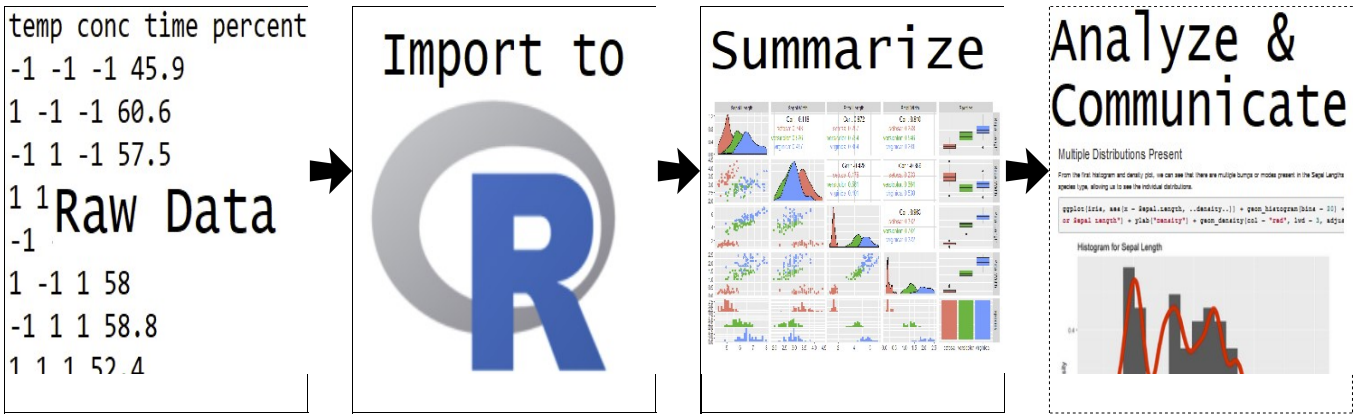
NC STATE UNIVERSITY

Summarizing Data: Descriptive Statistics

Justin Post

What is this course about?

Basic use of R for reading, manipulating, and plotting data!



What is this course about?

Basic use of R for reading, manipulating, and plotting data!

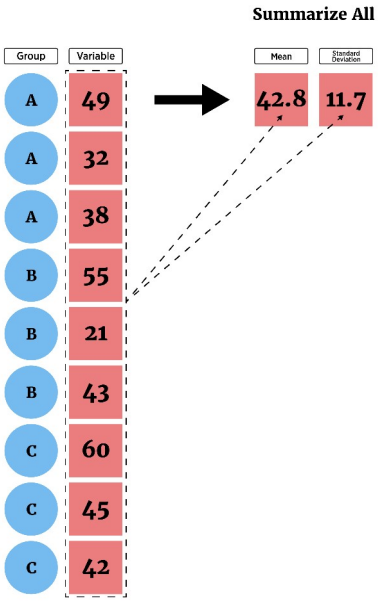
- read and write basic R programs
- import well formatted data into R
- do basic data manipulation in R
- **produce common numerical and graphical summaries in R**
- describe a use case of an analysis done in R

Where do we start?

- Understand types of data and their distributions

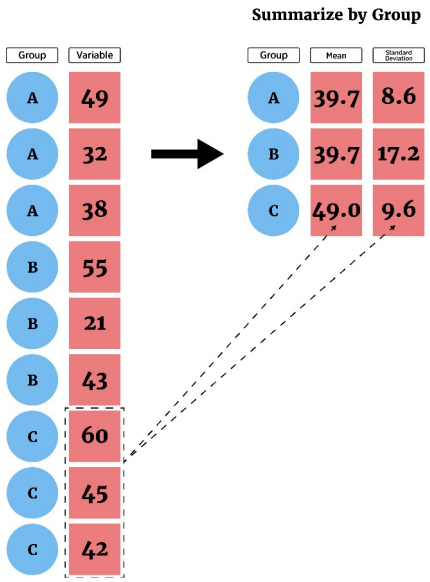
Where do we start?

