Simple_Regression_Intro

Emmenta Janneh

2024-02-01

We load in the data

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
hibbs <- read.csv("https://dcgerard.github.io/stat_415_615/data/hibbs.csv")
```

You can use `lm()` (linear model) function to find the OLS estimates. - First argument is a "formula" which uses a tilde ~ y ~ x - Second argument is the data frame containing the variables - You assign the output to some variables, so you can manipulate it later

```
lm_hibbs <- lm(vote ~ growth, data = hibbs)
lm_hibbs
```

```
##
## Call:
## lm(formula = vote ~ growth, data = hibbs)
##
## Coefficients:
## (Intercept)       growth
##      46.248        3.061
```

To interact with this lm object, it is wasiest to use the broom package. - `glance()`: will provide a 1 row summary of the model - `tidy()`: will provide one row per parameter estimate - `augment()`: will provide one row per observational unit

```r
library(broom)
```

```
## Warning: package 'broom' was built under R version 4.3.2
```

```r
tidy(lm_hibbs)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    46.2      1.62      28.5  8.41e-14
## 2 growth          3.06     0.696      4.40 6.10e- 4
```

```r
glance(lm_hibbs)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.580         0.550  3.76      19.3 0.000610     1  -42.8  91.7  94.0
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
augment(lm_hibbs)
```

```
## # A tibble: 16 x 8
##     vote growth .fitted .resid   .hat .sigma  .cooksd .std.resid
##    <dbl>  <dbl>   <dbl>  <dbl>  <dbl>  <dbl>    <dbl>      <dbl>
##  1  44.6   2.4     53.6 -8.99  0.0711   2.92 0.235       -2.48
##  2  57.8   2.89    55.1  2.67  0.0962   3.83 0.0296       0.746
##  3  49.9   0.85    48.8  1.06  0.100    3.89 0.00491      0.297
##  4  61.3   4.21    59.1  2.21  0.246    3.84 0.0742       0.675
##  5  49.6   3.02    55.5 -5.89  0.106    3.50 0.162       -1.66
##  6  61.8   3.62    57.3  4.46  0.164    3.66 0.165        1.30
##  7  49.0   1.08    49.6 -0.603 0.0854   3.90 0.00131     -0.168
##  8  44.7  -0.39    45.1 -0.354 0.242    3.90 0.00186     -0.108
##  9  59.2   3.86    58.1  1.11  0.194    3.89 0.0130       0.328
## 10  53.9   2.27    53.2  0.745 0.0672   3.90 0.00151      0.205
## 11  46.6   0.38    47.4 -0.861 0.141    3.90 0.00501     -0.247
## 12  54.7   1.04    49.4  5.31  0.0877   3.59 0.105        1.48
## 13  50.3   2.36    53.5 -3.20  0.0698   3.80 0.0292      -0.882
## 14  51.2   1.72    51.5 -0.272 0.0636   3.90 0.000189    -0.0746
## 15  46.3   0.1     46.6 -0.234 0.173    3.90 0.000488    -0.0683
## 16  52     0.95    49.2  2.84  0.0932   3.82 0.0324       0.794
```

Note: **We use `augment()` to get the residuals and fitted values.

Lets take a look at another sample

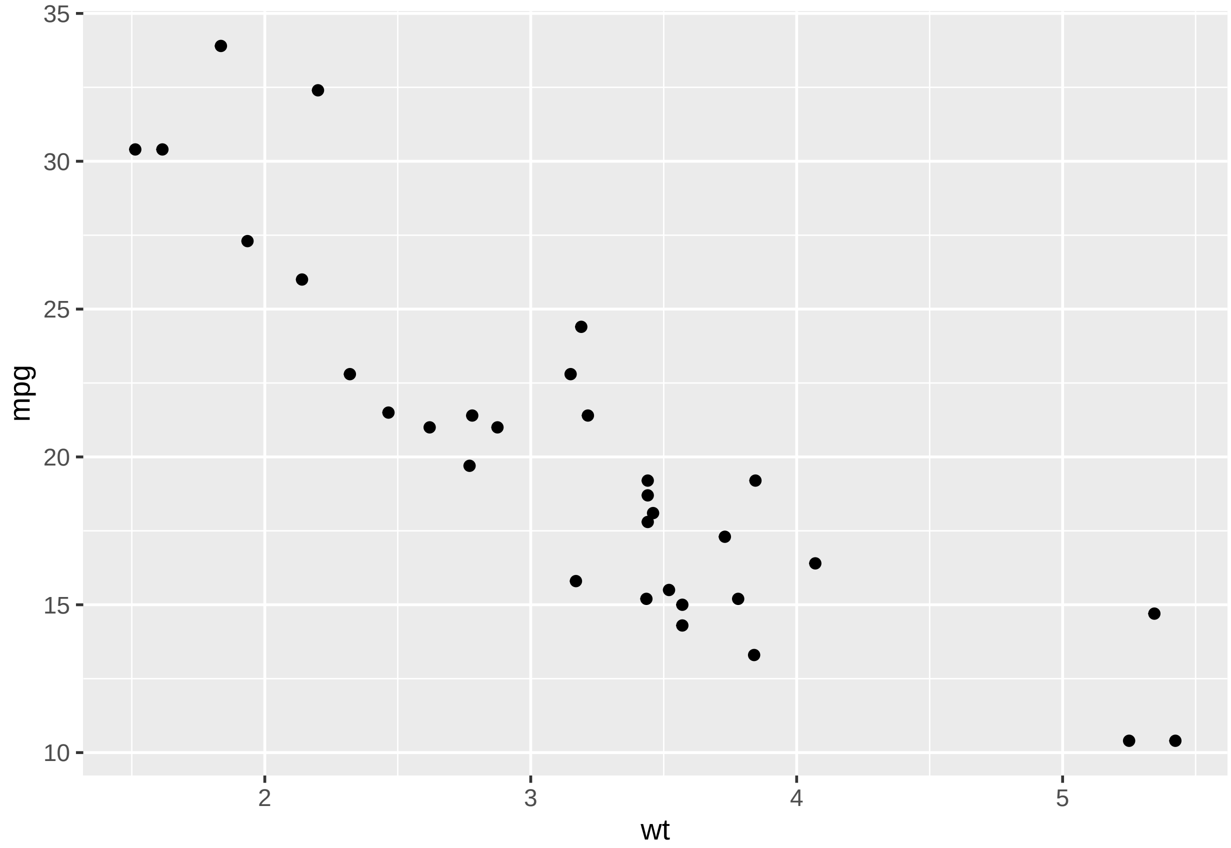
```r
data("mtcars")
lm_mtcars <- lm(mpg ~ wt, data = mtcars)
tidy(lm_mtcars)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    37.3      1.88      19.9  8.24e-19
## 2 wt             -5.34     0.559     -9.56 1.29e-10
```

Cars that weight 1000 pounds more have 5.3 worse MPG on average. 37.285 is the y-intercept of the regression line.

You should always plot the data BEFORE doing a regression

```
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point()
```



What if you want an estimated value at an X not in the data set? To predict what Y will be given a value of X, you create a new data frame with those values of X.

```
newdf <- data.frame(growth = c(1,2,3.3))
```

Then you feed this, as well as the lm object, into `predict()`

```
predict(object = lm_hibbs, newdata = newdf)
```

```
##        1        2        3
## 49.30818 52.36870 56.34739
```