Linear Regression with Categorical Variables

Nate Wells

Math 141, 2/21/22

Outline

In this lecture, we will...

Outline

In this lecture, we will. . .

- Create linear models with binary categorical explanatory variables
- Extend linear models to include arbitrary categorical explanatory variables

Section 1

Regression for Binary Categorical Variables

Overview of Regression for a Categorical Variable

- **Simple linear regression** model a linear relationship between two **quantitative** variables.

$$\hat{Y} = \beta_0 + \beta_1 X$$

Overview of Regression for a Categorical Variable

- **Simple linear regression** model a linear relationship between two **quantitative** variables.

$$\hat{Y} = \beta_0 + \beta_1 X$$

- **General Linear Regression** is a more flexible class of models that take the form:

$$\hat{Y} = \beta_0 + \beta_1 f_1(X_1) + \beta_2 f_2(X_2) + \cdots + \beta_p f_p(X_p)$$

where p is the number of variables present, $f_1, \ldots, f_p$ are functions of those variables, and $\beta_0, \beta_1, \ldots, \beta_p$ are fixed constants.

Overview of Regression for a Categorical Variable

- **Simple linear regression** model a linear relationship between two **quantitative** variables.

$$\hat{Y} = \beta_0 + \beta_1 X$$

- **General Linear Regression** is a more flexible class of models that take the form:

$$\hat{Y} = \beta_0 + \beta_1 f_1(X_1) + \beta_2 f_2(X_2) + \cdots + \beta_p f_p(X_p)$$

where p is the number of variables present, $f_1, \ldots, f_p$ are functions of those variables, and $\beta_0, \beta_1, \ldots, \beta_p$ are fixed constants.

- General linear regression requires a quantitative response variable, but allows us to:

Overview of Regression for a Categorical Variable

- **Simple linear regression** model a linear relationship between two **quantitative** variables.

$$\hat{Y} = \beta_0 + \beta_1 X$$

- **General Linear Regression** is a more flexible class of models that take the form:

$$\hat{Y} = \beta_0 + \beta_1 f_1(X_1) + \beta_2 f_2(X_2) + \cdots + \beta_p f_p(X_p)$$

  where p is the number of variables present, $f_1, \ldots, f_p$ are functions of those variables, and $\beta_0, \beta_1, \ldots, \beta_p$ are fixed constants.

- General linear regression requires a quantitative response variable, but allows us to:
  - Use either quantitative or categorical explanatory variables
  - Simultaneously include multiple explanatory variables
  - Model non-linear relationships between explanatory and response variables.

Overview of Regression for a Categorical Variable

- **Simple linear regression** model a linear relationship between two **quantitative** variables.

$$\hat{Y} = \beta_0 + \beta_1 X$$

- **General Linear Regression** is a more flexible class of models that take the form:

$$\hat{Y} = \beta_0 + \beta_1 f_1(X_1) + \beta_2 f_2(X_2) + \cdots + \beta_p f_p(X_p)$$

  where p is the number of variables present, $f_1, \ldots, f_p$ are functions of those variables, and $\beta_0, \beta_1, \ldots, \beta_p$ are fixed constants.

- General linear regression requires a quantitative response variable, but allows us to:
  - Use either quantitative or categorical explanatory variables
  - Simultaneously include multiple explanatory variables
  - Model non-linear relationships between explanatory and response variables.
- Today, we'll focus on just the first extension above: *using categorical explanatory variables.*

Caffeine Consumption

- Suppose we are interested in whether a 9am or 10am section of Math 141 consumes more caffeine on a typical day:

Caffeine Consumption

- Suppose we are interested in whether a 9am or 10am section of Math 141 consumes more caffeine on a typical day:
  - We can treat caffeine consumption in `mg` as the (quantitative) response variable, and `section` as the (categorical) explanatory variable.

Caffeine Consumption

- Suppose we are interested in whether a 9am or 10am section of Math 141 consumes more caffeine on a typical day:
    - We can treat caffeine consumption in `mg` as the (quantitative) response variable, and `section` as the (categorical) explanatory variable.
- We record caffeine consumption for 42 students:

Caffeine Consumption

- Suppose we are interested in whether a 9am or 10am section of Math 141 consumes more caffeine on a typical day:
  - We can treat caffeine consumption in `mg` as the (quantitative) response variable, and `section` as the (categorical) explanatory variable.
- We record caffeine consumption for 42 students:

```
## # A tibble: 42 x 2
##    section    mg
##    <fct>   <dbl>
##  1 10am        0
##  2 9am       200
##  3 10am        0
##  4 10am      600
##  5 10am      120
##  6 9am       400
##  7 9am       275
##  8 10am      100
##  9 10am      200
## 10 10am      175
## # ... with 32 more rows
```

Caffeine Consumption

- Suppose we are interested in whether a 9am or 10am section of Math 141 consumes more caffeine on a typical day:
  - We can treat caffeine consumption in `mg` as the (quantitative) response variable, and `section` as the (categorical) explanatory variable.

- We record caffeine consumption for 42 students:

```
## # A tibble: 42 x 2
##    section    mg
##    <fct>   <dbl>
##  1 10am        0
##  2 9am       200
##  3 10am        0
##  4 10am      600
##  5 10am      120
##  6 9am       400
##  7 9am       275
##  8 10am      100
##  9 10am      200
## 10 10am      175
## # ... with 32 more rows
```

- And compute relevant statistics:

Caffeine Consumption

- Suppose we are interested in whether a 9am or 10am section of Math 141 consumes more caffeine on a typical day:
  - We can treat caffeine consumption in `mg` as the (quantitative) response variable, and `section` as the (categorical) explanatory variable.

