UNIVERSITY of WEST FLORIDA

Categorical Predictors

Statistics for Data Science II

Introduction

Recall the general linear model,

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k$$

Until now, we have discussed *continuous* predictors.

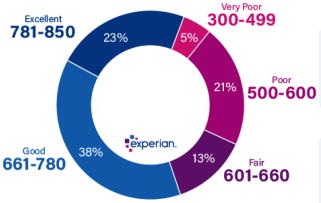
Now, let us discuss including *categorical*, or qualitative, predictors.

This means that we will include predictors that *categorize* the observations.

We can assign numbers to the categories, however, the numbers are *nominal*.

Introduction

Let us consider the categorization of credit scores by Experian [1].



If this were a predictor in our model, there would be 5 categories:

Very Poor, Poor, Fair, Good, Excellent

Introduction

If there are c classes in a categorical predictor, we include $c - 1$ in the model.

In the credit score example, there are $c = 5$ categories.

Thus, we would include $c - 1 = 4$ predictors in the model.

Dummy coding:

The $c - 1$ predictors included in the model will be binary indicators for category.

$$x_i = \begin{cases} 1 & \text{if category } i \\ 0 & \text{if another category} \end{cases}$$

Note that there are other types of coding, including effect coding.

Creating Dummy Variables

Continuing the credit score example, we would define the following indicators:

$$x_{VP} = \begin{cases} 1 & \text{if credit} = \text{“Very Poor”} \\ 0 & \text{otherwise} \end{cases} \qquad x_{P} = \begin{cases} 1 & \text{if credit} = \text{“Poor”} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{F} = \begin{cases} 1 & \text{if credit} = \text{“Fair”} \\ 0 & \text{otherwise} \end{cases} \qquad x_{G} = \begin{cases} 1 & \text{if credit} = \text{“Good”} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{E} = \begin{cases} 1 & \text{if credit} = \text{“Excellent”} \\ 0 & \text{otherwise} \end{cases}$$

This can be done through the `fastDummies` package in R.

Creating Dummy Variables

Example

Consider the data from the `palmerpenguin` package. Let's create a dataset with the variables `species`, `bill_length_mm`, and `flipper_length_mm`.

```
data <- as_tibble(penguins %>% select(species,
                                      bill_length_mm,
                                      flipper_length_mm))
```

This code creates a tibble from the `penguins` dataset that contains only the columns `species`, `bill_length_mm`, and `flipper_length_mm`.

Creating Dummy Variables

Example (Continued)

If we examine the first few observations of the tibble,

```
## # A tibble: 344 x 3
##   species bill_length_mm flipper_length_mm
##   <fct>            <dbl>              <int>
## 1 Adelie            39.1                181
## 2 Adelie            39.5                186
## 3 Adelie            40.3                195
## 4 Adelie            NA                   NA
## 5 Adelie            36.7                193
## # ... with 339 more rows
```

Creating Dummy Variables

Example (Continued)

How many species are contained in the dataset?

```
data %>% count(species)
```

```
## # A tibble: 3 x 2
##   species       n
##   <fct>     <int>
## 1 Adelie      152
## 2 Chinstrap    68
## 3 Gentoo      124
```

Creating Dummy Variables

Example (Continued)

We now use the `dummy_cols()` function in the `fastDummies` package to create our dummy variables

```
data <- dummy_cols(data, select_columns = "species")

colnames(data)
```

```
## [1] "species"           "bill_length_mm"    "flipper_length_mm"
## [4] "species_Adelie"    "species_Chinstrap" "species_Gentoo"
```

Modeling

We represent a categorical variable with c classes with $c - 1$ dummy variables in a model.

The last dummy variable not included is called the *reference group*.

How do we choose a reference group?

- It depends on the story being told / what is of interest.
- It does not affect the usefulness of the model, only the interpretations.

Modeling

Example

In the penguin data, suppose we want to model bill length as a function of flipper length and species. Let us use Adelies as the reference group.

```
m1 <- lm(bill_length_mm ~ flipper_length_mm + species_Chinstrap +
         species_Gentoo, data = data)

coefficients(m1)
```

```
##       (Intercept) flipper_length_mm species_Chinstrap    species_Gentoo
##        -2.0585910         0.2150524         8.7800997         2.8568909
```

Modeling

Example

What if we use Chinstraps as the reference group?

```
m2 <- lm(bill_length_mm ~ flipper_length_mm + species_Adelie +
         species_Gentoo, data = data)

coefficients(m2)
```

```
##       (Intercept) flipper_length_mm    species_Adelie    species_Gentoo
##         6.7215087         0.2150524        -8.7800997        -5.9232088
```

Modeling

Example:

Looking at these models side by side,

```
coefficients(m1)
```

```
##       (Intercept) flipper_length_mm species_Chinstrap    species_Gentoo
##        -2.0585910         0.2150524         8.7800997         2.8568909
```

```
coefficients(m2)
```

```
##       (Intercept) flipper_length_mm    species_Adelie    species_Gentoo
##         6.7215087         0.2150524        -8.7800997        -5.9232088
```

Interpretations

As stated before, the dummy variable not included is the *reference group*.

