Tidy data and data wrangling

Dr. Maria Tackett

09.05.19

Click for PDF of slides

Announcements

- Lab 01 due **TODAY at 11:59p**
  - Your lab must be on GitHub!
- Reading 02 on the course schedule - due Thursday

Check in

- Any questions on material from last time?

Identifying variables

Number of variables involved

- Univariate data analysis: distribution of single variable
- Bivariate data analysis: relationship between two variables
- Multivariate data analysis: relationship between many variables at once, usually focusing on the relationship between two while conditioning for others

Types of variables

- Numerical variables can be classified as continuous or discrete based on whether or not the variable can take on an infinite number of values or only non-negative whole numbers, respectively.
  - *height* is continuous
  - *number of siblings* is discrete
- If the variable is categorical, we can determine if it is ordinal based on whether or not the levels have a natural ordering.
  - *hair color* is unordered
  - *year in school* is ordinal

Visualizing numerical data

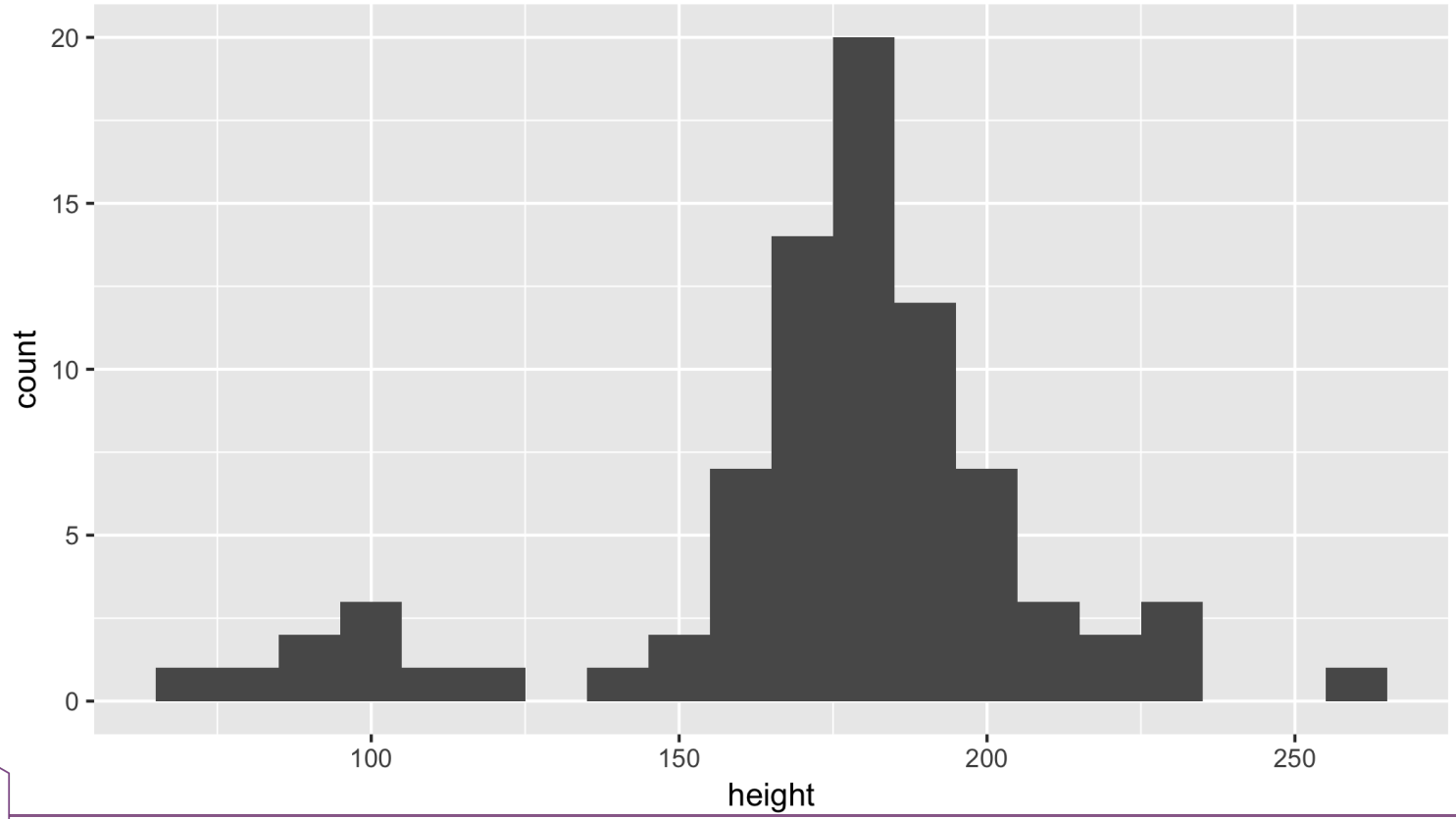
Describing shapes of numerical distributions

- **shape:**
  - skewness: right-skewed, left-skewed, symmetric (skew is to the side of the longer tail)
  - modality: unimodal, bimodal, multimodal, uniform
- **center:** mean (`mean`), median (`median`), mode (not always useful)
- **spread:** range (`range`), standard deviation (`sd`), inter-quartile range (`IQR`)
- **outliers**: observations outside of the usual pattern

Histograms

```
ggplot(data = starwars, mapping = aes(x = height)) +
  geom_histogram(binwidth = 10)
```

