Exploratory Data Analysis (EDA) Concepts

Justin Post

Recap!

- Data Science!!
- R Projects/Quarto/Git/GitHub for reproducibility/communication
- R Data Structures
    - Vectors, Matrices, Data Frames, Lists
- R Control Flow
    - if/then/else, loops, function writing
- Reading & Manipulating data with the tidyverse!
- Next: Gain meaningful insights from data through EDA
- Later: Dashboards, Predictive Modeling, & More

EDA Basics

- Get to know your data!
- EDA generally consists of a few steps:
  - Understand how your data is stored
  - Do basic data validation
  - Determine rate of missing values
  - Clean data up data as needed
  - Investigate distributions
    - Univariate measures/graphs
    - Multivariate measures/graphs
  - Apply transformations and repeat previous step

Understand How Data is Stored

Let's read in some data!

- Appendicitis Data

> This dataset was acquired in a retrospective study from a cohort of pediatric patients admitted with abdominal pain to Children's Hospital St. Hedwig in Regensburg, Germany. ... Alongside multiple US images for each subject, the dataset includes information encompassing laboratory tests, physical examination results, clinical scores, such as Alvarado and pediatric appendicitis scores, and expert-produced ultrasonographic findings. Lastly, the subjects were labeled w.r.t. three target variables: diagnosis (appendicitis vs. no appendicitis), management (surgical vs. conservative) and severity (complicated vs. uncomplicated or no appendicitis). ...

Understand How Data is Stored

```r
#download data to local folder
library(tidyverse)
library(readxl)
app_data <- read_excel("data/app_data.xlsx", sheet = 1)
```

- Column data types should make sense for what you expect!

```r
app_data
```

```
## # A tibble: 782 x 58
##      Age BMI                Sex   Height Weight Length_of_Stay Management Severity
##    <dbl> <chr>              <chr>  <dbl>  <dbl>          <dbl> <chr>      <chr>
## 1   12.7 16.8999999999999~ fema~    148   37                 3 conservat~ uncompl~
## 2   14.1 31.9               male     147   69.5               2 conservat~ uncompl~
## 3   14.1 23.3               fema~    163   62                 4 conservat~ uncompl~
## 4   16.4 20.6               fema~    165   56                 3 conservat~ uncompl~
## 5   11.1 16.8999999999999~ fema~    163   45                 3 conservat~ uncompl~
## # i 777 more rows
## # i 50 more variables: Diagnosis_Presumptive <chr>, Diagnosis <chr>,
## #   Alvarado_Score <dbl>, Paedriatic_Appendicitis_Score <dbl>,
## #   Appendix_on_US <chr>, Appendix_Diameter <dbl>, Migratory_Pain <chr>,
## #   Lower_Right_Abd_Pain <chr>, Contralateral_Rebound_Tenderness <chr>,
a <chr>, Loss_of_Appetite <chr>,
C_Count <dbl>, Neutrophil_Percentage <dbl>, ...
```

Understand How Data is Stored

- Check the structure of the data!

```
str(app_data)
```

```
## tibble [782 x 58] (S3: tbl_df/tbl/data.frame)
##  $ Age                              : num [1:782] 12.7 14.1 14.1 16.4 11.1 ...
##  $ BMI                              : chr [1:782] "16.899999999999999" "31.9" "23.3" "20.6" ...
##  $ Sex                              : chr [1:782] "female" "male" "female" "female" ...
##  $ Height                           : num [1:782] 148 147 163 165 163 121 140 NA 131 174 ...
##  $ Weight                           : num [1:782] 37 69.5 62 56 45 45 38.5 21.5 26.7 45.5 ...
##  $ Length_of_Stay                   : num [1:782] 3 2 4 3 3 3 3 2 3 3 ...
##  $ Management                       : chr [1:782] "conservative" "conservative" "conservative" "conservative" ...
##  $ Severity                         : chr [1:782] "uncomplicated" "uncomplicated" "uncomplicated" "uncomplicated" ...
##  $ Diagnosis_Presumptive            : chr [1:782] "appendicitis" "appendicitis" "appendicitis" "appendicitis" ...
##  $ Diagnosis                        : chr [1:782] "appendicitis" "no appendicitis" "no appendicitis" "no appendicitis" ..
##  $ Alvarado_Score                   : num [1:782] 4 5 5 7 5 6 5 3 7 4 ...
##  $ Paedriatic_Appendicitis_Score    : num [1:782] 3 4 3 6 6 7 6 3 6 4 ...
##  $ Appendix_on_US                   : chr [1:782] "yes" "no" "no" "no" ...
##  $ Appendix_Diameter                : num [1:782] 7.1 NA NA NA 7 NA NA NA 3.7 8 ...
##  $ Migratory_Pain                   : chr [1:782] "no" "yes" "no" "yes" ...
##  $ Lower_Right_Abd_Pain             : chr [1:782] "yes" "yes" "yes" "yes" ...
##  $ Contralateral_Rebound_Tenderness: chr [1:782] "yes" "yes" "yes" "no" ...
##  $ Coughing_Pain                    : chr [1:782] "no" "no" "no" "no" ...
                                    : chr [1:782] "no" "no" "no" "yes" ...
                                    : chr [1:782] "yes" "yes" "no" "yes" ...
                                    : num [1:782] 37 36.9 36.6 36 36.9 36.9 36.7 36.8 37.3 37.1 ...
##  $ WBC_Count                        : num [1:782] 7.7 8.1 13.2 11.4 8.1 9.5 10 8 20.9 5.8 ...
```