- Understand types of data and their distributions
- Numerical summaries (across subgroups)



Where do we start?

- Understand types of data and their distributions
- Numerical summaries (across subgroups)



Where do we start?

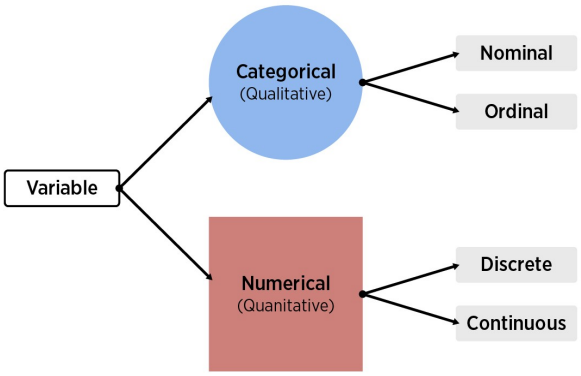
- Understand types of data and their distributions
- Numerical summaries (across subgroups)
    - Contingency Tables
    - Mean/Median
    - Standard Deviation/Variance/IQR
    - Quantiles/Percentiles

Where do we start?

- Understand types of data and their distributions
- Numerical summaries (across subgroups)
    - Contingency Tables
    - Mean/Median
    - Standard Deviation/Variance/IQR
    - Quantiles/Percentiles
- Graphical summaries (across subgroups)
    - Bar plots
    - Histograms
    - Box plots
    - Scatter plots

Understanding Data

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



Understanding Data

Common goal: Describe the **distribution** of the variable

- Distribution = pattern and frequency with which you observe a variable
- Categorical variable - describe relative frequency (or count) in each category
- Numeric variable - describe the shape, center, and spread

Contingency tables

Categorical variable - entries are a label or attribute

- Tables (contingency tables) via `table`
    - Show frequency/proportion of categories

Contingency tables

Categorical variable - entries are a label or attribute

- Tables (contingency tables) via `table`
    - Show frequency/proportion of categories
- Consider data on titanic passengers in `titanic.csv`

```
## # A tibble: 1,310 x 14
##   pclass survived name  sex      age sibsp parch ticket  fare cabin embarked
##    <dbl>    <dbl> <chr> <chr>  <dbl> <dbl> <dbl> <chr>  <dbl> <chr> <chr>
## 1      1        1 Alle~ fema~ 29         0     0 24160   211. B5    S
## 2      1        1 Alli~ male   0.917     1     2 113781  152. C22 ~ S
## 3      1        0 Alli~ fema~  2         1     2 113781  152. C22 ~ S
## 4      1        0 Alli~ male  30         1     2 113781  152. C22 ~ S
## 5      1        0 Alli~ fema~ 25         1     2 113781  152. C22 ~ S
## # ... with 1,305 more rows, and 3 more variables: boat <chr>, body <dbl>,
## #   home.dest <chr>
```

Contingency tables

- Create **one-way contingency tables** for each of three categorical variables:
  - embarked (where journey started)
  - survived (survive or not)
  - sex (Male or Female)

```
table(titanicData$embarked)
```

```
##
##   C   Q   S
## 270 123 914
```

```
table(titanicData$survived)
```

```
##
##   0   1
## 809 500
```

```
table(titanicData$sex)
```

```
##
## female   male
##    466    843
```

Two-way contingency tables

- Create **two-way contingency tables** for pairs of categorical variables

```
table(titanicData$survived,
      titanicData$sex)
```

```
##
##     female male
##   0    127  682
##   1    339  161
```

```
table(titanicData$survived,
      titanicData$embarked)
```

```
##
##       C   Q   S
##   0 120  79 610
##   1 150  44 304
```

```
table(titanicData$sex,
      titanicData$embarked)
```

```
##
##             C   Q   S
##   female 113  60 291
##   male   157  63 623
```

