Social Statistics

Introducing Regression

November 14, 2023

Merging Datasets

- I usually use `left_join()`. It keeps all observations in the first variable and keeps those observations in the second variable that have a match based on the `by` values:

```
gender_datasets <- left_join(gender_index, hdi_gender,
                             by = "country")
```

- Now repeat the process using the new `gender_datasets` dataframe and the `human_development_index` data set:

```
hdi <- left_join(gender_datasets, human_development_index,
                 by = "country")
```

Where We've Been

- Descriptive statistics gave us means, standard deviations
  - → "What are the spreads and the shapes of our observed distributions?"
- Probability gave us ways to use our sample statistics to predict ranges of possible population parameters
  - → "What is the likelihood of getting the values we observe?"
- Inference gave us tools to test significance
  - → "What is the likelihood of getting a value more extreme than the values we observe?"
  - → "How confident can we be that our observations differ from values of the null hypotheses?"

Two Things We Still Want

1. Better conclusions

- Asssociations peaked with correlation
- If correlation coefficient tells us that X and Y *tend to move together*, regression tells us *how much* they tend to move together

Start With Regression Basics

- Basic assumption (for now): The relationship between X and Y is linear
  - → HS Flashback: $y = mx + b$, where m is the slope and b is the intercept
- Linear relationship is regression equation:
  - → \(\large{\widehat{y_i} = \alpha + \beta X_i + \epsilon_i}\)
  - → Read as: *regress y on x*

Start With Regression Basics

- $\large{\widehat{y_i} = \alpha + \beta X_i + \epsilon_i}$
  - → $\widehat{y_i}$ = predicted outcome, the best guess
  - → α = intercept or constant, where the line hits the y-axis when x is 0
  - → β = the slope, the multiplier for every X, known as the coefficient
  - → X_i = the observed value of X
  - → ϵ_i = error (or residual), difference between observed and predicted values

Example from UN Human Development Project

- Before moving forward, we need to standardize the schooling values.
- Use the `scale()` function for this…

```r
hdi <- hdi |>
  mutate(std_schooling_expected = scale(schooling_expected))
```

- Mean of standardized variable should be 0. SD should be 1.

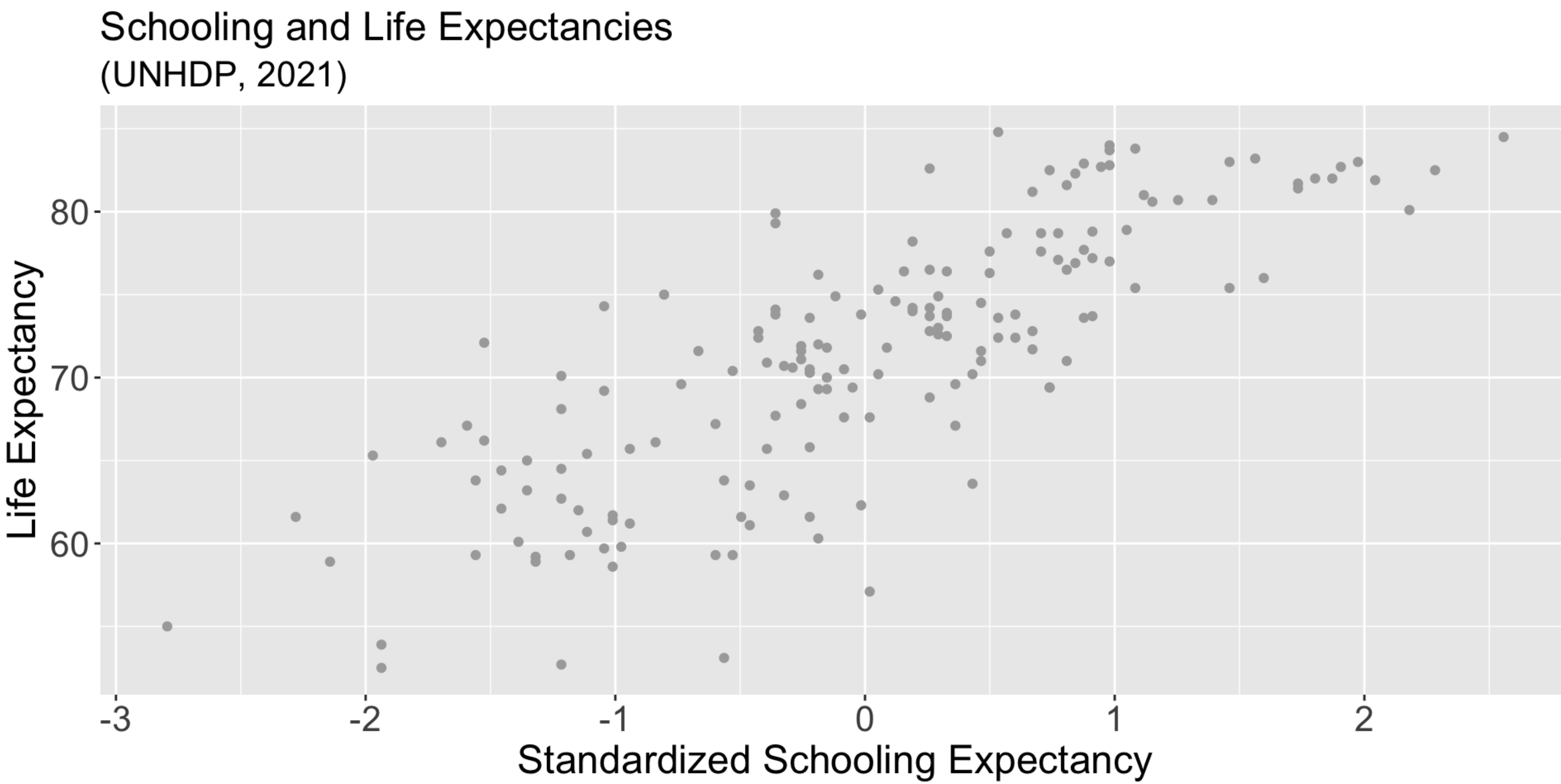
```r
mean(hdi$std_schooling_expected)
```

```
[1] 2.089321e-16
```