Convert Columns Explicitly

- `as.*()` family of functions can help coerce columns to the correct type

```
app_data <- app_data |>
  mutate(BMI = as.numeric(BMI),
         US_Number = as.character(US_Number))
app_data
```

```
## # A tibble: 782 x 58
##     Age   BMI Sex    Height Weight Length_of_Stay Management   Severity
##   <dbl> <dbl> <chr>   <dbl>  <dbl>          <dbl> <chr>        <chr>
## 1  12.7  16.9 female    148   37                3 conservative uncomplicated
## 2  14.1  31.9 male      147   69.5              2 conservative uncomplicated
## 3  14.1  23.3 female    163   62                4 conservative uncomplicated
## 4  16.4  20.6 female    165   56                3 conservative uncomplicated
## 5  11.1  16.9 female    163   45                3 conservative uncomplicated
## # i 777 more rows
## # i 50 more variables: Diagnosis_Presumptive <chr>, Diagnosis <chr>,
## #   Alvarado_Score <dbl>, Paedriatic_Appendicitis_Score <dbl>,
## #   Appendix_on_US <chr>, Appendix_Diameter <dbl>, Migratory_Pain <chr>,
## #   Lower_Right_Abd_Pain <chr>, Contralateral_Rebound_Tenderness <chr>,
## #   Coughing_Pain <chr>, Nausea <chr>, Loss_of_Appetite <chr>,
## #   Body_Temperature <dbl>, WBC_Count <dbl>, Neutrophil_Percentage <dbl>, ...
```

Do Basic Data Validation

- Can use the `psych::describe()` function
- Check that the min's, max's, etc. all make sense!

```
psych::describe(app_data)
```

```
##                                  vars   n   mean    sd median trimmed   mad
## Age                                 1 781  11.35  3.53  11.44   11.53  3.59
## BMI                                 2 755  18.91  4.39  18.06   18.43  3.91
## Sex*                                3 780   1.52  0.50   2.00    1.52  0.00
## Height                              4 756 148.02 19.73 149.65  149.33 19.50
## Weight                              5 779  43.17 17.39  41.40   42.18 18.68
## Length_of_Stay                      6 778   4.28  2.57   3.00    3.85  1.48
## Management*                         7 781   1.42  0.57   1.00    1.35  0.00
## Severity*                           8 781   1.85  0.36   2.00    1.93  0.00
## Diagnosis_Presumptive*              9 780   4.04  2.86   3.00    3.17  0.00
## Diagnosis*                         10 780   1.41  0.49   1.00    1.38  0.00
## Alvarado_Score                     11 730   5.92  2.16   6.00    5.96  2.97
## Paedriatic_Appendicitis_Score      12 730   5.25  1.96   5.00    5.21  1.48
## Appendix_on_US*                    13 777   1.65  0.48   2.00    1.69  0.00
## Appendix_Diameter                  14 498   7.76  2.54   7.50    7.63  2.22
## Migratory_Pain*                    15 773   1.27  0.45   1.00    1.22  0.00
## Lower_Right_Abd_Pain*              16 774   1.95  0.22   2.00    2.00  0.00
## Contralateral_Rebound_Tenderness*  17 767   1.39  0.49   1.00    1.36  0.00
##                                    18 766   1.28  0.45   1.00    1.23  0.00
##                                    19 774   1.59  0.49   2.00    1.61  0.00
##                                    20 772   1.51  0.50   2.00    1.51  0.00
## Body_Temperature                   21 775  37.40  0.90  37.20   37.36  0.74
```

Determine Rate of Missing Values

- Use `is.na()`

```
colSums(is.na(app_data))
```

```
##                             Age                              BMI
##                               1                               27
##                             Sex                           Height
##                               2                               26
##                          Weight                   Length_of_Stay
##                               3                                4
##                      Management                         Severity
##                               1                                1
##            Diagnosis_Presumptive                        Diagnosis
##                               2                                2
##                  Alvarado_Score    Paedriatic_Appendicitis_Score
##                              52                               52
##                  Appendix_on_US                Appendix_Diameter
##                               5                              284
##                  Migratory_Pain              Lower_Right_Abd_Pain
##                               9                                8
## Contralateral_Rebound_Tenderness                    Coughing_Pain
##                              15                               16
##                          Nausea                 Loss_of_Appetite
                                  8                               10
                                 re                        WBC_Count
                                  7                                6
##           Neutrophil_Percentage            Segmented_Neutrophils
```

Determine Rate of Missing Values

- Stay in the `tidyverse`

```
sum_na <- function(column){
  sum(is.na(column))
}
na_counts <- app_data |>
  summarize(across(everything(), sum_na))
na_counts
```

```
## # A tibble: 1 x 58
##     Age   BMI   Sex Height Weight Length_of_Stay Management Severity
##   <int> <int> <int>  <int>  <int>          <int>      <int>    <int>
## 1     1    27     2     26      3              4          1        1
## # i 50 more variables: Diagnosis_Presumptive <int>, Diagnosis <int>,
## #   Alvarado_Score <int>, Paedriatic_Appendicitis_Score <int>,
## #   Appendix_on_US <int>, Appendix_Diameter <int>, Migratory_Pain <int>,
## #   Lower_Right_Abd_Pain <int>, Contralateral_Rebound_Tenderness <int>,
## #   Coughing_Pain <int>, Nausea <int>, Loss_of_Appetite <int>,
## #   Body_Temperature <int>, WBC_Count <int>, Neutrophil_Percentage <int>,
## #   Segmented_Neutrophils <int>, Neutrophilia <int>, RBC_Count <int>, ...
```