Three-way contingency tables

- Create a **three-way contingency table** for three categorical variables

```
table(titanicData$sex, titanicData$embarked, titanicData$survived)
```

```
## , ,  = 0
##
##
##            C   Q   S
##   female  11  23  93
##   male   109  56 517
##
## , ,  = 1
##
##
##            C   Q   S
##   female 102  37 198
##   male    48   7 106
```

Three-way contingency tables

- Create a **three-way contingency table** for three categorical variables (order matters for output!)
- Example of an array! 3 dimensions `[ , , ]`

```
tab <- table(titanicData$sex, titanicData$embarked, titanicData$survived)

str(tab)
```

```
##  'table' int [1:2, 1:3, 1:2] 11 109 23 56 93 517 102 48 37 7 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ : chr [1:2] "female" "male"
##   ..$ : chr [1:3] "C" "Q" "S"
##   ..$ : chr [1:2] "0" "1"
```

Conditional contingency tables

- Can obtain **conditional** bivariate info!

```
##  'table' int [1:2, 1:3, 1:2] 11 109 23 56 93 517 102 48 37 7 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ : chr [1:2] "female" "male"
##   ..$ : chr [1:3] "C" "Q" "S"
##   ..$ : chr [1:2] "0" "1"
```

```
#returns embarked vs survived table for females
tab[1, , ]
```

```
##
##       0   1
##   C  11 102
##   Q  23  37
##   S  93 198
```

Conditional contingency tables

- Can obtain **conditional** bivariate info!

```
##  'table' int [1:2, 1:3, 1:2] 11 109 23 56 93 517 102 48 37 7 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ : chr [1:2] "female" "male"
##   ..$ : chr [1:3] "C" "Q" "S"
##   ..$ : chr [1:2] "0" "1"
```

```
#returns embarked vs survived table for males
tab[2, , ]
```

```
##
##       0   1
##   C 109  48
##   Q  56   7
##   S 517 106
```

Conditional contingency tables

- Can obtain **conditional** bivariate info!

```
##  'table' int [1:2, 1:3, 1:2] 11 109 23 56 93 517 102 48 37 7 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ : chr [1:2] "female" "male"
##   ..$ : chr [1:3] "C" "Q" "S"
##   ..$ : chr [1:2] "0" "1"
```

```
#returns survived vs sex table for embarked "C"
tab[, 1, ]
```

```
##
##            0   1
##   female  11 102
##   male   109  48
```

Conditional contingency tables

- Can obtain **conditional** univariate info too!

```
##  'table' int [1:2, 1:3, 1:2] 11 109 23 56 93 517 102 48 37 7 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ : chr [1:2] "female" "male"
##   ..$ : chr [1:3] "C" "Q" "S"
##   ..$ : chr [1:2] "0" "1"
```

```
#Survived status for males that embarked at "Q"
tab[2, 2, ]
```

```
##  0  1
## 56  7
```

Numerical summaries: Numeric variables

Numeric variable - entries are a numerical value where math can be performed

Single variable:

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

Numerical summaries: Numeric variables

Numeric variable - entries are a numerical value where math can be performed

Single variable:

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

Two Variables:

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

Numerical summaries: Numeric variables

- Look at carbon dioxide (CO2) uptake data set
    - Response recorded: `uptake` CO2 uptake rates in grass plants
    - Environment manipulated: `Treatment` - chilled/nonchilled
    - Ambient CO2 specified and measured: `conc`

```
CO2 <- tbl_df(CO2)
CO2
```

```
## # A tibble: 84 x 5
##   Plant Type   Treatment   conc uptake
##   <ord> <fct>  <fct>      <dbl>  <dbl>
## 1 Qn1   Quebec nonchilled    95   16
## 2 Qn1   Quebec nonchilled   175   30.4
## 3 Qn1   Quebec nonchilled   250   34.8
## 4 Qn1   Quebec nonchilled   350   37.2
## 5 Qn1   Quebec nonchilled   500   35.3
## # ... with 79 more rows
```

Measures of center

Mean & Median

```
mean(CO2$uptake)
```

```
## [1] 27.2131
```

```
#note you can easily get a trimmed mean
mean(CO2$uptake, trim = 0.05) #5% trimmed mean
```

```
## [1] 27.25263
```

```
median(CO2$uptake)
```

```
## [1] 28.3
```

