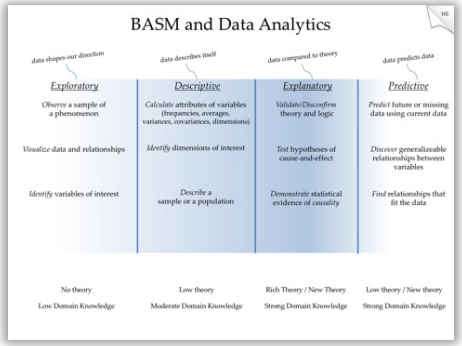
BACS:

Business Analytics Using Computational Statistics

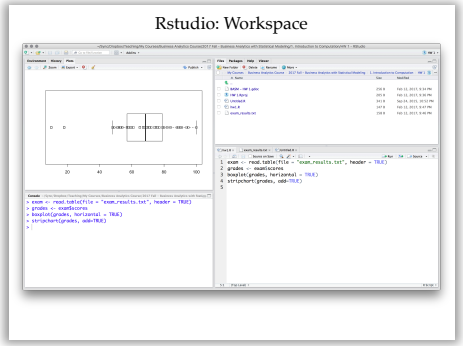
Week 1
Computation and Statistics

Week 2
Describing and Visualizing Data

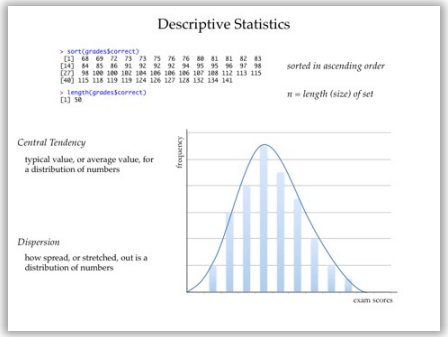
Computational Statistics



Our Tools



First Look at R



Canvas

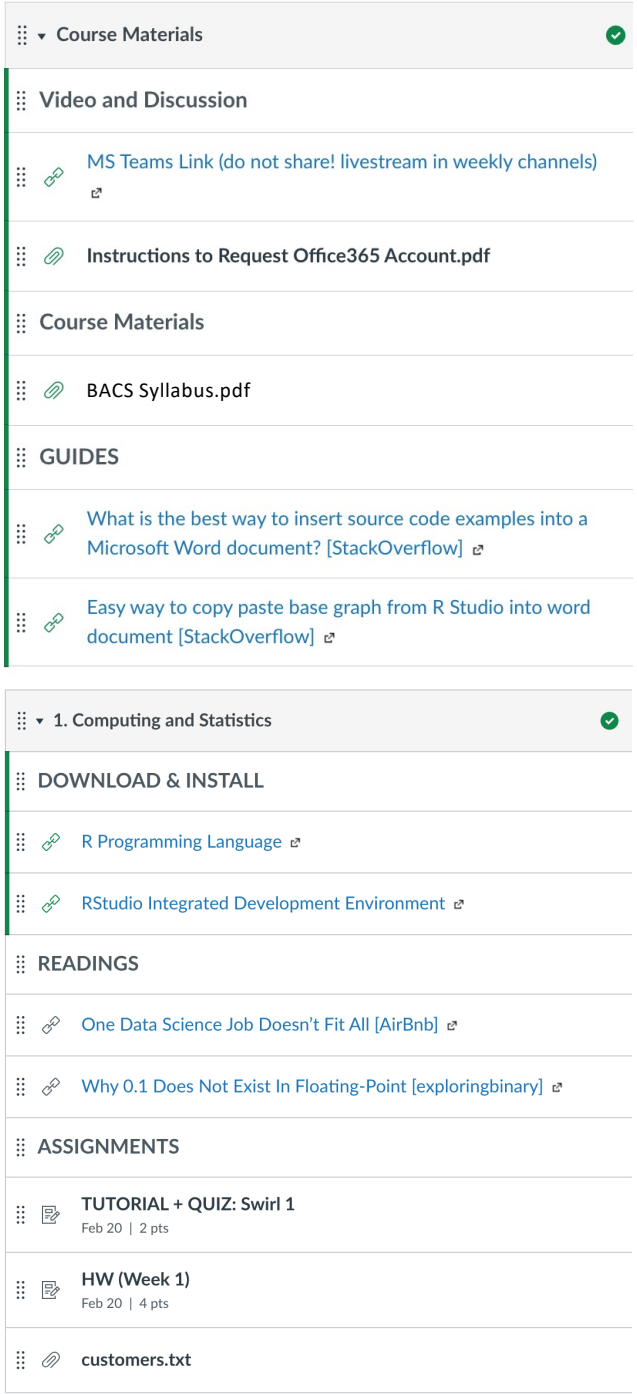
https://canvas.instructure.com/courses/4295061

