GETTING STARTED

Install the `janitor` package from GitHub:
`devtools::install_github("sfirke/janitor")`

CHRISTOPHER PRENER, PH.D.

FALL, 2017
LECTURE 17

QUANTITATIVE ANALYSIS

CHI-SQUARE

AGENDA

1. Front Matter
2. Chi-square Test Theory
3. Contingency Tables in `R`
4. Chi-square in `R`
5. Back Matter

1 FRONT MATTER

ANNOUNCEMENTS

Lab-16 and **all final project** materials are due by next Monday (12/18) at 4:00pm

Final project submissions on GitHub should include code and pdfs of handout, slides, and (if necessary) your paper.

Final grades available by end of business on Wednesday, 12/20

We have not hit the required response rate for course evals!

2 CHI-SQUARE TEST THEORY

IT'S BEEN ALL ABOUT THE MEAN

$$\bar{x} = \frac{\sum_{i=1}^{n} x}{n}$$

$$s = \sqrt{\frac{\sum_{i=1}^{n} (x - \bar{x})^2}{n - 1}}$$

IT'S BEEN ALL ABOUT THE MEAN

$$\bar{x} = \frac{\sum_{i=1}^{n} x}{n}$$

$$t = \frac{\bar{X}_a - \bar{X}_b}{\sqrt{\frac{s_p^2}{n_a} + \frac{s_p^2}{n_b}}}$$