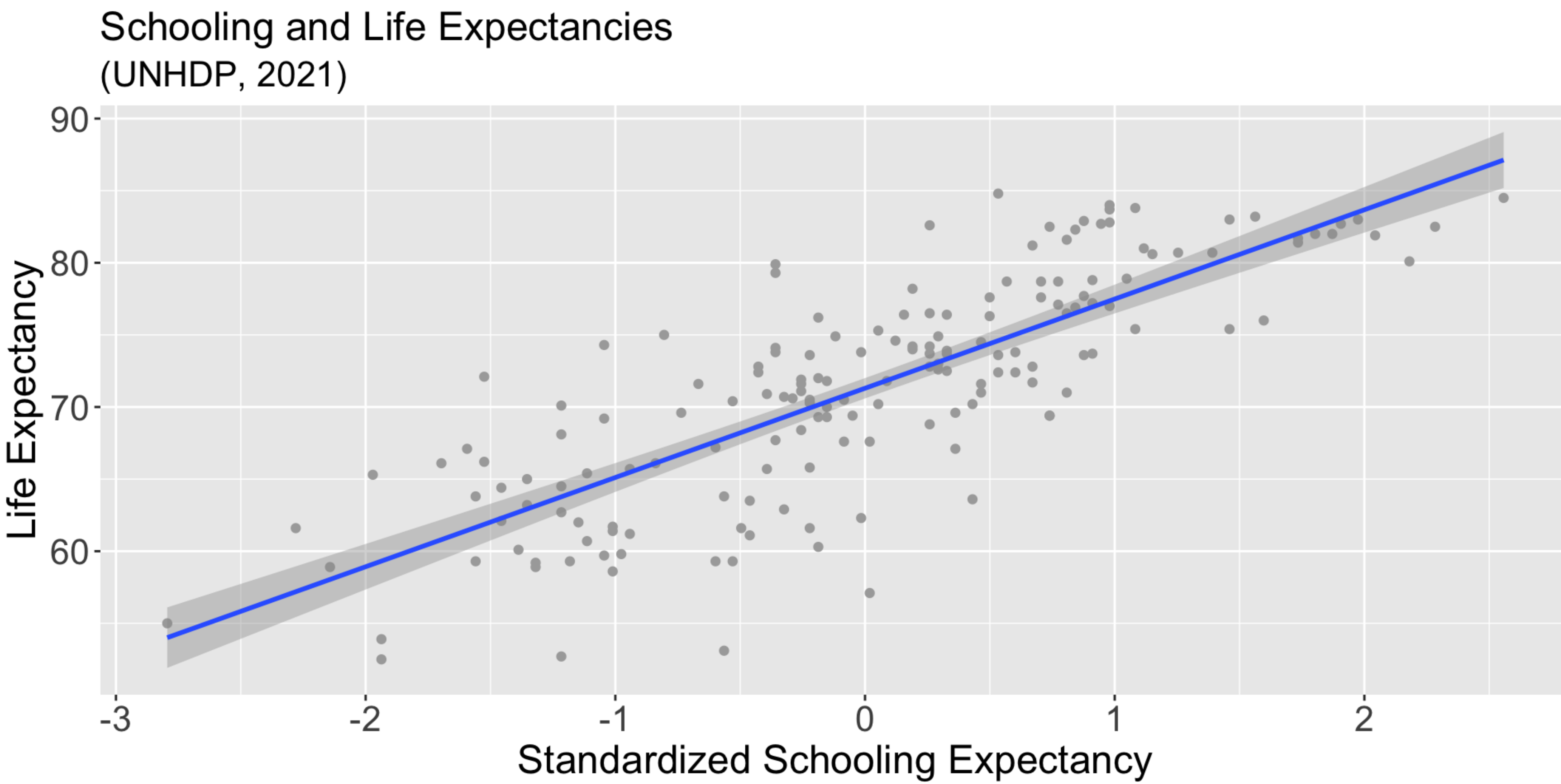
```r
sd(hdi$std_schooling_expected)
```

```
[1] 1
```

Example from UN Human Development Project



Example from UN Human Development Project



Fitting The Regression Line

- Recall that a *residual* is the difference between the observed value, y, and the predicted value on the line, $\widehat{y}$
- We want a line that makes every residual as small as possible
- Every observation has a residual. How do we combine them?
  - → Can't just add them up since negatives could cancel out positives
  - → Absolute values are the usual fix, but they don't help as much this time since they offer little guide for where to start with α and β

Fitting The Regression Line

- Sum of the squared residuals gets us closest
  - → $SSE = \sum{(y - \widehat{y})^2}$
  - → Line with the smallest sum has the *least squares*: why basic regression is called *Ordinary Least Squares*
- Squaring gives extra weight to biggest residuals (the observations that a given line does a particularly bad job at including)
- To find beta and alpha, we'll use basics we have seen: how the observed x's differ from the mean of x, how the observed y's differ from the mean of y, and how the distribution of x and y tend to move together

Fitting Beta and Alpha

- Let's try the example of regressing life expectancy in years on the standardized values of expected years of schooling
- Start with basic descriptives
  - → What's the correlation between the two variables?
  - → What are the mean and standard deviation of `std_schooling_expected`?
  - → What are the mean and standard deviation of `life_expectancy`?

Finding Beta and Alpha

```r
# Correlation
cor(hdi$std_schooling_expected, hdi$life_expectancy)
```

```
          [,1]
[1,] 0.8001698
```

- Interpretation?