- We record caffeine consumption for 42 students:

```
## # A tibble: 42 x 2
##    section    mg
##    <fct>   <dbl>
##  1 10am        0
##  2 9am       200
##  3 10am        0
##  4 10am      600
##  5 10am      120
##  6 9am       400
##  7 9am       275
##  8 10am      100
##  9 10am      200
## 10 10am      175
## # ... with 32 more rows
```

- And compute relevant statistics:

```
caffeine %>% group_by(section) %>%
  summarize(
    mean_score = mean(mg),
    sd_score = sd(mg),
    n = n() )
```

```
## # A tibble: 2 x 4
##   section mean_score sd_score     n
##   <fct>        <dbl>    <dbl> <int>
## 1 9am           193.     177.    21
## 2 10am          157.     174.    21
```

Caffeine Consumption

- Suppose we are interested in whether a 9am or 10am section of Math 141 consumes more caffeine on a typical day:
  - We can treat caffeine consumption in `mg` as the (quantitative) response variable, and `section` as the (categorical) explanatory variable.

- We record caffeine consumption for 42 students:

```
## # A tibble: 42 x 2
##    section    mg
##    <fct>   <dbl>
##  1 10am        0
##  2 9am       200
##  3 10am        0
##  4 10am      600
##  5 10am      120
##  6 9am       400
##  7 9am       275
##  8 10am      100
##  9 10am      200
## 10 10am      175
## # ... with 32 more rows
```

- And compute relevant statistics:

```
caffeine %>% group_by(section) %>%
  summarize(
    mean_score = mean(mg),
    sd_score = sd(mg),
    n = n() )
```

```
## # A tibble: 2 x 4
##   section mean_score sd_score     n
##   <fct>        <dbl>    <dbl> <int>
## 1 9am           193.     177.    21
## 2 10am          157.     174.    21
```

- Note that mean consumption is higher in the 9am section (but not much higher relative to standard deviation)

Caffeine Consumption

- Suppose we are interested in whether a 9am or 10am section of Math 141 consumes more caffeine on a typical day:
  - We can treat caffeine consumption in `mg` as the (quantitative) response variable, and `section` as the (categorical) explanatory variable.

- We record caffeine consumption for 42 students:

```
## # A tibble: 42 x 2
##    section    mg
##    <fct>   <dbl>
##  1 10am        0
##  2 9am       200
##  3 10am        0
##  4 10am      600
##  5 10am      120
##  6 9am       400
##  7 9am       275
##  8 10am      100
##  9 10am      200
## 10 10am      175
## # ... with 32 more rows
```

- And compute relevant statistics:

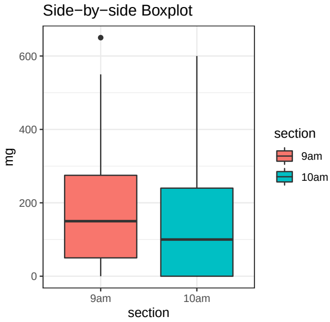
```
caffeine %>% group_by(section) %>%
  summarize(
    mean_score = mean(mg),
    sd_score = sd(mg),
    n = n() )
```

```
## # A tibble: 2 x 4
##   section mean_score sd_score     n
##   <fct>        <dbl>    <dbl> <int>
## 1 9am           193.     177.    21
## 2 10am          157.     174.    21
```

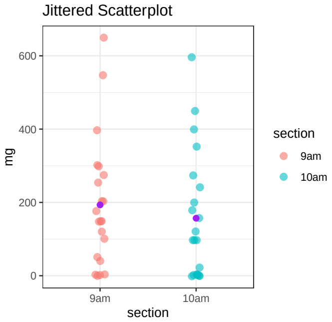
- Note that mean consumption is higher in the 9am section (but not much higher relative to standard deviation)

Visualizations

- Since the response is quantitative, and the explanatory is categorical, we can visualize either with side-by-side boxplots or with a jittered scatterplot:



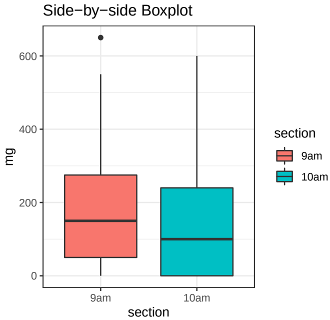
Horizontal bars indicate median mg for each section



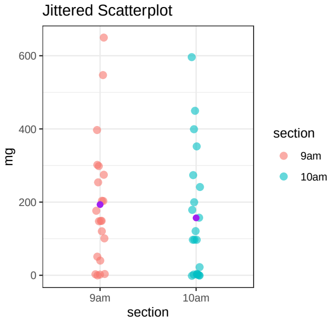
Purple dots indicate mean mg for each section

Visualizations

- Since the response is quantitative, and the explanatory is categorical, we can visualize either with side-by-side boxplots or with a jittered scatterplot:



Horizontal bars indicate median mg for each section



Purple dots indicate mean mg for each section

- Advantages of each type of plot?

Recoding Binary Variables

- A linear model for `mg` as a function of `section` is problematic:

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section}$$

Recoding Binary Variables

- A linear model for `mg` as a function of `section` is problematic:

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section}$$

  - `section` is categorical, so we can't add or multiply its values to get a number

Recoding Binary Variables

- A linear model for `mg` as a function of `section` is problematic:

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section}$$

  - `section` is categorical, so we can't add or multiply its values to get a number
  - But there is a relatively easy fix!

