HW2

Solutions

Details

Due Date

This assignment is due at 9:30 AM on Friday, Sept 20.

Grading

20% of your grade on this assignment is for completion. A quick pass will be made to ensure that you've made a reasonable attempt at all problems.

Some of the problems will be graded more carefully for correctness. In grading these problems, an emphasis will be placed on full explanations of your thought process. You usually won't need to write more than a few sentences for any given problem, but you should write complete sentences! Understanding and explaining the reasons behind your decisions is more important than making the "correct" decision.

Solutions to all problems will be provided.

Collaboration

You are allowed to work with others on this assignment, but you must complete and submit your own write up. You should not copy large blocks of code or written text from another student.

Sources

You may refer to class notes, our textbook, Wikipedia, etc.. All sources you refer to must be cited in the space I have provided at the end of this problem set.

In particular, you may find the following resources to be valuable:

- Courses assigned on DataCamp
- Example R code from class
- Cheat sheets and resources linked from [http://www.evanlray.com/stat340_f2019/resources.html]

Load Packages

The following R code loads packages needed in this assignment.

```r
library(readr)
library(dplyr)
library(ggplot2)
```

Conceptual Problems

If you prefer, you can write your answers to all conceptual problems by hand and turn in a physical copy. It's also fine if you want to write your answers up in LaTeX and push the pdf to GitHub.

Problem 1: Column space, fitted values, hat matrix, and projections.

Please complete the example problem we started in class on Friday, Sept. 13 (linked to from the schedule page). This will be graded for completion only, but it's important that you understand what's happening. I'm happy to talk through it with you.

Problem 2: One-way ANOVA

Suppose I conduct an experiment where a total of n subjects are randomly assigned to one of two groups (control and treatment). There are n_1 subjects in the control group and n_2 subjects in the treatment group (so $n_1 + n_2 = n$). For each subject i, I record a response variable y_i.

(a) Write down an appropriate model for these data for a single observation indexed by i. As part of your answer, please define x_i and specify what follows a normal distribution.

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \text{ where } x_i = \begin{cases} 1 \text{ if observation } i \text{ is in the treatment group} \\ 0 \text{ if observation } i \text{ is in the control group} \end{cases} \quad \text{and } \varepsilon_i \sim \text{Normal}(0, \sigma^2).$$

(b) Write down the model for these data in matrix form. As part of your answer, please define X and specify what follows a normal distribution. You may assume the first n_1 subjects were assigned to the control group and the remaining n_2 subjects were assigned to the treatment group. Since you don't know exact values for n_1 and n_2, you'll probably have to have some dots ($\vdots$) in your specification of X to indicate some rows that you aren't explicitly writing down.

$Y = X\beta + \varepsilon$, where

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \qquad X = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix} \text{ and } \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

In the design matrix X, the first n_1 rows have 0's in the second column and the remaining n_2 rows have 1's in the second column.

(c) By working through matrix calculations, find expressions for the coefficient estimates $\hat{\beta}$ in terms of $y_1, \dots, y_n$, n_1, and n_2.

$$
\begin{aligned}
\hat{\beta} &= (X'X)^{-1}X'y \\
&= \left(\begin{bmatrix} 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ \vdots & \vdots \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \\
&= \begin{bmatrix} n & n_2 \\ n_2 & n_2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=n_1+1}^n y_i \end{bmatrix} \\
&= \frac{1}{nn_2 - n_2 n_2} \begin{bmatrix} n_2 & -n_2 \\ -n_2 & n \end{bmatrix} \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=n_1+1}^n y_i \end{bmatrix} \\
&= \frac{1}{n_2(n - n_2)} \begin{bmatrix} n_2 \sum_{i=1}^n y_i - n_2 \sum_{i=n_1+1}^n y_i \\ -n_2 \sum_{i=1}^n y_i + n \sum_{i=n_1+1}^n y_i \end{bmatrix} \\
&= \frac{1}{n_1 n_2} \begin{bmatrix} n_2 \sum_{i=1}^{n_1} y_i \\ -n_2 \sum_{i=1}^n y_i + (n_1 + n_2) \sum_{i=n_1+1}^n y_i \end{bmatrix} \\
&= \frac{1}{n_1 n_2} \begin{bmatrix} n_2 \sum_{i=1}^{n_1} y_i \\ -n_2 \sum_{i=1}^n y_i + (n_1 + n_2) \sum_{i=n_1+1}^n y_i \end{bmatrix} \\
&= \frac{1}{n_1 n_2} \begin{bmatrix} n_2 \sum_{i=1}^{n_1} y_i \\ -n_2 \sum_{i=1}^{n_1} y_i + n_1 \sum_{i=n_1+1}^n y_i \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{n_1} \sum_{i=1}^{n_1} y_i \\ -\frac{1}{n_1} \sum_{i=1}^{n_1} y_i + \frac{1}{n_2} \sum_{i=n_1+1}^n y_i \end{bmatrix} \\
&= \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 - \bar{y}_1 \end{bmatrix}
\end{aligned}
$$

where $\bar{y}_1$ is the sample mean of observed responses for the control group and $\bar{y}_2$ is the sample mean of observed responses from the treatment group.

Applied Problems

Problem 3: Two-way ANOVA

I made this example up. Suppose a study was done to evaluate two possible factors on heart health: a new medication, and an exercise program. 15 subjects were recruited for the study, and randomly assigned to one of three groups (with 5 subjects in each group):

- control: participants did not receive the medication or participate in the exercise program.
- medicine: participants took the medication, but did not exercise
- both: participants took the medication and also exercised.

The following R code creates a manufactured data set with made up numbers. The response `health` represents a measure of heart health where a larger number is better. There is a column called `medication` which has the value `yes` if the participant received medication and `no` if they did not. Similarly, the column called `exercise` has the value `yes` if the participant was in the exercise program and `no` if they were not.

```r
set.seed(19842)

study_data <- data.frame(
  health = rnorm(15, mean = c(rep(50, 5), rep(70, 5), rep(85,5)), sd = 5),
  medication = c(rep("no", 5), rep("yes", 5), rep("yes", 5)),
  exercise = c(rep("no", 5), rep("no", 5), rep("yes", 5))
)
```

(a) Consider an additive two-way ANOVA model for these data:

$$
\begin{aligned}
y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i, \text{ where} \\
x_{i1} &= \begin{cases} 0 \text{ if subject } i \text{ did not take the medication} \\ 1 \text{ if subject } i \text{ did take the medication} \end{cases} \\
x_{i2} &= \begin{cases} 0 \text{ if subject } i \text{ did not participate in the exercise program} \\ 1 \text{ if subject } i \text{ did participate in the exercise program} \end{cases} \\
\varepsilon_i &\sim \text{Normal}(0, \sigma^2)
\end{aligned}
$$

i. Based on this model specification, find the mean health score for the following four combinations, in terms of β_0, β_1, and β_2:

- No medication, no exercise: β_0
- Medication, no exercise: $\beta_0 + \beta_1$
- No medication, exercise: $\beta_0 + \beta_2$
- Medication, exercise: $\beta_0 + \beta_1 + \beta_2$

ii. In this model specification, what are the interpretations of each of β_0, β_1, and β_2? It may help to compare your answers to part i.

β_0 represents the mean health score for people in the control group (no medication and no exercise).

β_1 represents the difference in mean health scores between groups that are taking medicine and are not taking medicine, holding fixed whether or not the person is exercising.

β_2 represents the difference in mean health scores between groups that are participating in the exercise program and are not participating in the exercise program, holding fixed whether they are taking the medication.

iii. Fit the additive model specified above to these data, and print out a model summary.

```r
model_fit <- lm(health ~ medication + exercise, data = study_data)
summary(model_fit)
```

```
##
## Call:
## lm(formula = health ~ medication + exercise, data = study_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.6568  -1.9444  -0.0702   1.7293   9.1401
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)     50.087      2.499  20.041 1.36e-10 ***
## medicationyes   22.128      3.534   6.261 4.19e-05 ***
## exerciseyes     14.216      3.534   4.022  0.00169 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 5.588 on 12 degrees of freedom
## Multiple R-squared:  0.8995, Adjusted R-squared:  0.8828 
## F-statistic:  53.7 on 2 and 12 DF,  p-value: 1.03e-06
```

iv. Extract the design matrix X from the model fit object and print it out. Does this matrix have full column rank? If so, explain how you can tell; if not, find a way to express one of the columns of X as a linear combination of the others.

```r
X <- model.matrix(model_fit)
X
```

```
##    (Intercept) medicationyes exerciseyes
## 1            1             0           0
## 2            1             0           0
## 3            1             0           0
## 4            1             0           0
## 5            1             0           0
## 6            1             1           0
## 7            1             1           0
## 8            1             1           0
## 9            1             1           0
## 10           1             1           0
## 11           1             1           1
## 12           1             1           1
## 13           1             1           1
## 14           1             1           1
## 15           1             1           1
## attr(,"assign")
## [1] 0 1 2
## attr(,"contrasts")
## attr(,"contrasts")$medication
## [1] "contr.treatment"
## 
## attr(,"contrasts")$exercise
## [1] "contr.treatment"
```

```r
solve(t(X) %*% X)
```

```
##               (Intercept) medicationyes exerciseyes
## (Intercept)           0.2          -0.2         0.0
## medicationyes        -0.2           0.4        -0.2
## exerciseyes           0.0          -0.2         0.4
```

Yes, X has full column rank. There is no way to obtain any of the columns as a linear combination of the other two. We also confirmed that $X'X$ was invertible. Note that the model summary output contained estimates for all three model parameters.

v. Find the residual sum of squares from your model fit in part iii.

```
sum((study_data$health - predict(model_fit))^2)
```

```
## [1] 374.7673
```

(b) Now consider a model with interactions:

$$
\begin{aligned}
y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i1} x_{i2} + \varepsilon_i, \text{ where} \\
x_{i1} &= \begin{cases} 0 \text{ if subject } i \text{ did not take the medication} \\ 1 \text{ if subject } i \text{ did take the medication} \end{cases} \\
x_{i2} &= \begin{cases} 0 \text{ if subject } i \text{ did not participate in the exercise program} \\ 1 \text{ if subject } i \text{ did participate in the exercise program} \end{cases} \\
\varepsilon_i &\sim \text{Normal}(0, \sigma^2)
\end{aligned}
$$

i. Based on this model specification, find the mean health score for the following four combinations, in terms of β_0, β_1, β_2, and β_3:

- No medication, no exercise: β_0
- Medication, no exercise: $\beta_0 + \beta_1$
- No medication, exercise: $\beta_0 + \beta_2$
- Medication, exercise: $\beta_0 + \beta_1 + \beta_2 + \beta_3$

ii. In this model specification, what are the interpretations of each of β_0, β_1, β_2, and β_3? It may help to compare your answers to part i.