Measures of spread

Variance, Standard Deviation, Quartiles, & IQR

```
#quartiles and mean
summary(CO2$uptake)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.70   17.90   28.30   27.21   37.12   45.50
```

```
var(CO2$uptake)
```

```
## [1] 116.9515
```

```
sd(CO2$uptake)
```

```
## [1] 10.81441
```

```
IQR(CO2$uptake)
```

```
## [1] 19.225
```

```
quantile(CO2$uptake, probs = c(0.1, 0.2))
```

```
##   10%   20%
## 12.36 15.64
```

Measures of linear relationship

Covariance & Correlation

```
cov(CO2$conc, CO2$uptake)
```

```
## [1] 1552.687
```

```
cor(CO2$conc, CO2$uptake)
```

```
## [1] 0.4851774
```

Numerical summaries: Numeric variables

Usually want summaries for different **subgroups of data**

- Ex: Get similar uptake summaries for each **Treatment**

Numerical summaries: Numeric variables

Usually want summaries for different **subgroups of data**

- Ex: Get similar uptake summaries for each **Treatment**
- `dplyr` easy to use but can only return one value

Numerical summaries: Numeric variables

Usually want summaries for different **subgroups of data**

- Ex: Get similar uptake summaries for each **Treatment**
- `dplyr` easy to use but can only return one value

Idea:

- Use `group_by` to create subgroups associated with the data frame
- Use `summarize` to create basic summaries for each subgroup

Summarizing across groups

- Ex: Get similar uptake summaries for each **Treatment**

```
CO2 %>% group_by(Treatment) %>%
    summarise(avg = mean(uptake), med = median(uptake), var = var(uptake))
```

```
## # A tibble: 2 x 4
##   Treatment    avg   med   var
##   <fct>      <dbl> <dbl> <dbl>
## 1 nonchilled  30.6  31.3  94.2
## 2 chilled     23.8  19.7 118.
```

Summarizing across groups

- Ex: Get similar uptake summaries for each **Treatment** and **Concentration**

```
CO2 %>% group_by(Treatment, conc) %>%
        summarise(avg = mean(uptake), med = median(uptake), var = var(uptake))
```

```
## # A tibble: 14 x 5
## # Groups:   Treatment [2]
##    Treatment   conc   avg   med    var
##    <fct>      <dbl> <dbl> <dbl>  <dbl>
##  1 nonchilled    95  13.3  12.8   5.75
##  2 nonchilled   175  25.1  24.6  32.6
##  3 nonchilled   250  32.5  32.7  35.1
##  4 nonchilled   350  35.1  34.5  37.4
##  5 nonchilled   500  35.1  33.8  31.9
##  6 nonchilled   675  36.0  35.8  40.2
##  7 nonchilled  1000  37.4  37.6  49.8
##  8 chilled       95  11.2  10.6   8.18
##  9 chilled      175  19.4  19.5  34.7
## 10 chilled      250  25.3  24.2 112.
## 11 chilled      350  26.2  26.4 117.
## 12 chilled      500  26.6  26   131.
## 13 chilled      675  27.9  28.8 120.
## 14 chilled     1000  29.8  30.3 154.
```

Summarizing across groups

`dplyr` has variations on `summarise` that can be used:

- `summarise_all()` - Apply functions to every column
- `summarise_at()` - Apply functions to specific columns
- `summarise_if()` - Apply functions to all columns of one type

Summarizing across groups

- Ex: Get similar uptake summaries for each **Treatment**
- Built-in `aggregate()` function more general

Summarizing across groups

- Ex: Get similar uptake summaries for each **Treatment**
- Built-in `aggregate()` function more general
- Basic use gives response (`x`) and a `list` of variables to group `by`

```
aggregate(x = CO2$uptake, by = list(CO2$Treatment), FUN = summary)
```

```
##      Group.1   x.Min. x.1st Qu. x.Median   x.Mean x.3rd Qu.   x.Max.
## 1 nonchilled 10.60000  26.47500 31.30000 30.64286  38.70000 45.50000
## 2    chilled  7.70000  14.52500 19.70000 23.78333  34.90000 42.40000
```