Recoding Binary Variables

- A linear model for `mg` as a function of `section` is problematic:

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section}$$

  - `section` is categorical, so we can't add or multiply its values to get a number
  - But there is a relatively easy fix!
- We can **recode** the levels of `section` as a numeric **indicator** variable

Recoding Binary Variables

- A linear model for `mg` as a function of `section` is problematic:

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section}$$

  - `section` is categorical, so we can't add or multiply its values to get a number
  - But there is a relatively easy fix!

- We can **recode** the levels of `section` as a numeric **indicator** variable

```
caffeine <- caffeine %>% mutate(
  section_10am = ifelse(section == "10am", 1, 0) )
```

Recoding Binary Variables

- A linear model for `mg` as a function of `section` is problematic:

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section}$$

  - `section` is categorical, so we can't add or multiply its values to get a number
  - But there is a relatively easy fix!
- We can **recode** the levels of `section` as a numeric **indicator** variable

```
caffeine <- caffeine %>% mutate(
  section_10am = ifelse(section == "10am", 1, 0) )
```

```
## # A tibble: 42 x 3
##    section_10am section    mg
##           <dbl> <fct>   <dbl>
##  1            0 9am       300
##  2            0 9am       175
##  3            0 9am       150
##  4            0 9am       300
##  5            0 9am         0
##  6            1 10am       25
##  7            0 9am       200
##  8            1 10am      275
##  9            0 9am       200
## 10            1 10am      100
## # ... with 32 more rows
```

- The variable `section_10am` takes the value...
  - 1, if a student is in the 10am section
  - 0, if a student is in the 9am section

Recoding Binary Variables

- A linear model for `mg` as a function of `section` is problematic:

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section}$$

  - `section` is categorical, so we can't add or multiply its values to get a number
  - But there is a relatively easy fix!

- We can **recode** the levels of `section` as a numeric **indicator** variable

```
caffeine <- caffeine %>% mutate(
  section_10am = ifelse(section == "10am", 1, 0) )
```

```
## # A tibble: 42 x 3
##    section_10am section    mg
##           <dbl> <fct>   <dbl>
##  1            0 9am       300
##  2            0 9am       175
##  3            0 9am       150
##  4            0 9am       300
##  5            0 9am         0
##  6            1 10am       25
##  7            0 9am       200
##  8            1 10am      275
##  9            0 9am       200
## 10            1 10am      100
## # ... with 32 more rows
```

- The variable `section_10am` takes the value...
  - 1, if a student is in the 10am section
  - 0, if a student is in the 9am section
- This choice was somewhat arbitrary.
  - We could have instead created a variable called `section_9am` that takes the value 1 if a student is in the 9am section.

Linear Models for Binary Categorical Variables

- After recoding, a linear equation is now possible:

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section\_10am}$$

Linear Models for Binary Categorical Variables

- After recoding, a linear equation is now possible:

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section\_10am}$$

Linear Models for Binary Categorical Variables

- After recoding, a linear equation is now possible:

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section\_10am}$$

- For example, suppose

$$\hat{\text{mg}} = 193 - 36 \cdot \text{section\_10am}$$

Linear Models for Binary Categorical Variables

- After recoding, a linear equation is now possible:

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section\_10am}$$

- For example, suppose

$$\hat{\text{mg}} = 193 - 36 \cdot \text{section\_10am}$$

  - If a student is in the 10am section, then `section_10am = 1`, then the model predicts

$$\hat{\text{mg}} = 193 - 36 \cdot 1 = 157$$

Linear Models for Binary Categorical Variables

- After recoding, a linear equation is now possible:

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section\_10am}$$

- For example, suppose

$$\hat{\text{mg}} = 193 - 36 \cdot \text{section\_10am}$$

  - If a student is in the 10am section, then `section_10am = 1`, then the model predicts

$$\hat{\text{mg}} = 193 - 36 \cdot 1 = 157$$

  - If a student is in the 9am section, then `section_10am = 0`, then the model predicts

$$\hat{\text{mg}} = 193 - 36 \cdot 0 = 193$$

Linear Models for Binary Categorical Variables

- After recoding, a linear equation is now possible:

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section\_10am}$$

- For example, suppose

$$\hat{\text{mg}} = 193 - 36 \cdot \text{section\_10am}$$

  - If a student is in the 10am section, then `section_10am = 1`, then the model predicts

$$\hat{\text{mg}} = 193 - 36 \cdot 1 = 157$$

  - If a student is in the 9am section, then `section_10am = 0`, then the model predicts

$$\hat{\text{mg}} = 193 - 36 \cdot 0 = 193$$

- The value of β_0 is the prediction for students *not* in the 10am section. This is the **baseline** prediction. (The baseline is 193mg)

Linear Models for Binary Categorical Variables

- After recoding, a linear equation is now possible:

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section\_10am}$$

- For example, suppose

$$\hat{\text{mg}} = 193 - 36 \cdot \text{section\_10am}$$

  - If a student is in the 10am section, then `section_10am = 1`, then the model predicts

$$\hat{\text{mg}} = 193 - 36 \cdot 1 = 157$$

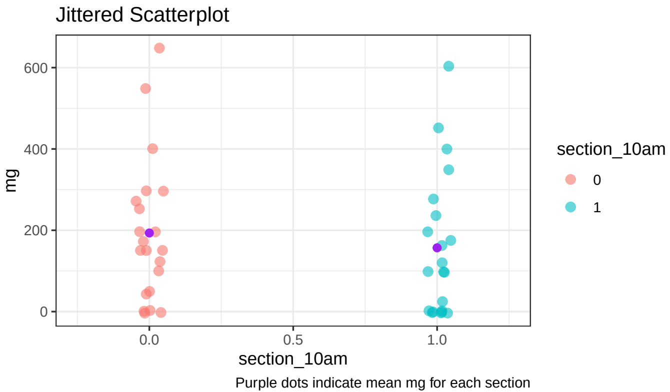
  - If a student is in the 9am section, then `section_10am = 0`, then the model predicts

$$\hat{\text{mg}} = 193 - 36 \cdot 0 = 193$$

- The value of β_0 is the prediction for students *not* in the 10am section. This is the **baseline** prediction. (The baseline is 193mg)
- The value of β_1 is the **change** in prediction for a student in the 10am section, relative to the baseline. (The change is -36mg)

