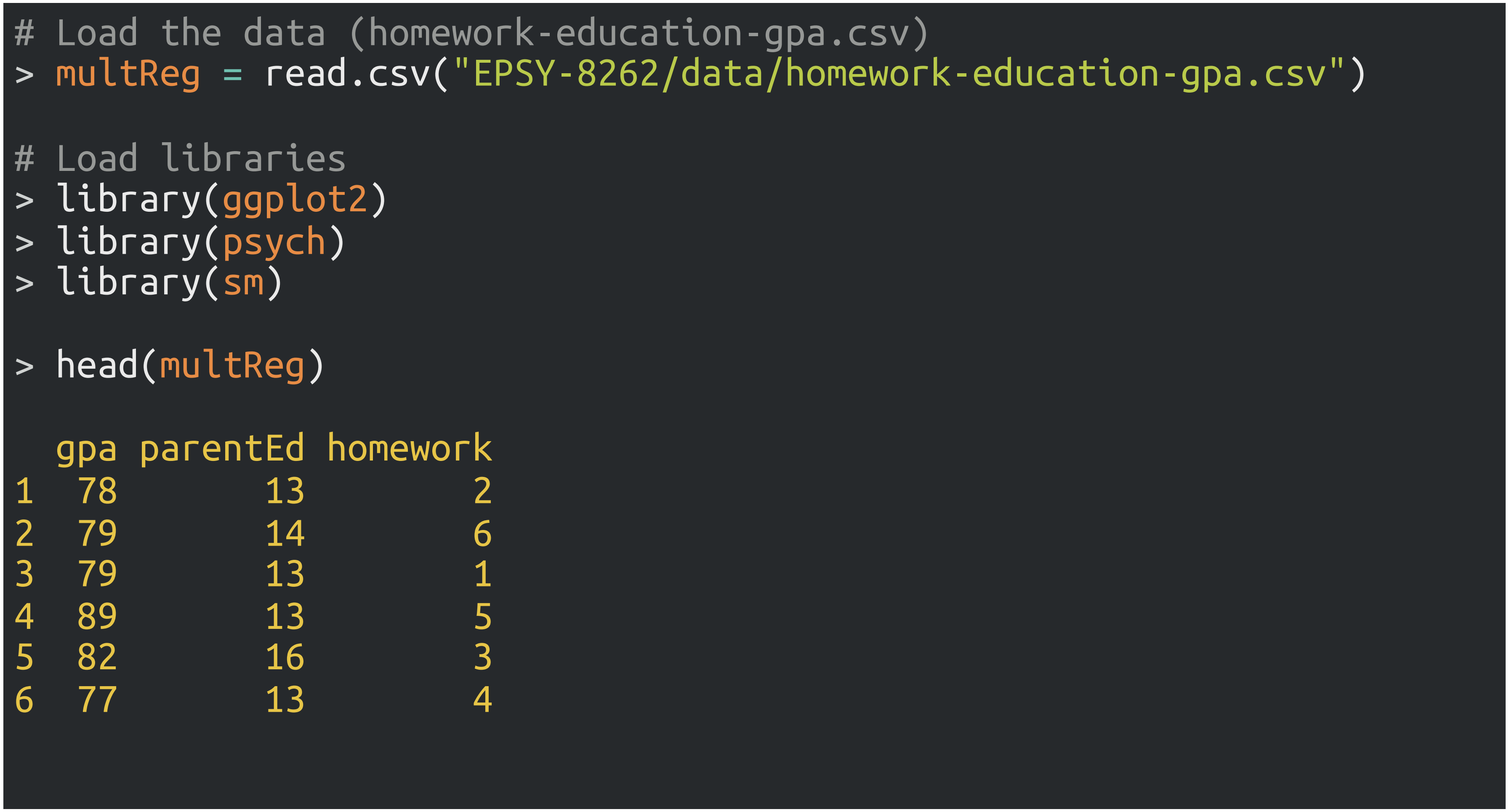
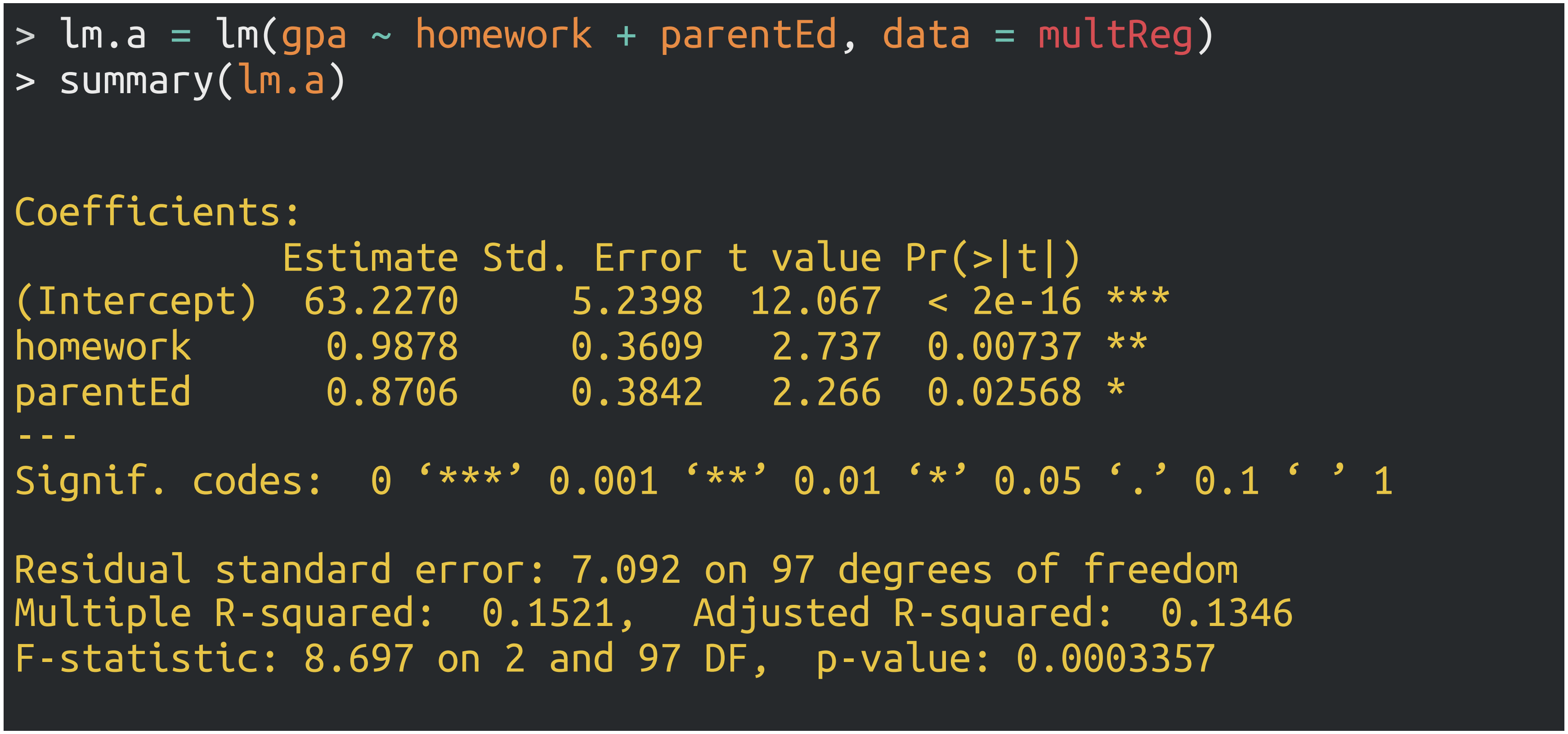
Presenting Multiple Regression Results

Read in Data and Load Libraries



```
# Load the data (homework-education-gpa.csv)
> multReg = read.csv("EPSY-8262/data/homework-education-gpa.csv")

# Load libraries
> library(ggplot2)
> library(psych)
> library(sm)

> head(multReg)

  gpa parentEd homework
1  78       13        2
2  79       14        6
3  79       13        1
4  89       13        5
5  82       16        3
6  77       13        4
```

