Assignment 1

ANSWER KEY

2023-10-13

Questions

1. Using the `vtable` package, create a table of summary statistics from the `econmath` data that includes the mean, standard deviation, minimum, and maximum for variables: *score*, *hsgpa*, *study*, *age*.

```
econmath %>%
  select(score, hsgpa, study, age) %>%
  sumtable(summ=c('mean(x)','sd(x)','min(x)','max(x)'))
```

2. Compute the summary statistics table as you did in (1), but group the data by whether or not the student took a high school economics course. Comment on the differences across groups in the mean of these variables.

There are some differences, but they are very minor especially in relation to the standard deviations of these variables. So it appears that the students who took economics in high school are essentially identical to those who did not.

```
econmath %>%
  select(score, hsgpa, study, age, econhs) %>%
  sumtable(summ=c('mean(x)','sd(x)','min(x)','max(x)'),
```

Table 1: Summary Statistics

Variable	Mean	Sd	Min	Max
score	73	13	20	98
hsgpa	3.3	0.34	2.4	4.3
study	14	7.8	0	50
age	19	0.94	18	29

Table 2: Summary Statistics

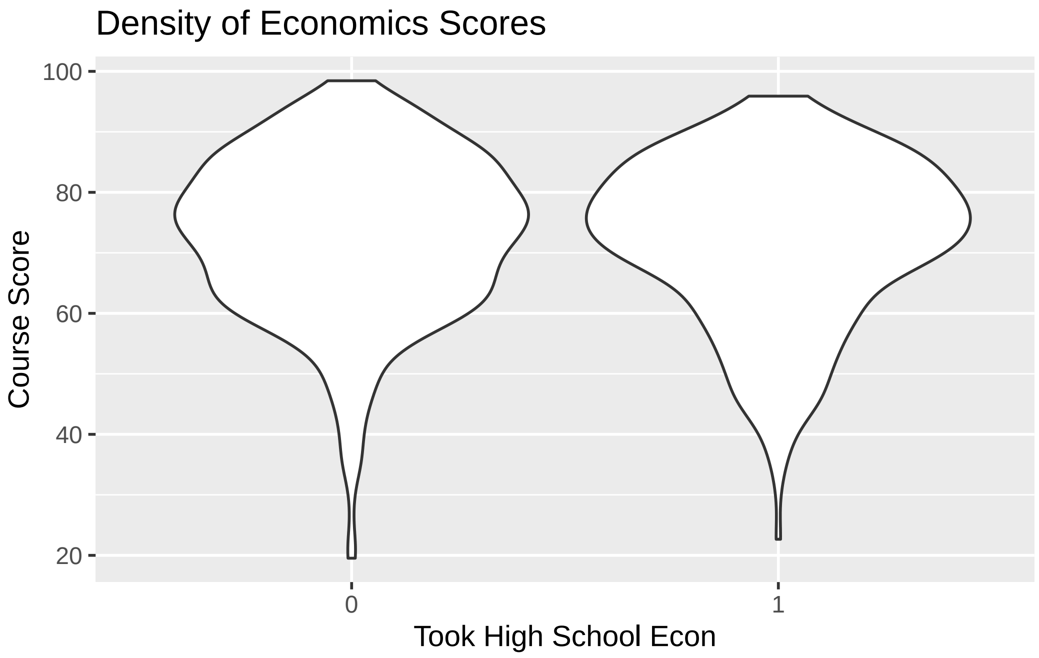
econhs	0				1			
Variable	Mean	Sd	Min	Max	Mean	Sd	Min	Max
score	73	13	20	98	72	13	23	96
hsgpa	3.3	0.34	2.4	4.1	3.4	0.35	2.4	4.3
study	14	7.7	0	48	14	8	0	50
age	19	0.94	18	29	19	0.94	18	28

```
             group = 'econhs')
```