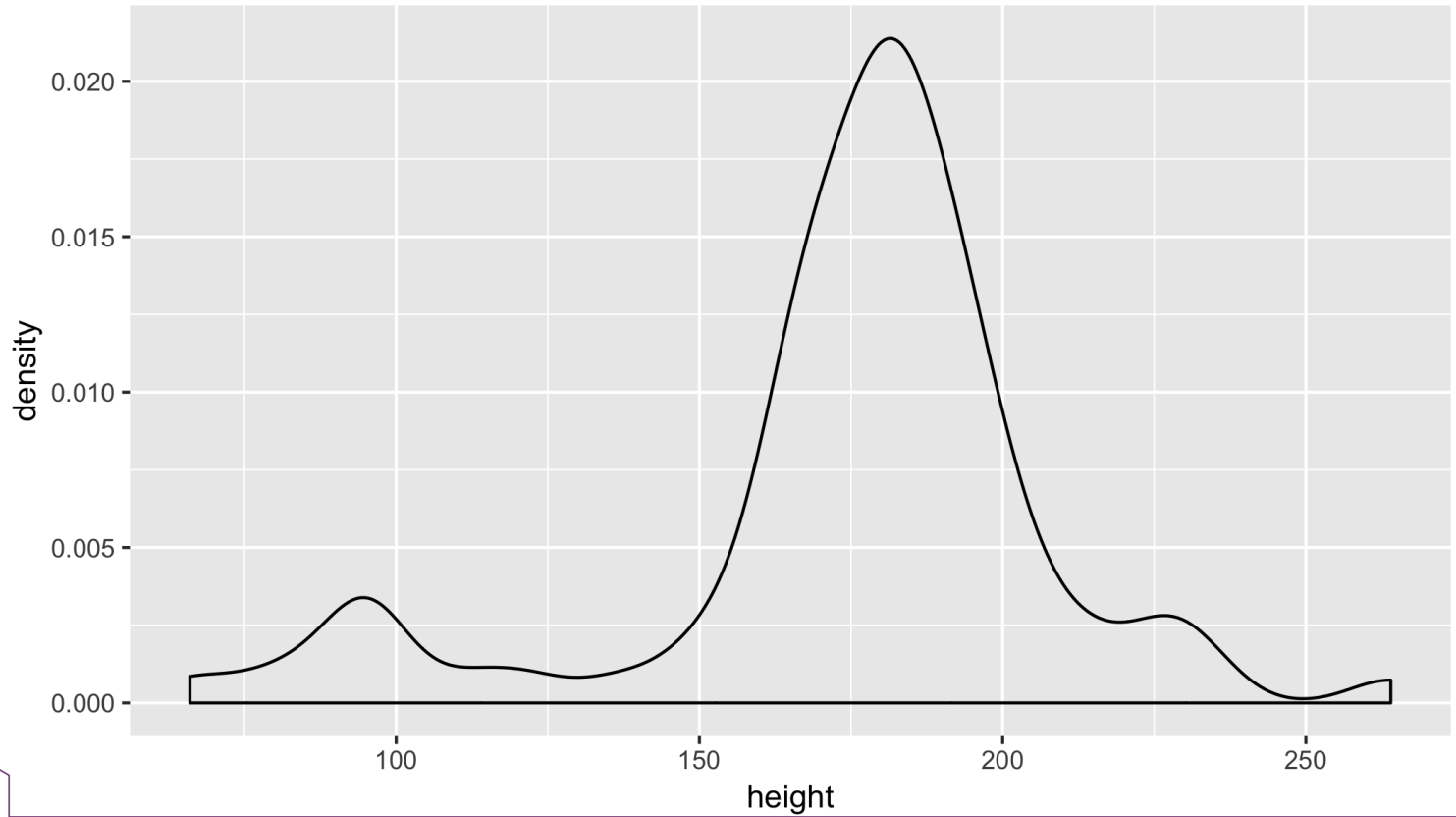
```
## Warning: Removed 6 rows containing non-finite values (stat_bin).
```



Density plots

```
ggplot(data = starwars, mapping = aes(x = height)) +
  geom_density()
```

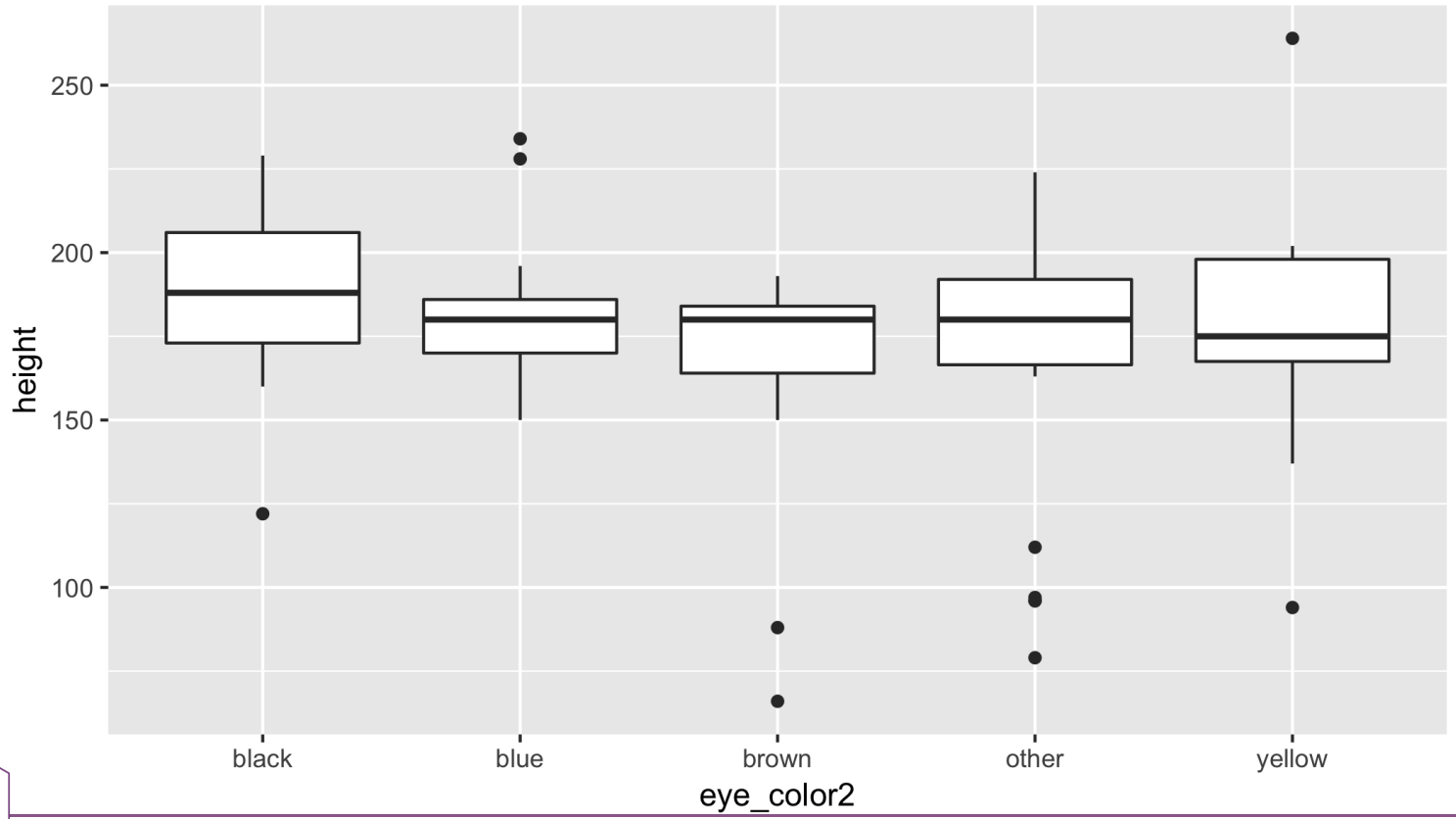
```
## Warning: Removed 6 rows containing non-finite values (stat_density).
```