Clean Up Data As Needed

- Can remove rows with missing using `tidyr::drop_na()` function

```
names(app_data)[na_counts < 30]
```

```
##  [1] "Age"                          "BMI"
##  [3] "Sex"                          "Height"
##  [5] "Weight"                       "Length_of_Stay"
##  [7] "Management"                   "Severity"
##  [9] "Diagnosis_Presumptive"        "Diagnosis"
## [11] "Appendix_on_US"               "Migratory_Pain"
## [13] "Lower_Right_Abd_Pain"         "Contralateral_Rebound_Tenderness"
## [15] "Coughing_Pain"                "Nausea"
## [17] "Loss_of_Appetite"             "Body_Temperature"
## [19] "WBC_Count"                    "RBC_Count"
## [21] "Hemoglobin"                   "RDW"
## [23] "Thrombocyte_Count"            "CRP"
## [25] "Dysuria"                      "Stool"
## [27] "Peritonitis"                  "US_Performed"
## [29] "US_Number"
```

Clean Up Data As Needed

- Can remove rows with missing using `tidyr::drop_na()` function

```
app_data |>
  drop_na(names(app_data)[na_counts < 30])
```

```
## # A tibble: 674 x 58
##      Age   BMI Sex    Height Weight Length_of_Stay Management   Severity
##    <dbl> <dbl> <chr>   <dbl>  <dbl>          <dbl> <chr>        <chr>
## 1   12.7  16.9 female    148   37                3 conservative uncomplicated
## 2   14.1  31.9 male      147   69.5              2 conservative uncomplicated
## 3   14.1  23.3 female    163   62                4 conservative uncomplicated
## 4   16.4  20.6 female    165   56                3 conservative uncomplicated
## 5   11.1  16.9 female    163   45                3 conservative uncomplicated
## # i 669 more rows
## # i 50 more variables: Diagnosis_Presumptive <chr>, Diagnosis <chr>,
## #   Alvarado_Score <dbl>, Paedriatic_Appendicitis_Score <dbl>,
## #   Appendix_on_US <chr>, Appendix_Diameter <dbl>, Migratory_Pain <chr>,
## #   Lower_Right_Abd_Pain <chr>, Contralateral_Rebound_Tenderness <chr>,
## #   Coughing_Pain <chr>, Nausea <chr>, Loss_of_Appetite <chr>,
## #   Body_Temperature <dbl>, WBC_Count <dbl>, Neutrophil_Percentage <dbl>, ...
```

May Want to Impute Values

- We lose information when removing rows!
- Can **impute** missing values with `tidyr::replace_na()`

```
app_data <- app_data |>
  replace_na(list(BMI = mean(app_data$BMI, na.rm = TRUE),
                  Height = mean(app_data$Height, na.rm = TRUE)))
app_data
```

```
## # A tibble: 782 x 58
##     Age   BMI Sex    Height Weight Length_of_Stay Management   Severity
##   <dbl> <dbl> <chr>   <dbl> <dbl>           <dbl> <chr>        <chr>
## 1  12.7  16.9 female    148  37                 3 conservative uncomplicated
## 2  14.1  31.9 male      147  69.5               2 conservative uncomplicated
## 3  14.1  23.3 female    163  62                 4 conservative uncomplicated
## 4  16.4  20.6 female    165  56                 3 conservative uncomplicated
## 5  11.1  16.9 female    163  45                 3 conservative uncomplicated
## # i 777 more rows
## # i 50 more variables: Diagnosis_Presumptive <chr>, Diagnosis <chr>,
## #   Alvarado_Score <dbl>, Paedriatic_Appendicitis_Score <dbl>,
## #   Appendix_on_US <chr>, Appendix_Diameter <dbl>, Migratory_Pain <chr>,
## #   Lower_Right_Abd_Pain <chr>, Contralateral_Rebound_Tenderness <chr>,
## #   Coughing_Pain <chr>, Nausea <chr>, Loss_of_Appetite <chr>,
C_Count <dbl>, Neutrophil_Percentage <dbl>, ...
```

EDA Basics

- Get to know your data!

- EDA generally consists of a few steps:

    - Understand how your data is stored
    - Do basic data validation
    - Determine rate of missing values
    - Clean data up data as needed
    - Investigate distributions
        - Univariate measures/graphs
        - Multivariate measures/graphs
    - Apply transformations and repeat previous step

Investigate distributions

- How to summarize data depends on the type of data
    - Categorical (Qualitative) variable - entries are a label or attribute
    - Numeric (Quantitative) variable - entries are a numerical value where math can be performed

NC STATE UNIVERSITY

Investigate distributions

- How to summarize data depends on the type of data
    - Categorical (Qualitative) variable - entries are a label or attribute
    - Numeric (Quantitative) variable - entries are a numerical value where math can be performed
- Numerical summaries (across subgroups)
    - Contingency Tables (for categorical data)
    - Mean/Median
    - Standard Deviation/Variance/IQR
    - Quantiles/Percentiles
- Graphical summaries (across subgroups)
    - Bar plots (for categorical data)
    - Histograms
    - Box plots
    - Scatter plots

Categorical Data

Goal: Describe the **distribution** of the variable

- Distribution = pattern and frequency with which you observe a variable
- Categorical variable - entries are a label or attribute
    - Describe the relative frequency (or count) for each category

Variables of interest for this section:

- `Sex`, `Diagnosis`, `Severity`

Factors

A factor variable is really useful for certain categorical variables!

