Lab 3: Linear Regression

Rafael Campos-Gottardo[*]

2025-03-22

Linear Regression

Linear regression represents one of the most important tools in a political scienctists quantitative toolkit. Linear regression is used by most political scienctists for descriptive, preditive, and causal inference. At its most basic linear regression is a tool to summarize how the mean of an outcome variable (Y) changes based on a linear function of predictor variables $(\mathbf{X})$. In other words regression estimates the difference in the mean outcome variable resulting from a one-unit change in the predictor variables. We can use our class dataset to illustrate this feature of regression.

Let's look at the difference in means of our social media usage variable by gender. We can use the summary table from our `lab3.R` script.

```
mean_social_use
```

```
## # A tibble: 3 x 5
##   Gender  Mean Standard_Deviation   Min   Max
##   <chr>  <dbl>              <dbl> <dbl> <dbl>
## 1 Man    12.9                9.77     3    35
## 2 Woman   9.34               4.39     3    21
## 3 <NA>    4                  2.83     2     6
```

Let's extract the mean value of social media usage for the men and women in our class and find the difference in means.

```
mean_social_use$Mean[2] - mean_social_use$Mean[1]
```

```
## [1] -3.55625
```

This tells us that on average the men in this class use social media for 3.56 more hours a week than the women. However, calcuating the difference in means using this method does not tell us whether this difference in means is statistically signficant. In order to determine that this difference is signficantly different from 0, we can use a bivariate linear regression model to calcuate the standard error and p-value. To fit a linear regression model in R we use the `lm()` function.

The `lm()` function by default requires two arguements first it requires a formula object and second it requires a data arguement. A formula object includes a set of one or more X variables and a Y variable separated by a tilda (`~`). The basic struture of a bivariate formula object is `Y ~ X` and a multivariate model is `Y ~ X_1 + X_2 + X_3`. We can write the formula directly into the regression model or save it first as a formula object.

```
# We can save our formula object
bivariate_formula <- social_use ~ Gender

# And use the class function to see that this is a formula
class(bivariate_formula)
```

[*] This lab guide builds on those created by Aaron Erlich and Colin Scott. I would also like to acknowledge Brendan Szendro, Nicholas Vaxelaire, Guila Cohen, and Natalie Delmonte for their help developing this lab guide.

```
## [1] "formula"
```

Now that we have created a formula object we can use the `lm()` function to regress `social_use` (Y) on `Gender` (X).

```r
bivariate_model <- lm(bivariate_formula, data = class_df)

# Now we can use the summary function to summarize our model

summary(bivariate_model)
```

```
##
## Call:
## lm(formula = bivariate_formula, data = class_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9000 -2.9000 -1.3437  0.6563 22.1000
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   12.900      1.908   6.761 4.04e-08 ***
## GenderWoman   -3.556      2.186  -1.627    0.112
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.033 on 40 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.06207,    Adjusted R-squared:  0.03862
## F-statistic: 2.647 on 1 and 40 DF,  p-value: 0.1116
```

What do we notice about our regression model?

We will go through this output line by line.

1. The call at the top repeats the code we used to fit our regression model.

2. The residuals indicate the distance between each point and our regression line. When we graph our regression model this will become more clear.

3. The coefficients are the most important part of a regression model for interpretation. The `(Intercept)` tells us the value of our dependent variable when our X variable is equal to 0. The `GenderWoman` coefficient tells us the difference in the mean value from man (the reference category) to woman. The estimate column tells us the actual values. The estimate for the intercept indicates that the average value of social use for men (the reference category) is 12.90. The estimate for the `GenderWoman` coefficient tells us that the average value for women is 3.56 hours lower then for men. The `Std. Error` is used to calculate the `t.value` and confidence intervals. The `t.value` is used to calculate the `p-value`. Finally, the p-value (`Pr(>|t|)`) indicates the probability that the coefficient is statistically different from 0 based on the standard error. You will notice in this model that the Gender coefficient is not different from 0. We will discuss the Estimates in more detail shortly.

4. The `Signif. codes` provide a short hand that can be used to determine if a variable is statistically significant. The general rule of thumb is that if a variable has at least one star next to it then the coefficient is statistically different from 0.