β_0 represents the mean health score for people in the control group (no medication and no exercise).

β_1 represents the difference in mean health scores between people who aren't in the exercise program but take medication and people who aren't in the exercise program and also don't take medication.

β_2 represents the difference in mean health scores between people who are participating in the exercise program but not taking medicine and people who are not participating int he exercise program and also don't take medicine.

β_3 represents the additional change in mean health scores due to adding medication when someone is exercising relative to when they are not exercising. Equivalently, β_3 represents the additional change in mean health scores due to adding participation in an exercise program when someone is taking the medication relative to when they are not taking the medication.

iii. Fit the interactions model specified above to these data, and print out a model summary.

```
model_fit <- lm(health ~ medication * exercise, data = study_data)
summary(model_fit)
```

```
##
## Call:
## lm(formula = health ~ medication * exercise, data = study_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.6568  -1.9444  -0.0702   1.7293   9.1401
```

```
##
## Coefficients: (1 not defined because of singularities)
##                            Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                  50.087      2.499  20.041 1.36e-10 ***
## medicationyes                22.128      3.534   6.261 4.19e-05 ***
## exerciseyes                  14.216      3.534   4.022  0.00169 ** 
## medicationyes:exerciseyes        NA         NA      NA       NA    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.588 on 12 degrees of freedom
## Multiple R-squared:  0.8995, Adjusted R-squared:  0.8828 
## F-statistic:  53.7 on 2 and 12 DF,  p-value: 1.03e-06
```

iv. Extract the design matrix from the model fit object and print it out. Does this matrix have full column rank? If so, explain how you can tell; if not, find a way to express one of the columns of X as a linear combination of the others.

```
X <- model.matrix(model_fit)
X
```

```
##    (Intercept) medicationyes exerciseyes medicationyes:exerciseyes
## 1            1             0           0                         0
## 2            1             0           0                         0
## 3            1             0           0                         0
## 4            1             0           0                         0
## 5            1             0           0                         0
## 6            1             1           0                         0
## 7            1             1           0                         0
## 8            1             1           0                         0
## 9            1             1           0                         0
## 10           1             1           0                         0
## 11           1             1           1                         1
## 12           1             1           1                         1
## 13           1             1           1                         1
## 14           1             1           1                         1
## 15           1             1           1                         1
## attr(,"assign")
## [1] 0 1 2 3
## attr(,"contrasts")
## attr(,"contrasts")$medication
## [1] "contr.treatment"
##
## attr(,"contrasts")$exercise
## [1] "contr.treatment"
```

```
solve(t(X) %*% X)
```

```
## Error in solve.default(t(X) %*% X): Lapack routine dgesv: system is exactly singular: U[4,4] = 0
```

This design matrix does not have full columnn rank: the third and fourth columns are equal to each other.

v. Find two different sets of parameter estimates $\hat{\beta}_0$, $\hat{\beta}_1$, $\hat{\beta}_2$, and $\hat{\beta}_3$ that give the same residual sum of squares that you obtained in part (a) v.

```
beta_hat_a <- matrix(coef(model_fit))
beta_hat_a[4, 1] <- 0
sum((study_data$health - X %*% beta_hat_a)^2)
```

```
## [1] 374.7673
```

```
beta_hat_b <- beta_hat_a
beta_hat_a[4, 1] <- beta_hat_a[3, 1]
beta_hat_a[3, 1] <- 0
sum((study_data$health - X %*% beta_hat_b)^2)
```

```
## [1] 374.7673
```

```
beta_hat_a
```

```
##          [,1]
## [1,] 50.08732
## [2,] 22.12768
## [3,]  0.00000
## [4,] 14.21634
```

```
beta_hat_b
```

```
##          [,1]
## [1,] 50.08732
## [2,] 22.12768
## [3,] 14.21634
## [4,]  0.00000
```

Problem 4: Polynomial Regression Example 1

The following R code loads in a data set with measurements of the tensile strength of paper (`tensile`, in units of pounds per square inch), and the percent of hardwood in the batch of pulp that was used to produce the paper (`hardwood`), for 19 different samples of paper with different percent hardwoods.

```
paper <- read_csv("http://www.evanlray.com/data/BSDA/Hardwood.csv")
```

```
## Parsed with column specification:
## cols(
##   tensile = col_double(),
##   hardwood = col_double()
## )
```

References:

These data, and the description above, come from the R package for "Basic Statistics and Data Analysis" by Alan T. Arnholt: https://alanarnholt.github.io/BSDA/

(a) Fit and summarize polynomial regression models of degree 2, 3, and 4.

For each of these three candidate models, please produce:

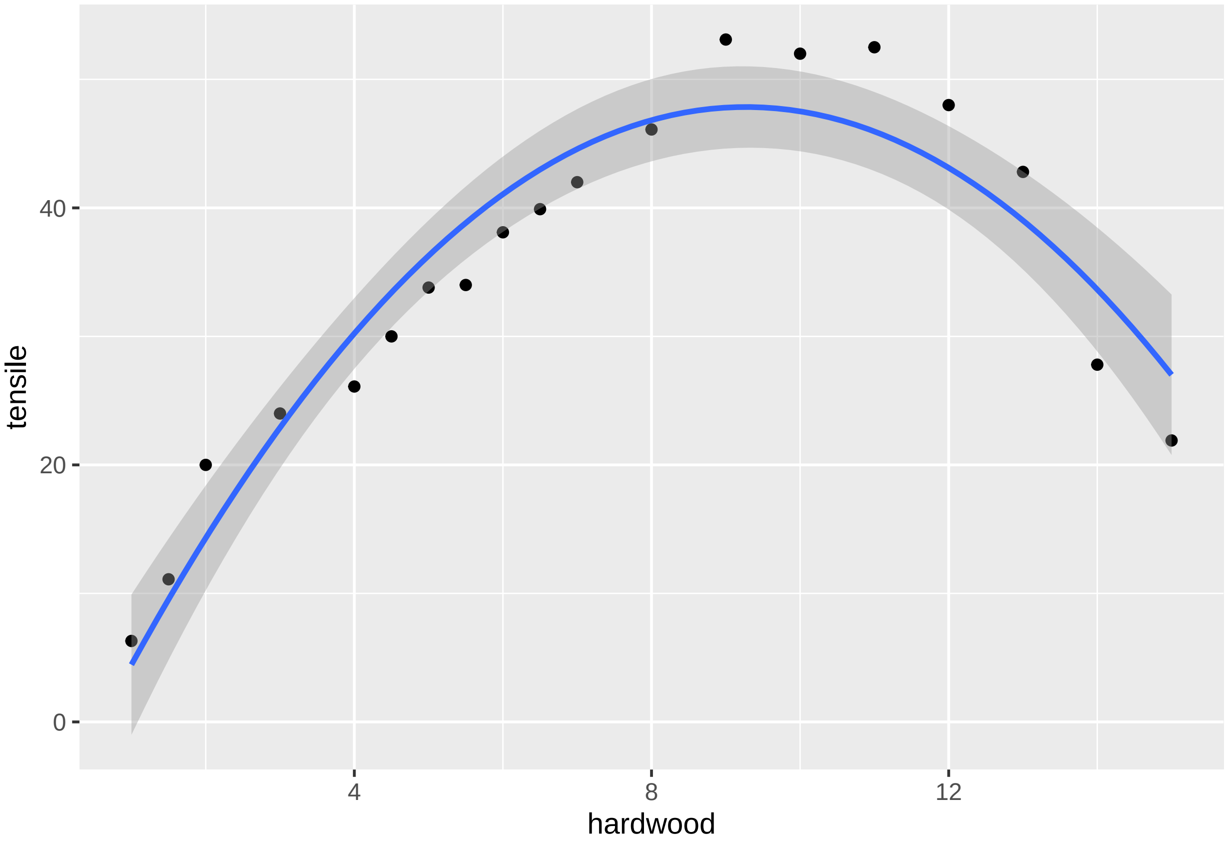
- Output from the `summary` function that you could use to conduct relevant hypothesis tests
- A scatter plot of the data with the estimated curve overlaid on top
- A plot of either residuals vs. fitted values or residuals vs. percent hardwood in the pulp (your choice)
- The residual sum of squares (RSS). (You should also know how to find the R^2 and residual standard error (RSE) in the `summary` output.)