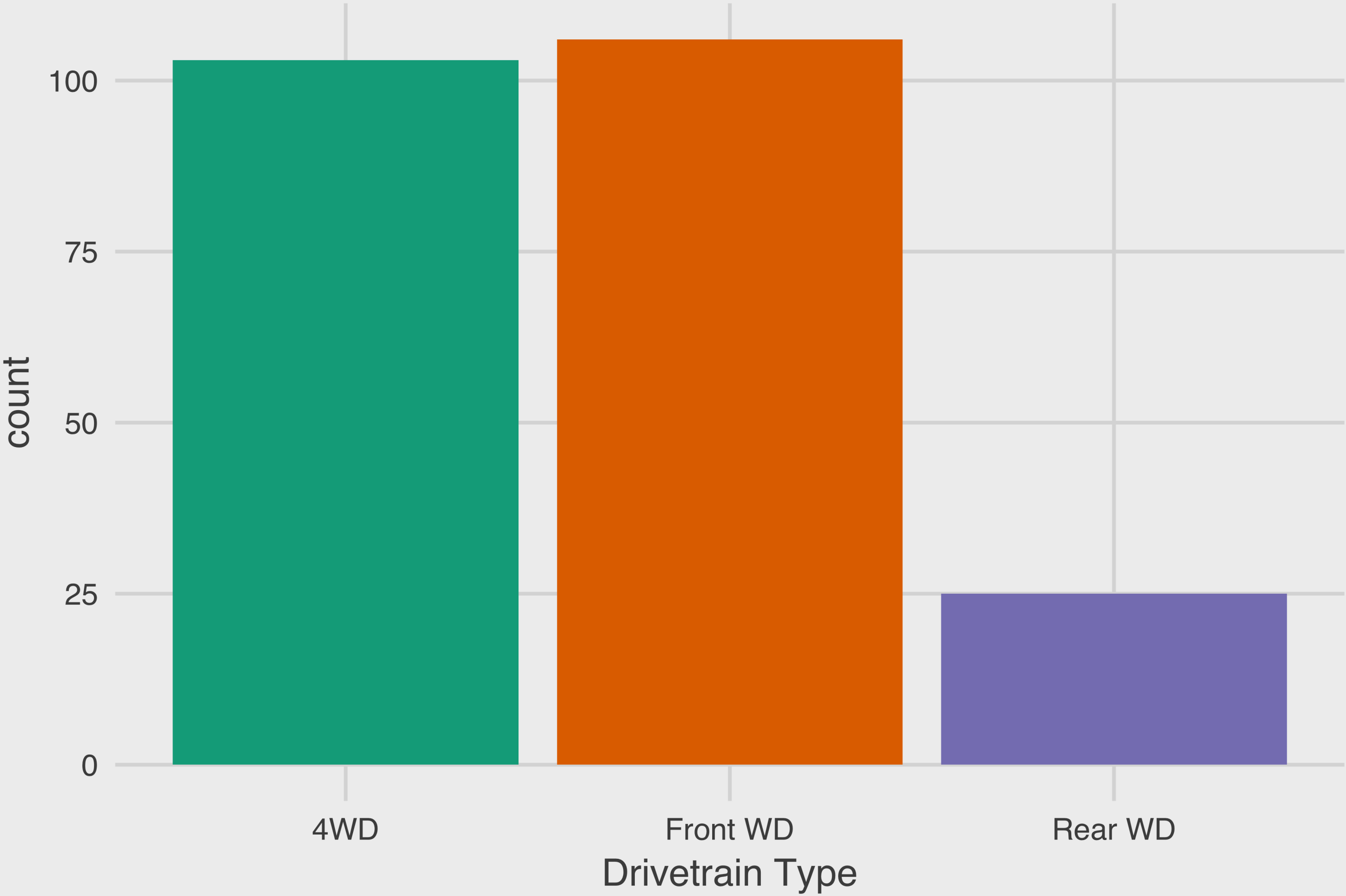
IT'S BEEN ALL ABOUT THE MEAN

$$\bar{x} = \frac{\sum_{i=1}^{n} x}{n}$$

$$r = \frac{\sum_{i=1}^{n} (x - \bar{x})(y - \bar{y})}{(n-1)s_x s_y}$$

Drivetrain Variable from ggplot2's mpg Data