Side-by-side box plots

```
ggplot(data = starwars, mapping = aes(y = height, x = eye_color2)) +
  geom_boxplot()
```

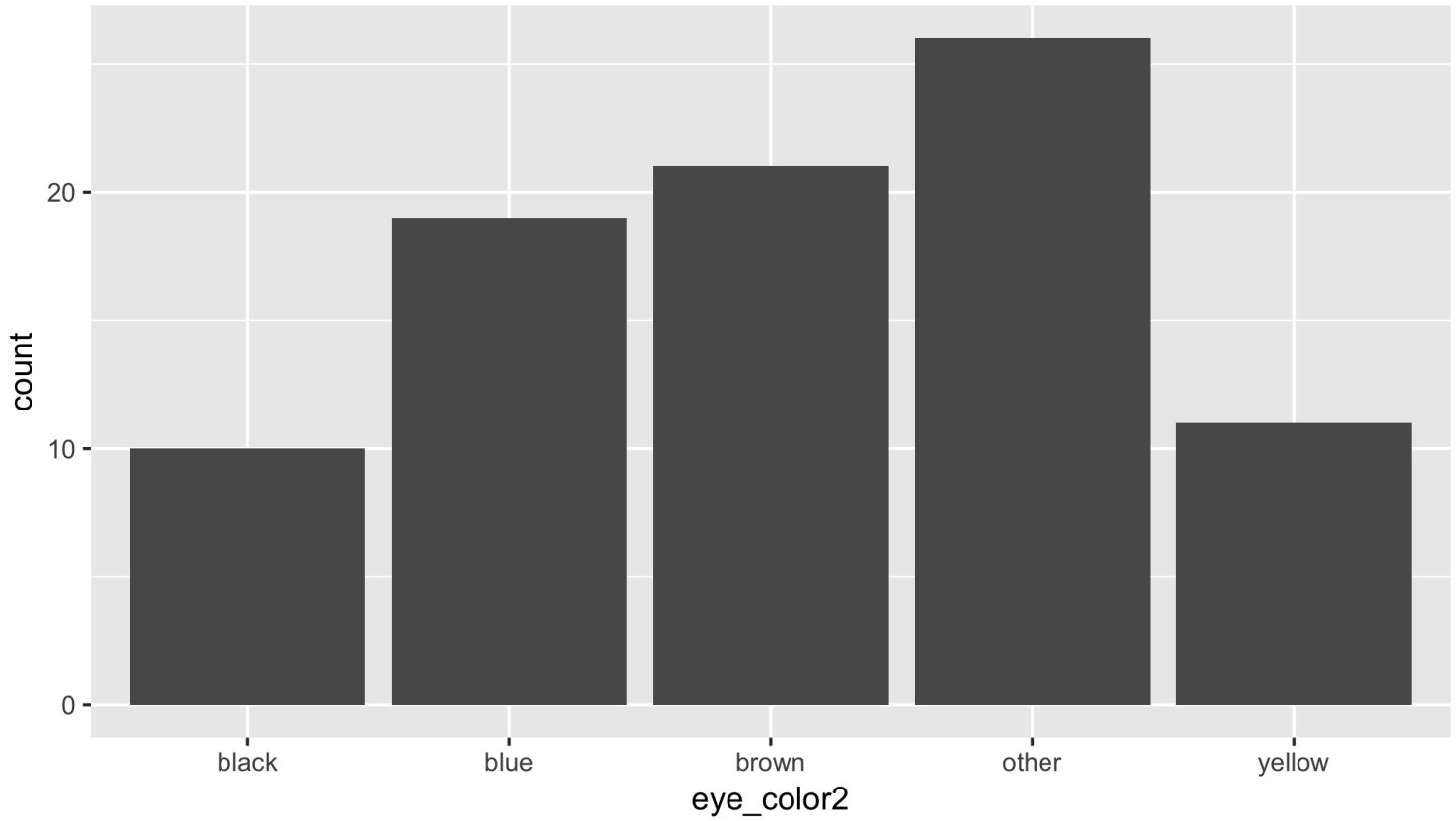
```
## Warning: Removed 6 rows containing non-finite values (stat_boxplot).
```



Visualizing categorical data

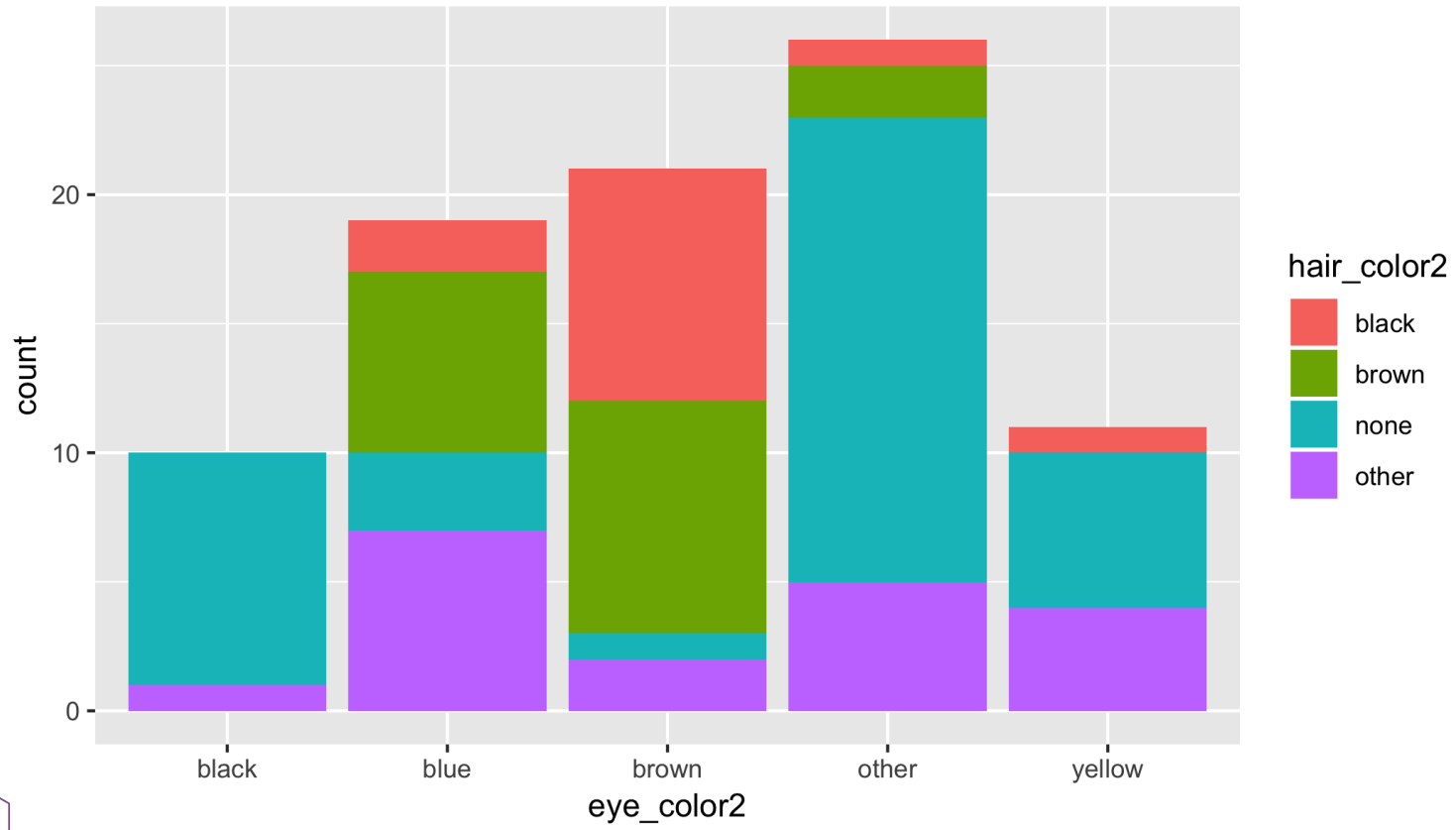
Bar plots

```
ggplot(data = starwars, mapping = aes(x = eye_color2)) +
  geom_bar()
```



