

Intro to R

Intro to R

- R is a language and environment that allows:
 - Data management
 - Graphs and tables
 - Statistical analyses
 - You will need: some basic statistics
 - We will discuss these
- R is open source and runs on Windows, Mac, Linux systems

R Environment

- R is an integrated software suite that includes:
 - Effective data handling
 - A suite of operators for array/matrix calculations
 - Intermediate tools for data analysis
 - Graphical facilities
 - Simple and effective programming language which includes conditionals, loops, functions, I/O

R

-
- Goals for this section of the course include:
 - Becoming familiar with Statistical Packages
 - Creating new Datasets
 - Importing & exporting Datasets
 - Manipulating data in a Dataset
 - Basic analysis of data (mainly descriptive statistics with some inferential statistics)
 - An overview of R's advanced features

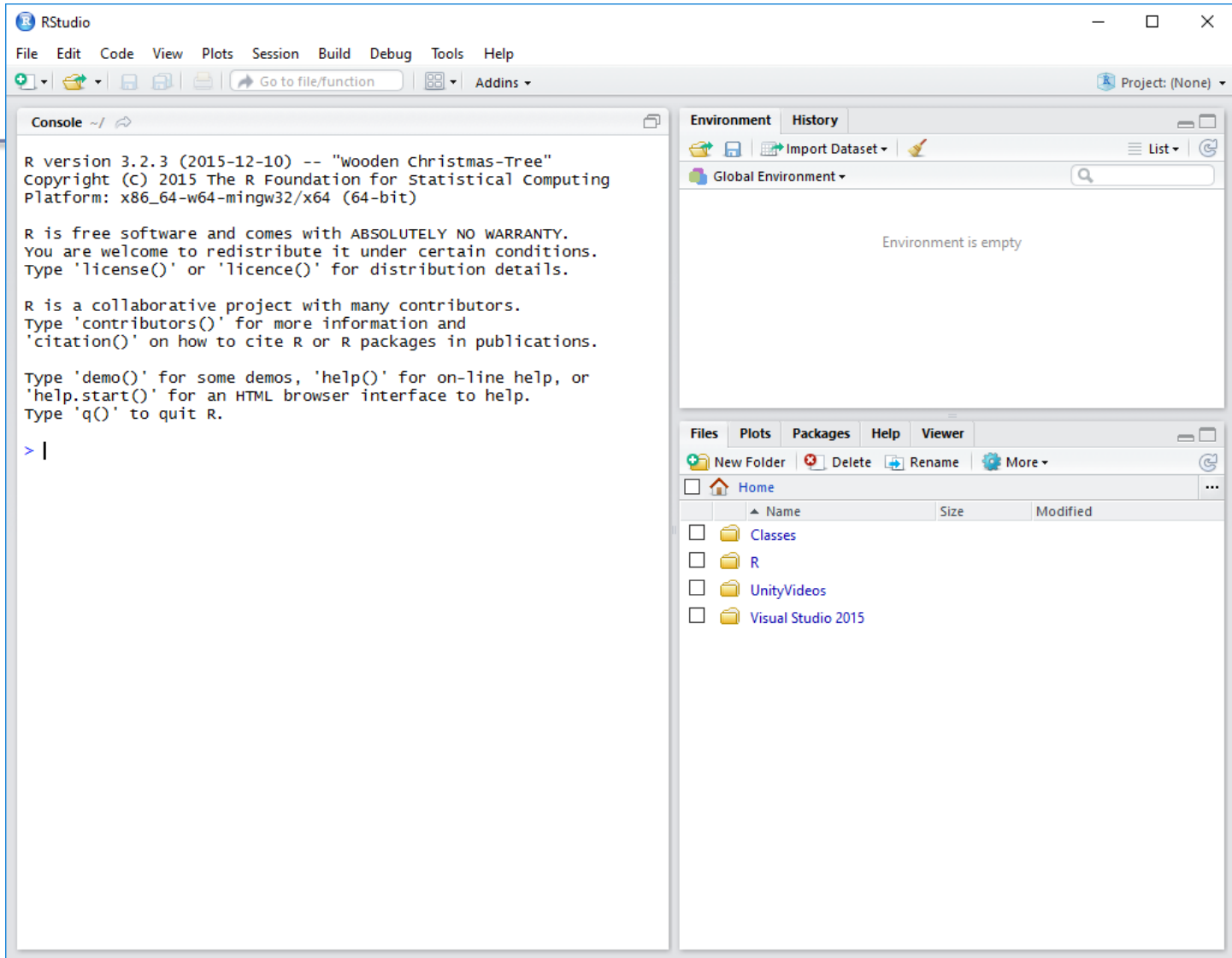
Note: This is not a statistics course such as Math 207. We will only concentrate on basic statistical concepts.