Finding Beta and Alpha

```r
# Mean and Standard Deviation of X
mean(hdi$std_schooling_expected)
```

```
[1] 2.089321e-16
```

```r
sd(hdi$std_schooling_expected)
```

```
[1] 1
```

```r
# Mean and Standard Deviation of Y
mean(hdi$life_expectancy)
```

```
[1] 71.29941
```

```r
sd(hdi$life_expectancy)
```

```
[1] 7.733692
```

Fitting The Regression Line

- We have all we need to find beta:
  - → $\beta = cor_{xy} \frac{s_{y}}{s_{x}}$
- And beta will be the missing piece to help us find alpha:
  - → $\alpha = \bar{y} - \beta \bar{x}$

Finding Beta

- \(\large{\beta = cor_{xy} \frac {s_{y}}{s_{x}}}\)

```r
beta <- cor(hdi$std_schooling_expected,
      hdi$life_expectancy) *
     (sd(hdi$life_expectancy) /
      sd(hdi$std_schooling_expected))

beta
```

```
         [,1]
[1,] 6.188267
```

Interpreting Beta

- Every one unit increase in the value of X is associated with an increase of beta in the predicted value of Y, on average
  - → In this model, a one standard deviation increase in schooling expectancy is associated with an increase of 6.188267 years in life expectancy, on average
- And since we are working with linear regression, a one unit decrease in the value of X is associated with a decrease of beta in the predicted value of Y, on average
  - → In this model, a one standard deviation decrease in schooling expectancy is associated with a decrease of 6.188267 years in life expectancy, on average

Finding Alpha

- $\large{\alpha = \bar{y} - \beta \bar{x}}$

```
alpha <- mean(hdi$life_expectancy) -
      beta*(mean(hdi$std_schooling_expected))

alpha
```

```
         [,1]
[1,] 71.29941
```

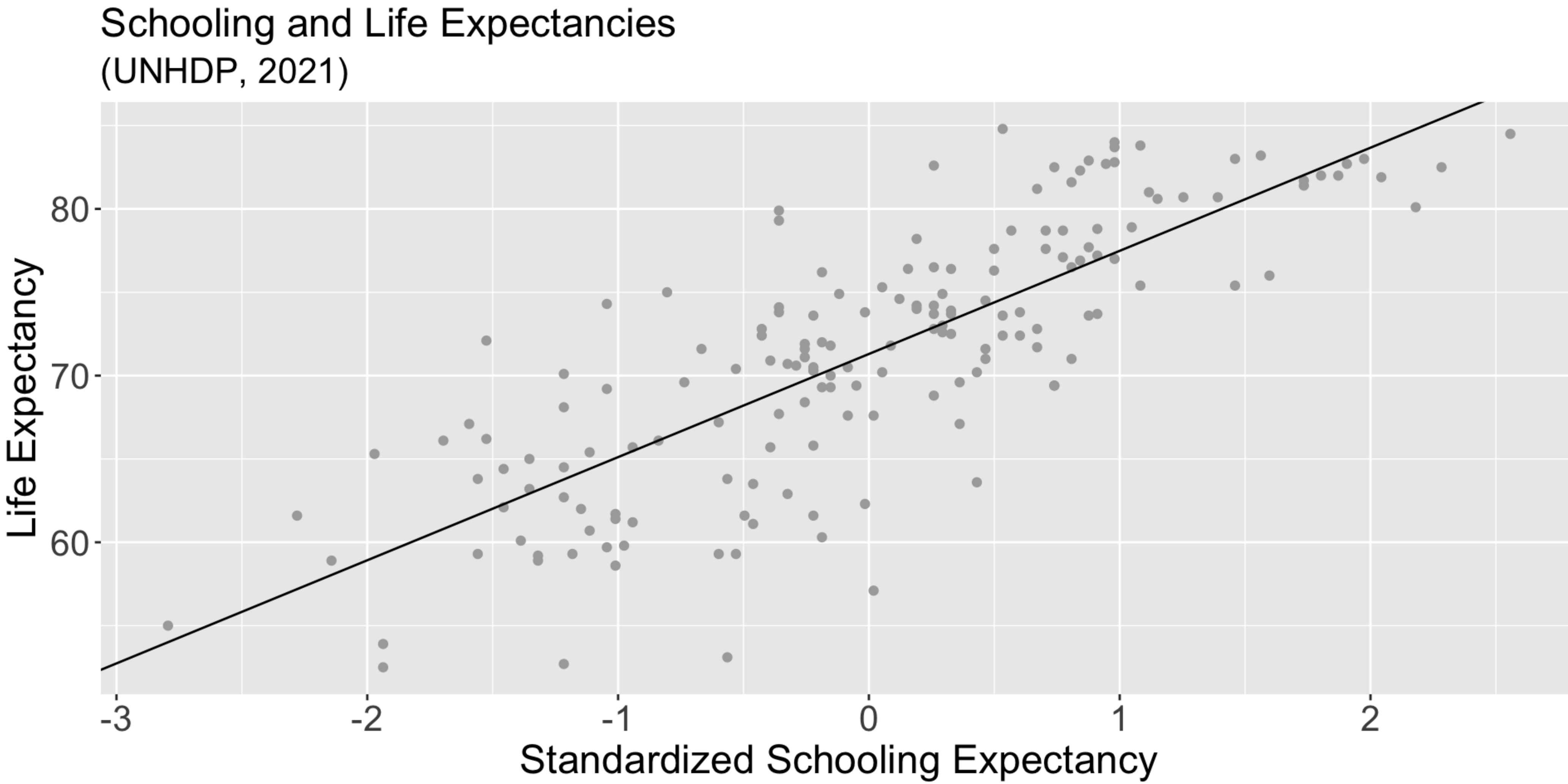
- When X is 0, our model predicts that Y should be 71.29941
- In this case (since x is standardized with a mean of 0), a country with a schooling expectancy at the average of the distribution would be predicted to have a life expectancy of 71.29941 years.