5. The `Residual standard error` is a measure of variation of the residuals. Residuals are the distance

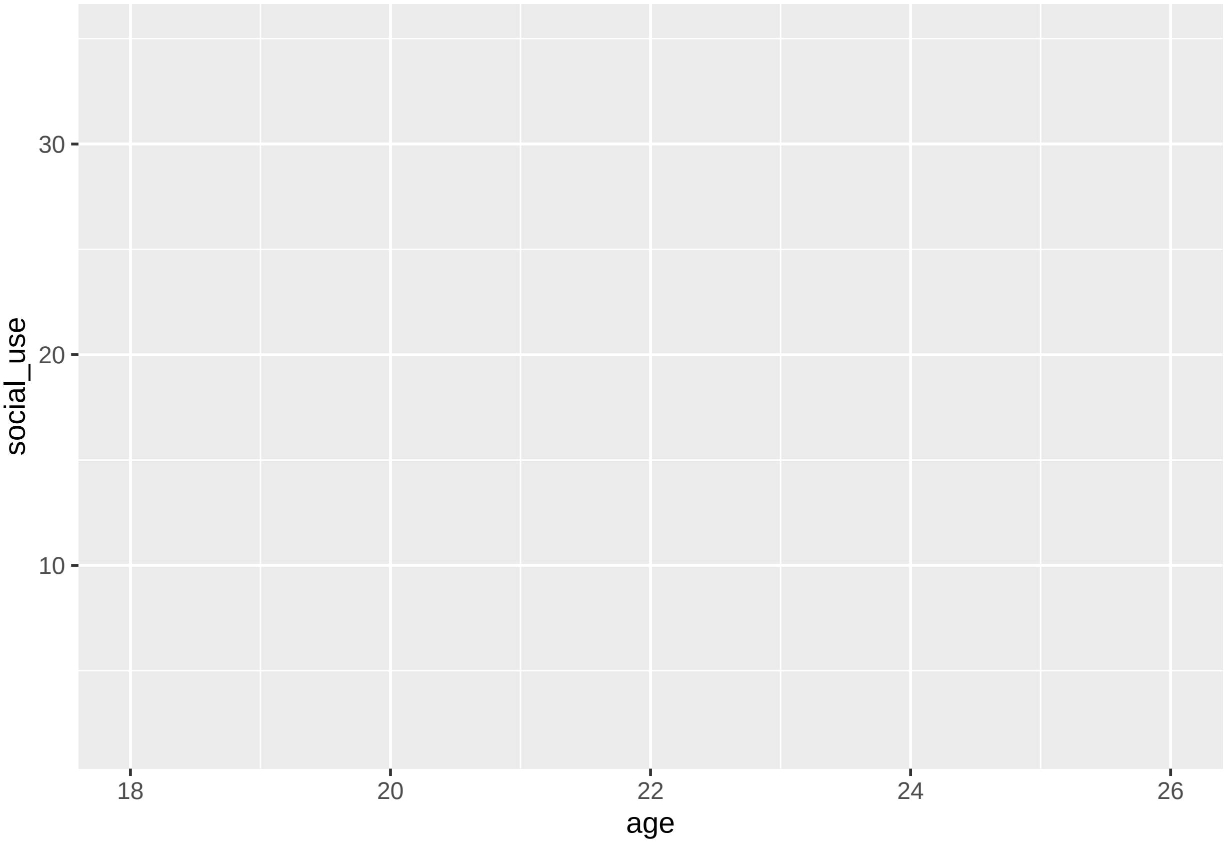
Visualization of Regression Models

In order to better understand the coefficients we can visualize the difference in means using a graph.

Most graphing in R is done using a package called `ggplot`, which is part of the `tidyverse`. However, `ggplot` was developed before the `tidyverse` added the pipe (`%>%`) therefore it uses a plus (`+`) to serve the same purpose. This is an odd feature of `ggplot` that may take some getting used.

To use `ggplot` we always start by piping the dataset into the `ggplot()` function. In the `ggplot()` function we can then define our X and Y variables using the `aes()` function. For this graph our X variable is Gender and our Y variable is social media usage. We are also going to define the colour parameter so that the points in our graph are coloured by gender for easier visualization.