> $\hat{\beta}_i$ is the difference between group i and the reference group, after adjusting for all other terms in the model.

The y-intercept of a model is the average outcome when all predictors are equal to 0.

> Thus, the y-intercept is the average of the reference group, when all terms in the model are 0.

Interpretations

Example:

Recall the model created with the penguin data,

$$\hat{y} = -2.06 + 0.22x_{\text{flipper}} + 8.78x_{\text{Chinstrap}} + 2.86x_{\text{Gentoo}}$$

For a 1 mm increase in flipper length, bill length increases by 0.22 mm.

Chinstraps, on average, have bills 8.78 mm longer than Adelies.

Gentoos, on average, have bills 2.86 mm longer than Adelies.

When flipper length is 0 mm, Adelies have an average bill length of -2.06 mm.

Testing for Significance

Recall that we can test the significance of a predictor using the t test that is output by the `summary()` function.

Before we can talk about the individual t tests for categorical predictors, we must learn about the "global" test for significance.

We will use ANOVA to determine if there is overall significance of the categorical predictor and we will use the t test as a posthoc test.

i.e., ANOVA: $\beta_{c_1} = \beta_{c_2} = \ldots = 0$ and t-test: $\beta_{c_i} = \mu_{c_i} - \mu_{c_{\text{ref}}} = 0$.

Testing for Significance

To perform the global test for significance, we will construct two models:

M1 (full): model including the categorical predictor

M2 (reduced): model without the categorical predictor

Example:

```
full <- lm(bill_length_mm ~ flipper_length_mm + species_Chinstrap +
          species_Gentoo, data = data)

reduced <- lm(bill_length_mm ~ flipper_length_mm, data = data)
```

Testing for Significance

Then, we will use the `anova()` function to compare the two models.

Example:

```
anova(reduced, full)
```

```
## Analysis of Variance Table
##
## Model 1: bill_length_mm ~ flipper_length_mm
## Model 2: bill_length_mm ~ flipper_length_mm + species_Chinstrap + species_Gentoo
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    340 5787.8
## 2    338 2278.3  2    3509.4 260.32 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Testing for Significance

Hypotheses

H_0 : $\beta_{\text{Chinstrap}} = \beta_{\text{Gentoo}} = 0$

H_1 : at least one $\beta_i \neq 0$, $i = \{\text{Chinstrap, Gentoo}\}$

Test Statistic and p-Value

$F_0 = 260.32$ ($p < 0.001$)

Rejection Region

Reject H_0 if $p < \alpha$; $\alpha = 0.05$.

Conclusion / Interpretation

Reject H_0. Species is a significant predictor of bill length.

Testing for Significance

If ANOVA does not show significance: **do not** look at t tests for pairwise comparisons.

If ANOVA shows significance: **can** look at t tests for pairwise comparisons.

Note that this puts us back in the world of multiple testing.

Recall that we should adjust to avoid inflation of the type I error (α).

Bonferroni correction:

$$\alpha_B = \frac{\alpha}{k}$$

where k is the number of tests being performed

Testing for Significance

Example:

If we want to look at only two comparisons, we will use $\alpha_B = 0.05/2 = 0.025$.

If we want to look at all three comparisons, we will use $\alpha_B = 0.05/3 = 0.017$.

```
summary(m1)[[4]]
```

```
##                     Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)       -2.0585910  4.0385503 -0.5097351 6.105697e-01
## flipper_length_mm  0.2150524  0.0212316 10.1288820 3.117940e-21
## species_Chinstrap  8.7800997  0.3991228 21.9984916 3.394765e-67
## species_Gentoo     2.8568909  0.6586091  4.3377638 1.902820e-05
```

Testing for Significance

Example:

```
summary(m1)[[4]]
```

```
##                     Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)       -2.0585910  4.0385503 -0.5097351 6.105697e-01
## flipper_length_mm  0.2150524  0.0212316 10.1288820 3.117940e-21
## species_Chinstrap  8.7800997  0.3991228 21.9984916 3.394765e-67
## species_Gentoo     2.8568909  0.6586091  4.3377638 1.902820e-05
```

```
summary(m2)[[4]]
```

```
##                     Estimate Std. Error    t value     Pr(>|t|)
## (Intercept)        6.7215087  4.1695509   1.612046 1.078852e-01
## flipper_length_mm  0.2150524  0.0212316  10.128882 3.117940e-21
## species_Adelie    -8.7800997  0.3991228 -21.998492 3.394765e-67
## species_Gentoo    -5.9232088  0.5997202  -9.876620 2.226626e-20
```

Confidence Intervals

Like in the t test, the confidence intervals for β_{c_i} for categorical predictors correspond to confidence intervals for $\mu_{c_i} - \mu_{c_{\text{ref}}}$.

We can request confidence intervals from the `confint()` function.

Note that by default, `confint()` returns the 95% confidence interval.

We can change this using the `level` option.

e.g., `confint(m1, level = 0.72)`

Confidence Intervals

Example:

Looking at confidence intervals for the penguin data,