Factor - special class of vector with a `levels` attribute

- Can have more descriptive labels, ordering of categories, etc.
- Levels define **all** possible values for that variable
    - Great for variable like `Day` (Monday, Tuesday, …, Sunday)
    - Not great for variable like `Name` where new values may come up
- Great for plotting as you can order the levels and give nicer labels

Factors

- Let's create factor versions of our three variables

```
unique(app_data$Sex)
```

```
## [1] "female" "male"   NA
```

```
unique(app_data$Diagnosis)
```

```
## [1] "appendicitis"    "no appendicitis" NA
```

```
unique(app_data$Severity)
```

```
## [1] "uncomplicated" NA              "complicated"
```

- Now we can use `factor()` or `as.factor()` to coerce the character variables

Factors

- Let's create factor versions of our three variables

```
app_data |>
  mutate(SexF = factor(Sex, levels = c("female", "male"), labels = c("Female", "Male")),
         DiagnosisF = as.factor(Diagnosis),
         SeverityF = as.factor(Severity)) |>
  select(SexF, DiagnosisF, SeverityF)
```

```
## # A tibble: 782 x 3
##   SexF   DiagnosisF      SeverityF
##   <fct>  <fct>           <fct>
## 1 Female appendicitis    uncomplicated
## 2 Male   no appendicitis uncomplicated
## 3 Female no appendicitis uncomplicated
## 4 Female no appendicitis uncomplicated
## 5 Female appendicitis    uncomplicated
## # i 777 more rows
```

NC STATE UNIVERSITY

Contingency Tables

- Summarize categorical data by looking at counts!

```
app_data |>
  group_by(SexF) |>
  drop_na(SexF) |>
  summarize(count = n())
```

```
## # A tibble: 2 x 2
##   SexF   count
##   <fct>  <int>
## 1 Female   377
## 2 Male     403
```

```
app_data |>
  group_by(DiagnosisF) |>
  drop_na(DiagnosisF) |>
  summarize(count = n())
```

```
## # A tibble: 2 x 2
##   DiagnosisF      count
##   <fct>           <int>
## 1 appendicitis      463
## 2 no appendicitis   317
```

Contingency Tables

- Summarize categorical data by looking at counts across combinations of variables!

```
app_data |>
  group_by(SexF, DiagnosisF) |>
  drop_na(SexF, DiagnosisF) |>
  summarize(count = n()) |>
  pivot_wider(names_from = DiagnosisF, values_from = count)
```

```
## # A tibble: 2 x 3
## # Groups:   SexF [2]
##   SexF   appendicitis `no appendicitis`
##   <fct>         <int>              <int>
## 1 Female          200                176
## 2 Male            262                141
```

Contingency Tables

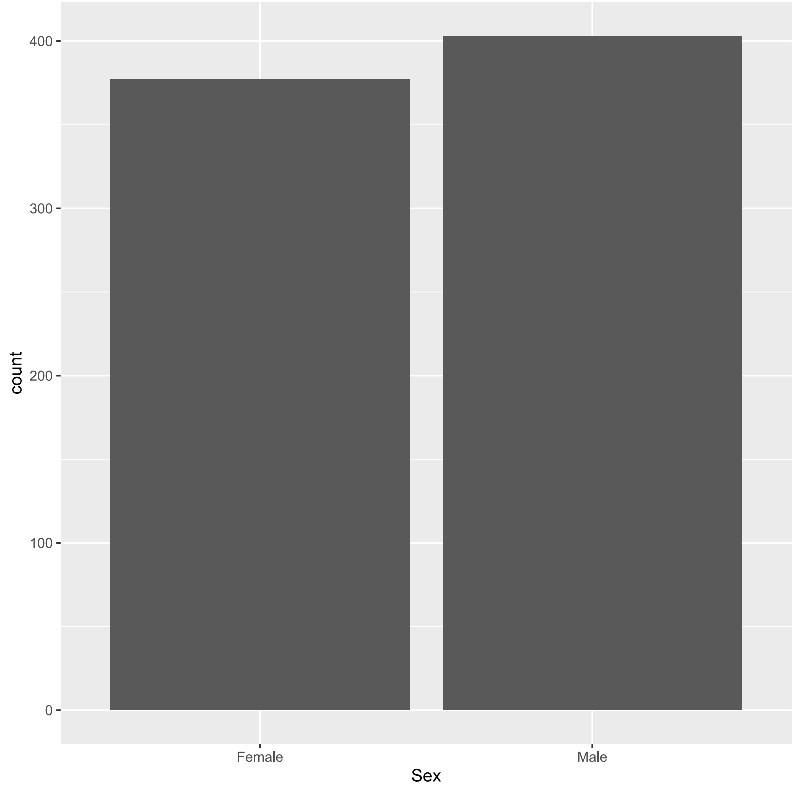
- Summarize categorical data by looking at counts across combinations of variables!

```
app_data |>
  group_by(SexF, DiagnosisF, SeverityF) |>
  drop_na(SexF, DiagnosisF, SeverityF) |>
  summarize(count = n()) |>
  pivot_wider(names_from = DiagnosisF, values_from = count)
```

```
## # A tibble: 4 x 4
## # Groups:   SexF [2]
##   SexF   SeverityF     appendicitis `no appendicitis`
##   <fct>  <fct>                <int>             <int>
## 1 Female complicated             55                 1
## 2 Female uncomplicated          145               175
## 3 Male   complicated             63                NA
## 4 Male   uncomplicated          199               141
```