3. Using the `ggplot2` package, produce a violin plot of *score* across the two values of *econhs* (note, you may need to look up violin plots to familiarize yourself with them). Create a title for the graph and relabel the x and y axes with more intuitive names. Describe the relationship between these two variables. [NOTE: when you define the aesthetics in your plot, you will need to declare *econhs* as a factor variable using `as.factor(econhs)`]

A violin plot is an alternative to a boxplot that shows the distribution of a variable across different groups. In this case, it plots the distribution of score based on whether the students have taken economics in high school or not. What you can see from this plot is that the means are mostly in line (as we knew from the table in the previous question), but the distribution for those who have taken economics in high school is more concentrated around the middle, and has a longer tail on the low end. This means that for the most part they are academically more similar to each other with the exception of a few low performers.

```
ggplot(econmath, aes(y = score, x = as.factor(econhs))) +
  geom_violin() +
  labs(title = "Density of Economics Scores",
       x = "Took High School Econ", y = "Course Score")
```

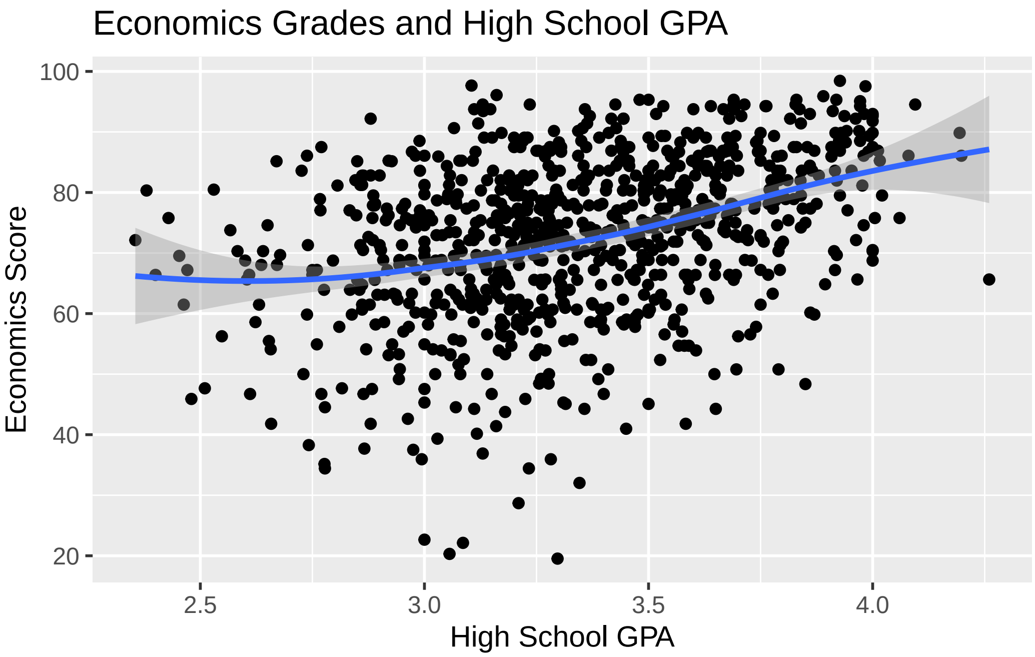


4. Using the `ggplot2` package, produce a scatterplot with *score* on the y-axis and *hsgpa* on the x-axis. Layer on top of that a **loess** regression line (again, look up what a loess function is). Create a title for the graph and relabel the x and y axes with more intuitive names. Describe the relationship between these two variables.

