Intro to DS
Week 2

Jeff Saltz/Jeff Stanton

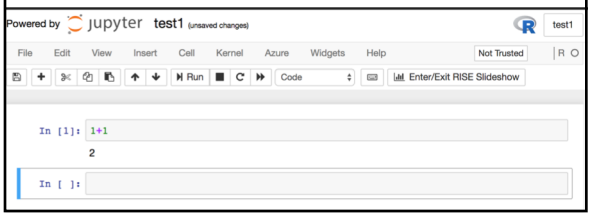
School of Information Studies
SYRACUSE UNIVERSITY
ischool.syr.edu

Objectives for This Week

- Explain/use basic concepts and techniques of data modeling
- Explain and use data frames in R
  - Data organization to facilitate R analysis
  - Rows and columns
  - Consistent mode type by attribute/variable

What are Markdown files and Notebooks?

- Provide Code, Comments and outputs in a single file
- Cells are code and output or markdown (comments)
- The most popular notebooks are Jupyter Notebooks
- Rmarkdown is part of RStudio



Integrated Code & Output

Powered by jupyter test1 (autosaved)

File Edit View Insert Cell Kernel Azure Widgets Help Trusted

Run Code Enter/Exit RISE Slideshow

Explore MPG, but round the values first

In [8]:
```
myCars <- mtcars
myCars$roundMPG <- round(myCars$mpg, -1)
str(myCars)
```

```
'data.frame':   32 obs. of  12 variables:
 $ mpg     : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl     : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp    : num  160 160 108 258 360 ...
 $ hp      : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat    : num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt      : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec    : num  16.5 17 18.6 19.4 17 ...
 $ vs      : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am      : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear    : num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb    : num  4 4 1 1 2 1 4 2 2 4 ...
 $ roundMPG: num  20 20 20 20 20 20 10 20 20 20 ...
```

How to use the concept of cells & code?

- Break logical code into different cells
- Use markdown cells as comments / explanation

Powered by jupyter test1 (autosaved)

File Edit View Insert Cell Kernel Azure Widgets Help Trusted

Run Markdown Enter/Exit RISE Slideshow

Explore MPG, but round the values first

In [8]:
```
myCars <- mtcars
myCars$roundMPG <- round(myCars$mpg, -1)
str(myCars)
```

```
'data.frame':   32 obs. of  12 variables:
 $ mpg     : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl     : num  6 6 4 6 8 6 8 4 4 6 ...
```

Not a Good Notebook

Use Cells!!!

```
In [6]: library(tidyverse)
        value <- df %>%
          #only look at Thursday, ignore spaces
          filter(str_trim(day_of_week)=="THURSDAY") %>%

          #look at vehicle count
          pull(vehicle_count) %>%

          #take the mean, store it in value
          mean(na.rm=TRUE)

        paste("mean vehicle count", value)
        'mean vehicle count 1.90728476821192'

In [7]: #an alternative way to calculate the mean
        mean( df$vehicle_count[str_trim(df$day_of_week)=="THURSDAY"], na.rm=TRUE)
        1.90728476821192
        #Step 2: Investigating the data frame by answering the following questions with explicit R code:

In [9]: #  A.   What was the total number of accidents with injuries?
        df %>% filter (injury=="YES") %>% summarize( count=n()) %>% pull(count)
        301

In [11]: #  B.   How many accidents happen on Friday?
         df %>%
           filter (str_trim(df$day_of_week)=="FRIDAY") %>%
           summarize( count=n()) %>%
           pull(count)
         151

In [15]: #  H.   Use hist( ) to make a histogram of the number of vehicles
         #       in accidents on Friday.
         df %>%
           filter(str_trim(df$day_of_week)=="FRIDAY") %>%
           pull(vehicle_count) %>%
           hist(main="# of vehicles in Accidencts on Friday")
```

Output PDF in RStudio?