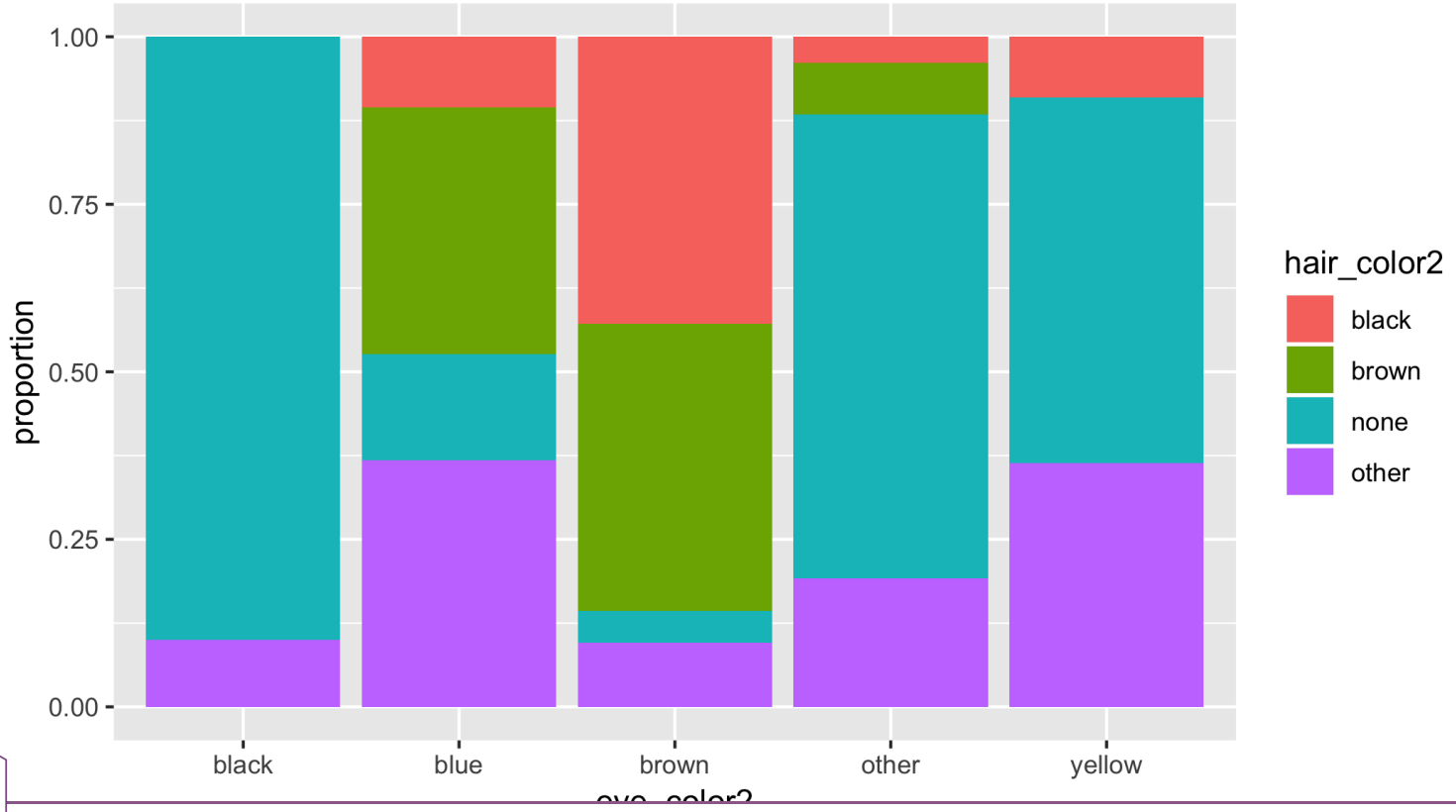
Segmented bar plots, counts

```
ggplot(data = starwars, mapping = aes(x = eye_color2, fill = hair_color2)) +
  geom_bar()
```