Fitting the Multiple Regression Model Using R



```
> lm.a = lm(gpa ~ homework + parentEd, data = multReg)
> summary(lm.a)


Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  63.2270     5.2398  12.067  < 2e-16 ***
homework      0.9878     0.3609   2.737  0.00737 **
parentEd      0.8706     0.3842   2.266  0.02568 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.092 on 97 degrees of freedom
Multiple R-squared:  0.1521,   Adjusted R-squared:  0.1346
F-statistic: 8.697 on 2 and 97 DF,  p-value: 0.0003357
```

$$\hat{\text{GPA}} = 63.23 + 0.99(\text{homework}) + 0.87(\text{parentEd})$$

Predictions

$$\mathrm{G\hat{P}A} = 63.23 + 0.99(\mathrm{homework}) + 0.87(\mathrm{parentEd})$$

Let's predict the GPA for three students who spend differing amounts of time on homework,

HW = 1
HW = 2
HW = 3

Let's assume that these students all have a parentEd value of 12 (years of schooling for the parent with the most education is 12 years)

homework	ParentEd	Predicted GPA
1	12	$63.23 + 0.99(1) + 0.87(12) = 74.66$
2	12	$63.23 + 0.99(2) + 0.87(12) = 75.65$
3	12	$63.23 + 0.99(3) + 0.87(12) = 76.64$

$$\hat{\text{GPA}} = 63.23 + 0.99(\text{homework}) + 0.87(\text{parentEd})$$

homework		ParentEd	Predicted GPA	
1	+1	12	63.23 + 0.99(1) + 0.87(12) = 74.66	+0.99
2	+1	12	63.23 + 0.99(2) + 0.87(12) = 75.65	+0.99
3		12	63.23 + 0.99(3) + 0.87(12) = 76.64	

A one-hour difference in the time students spend on homework is associated with a 0.99-unit difference in GPA…**controlling for** differences in parent education by **holding that value constant**.

homework		ParentEd	Predicted GPA	
1	+1	13	63.23 + 0.99(1) + 0.87(13) = 75.53	+0.99
2	+1	13	63.23 + 0.99(2) + 0.87(13) = 76.52	+0.99
3		13	63.23 + 0.99(3) + 0.87(13) = 77.51	

This will be true, regardless of which value we pick for parentEd.

$$\hat{\text{GPA}} = 63.23 + 0.99(\text{homework}) + 0.87(\text{parentEd})$$

What happens if we pick a fixed value for homework and look at how the predicted GPA varies for different values of parentEd?

homework	ParentEd		Predicted GPA	
5	13			
		+1		?
5	14			
		+1		?
5	15			

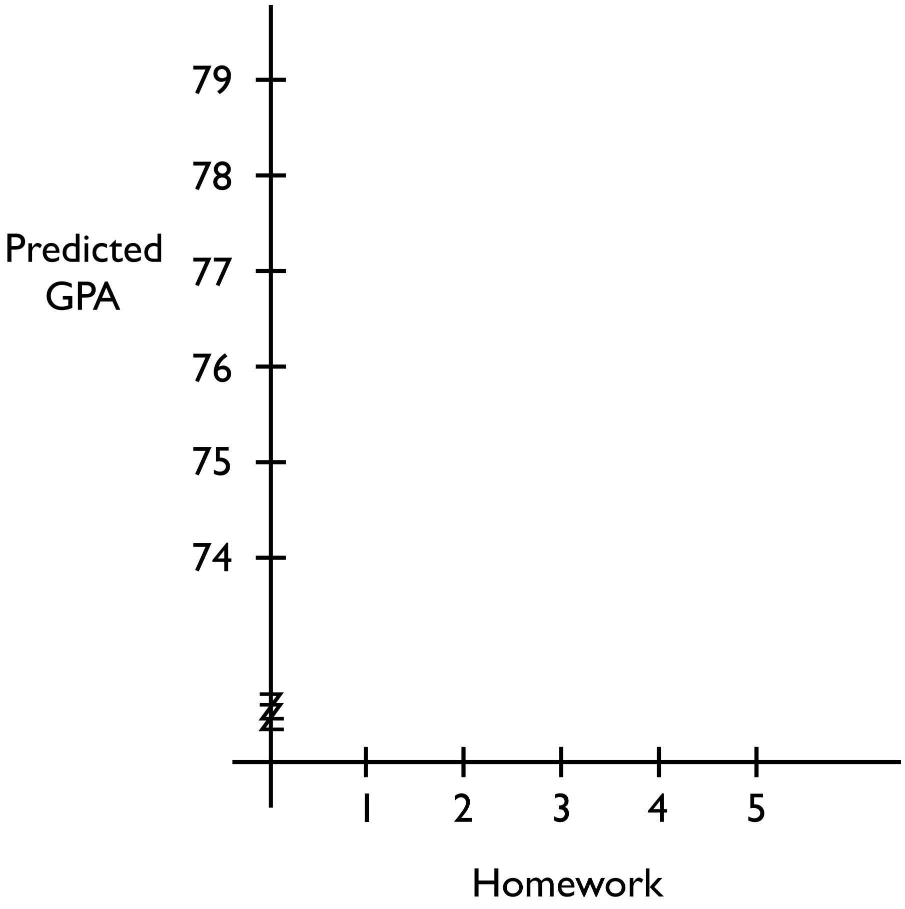
Try it and convince yourself!

homework	ParentEd	Predicted GPA
1	12	74.66
2	12	75.65
3	12	76.64

homework	ParentEd	Predicted GPA
1	13	75.53
2	13	76.52
3	13	77.51

Sketch the scatterplot to display the ordered pairs (*homework, predicted GPA*) for those students whose parentEd value is 12.

Add the ordered pairs (*homework, predicted GPA*) for those students whose parentEd value is 16 to the plot, but use a different symbol.