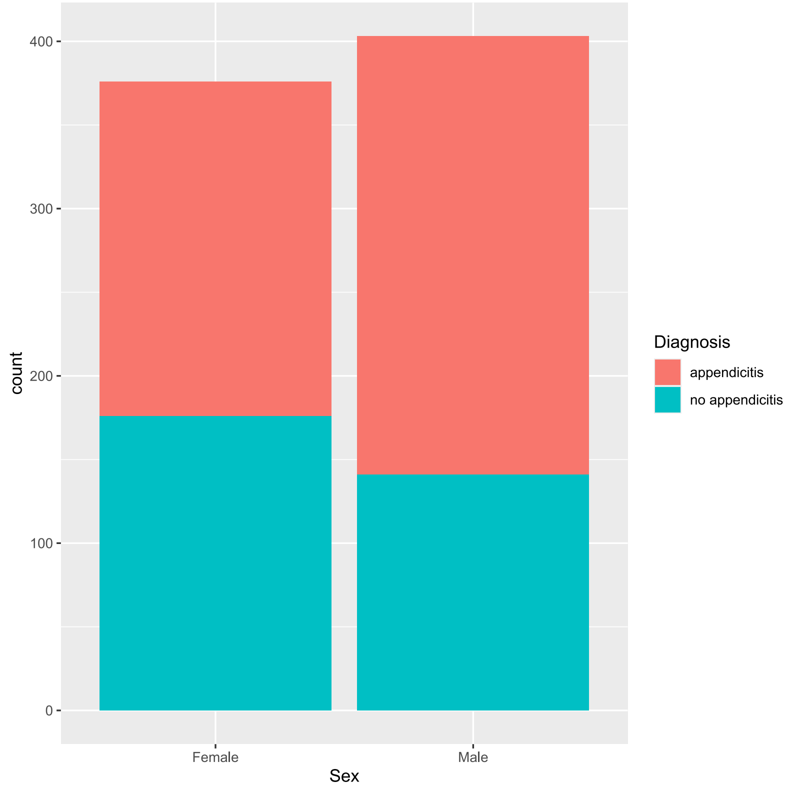
Bar Charts

- Main visual used is a bar plot! Simply displays our counts with bars.



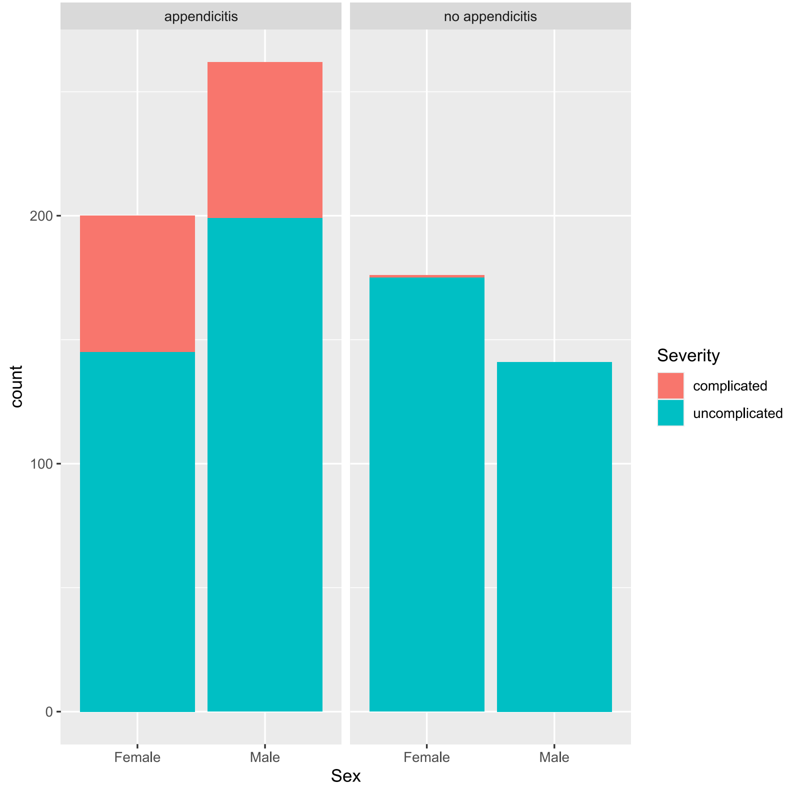
Bar Charts

- Main visual used is a bar plot! Simply displays our counts with bars.



Bar Charts

- Main visual used is a bar plot! Simply displays our counts with bars.