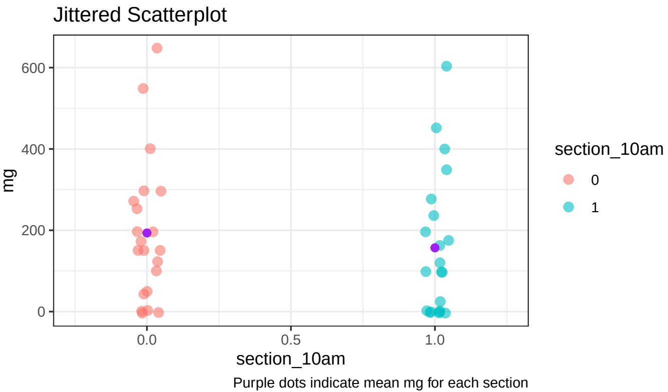
Least Squares Regression

- Consider the jittered scatterplot for `mg` and `section_10am`



Least Squares Regression

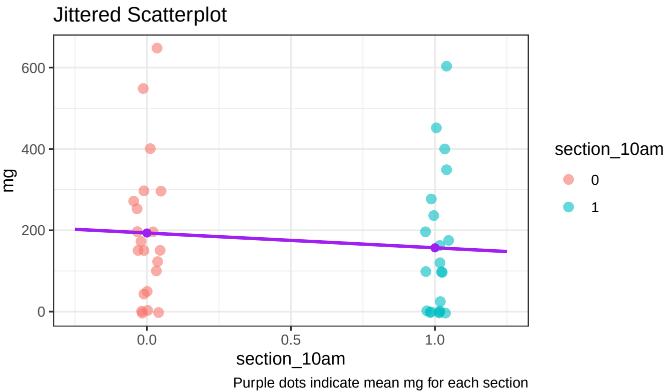
- Consider the jittered scatterplot for `mg` and `section_10am`



- Since this is a scatterplot of two *quantitative* variables, we can find the line of best fit!

Least Squares Regression

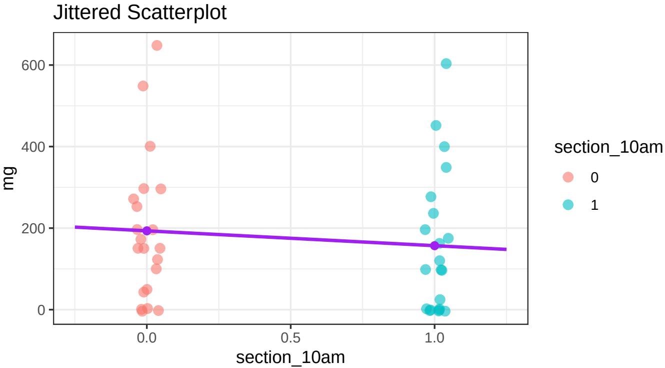
- Consider the jittered scatterplot for `mg` and `section_10am`



- Since this is a scatterplot of two *quantitative* variables, we can find the line of best fit!

Least Squares Regression

- Consider the jittered scatterplot for `mg` and `section_10am`



Purple dots indicate mean mg for each section

- The line of best fit passes through the mean `mg` in each section!

Properties of Least Squares Regression when X is binary

- In general, the slope β_1 of a regression line is the average change in Y per unit increase in X.

Properties of Least Squares Regression when X is binary

- In general, the slope β_1 of a regression line is the average change in Y per unit increase in X.
  - If X is binary, it can only take two values: 0 and 1.
  - Increasing X by 1 exactly corresponds to changing from the first level of X to the second level of X.

Properties of Least Squares Regression when X is binary

- In general, the slope β_1 of a regression line is the average change in Y per unit increase in X.
  - If X is binary, it can only take two values: 0 and 1.
  - Increasing X by 1 exactly corresponds to changing from the first level of X to the second level of X.
- The intercept β_0 of a regression line is the predicted value when $X = 0$.

Properties of Least Squares Regression when X is binary

- In general, the slope β_1 of a regression line is the average change in Y per unit increase in X.
  - If X is binary, it can only take two values: 0 and 1.
  - Increasing X by 1 exactly corresponds to changing from the first level of X to the second level of X.
- The intercept β_0 of a regression line is the predicted value when $X = 0$.
  - If X is binary, the best prediction for Y when $X = 0$ is the mean value of Y when $X = 0$

