Fitting Many Linear Models: Takeaways

by Dataquest Labs, Inc. - All rights reserved © 2020

Syntax

EVALUATING BIVARIATE RELATIONSHIPS

- Build scatterplot of response and predictor variable for all groups in a categorical variable:

```
library(ggplot2)
ggplot(data = df,
       aes(x = predictor, y = response)) +
  geom_point() +
  scale_y_continuous(labels = scales::comma) +
  facet_wrap(~ categorical_variable, ncol = 2)
```

- Generate tidy dataframe of coefficient-related statistics with confidence intervals:

```
library(broom)
tidy(x = lm_fit, conf.int = TRUE)
```

- Generate tidy dataframe of linear model summary statistics:

```
glance(x = lm_fit)
```

- Augment dataframe with linear model statistics:

```
augment(x = lm_fit, data = df)
```

- Create a nested dataframe:

```
library(tidyr)
library(dplyr)
df_nested <- df %>%
  group_by(categorical_variable) %>%
  nest()
```

- Generate many linear models using a nested dataframe:

```
library(tidyr)
library(dplyr)
df_nested <- df %>%
  group_by(categorical_variable) %>%
  nest() %>%
  mutate(linear_model = map(.x = data,
                        .f = ~lm(response ~ predictor,
                        data = .)))
```

- Generate list-column of tidy coefficients summaries with confidence intervals:

```
df_nested <- df %>%
  group_by(categorical_variable) %>%
  nest() %>%
  mutate(linear_model = map(.x = data,
                        .f = ~lm(response ~ predictor,
                        data = .))) %>%
  mutate(tidy_coefficients = map(.x = linear_model,
                              .f = tidy,
                              conf.int = TRUE))
```

- Unnest list-column of tidy coefficients summaries to return a tidy dataframe:

```
tidy_coefficients <- df_nested %>%
  select(categorical_variable, tidy_coefficients) %>%
  unnest(cols = tidy_coefficients)
```

- Filter tidied coefficients dataframe to return slope estimate:

```
slope <- tidy_coefficients %>%
  filter(term == "predictor_variable") %>%
  arrange(estimate)
```

- Generate list-column of tidy summary statistics with broom glance:

```
df_nested <- df %>%
  group_by(categorical_variable) %>%
  nest() %>%
  mutate(linear_model = map(.x = data,
                        .f = ~lm(response ~ predictor,
                        data = .))) %>%
  mutate(tidy_summary_stats = map(.x = linear_model,
                              .f = glance))
```

- Unnest list-column of tidy summary statistics to return a tidy dataframe:

```
df_summary_stats <- df_nested %>%
  select(categorical_variable, tidy_summary_stats) %>%
  unnest(cols = tidy_summary_stats)
```

- Augment many nested dataframes with linear model statistics:

```
df_nested <- bdf %>%
  group_by(categorical_variable) %>%
  nest() %>%
  mutate(linear_model = map(.x = data,
                            .f = ~lm(response ~ predictor,
                                     data = .))) %>%
  mutate(data_augmented = map2(.x = linear_model,
                          .y = data,
                          .f = augment))
```

- Unnest many augmented dataframes to return a single dataframe:

```
df_augmented <- df_nested %>%
  select(categorical_variable, data_augmented) %>%
  unnest(data_augmented)
```

Concepts

- **Nested data (nesting):** Nesting is performed with the function nest() from the tidyverse tidyr package. Nesting creates a "list-column" of data frames or model objects. These list-columns exist in a single dataframe that has one row per group, or category. The dataframe contains a special list-column "data" where *each observation is itself a dataframe*. This dataframe may also contain nested model objects where each observation contains regression statistics specific to the associated nested dataframe.
- **Unnested data (unnesting):** The unnest() function flattens a list-column variable in to a regular dataframe. This can be used to return a single tidy dataframe that includes tidy coefficient summaries for many models, or tidy summary statistics for many models.

When the augment() function is used unnest() returns a single dataframe that has been augmented with regression statistics specific to each categorical variable in the dataset.

- **List-column:** List-columns are variables where each observation is a list of lists. These list-columns can contain nested dataframes or model objects. List-columns are useful data structures because they enable us to iterate over each observation in a dataframe with map() and apply a function like lm() or tidy().

Resources

- The broom package on the tidyverse website.
- Vignette on the broom package.
- Vignette on the broom and dplyr package.
- The broom package on GitHub.
- Vignette on nested dataframes.
- Chapter on Many Models from Hadley Wickham's book R for Data Science.