AE 20: Multiple regression I

Dav King

3/26/2022

```
library(tidyverse)
library(tidymodels)
library(scatterplot3d)
```

Reminder

- Lab Due on Friday at 11:59 PM.
- Discussion of Peer Review.

Learning goals

- Gain proficiency with multiple linear regression
- Review what a dummy variable is

To begin, we'll work with a dataset with information on the price of sports cars new set of pokemon data.

```
sports_car_prices <- read_csv("sportscars.csv")
pokemon <- read_csv("pokemon150.csv")
```

The linear model with one predictor

- Previously, we were interested in the
$$\beta_0$$
(population parameter for the intercept) and the
$$\beta_1$$
(population parameter for the slope) in the following model:

$$\hat{y} = \beta_0 + \beta_1 \ x + \epsilon$$

- Unfortunately, we can't get these values
- So we use sample statistics to estimate them:

$$\hat{y} = b_0 + b_1 \ x$$

The linear model with multiple predictors

$$\hat{y} = \beta_0 + \beta_1\ x_1 + \beta_2\ x_2 + \cdots + \beta_k\ x_k + \epsilon$$

- Sample model that we use to estimate the population model:

$$\hat{y} = b_0 + b_1\ x_1 + b_2\ x_2 + \cdots + b_k\ x_k$$

An example

The file `sportscars.csv` contains prices for Porsche and Jaguar cars for sale on cars.com.

`car`: car make (Jaguar or Porsche)

`price`: price in USD

`age`: age of the car in years

`mileage`: previous miles driven

The linear model with a single predictor

```
prices_model <- linear_reg() %>%
  set_engine("lm") %>%
  fit(price ~ age, data = sports_car_prices)
tidy(prices_model)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   53246.     3322.     16.0  5.70e-23
## 2 age           -2149.      466.     -4.62 2.22e- 5
```

But is the age the only variable that predicts price?

The linear model with multiple predictors

$$\hat{y} = \beta_0 + \beta_1\ x_1 + \beta_2\ x_2 + \cdots + \beta_k\ x_k + \epsilon$$

- Sample model that we use to estimate the population model:

$$\hat{y} = b_0 + b_1\ x_1 + b_2\ x_2 + \cdots + b_k\ x_k$$

Let's add a variable.

Multiple Regression

```r
m_main <- linear_reg() %>%
  set_engine("lm") %>%
  fit(price ~ age + car, data = sports_car_prices)
m_main %>%
  tidy() %>%
  select(term, estimate)
```

```
## # A tibble: 3 x 2
##   term        estimate
##   <chr>          <dbl>
## 1 (Intercept)   44310.
## 2 age           -2487.
## 3 carPorsche    21648.
```

Linear model:

$$\widehat{price} = 44310 - 2487\ age + 21648\ carPorsche$$

- Plug in 0 for `carPorsche` to get the linear model for Jaguars.
- Plug in 1 for `carPorsche` to get the linear model for Porsches.
- Jaguar:

$$\widehat{price} = 44310 - 2487\ age + 21648 * 0 = 44310 - 2487 * age$$

- Porsche:

$$\widehat{price} = 44310 - 2487\ age + 21648 * 1 = 65958 - 2487\ age$$

- Rate of change in price as the age of the car increases does not depend on make of car (same slopes)
- Porsches are consistently more expensive than Jaguars (different intercepts)

Main effects, numerical and categorical predictors

```r
m_main %>%
  tidy() %>%
  select(term, estimate)
```

```
## # A tibble: 3 x 2
##   term        estimate
##   <chr>          <dbl>
## 1 (Intercept)   44310.
## 2 age           -2487.
## 3 carPorsche    21648.
```

```r
m_main_coefs <- m_main %>%
  tidy() %>%
  select(term, estimate)
m_main_coefs
```

```
## # A tibble: 3 x 2
##   term        estimate
##   <chr>          <dbl>
## 1 (Intercept)   44310.
## 2 age           -2487.
## 3 carPorsche    21648.
```

- **All else held constant**, for each additional year of a car's age, the price of the car is predicted to decrease, on average, by $2,487.
- **All else held constant**, Porsches are predicted, on average, to have a price that is $21,648 greater than Jaguars.
- Jaguars that have an age of 0 are predicted, on average, to have a price of $44,310.