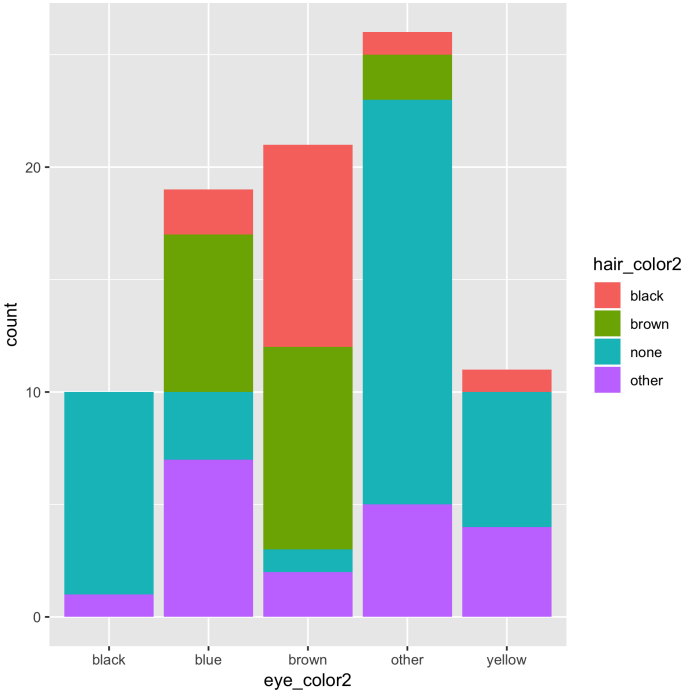


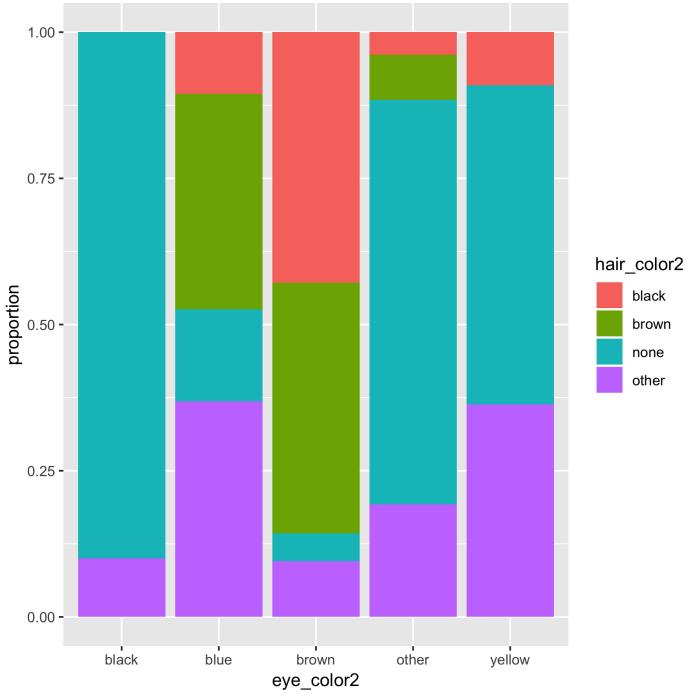
Segmented bar plots, proportions

```
ggplot(data = starwars, mapping = aes(x = eye_color2, fill = hair_color2)) +
  geom_bar(position = "fill") +
  labs(y = "proportion")
```



Which bar plot is a more useful representation for visualizing the relationship between eye color and hair color? Why?





Tidy data

Tidy data

> Happy families are all alike; every unhappy family is unhappy in its own way.
>
> Leo Tolstoy

Characteristics of tidy data:

- Each variable forms a column.
- Each observation forms a row.
- Each type of observational unit forms a table.

Characteristics of untidy data:

!@#$%^&*()

Summary tables

Is each of the following a dataset? Which is a summary table?

```
## # A tibble: 87 x 3
##    name               height  mass
##    <chr>               <int> <dbl>
##  1 Luke Skywalker        172    77
##  2 C-3PO                 167    75
##  3 R2-D2                  96    32
##  4 Darth Vader           202   136
##  5 Leia Organa           150    49
##  6 Owen Lars             178   120
##  7 Beru Whitesun lars    165    75
##  8 R5-D4                  97    32
##  9 Biggs Darklighter     183    84
## 10 Obi-Wan Kenobi        182    77
## # … with 77 more rows
```

```
## # A tibble: 10 x 2
##    species   avg_height
##    <chr>          <dbl>
##  1 <NA>             160
##  2 Aleena            79
##  3 Besalisk         198
##  4 Cerean           198
##  5 Chagrian         196
##  6 Clawdite         168
##  7 Droid            140
##  8 Dug              112
##  9 Ewok              88
## 10 Geonosian        183
```

Pipes

Where does the name come from?

The pipe operator is implemented in the package **magrittr**, it's pronounced "and then".

https://en.wikipedia.org/wiki/The_Treachery_of_Images

Review: How does a pipe work?

- You can think about the following sequence of actions - find key, unlock car, start car, drive to school, park.
- Expressed as a set of nested functions in R pseudocode this would look like:

```
park(drive(start_car(find("keys")), to = "campus"))
```

- Writing it out using pipes give it a more natural (and easier to read) structure:

```
find("keys") %>%
  start_car() %>%
  drive(to = "campus") %>%
  park()
```

What about other arguments?

To send results to a function argument other than first one or to use the previous result for multiple arguments, use "`.`"

```
starwars %>%
  filter(species == "Human") %>%
  lm(mass ~ height, data = .)
```

```
##
## Call:
## lm(formula = mass ~ height, data = .)
##
## Coefficients:
## (Intercept)       height
##     -116.58         1.11
```

Data wrangling

Bike crashes in NC 2007 - 2014

The dataset is in the **dsbox** package:

```
library(dsbox)
ncbikecrash
```

- The dataset contains all North Carolina bike crash data from 2007-2014.
- Data downloaded on Sep 6, 2018.

Variables

View the names of variables via

```
names(ncbikecrash)
```

```
##  [1] "object_id"           "city"                "county"
##  [4] "region"              "development"         "locality"
##  [7] "on_road"             "rural_urban"         "speed_limit"
## [10] "traffic_control"     "weather"             "workzone"
## [13] "bike_age"            "bike_age_group"      "bike_alcohol"
## [16] "bike_alcohol_drugs"  "bike_direction"      "bike_injury"
## [19] "bike_position"       "bike_race"           "bike_sex"
## [22] "driver_age"          "driver_age_group"    "driver_alcohol"
## [25] "driver_alcohol_drugs" "driver_est_speed"   "driver_injury"
## [28] "driver_race"         "driver_sex"          "driver_vehicle_type"
## [31] "crash_alcohol"       "crash_date"          "crash_day"
## [34] "crash_group"         "crash_hour"          "crash_location"
## [37] "crash_month"         "crash_severity"      "crash_time"
## [40] "crash_type"          "crash_year"          "ambulance_req"
## [43] "hit_run"             "light_condition"     "road_character"
## [46] "road_class"          "road_condition"      "road_configuration"
## [49] "road_defects"        "road_feature"        "road_surface"
## [52] "num_bikes_ai"        "num_bikes_bi"        "num_bikes_ci"
## [55] "num_bikes_ki"        "num_bikes_no"        "num_bikes_to"
## [58] "num_bikes_ui"        "num_lanes"           "num_units"
## [61] "distance_mi_from"    "frm_road"            "rte_invd_cd"
## [64] "towrd_road"          "geo_point"           "geo_shape"
```