Original factor with mean of 1.667

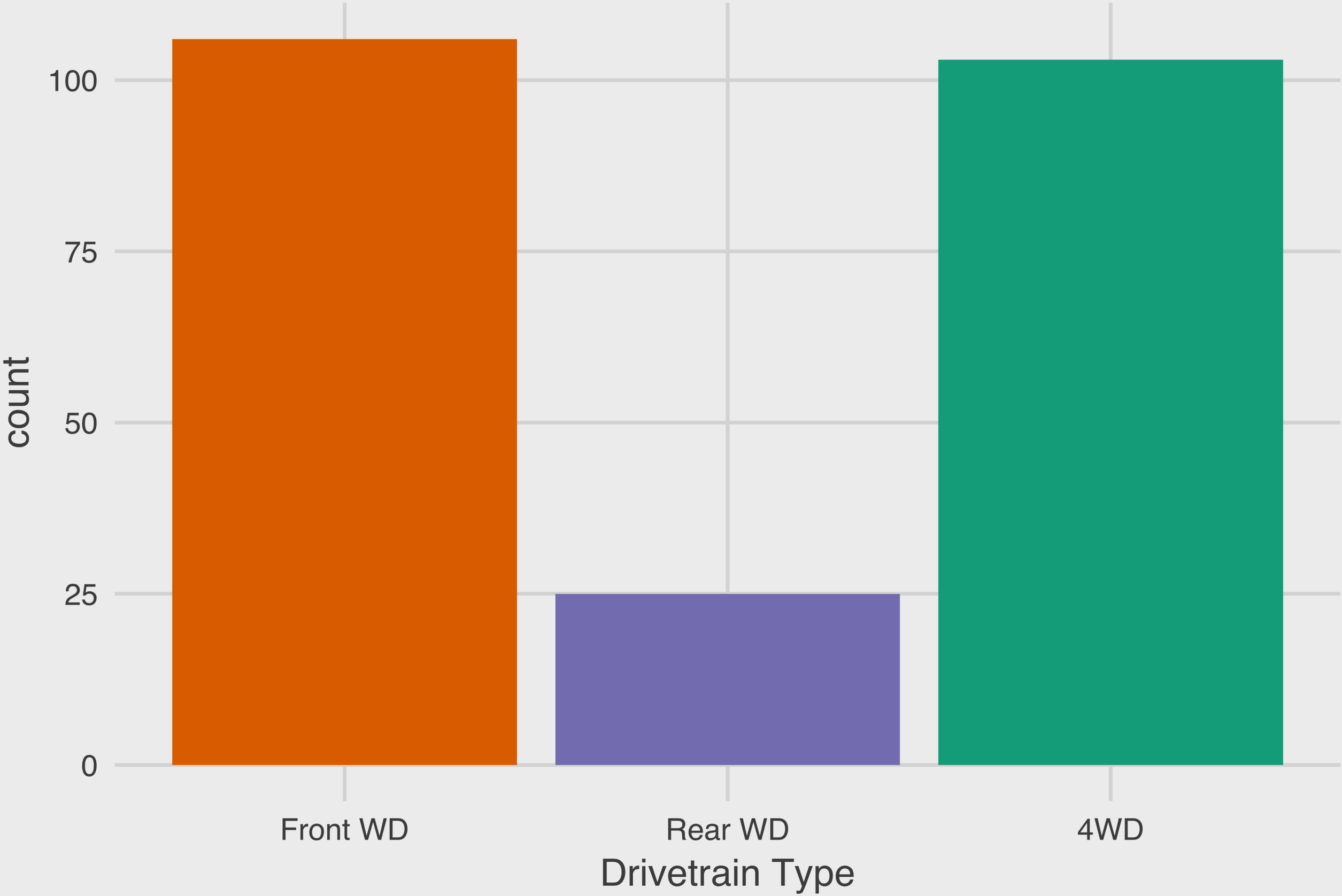


Drivetrain Variable from ggplot2's mpg Data

Original factor with mean of 1.667

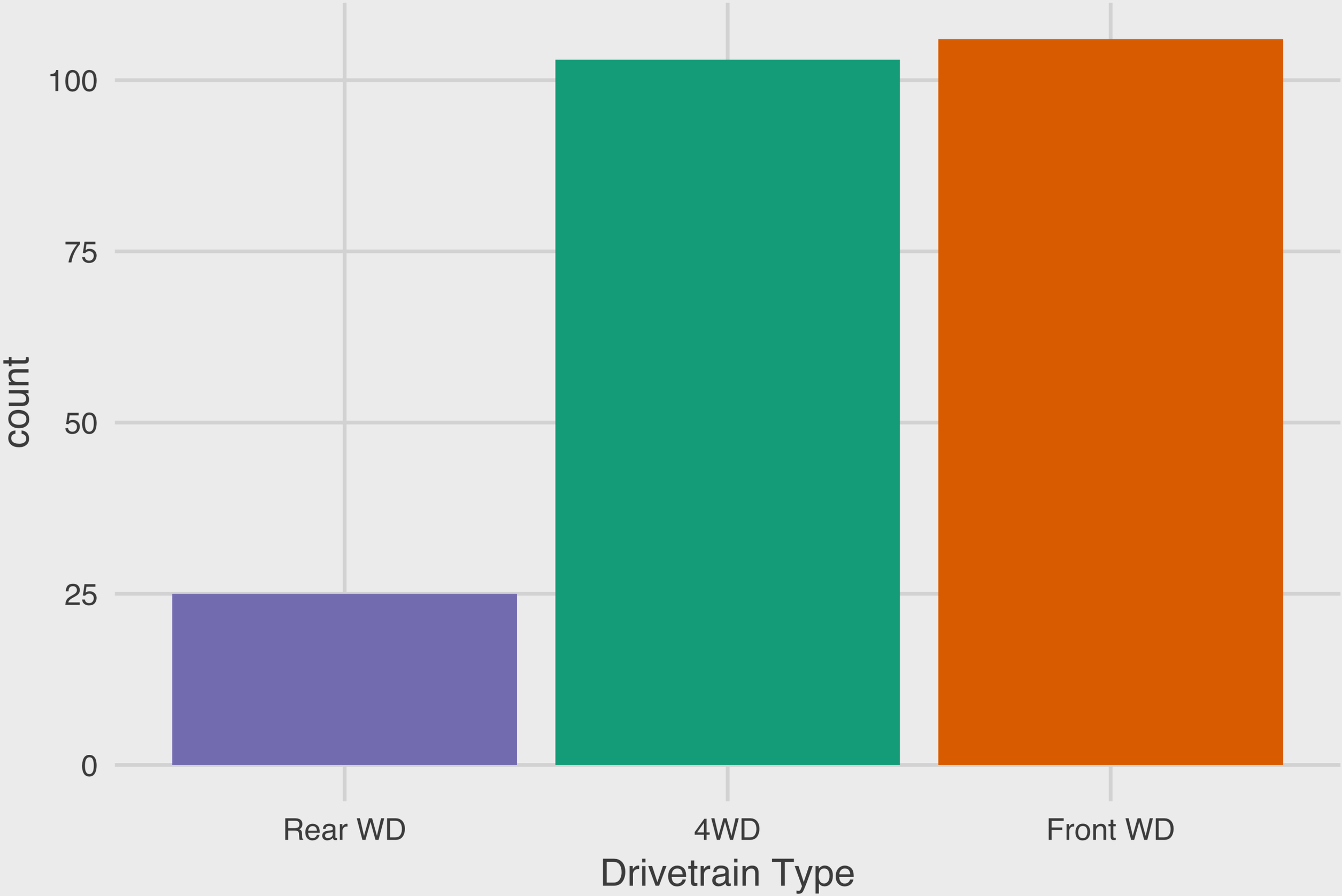
Drivetrain Variable from ggplot2's mpg Data