Link to our MS Teams space to discuss/watch livestream →

See guide on how to apply/activate your **Office 365** account →

Do readings/viewings before starting assignments →

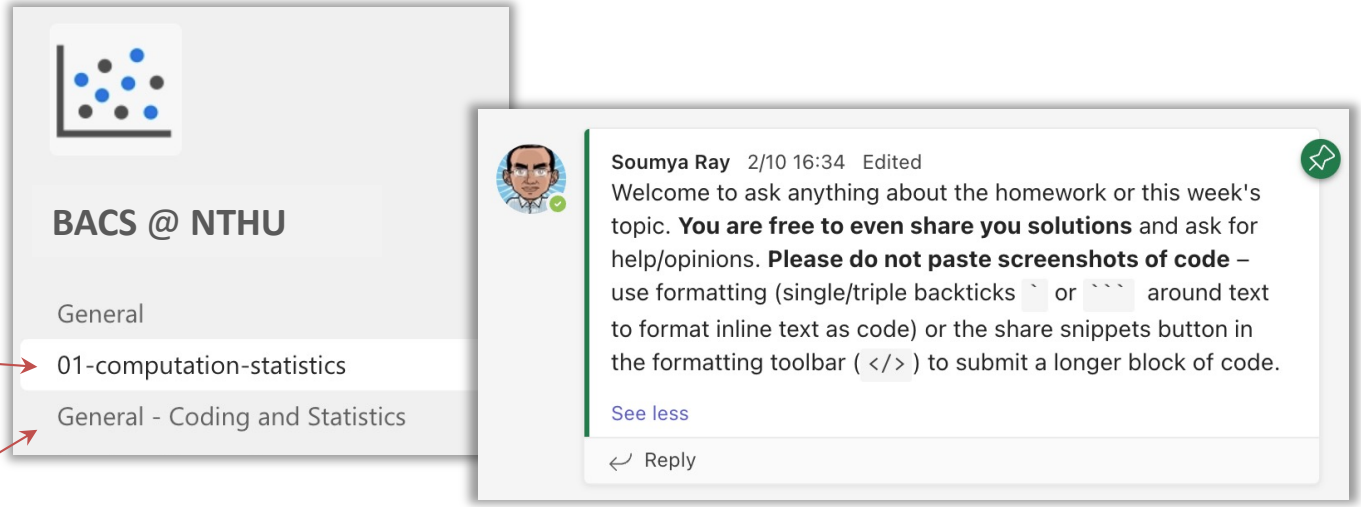
Submit assignment solutions as **PDF** reports →
Download data files for assignments



MS Teams

We will use MS Teams for all **discussion** and **livestreams**

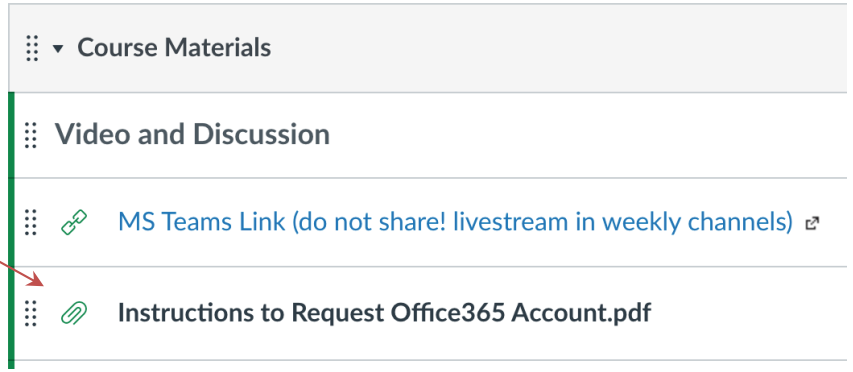
See weekly **channels** for *discussions* and *livestream*.