Degree 2:

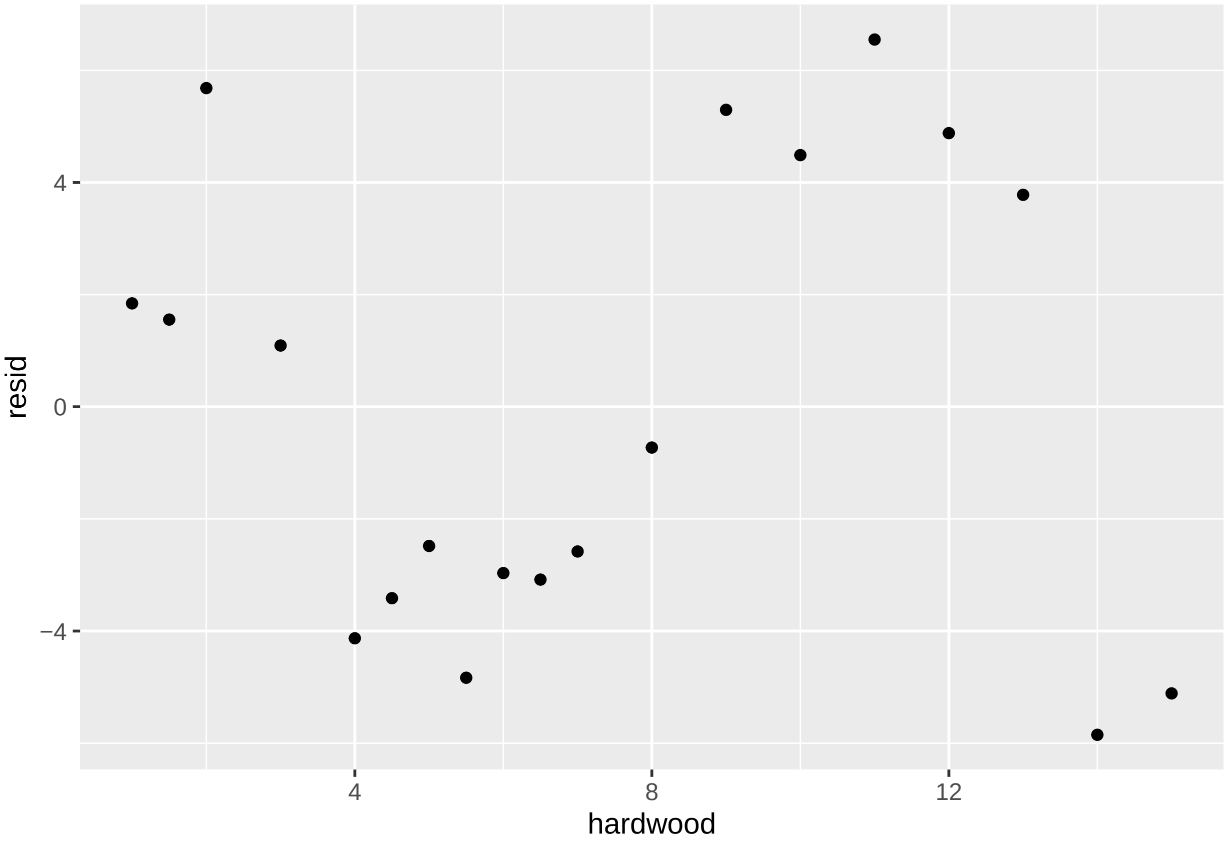
```
fit2 <- lm(tensile ~ poly(hardwood, 2, raw = TRUE), data = paper)
summary(fit2)
```

```
##
## Call:
## lm(formula = tensile ~ poly(hardwood, 2, raw = TRUE), data = paper)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8503 -3.2482 -0.7267  4.1350  6.5506
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -6.67419    3.39971  -1.963   0.0673 .
## poly(hardwood, 2, raw = TRUE)1 11.76401    1.00278  11.731 2.85e-09 ***
## poly(hardwood, 2, raw = TRUE)2 -0.63455    0.06179 -10.270 1.89e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.42 on 16 degrees of freedom
## Multiple R-squared:  0.9085, Adjusted R-squared:  0.8971
## F-statistic: 79.43 on 2 and 16 DF,  p-value: 4.912e-09
```

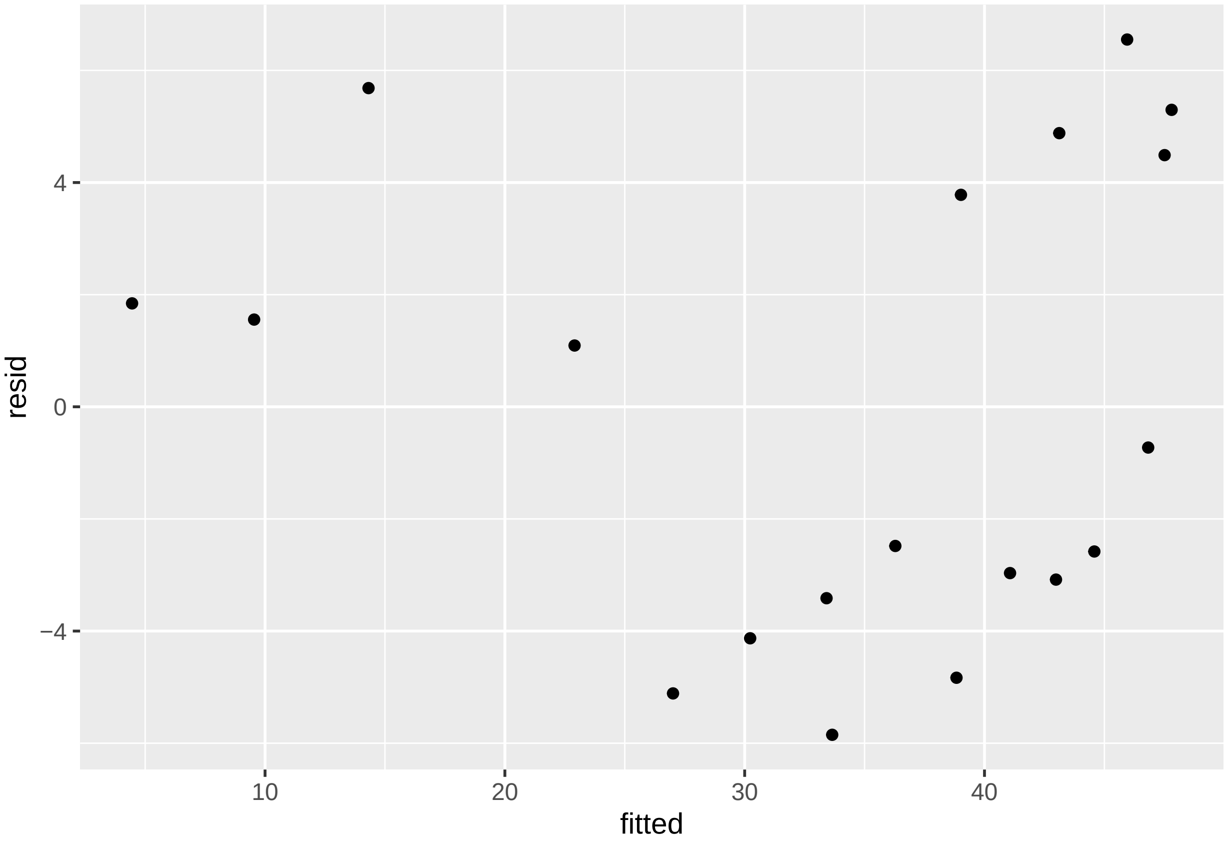
```
paper <- paper %>%
  mutate(
    resid = residuals(fit2),
    fitted = predict(fit2)
  )

ggplot(data = paper, mapping = aes(x = hardwood, y = tensile)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2, raw = TRUE))
```



```
ggplot(data = paper, mapping  = aes(x = hardwood, y = resid)) +
  geom_point()
```



```
ggplot(data = paper, mapping  = aes(x = fitted, y = resid)) +
  geom_point()
```



```
# Training RSS
paper %>% summarize(train_RSS = mean(resid^2))
```

```
## # A tibble: 1 x 1
##   train_RSS
##       <dbl>
## 1      16.5
```