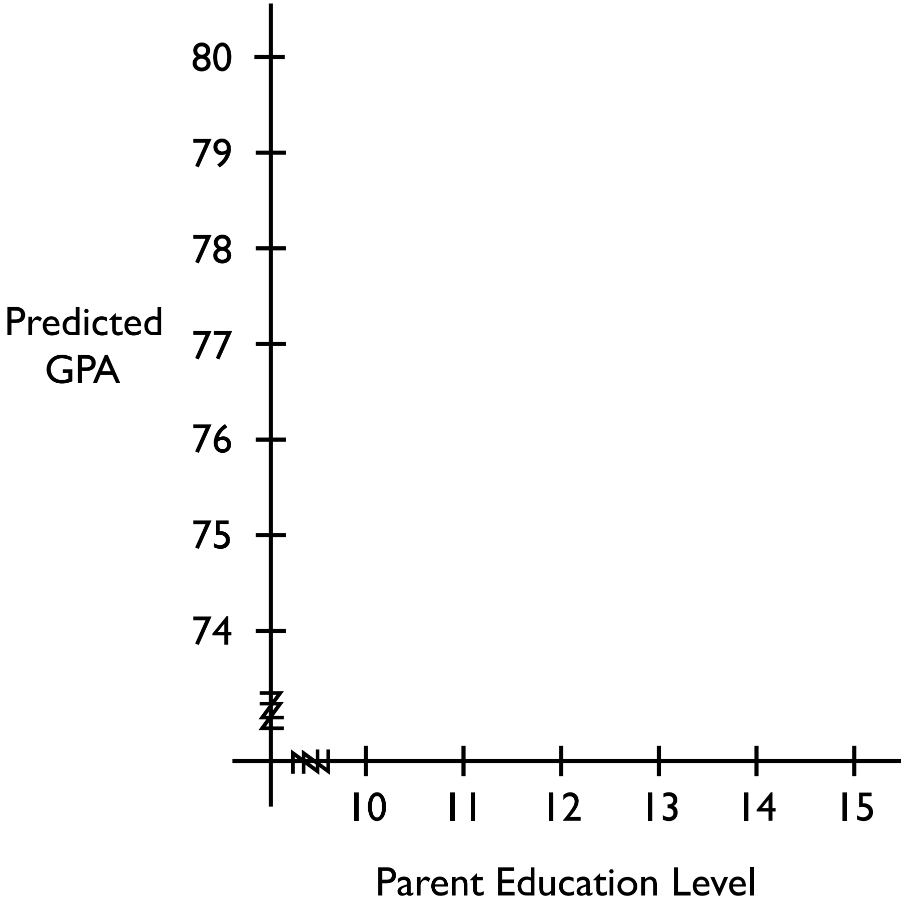


homework	ParentEd	Predicted GPA
1	12	74.66
1	13	75.53
1	14	76.40

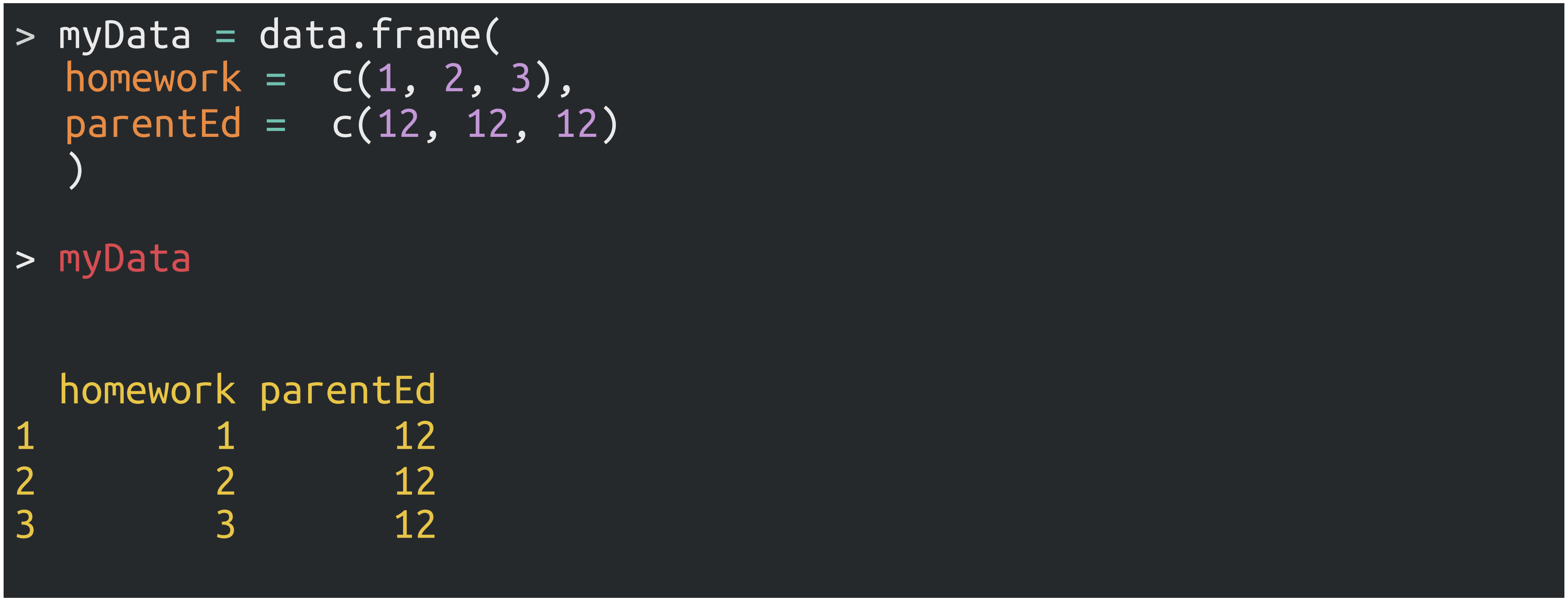
homework	ParentEd	Predicted GPA
5	12	78.61
5	13	79.48
5	14	80.35

Sketch the scatterplot to display the ordered pairs (*parentEd, predicted GPA*) for those students whose homework value is 1.

Add the ordered pairs (*parentEd, predicted GPA*) for those students whose homework value is 5 to the plot, but use a different symbol.



Using R to get Predictions

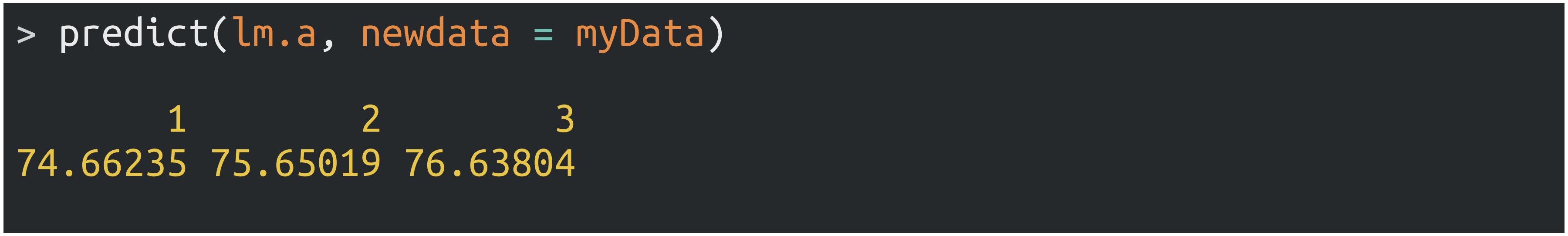


```
> myData = data.frame(
   homework =  c(1, 2, 3),
   parentEd =  c(12, 12, 12)
   )

> myData


  homework parentEd
1        1       12
2        2       12
3        3       12
```

We create a new data frame from which we are going to predict GPAs. The variables in this data frame need to have the exact same names as the predictors in your `lm()` model.

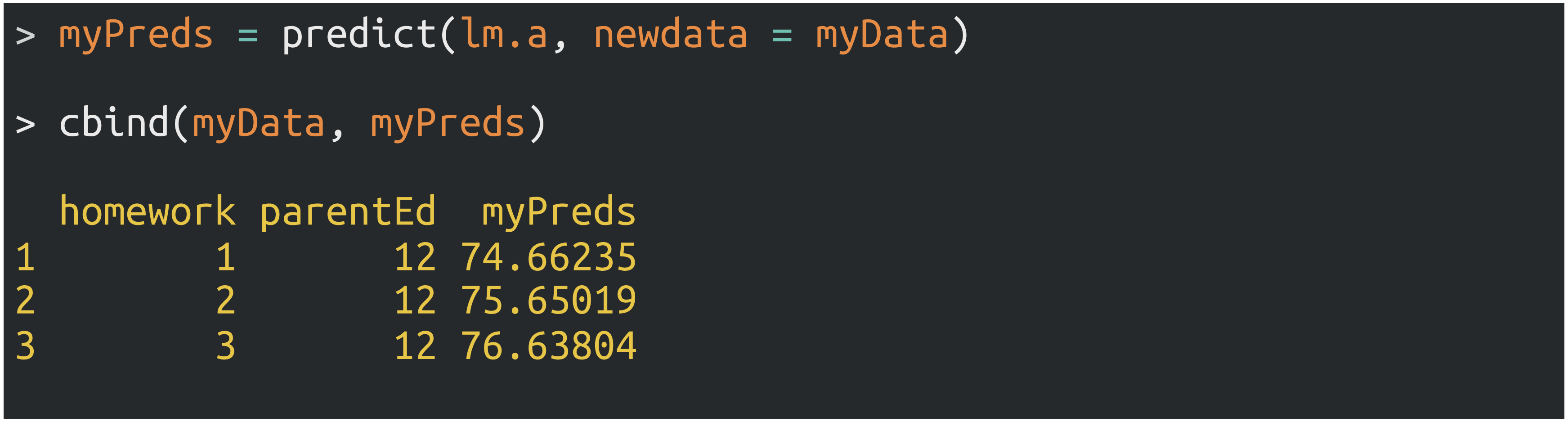
```
lm(gpa ~ homework + parentEd)
```

Use the `predict()` function to obtain predictions. This function takes the name of the fitted model and the argument `newdata=` which gives the name of the data frame from which we are predicting.



```
> predict(lm.a, newdata = myData)

       1        2        3 
74.66235 75.65019 76.63804 
```

Here we append the predictions to the original data frame to make it more readable.