R Resources

- Web site resources:
 - R console application only
 - <https://cran.r-project.org/>
 - Rstudio IDE
 - <https://www.rstudio.com/products/rstudio/download/>
 - <https://cran.rstudio.com/>
 - R documentation
 - <http://www.tutorialspoint.com/r/index.htm>
 - <http://www.cyclismo.org/tutorial/R/index.html>

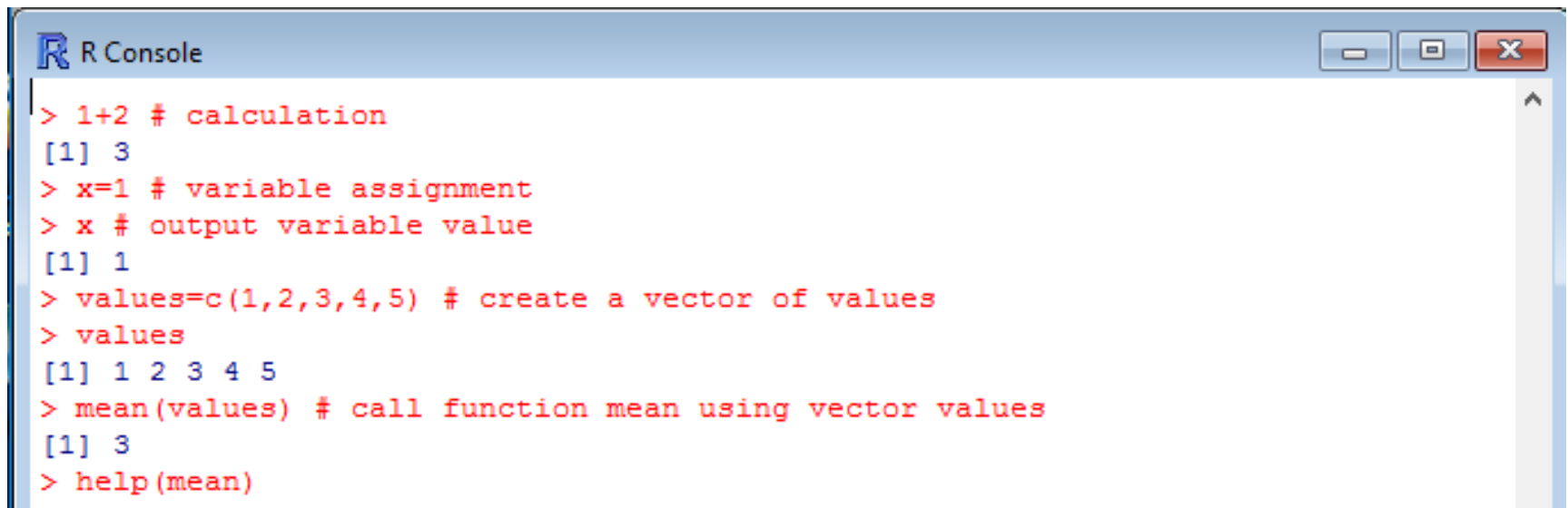
<https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf>

Open RStudio



R Session

- Start an RStudio session
- We will use the console window of RStudio



```
R Console
> 1+2 # calculation
[1] 3
> x=1 # variable assignment
> x # output variable value
[1] 1
> values=c(1,2,3,4,5) # create a vector of values
> values
[1] 1 2 3 4 5
> mean(values) # call function mean using vector values
[1] 3
> help(mean)
```

Basic Datatypes

- There are four basic datatypes in R:
 - **Numeric:** numbers with decimal points
 - **Logical:** binary – true or false
 - **Character:** any text
 - **Integer:** whole numbers only

Basic Datatypes

Numeric

- Numeric – the default datatype for numbers
 - Contains a decimal point

```
> x=10.5 # numeric
> k=1 # still numeric
> is.integer(k)
[1] FALSE
> |
```