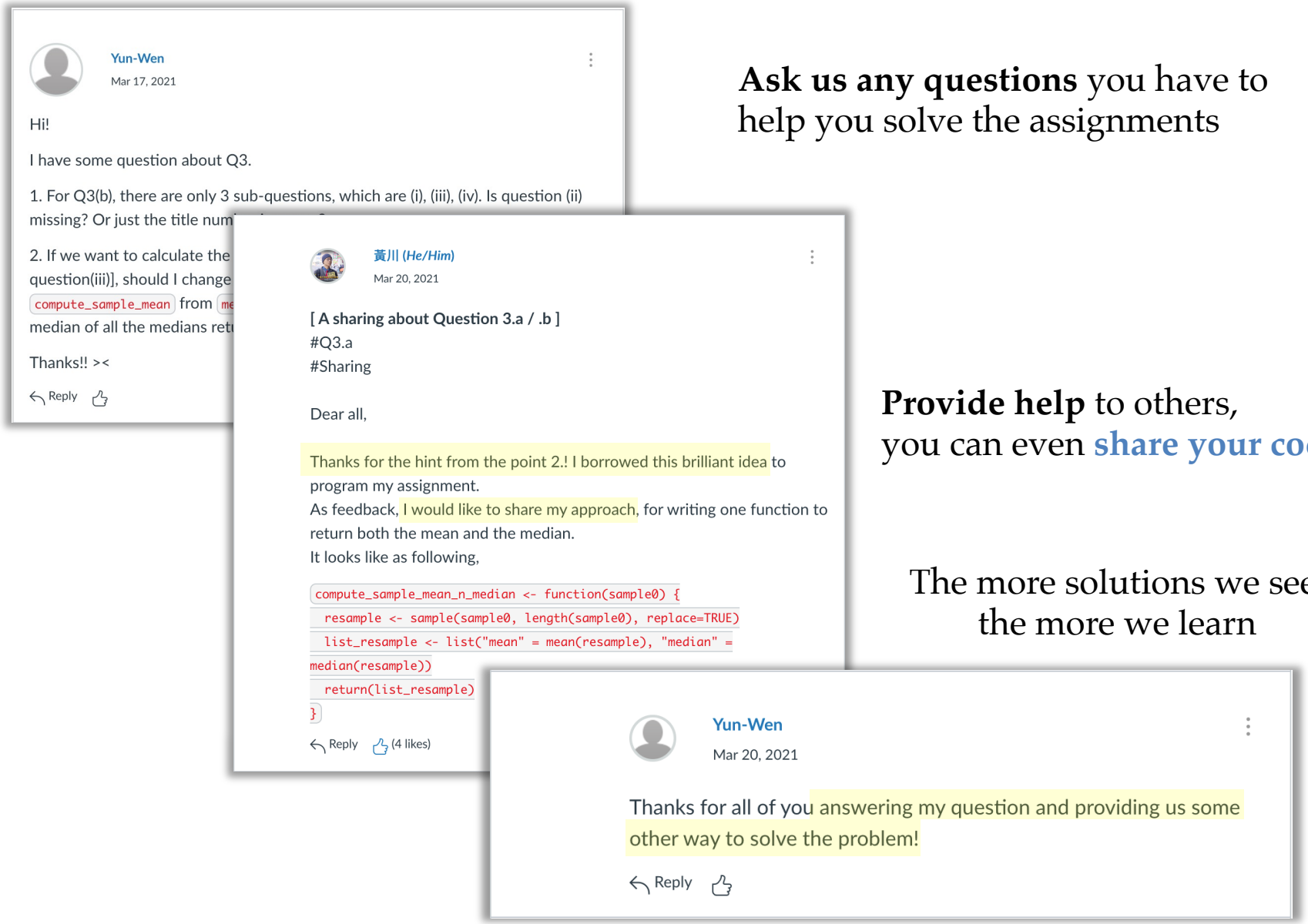
Ask or suggest coding & statistics resources

Ask/share new issues as new conversations

Log into MS Team using your NTHU Office365 account
- see **PDF on Canvas** on how to apply/activate your 365 account



Public Sharing is not Cheating!