Properties of Least Squares Regression when X is binary

- In general, the slope β_1 of a regression line is the average change in Y per unit increase in X.
  - If X is binary, it can only take two values: 0 and 1.
  - Increasing X by 1 exactly corresponds to changing from the first level of X to the second level of X.
- The intercept β_0 of a regression line is the predicted value when $X = 0$.
  - If X is binary, the best prediction for Y when $X = 0$ is the mean value of Y when $X = 0$
- If Y is a quantitative response variable and X is a binary numeric variable, then the least squares regression line is

$$\hat{Y} = \beta_0 + \beta_1 X$$

Properties of Least Squares Regression when X is binary

- In general, the slope β_1 of a regression line is the average change in Y per unit increase in X.
  - If X is binary, it can only take two values: 0 and 1.
  - Increasing X by 1 exactly corresponds to changing from the first level of X to the second level of X.
- The intercept β_0 of a regression line is the predicted value when $X = 0$.
  - If X is binary, the best prediction for Y when $X = 0$ is the mean value of Y when $X = 0$
- If Y is a quantitative response variable and X is a binary numeric variable, then the least squares regression line is

$$\hat{Y} = \beta_0 + \beta_1 X$$

  - β_0 is the mean of Y when $X = 0$
  - β_1 is the difference in means of Y between when $X = 1$ and $X = 0$.
  - $\beta_0 + \beta_1$ is the mean of Y when $X = 1$.

Finding Least Squaress Line (by hand)

- Since β_0, β_1 only require us to know the mean of Y when X is 0 and 1, we can compute the least squares line by hand:

Finding Least Squaress Line (by hand)

- Since β_0, β_1 only require us to know the mean of Y when X is 0 and 1, we can compute the least squares line by hand:

```
caffeine %>% group_by(section) %>% summarize(mean = mean(mg))
```

```
## # A tibble: 2 x 2
##   section  mean
##   <fct>   <dbl>
## 1 9am      193.
## 2 10am     157.
```

Finding Least Squaress Line (by hand)

- Since β_0, β_1 only require us to know the mean of Y when X is 0 and 1, we can compute the least squares line by hand:

```
caffeine %>% group_by(section) %>% summarize(mean = mean(mg))
```

```
## # A tibble: 2 x 2
##   section  mean
##   <fct>   <dbl>
## 1 9am      193.
## 2 10am     157.
```

$$\hat{\text{mg}} = 193 - 36 \cdot \text{section\_10am} \qquad \text{Since } 193 - 157 = 36$$

Finding Least Squaress Line (using R)

- But we can also use `lm` in R.
  - R will even automatically convert binary categorical variables to numeric indicators:

Finding Least Squaress Line (using R)

- But we can also use `lm` in R.
  - R will even automatically convert binary categorical variables to numeric indicators:

```
caf_mod <- lm(mg ~ section, data = caffeine)
get_regression_table(caf_mod)
```

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	193.333	38.224	5.058	0.000	116.080	270.587
section10am	-36.429	54.057	-0.674	0.504	-145.681	72.824

Finding Least Squaress Line (using R)

- But we can also use `lm` in R.
  - R will even automatically convert binary categorical variables to numeric indicators:

```
caf_mod <- lm(mg ~ section, data = caffeine)
get_regression_table(caf_mod)
```

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	193.333	38.224	5.058	0.000	116.080	270.587
section10am	-36.429	54.057	-0.674	0.504	-145.681	72.824

- But R made a choice here about which level to code as a 0 and which to code as a 1
  - It coded the 9am section as 0 and the 10am section as 1 (how do I know?)

Finding Least Squaress Line (using R)

- But we can also use `lm` in R.
    - R will even automatically convert binary categorical variables to numeric indicators:

```
caf_mod <- lm(mg ~ section, data = caffeine)
get_regression_table(caf_mod)
```

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	193.333	38.224	5.058	0.000	116.080	270.587
section10am	-36.429	54.057	-0.674	0.504	-145.681	72.824

- But R made a choice here about which level to code as a 0 and which to code as a 1
    - It coded the 9am section as 0 and the 10am section as 1 (how do I know?)
    - In general, R will code the first level of a factor as 0, and the second as a 1.

Finding Least Squaress Line (using R)

- But we can also use `lm` in R.
  - R will even automatically convert binary categorical variables to numeric indicators:

```
caf_mod <- lm(mg ~ section, data = caffeine)
get_regression_table(caf_mod)
```

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	193.333	38.224	5.058	0.000	116.080	270.587
section10am	-36.429	54.057	-0.674	0.504	-145.681	72.824

- But R made a choice here about which level to code as a 0 and which to code as a 1
  - It coded the 9am section as 0 and the 10am section as 1 (how do I know?)
  - In general, R will code the first level of a factor as 0, and the second as a 1.
  - If no order is provided, it will use alphabetical order.

Finding Least Squaress Line (using R)

- But we can also use `lm` in R.
  - R will even automatically convert binary categorical variables to numeric indicators:

```
caf_mod <- lm(mg ~ section, data = caffeine)
get_regression_table(caf_mod)
```

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	193.333	38.224	5.058	0.000	116.080	270.587
section10am	-36.429	54.057	-0.674	0.504	-145.681	72.824

- But R made a choice here about which level to code as a 0 and which to code as a 1
  - It coded the 9am section as 0 and the 10am section as 1 (how do I know?)
  - In general, R will code the first level of a factor as 0, and the second as a 1.
  - If no order is provided, it will use alphabetical order.
  - If you want to change the order, you need to `mutate` the data frame using `fct_relevel`

Section 2

Linear Regression with Multi-level Categorical Variables

More Classes

- Suppose we also have data on caffeine consumption from a 3rd section of math 141.

```
## # A tibble: 63 x 2
##    section      mg
##    <fct>     <dbl>
##  1 Nate_10am     0
##  2 Nate_9am    550
##  3 Nate_10am     0
##  4 Nate_10am     0
##  5 Nick_9am      0
##  6 Nate_9am     40
##  7 Nate_9am    400
##  8 Nick_9am    100
##  9 Nate_10am   200
## 10 Nate_9am    100
## # ... with 53 more rows
```

```
caffeine3 %>% group_by(section) %>%
  summarize(mean_mg = mean(mg), sd_mg = sd(mg))
```

```
## # A tibble: 3 x 3
##   section   mean_mg sd_mg
##   <fct>       <dbl> <dbl>
## 1 Nick_9am     195.  164.
## 2 Nate_9am     193.  177.
## 3 Nate_10am    157.  174.
```

- Goal: Create a linear model that takes `section` as input and returns a predicted `mg` as output.