Numeric Data

Goal: Describe the **distribution** of the variable

- Distribution = pattern and frequency with which you observe a variable
- Numeric variable - entries are a numerical value where math can be performed

For a single numeric variable, describe the distribution via

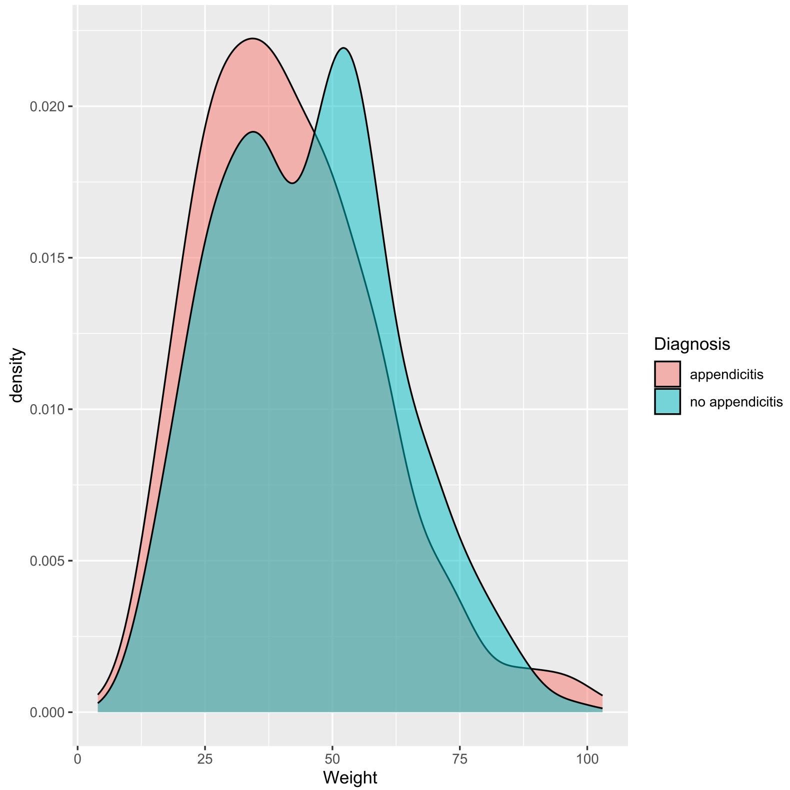
- Shape: Histogram, Density plot, …
- Measures of center: Mean, Median, …
- Measures of spread: Variance, Standard Deviation, Quartiles, IQR, …

For two numeric variables, describe the distribution via

- Shape: Scatter plot, …
- Measures of linear relationship: Covariance, Correlation

Summarizing Center and Spread

- We summarize center and spread for a numeric variable because it is difficult to compare entire distributions!
    - Consider the distributions of `Weight` for those with appendicitis and those without



Summarizing Center and Spread

- Mean and Median give good measures of the 'middle' type observations

```
app_data |>
  group_by(Diagnosis) |>
  drop_na(Diagnosis, Weight) |>
  summarize(mean_weight = mean(Weight),
            median_weight = median(Weight))
```

```
## # A tibble: 2 x 3
##   Diagnosis       mean_weight median_weight
##   <chr>                 <dbl>         <dbl>
## 1 appendicitis           41.7          39.5
## 2 no appendicitis        45.3          46.3
```

Summarizing Center and Spread

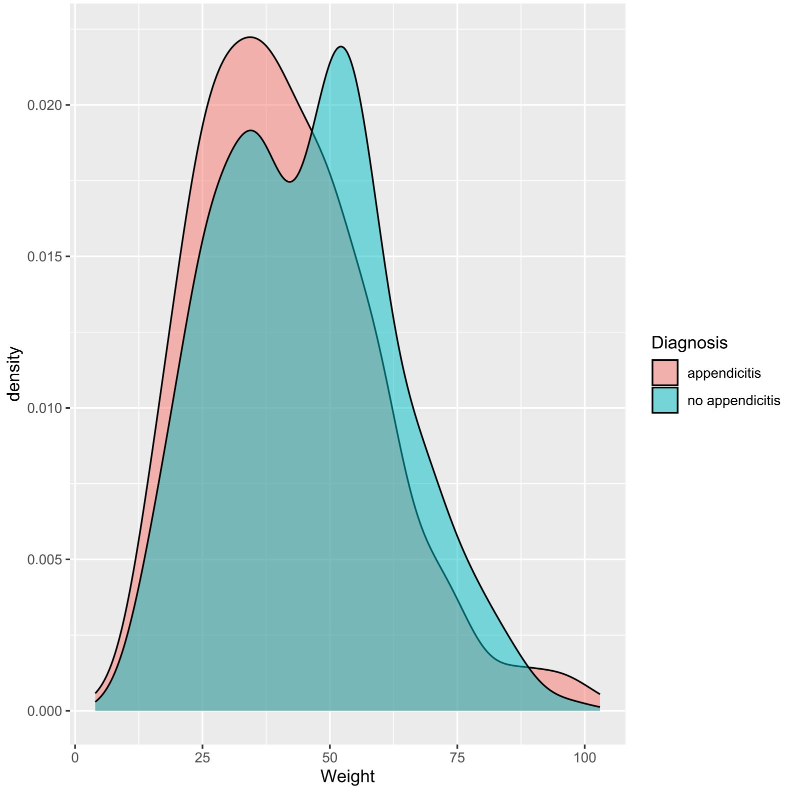
- Of course we need to understand the variability we see as well! Variance, standard deviation, and IQR are good measures of that.