Ask us any questions you have to help you solve the assignments

Provide help to others, you can even **share your code!**

The more solutions we see, the more we learn

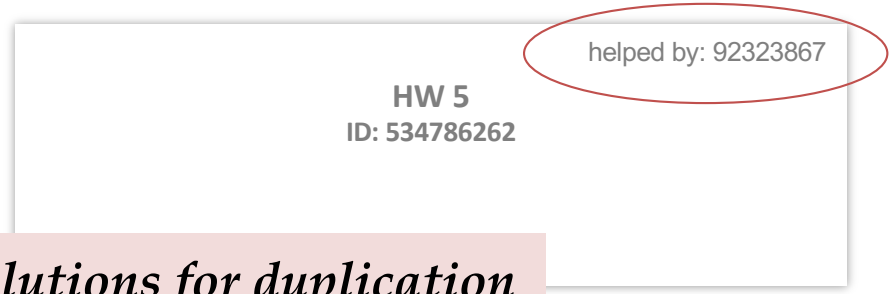
Working on Homework Together is Encouraged

Be **inspired by others' solutions**, but **do not use someone else's code exactly**

Play with your own implementation of their idea
Show your originality in thinking *the solution through*

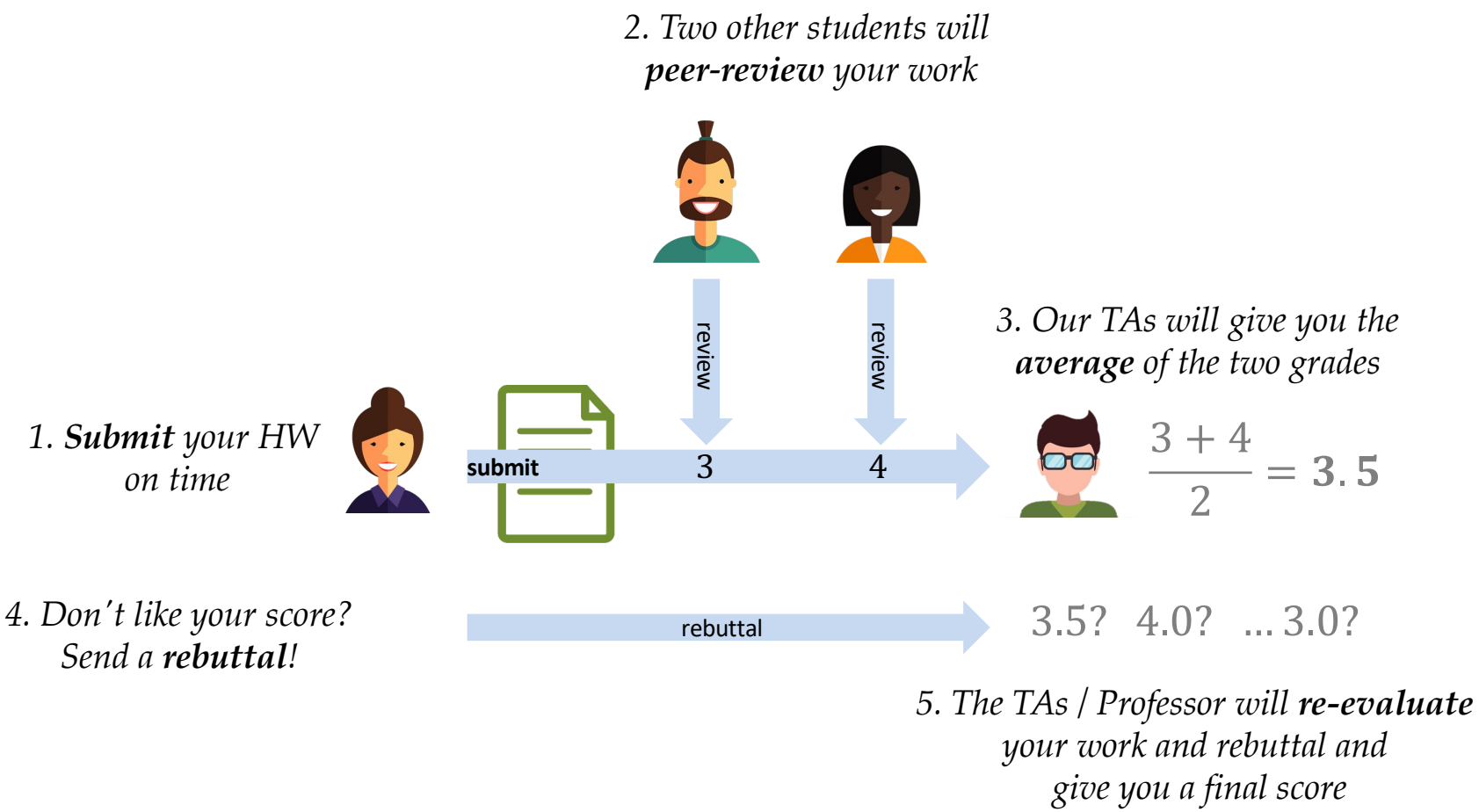
You can **work with others** on HWs, but **do not just copy/paste their answers**