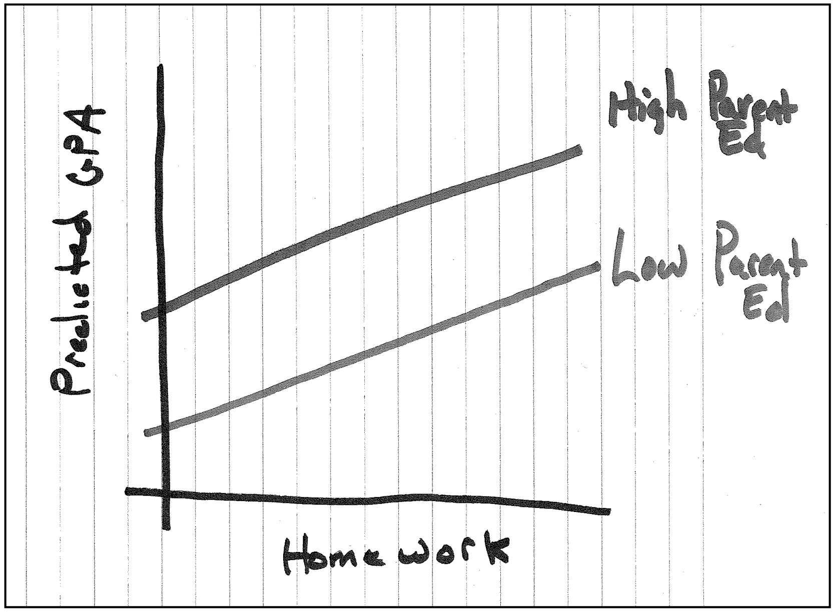
```
> myPreds = predict(lm.a, newdata = myData)

> cbind(myData, myPreds)

  homework parentEd  myPreds
1        1       12 74.66235
2        2       12 75.65019
3        3       12 76.63804
```

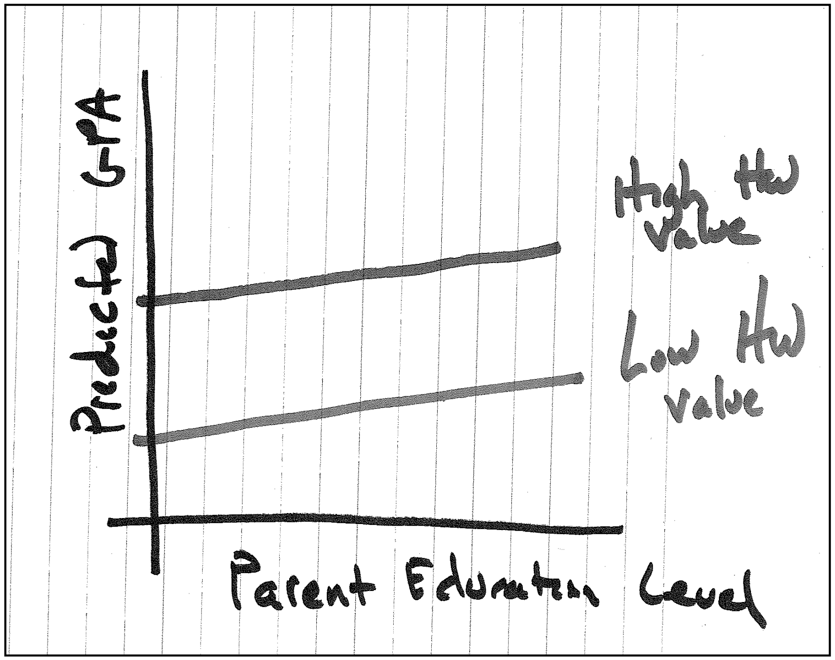
Try using R to compute the predicted values for GPA for the following students.

homework	ParentEd
5	13
5	14
5	15

Considering a Plot of the Results

$$\hat{\text{GPA}} = 63.23 + 0.99(\text{homework}) + 0.87(\text{parentEd})$$



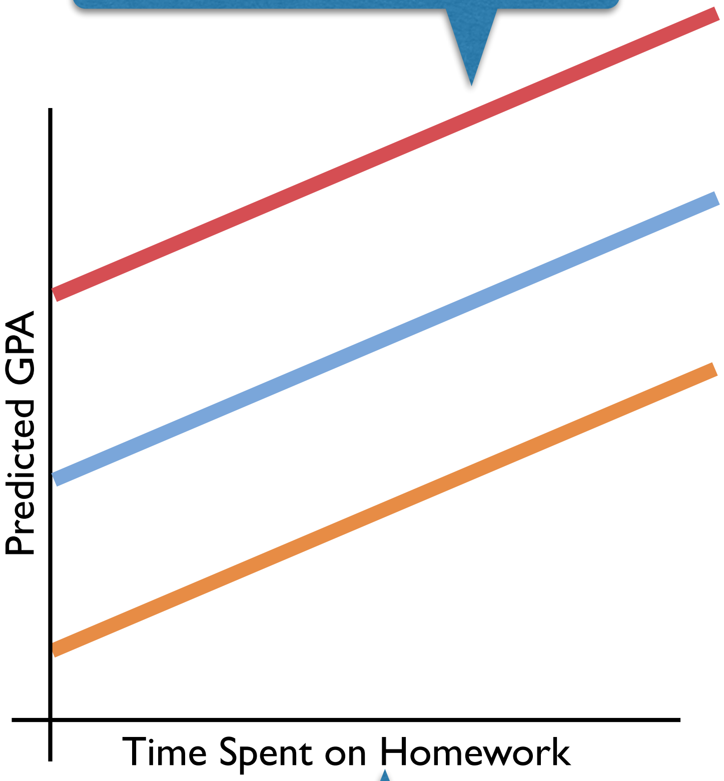


With two (or more) effects in the model we have multiple displays of the fitted lines that are possible.

- Which predictor do you want to display on the *x*-axis?
- How many levels of the remaining predictors do you want to display?

Planning the Plot

Any variable we are showing via different lines (parentEd) will have discrete values.



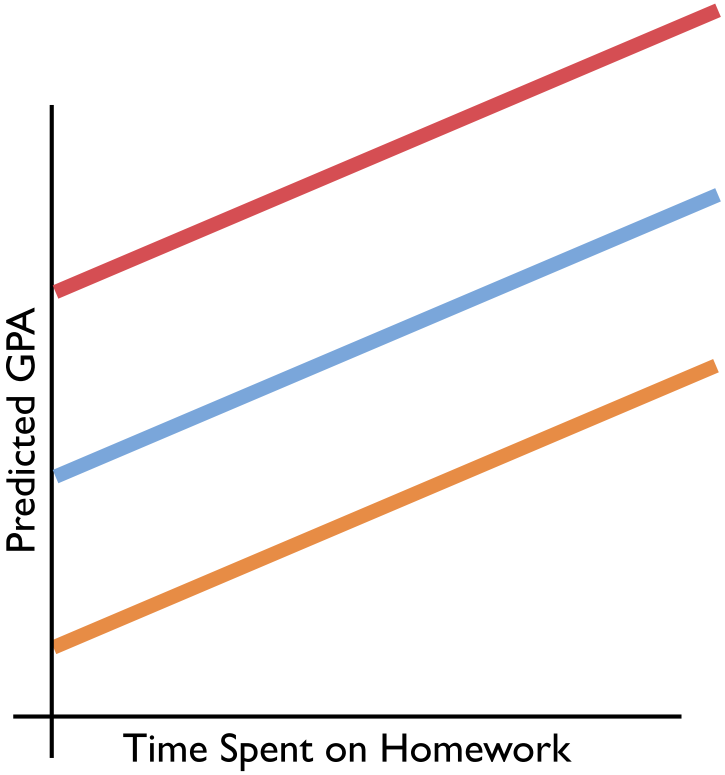
The variable on the *x*-axis (homework) will be continuous.