- Knit to PDF
- Might need to do the following commands in R:

  install.packages("tinytex")
  library(tinytex)
  tinytex::install_tinytex()

Data Frames in R

ischool.syr.edu

School of Information Studies
SYRACUSE UNIVERSITY

Rows and Columns

One of the most basic and widely used methods of representing data is to use rows and columns, where each row is a case/instance and each column is a variable/attribute. Most spreadsheets arrange their data in rows and columns, although spreadsheets don't usually refer to these as cases or variables. R represents rows and columns in an object called a *data frame*.

- Data frames
  - Context: What is the data set about?
  - Content: What is contained in the columns?
  - Mode/Type: What are the data types of the columns
- Data frames facilitate analysis in R
  - Enforcement of rectangle
  - Row and column naming
  - Single mode/type by attribute/variable

An Example Dataset: Context

Name	Age	Gender	Weight
Dad	43	Male	188
Mom	42	Female	136
Sis	12	Female	83
Bro	8	Male	61
Dog	5	Female	44

An Example Dataset: Characteristics

Two-dimensions: rows and columns

Name	Age	Gender	Weight
Dad	43	Male	188
Mom	42	Female	136
Sis	12	Female	83
Bro	8	Male	61
Dog	5	Female	44

An Example Dataset: Characteristics

- Rows (data)
  - Cases
  - Instances
  - Observations

Name	Age	Gender	Weight
Dad	43	Male	188
Mom	42	Female	136
Sis	12	Female	83
Bro	8	Male	61
Dog	5	Female	44

Note: Name Age Gender Weight is **not** a data row.

An Example Dataset: Characteristics

- Columns (data)
  - **Variable name**
  - Attributes
  - Variables

Name	Age	Gender	Weight
Dad	43	Male	188
Mom	42	Female	136
Sis	12	Female	83
Bro	8	Male	61
Dog	5	Female	44

An Example Dataset: Characteristics

- Columns (data)
  - **Attributes**
  - **Variables**
  - Variable name

Name	Age	Gender	Weight
Dad	43	Male	188
Mom	42	Female	136
Sis	12	Female	83
Bro	8	Male	61
Dog	5	Female	44

- Note: Name Age Gender Weight are **not** data

An Example Dataset: Characteristics

- In a well-structured data set, each row has a unique identifier (case label).
- R supports "row names" or you can build in a data field to serve as unique ID
- R does NOT enforce the uniqueness of a row, as some DBs do

Name	Age	Gender	Weight
Dad	43	Male	188
Mom	42	Female	136
Sis	12	Female	83
Bro	8	Male	61
Dog	5	Female	44

An Example Dataset: Characteristics

- Each column has the same type/mode of data.
- Each column has the same number of entries.

Name	Age	Gender	Weight
Dad	43	Male	188
Mom	42	Female	136
Sis	12	Female	83
Bro	8	Male	61
Dog	5	Female	44

Creating a dataset in R

- Data set: How does this get built in R?
  - Create a vector for each variable (column).
  - Create a data frame to combine individual vectors.

Name	Age	Gender	Weight
Dad	43	Male	188
Mom	42	Female	136
Sis	12	Female	83
Bro	8	Male	61
Dog	5	Female	44

Question:

- How would you represent the following data in a data frame?
  - Students in a class
    - For each student, we have a student ID and a GPA.
    - Student 1: ID: N1; GPA: 3.8
    - Student 2: ID: N2; GPA: 4.0
    - Student 3: ID: N3; GPA: 3.3
    - Student 4: ID: N4; GPA: 3.5
    - Student 5: ID: N5; GPA: 3.9

→Create a grid (table) to show this information

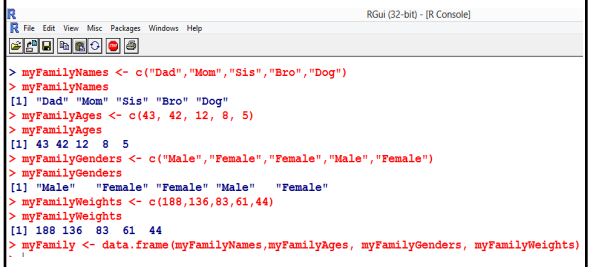
Answer:

- How would you represent the following data in a data frame?