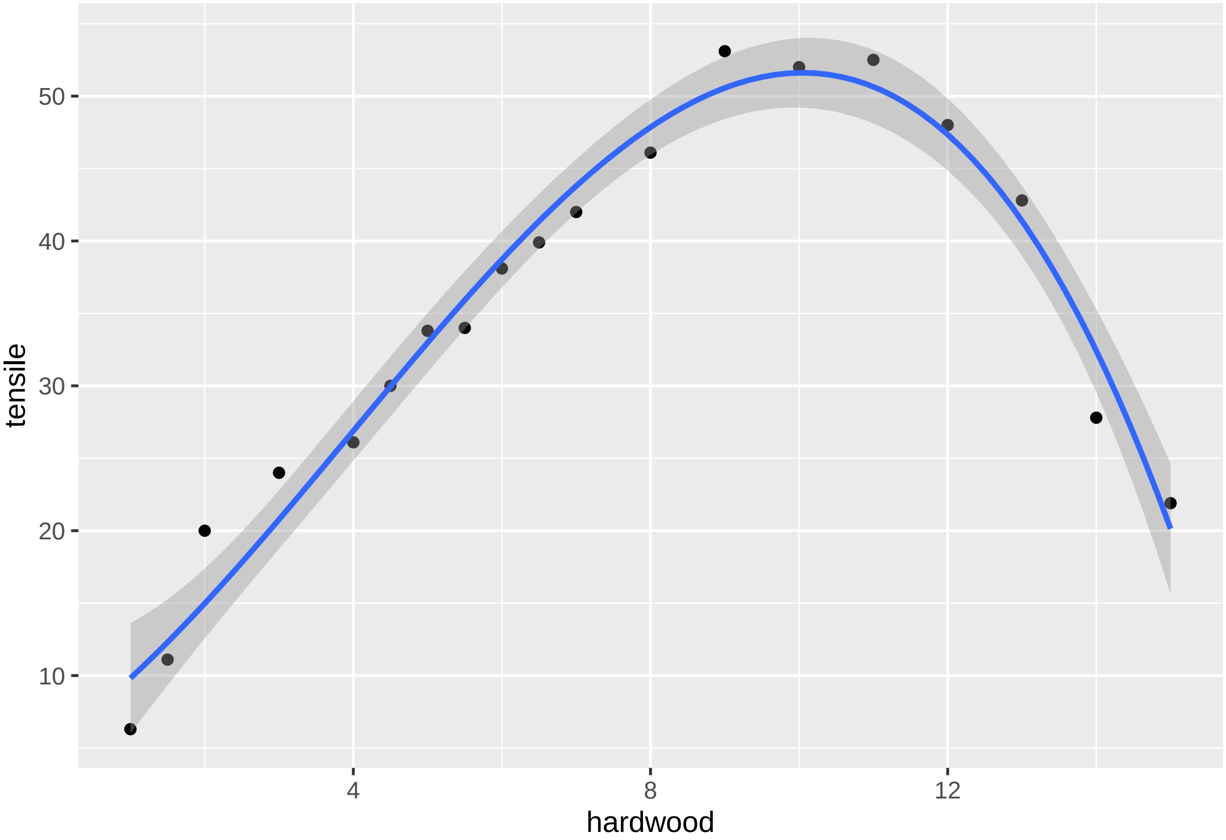
Degree 3:

```
fit3 <- lm(tensile ~ poly(hardwood, 3, raw = TRUE), data = paper)
summary(fit3)
```

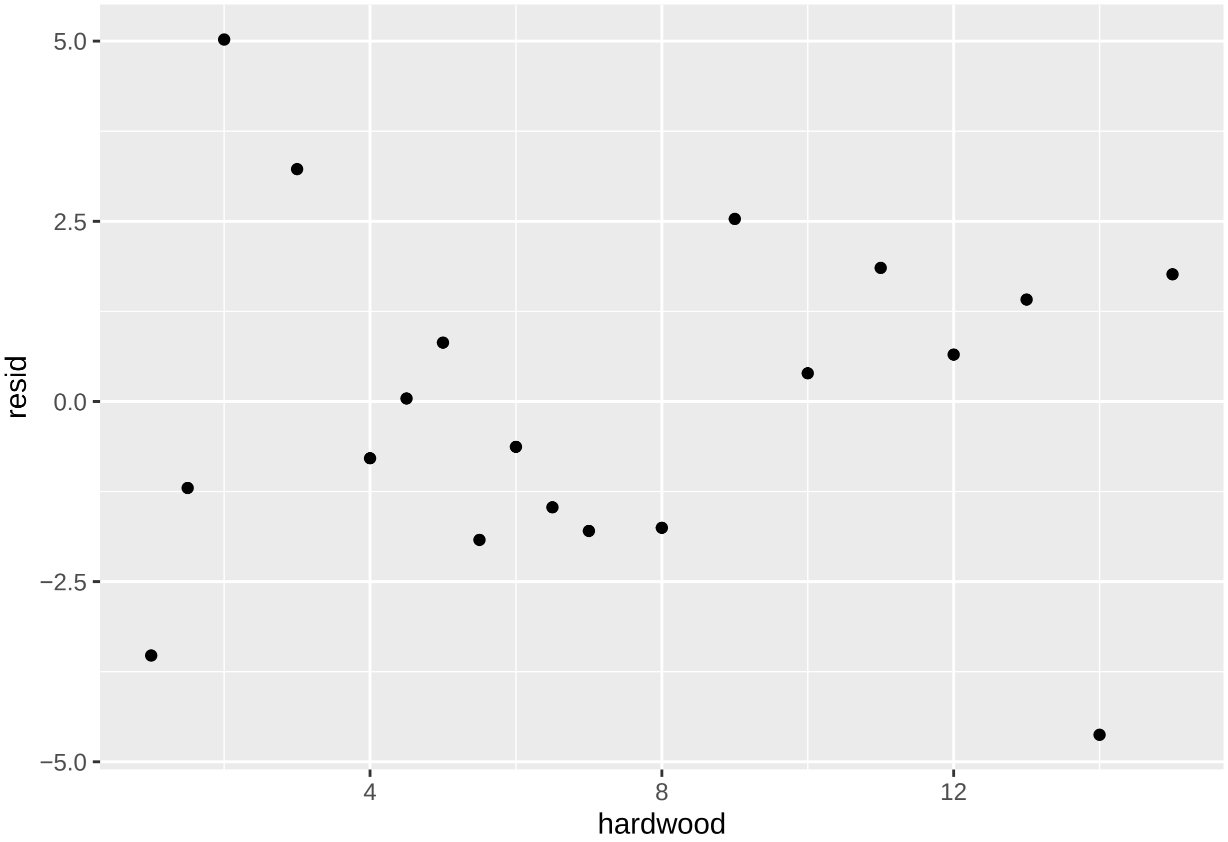
```
##
## Call:
## lm(formula = tensile ~ poly(hardwood, 3, raw = TRUE), data = paper)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6250 -1.6109  0.0413  1.5892  5.0216
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     5.648395   2.954663   1.912   0.0752 .
## poly(hardwood, 3, raw = TRUE)1  3.578489   1.565854   2.285   0.0373 *
## poly(hardwood, 3, raw = TRUE)2  0.653635   0.231330   2.826   0.0128 *
## poly(hardwood, 3, raw = TRUE)3 -0.055188   0.009789  -5.638 4.72e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.585 on 15 degrees of freedom
## Multiple R-squared:  0.9707, Adjusted R-squared:  0.9648
## F-statistic: 165.4 on 3 and 15 DF,  p-value: 1.025e-11
```

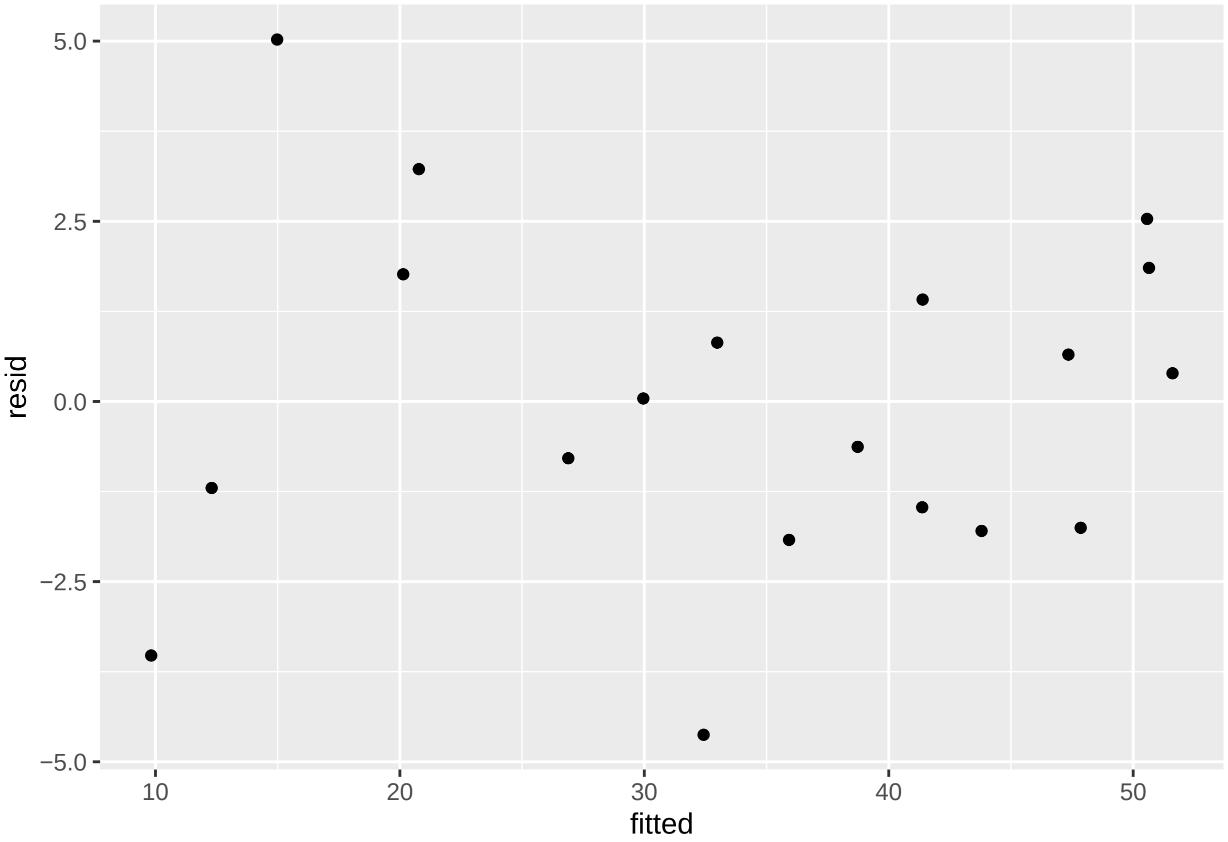
```
paper <- paper %>%
  mutate(
    resid = residuals(fit3),
    fitted = predict(fit3)
  )

ggplot(data = paper, mapping = aes(x = hardwood, y = tensile)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ poly(x, 3, raw = TRUE))
```



```
ggplot(data = paper, mapping  = aes(x = hardwood, y = resid)) +
  geom_point()
```



```
ggplot(data = paper, mapping  = aes(x = fitted, y = resid)) +
  geom_point()
```



```
# Training RSS
paper %>% summarize(train_RSS = mean(resid^2))
```