Viewing your data

- In the Environment, after loading with **`data(ncbikecrash)`**, and click on the name of the data frame to view it in the data viewer
- Use the **`glimpse`** function to take a peek

```
glimpse(ncbikecrash)
```

```
## Observations: 7,467
## Variables: 66
## $ object_id            <int> 1686, 1674, 1673, 1687, 1653, 1665, 1642, 1…
## $ city                 <chr> "None - Rural Crash", "Henderson", "None - …
## $ county               <chr> "Wayne", "Vance", "Lincoln", "Columbus", "N…
## $ region               <chr> "Coastal", "Piedmont", "Piedmont", "Coastal…
## $ development          <chr> "Farms, Woods, Pastures", "Residential", "F…
## $ locality             <chr> "Rural (<30% Developed)", "Mixed (30% To 70…
## $ on_road              <chr> "SR 1915", "NICHOLAS ST", "US 321", "W BURK…
## $ rural_urban          <chr> "Rural", "Urban", "Rural", "Urban", "Urban"…
## $ speed_limit          <chr> "50 - 55  MPH", "30 - 35  MPH", "50 - 55  M…
## $ traffic_control      <chr> "No Control Present", "Stop Sign", "Double …
## $ weather              <chr> "Clear", "Clear", "Clear", "Rain", "Clear",…
## $ workzone             <chr> "No", "No", "No", "No", "No", "No", "No", "…
## $ bike_age             <chr> "52", "66", "33", "52", "22", "15", "41", "…
## $ bike_age_group       <chr> "50-59", "60-69", "30-39", "50-59", "20-24"…
## $ bike_alcohol         <chr> "No", "No", "No", "Yes", "No", "No", "No", …
## $ bike_alcohol_drugs   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…
```

A Grammar of Data Manipulation

dplyr is based on the concepts of functions as verbs that manipulate data frames.

A Grammar of Data Manipulation

dplyr is based on the concepts of functions as verbs that manipulate data frames.

dplyr rules for functions

- First argument is *always* a data frame
- Subsequent arguments say what to do with that data frame
- Always returns a data frame

A note on piping and layering

- The **%>%** operator in **dplyr** functions is called the **pipe operator**. This means you "pipe" the output of the previous line of code as the first input of the next line of code.

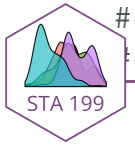
- The **+** operator in **ggplot2** functions is used for "layering". This means you create the plot in layers, separated by **+**.

`filter` to select a subset of rows

for crashes in Durham County

```
ncbikecrash %>%
  filter(county == "Durham")
```

```
## # A tibble: 340 x 66
##    object_id city  county region development locality on_road rural_urban
##        <int> <chr> <chr>  <chr>  <chr>       <chr>    <chr>   <chr>
##  1      2452 Durh… Durham Piedm… Residential Urban (… <NA>    Urban
##  2      2441 Durh… Durham Piedm… Commercial  Urban (… <NA>    Urban
##  3      2466 Durh… Durham Piedm… Commercial  Urban (… <NA>    Urban
##  4       549 Durh… Durham Piedm… Residential Urban (… PARK A… Urban
##  5       598 Durh… Durham Piedm… Residential Urban (… BELT S… Urban
##  6       603 Durh… Durham Piedm… Residential Urban (… HINSON… Urban
##  7      3974 Durh… Durham Piedm… Commercial  Urban (… <NA>    Urban
##  8      7134 Durh… Durham Piedm… Commercial  Urban (… <NA>    Urban
##  9      1670 Durh… Durham Piedm… Commercial  Urban (… INFINI… Urban
## 10      1773 Durh… Durham Piedm… Residential Urban (… <NA>    Urban
## # … with 330 more rows, and 58 more variables: speed_limit <chr>,
## #   traffic_control <chr>, weather <chr>, workzone <chr>, bike_age <chr>,
## #   bike_age_group <chr>, bike_alcohol <chr>, bike_alcohol_drugs <chr>,
## #   bike_direction <chr>, bike_injury <chr>, bike_position <chr>,
## #   bike_race <chr>, bike_sex <chr>, driver_age <chr>,
## #   driver_age_group <chr>, driver_alcohol <chr>,
```

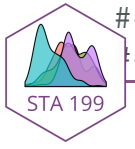


`filter` for many conditions at once

for crashes in Durham County where biker was 30-39 years old

```
ncbikecrash %>%
  filter(county == "Durham", bike_age_group == "30-39")
```

```
## # A tibble: 47 x 66
##    object_id city  county region development locality on_road rural_urban
##        <int> <chr> <chr>  <chr>  <chr>       <chr>    <chr>   <chr>
##  1      1244 Durh… Durham Piedm… Residential Urban (… HILLAN… Urban
##  2      3227 Durh… Durham Piedm… Residential Urban (… <NA>    Urban
##  3      3449 Durh… Durham Piedm… Commercial  Urban (… <NA>    Urban
##  4      1138 Durh… Durham Piedm… Commercial  Urban (… CORNWA… Urban
##  5      1424 Durh… Durham Piedm… Commercial  Urban (… PVA 40… Urban
##  6       198 Durh… Durham Piedm… Commercial  Urban (… EAST C… Urban
##  7      4202 None… Durham Piedm… Farms, Woo… Rural (… <NA>    Rural
##  8      6464 Durh… Durham Piedm… Residential Urban (… <NA>    Urban
##  9      1869 Durh… Durham Piedm… Commercial  Urban (… <NA>    Urban
## 10      1044 Durh… Durham Piedm… Residential Urban (… W CLUB… Urban
## # … with 37 more rows, and 58 more variables: speed_limit <chr>,
## #   traffic_control <chr>, weather <chr>, workzone <chr>, bike_age <chr>,
## #   bike_age_group <chr>, bike_alcohol <chr>, bike_alcohol_drugs <chr>,
## #   bike_direction <chr>, bike_injury <chr>, bike_position <chr>,
## #   bike_race <chr>, bike_sex <chr>, driver_age <chr>,
## #   driver_age_group <chr>, driver_alcohol <chr>,
```



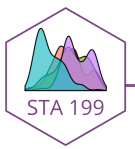
Logical operators in R

operator	definition	operator	definition
`<`	less than	`x | y`	`x` OR `y`
`<=`	less than or equal to	`is.na(x)`	test if `x` is `NA`
`>`	greater than	`!is.na(x)`	test if `x` is not `NA`
`>=`	greater than or equal to	`x %in% y`	test if `x` is in `y`
`==`	exactly equal to	`!(x %in% y)`	test if `x` is not in `y`
`!=`	not equal to	`!x`	not `x`
`x & y`	`x` AND `y`		

`select` to keep variables

```
ncbikecrash %>%
  filter(county == "Durham", bike_age_group == "30-39") %>%
  select(locality, speed_limit)
```