Basic Datatypes

Logical

- Logical – is either TRUE or FALSE

```
> x = 1; y = 2; z = 1 # assign values to variables
> a = x < y # is x smaller than y ?
> a
[1] TRUE
> b = y == z # is y equal to z ?
> b
[1] FALSE
> |
```

Basic Datatypes

Character

- Character – is used to represent **text** values

```
> firstName = "Computer"
> lastName = " Science"
> firstName
[1] "Computer"
> paste (firstName, lastName) # concatenates values together
[1] "Computer Science"
> pi = as.character (3.14) # force 3.14 to be string
> class (pi)
[1] "character"
> pi * 2 # what happens
```

Basic Datatypes

Integer

- Integer – created using `as.integer ()` function or suffix `L` as in `2L`
 - No decimal point
 - Only use integer in interface with another software package or to save space (memory)

```
> k=as.integer(1)
> k
[1] 1
> is.integer(k)
[1] TRUE
> x=2
> is.integer(x)
[1] FALSE
> j=2L
> is.integer(j)
[1] TRUE
> j
[1] 2
```

Data Structures

<http://adv-r.had.co.nz/Data-structures.html>

- Combine multiple pieces of data into one variable
- Atomic Vector – often just called *vector*
 - Sequence of data of the same type (1, 2, 3, 9)
- Generic Vector/Lists
 - Sequence of data of many types (100, 200, "oak")
- Matrix
 - Grid of data of the same type $\begin{bmatrix} 1 & 9 \\ 2 & 3 \end{bmatrix}$
- Data Frame
 - Grid of data of many types $\begin{bmatrix} 100 & 200 & \text{"oak"} \\ 32 & 40 & \text{"maple"} \end{bmatrix}$

Vector

- A sequence of data of the same type
- Six types of atomic vectors

1. Logical

```
> v1=c(1,2,3)
```

2. Integer

```
> v2=4:6
```

3. Double (Numeric)

```
> v3=7.1:10.1
```

4. Character

```
> v4=seq(1.1,1.9,by=0.1)
```

```
> v3
```

```
[1] 7.1 8.1 9.1 10.1
```

5. Complex

```
> v4
```

```
[1] 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9
```

6. Raw

- For now we will concern ourselves with 1-4.

Measures of Central Tendency

- Used to describe the center of a distribution
- Define each of the following:
 - Mean
 - Median
 - Mode

Problems

- 1) Create a vector of ages in a variable called age with the following integer values: 18, 19, 18, 21, 22, 23, 19, 18
- 2) Compute the mean and median of the age values
- 3) Compute the mean of the first 1000 natural numbers

Problem

- Given the following dataset, find the mean, median, and mode of the Age variable using R

Breed	Age	Weight
Collie	2	23.2
Collie	3	35.7
Setter	5	45.4
Shepard	1	65.9
Setter	2	72.2

An R Solution

- First of all, what do we expect the answers to be?
- Let's use R to check expected results:
 1. Create a vector **age** with the Age values
 2. Call function mean
 3. Call function median
 4. Call function mode

Did we get our expected results?

Data Frame

- A data frame is a two-dimensional (2D) structure where
 - column data refers to a variable
 - row data refers to an observation or a case
- Column names are to be unique non-empty.
- Row names are optional but should be unique.
- Allowable types of variable info: numeric, factor or character type.