Re-leveled factor with mean of 1.987



Drivetrain Variable from ggplot2's mpg Data

Re-leveled factor with mean of 2.346



3 CONTINGENCY TABLES IN R

ONE-WAY TABLES IN R

f(x) `tabyl(.data, varName)`

Parameters:

- `.dat`
- `varN` `r`, or `char`

Available in `janitor`
Download via GitHub (for now)

ONE-WAY TABLES IN R

`tabyl(.data, varName)`

Parameters:

- `.data` is a data frame or table (can be used in a pipe)
- `varName` is the variable name you want to analyze (numeric, factor, or character)

ONE-WAY TABLES IN R

f(x) `tabyl(.data, varName)`

Using the `cyl` variable from `ggplot2`'s `mpg` data:

```
> tabyl(mpg, cyl)
 cyl  n    percent
   4 81 0.34615385
   5  4 0.01709402
   6 79 0.33760684
   8 70 0.29914530
```

ONE-WAY TABLES IN R

f(x) `tabyl(.data, varName)`

Using the `cyl` variable from `ggplot2`'s `mpg` data in a pipe:

```
> mpg %>%
+   tabyl(cyl)
 cyl  n    percent
   4 81 0.34615385
   5  4 0.01709402
   6 79 0.33760684
   8 70 0.29914530
```

TWO-WAY TABLES IN R

f(x) `tabyl(.data, xvar, yvar)`

Parameters:

- `.data` is a data frame or table (can be used in a pipe)
- `xvar` is the variable name of the first variable you want to analyze (numeric, factor, or character)
- `yvar` is the variable name of the second variable you want to analyze (numeric, factor, or character)

TWO-WAY TABLES IN R

f(x) `tabyl(.data, xvar, yvar)`

Using the `cyl` and `drv` variables from `ggplot2`'s `mpg` data:

```
> tabyl(mpg, cyl, drv)
 cyl  4  f  r
   4 23 58  0
   5  0  4  0
   6 32 43  4
   8 48  1 21
```

ADDING TOTALS

f(x) `adorn_totals(where = position)`

Parameters:

- `position` is one of "`row`" (for row totals), "`col`" (for column totals), or both combined together with the concatenate function.

