```

Under this model, the R^2 value asserts that 46.74% of the total variation in y is accounted for by x, which suggests a somewhat weak relationship. The RSE of 0.4814 describes the variability among the errors/residuals, which appears rather high, given the units of the model. Further, $\hat{\beta}_0 = -1.01885$ and $\hat{\beta}_1 = 0.49947$, both of which are slightly smaller compared to their counterparts found in part (c).

f)

```
plot(x, y)
abline(model, lwd = 2, col = "red")
abline(-1, 0.5, lwd = 2, col = "blue")
legend(-1, legend = c("fitted model", "pop. regression"), col=c("red", "blue"), lwd=2)
```



Note that the fitted regression model nearly overlaps the population regression model.

g)

```
model2 <- lm(y ~ x + I(x^2))
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2))
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98252 -0.31270 -0.06441  0.29014  1.13500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.97164    0.05883 -16.517  < 2e-16 ***
## x            0.50858    0.05399   9.420  2.4e-15 ***
## I(x^2)      -0.05946    0.04238  -1.403    0.164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.479 on 97 degrees of freedom
## Multiple R-squared:  0.4779, Adjusted R-squared:  0.4672
## F-statistic:  44.4 on 2 and 97 DF,  p-value: 2.038e-14
```

Although the R^2 value slightly increased to 47.79%, the p-value attached to the squared term (i.e. 0.164) yields insignificant results. This model might fit the data slightly better, but it is unnecessary to include the higher order term. Therefore, drop the quadratic term and conclude the linear model.

h)

```r
# Moving the original eps' variance to the right one decimal place to decrease sd
eps2 <- rnorm(n = 100, mean = 0, sd = sqrt(0.025))
y2 <- -1 + 0.5*x + eps2
model3 <- lm(y2 ~ x)
summary(model3)
```

```
##
## Call:
## lm(formula = y2 ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46076 -0.07626 -0.00717  0.10265  0.41767
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.99567    0.01656  -60.14   <2e-16 ***
## x            0.50335    0.01839   27.37   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1643 on 98 degrees of freedom
## Multiple R-squared:  0.8843, Adjusted R-squared:  0.8832
## F-statistic: 749.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

By decreasing the variance in *eps* by assigning a smaller value (i.e. 0.025 this time) and then applying this change to the response, y, the R^2 value increases nearly two-fold to 88.43%. The RSE has decreased down to 0.1643 , reflecting the **decrease in noise**. The coefficients have converged closer to the population values, as well. The overall point is that that this model fits better, as errors are minimized.

i)

```r
# Multiplying the original eps' variance by 2 to increase sd
eps3 <- rnorm(n = 100, mean = 0, sd = sqrt(0.5))
y3 <- -1 + 0.5*x + eps3
model4 <- lm(y3 ~ x)
summary(model4)
```

```
##
## Call:
## lm(formula = y3 ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7793 -0.3856 -0.0267  0.4758  1.3286
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.95922    0.07091 -13.527  < 2e-16 ***
## x            0.46062    0.07876   5.848 6.55e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7039 on 98 degrees of freedom
## Multiple R-squared:  0.2587, Adjusted R-squared:  0.2512
## F-statistic:  34.2 on 1 and 98 DF,  p-value: 6.553e-08
```

By increasing the variance in *eps* by assigning twice the original value (i.e. 0.5 this time) and then applying this change to the response, y, the R^2 value decreases drastically to 25.87%. The RSE has increased to 0.7039, reflecting the **increase in noise**. The coefficients have tended away from the population values, as well. Overall, this model displays an ill fit of the data as a result of larger error terms.

j)

```r
# CI for the original data:
confint(model)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.1150804 -0.9226122
## x            0.3925794  0.6063602
```

```r
# CI for the less noisy data:
confint(model3)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0285258 -0.9628195
## x            0.4668557  0.5398379
```

```r
# CI for the noisier data:
confint(model4)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0999424 -0.8185064
## x            0.3043238  0.6169242
```

Clearly, when comparing the three confidence intervals, we see a pattern related to the errors. The confidence interval for the less noisy data is narrower than the original, and the confidence interval for the noisy data is wider than the original. This is directly related to the standard error within the construction of the confidence intervals. In sum, as error variance increases, the associated confidence intervals widen (and vice versa). Notably, all 3 confidence intervals seem to be centered around the population values of β_0 and β_1, as predicted.