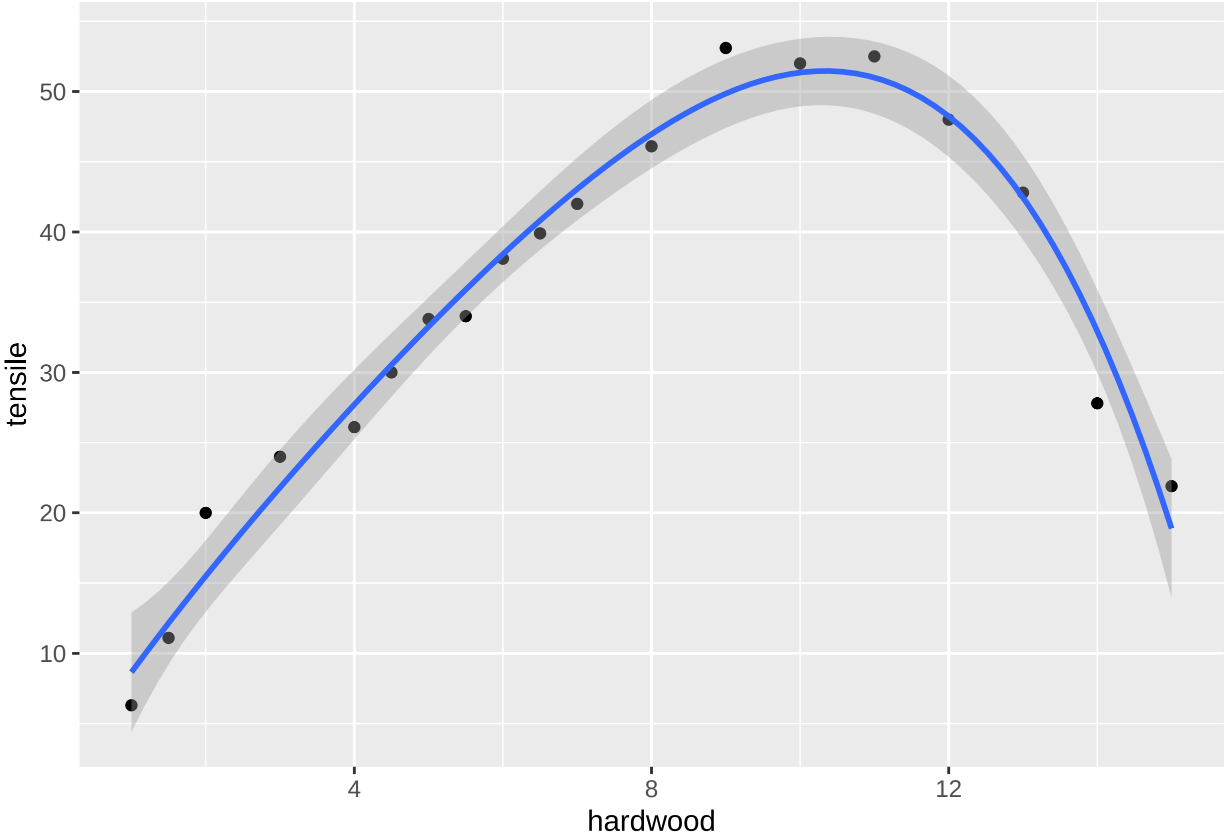
```
## # A tibble: 1 x 1
##   train_RSS
##       <dbl>
## 1      5.28
```

Degree 4:

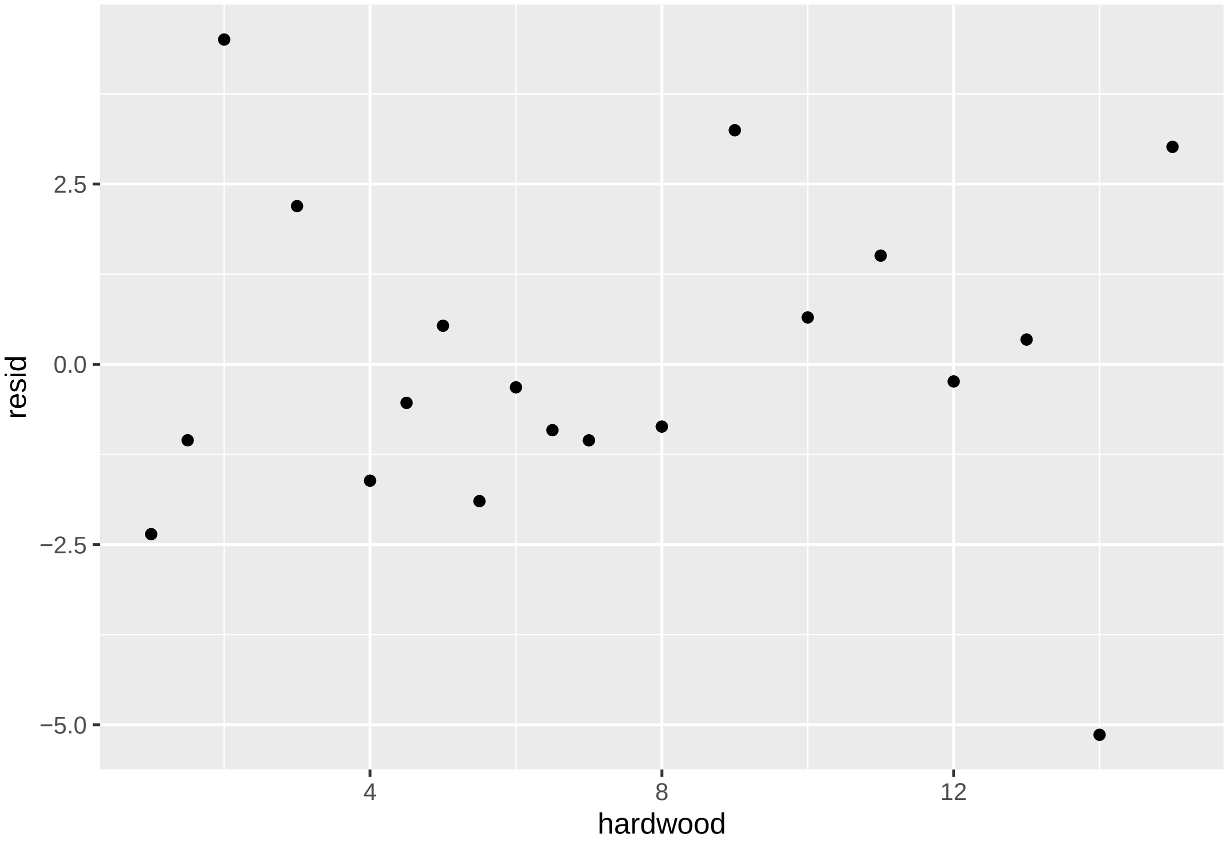
```
fit4 <- lm(tensile ~ poly(hardwood, 4, raw = TRUE), data = paper)
summary(fit4)
```

```
##
## Call:
## lm(formula = tensile ~ poly(hardwood, 4, raw = TRUE), data = paper)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1384 -1.0550 -0.3203  1.0779  4.5030
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.070958   4.679866   0.229   0.8223
## poly(hardwood, 4, raw = TRUE)1  8.049481   3.902170   2.063   0.0582 .
## poly(hardwood, 4, raw = TRUE)2 -0.517352   0.966390  -0.535   0.6008
## poly(hardwood, 4, raw = TRUE)3  0.056575   0.090164   0.627   0.5405
## poly(hardwood, 4, raw = TRUE)4 -0.003505   0.002812  -1.247   0.2330
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.539 on 14 degrees of freedom
## Multiple R-squared:  0.9736, Adjusted R-squared:  0.9661
## F-statistic: 129.1 on 4 and 14 DF,  p-value: 6.994e-11
```