The loess (or lowess) is a non-parametric, data-driven technique that produces regression lines based on a local weighted average of the data near each value of x. When you connect them together you get the regression line plotted here. In this case it shows that at low values of high school GPA the two variables are unrelated, but at higher values they are positively related - a non-linear relationship.

```
p<-ggplot(econmath, aes(y = score, x = hsgpa)) +
  geom_point() +
  geom_smooth(method = "loess") +
  labs(title = "Economics Grades and High School GPA",
       x = "High School GPA", y = "Economics Score")
p
```

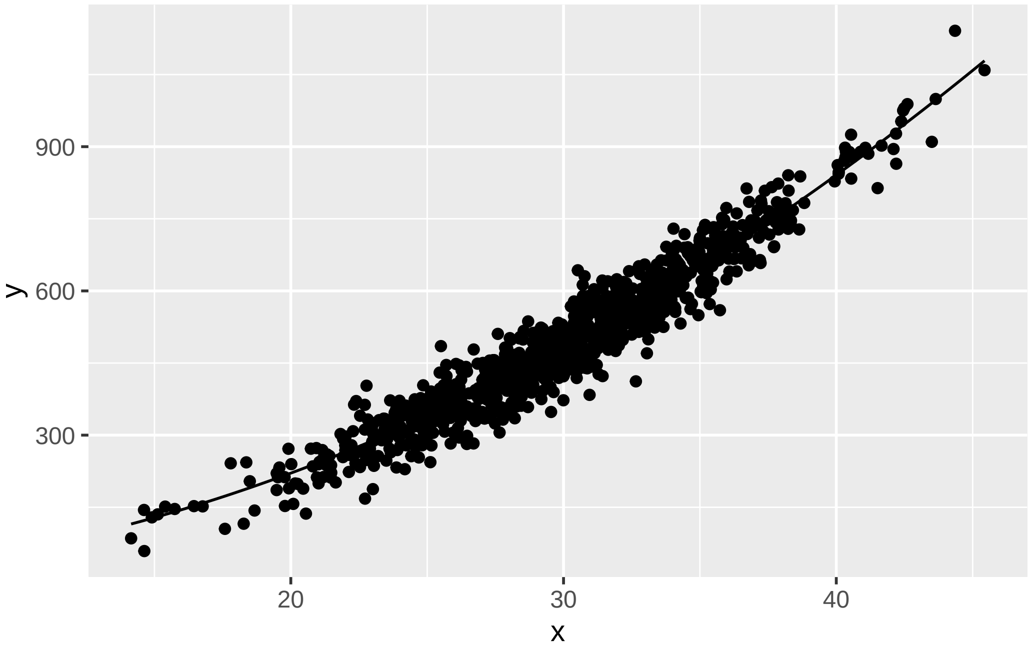
```
`geom_smooth()` using formula = 'y ~ x'
```



5. Suppose that the process that generates a set of data is $y = 1 + x + 0.5 * x^2 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 40^2)$, and $x \sim \mathcal{N}(30, 5^2)$. This means that the Conditional Expectation Function (CEF) is $E[y|x] = 1 + x + 0.5 * x^2$. The code below creates the data for x and y. Plot the conditional expectation function on top of a scatterplot of the data.

```
data <- tibble(x = rnorm(1000,30,5),
               y = 1 + x + 0.5*x^2 + rnorm(1000,0,40))

ggplot(data, aes(x = x, y = y)) +
  geom_point() +
  geom_function(fun = function(x) 1 + x + 0.5*x^2)
```



6. Suppose you are interested in the Population Regression of y on x as an approximation of the CEF. Compute the population regression slope and intercept. A useful piece of information for this question is that for a Normal random variable x, the covariance between x and x^2 is $(E[x])^3 + 3E[x]Var[x] - E[x]((E[x])^2 + Var[x])$.

The slope in a bivariate population regression is

$$\beta = \frac{cov(x,y)}{var(x)}$$

Substitute the definition of y from the question

$$\beta = \frac{cov(x, 1 + x + 0.5x^2)}{var(x)}$$

$$= \frac{cov(x,1)}{var(x)} + \frac{cov(x,x)}{var(x)} + \frac{0.5cov(x,x^2)}{var(x)}$$

$$= 1 + \frac{0.5cov(x,x^2)}{var(x)}$$

As given in the question, the covariance between x and x^2 is $(E[x])^3 + 3E[x]Var[x] - E[x]((E[x])^2 + Var[x])$, so

$$\beta = 1 + \frac{0.5((E[x])^3 + 3E[x]Var[x] - E[x]((E[x])^2 + Var[x]))}{var(x)}$$