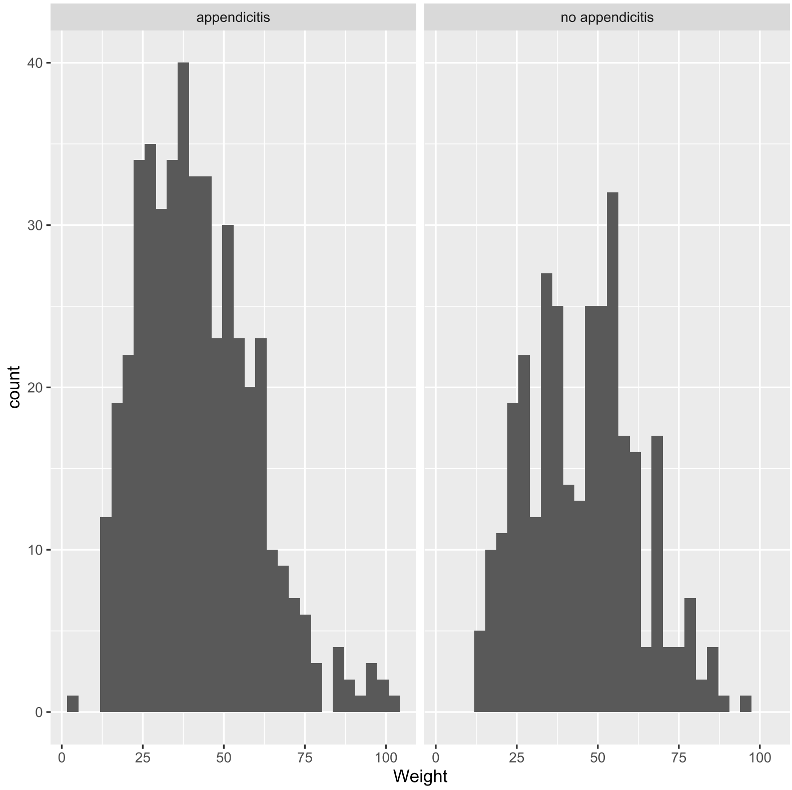
```
app_data |>
  group_by(Diagnosis) |>
  drop_na(Diagnosis, Weight) |>
  summarize(across(Weight, .fns = list("mean" = mean,
                                       "median" = median,
                                       "var" = var,
                                       "sd" = sd,
                                       "IQR" = IQR), .names = "{.fn}_{.col}"))
```

```
## # A tibble: 2 x 6
##   Diagnosis       mean_Weight median_Weight var_Weight sd_Weight IQR_Weight
##   <chr>                 <dbl>         <dbl>      <dbl>     <dbl>      <dbl>
## 1 appendicitis           41.7          39.5       305.      17.5       23.4
## 2 no appendicitis        45.3          46.3       293.      17.1       23.5
```

Summarizing Shape

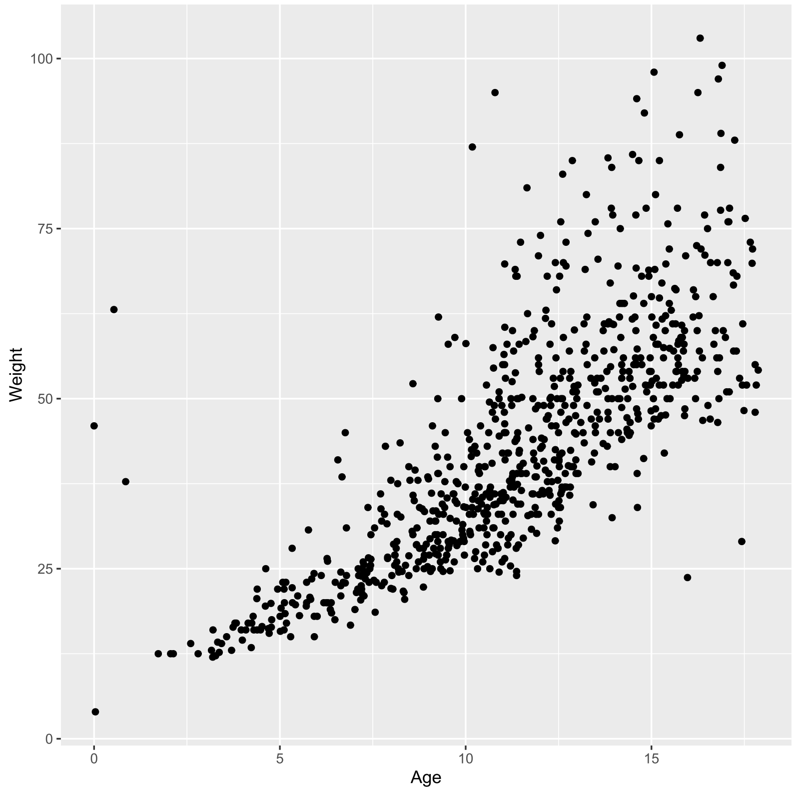
- Most easily done via histograms and density plots
  - Histograms are more variable, which can be bad!





Summarizing Two Numeric Variables

- To look at the distribution of two numeric variables together, we usually look at a scatter plot!