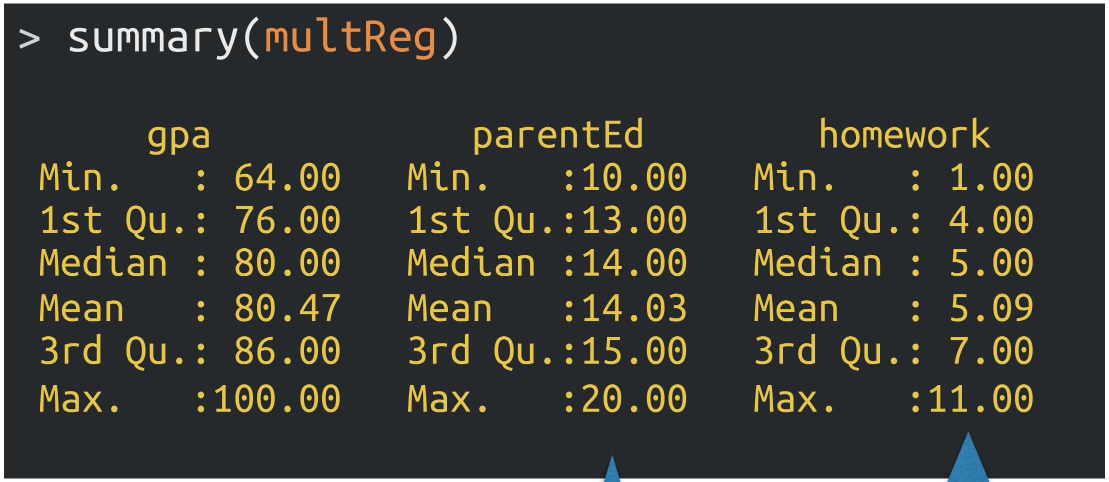
- Which predictor do you want to display on the *x*-axis?

  Homework

- How many levels of the remaining predictors do you want to display?

  Let's show 3 different values of parentEd





```
> summary(multReg)

      gpa             parentEd         homework    
 Min.   : 64.00   Min.   :10.00   Min.   : 1.00  
 1st Qu.: 76.00   1st Qu.:13.00   1st Qu.: 4.00  
 Median : 80.00   Median :14.00   Median : 5.00  
 Mean   : 80.47   Mean   :14.03   Mean   : 5.09  
 3rd Qu.: 86.00   3rd Qu.:15.00   3rd Qu.: 7.00  
 Max.   :100.00   Max.   :20.00   Max.   :11.00  
```

Values should be between 10 and 20, but should be interpretable.

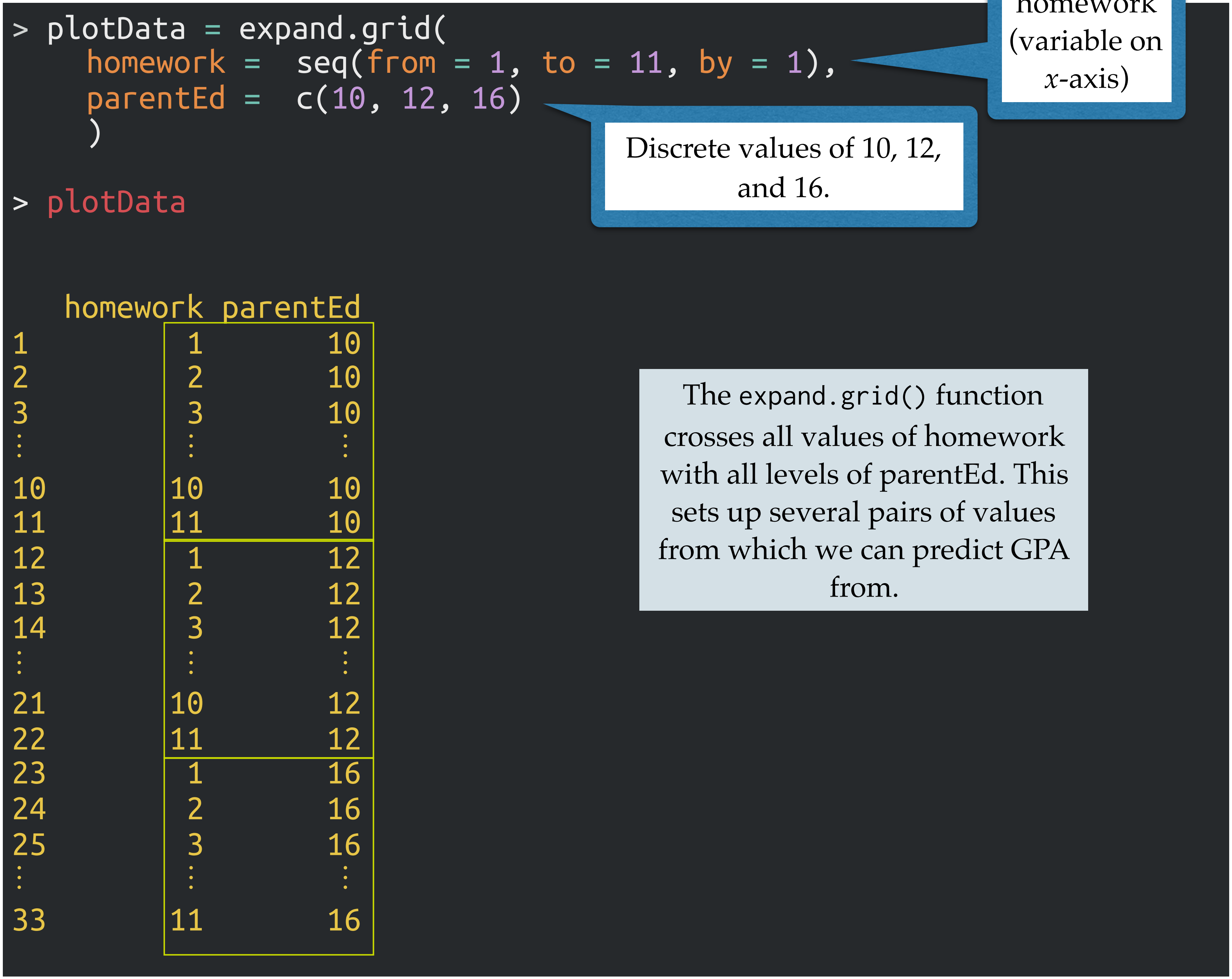
Range of homework (variable on *x*-axis)

- What **range of values** should we use for the predictor on the *x*-axis?

  0-11

- Which discrete values should we choose for the the remaining predictors?

  10 (some high school), 12 (high school), 16 (undergraduate)

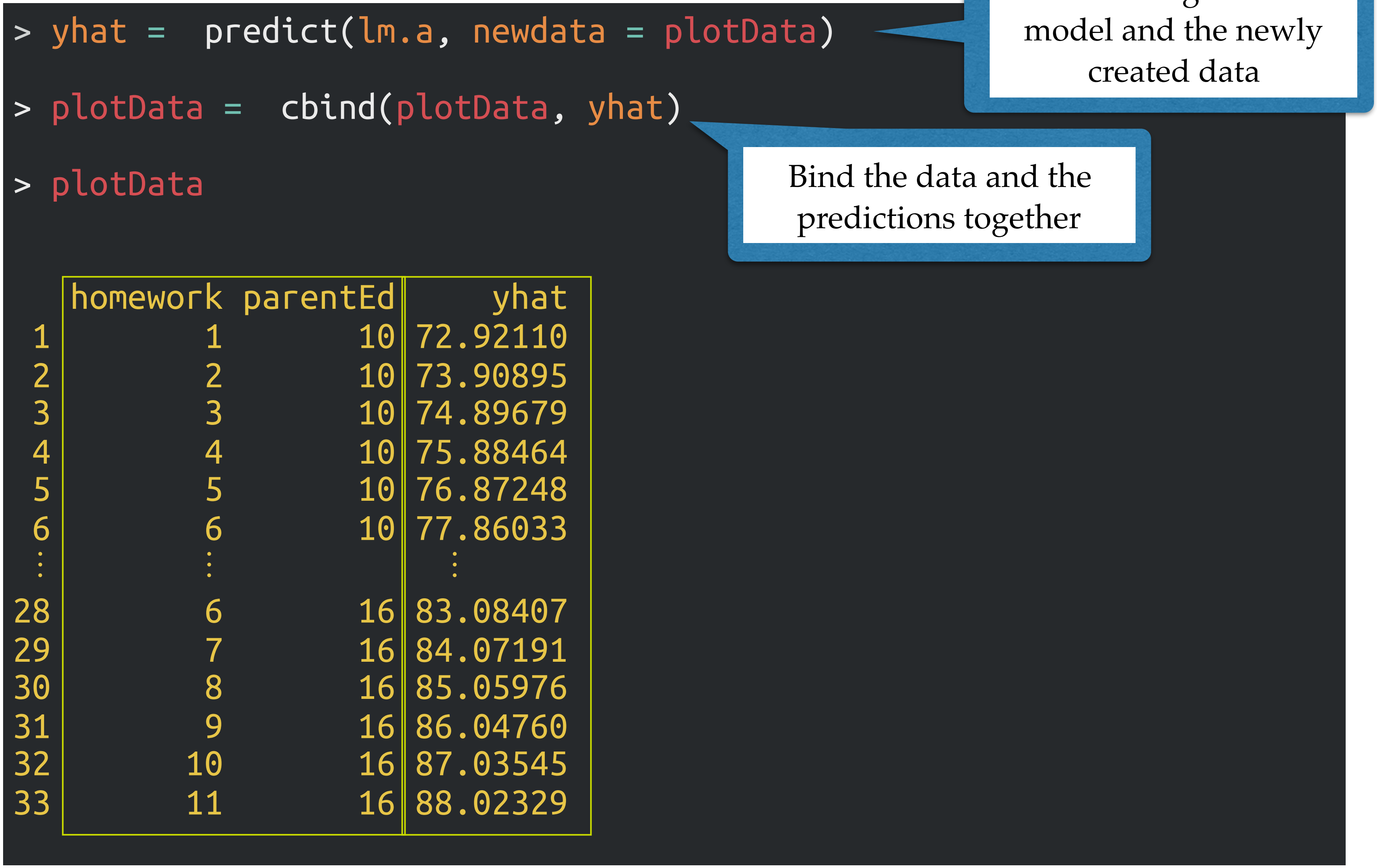


```
> plotData = expand.grid(
    homework =  seq(from = 1, to = 11, by = 1),
    parentEd =  c(10, 12, 16)
    )

> plotData


   homework  parentEd
1         1        10
2         2        10
3         3        10
⋮         ⋮         ⋮
10       10        10
11       11        10
12        1        12
13        2        12
14        3        12
⋮         ⋮         ⋮
21       10        12
22       11        12
23        1        16
24        2        16
25        3        16
⋮         ⋮         ⋮
33       11        16
```