We know all the values in this function from the information given in the question. Substituting them in, we get

$$\beta = 1 + \frac{0.5(30)^3 + 3 * 30 * 25 - 30 * (30^2 + 25))}{25} = 31$$

The intercept is $\alpha = E[y] - \beta E[x]$

Again substituting in for y we get

$$\alpha = E[1 + x + 0.5x^2 + \epsilon] - \beta E[x]$$
$$\alpha = 1 + E[x] + 0.5E[x^2] - \beta E[x]$$

Using the fact that the variance of x is $var(x) = E[x^2] - (E[x])^2$, we can sub in for $E[x^2]$

$$\alpha = 1 + E[x] + 0.5(var(x) + (E[x])^2) - \beta E[x]$$

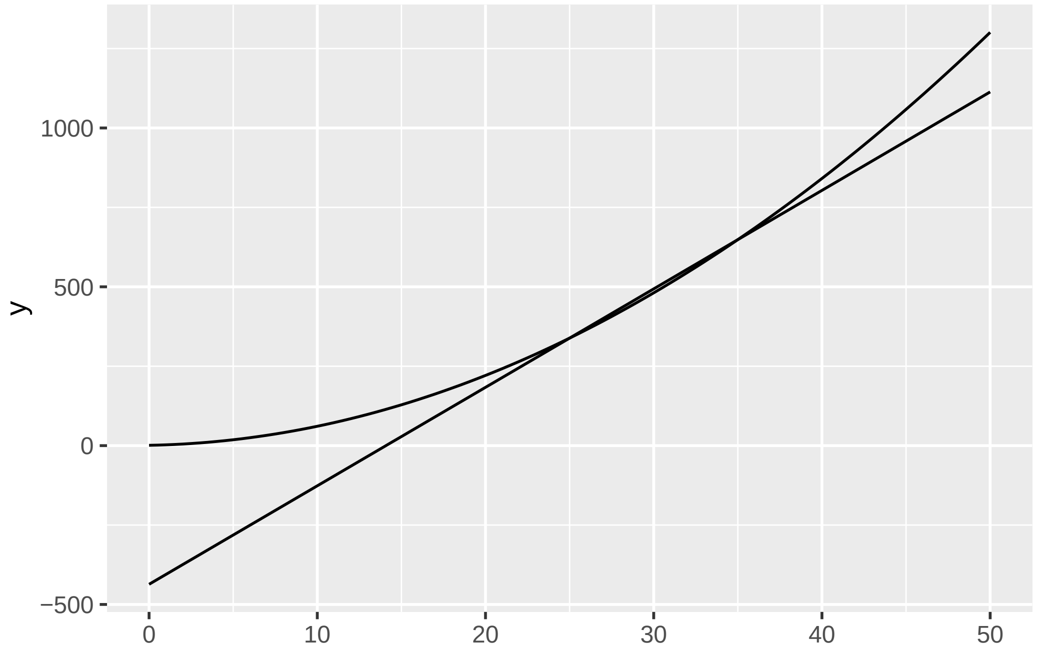
Using the values for each of these components, we get

$$\alpha = 1 + 30 + 0.5(25 + 30^2) - 31 * 30 = -436.5$$

7. Plot the Population Regression Function (PRF) with the CEF and comment on the quality of the approximation.

In this case the PRF approximates the CEF fairly well, considering it is a linear approximation to a non-linear function. In particular, it captures the essential parts of the positive relationship, though it does not capture the tails extremely well.

```r
ggplot() +
  geom_function(fun = function(x) 1 + x + 0.5*x^2) +
  geom_function(fun = function(x) -436.5 + 31*x ) +
  xlim(0,50)
```


y
1000
500
0
−500
0
10
20
30
40
50