Fitting The Regression Line

- Now we have our line: y = 71.29941 + 6.188267 \(X\)
- Let's add it to our plot using `geom_abline()`:

```
schooling_life_plot1 <- ggplot(hdi, aes(
     x = std_schooling_expected, y = life_expectancy))

schooling_life_plot1 + geom_point(color = "Dark Gray") +
     labs(x = "Standardized Schooling Expectancy",
      y = "Life Expectancy",
      title = "Schooling and Life Expectancies",
      subtitle = "(UNHDP, 2021)") +
     geom_abline(intercept = 71.29941, slope = 6.188267)
```

Fitting The Regression Line

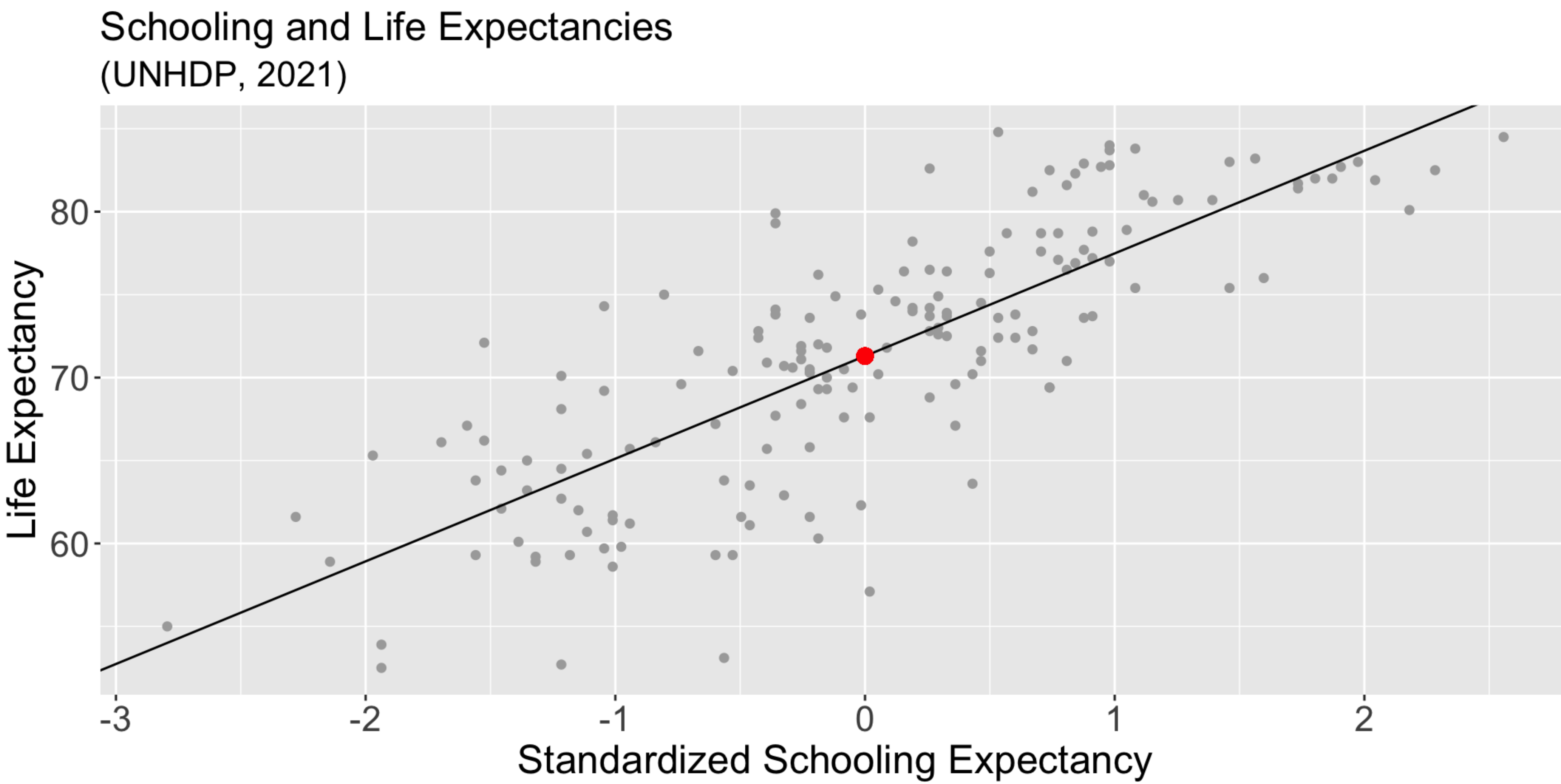


Predicting Values of Y

- If the line is correct, there should be a point on the line where X=0 and Y=71.29941

```
schooling_life_plot1 + geom_point(color = "Dark Gray") +
     labs(x = "Standardized Schooling Expectancy",
      y = "Life Expectancy",
      title = "Schooling and Life Expectancies",
      subtitle = "(UNHDP, 2021)") +
     geom_abline(intercept = 71.29941, slope = 6.188267) +
     geom_point(x = 0, y = 71.29941, color = "Red", size = 3)
```

Predicting Values of Y



Predicting Values of Y

- Digging Deeper: when \(\large{x}\) increases by 1, \(\large{\widehat{y}}\) is expected to increase by 6.188267
- So if \(\large{x}\) is 1 standard deviation above the mean, what is \(\large{\widehat{y}}\)? And if \(\large{x}\) is 1 standard deviation below the mean, what is \(\large{\widehat{y}}\)?
- Prediction always has to start with value of \(\large{\alpha}\)!

```
predicted_y_plus1sd <- alpha + beta*1
predicted_y_plus1sd
```

```
        [,1]
[1,] 77.48768
```

```
predicted_y_minus1sd <- alpha + beta*-1
predicted_y_minus1sd
```