Student ID	Student GPA
N1	3.8
N2	4.0
N3	3.3
N4	3.5
N5	3.9

Creating a Dataframe in R

1) Create Vectors
2) Use the data.frame function



Viewing a Dataframe

Display the contents of the data object MyFamily.

```
> myFamily <- data.frame(myFamilyNames,myFamilyAges, myFamilyGenders, myFamilyWeights)
> myFamily
  myFamilyNames myFamilyAges myFamilyGenders myFamilyWeights
1           Dad           43            Male            188
2           Mom           42          Female            136
3           Sis           12          Female             83
4           Bro            8            Male             61
5           Dog            5          Female             44
>
```

Using the R "Str" (Structure) Command

```
> myFamily
  myFamilyNames myFamilyAges myFamilyGenders myFamilyWeights
1           Dad           43            Male            188
2           Mom           42          Female            136
3           Sis           12          Female             83
4           Bro            8            Male             61
5           Dog            5          Female             44
> str(myFamily)
'data.frame':   5 obs. of  4 variables:
 $ myFamilyNames  : Factor w/ 5 levels "Bro","Dad","Dog",..: 2 4 5 1
 $ myFamilyAges   : num  43 42 12 8 5
 $ myFamilyGenders: Factor w/ 2 levels "Female","Male": 2 1 1 2 1
 $ myFamilyWeights: num  188 136 83 61 44
```

What does the structure function tell us about the data object myFamily?

- Confirmation that MyFamily is a data frame;
- MyFamily has five observations (cases/instances) and four variables.
- "$" for each variable/component column with descriptive information.
- Each of the variables has a mode or type (same mode within a variable/column).
- Variable is either a "factor" or "num".
- "Factor" variable has a "level".
- "Level" describes the options within a variable.
- "num" variable indicates "numeric".

Using the R Summary Command

```
> summary(myFamily)
 myFamilyNames  myFamilyAges myFamilyGenders myFamilyWeights
 Bro:1         Min.   : 5    Female:3        Min.   : 44.0
 Dad:1         1st Qu.: 8    Male  :2        1st Qu.: 61.0
 Dog:1         Median :12                    Median : 83.0
 Mom:1         Mean   :22                    Mean   :102.4
 Sis:1         3rd Qu.:42                    3rd Qu.:136.0
               Max.   :43                    Max.   :188.0
```

What does the summary function tell us about the data object myFamily?

- "Factor" variables list variable names (e.g., MyFamilyNames) along with the number of occurrences of cases that are coded within that factor.
- Numeric variables have six different calculated quantities that help summarize the variable:
  - Min / Max—minimum (or maximum) value of all cases
  - 1st Qu—dividing line at the top of the 1st quartile
  - Median—value of the case that splits the whole group in half
  - Mean—numeric average
  - 3rd Qu—3rd quartile

Accessing Dataframes

Accessing Dataframes as a vector:

```
#returns the second element in myFamilyAges
myFamily$myFamilyAges[2]
```

Accessing Dataframes as a matrix:

```
#returns the data element in the first row and first column
myFamily[1,1]

#returns the first row
myFamily[1,]

#returns the first column
myFamily[,1]

#returns everything but the 1st row  (i.e. deletes the 1st  row)
myFamily[-1,]

#returns everything but the first
> myFamily[,-1]
```

Sorting a Dataframe

Why doesn't 'sort' work?