ADDING TOTALS

f(x) `adorn_totals(where = position)`

Using the `cyl` and `drv` variables from `ggplot2`'s `mpg` data:

```
> mpg %>%
+   tabyl(cyl, drv) %>%
+   adorn_totals(where = ‘‘row’’)
```

Should be used in a pipeline after the `tabyl()` function but before any other adornment functions!

ADDING TOTALS

```
> mpg %>%
+   tabyl(cyl, drv) %>%
+   adorn_totals(where = "row")
   cyl   4   f   r
     4  23  58  0
     5   0   4  0
     6  32  43  4
     8  48   1 21
 Total 103 106 25
```

ADDING TOTALS

```
> mpg %>%
+   tabyl(cyl, drv) %>%
+   adorn_totals(where = "col")
 cyl  4  f  r Total
   4 23 58  0    81
   5  0  4  0     4
   6 32 43  4    79
   8 48  1 21    70
```

ADDING TOTALS

```
> mpg %>%
+   tabyl(cyl, drv) %>%
+   adorn_totals(where = c(“row”, “col”))
   cyl   4   f   r Total
     4  23  58   0    81
     5   0   4   0     4
     6  32  43   4    79
     8  48   1  21    70
 Total 103 106  25   234
```

ADDING PERCENTAGES

f(x) `adorn_percentages(denominator = pctType)`

Parameters:

- `pctType` is one of ‘‘`row`’’ (for row percents), ‘‘`col`’’ (for column percents), or ‘‘`all`’’ (for all percentages).

ADDING PERCENTAGES

f(x) `adorn_percentages(denominator = pctType)`

Using the `cyl` and `drv` variables from `ggplot2`'s `mpg` data:

```
> mpg %>%
+   tabyl(cyl, drv) %>%
+   adorn_totals(where = c(“row”, “col”)) %>%
+   adorn_percentages(denominator = “row”)
```

Should be used in a pipeline after the `tabyl()` function and `adorn_totals()` (if used)!

ADDING PERCENTAGES

```
> mpg %>%
+   tabyl(cyl, drv) %>%
+   adorn_totals(where = c(“row”, “col”)) %>%
+   adorn_percentages(denominator = “row”)
```

```
   cyl         4          f          r Total
     4 0.2839506 0.71604938 0.00000000     1
     5 0.0000000 1.00000000 0.00000000     1
     6 0.4050633 0.54430380 0.05063291     1
     8 0.6857143 0.01428571 0.30000000     1
 Total 0.4401709 0.45299145 0.10683761     1
```

FORMATTING PERCENTAGES

f(x) `adorn_pct_formatting(digits = val)`

Parameters:

- `val` is the number of significant digits you want your percentage values rounded to.

FORMATTING PERCENTAGES

f(x) `adorn_pct_formatting(digits = val)`

Using the `cyl` and `drv` variables from `ggplot2`'s `mpg` data:

```
> mpg %>%
+   tabyl(cyl, drv) %>%
+   adorn_totals(where = c("row", "col")) %>%
+   adorn_percentages(denominator = "row") %>%
+   adorn_pct_formatting(digits = 3)
```

Should be used in a pipeline after the `tabyl()` function and `adorn_totals()` (if used)!

ADDING PERCENTAGES

```
> mpg %>%
+   tabyl(cyl, drv) %>%
+   adorn_totals(where = c(“row”, “col”)) %>%
+   adorn_percentages(denominator = “row”) %>%
+   adorn_pct_formatting(digits = 3)
   cyl       4        f       r    Total
     4 28.395%  71.605%  0.000% 100.000%
     5  0.000% 100.000%  0.000% 100.000%
     6 40.506%  54.430%  5.063% 100.000%
     8 68.571%   1.429% 30.000% 100.000%
 Total 44.017%  45.299% 10.684% 100.000%
```

ADDING FREQUENCIES

f(x) `adorn_ns(position = position)`

Parameters:

- `position` refers to the placement of the frequency values; can either be "`front`" or "`rear`".