Summarizing Two Numeric Variables

- Again, difficult to describe the relationship generally!
    - Numerically we commonly describe the 'linear-ness' of the relationship
    - Done through covariance and correlation

```
app_data |>
  drop_na(Weight, Age) |>
  summarize(cov = cov(Weight, Age), corr = cor(Weight, Age))
```

```
## # A tibble: 1 x 2
##     cov  corr
##   <dbl> <dbl>
## 1  47.0 0.766
```

Summarizing Two Numeric Variables

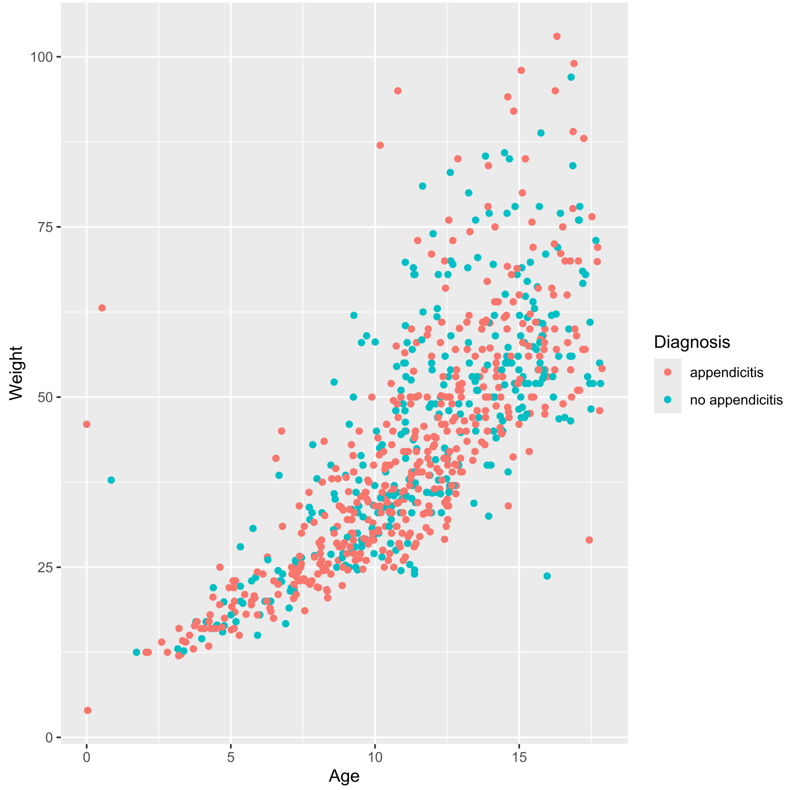
- Again, difficult to describe the relationship generally!
  - Numerically we commonly describe the 'linear-ness' of the relationship
  - Done through covariance and correlation

```
app_data |>
  drop_na(Weight, Age) |>
  summarize(cov = cov(Weight, Age), corr = cor(Weight, Age))
```

```
## # A tibble: 1 x 2
##     cov  corr
##   <dbl> <dbl>
## 1  47.0 0.766
```

Summarizing Two Numeric Variables

- Of course we want to bring in subgroups to compare them!



Summarizing Two Numeric Variables

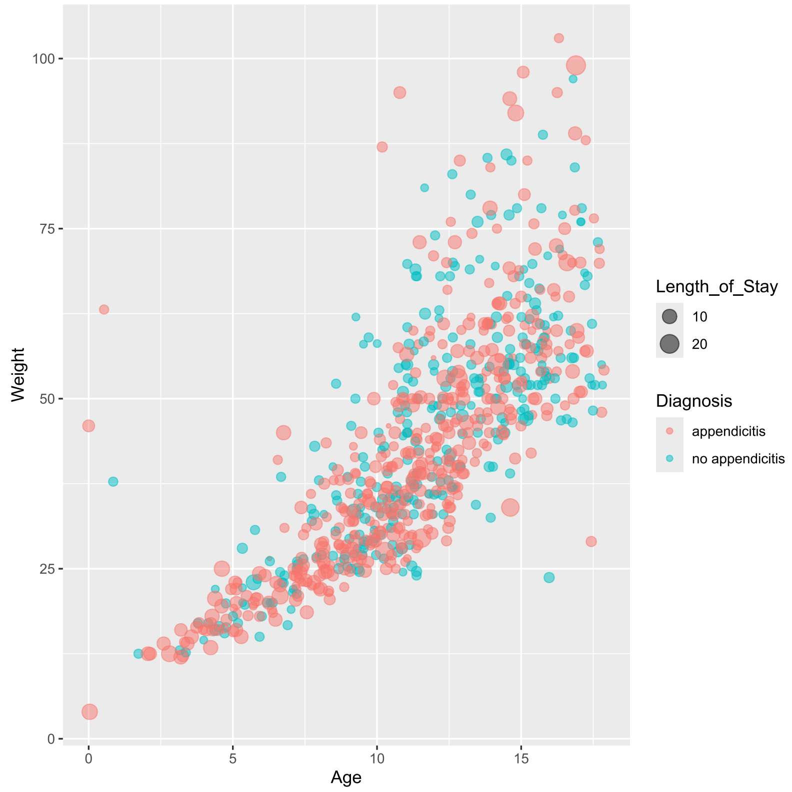
- Summarize based on groups!

```
app_data |>
  drop_na(Weight, Age, Diagnosis) |>
  group_by(Diagnosis) |>
  summarize(cov = cov(Weight, Age), corr = cor(Weight, Age))
```

```
## # A tibble: 2 x 3
##   Diagnosis         cov  corr
##   <chr>           <dbl> <dbl>
## 1 appendicitis     48.0 0.775
## 2 no appendicitis  44.3 0.748
```

Summarizing Two Numeric Variables

- We can do really interesting stuff to add in additional variables (like a third numeric variable)



Recap

- EDA is often the first step to an analysis:
  - Understand how your data is stored
  - Do basic data validation
  - Determine rate of missing values
  - Clean data up data as needed
  - Investigate distributions
    - Univariate measures/graphs
    - Multivariate measures/graphs
  - Apply transformations and repeat previous step