Range of homework (variable on *x*-axis)

Discrete values of 10, 12, and 16.

The `expand.grid()` function crosses all values of homework with all levels of parentEd. This sets up several pairs of values from which we can predict GPA from.

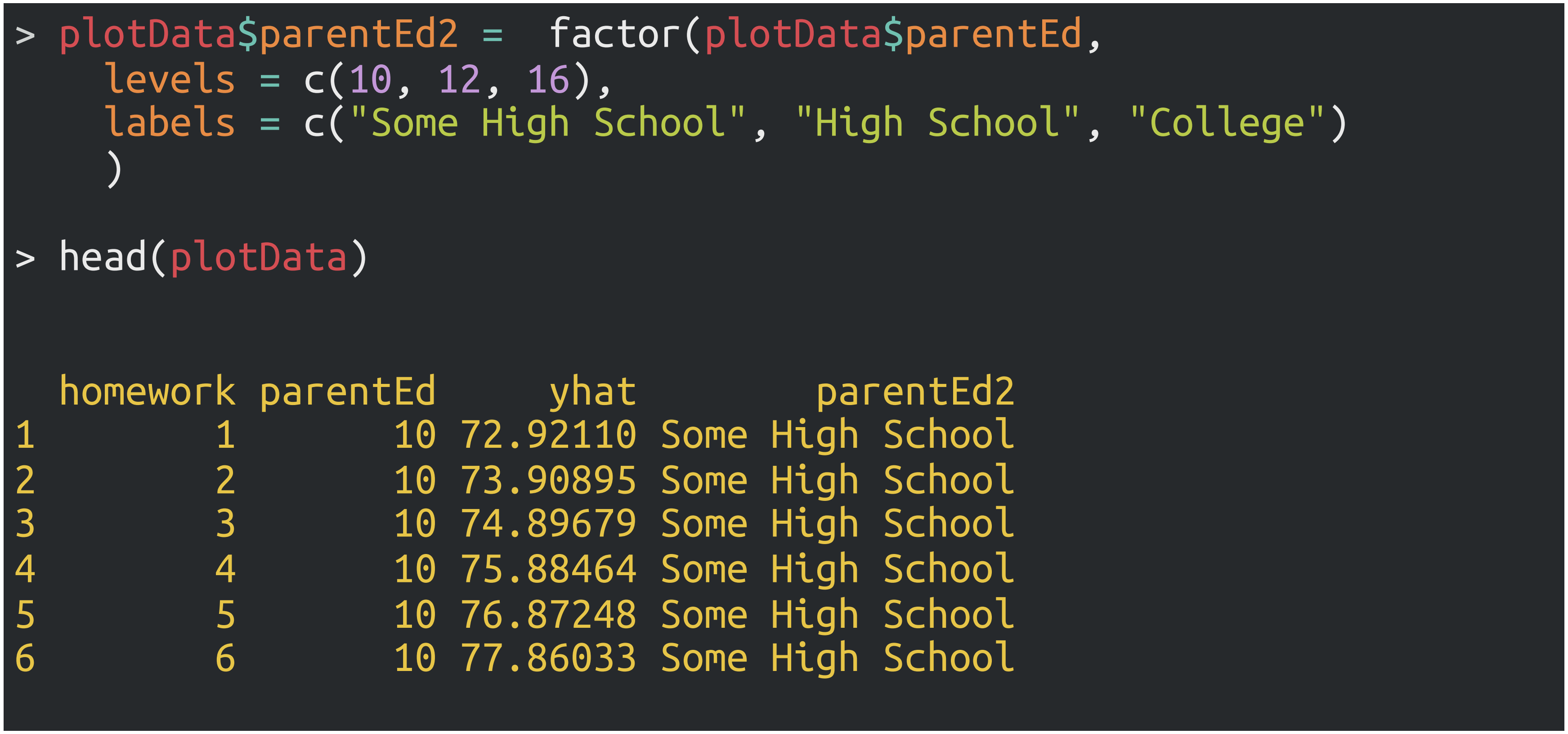


```
> yhat =  predict(lm.a, newdata = plotData)

> plotData =  cbind(plotData, yhat)

> plotData


   homework parentEd     yhat
 1        1       10 72.92110
 2        2       10 73.90895
 3        3       10 74.89679
 4        4       10 75.88464
 5        5       10 76.87248
 6        6       10 77.86033
 ⋮        ⋮                ⋮
28        6       16 83.08407
29        7       16 84.07191
30        8       16 85.05976
31        9       16 86.04760
32       10       16 87.03545
33       11       16 88.02329
```