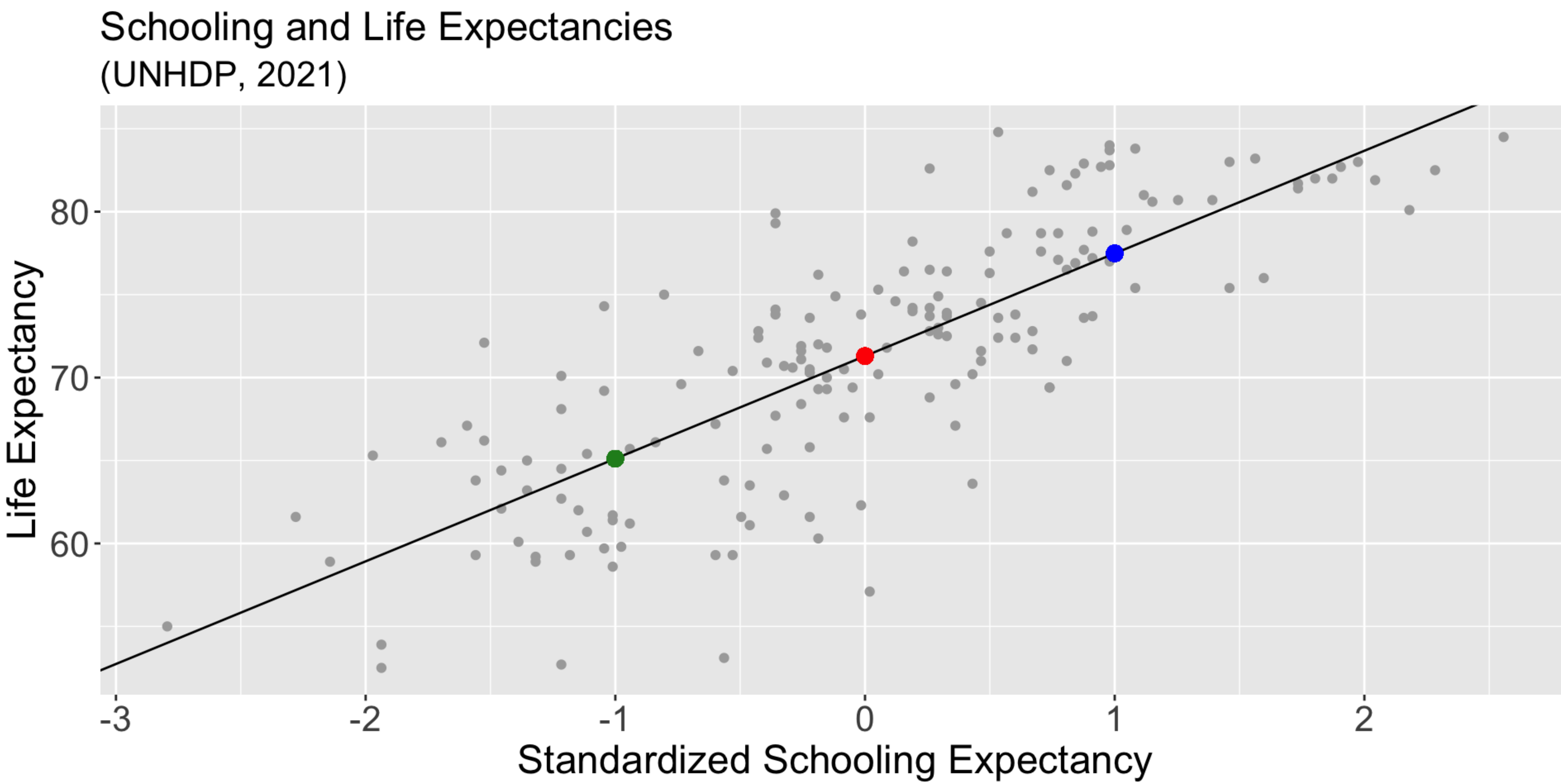
```
        [,1]
[1,] 65.11114
```

Predicting Values of Y

Put these points on our plot…

```
schooling_life_plot1 + geom_point(color = "Dark Gray") +
labs(x = "Standardized Schooling Expectancy",
y = "Life Expectancy",
      title = "Schooling and Life Expectancies",
      subtitle = "(UNHDP, 2021)") +
geom_abline(intercept = 71.29941, slope = 6.188267) +
geom_point(x = 0, y = 71.29941, color = "Red", size = 3) +
geom_point(x = 1, y = 77.48768, color = "Blue", size = 3) +
geom_point(x = -1, y = 65.11114, color = "Forest Green",
  size = 3)
```

Predicting Values of Y



Regression in R

- As always, R makes this easier. Meet the `lm()` command.

```
# Start by saving the model as an object:

schooling_life_model1 <-
      lm(life_expectancy ~ std_schooling_expected,
        data = hdi)
```

```
# Then look at the summary of the saved model:

summary(schooling_life_model1)
```

Regression in R

Regression in R

R’s Regression Output - Std Error

R’s Regression Output - Std Error

- $\Large{se = \frac{s} {\sqrt{\sum{ (x - \bar{x})^2}}}}$
- $\Large{s = \sqrt {\frac {\sum{(y - \widehat{y})^2}}{n-2}}}$
- The standard error formula uses the predicted values of y to calculate the residuals
- R makes it easy to save all the predicted values from a model:

```
hdi$predicted_life_expectancy <-
    schooling_life_model1$fitted.values
```

R’s Regression Output - Std Error

- Now you can plug in the predicted values to the rest of the standard error equation:

```
se_numerator <- sqrt(sum((hdi$life_expectancy -
  hdi$predicted_life_expectancy)^2) /
    (length(hdi$life_expectancy) - 2))

se_denominator <- sqrt(sum((hdi$std_schooling_expected -
  mean(hdi$std_schooling_expected))^2))

se <- se_numerator / se_denominator

se
```

```
[1] 0.3578653
```