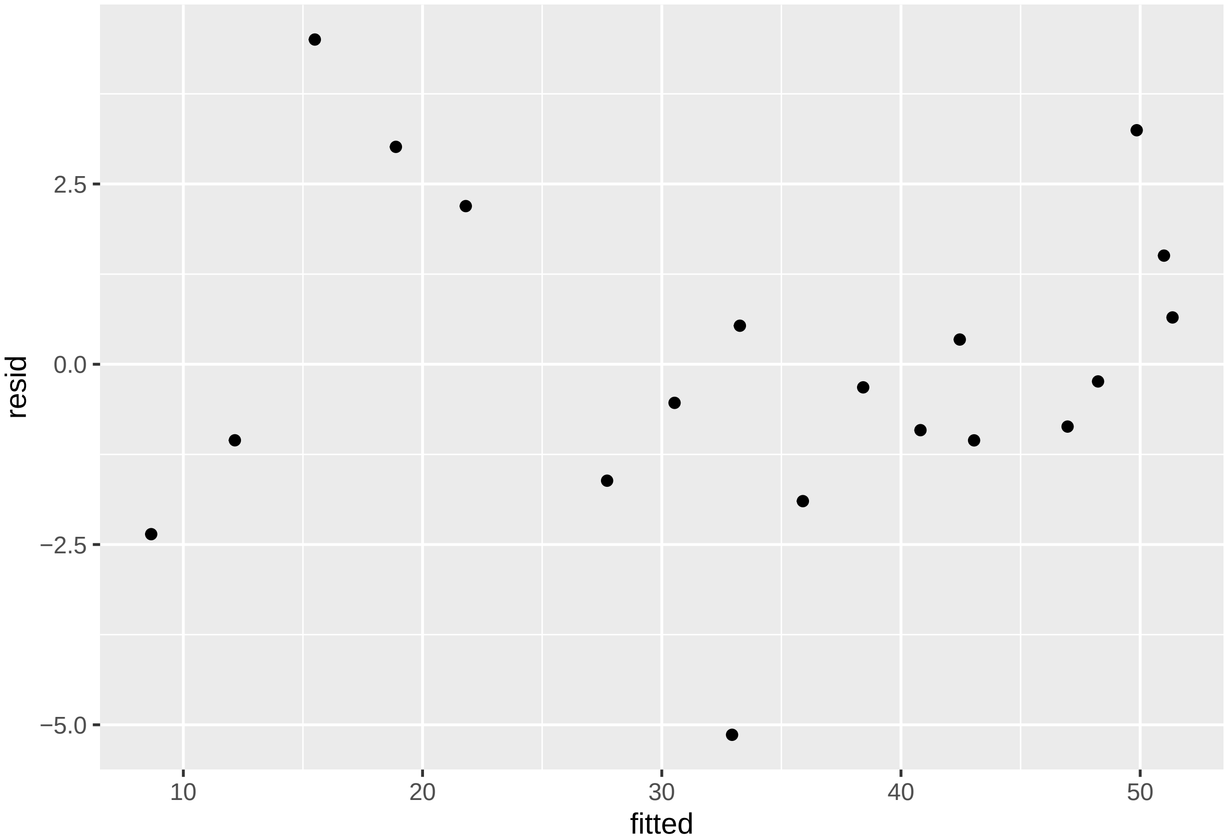
```
paper <- paper %>%
  mutate(
    resid = residuals(fit4),
    fitted = predict(fit4)
  )

ggplot(data = paper, mapping = aes(x = hardwood, y = tensile)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ poly(x, 4, raw = TRUE))
```



```
ggplot(data = paper, mapping  = aes(x = hardwood, y = resid)) +
  geom_point()
```



```r
ggplot(data = paper, mapping  = aes(x = fitted, y = resid)) +
  geom_point()
```



```
# Training RSS
paper %>% summarize(train_RSS = mean(resid^2))
```

```
## # A tibble: 1 x 1
##   train_RSS
##       <dbl>
## 1      4.75
```

(b) Based on your results above, which model do you prefer?

You should justify your decision with reference to a discussion of a couple of the plots you created and a comparison of RSS. You can also discuss hypothesis test results if you want, but that's not required. You don't have to discuss all of the plots and all of the statistics, but you should know how to interpret them all.

I prefer the degree 3 polynomial. The plots indicate a clear lack of fit for the degree 2 polynomial. In the plot with the estimated curve overlaid on the scatter plot, we see that the estimated curve is generally too low for the first few observations, too high for the next few observations, too low again near the peak, and too high for the last few observations. This shows up in the residual plots with a sinusoidal pattern for the residuals vs. explanatory variable, and a curved pattern for residuals vs. predicted. These trends were not as strong in the plots for the degree 3 and degree 4 models. There wasn't much of a difference in the plots for the degree 3 and degree 4 models, and so I prefer the simpler of the two.

The hypothesis test results indicate that the degree 2 and degree 3 terms are "statistically significant". The degree 4 term does not account for a statistically significant amount of variation in the response variable if the degree 2 and degree 3 terms are included.

As we've discussed, it's not useful to compare training set RSS, R^2, or RSE for models with different levels of flexibility. The R^2 was always higher for higher degree models, and the RSS and RSE were always lower for higher degree models. This did not indicate a meaningfully improved fit though.

(c) Suppose I got a new sample based on a new batch of pulp from the same factory. Do you have confidence that the predicted tensile strength from your selected model from part (b) would be similar to the observed values of tensile strength in my new sample? Why or why not? Just write a sentence or two.

Two possible answers:

1. Based on the plots, it does not appear that we have overfit our training data set. I feel fairly confident that predictions for a test set observation would perform well.
2. There could be systematic differences between different batches of pulp. Since our model was fit based on samples from a single batch of pulp, I will always be cautious about extrapolation to different batches. I would really like to have data from multiple batches included in my training data set.

(d) Extract the model matrix from your degree 2 polynomial fit, and use it to find the coefficient estimates $\hat{\beta}$, the hat matrix H, and the fitted values $\hat{y}$ through direct matrix manipulations.

```
X <- model.matrix(fit2)
y <- matrix(paper$tensile)

beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y
beta_hat
```

```
##                                     [,1]
## (Intercept)                   -6.6741916
## poly(hardwood, 2, raw = TRUE)1 11.7640057
## poly(hardwood, 2, raw = TRUE)2 -0.6345492
```

```
H <- X %*% solve(t(X) %*% X) %*% t(X)
H
```

```
##               1            2            3            4            5
## 1   0.339085846  0.291660100  0.247103621  0.166598463  0.097570372
## 2   0.291660100  0.253559071  0.217680184  0.152588836  0.096386055
## 3   0.247103621  0.217680184  0.189880611  0.139153053  0.094920948
## 4   0.166598463  0.152588836  0.139153053  0.114003020  0.091148363
## 5   0.097570372  0.096386055  0.094920948  0.091148363  0.086252617
## 6   0.067360227  0.071617877  0.075240690  0.080581800  0.083383559
## 7   0.040019348  0.049071841  0.057184295  0.070589082  0.080233711
## 8   0.015547737  0.028747947  0.040751763  0.061170208  0.076803072
## 9  -0.006054608  0.010646195  0.025943094  0.052325178  0.073091643
## 10 -0.024787686 -0.005233415  0.012758288  0.044053992  0.069099424
## 11 -0.040651497 -0.018890883  0.001197346  0.036356650  0.064826415
## 12 -0.063771319 -0.039539395 -0.017052950  0.022683498  0.055438026
## 13 -0.075414074 -0.051299339 -0.028807794  0.011305723  0.044926477
## 14 -0.075579761 -0.054170715 -0.034067185  0.002223324  0.033291767
## 15 -0.064268382 -0.048153524 -0.032831124 -0.004563698  0.020533895
## 16 -0.041479935 -0.033247766 -0.025099611 -0.009055345  0.006652864
## 17 -0.007214421 -0.009453440 -0.010872645 -0.011251614 -0.008351329
## 18  0.038528161  0.023229453  0.009849773 -0.011152508 -0.024478682
## 19  0.095747809  0.064800914  0.037067643 -0.008758025 -0.041729197
##               6            7            8            9           10
## 1   0.067360227  0.040019348  0.015547737 -0.006054608 -0.024787686
## 2   0.071617877  0.049071841  0.028747947  0.010646195 -0.005233415
## 3   0.075240690  0.057184295  0.040751763  0.025943094  0.012758288
## 4   0.080581800  0.070589082  0.061170208  0.052325178  0.044053992
## 5   0.083383559  0.080233711  0.076803072  0.073091643  0.069099424
## 6   0.083832181  0.083645965  0.082824911  0.081369019  0.079278289
## 7   0.083645965  0.086118180  0.087650355  0.088242490  0.087894586
## 8   0.082824911  0.087650355  0.091279404  0.093712057  0.094948315
## 9   0.081369019  0.088242490  0.093712057  0.097777719  0.100439477
## 10  0.079278289  0.087894586  0.094948315  0.100439477  0.104368070
## 11  0.076552721  0.086606642  0.094988178  0.101697329  0.106734096
## 12  0.069197070  0.081210634  0.091478718  0.100001321  0.106778445
## 13  0.059302068  0.072054468  0.083183677  0.092689695  0.100572523
## 14  0.046867713  0.059138142  0.070103055  0.079762450  0.088116329
## 15  0.031894005  0.042461657  0.052236851  0.061219587  0.069409865
## 16  0.014380946  0.022025014  0.029585067  0.037061106  0.044453130
## 17 -0.005671466 -0.002171789  0.002147702  0.007287006  0.013246124
## 18 -0.028263230 -0.030128750 -0.030075244 -0.028102712 -0.024211153
## 19 -0.053394346 -0.061845871 -0.067083772 -0.069108048 -0.067918701
##              11           12          13           14           15
## 1  -0.040651497 -0.063771319 -0.07541407 -0.075579761 -0.064268382
## 2  -0.018890883 -0.039539395 -0.05129934 -0.054170715 -0.048153524
## 3   0.001197346 -0.017052950 -0.02880779 -0.034067185 -0.032831124
## 4   0.036356650  0.022683498  0.01130572  0.002223324 -0.004563698
## 5   0.064826415  0.055438026  0.04492648  0.033291767  0.020533895
## 6   0.076552721  0.069197070  0.05930207  0.046867713  0.031894005
## 7   0.086606642  0.081210634  0.07205447  0.059138142  0.042461657
## 8   0.094988178  0.091478718  0.08318368  0.070103055  0.052236851
## 9   0.101697329  0.100001321  0.09268970  0.079762450  0.061219587
## 10  0.106734096  0.106778445  0.10057252  0.088116329  0.069409865
## 11  0.110098478  0.111810088  0.10683216  0.095164692  0.076807685
## 12  0.111810088  0.116636935  0.11448186  0.105344866  0.089225951
## 13  0.106832159  0.114481861  0.11563880  0.110302974  0.098474386
## 14  0.095164692  0.105344866  0.11030297  0.110039014  0.104552988
## 15  0.076807685  0.089225951  0.09847439  0.104552988  0.107461758
## 16  0.051761140  0.066125116  0.08015303  0.093844894  0.107200697
## 17  0.020025056  0.036042361  0.05533892  0.077914734  0.103769803
## 18 -0.018400567 -0.001022316  0.02403204  0.056762507  0.097169078
## 19 -0.063515729 -0.045068912 -0.01376760  0.030388212  0.087398520
##              16           17           18           19
## 1  -0.041479935 -0.007214421  0.038528161  0.095747809
## 2  -0.033247766 -0.009453440  0.023229453  0.064800914
## 3  -0.025099611 -0.010872645  0.009849773  0.037067643
## 4  -0.009055345 -0.011251614 -0.011152508 -0.008758025
## 5   0.006652864 -0.008351329 -0.024478682 -0.041729197
## 6   0.014380946 -0.005671466 -0.028263230 -0.053394346
## 7   0.022025014 -0.002171789 -0.030128750 -0.061845871
## 8   0.029585067  0.002147702 -0.030075244 -0.067083772
## 9   0.037061106  0.007287006 -0.028102712 -0.069108048
## 10  0.044453130  0.013246124 -0.024211153 -0.067918701
## 11  0.051761140  0.020025056 -0.018400567 -0.063515729
## 12  0.066125116  0.036042361 -0.001022316 -0.045068912
## 13  0.080153034  0.055338920  0.024032042 -0.013767598
## 14  0.093844894  0.077914734  0.056762507  0.030388212
## 15  0.107200697  0.103769803  0.097169078  0.087398520
## 16  0.120220441  0.132904127  0.145251755  0.157263325
## 17  0.132904127  0.165317705  0.201010539  0.239982627
## 18  0.145251755  0.201010539  0.264445429  0.335556426
## 19  0.157263325  0.239982627  0.335556426  0.443984722
```

```
y_hat <- X %*% beta_hat
y_hat
```

```
##         [,1]
## 1   4.455265
## 2   9.544081
## 3  14.315623
## 4  22.906883
## 5  30.229044
## 6  33.414213
## 7  36.282107
## 8  38.832727
## 9  41.066072
## 10 42.982143
## 11 44.580939
## 12 46.826707
## 13 47.803377
## 14 47.510948
## 15 45.949421
## 16 43.118796
## 17 39.019072
## 18 33.650250
## 19 27.012330
```

```
y_hat_v2 <- H %*% y
y_hat_v2
```

```
##         [,1]
## 1   4.455265
## 2   9.544081
## 3  14.315623
## 4  22.906883
## 5  30.229044
## 6  33.414213
## 7  36.282107
## 8  38.832727
## 9  41.066072
## 10 42.982143
## 11 44.580939
## 12 46.826707
## 13 47.803377
## 14 47.510948
## 15 45.949421
## 16 43.118796
## 17 39.019072
## 18 33.650250
## 19 27.012330
```