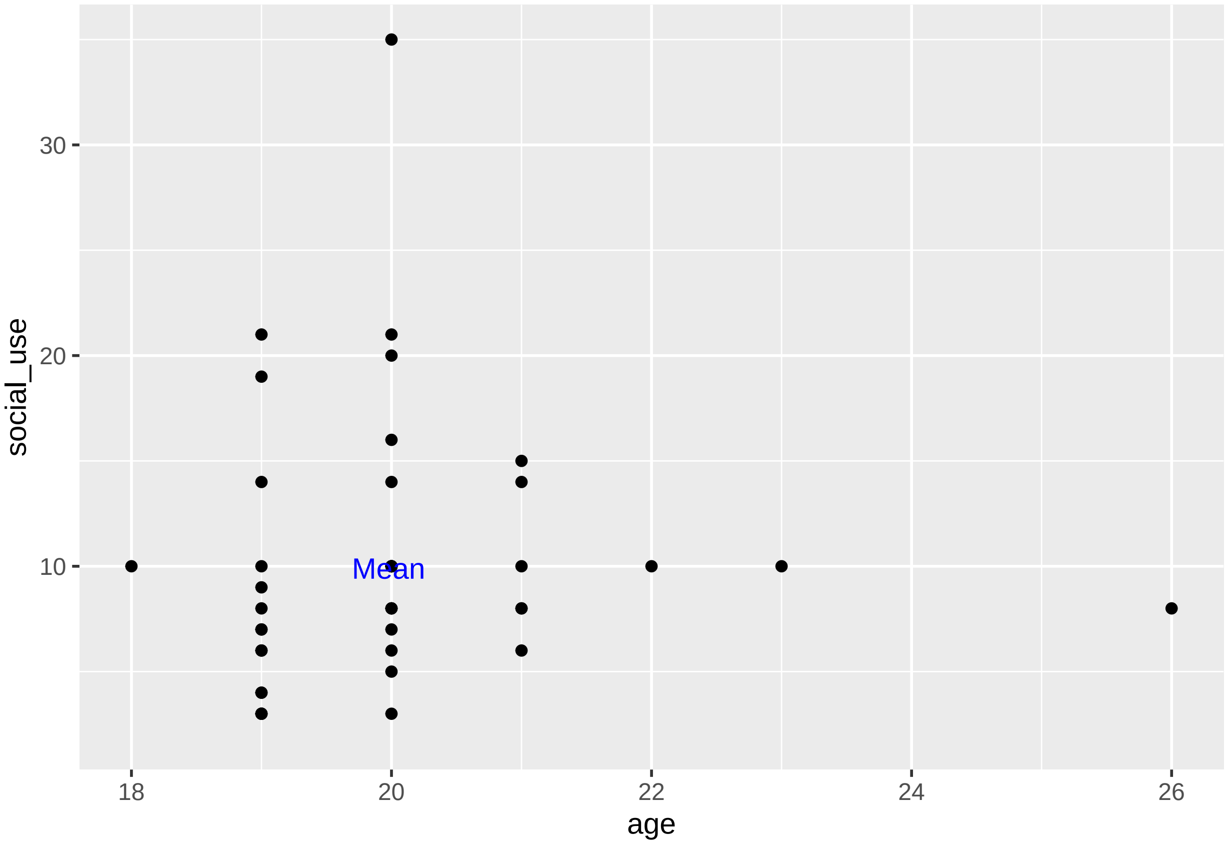
```
class_df %>%
  ggplot(aes(x = age, y = social_use, colour = Gender))
```