> sort(mtcars)

Error in `[.data.frame`(x, order(x, na.last = na.last, decreasing = decreasing)) :

undefined columns selected

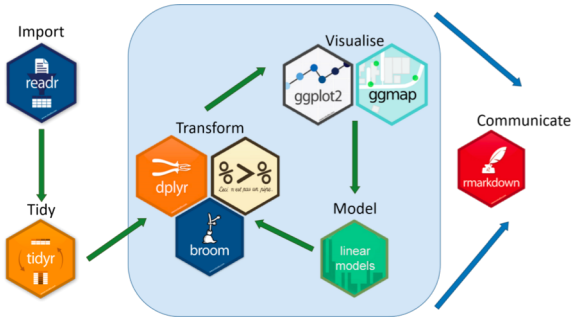
Sorting a Dataframe

```
#sorting
indexes <- order(mtcars[,"mpg"])
indexes[1]
[15]

mtcars[15,]
                    mpg cyl disp  hp drat   wt  qsec vs am gear carb
Cadillac Fleetwood 10.4   8  472 205 2.93 5.25 17.98  0  0    3    4

sortedDF <- mtcars[indexes,]
```

The Tidyverse (partial view)



Tibbles vs. DataFrames

- A "tibble" updates the behavior of a dataframe to make the structure more useful.
- Tibbles are part of the "tidyverse."
- Try 'glimpse' (vs 'str')

```
#> # A tibble: 1,000 x 5
#>   a                   b             c     d e
#>   <dttm>              <date>      <int> <dbl> <chr
#> 1 2019-01-08 17:33:11 2019-01-15      1 0.368 h
#> 2 2019-01-09 11:38:20 2019-01-20      2 0.612 n
#> 3 2019-01-09 06:02:00 2019-01-30      3 0.415 l
#> 4 2019-01-08 19:23:17 2019-01-29      4 0.212 x
#> 5 2019-01-08 15:47:33 2019-01-26      5 0.733 a
#> 6 2019-01-09 02:48:30 2019-01-22      6 0.460 v
#> # … with 994 more rows
```

Tibbles:

- Print nicely on the console
- Provide type information
- Tibbles avoid unnecessary type conversions and support improved column naming

Exploring a DataFrame

How to explore a dataframe

```
#Base R
str(mtcars)
```

'data.frame': 32 obs. of 11 variables:
$ mpg : num 21 21 22.8…
$ cyl : num 6 6 4 6 8 6…

```
#The tidyverse way
library(tidyverse)
glimpse(mtcars)
```

Rows: 32
Columns: 11
$ mpg <dbl> 21.0, 21.0, 22.8, 21.4…
$ cyl <dbl> 6, 6, 4, 6, 8, 6, 8, 4, 4…

Selecting a Column

How to select one column

```
#Base R
mtcars[ , "mpg"]
mtcars[ , 1]
mtcars$mpg
```

[1] 21.0 21.0 22.8 21.4 18.7...

```
#The tidyverse way
select(mtcars, "mpg")
select(mtcars, mpg)
```

	mpg
Mazda RX4	21.0
Mazda RX4 Wag	21.0
Datsun 710	22.8

Selecting Columns

How to select more than one column

```
#Base R
mtcars[ , c("mpg", "hp")]
mtcars[ ,c( 1,4)]
```

	mpg	hp
Mazda RX4	21.0	110
Mazda RX4 Wag	21.0	110
Datsun 710	22.8	93

```
#The tidyverse way
select(mtcars, c("mpg", "hp"))
select(mtcars, mpg, hp)
```

	mpg	hp
Mazda RX4	21.0	110
Mazda RX4 Wag	21.0	110
Datsun 710 2	2.8	93

Selecting Rows

How to select one or more rows

```
#Base R
mtcars[ 1:3,]
```

	mpg	cyl	disp	hp
Mazda RX4	21.0	6	160	110
Mazda RX4 Wag	21.0	6	160	110

```
#The tidyverse way
slice(mtcars,1:3)
```

	mpg	cyl	disp	hp
Mazda RX4	21.0	6	160	110
Mazda RX4 Wag	21.0	6	160	110

Tidyverse Pipes

Passes data from the command before to the one after
- Data flows left to right

Allows the code read more like a sentence

"%>%" is the "symbol" for a pipe

Example: min(x) is the same as:

#(x data get passed to "min" function)

```
x %>% min
```

#Simple example

```
x <- c(4,1,10,5)
x %>% min
[1] 1
```

#Dataframes example

```
mtcars %>% slice(1:3)
```

	mpg	cyl	disp	hp
Mazda RX4	21.0	6	160	110
Mazda RX4 Wag	21.0	6	160	110

Filtering DataFrames

(i.e., Subset Rows Based on a Condition)

How to create a subset of rows based on a condition for a column

#Base R

```
mtcars[mtcars$mpg > 28, ]
```

	mpg	cyl	disp	hp	drat
Fiat 128	32.4	4	78.7	66	4.08
Honda Civic	30.4	4	75.7	52	4.93

#The tidyverse way

```
mtcars %>% filter(mpg > 28)
```

	mpg	cyl	disp	hp	drat
Fiat 128	32.4	4	78.7	66	4.08
Honda Civic	30.4	4	75.7	52	4.93

Adding a Column

How to create a subset of rows based on a condition for a column

#Base R

```
myCars <- mtcars
carName=rownames(myCars)
myCars$carName <- carName
myCars [1,
  c("mpg", "cyl", "carName", "hp")]
```

	mpg	cyl	carName	hp
Mazda RX4	21	6	Mazda RX4	110

#The tidyverse way

```
myCars <- mtcars
myCars <-
rownames_to_column(myCars, var =
"carName")
myCars %>%
  slice(1) %>%
  select(mpg, cyl, carName, hp)
```

	mpg	cyl	carName	hp
1	21	6	Mazda RX4	110

Grouping Information

Do a summary-type report

```
#Base R
mean(myCars[myCars$cyl==4,"mpg"])
[1] 26.66364

mean(myCars[myCars$cyl==6,"mpg"])
[1] 19.74286

mean(myCars[myCars$cyl==8,"mpg"])
[1] 15.1
```

```
#The tidyverse way
myCars %>%
    group_by(cyl) %>%
        summarize(mpg=mean(mpg),
                  hp=mean(hp),
                  .groups = 'drop')
# A tibble: 3 x 3
  cyl   mpg   hp
  <dbl> <dbl> <dbl>
1 4     26.7  82.6
2 6     19.7  122
3 8     15.1  209
```

Sorting a DataFrame

```
#Base R
indexes <- order(mtcars[,"mpg"])
sortedDF <- mtcars[indexes,]
str(sortedDF)

"data.frame": 32 obs. of 11 variables:
 $ mpg : num 10.4 10.4 13.3…
 $ cyl : num 8 8 8 8 8 8…
```

```
#The tidyverse way
sortedDF <- mtcars %>%
                arrange(mpg)
glimpse(sortedDF)

Rows: 32
Columns: 11
$ mpg  <dbl> 10.4, 10.4, 13.3,…
$ cyl  <dbl> 8, 8, 8, 8, 8, 8, 8,…
```

Sample Questions

#Given the following lines of code were executed

```
names <- c("Jeff", "Pat", "Joe")
height <- c(100,103,120)
myFamily <- data.frame(names, height)
```

#What is returned from the following commands?

```
myFamily[1,1]
myFamily[1,]
select(myFamily, height)
myFamily %>% slice( c(1,3) )
```

Sample Answers

myFamily[1,1]
[1] "Jeff"

myFamily[1,]
names height
1 Jeff 100

select(myFamily, height)
height
1 100
2 103
3 120

myFamily %>% slice(c(1,3))
names height
1 Jeff 100
2 Joe 120

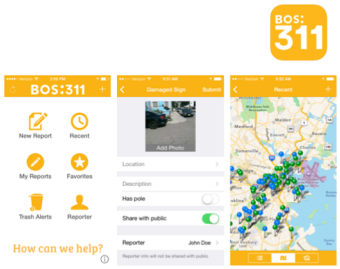
Data Science in the Real World

ischool.syr.edu

School of Information Studies
SYRACUSE UNIVERSITY

Boston Potholes: Discussion

The BOS:311 app helps residents and visitors improve City neighborhoods. You can report non-emergency issues, like potholes and graffiti.



Questions

- Example Focus – Fix potholes
- Where's the data science?
- What might be an issue?