Summarizing across groups

- Ex: Get similar uptake summaries for each **Treatment**
- Built-in `aggregate()` function more general
- Commonly used with `formula` notation!

```
aggregate(uptake ~ Treatment, data = CO2, FUN = summary)
```

```
##    Treatment uptake.Min. uptake.1st Qu. uptake.Median uptake.Mean
## 1 nonchilled    10.60000       26.47500      31.30000    30.64286
## 2    chilled     7.70000       14.52500      19.70000    23.78333
##   uptake.3rd Qu. uptake.Max.
## 1       38.70000    45.50000
## 2       34.90000    42.40000
```

Summarizing across groups

- Ex: Get similar uptake summaries for each **Treatment**
- Built-in `aggregate()` function more general
- Commonly used with `formula` notation!

```
aggregate(uptake ~ Treatment, data = CO2, FUN = summary)
```

`uptake ~ Treatment` - formula notation in R

- Idea: uptake (LHS) modeled by Treatment levels (RHS)

Summarizing across groups

- Ex: Get similar uptake summaries for each **Treatment** and **Concentration**
- Built-in `aggregate()` function more general
- Commonly used with `formula` notation!

```
aggregate(uptake ~ Treatment + conc, data = CO2, FUN = summary)
```

`uptake ~ Treatment + conc` model uptake by levels of Treatment and conc

Summarizing across groups

- Ex: Get similar uptake summaries for each **Treatment** and **Concentration**

```
aggregate(uptake ~ Treatment + conc, data = CO2, FUN = summary)
```

```
##     Treatment conc uptake.Min. uptake.1st Qu. uptake.Median uptake.Mean
## 1  nonchilled   95    10.60000       11.47500      12.80000    13.28333
## 2     chilled   95     7.70000        9.60000      10.55000    11.23333
## 3  nonchilled  175    19.20000       20.05000      24.65000    25.11667
## 4     chilled  175    11.40000       15.67500      19.50000    19.45000
## 5  nonchilled  250    25.80000       27.30000      32.70000    32.46667
## 6     chilled  250    12.30000       17.95000      24.20000    25.28333
## 7  nonchilled  350    27.90000       30.45000      34.50000    35.13333
## 8     chilled  350    13.00000       18.15000      26.45000    26.20000
## 9  nonchilled  500    28.50000       31.27500      33.85000    35.10000
## 10    chilled  500    12.50000       18.30000      26.00000    26.65000
## 11 nonchilled  675    28.10000       31.42500      35.80000    36.01667
## 12    chilled  675    13.70000       19.72500      28.80000    27.88333
## 13 nonchilled 1000    27.80000       32.50000      37.60000    37.38333
## 14    chilled 1000    14.40000       20.40000      30.30000    29.78333
##    uptake.3rd Qu. uptake.Max.
## 1        15.40000    16.20000
## 2        13.30000    15.10000
## 3        29.62500    32.40000
## 4        23.32500    27.30000
## 5        36.52500    40.30000
## 6        33.82500    38.10000
## 7        40.65000    42.10000
## 8        34.45000    38.80000
## 9        39.27500    42.90000
## 10       37.07500    38.90000
## 11       40.85000    43.90000
## 12       36.97500    39.60000
## 13       43.15000    45.50000
## 14       40.72500    42.40000
```

Recap/Next Up!

- Understand types of data and their distributions
- Numerical summaries
  - Contingency Tables: `table`
  - Mean/Median: `mean`, `median`
  - Standard Deviation/Variance/IQR: `sd`, `var`, `IQR`
  - Quantiles/Percentiles: `quantile`
- Across subgroups with `dplyr::group_by` and `dplyr::summarize` or `aggregate`
- Graphical summaries (across subgroups)