Multi-level Model, First Attempt

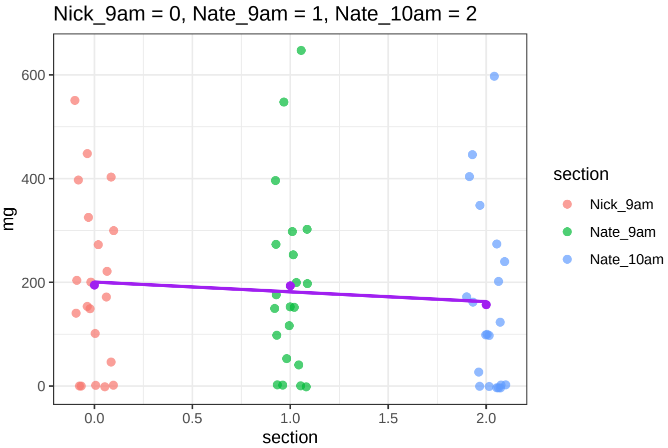
- We could try to recode levels by converting to the integers 0, 1 and 2.

Multi-level Model, First Attempt

- We could try to recode levels by converting to the integers 0, 1 and 2.
    - But this would be very problematic for creating the line of best fit. Why?

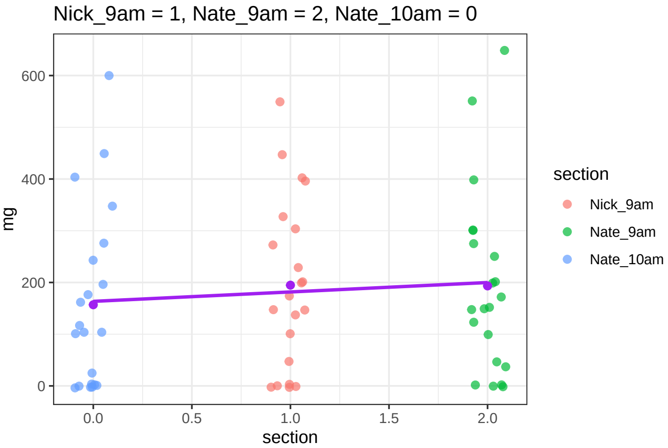
Multi-level Model, First Attempt

- We could try to recode levels by converting to the integers 0, 1 and 2.
  - But this would be very problematic for creating the line of best fit. Why?



Multi-level Model, First Attempt

- We could try to recode levels by converting to the integers 0, 1 and 2.
  - But this would be very problematic for creating the line of best fit. Why?



One-Hot Encoding

- Instead of defining a single numeric variable to encode all levels, we need a binary indicator variable *for each level*:

One-Hot Encoding

- Instead of defining a single numeric variable to encode all levels, we need a binary indicator variable *for each level*:
  - `section_Nick_9am` is 1 if the student is in Nick's 9am section, and 0 otherwise.

One-Hot Encoding

- Instead of defining a single numeric variable to encode all levels, we need a binary indicator variable *for each level*:
  - `section_Nick_9am` is 1 if the student is in Nick's 9am section, and 0 otherwise.
  - `section_Nate_9am` is 1 if the student is in Nate's 9am section, and 0 otherwise.

One-Hot Encoding

- Instead of defining a single numeric variable to encode all levels, we need a binary indicator variable *for each level*:
    - `section_Nick_9am` is 1 if the student is in Nick's 9am section, and 0 otherwise.
    - `section_Nate_9am` is 1 if the student is in Nate's 9am section, and 0 otherwise.
    - `section_Nate_10am` is 1 if the student is in Nate's 10am section, and 0 otherwise.

One-Hot Encoding

- Instead of defining a single numeric variable to encode all levels, we need a binary indicator variable *for each level*:
    - `section_Nick_9am` is 1 if the student is in Nick's 9am section, and 0 otherwise.
    - `section_Nate_9am` is 1 if the student is in Nate's 9am section, and 0 otherwise.
    - `section_Nate_10am` is 1 if the student is in Nate's 10am section, and 0 otherwise.
    - Note that for a given student, *exactly* one of these variables is 1, and the other two are 0.

One-Hot Encoding

- Instead of defining a single numeric variable to encode all levels, we need a binary indicator variable *for each level*:
  - `section_Nick_9am` is 1 if the student is in Nick's 9am section, and 0 otherwise.
  - `section_Nate_9am` is 1 if the student is in Nate's 9am section, and 0 otherwise.
  - `section_Nate_10am` is 1 if the student is in Nate's 10am section, and 0 otherwise.
  - Note that for a given student, *exactly* one of these variables is 1, and the other two are 0.

```
caffeine3 <- caffeine3 %>% mutate(
  section_Nick_9am = ifelse(section == "Nick_9am", 1, 0),
  section_Nate_9am = ifelse(section == "Nate_9am", 1, 0),
  section_Nate_10am = ifelse(section == "Nate_10am", 1, 0))
```

```
## # A tibble: 63 x 5
##   section   section_Nick_9am section_Nate_9am section_Nate_10am    mg
##   <fct>                <dbl>            <dbl>             <dbl> <dbl>
## 1 Nick_9am                 1                0                 0   140
## 2 Nate_10am                0                0                 1   600
## 3 Nate_10am                0                0                 1     0
## 4 Nick_9am                 1                0                 0   150
## 5 Nate_9am                 0                1                 0    50
## # ... with 58 more rows
```