ADDING FREQUENCIES

f(x)

`adorn_ns(position = *position*)`

Using the `cyl` and `drv` variables from `ggplot2`'s `mpg` data:

```
> mpg %>%
+   tabyl(cyl, drv) %>%
+   adorn_totals(where = c(“row”, “col”)) %>%
+   adorn_percentages(“row”) %>%
+   adorn_pct_formatting(digits = 3) %>%
+   adorn_ns(position = “front”)
```

Should be used in a pipeline after the `tabyl()` function and `adorn_totals()` (if used)!

ADDING PERCENTAGES

```
> mpg %>%
+   tabyl(cyl, drv) %>%
+   adorn_totals(where = c(“row”, “col”)) %>%
+   adorn_percentages(“row”) %>%
+   adorn_pct_formatting(digits = 3) %>%
+   adorn_ns(position = “front”)
   cyl              4              f              r          Total
     4  23 (28.395%)  58 (71.605%)  0 (0.000%)  81 (100.000%)
     5   0 (0.000%)    4 (100.000%)  0 (0.000%)   4 (100.000%)
     6  32 (40.506%)  43 (54.430%)  4 (5.063%)  79 (100.000%)
     8  48 (68.571%)   1 (1.429%)  21 (30.000%)  70 (100.000%)
 Total 103 (44.017%) 106 (45.299%) 25 (10.684%) 234 (100.000%)
```

CONVERTING TO LaTeX

```
> mpg %>%
+    tabyl(cyl, drv) %>%
+    adorn_totals(where = c(“row”, “col”)) %>%
+    adorn_percentages(“row”) %>%
+    adorn_pct_formatting(digits = 3) %>%
+    adorn_ns(position = “front”) -> table

> stargazer(table, title = “Cylinders by Drivetrain”,
     summary = FALSE)
```

<<<<< OUTPUT OMITTED >>>>>

4 CHI-SQUARE TEST IN R

HYPOTHESES

- H_0 is that there **is no** meaningful relationship between x and y
- H_1 is that there **is** a meaningful relationship between x and y

ASSUMPTIONS

- Discrete (nominal or ordinal) data for both x and y
- Independence between x and y
- Sample size greater than 30
- Less than 20% of cells can have an expected count of less than 5 cases, and no cell should have an expected count less than 1
    - These are known as the “Cochran conditions”
    - Cochran himself acknowledged that 5 was an arbitrary value.

CHI-SQUARE TEST

f(x) `chisq.test(`*xvar*`, `*yvar*`)`

Parameters:

- `xvar` ... be
  spec

Available in `stats`
Included in standard distributions of R

CHI-SQUARE TEST

f(x) `chisq.test(`*xvar, yvar*`)`

Parameters:

- `xvar` and `yvar` are the two variables to be tested; they must both be specified with the data frame and the dollar sign

CHI-SQUARE TEST

f(x) `chisq.test(`*xvar*`, `*yvar*`)`

Using the `cyl` and `drv` variable from `ggplot2`'s `mpg` data:

```
> chisq.test(mpg$cyl, mpg$drv)

<<<<< OUTPUT OMITTED >>>>>>
```

Can be used with numeric, factor, or character variables.

CHI-SQUARE TEST

```
> chisq.test(mpg$cyl, mpg$drv)

    Pearson's Chi-squared test

data:  mpg$cyl and mpg$drv
X-squared = 98.136, df = 6, p-value < 2.2e-16

Warning message:
In chisq.test(mpg$cyl, mpg$drv) :
  Chi-squared approximation may be incorrect
```

CHI-SQUARE TEST

```
> chisq.test(mpg$cyl, mpg$drv)

    Pearson's Chi-squared test

data:  mpg$cyl and mpg$drv
X-squared = 98.136, df = 6, p-value < 2.2e-16
```

How would you interpret this result?