Predict using the fitted model and the newly created data

Bind the data and the predictions together

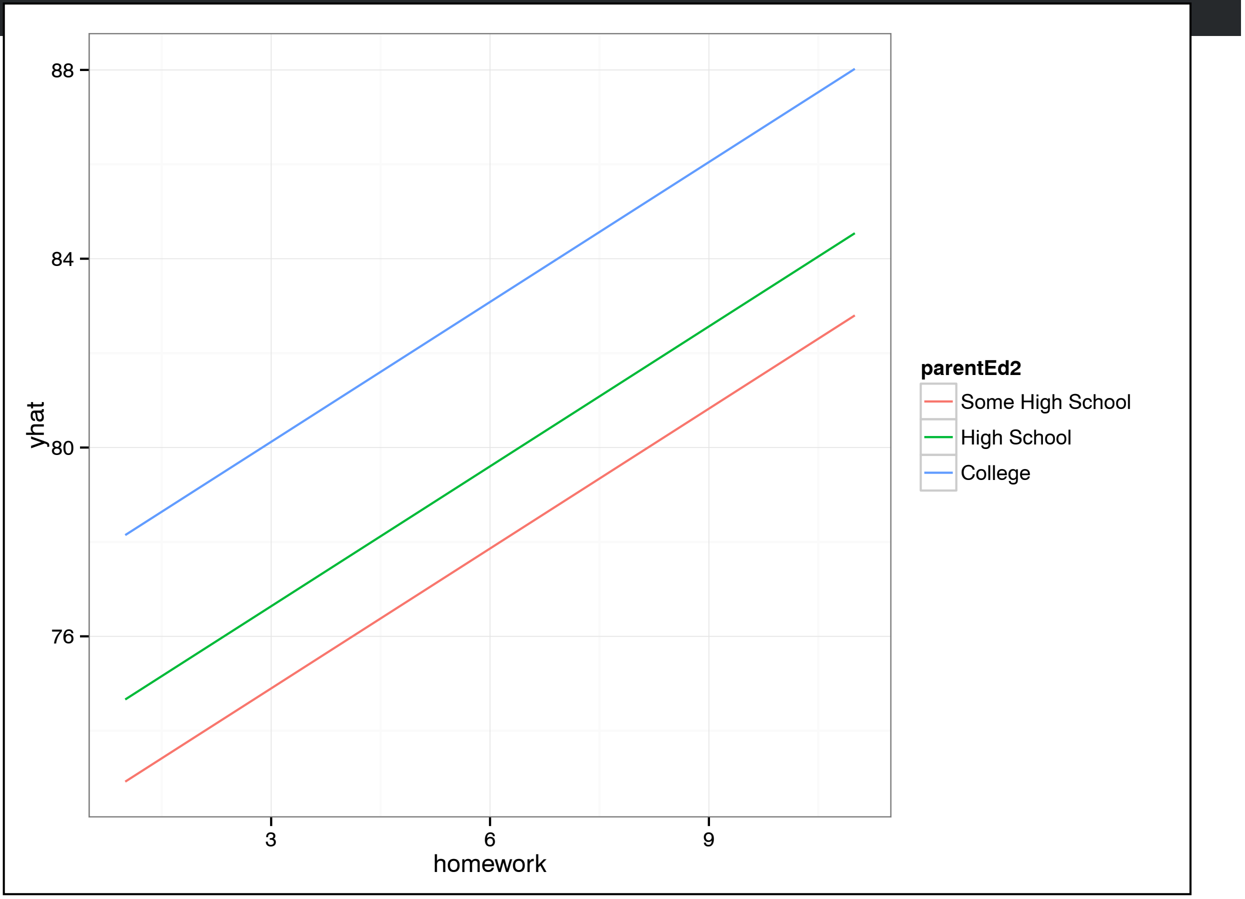
After predicting, we can coerce any variable with discrete values (parentEd) into a factor. This will help auto-create a legend when we plot it later on.

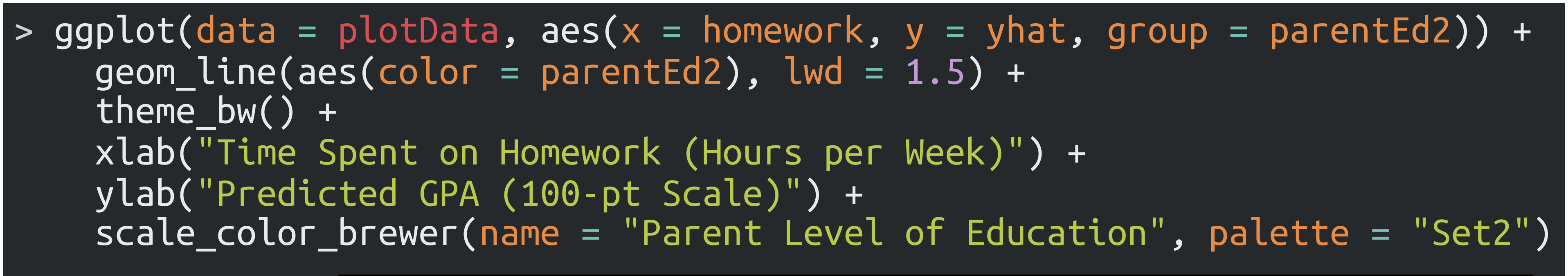


```
> plotData$parentEd2 =  factor(plotData$parentEd,
    levels = c(10, 12, 16),
    labels = c("Some High School", "High School", "College")
    )

> head(plotData)


  homework parentEd     yhat        parentEd2
1        1       10 72.92110 Some High School
2        2       10 73.90895 Some High School
3        3       10 74.89679 Some High School
4        4       10 75.88464 Some High School
5        5       10 76.87248 Some High School
6        6       10 77.86033 Some High School
```

Here we use the group= aesthetic to draw different lines for each value of parentEd2. We then use color= to color the lines different colors.

```
> ggplot(data = plotData, aes(x = homework, y = yhat, group = parentEd2)) +
     geom_line(aes(color = parentEd2)) +
     theme_bw()
```





```
> ggplot(data = plotData, aes(x = homework, y = yhat, group = parentEd2)) +
    geom_line(aes(color = parentEd2), lwd = 1.5) +
    theme_bw() +
    xlab("Time Spent on Homework (Hours per Week)") +
    ylab("Predicted GPA (100-pt Scale)") +
    scale_color_brewer(name = "Parent Level of Education", palette = "Set2")
```

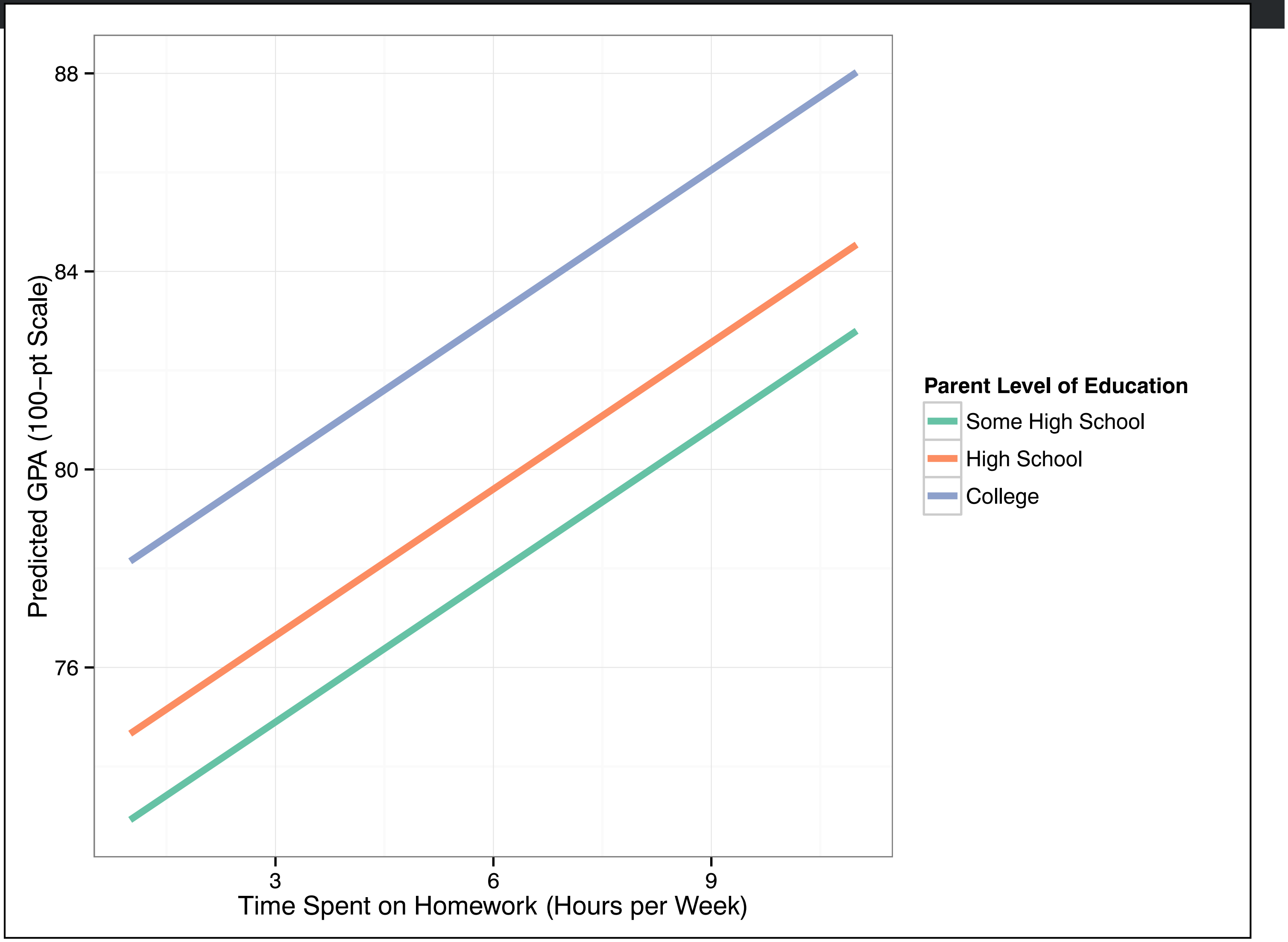


Figure 1. *Predicted GPA as a function of parent level of education for students who spend one and ten hours a week on homework.*