You only needed to calculate $\hat{y}$ one way.

Problem 5: Polynomial Regression Example 2

The United Nations Development Programme (UNDP) uses the Human Development Index (HDI) in an attempt to summarize in one number the progress in health, education, and economics of a country. In 2012, the HDI was as high as 0.955 for Norway and as low as 0.304 for Niger. The gross national income per capita, by contrast, is often used to summarize the overall economic strength of a country. In this example we will consider models between these variables, with `gni_per_cap` as the explanatory variable and `hdl` as the response variable.

The following R chunk reads in data including the HDI and gross national income per capita for 187 countries as of 2012, and creates an initial plot.

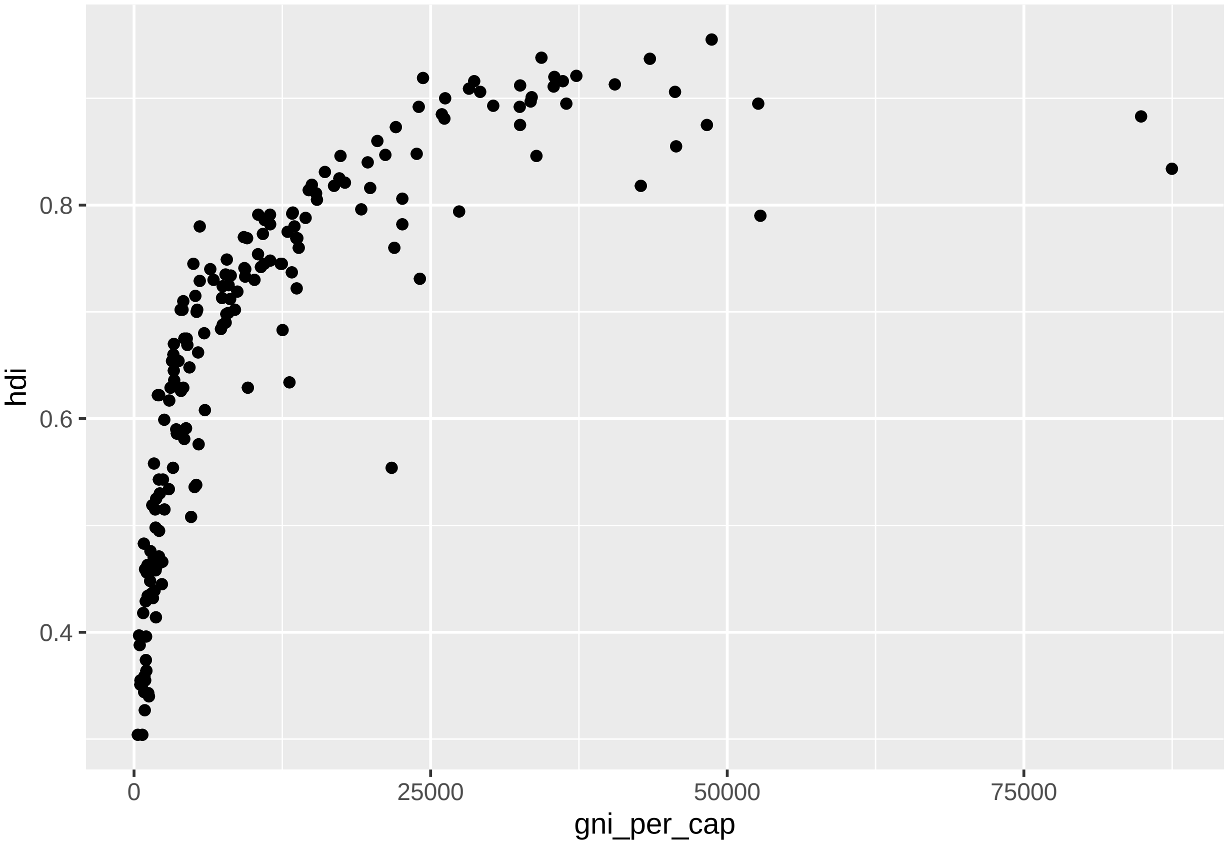
```
country_data <- read_csv("https://mhc-stat140-2017.github.io/data/sdm4/HDI_2012.csv")
```

```
## Parsed with column specification:
## cols(
##   `HDI rank` = col_double(),
##   Country = col_character(),
##   Abbreviation = col_character(),
##   HDI = col_double(),
##   `Life Expectancy` = col_double(),
##   `Mean School yrs 2010` = col_double(),
##   `Exp school yrs 2011` = col_double(),
##   `GNI/cap2012` = col_double(),
##   `GNI rank-HDIrank` = col_double(),
##   `NonIncome HDI 2012` = col_double(),
##   Type = col_character()
## )
```

```
names(country_data) <- c("hdi_rank", "country", "country_abbr", "hdi", "life_expectancy", "mean_school_
head(country_data)
```

```
## # A tibble: 6 x 11
##   hdi_rank country country_abbr   hdi life_expectancy mean_school_yea~
##      <dbl> <chr>   <chr>        <dbl>           <dbl>            <dbl>
## 1      175 Afghan~ AFG          0.374            49.1              3.1
## 2       70 Albania ALB          0.749            77.1             10.4
## 3       93 Algeria DZA          0.713            73.4              7.6
## 4       33 Andorra AND          0.846            81.1             10.4
## 5      148 Angola  AGO          0.508            51.5              4.7
## 6       67 Antigu~ ATG          0.76             72.8              8.9
## # ... with 5 more variables: exp_chool_years <dbl>, gni_per_cap <dbl>,
## #   gni_rank_minus_hdi_rank <dbl>, non_income_hdi <dbl>, type <chr>
```

```
ggplot(data = country_data, mapping = aes(x = gni_per_cap, y = hdi)) +
  geom_point()
```



References:

These data, and the description above, are from Statistics: Data and Models, 4th edition, by De Veaux et al.