You will notice that this code produces a blank graph. This is because we have not specified any geom layers for our graph. We want to make a scatterplot with a regression line on it. To do this we use the `geom_point()` function and add it to the graph using the `+`. We also added `Mean` which is at the mean of X and Y.

```
class_df %>%
  ggplot(aes(x = age, y = social_use)) +
  geom_point() +
  annotate("text", x = mean(class_df$age, na.rm = TRUE),
           y = mean(class_df$social_use, na.rm = TRUE),
           label = "Mean", col = "blue")
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



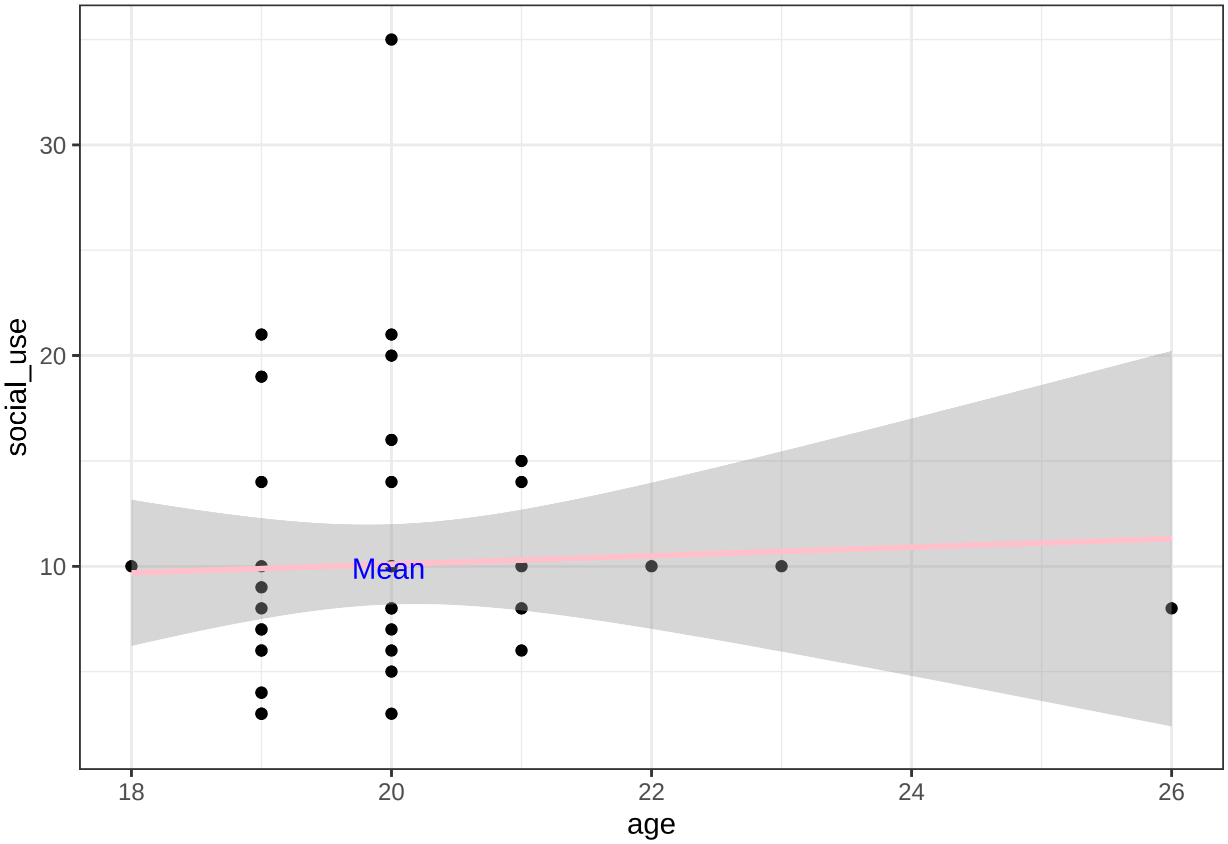
Now we can add a regression line using `geom_smooth()`.

```r
class_df %>%
  ggplot(aes(x = age, y = social_use)) +
  geom_point() +
   geom_smooth(method = "lm", col = "pink") +
  annotate("text", x = mean(class_df$age, na.rm = TRUE),
           y = mean(class_df$social_use, na.rm = TRUE),
           label = "Mean", col = "blue") +
  theme_bw()
```

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

## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Notice how the regression line goes through the mean of X and Y.

Multiple regression

Now that we have learned the basics of linear regression what is multiple regression. Similarly to a bivariate regression model, a multivariate model is a difference in means estimator. However, in the multivariate model we estimate the change in the mean of the Y variable based on a linear function of multiple X variables. In order to fit a multivariate regression model we use the same function as a bivariate regression model.

```
multivariate_model <- lm(social_use ~ Gender + age, data = class_df)

summary(multivariate_model)
```

```
##
## Call:
## lm(formula = social_use ~ Gender + age, data = class_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.9519 -3.0329 -1.3194  0.8752 22.0481
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.7613    14.3333   0.541    0.591
## GenderWoman  -3.6325     2.2200  -1.636    0.110
## age           0.2595     0.7173   0.362    0.719
##
## Residual standard error: 6.1 on 39 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.0652, Adjusted R-squared:  0.01727
## F-statistic:  1.36 on 2 and 39 DF,  p-value: 0.2685
```

For this model each estimate (β coefficient) represents the average difference in Y given a one unit change in the X value holding the other variables constant. Therefore, we can interpret the coefficient on Gender as women use social media on average 3.63 hours less per week then men, while controlling for age (or holding age constant).