Adjusted R-Squared

- The strength of the fit of a linear model is commonly evaluated using R^2.
- It tells us what percentage of the variability in the response variable is explained by the model. The remainder of the variability is unexplained.

Please recall:

- We can write explained variation using the following ratio of sums of squares:

$$R^2 = 1 - \left(\frac{SS\_Error}{SS\_Total}\right)$$

where SS_{Error} is the sum of squared residuals and SS_{Total} is the total variance in the response variable.

Adjusted R^2

$$R^2\_adj = 1 - \left(\frac{SS\_Error}{SS\_Total} \times \frac{n-1}{n-k-1}\right),$$

where n is the number of observations and k is the number of predictors in the model.

- Adjusted R^2 doesn't increase if the new variable does not provide any new information or is completely unrelated and can even decrease.
- This makes adjusted R^2 a preferable metric for model selection in multiple regression models.

Let's find the R^2 and adjusted R^2 for the `m_main` model we built.

```
glance(m_main)$r.squared
```

```
## [1] 0.6071375
```

```r
glance(m_main)$adj.r.squared
```

```
## [1] 0.5933529
```

Exercises

Now, let's do some exercises with the Pokemon data.

Exercise 1)

Are height and weight correlated with a Pokemon's hit points? Run a bivariate linear regression model for each of these. Do you find statistically significant results?

```r
hp_height <- linear_reg() %>%
  set_engine("lm") %>%
  fit(hp ~ height_m, data = pokemon)
hp_weight <- linear_reg() %>%
  set_engine("lm") %>%
  fit(hp ~ weight_kg, data = pokemon)
tidy(hp_height)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)     53.9      2.57     21.0  3.16e-46
## 2 height_m        10.9      1.44      7.52 4.90e-12
```

```r
tidy(hp_weight)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   60.1      1.98       30.4  1.43e-65
## 2 weight_kg      0.120    0.0151      7.94 4.59e-13
```

Yes, for each of these P is (much) less than 0.05; thus, we can reject H_0.

Exercise 2)

Other variables may be correlated with `hp`, e.g. a pokemon's legendary status.

Do legendary pokemon have higher `hp` than non-legendary pokemon? Compare mean `hp` between groups to support your answer.

```r
x <- pokemon %>%
  filter(is_legendary == 0) %>%
  pull(hp)
y <- pokemon %>%
  filter(is_legendary == 1) %>%
  pull(hp)
t.test(x, y, var.equal = T, alternative = "less")
```

```
##
##  Two Sample t-test
##
## data:  x and y
## t = -4.7458, df = 148, p-value = 2.425e-06
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf -27.3574
## sample estimates:
## mean of x mean of y
##  66.41958 108.42857
```

```
#Even if you wanted me to be boring and just compare the means of the two
#groups without a statistical test, this model still puts them out
```

Yes, legendary pokemon have a higher `hp`. Their mean `hp` is 108.429, while the mean `hp` of non-legendary pokemon is 66.420 - a difference with a p-value of essentially zero.

Write down a model to predict a pokemon's hitpoints based on their height, weight, legendary status (use x, y, β notation). Define each variable.

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + b_3x_3$$

Where $\hat{y}$ is our predicted hp, b_0 is our predicted hp for a non-legendary pokemon that is 0 meters tall and weighs 0 kilograms, b_1 is the amount that each unit increase in height x_1 increases the pokemon's predicted hp, b_2 is the amount that each unit increase in weight x_2 increases the pokemon's predicted hp, and b_3 is the amount that a pokemon being legendary (x_3) increases the pokemon's predicted hp.

Exercise 3)

Use `tidymodel` syntax to build a linear model with *all three variables* and estimate each β. Then, find and interpret the adjusted R^2.

```
hp_model <- linear_reg() %>%
  set_engine("lm") %>%
  fit(hp ~ height_m + weight_kg + is_legendary, data = pokemon)
tidy(hp_model)
```

```
## # A tibble: 4 x 5
##   term         estimate std.error statistic  p.value
##   <chr>           <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   55.9       2.91      19.2   7.45e-42
## 2 height_m       5.10      2.58       1.98  5.00e- 2
## 3 weight_kg      0.0790    0.0205     3.86  1.71e- 4
## 4 is_legendary   5.31     11.5        0.462 6.45e- 1
```

```
glance(hp_model)$adj.r.squared
```

```
## [1] 0.3304261
```

Interpret the meaning of your estimates and write a brief description below.