(a) Fit a degree 8 polynomial and use the summary function to print out a summary of the model fit. If you used a hypothesis test at the $\alpha = 0.05$ significance level to pick a polynomial model for these data, which model would you select? You don't need to conduct any additional model checks in this part of the problem.

```
fit8 <- lm(hdi ~ poly(gni_per_cap, 8, raw = TRUE), data = country_data)
summary(fit8)
```

```
##
## Call:
## lm(formula = hdi ~ poly(gni_per_cap, 8, raw = TRUE), data = country_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.274724 -0.025835  0.005165  0.034101  0.116333
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        2.806e-01  1.765e-02  15.898  < 2e-16
## poly(gni_per_cap, 8, raw = TRUE)1  1.459e-04  1.404e-05  10.393  < 2e-16
## poly(gni_per_cap, 8, raw = TRUE)2 -1.943e-08  3.310e-09  -5.871 2.08e-08
## poly(gni_per_cap, 8, raw = TRUE)3  1.394e-12  3.408e-13   4.092 6.48e-05
## poly(gni_per_cap, 8, raw = TRUE)4 -5.648e-17  1.794e-17  -3.148  0.00193
## poly(gni_per_cap, 8, raw = TRUE)5  1.338e-21  5.163e-22   2.592  0.01033
## poly(gni_per_cap, 8, raw = TRUE)6 -1.834e-26  8.157e-27  -2.248  0.02582
## poly(gni_per_cap, 8, raw = TRUE)7  1.337e-31  6.593e-32   2.028  0.04405
## poly(gni_per_cap, 8, raw = TRUE)8 -3.996e-37  2.121e-37  -1.884  0.06125
##
## (Intercept)                       ***
## poly(gni_per_cap, 8, raw = TRUE)1 ***
## poly(gni_per_cap, 8, raw = TRUE)2 ***
## poly(gni_per_cap, 8, raw = TRUE)3 ***
## poly(gni_per_cap, 8, raw = TRUE)4 **
## poly(gni_per_cap, 8, raw = TRUE)5 *
## poly(gni_per_cap, 8, raw = TRUE)6 *
## poly(gni_per_cap, 8, raw = TRUE)7 *
## poly(gni_per_cap, 8, raw = TRUE)8 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05317 on 178 degrees of freedom
## Multiple R-squared:  0.9072, Adjusted R-squared:  0.903
## F-statistic: 217.6 on 8 and 178 DF,  p-value: < 2.2e-16
```

Based on the output above, it looks like we could drop the degree 8 term from this model. More specifically, we could conduct the following hypothesis test:

$H_0 : \beta_8 = 0$ vs. $H_A : \beta_8 \neq 0$

With a p-value of 0.061235, we fail to reject the null hypothesis at a signficance level of $\alpha = 0.05$.

I didn't ask you to do this, but it might be nice to check out the degree 7 model fit as well:

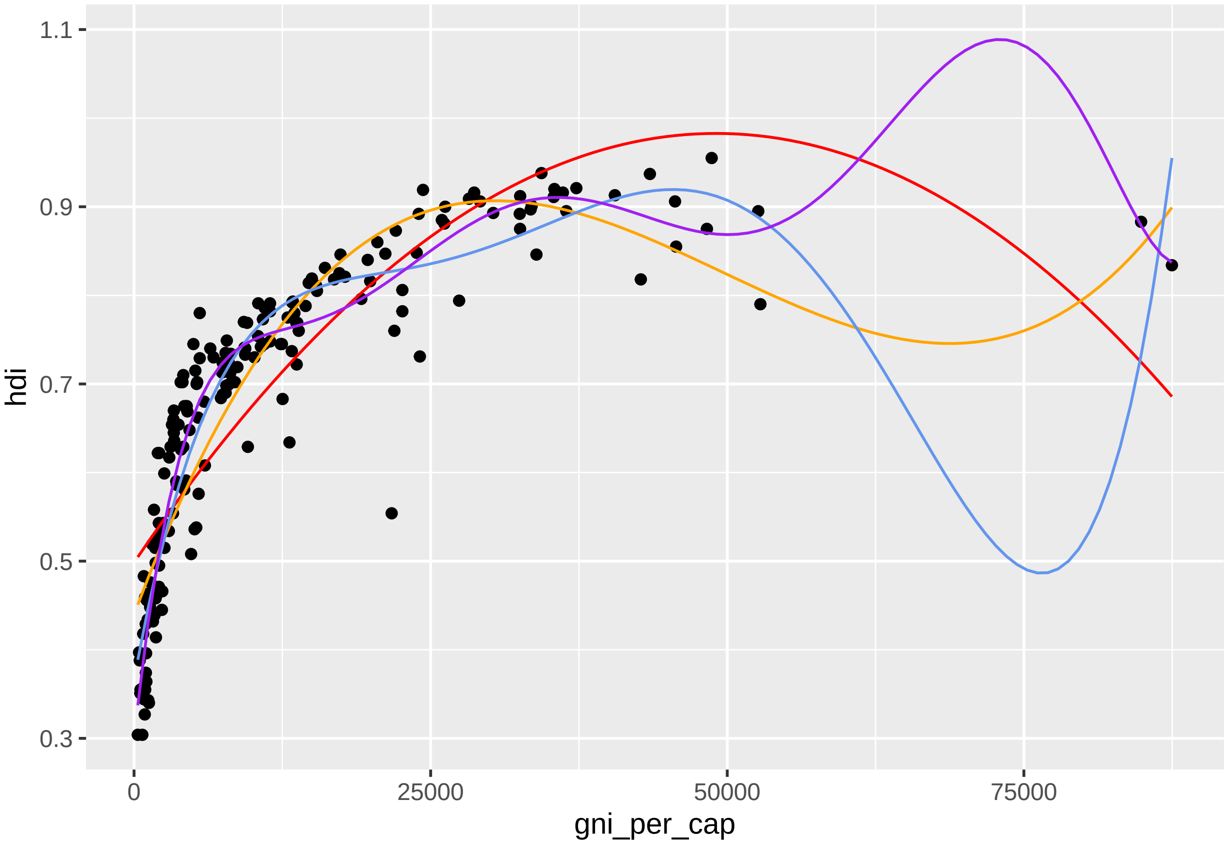
```
fit7 <- lm(hdi ~ poly(gni_per_cap, 7), data = country_data)
summary(fit7)
```

```
##
## Call:
## lm(formula = hdi ~ poly(gni_per_cap, 7), data = country_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.264274 -0.029135  0.004283  0.034270  0.120305
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.674904   0.003916 172.364  < 2e-16 ***
## poly(gni_per_cap, 7)1  1.678176   0.053545  31.342  < 2e-16 ***
## poly(gni_per_cap, 7)2 -1.090602   0.053545 -20.368  < 2e-16 ***
## poly(gni_per_cap, 7)3  0.693178   0.053545  12.946  < 2e-16 ***
## poly(gni_per_cap, 7)4 -0.403564   0.053545  -7.537 2.31e-12 ***
## poly(gni_per_cap, 7)5  0.333508   0.053545   6.229 3.27e-09 ***
## poly(gni_per_cap, 7)6 -0.347313   0.053545  -6.486 8.31e-10 ***
## poly(gni_per_cap, 7)7  0.170048   0.053545   3.176  0.00176 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05354 on 179 degrees of freedom
## Multiple R-squared:  0.9054, Adjusted R-squared:  0.9017
## F-statistic: 244.6 on 7 and 179 DF,  p-value: < 2.2e-16
```

Based on the hypothesis tests, it looks like we could use a degree 7 polynomial for these data. Of course, in reality we should always look at some plots too - see the next part.

(b) Create four additional plots (or if you prefer, one additional plot with four curves on it) displaying polynomial fits of degrees 2, 3, 5, and 7 overlaid on the scatterplot(s). Do any of these models provide an adequate summary of the data? If I found data for a few additional countries, would you have confidence that the predicted `hdl` from the best polynomial model from part (a) would be similar to the observed values of `hdl` in my new sample?

```r
pred2 <- function(x) {
  fit <- lm(hdi ~ poly(gni_per_cap, degree = 2, raw = TRUE), data = country_data)
  predict(fit, newdata = data.frame(gni_per_cap = x))
}
pred3 <- function(x) {
  fit <- lm(hdi ~ poly(gni_per_cap, degree = 3, raw = TRUE), data = country_data)
  predict(fit, newdata = data.frame(gni_per_cap = x))
}
pred5 <- function(x) {
  fit <- lm(hdi ~ poly(gni_per_cap, degree = 5, raw = TRUE), data = country_data)
  predict(fit, newdata = data.frame(gni_per_cap = x))
}
pred7 <- function(x) {
  fit <- lm(hdi ~ poly(gni_per_cap, degree = 7, raw = TRUE), data = country_data)
  predict(fit, newdata = data.frame(gni_per_cap = x))
}
ggplot(data = country_data, mapping = aes(x = gni_per_cap, y = hdi)) +
  geom_point() +
  stat_function(fun = pred2, color = "red") +
  stat_function(fun = pred3, color = "orange") +
  stat_function(fun = pred5, color = "cornflowerblue") +
  stat_function(fun = pred7, color = "purple")
```



None of these polynomial models seem very good. In particular, the degree 7 polynomial (plotted in purple) that was selected by the hypothesis tests above seems much too wiggly and is definitely overfitting the two data points with large values of `gni_per_cap`, and makes unreasonable predictions in the range of 50,000 to 80,000. The predicted value for a new observation might be ok for values of GNI per capita less than about 37500, but I would not trust the predictions from any of these models along the full range of values of GNI per capita.

Collaboration and Sources

If you worked with any other students on this assignment, please list their names here.

If you referred to any sources (including our text book), please list them here. No need to get into formal citation formats, just list the name of the book(s) you used or provide a link to any online resources you used.