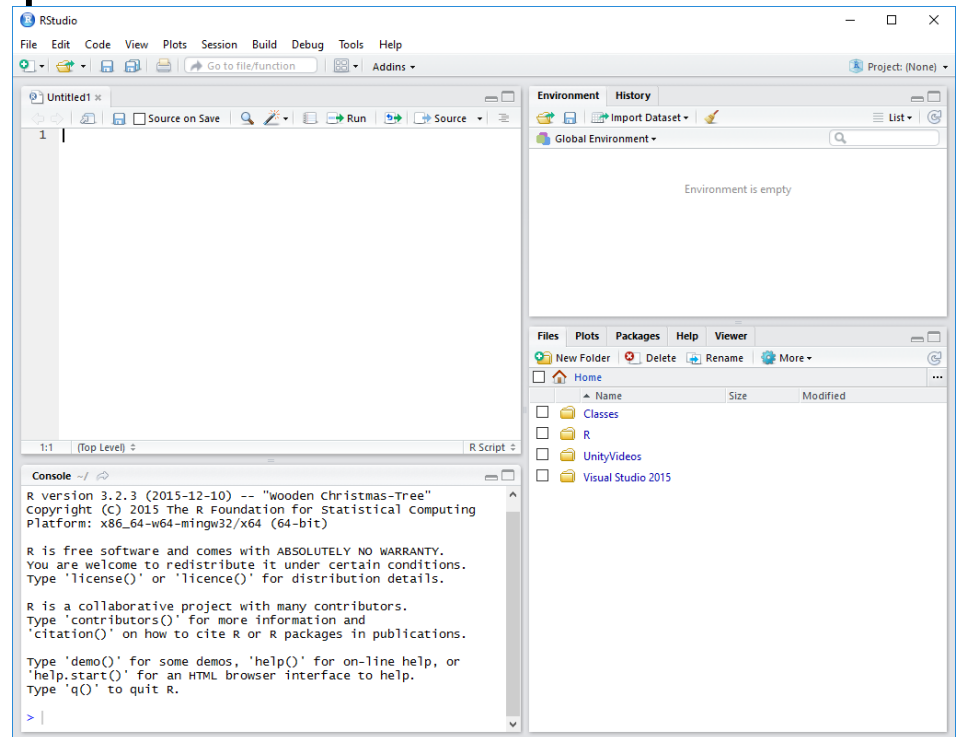
Dog Data Frame Example

- What type is Breed?
Age?
Weight?

Breed	Age	Weight
Collie	2	23.2
Collie	3	35.7
Setter	5	45.4
Shepard	1	65.9
Setter	2	72.2

Dog Data Frame

- We are going to start creating scripts in Rstudio
- File->New File->R Script



Dog Data Frame

- In the Untitled script window, type the following R script

```
# Create the data frame for dog data.
```

```
breed = c("Collie", "Collie", "Setter", "Shepard", "Setter")
```

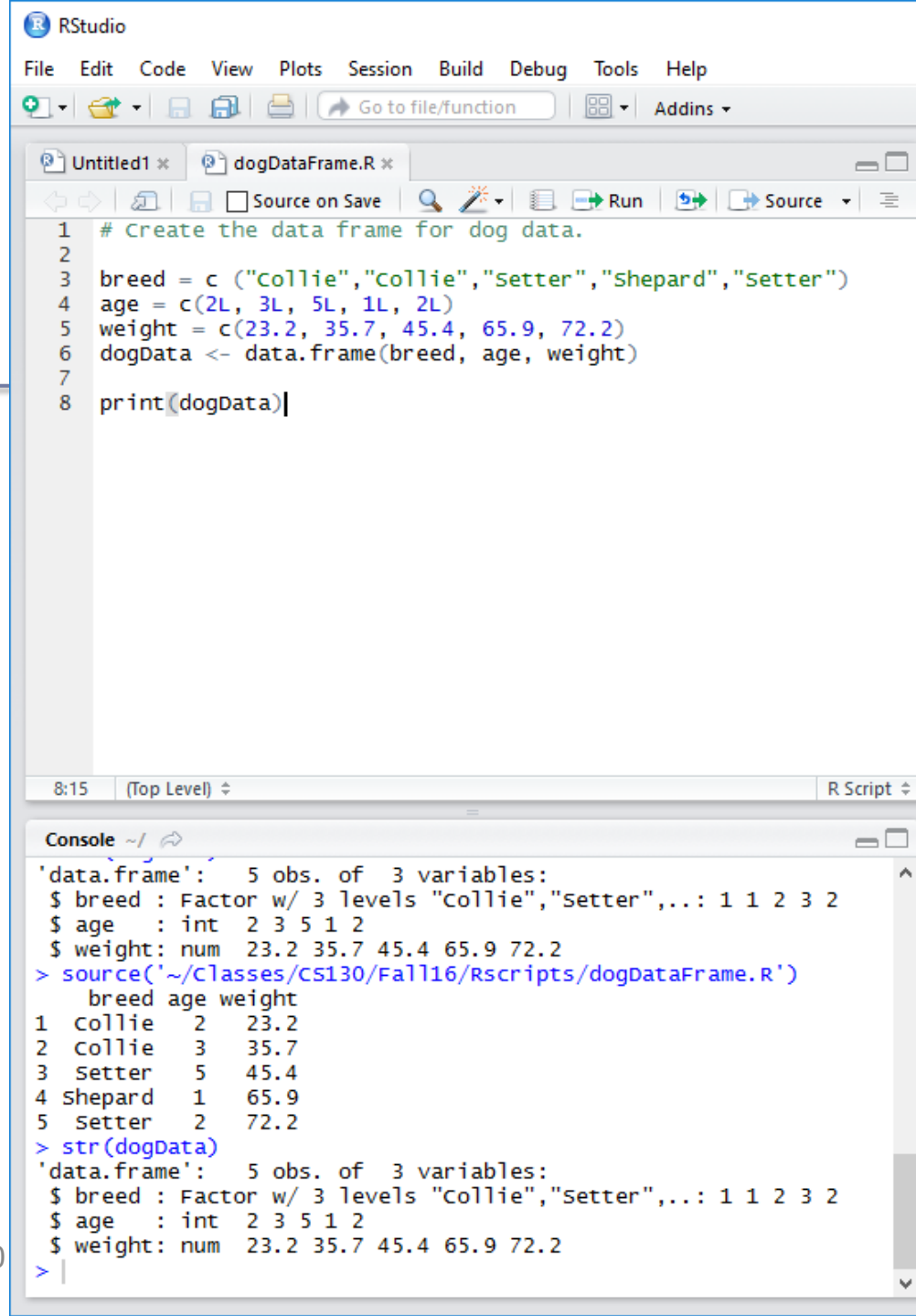
```
age = c(2L, 3L, 5L, 1L, 2L)
```

```
weight = c(23.2, 35.7, 45.4, 65.9, 72.2)
```

```
dogData <- data.frame(breed, age, weight)
```

```
print(dogData)
```