R’s Regression Output - Std Error

```
Call:
lm(formula = life_expectancy ~ std_schooling_expected, data = hdi)

Residuals:
     Min       1Q   Median       3Q      Max 
-14.7062  -3.1567   0.3007   2.7595  10.8199 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)             71.2994     0.3568  199.82   <2e-16 ***
std_schooling_expected   6.1883     0.3579   17.29   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.652 on 168 degrees of freedom
Multiple R-squared:  0.6403,	Adjusted R-squared:  0.6381 
F-statistic:   299 on 1 and 168 DF,  p-value: < 2.2e-16
```

R’s Regression Output - T Value

R's Regression Output - T Value

- t = coefficient estimate / standard error

```r
6.1883 / .3579
```

```
[1] 17.29058
```

R's Regression Output - T Value

```
Call:
lm(formula = life_expectancy ~ std_schooling_expected, data = hdi)

Residuals:
     Min       1Q   Median       3Q      Max
-14.7062  -3.1567   0.3007   2.7595  10.8199

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)             71.2994     0.3568  199.82   <2e-16 ***
std_schooling_expected   6.1883     0.3579   17.29   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.652 on 168 degrees of freedom
Multiple R-squared:  0.6403,    Adjusted R-squared:  0.6381
F-statistic:   299 on 1 and 168 DF,  p-value: < 2.2e-16
```

R’s Regression Output - P Value

R's Regression Output - P Value

```r
# Area in right tail:
pr_tail <- 1 - pt(17.29, df = 168)

# Area in both tails (what output gives):
2 * pr_tail
```

```
[1] 0
```

- Can we reject the null hypothesis that the coefficient for `std_schooling_expected` is different from 0?
  - → Yes, because `Pr(>|t|)` is less than .05
- Note the stars!

R’s Regression Output - P Value

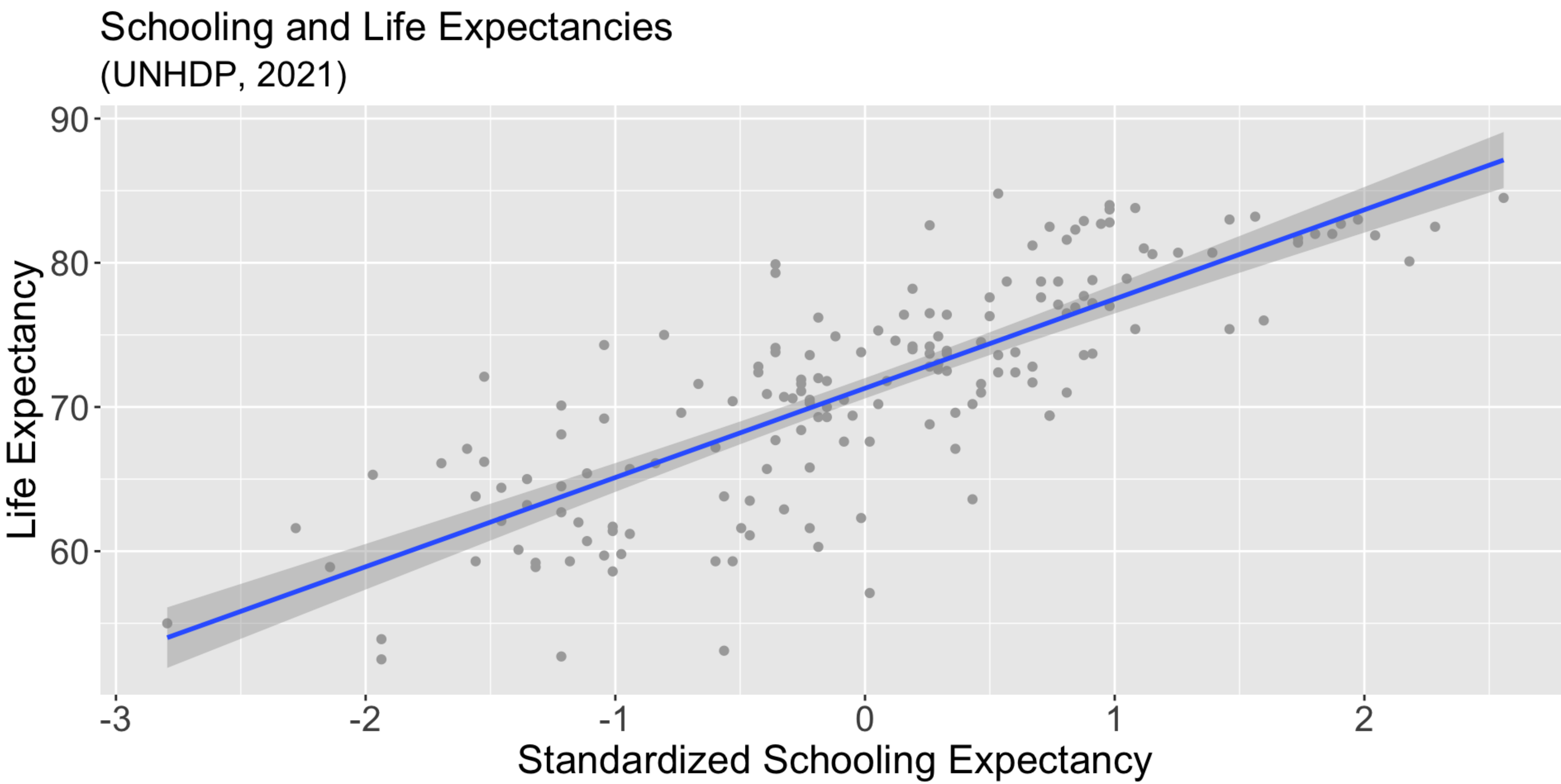
```
Call:
lm(formula = life_expectancy ~ std_schooling_expected, data = hdi)

Residuals:
     Min       1Q   Median       3Q      Max 
-14.7062  -3.1567   0.3007   2.7595  10.8199 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)             71.2994     0.3568  199.82   <2e-16 ***
std_schooling_expected   6.1883     0.3579   17.29   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.652 on 168 degrees of freedom
Multiple R-squared:  0.6403,	Adjusted R-squared:  0.6381 
F-statistic:   299 on 1 and 168 DF,  p-value: < 2.2e-16
```