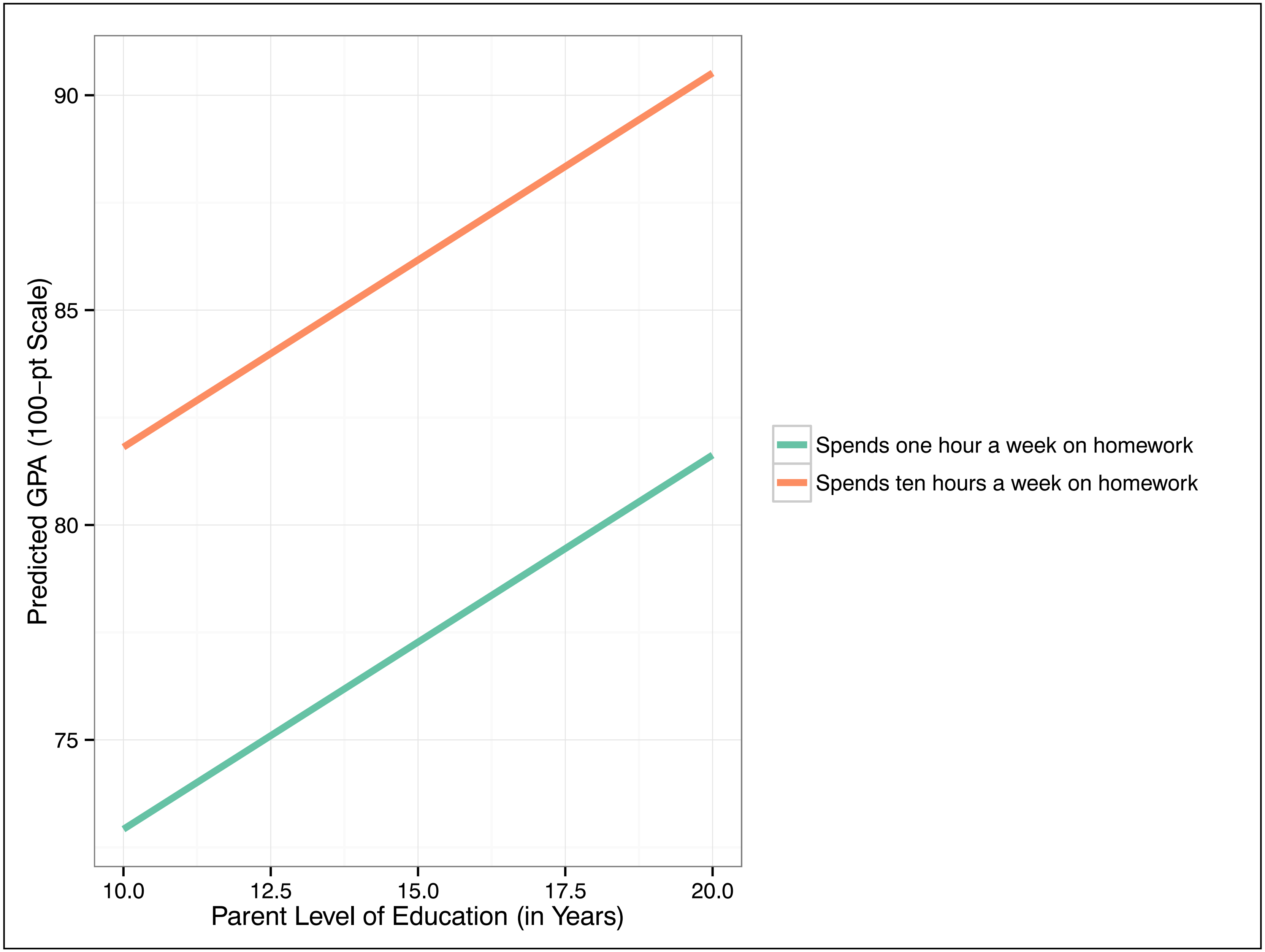
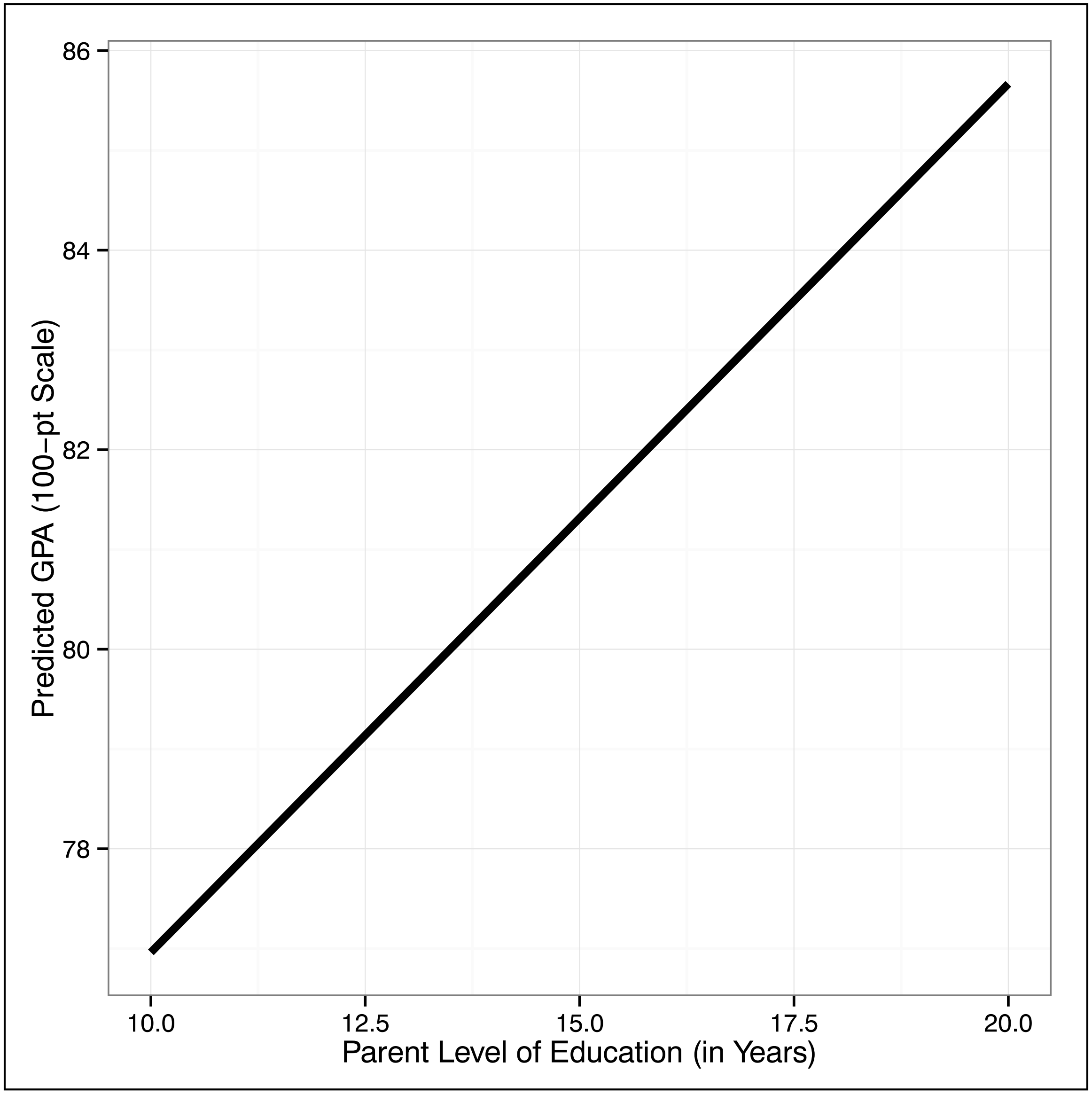


Figure 1. *Predicted GPA as a function of parent level of education. Time spent on homework is patrolled out of the model by fixing this variable to its mean value of 5.09.*


Predicted GPA (100−pt Scale)
Parent Level of Education (in Years)