Intercept: 55.9: we expect a non-legendary pokemon with height and weight 0 to have 55.9 hp.

Slopes:

- 5.10: for each additional meter of height, we predict a corresponding increase of 5.10 hp for the pokemon.
- 0.079: for each additional kilogram of weight, we predict a corresponding increase of 0.079 hp for the pokemon.

= 5.31: we predict that being legendary will increase a pokemon's hp by 5.31.

Adjusted R^2: 0.33: 33% of the variance in Pokemon hp can be explained by its height, weight, and legendary status.

Exercise 4)

Some think that certain pokemon types have higher `hp` than others.

First, explore where there is a different `mean_hp` by `type_1`.

```
pokemon %>%
  group_by(type_1) %>%
  summarize(mean_hp = mean(hp))
```

```
## # A tibble: 18 x 2
##    type_1   mean_hp
##    <chr>      <dbl>
##  1 Bug         58.5
##  2 Dark        85.2
##  3 Dragon      83.6
##  4 Electric    72.9
##  5 Fairy       73.5
##  6 Fighting    70.8
##  7 Fire        75.7
##  8 Flying      79
##  9 Ghost       65
## 10 Grass       63.9
## 11 Ground      68.8
## 12 Ice         61.7
## 13 Normal      70.2
## 14 Poison      71.8
## 15 Psychic     65.3
## 16 Rock        76.5
## 17 Steel       66.2
## 18 Water       60.6
```

```
hp_means <- aov(hp ~ type_1, data = pokemon)
tidy(hp_means)
```

```
## # A tibble: 2 x 6
##   term         df  sumsq meansq statistic p.value
##   <chr>     <dbl>  <dbl>  <dbl>     <dbl>   <dbl>
## 1 type_1       17  7586.   446.     0.722   0.776
## 2 Residuals   132 81577.   618.    NA      NA
```

While it is clear from just the means themselves that there is some nominal difference in means by the type of pokemon, it is not statistically significant according to this one-way ANOVA test.

Then, please construct a linear model in `R` to determine the effect of pokemon type on `hp`.

```
type_model <- linear_reg() %>%
  set_engine("lm") %>%
  fit(hp ~ type_1, data = pokemon)
tidy(type_model)
```

```
## # A tibble: 18 x 5
##    term           estimate std.error statistic  p.value
##    <chr>             <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)       58.5       6.89     8.49  3.73e-14
##  2 type_1Dark        26.7      13.1      2.04  4.35e- 2
##  3 type_1Dragon      25.1      11.2      2.25  2.64e- 2
##  4 type_1Electric    14.3      11.7      1.23  2.21e- 1
##  5 type_1Fairy       15.0      14.2      1.05  2.94e- 1
##  6 type_1Fighting    12.2      11.2      1.09  2.76e- 1
##  7 type_1Fire        17.2      11.7      1.47  1.43e- 1
##  8 type_1Flying      20.5      25.8      0.793 4.29e- 1
##  9 type_1Ghost        6.46     13.1      0.494 6.22e- 1
## 10 type_1Grass        5.40      9.16     0.590 5.56e- 1
## 11 type_1Ground      10.3      13.1      0.784 4.34e- 1
## 12 type_1Ice          3.18     11.7      0.272 7.86e- 1
## 13 type_1Normal      11.7       8.95     1.30  1.94e- 1
## 14 type_1Poison      13.2      14.2      0.929 3.54e- 1
## 15 type_1Psychic      6.79      9.42     0.721 4.72e- 1
## 16 type_1Rock        18.0      12.3      1.46  1.46e- 1
## 17 type_1Steel        7.71     14.2      0.543 5.88e- 1
## 18 type_1Water        2.06      9.42     0.219 8.27e- 1
```

Interpret the meaning of your estimates and write a brief description below. Are any types missing?

One type is missing - bug, the first alphabetically and lowest in terms of hp, which reflects our intercept. We expect a bug type pokemon to have an hp of 58.5, and we expect the hp of each other type of pokemon to increase above that by its `estimate` given in the table.