Plotting Regressions

- More common to use `geom_smooth(method = lm)` than `geom_abline()`:

```
schooling_life_plot1 + geom_point(color = "Dark Gray") +
     labs(x = "Standardized Schooling Expectancy",
          y = "Life Expectancy",
      title = "Schooling and Life Expectancies",
      subtitle = "(UNHDP, 2021)") +
      geom_smooth(method = lm)
```

Plotting Regressions



Exercise 1

- Regress the gender inequality index (`gender_inequality_index`) on the average years of schooling completed by female residents (`schooling_mean_female`).

```
female_inequality_schooling_model <-
      lm(gender_inequality_index ~ schooling_mean_female,
            data = hdi)
```

```
summary(female_inequality_schooling_model)
```

Exercise 1

```
Call:
lm(formula = gender_inequality_index ~ schooling_mean_female,
    data = hdi)

Residuals:
    Min      1Q  Median      3Q     Max
-23.709  -8.009  -0.590   7.384  41.657

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)            75.7103     2.2197   34.11   <2e-16 ***
schooling_mean_female  -4.7189     0.2356  -20.03   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.75 on 167 degrees of freedom
```

Exercise 1

- Inequality Index = 75.7103 + (-4.7189 \(\times\) Schooling Mean Female)
- An increase of one year in the average years of schooling completed by female residents is associated with a decrease in the gender inequality index of 4.72, on average.
- In the US, the average years of schooling for females residents is 13.7. What is the US' predicted value on the gender inequality index?

```
75.7103 + (-4.7189*13.7)
```

```
[1] 11.06137
```

Exercise 1

- How does the predicted value of the gender inequality index compared to the observed value?

```
hdi |>
  filter(country == "United States") |>
  select(gender_inequality_index)
```

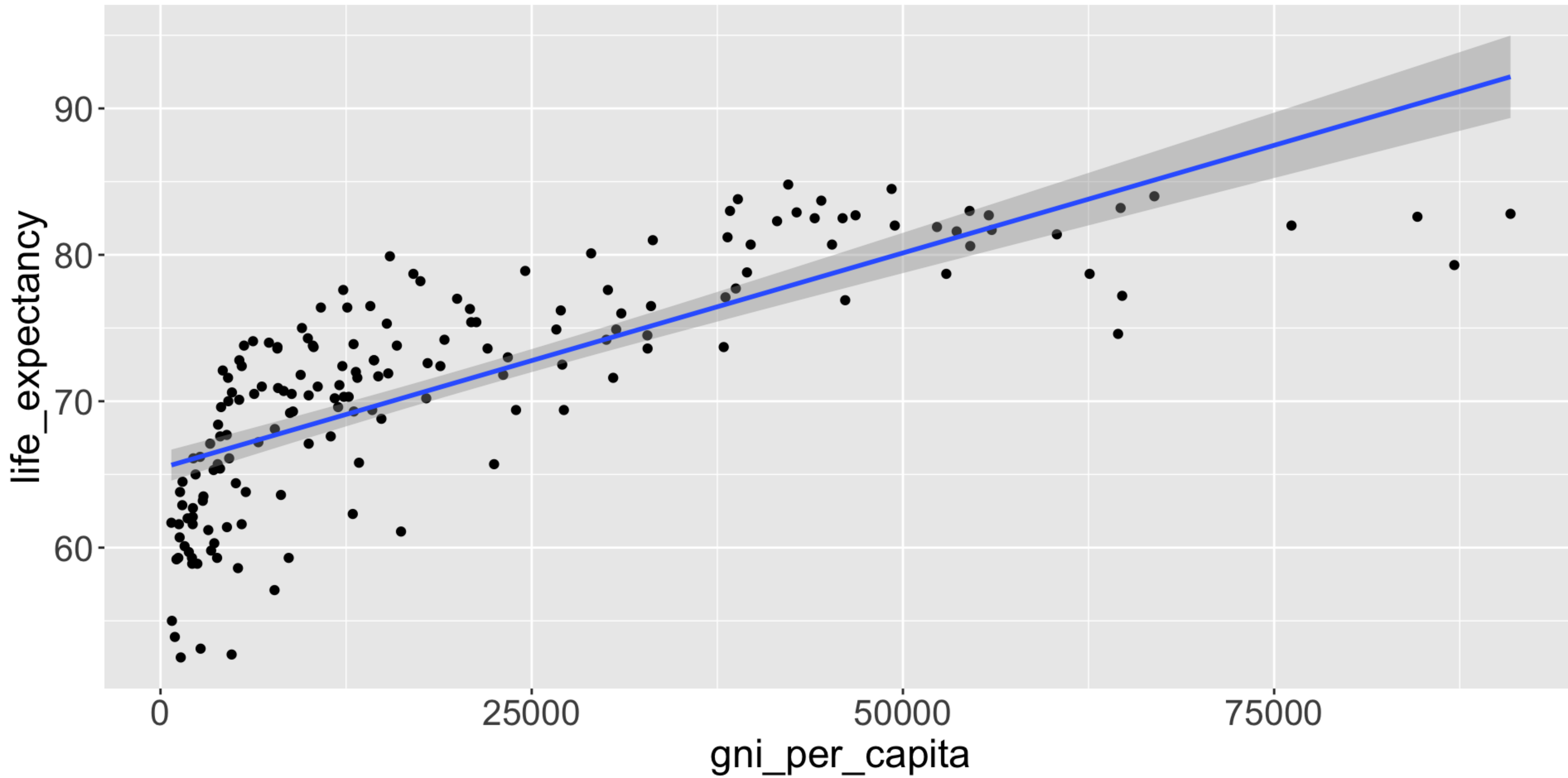
```
# A tibble: 1 × 1
  gender_inequality_index
                    <dbl>
1                    17.9
```