Give them credit on your assignment *(student ID)*
to let us know they were very helpful

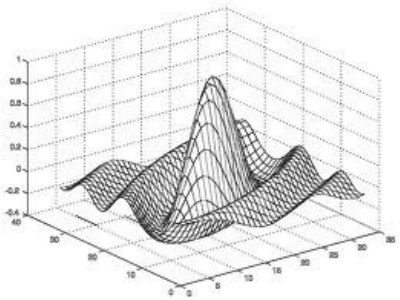


We will regularly check solutions for duplication

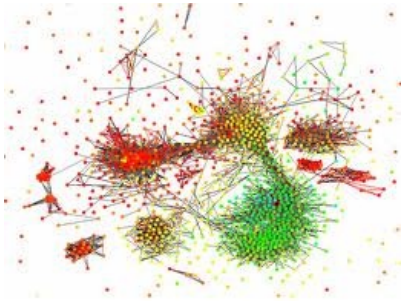
Peer Review (from Week 2 onwards)

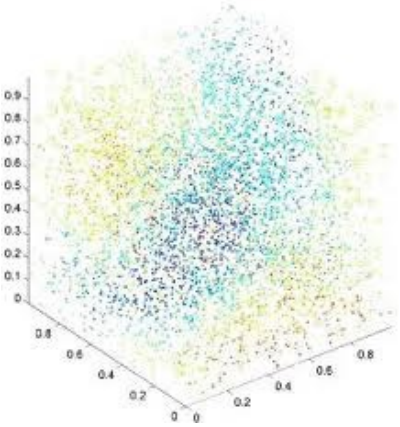


BACS: Understanding **Shape** and **Relationships** in Data

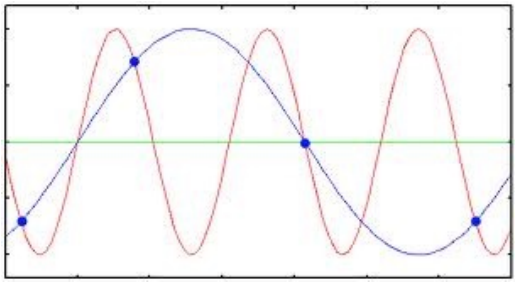


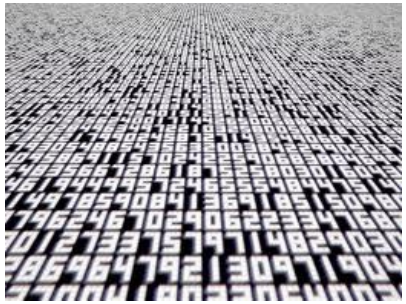
Uncovering
geometry and ***dimensionality***
of data