```
confint(m1)
```

```
##                         2.5 %    97.5 %
## (Intercept)       -10.0024490 5.8852671
## flipper_length_mm   0.1732897 0.2568151
## species_Chinstrap   7.9950222 9.5651771
## species_Gentoo      1.5614019 4.1523799
```

Confidence Intervals

Example:

```
confint(m1)
```

```
##                          2.5 %    97.5 %
## (Intercept)        -10.0024490 5.8852671
## flipper_length_mm    0.1732897 0.2568151
## species_Chinstrap    7.9950222 9.5651771
## species_Gentoo       1.5614019 4.1523799
```

```
confint(m2)
```

```
##                          2.5 %     97.5 %
## (Intercept)        -1.4800285 14.9230458
## flipper_length_mm   0.1732897  0.2568151
## species_Adelie     -9.5651771 -7.9950222
## species_Gentoo     -7.1028628 -4.7435548
```

Visualizing the Model

Including categorical predictors in the model varies the y-intercept.

Example:

```
coefficients(m1)
```

```
##       (Intercept) flipper_length_mm species_Chinstrap    species_Gentoo
##        -2.0585910         0.2150524         8.7800997         2.8568909
```

$$\hat{y}_{\text{Chinstraps}} = 6.72 + 0.22x_{\text{flipper}}$$

$$\hat{y}_{\text{Gentoos}} = 0.80 + 0.22x_{\text{flipper}}$$

$$\hat{y}_{\text{Adelies}} = -2.06 + 0.22x_{\text{flipper}}$$

Visualizing the Model

This also means that we can plot the separate regression lines on a graph.

Example:

We use the coefficients to create the predicted value, $\hat{y}$,

```
c1 <- coefficients(m1)
```

```
##       (Intercept) flipper_length_mm species_Chinstrap    species_Gentoo
##        -2.0585910         0.2150524         8.7800997         2.8568909
```

```
data <- data %>% mutate(
          predAdelie = c1[[1]] + c1[[2]]*flipper_length_mm,
          predChinstrap = c1[[1]] + c1[[2]]*data$flipper_length_mm + c1[[3]],
          predGentoo = c1[[1]] + c1[[2]]*data$flipper_length_mm + c1[[4]]
        )
```

Visualizing the Model

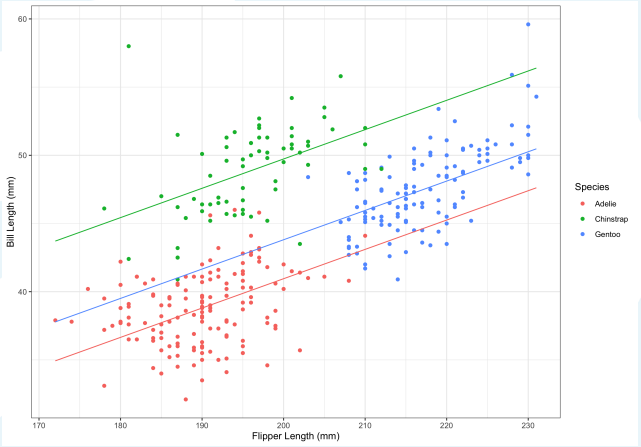
Example:

The following code will create the graph with the different regression lines:

```
p <- ggplot(data, aes(x=flipper_length_mm, y=bill_length_mm, color=Species)) +
  geom_point() +
  geom_line(aes(y = predAdelie), color = "#F8766D", linetype = "solid") +
  geom_line(aes(y = predChinstrap), color = "#00BA38", linetype = "solid") +
  geom_line(aes(y = predGentoo), color = "#619CFF", linetype = "solid") +
  xlab("Flipper Length (mm)") +
  ylab("Bill Length (mm)") +
  theme_bw()
```

Visualizing the Model

Example:



Future Considerations

As we increase the number of terms in our model, visualizations can quickly become more complicated.

If there are multiple continuous predictors:

- We allow only one to vary on the x axis.
- We plug in specific values for the ones we are holding constant.
  - e.g., the mean, median, or "known" value of interest

Future Considerations

As we increase the number of terms in our model, visualizations can quickly become more complicated.

If there are multiple categorical predictors:

- We can create lines for every combination.
  - e.g., male Adelies, female Adelies, male Chinstraps, etc.
- We can hold some constant and create lines for the others.
  - e.g., the graph itself is for males, then have lines for species.

Future Considerations

What if there are a ton of categories?

e.g., state

We need to make sure there are enough observations in each category to warrant inclusion in the model.

If not enough observations, we can condense categories.

e.g., Likert scale: combine "agree" and "strongly agree"

We also need to keep interpretability/generalizability in mind.