Multi-level Model, Second Attempt

- We can define a multivariate linear model for `mg` as a function of `section` by

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section\_Nate\_9am} + \beta_2 \cdot \text{section\_Nate\_10am}$$

Multi-level Model, Second Attempt

- We can define a multivariate linear model for `mg` as a function of `section` by

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section\_Nate\_9am} + \beta_2 \cdot \text{section\_Nate\_10am}$$

- For example, suppose

$$\hat{\text{mg}} = 195 - 1.5 \cdot \text{section\_Nate\_9am} - 38 \cdot \text{section\_Nate\_10am}$$

Multi-level Model, Second Attempt

- We can define a multivariate linear model for `mg` as a function of `section` by

$$\hat{mg} = \beta_0 + \beta_1 \cdot \text{section\_Nate\_9am} + \beta_2 \cdot \text{section\_Nate\_10am}$$

- For example, suppose

$$\hat{mg} = 195 - 1.5 \cdot \text{section\_Nate\_9am} - 38 \cdot \text{section\_Nate\_10am}$$

- What is the predicted caffeine consumption for a student in ...
    - Nate's 9am section?
    - Nate's 10am section?
    - Nick's 9am section?

Multi-level Model, Second Attempt

- We can define a multivariate linear model for `mg` as a function of `section` by

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section\_Nate\_9am} + \beta_2 \cdot \text{section\_Nate\_10am}$$

- For example, suppose

$$\hat{\text{mg}} = 195 - 1.5 \cdot \text{section\_Nate\_9am} - 38 \cdot \text{section\_Nate\_10am}$$

- What is the predicted caffeine consumption for a student in . . .
  - Nate's 9am section?
  - Nate's 10am section?
  - Nick's 9am section?
- Where did the indicator for Nick's 9am section go in the formula for the model???

Multi-level Model, Second Attempt

- We can define a multivariate linear model for `mg` as a function of `section` by

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section\_Nate\_9am} + \beta_2 \cdot \text{section\_Nate\_10am}$$

- For example, suppose

$$\hat{\text{mg}} = 195 - 1.5 \cdot \text{section\_Nate\_9am} - 38 \cdot \text{section\_Nate\_10am}$$

- What is the predicted caffeine consumption for a student in ...
  - Nate's 9am section?
  - Nate's 10am section?
  - Nick's 9am section?
- Where did the indicator for Nick's 9am section go in the formula for the model???
  - Nick's 9am section is treated as the **baseline**, and so does not need its own indicator.

Multi-level Model, Second Attempt

- We can define a multivariate linear model for `mg` as a function of `section` by

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section\_Nate\_9am} + \beta_2 \cdot \text{section\_Nate\_10am}$$

- For example, suppose

$$\hat{\text{mg}} = 195 - 1.5 \cdot \text{section\_Nate\_9am} - 38 \cdot \text{section\_Nate\_10am}$$

- What is the predicted caffeine consumption for a student in ...
  - Nate's 9am section?
  - Nate's 10am section?
  - Nick's 9am section?
- Where did the indicator for Nick's 9am section go in the formula for the model???
  - Nick's 9am section is treated as the **baseline**, and so does not need its own indicator.
  - The intercept is the prediction for the baseline.

Multi-level Model, Second Attempt

- We can define a multivariate linear model for `mg` as a function of `section` by

$$\hat{\text{mg}} = \beta_0 + \beta_1 \cdot \text{section\_Nate\_9am} + \beta_2 \cdot \text{section\_Nate\_10am}$$

- For example, suppose

$$\hat{\text{mg}} = 195 - 1.5 \cdot \text{section\_Nate\_9am} - 38 \cdot \text{section\_Nate\_10am}$$

- What is the predicted caffeine consumption for a student in ...
  - Nate's 9am section?
  - Nate's 10am section?
  - Nick's 9am section?
- Where did the indicator for Nick's 9am section go in the formula for the model???
  - Nick's 9am section is treated as the **baseline**, and so does not need its own indicator.
  - The intercept is the prediction for the baseline.
  - Slopes on the other indicator variables correspond to differences from the baseline.

Multi-level Linear Model in R

- As with quantitative $\sim$ quantitative, and quantitative $\sim$ binary, we can use the `lm` function to create linear models for quantitative $\sim$ multilevel in R

Multi-level Linear Model in R

- As with quantitative ∼ quantitative, and quantitative ∼ binary, we can use the `lm` function to create linear models for quantitative ∼ multilevel in R

```
caf_mod3 <- lm(mg ~ section, data = caffeine3)
get_regression_table(caf_mod3)
```

Multi-level Linear Model in R

- As with quantitative ~ quantitative, and quantitative ~ binary, we can use the `lm` function to create linear models for quantitative ~ multilevel in R

```
caf_mod3 <- lm(mg ~ section, data = caffeine3)
get_regression_table(caf_mod3)
```

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	194.762	37.431	5.203	0.000	119.890	269.634
sectionNate_9am	-1.429	52.935	-0.027	0.979	-107.314	104.457
sectionNate_10am	-37.857	52.935	-0.715	0.477	-143.742	68.028