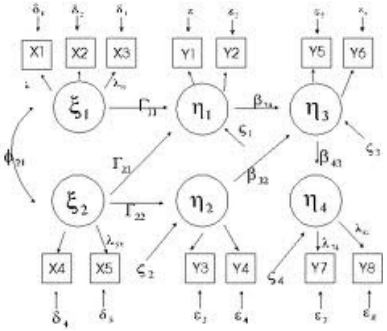


Removing the ***noise***
Finding the ***signal***





Understanding ***relationships***
Defining complex ***models***



BACS and Statistics

sample data suggests new directions	**sample data** describes itself	**sample data** compared to **theory**	**new data + model** predicts **outcomes**
Exploration	***Description***	***Inference***	***Prediction***
Observe a sample of a phenomenon	*Calculate* attributes of variables (frequencies, averages, variances, covariances, dimensions)	*Validate/Disconfirm* theory and logic	*Predict* future or missing data using current data
Visualize data and relationships	*Identify* dimensions of interest	*Test* hypotheses of cause-and-effect	*Discover* generalizeable relationships between variables
Identify variables of interest	*Describe* a sample's statistics	*Estimate* population statistics	*Find* relationships that might fit the data
No theory available	**Low theory**	**Rich Theory / New Theory**	**Low theory / New theory**
Low Domain Knowledge	**Moderate Domain Knowledge**	**Strong Domain Knowledge**	**Strong Domain Knowledge**

Class Syllabus

1. Computation

Computation and Statistics
Learning Computation
Exploration, Inference, and Prediction
Our Tools: R and Rstudio

Description and Simulation
Kernel Density Plots / Histograms
Simulating Distributions
Inferential Statistics

Computational Intervals
Functions and Iterations
Describing Distributions
Confidence Intervals
Resampling

2. Non-parametrics Tests

Bootstrapping
Review of Descriptives
Classical Hypothesis Testing
Bootstrapping the Alternative

Nonparametric Testing
Bootstrapped Hypothesis Testing
Empirical Distributions and Power

Permutation Tests
Reshaping Data
Permutation of Data Samples
Wilcoxon Test: Permutation vs. Sum of Ranks

Multigroup Tests
Normality and Quantiles – the QQ Plot
ANOVA: Parametric Test for Multiple Groups
Kruskal Wallis: Nonparametric Test of Independent Groups

3. Models: Signal vs. Noise

Data Similarity
Data as Vectors
Similarity: Cosine, Correlation
Item-Item Collaborative Filtering

Linear Regression
Review of Linear Regression
Geometric Perspective of Regression
Linear Algebraic Representation of Regression

Applied Regression
The Hat Matrix
Diagnosing and Managing Non-Linearity
Diagnosing and Managing Multi-Collinearity

Moderation and Mediation
The Contingency Perspective as Moderation
Partial Orthogonalization
Bootstrapped Test of Indirect Effects

4. Data Dimensions

Composites and Components
Multi-item Constructs
Principal Components
Transforming Dimensions
Reducing Dimensions

Principal Components Analysis
Composite Variables
Composites vs. Factors
Component Rotation as Perspective

Structural Equation Modeling
Structural Models
Composite Structural Models
Common Factor Structural Models

5. Learning and Predicting

Predictions
Out-of-sample Predictions
Split-sample Testing
k-Fold Cross Validation

Ensemble Predictions
Stable vs. Unstable Algorithms
Bagging Algorithms
Boosting Algorithms

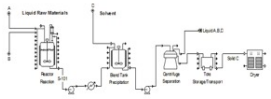
Validation and Conclusions
Hyperparameter Tuning
Validation Sets
What's Next?

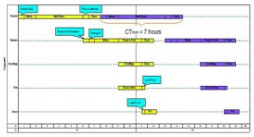
Computational Statistics in Practice

Management

Scientific Management

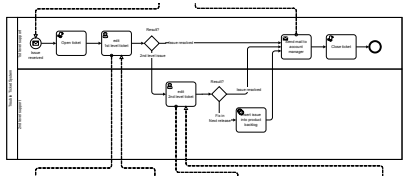
"Scientific Management"
"Inventory Control"
"Total Quality Management"





Business Process Management

1. Poorly defined and inconsistent practices
2. Repeatable practices at the workgroup level
3. Standard organization-wide end-to-end processes
4. **Statistically-managed and predictable processes**
5. Continuous process innovation and optimization