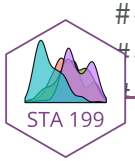
```
## # A tibble: 47 x 2
##    locality                speed_limit
##    <chr>                   <chr>
##  1 Urban (>70% Developed) 30 - 35  MPH
##  2 Urban (>70% Developed) 40 - 45  MPH
##  3 Urban (>70% Developed) 30 - 35  MPH
##  4 Urban (>70% Developed) 30 - 35  MPH
##  5 Urban (>70% Developed) 5 - 15 MPH
##  6 Urban (>70% Developed) 20 - 25  MPH
##  7 Rural (<30% Developed) 40 - 45  MPH
##  8 Urban (>70% Developed) 30 - 35  MPH
##  9 Urban (>70% Developed) 30 - 35  MPH
## 10 Urban (>70% Developed) 30 - 35  MPH
## # … with 37 more rows
```



`select` to exclude variables

```
ncbikecrash %>%
  select(-object_id)
```

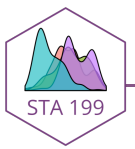
```
## # A tibble: 7,467 x 65
##    city  county region development locality on_road rural_urban speed_limit
##    <chr> <chr>  <chr>  <chr>       <chr>    <chr>   <chr>       <chr>
##  1 None… Wayne  Coast… Farms, Woo… Rural (… SR 1915 Rural       50 - 55  M…
##  2 Hend… Vance  Piedm… Residential Mixed (… NICHOL… Urban       30 - 35  M…
##  3 None… Linco… Piedm… Farms, Woo… Rural (… US 321  Rural       50 - 55  M…
##  4 Whit… Colum… Coast… Commercial  Urban (… W BURK… Urban       30 - 35  M…
##  5 Wilm… New H… Coast… Residential Urban (… RACINE… Urban       <NA>
##  6 None… Robes… Coast… Farms, Woo… Rural (… SR 1513 Rural       50 - 55  M…
##  7 None… Richm… Piedm… Residential Mixed (… SR 1903 Rural       30 - 35  M…
##  8 Rale… Wake   Piedm… Commercial  Urban (… PERSON… Urban       30 - 35  M…
##  9 Whit… Colum… Coast… Residential Rural (… FLOWER… Urban       30 - 35  M…
## 10 New … Craven Coast… Residential Urban (… SUTTON… Urban       20 - 25  M…
## # … with 7,457 more rows, and 57 more variables: traffic_control <chr>,
## #   weather <chr>, workzone <chr>, bike_age <chr>, bike_age_group <chr>,
## #   bike_alcohol <chr>, bike_alcohol_drugs <chr>, bike_direction <chr>,
## #   bike_injury <chr>, bike_position <chr>, bike_race <chr>,
## #   bike_sex <chr>, driver_age <chr>, driver_age_group <chr>,
## #   driver_alcohol <chr>, driver_alcohol_drugs <chr>,
## #   driver_est_speed <chr>, driver_injury <chr>, driver_race <chr>,
## #   driver_sex <chr>, driver_vehicle_type <chr>, crash_alcohol <chr>,
## #   crash_date <chr>, crash_day <chr>, crash_group <chr>
```



`select` a range of variables

```
ncbikecrash %>%
  select(city:locality)
```

```
## # A tibble: 7,467 x 5
##    city            county      region  development        locality
##    <chr>           <chr>       <chr>   <chr>              <chr>
##  1 None - Rural … Wayne       Coastal Farms, Woods, Pa… Rural (<30% Develop…
##  2 Henderson      Vance       Piedmo… Residential        Mixed (30% To 70% D…
##  3 None - Rural … Lincoln     Piedmo… Farms, Woods, Pa… Rural (<30% Develop…
##  4 Whiteville     Columbus    Coastal Commercial         Urban (>70% Develop…
##  5 Wilmington     New Hanov… Coastal Residential        Urban (>70% Develop…
##  6 None - Rural … Robeson     Coastal Farms, Woods, Pa… Rural (<30% Develop…
##  7 None - Rural … Richmond    Piedmo… Residential        Mixed (30% To 70% D…
##  8 Raleigh        Wake        Piedmo… Commercial         Urban (>70% Develop…
##  9 Whiteville     Columbus    Coastal Residential        Rural (<30% Develop…
## 10 New Bern       Craven      Coastal Residential        Urban (>70% Develop…
## # … with 7,457 more rows
```

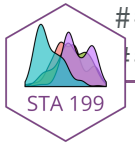


`slice` for certain row numbers

First five rows

```
ncbikecrash %>%
  slice(1:5)
```

```
## # A tibble: 5 x 66
##   object_id city  county region development locality on_road rural_urban
##       <int> <chr> <chr>  <chr>  <chr>       <chr>    <chr>   <chr>
## 1      1686 None… Wayne  Coast… Farms, Woo… Rural (… SR 1915 Rural
## 2      1674 Hend… Vance  Piedm… Residential Mixed (… NICHOL… Urban
## 3      1673 None… Linco… Piedm… Farms, Woo… Rural (… US 321  Rural
## 4      1687 Whit… Colum… Coast… Commercial  Urban (… W BURK… Urban
## 5      1653 Wilm… New H… Coast… Residential Urban (… RACINE… Urban
## # … with 58 more variables: speed_limit <chr>, traffic_control <chr>,
## #   weather <chr>, workzone <chr>, bike_age <chr>, bike_age_group <chr>,
## #   bike_alcohol <chr>, bike_alcohol_drugs <chr>, bike_direction <chr>,
## #   bike_injury <chr>, bike_position <chr>, bike_race <chr>,
## #   bike_sex <chr>, driver_age <chr>, driver_age_group <chr>,
## #   driver_alcohol <chr>, driver_alcohol_drugs <chr>,
## #   driver_est_speed <chr>, driver_injury <chr>, driver_race <chr>,
## #   driver_sex <chr>, driver_vehicle_type <chr>, crash_alcohol <chr>,
## #   crash_date <chr>, crash_day <chr>, crash_group <chr>,
## #   crash_hour <int>, crash_location <chr>, crash_month <chr>,
## #   crash_severity <chr>, crash_time <drtn>, crash_type <chr>,
```