Exercise 2

- What would you expect about the relationship between `gni_per_capita` and `life_expectancy`?

```r
income_life_expectancy_plot <- ggplot(hdi, aes(x = gni_per_capita,
                                          y = life_expectancy)) + geom_point() +
  geom_smooth(method = lm)

income_life_expectancy_plot
```

Exercise 2

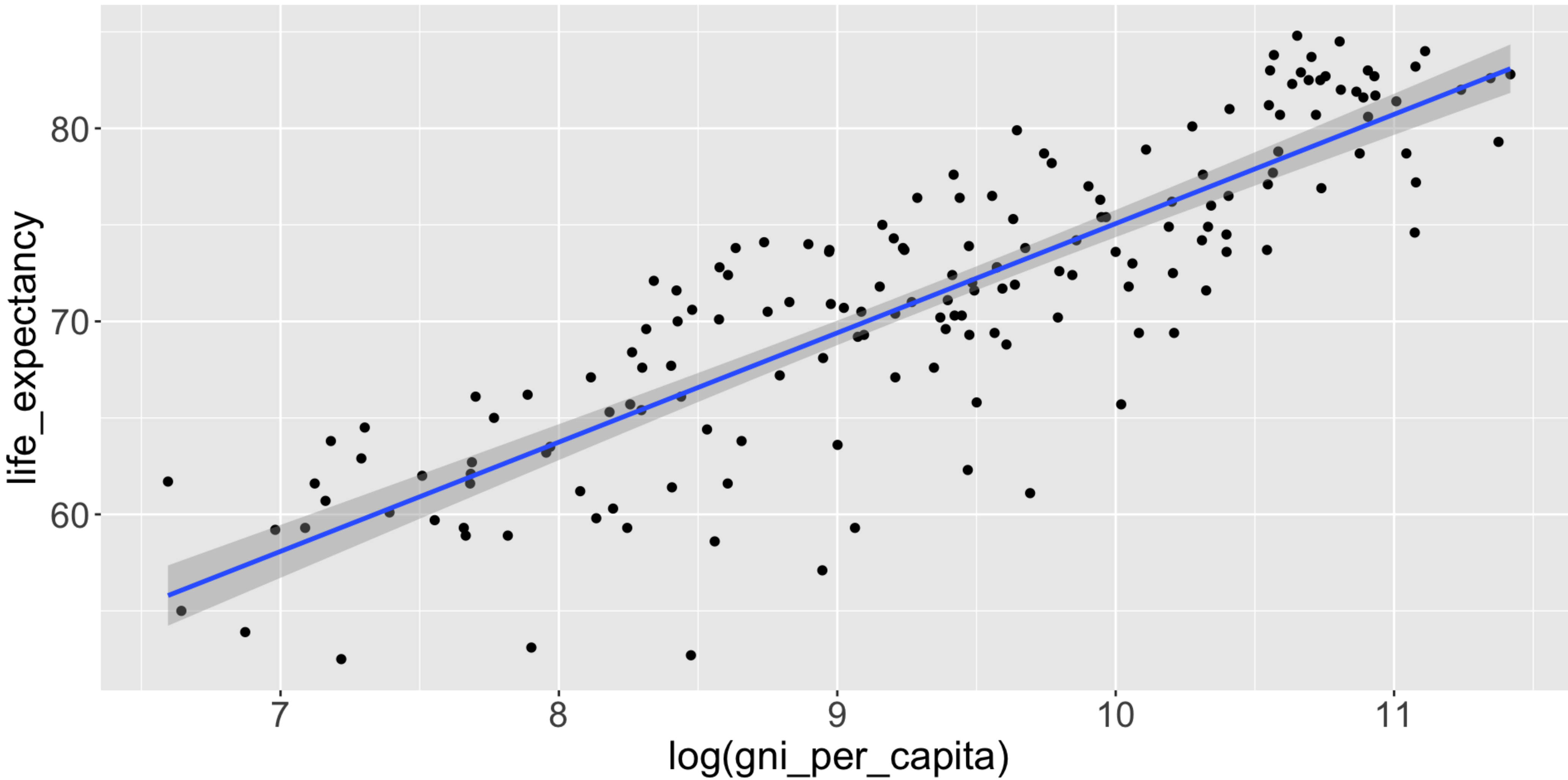


90
80
70
60
life_expectancy
0
25000
50000
75000
gni_per_capita

Exercise 2

```
income_log_life_expectancy_plot <- ggplot(hdi,
                                          aes(x = log(gni_per_capita),
                                           y = life_expectancy)) + geom_point() +
  geom_smooth(method = lm)

income_log_life_expectancy_plot
```

Exercise 2



Exercise 2

- Try the regression model using `life_expectancy` and `log(gni_per_capita)`…

```
income_life_expectancy_model <-
      lm(life_expectancy ~ log(gni_per_capita),
            data = hdi)
```

Exercise 2

```r
summary(income_life_expectancy_model)
```

```

Call:
lm(formula = life_expectancy ~ log(gni_per_capita), data = hdi)

Residuals:
     Min       1Q   Median       3Q      Max 
-13.7283  -2.3303   0.2772   3.0502   6.8427 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)          18.4541     2.4978   7.388 6.65e-12 ***
log(gni_per_capita)   5.6612     0.2655  21.321  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.029 on 168 degrees of freedom
Multiple R-squared:  0.7301,	Adjusted R-squared:  0.7285 
```

Exercise 2

- An increase in one unit of log gross national income is associated with an increase of 5.6612 years in life expectancy, on average. This increase is significant.
- A ten percent increase in gross national income is associated with a significant increase of 5.6612 years in life expectancy, on average.
- What is the predicted life expectancy for the United States?

```
log(hdi$gni_per_capita[hdi$country=="United States"])
```

```
[1] 11.07852
```

- To “exponentiate” logs…

```
exp(11.07852)
```

```
[1] 64764.96
```

Exercise 2

```
18.4541 + (5.6612*11.07852)
```

```
[1] 81.17182
```