Execute the script



```
RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
Addins
Untitled1 x dogDataframe.R x
Source on Save Run Source
1 # Create the data frame for dog data.
2
3 breed = c ("collie","collie","setter","shepard","setter")
4 age = c(2L, 3L, 5L, 1L, 2L)
5 weight = c(23.2, 35.7, 45.4, 65.9, 72.2)
6 dogData <- data.frame(breed, age, weight)
7
8 print(dogData)

8:15 (Top Level) R Script

Console ~/
'data.frame': 5 obs. of 3 variables:
 $ breed : Factor w/ 3 levels "collie","setter",...: 1 1 2 3 2
 $ age : int 2 3 5 1 2
 $ weight: num 23.2 35.7 45.4 65.9 72.2
> source('~/.Classes/CS130/Fall16/Rscripts/dogDataframe.R')
 breed age weight
1 collie 2 23.2
2 collie 3 35.7
3 setter 5 45.4
4 shepard 1 65.9
5 setter 2 72.2
> str(dogData)
'data.frame': 5 obs. of 3 variables:
 $ breed : Factor w/ 3 levels "collie","setter",...: 1 1 2 3 2
 $ age : int 2 3 5 1 2
 $ weight: num 23.2 35.7 45.4 65.9 72.2
> |
```

Problems

- Find the mean and median of the age and weight variables. Use the console window to do this.

Hint: Variables of a Data Frame can be specified as `dataframe$variable` (e.g. `dogData$age`)

Variables in R

- Let's define the following terms
- Variable
 - Categorical (or Qualitative) Variable
 - Nominal
 - Ordinal
 - Quantitative Variables
 - Numeric
 - Discrete
 - Continuous

Qualitative vs. Quantitative

- Qualitative: classify individuals into categories
- Quantitative: tell how much or how many of something there is

- Which are qualitative and which are quantitative?
 - Person's Age
 - Person's Gender
 - Mileage (in miles per gallon) of a car
 - Color of a car

Qualitative: Ordinal vs. Nominal

- Ordinal variables:
 - One whose categories have a natural ordering
 - Example: grades
- Nominal variables:
 - One whose categories have no natural ordering
 - Example: state of residence

Factor

- Factors are used to represent categorical data.
- Can be:
 - Ordered – use `ordered()`
 - Unordered – use `factor()`
- Factors are stored as integers, and have labels associated with these unique integers
- Once created, factors can only contain a pre-defined set of values, known as levels. By default, R sorts levels in alphabetical order

Create Ordinal Values

<http://www.statmethods.net/input/valuelabels.html>

```
classRank=c(1, 1, 2, 1, 3)
```

```
classRankOrdinal = ordered(classRank,  
levels=c(1,2,3,4),  
labels=c("Fr", "So", "Jr", "Sr") )
```

```
print(classRankOrdinal)
```

```
barplot(summary(classRankOrdinal))
```

Why do we want ordinal values?