CHI-SQUARE TEST

```
> chisq.test(mpg$cyl, mpg$drv)

    Pearson's Chi-squared test

data:  mpg$cyl and mpg$drv
X-squared = 98.136, df = 6, p-value < 2.2e-16
```

The chi-square test ($\chi^2 = 98.136,\ df = 6,\ p < .001$) indicates that there is substantial variation in cylinders by drive train type.

COCHRAN CONDITIONS

f(x) `model$expected`

Using the `cyl` and `drv` variable from `ggplot2`'s `mpg` data:

```
> model <- chisq.test(mpg$cyl, mpg$drv)
> model$expected
```

<<<<< OUTPUT OMITTED >>>>>

Execute the chi-squared test twice, once to get the standard output and once to check the expected frequencies.

COCHRAN CONDITIONS

```
> model <- chisq.test(mpg$cyl, mpg$drv)
Warning message:
In chisq.test(mpg$cyl, mpg$drv) :
  Chi-squared approximation may be incorrect
```

```
> model$expected
        mpg$drv
mpg$cyl         4         f         r
      4 35.653846 36.692308 8.6538462
      5  1.760684  1.811966 0.4273504
      6 34.773504 35.786325 8.4401709
      8 30.811966 31.709402 7.4786325
```

COCHRAN CONDITIONS

```
> model <- chisq.test(mpg$cyl, mpg$drv)
Warning message:
In chisq.test(mpg$cyl, mpg$drv) :
  Chi-squared approximation may be incorrect
```

```
> model$expected
        mpg$drv
mpg$cyl         4         f         r
      4 35.653846 36.692308 8.6538462
      5  1.760684  1.811966 0.4273504
      6 34.773504 35.786325 8.4401709
      8 30.811966 31.709402 7.4786325
```

3 of the 12 cells (or 25%) are less 5, and 1 cell is less than one, violating Cochran's conditions

COCHRAN CONDITIONS

```
> model <- chisq.test(mpg$cyl, mpg$drv)
Warning message:
In chisq.test(mpg$cyl, mpg$drv) :
  Chi-squared approximation may be incorrect
```

```
> model$expected
        mpg$drv
mpg$cyl         4         f         r
      4 35.653846 36.692308 8.6538462
      5  1.760684  1.811966 0.4273504
      6 34.773504 35.786325 8.4401709
      8 30.811966 31.709402 7.4786325
```

FISHER'S EXACT TEST

f(x) `fisher.test(`*xvar*`, `*yvar*`, simulate.p.value = TRUE)`

Parameters:

- `xvar` and `yvar` are the two variables to be tested; they must both be specified with the data frame and the dollar sign
- `simulate.p.value` uses a Monte Carlo simulation process to find the best p-value; the alternative (if `FALSE`) is far more computationally consuming (in terms of time and computer processing power)

FISHER'S EXACT TEST

f(x) `fisher.test(`*xvar*`,` *yvar*`, simulate.p.value = TRUE)`

Using the `cyl` and `drv` variable from `ggplot2`'s `mpg` data:

```
> fisher.test(mpg$cyl, mpg$drv, simulate.p.value =
    TRUE)
```

<<<<< OUTPUT OMITTED >>>>>

Use this test to fine the p-value if the Cochran conditions are not met.

FISHER'S EXACT TEST

```
> fisher.test(mpg$cyl, mpg$drv, simulate.p.value = TRUE)

   Fisher's Exact Test for Count Data with simulated p-value (based
on 2000 replicates)

data:  mpg$cyl and mpg$drv
p-value = 0.0004998
alternative hypothesis: two.sided
```

5 BACK MATTER

WHAT WE COVERED TODAY

2. Chi-square Test Theory

3. Contingency Tables in `R`

4. Chi-square in `R`

REMINDERS

Lab-16 and **all final project** materials are due by next Monday (12/18) at 4:00pm

Final project submissions on GitHub should include code and pdfs of handout, slides, and (if necessary) your paper.

Final grades available by end of business on Wednesday, 12/20

We have not hit the required response rate for course evals!