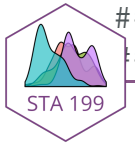


`slice` for certain row numbers

Last five rows

```
last_row <- nrow(ncbikecrash)
ncbikecrash %>%
  slice((last_row - 4):last_row)
```

```
## # A tibble: 5 x 66
##   object_id city  county region development locality on_road rural_urban
##       <int> <chr> <chr>  <chr>  <chr>       <chr>    <chr>   <chr>
## 1      6989 High… Guilf… Piedm… Residential Urban (… <NA>    Urban
## 2      6991 Wilm… New H… Coast… Residential Urban (… <NA>    Urban
## 3      6995 Kins… Lenoir Coast… Commercial  Urban (… <NA>    Urban
## 4      6998 Faye… Cumbe… Coast… Residential Urban (… <NA>    Urban
## 5      7000 None… Onslow Coast… Farms, Woo… Rural (… <NA>    Rural
## # … with 58 more variables: speed_limit <chr>, traffic_control <chr>,
## #   weather <chr>, workzone <chr>, bike_age <chr>, bike_age_group <chr>,
## #   bike_alcohol <chr>, bike_alcohol_drugs <chr>, bike_direction <chr>,
## #   bike_injury <chr>, bike_position <chr>, bike_race <chr>,
## #   bike_sex <chr>, driver_age <chr>, driver_age_group <chr>,
## #   driver_alcohol <chr>, driver_alcohol_drugs <chr>,
## #   driver_est_speed <chr>, driver_injury <chr>, driver_race <chr>,
## #   driver_sex <chr>, driver_vehicle_type <chr>, crash_alcohol <chr>,
## #   crash_date <chr>, crash_day <chr>, crash_group <chr>,
## #   crash_hour <int>, crash_location <chr>, crash_month <chr>,
```



`pull` to extract a column as a vector

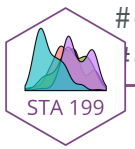
```
ncbikecrash %>%
  slice(1:6) %>%
  pull(locality)
```

```
## [1] "Rural (<30% Developed)"       "Mixed (30% To 70% Developed)"
## [3] "Rural (<30% Developed)"       "Urban (>70% Developed)"
## [5] "Urban (>70% Developed)"       "Rural (<30% Developed)"
```

vs.

```
ncbikecrash %>%
  slice(1:6) %>%
  select(locality)
```

```
## # A tibble: 6 x 1
##   locality
##   <chr>
## 1 Rural (<30% Developed)
## 2 Mixed (30% To 70% Developed)
## 3 Rural (<30% Developed)
## 4 Urban (>70% Developed)
## 5 Urban (>70% Developed)
## 6 Rural (<30% Developed)
```



`sample_n` / `sample_frac` for a random sample

- **`sample_n`**: randomly sample 5 observations

```
ncbikecrash_n5 <- ncbikecrash %>%
  sample_n(5, replace = FALSE)
dim(ncbikecrash_n5)
```

```
## [1]  5 66
```

- **`sample_frac`**: randomly sample 20% of observations

```
ncbikecrash_perc20 <-ncbikecrash %>%
  sample_frac(0.2, replace = FALSE)
dim(ncbikecrash_perc20)
```

```
## [1] 1493   66
```

`distinct` to filter for unique rows

And **`arrange`** to order alphabetically

```
ncbikecrash %>%
  select(county, city) %>%
  distinct() %>%
  arrange(county, city)
```

```
## # A tibble: 391 x 2
##    county    city
##    <chr>     <chr>
##  1 Alamance  Alamance
##  2 Alamance  Burlington
##  3 Alamance  Elon
##  4 Alamance  Elon College
##  5 Alamance  Gibsonville
##  6 Alamance  Graham
##  7 Alamance  Green Level
##  8 Alamance  Mebane
##  9 Alamance  None - Rural Crash
## 10 Alexander None - Rural Crash
## # … with 381 more rows
```

`summarise` to reduce variables to values

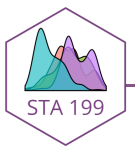
```
ncbikecrash %>%
  summarise(avg_hr = mean(crash_hour))
```

```
## # A tibble: 1 x 1
##   avg_hr
##    <dbl>
## 1   14.7
```

`group_by` to do calculations on groups

```
ncbikecrash %>%
  group_by(hit_run) %>%
  summarise(avg_hr = mean(crash_hour))
```

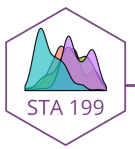
```
## # A tibble: 2 x 2
##   hit_run avg_hr
##   <chr>    <dbl>
## 1 No        14.6
## 2 Yes       15.0
```



`count` observations in groups

```
ncbikecrash %>%
  count(driver_alcohol_drugs)
```

```
## # A tibble: 6 x 2
##   driver_alcohol_drugs                      n
##   <chr>                                 <int>
## 1 <NA>                                   6654
## 2 Missing                                  99
## 3 No                                      695
## 4 Yes-Alcohol,  impairment suspected       12
## 5 Yes-Alcohol, no impairment detected       3
## 6 Yes-Drugs, impairment suspected           4
```



`mutate` to add new variables

```
ncbikecrash %>%
  mutate(driver_alcohol_drugs_simplified = case_when(
    is.na(driver_alcohol_drugs)              ~ driver_alcohol_drugs,
    driver_alcohol_drugs == "Missing"        ~ NA_character_,
    str_detect(driver_alcohol_drugs, "Yes") ~ "Yes",
    TRUE                                     ~ "No"
  ))
```

"Save" when you `mutate`

Most often when you define a new variable with **`mutate`** you'll also want to save the resulting data frame, often by writing over the original data frame.

```
ncbikecrash <- ncbikecrash %>%
  mutate(driver_alcohol_drugs_simplified = case_when(
    is.na(driver_alcohol_drugs)              ~ driver_alcohol_drugs,
    driver_alcohol_drugs == "Missing"        ~ NA_character_,
    str_detect(driver_alcohol_drugs, "Yes") ~ "Yes",
    TRUE                                     ~ "No"
  ))
```

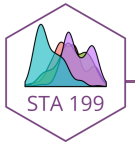
Check before you move on

```
ncbikecrash %>%
  count(driver_alcohol_drugs, driver_alcohol_drugs_simplified)
```

```
## # A tibble: 6 x 3
##   driver_alcohol_drugs                 driver_alcohol_drugs_simplified     n
##   <chr>                                <chr>                           <int>
## 1 <NA>                                 <NA>                             6654
## 2 Missing                              <NA>                               99
## 3 No                                   No                                695
## 4 Yes-Alcohol,  impairment suspected   Yes                                12
## 5 Yes-Alcohol, no impairment detected  Yes                                 3
## 6 Yes-Drugs, impairment suspected      Yes                                 4
```

```
ncbikecrash %>%
  count(driver_alcohol_drugs_simplified)
```

```
## # A tibble: 3 x 2
##   driver_alcohol_drugs_simplified     n
##   <chr>                           <int>
## 1 <NA>                             6753
## 2 No                                695
## 3 Yes                                19
```



AE 04 - NC bike crashes

- Copy the NC Bike Crashes project on RStudio Cloud
- For each question you work on, set the **`eval`** chunk option to **`TRUE`** and knit

Before next class

- You will get your teams in lab tomorrow!