Multi-level Linear Model in R

- As with quantitative ~ quantitative, and quantitative ~ binary, we can use the `lm` function to create linear models for quantitative ~ multilevel in R

```
caf_mod3 <- lm(mg ~ section, data = caffeine3)
get_regression_table(caf_mod3)
```

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	194.762	37.431	5.203	0.000	119.890	269.634
sectionNate_9am	-1.429	52.935	-0.027	0.979	-107.314	104.457
sectionNate_10am	-37.857	52.935	-0.715	0.477	-143.742	68.028

- Let's compare to some statistics we've already computed:

section	mean_mg	diff_from_baseline
Nick_9am	194.7619	0.000000
Nate_9am	193.3333	-1.428571
Nate_10am	156.9048	-37.857143

Multi-level Linear Model in R

- As with quantitative ~ quantitative, and quantitative ~ binary, we can use the `lm` function to create linear models for quantitative ~ multilevel in R

```
caf_mod3 <- lm(mg ~ section, data = caffeine3)
get_regression_table(caf_mod3)
```

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	194.762	37.431	5.203	0.000	119.890	269.634
sectionNate_9am	-1.429	52.935	-0.027	0.979	-107.314	104.457
sectionNate_10am	-37.857	52.935	-0.715	0.477	-143.742	68.028

- Let's compare to some statistics we've already computed:

section	mean_mg	diff_from_baseline
Nick_9am	194.7619	0.000000
Nate_9am	193.3333	-1.428571
Nate_10am	156.9048	-37.857143

- The intercept is the mean value of the response for the baseline level.

Multi-level Linear Model in R

- As with quantitative ∼ quantitative, and quantitative ∼ binary, we can use the `lm` function to create linear models for quantitative ∼ multilevel in R

```
caf_mod3 <- lm(mg ~ section, data = caffeine3)
get_regression_table(caf_mod3)
```

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	194.762	37.431	5.203	0.000	119.890	269.634
sectionNate_9am	-1.429	52.935	-0.027	0.979	-107.314	104.457
sectionNate_10am	-37.857	52.935	-0.715	0.477	-143.742	68.028

- Let's compare to some statistics we've already computed:

section	mean_mg	diff_from_baseline
Nick_9am	194.7619	0.000000
Nate_9am	193.3333	-1.428571
Nate_10am	156.9048	-37.857143

- The intercept is the mean value of the response for the baseline level.
- The slopes are the difference in mean values between the indicated level and the baseline.

Residuals for Multi-level Models

- As with simple linear regression for quantitative $\sim$ quantitative, we can get residuals for each observation:

Residuals for Multi-level Models

- As with simple linear regression for quantitative $\sim$ quantitative, we can get residuals for each observation:

```
get_regression_points(caf_mod3)
```

Residuals for Multi-level Models

- As with simple linear regression for quantitative ~ quantitative, we can get residuals for each observation:

```
get_regression_points(caf_mod3)
```

```
## # A tibble: 63 x 5
##       ID    mg section   mg_hat residual
##    <int> <dbl> <fct>      <dbl>    <dbl>
##  1    57   225 Nick_9am    195.     30.2
##  2     4   150 Nate_9am    193.    -43.3
##  3    39     0 Nate_10am   157.   -157.
##  4     1   550 Nate_9am    193.    357.
##  5    34     0 Nate_10am   157.   -157.
##  6    23     0 Nate_10am   157.   -157.
##  7    43     0 Nick_9am    195.   -195.
##  8    14    40 Nate_9am    193.   -153.
##  9    18   400 Nate_9am    193.    207.
## 10    51   100 Nick_9am    195.    -94.8
## # ... with 53 more rows
```

Residuals for Multi-level Models

- As with simple linear regression for quantitative ~ quantitative, we can get residuals for each observation:

```
get_regression_points(caf_mod3)
```

```
## # A tibble: 63 x 5
##       ID    mg section   mg_hat residual
##    <int> <dbl> <fct>      <dbl>    <dbl>
##  1    57   225 Nick_9am    195.     30.2
##  2     4   150 Nate_9am    193.    -43.3
##  3    39     0 Nate_10am   157.   -157.
##  4     1   550 Nate_9am    193.    357.
##  5    34     0 Nate_10am   157.   -157.
##  6    23     0 Nate_10am   157.   -157.
##  7    43     0 Nick_9am    195.   -195.
##  8    14    40 Nate_9am    193.   -153.
##  9    18   400 Nate_9am    193.    207.
## 10    51   100 Nick_9am    195.    -94.8
## # ... with 53 more rows
```

- Recall, residuals are the difference between the observed and predicted values

Residuals for Multi-level Models

- As with simple linear regression for quantitative ~ quantitative, we can get residuals for each observation:

```
get_regression_points(caf_mod3)
```

```
## # A tibble: 63 x 5
##       ID    mg section   mg_hat residual
##    <int> <dbl> <fct>      <dbl>    <dbl>
##  1    57   225 Nick_9am    195.     30.2
##  2     4   150 Nate_9am    193.    -43.3
##  3    39     0 Nate_10am   157.   -157.
##  4     1   550 Nate_9am    193.    357.
##  5    34     0 Nate_10am   157.   -157.
##  6    23     0 Nate_10am   157.   -157.
##  7    43     0 Nick_9am    195.   -195.
##  8    14    40 Nate_9am    193.   -153.
##  9    18   400 Nate_9am    193.    207.
## 10    51   100 Nick_9am    195.    -94.8
## # ... with 53 more rows
```

- Recall, residuals are the difference between the observed and predicted values
- Here, residual tells us the difference between a student's actual mg consumed and the mean mg for that student's class.