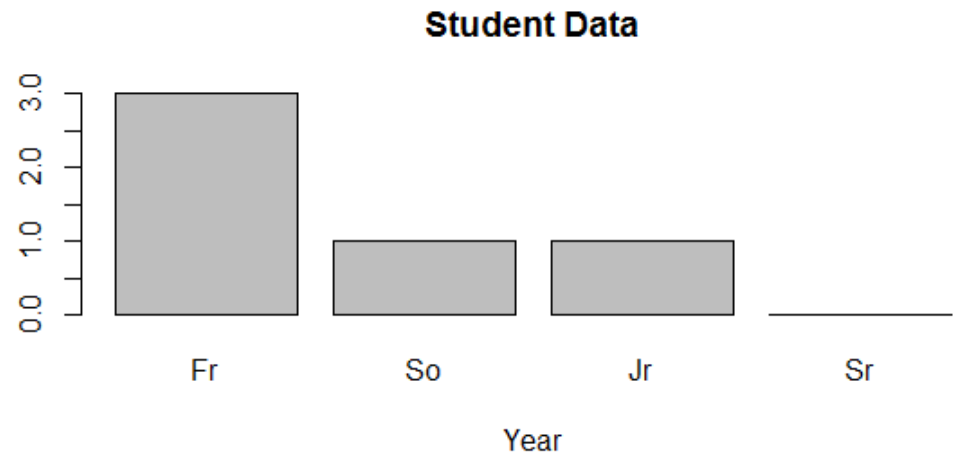
```
classRankNotOrdinal=("Fr", "Fr", "So", "Fr", "Jr")  
barplot(table(classRankNotOrdinal))
```

Bar Chart

<http://statmethods.net/graphs/bar.html>

- A **bar chart** or **bar graph** is a chart that presents grouped data with rectangular bars with lengths proportional to the values that they represent.
- function `table` returns a vector of frequency data

```
> barplot(table(classRankOrdinal),  
main = "Student Data",  
xlab = "Year")
```



Quantitative

- Discrete variables: Variables whose possible values can be listed
 - Example: number of children
- Continuous variables: Variables that can take any value in an interval
 - Example: height of a person

Problem

- Using the command `str(dogData)`, identify:
 - variable name
 - quantitative or qualitative
 - discrete, continuous, neither
 - nominal, ordinal, neither
- A specific variable can be selected and passed to the class function. Pass the variable `age` of `dogData` to `class`. What does the result tell us?

Importing Data into R

- `getwd()`
- `data = read.table("filename.txt", header=FALSE)`
- Copy `testData.txt` from CS130 Public to the location provided by `getwd()`
- Open `testData.txt` in a text editor
- `testData = read.table("testData.txt", header=TRUE)`
- `print(testData)`
- `str(testData)`

Candy Dataset Example

<http://zeus.cs.pacificu.edu/chadd/cs130w17/candy.txt>

This file contains a header

Brand	Name	ServingPerPkg	OzPerPkg	Calories	TotalFatInGrams	SatFatInGrams
M&M/Mars	Snickers Peanut Butter	1.0	2.00	310	20.0	7.0
Hershey	Cookies 'n Mint	1.0	1.55	230	12.0	6.0
Hershey	Cadbury Dairy Milk	3.5	5.00	220	12.0	8.0
M&M/Mars	Snickers	3.0	3.70	170	8.0	3.0
Charms	Sugar Daddy	1.0	1.70	200	2.5	2.5

Write dataframe to file

```
write.table( dataframe, "file.txt")  
getwd()
```

```
write.table(candy, "candy.txt")
```

Go to Documents and open candy.txt in a text editor

Problem

- Identify each of the following for Total Fat in Grams:
 - Minimum:
 - Maximum:
 - Mean:
 - Standard Deviation:

Use the help feature!