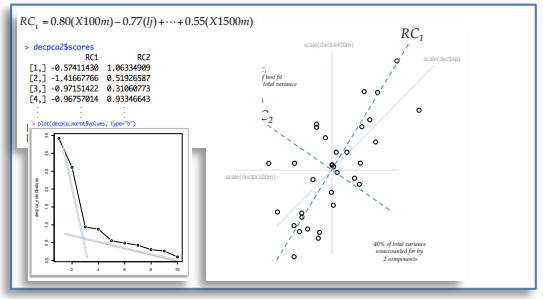
Developing Products/Services

Service Features

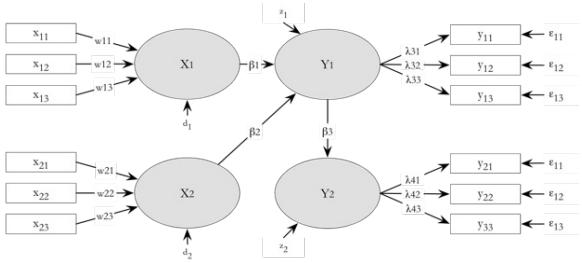
Product/Service Testing

Scientific Research

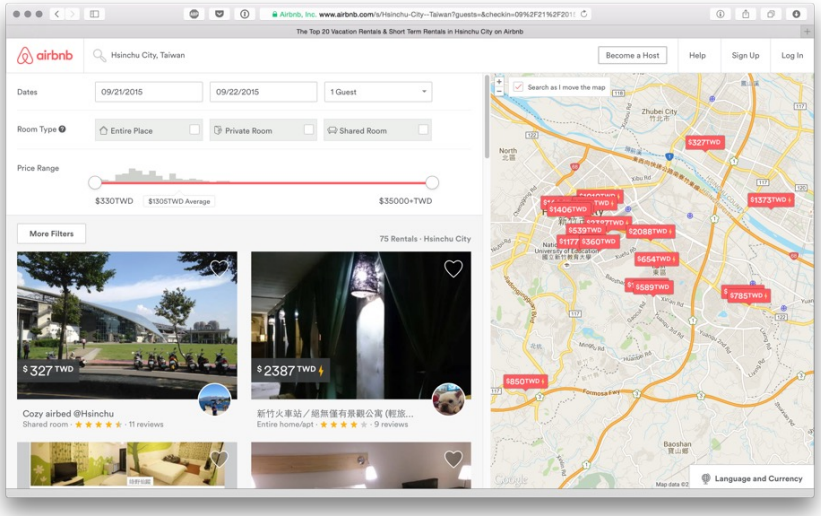
Data Exploration

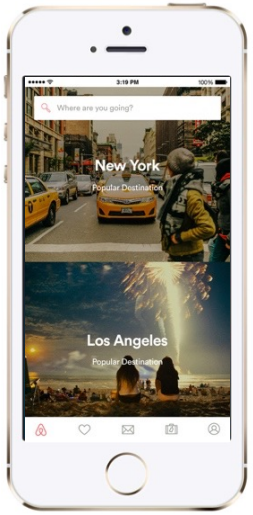


Complex Models



Data Science & Analytics





Job Openings

- **Analytics (Analyst)**
- **Inference**
- **Algorithms**

🤔

Are these all different from each other?

Why does AirBnB make these distinctions?

Business Analysts and Data Scientists

	Analytics	Inference	Algorithms
Goals	Discover insights Give actionable advice	Create models, experiments Inform product development	Develop and deploy machine-learning models
Tools	**GUI / spreadsheet** *(Excel, Tableau, PowerBI, …)*	**Programming** *(Python, R, …)*	**Programming** *(Python, Scala, C++, …)*
Programming	**Basic** *(simple scripts)*	**Intermediate** *(reproducible code)*	**Advanced** *(deployable code)*
Math & Statistics	**Basic - Intermediate** *(bit of everything)*	**Intermediate** *(focus on: causal inference, machine learning, linear algebra)*	**Advanced** *(focus on: AI and deep learning)*
Tasks	Data cleaning, transforming Exploring Data Diagramming	Model data Develop metrics Design experiments	Build machine-learning models Refine and select best models
Audience	Non-technical & Technical	Non-technical & Technical	Non-technical & Technical

Data Scientist

Commonly sought after abilities:

- "Ability to write **clean and concise code**"

  🤔 *Why is coding important? Why is "clean and concise" code important?*

- "Strong **analytical** and **communication skills**"

  🤔 *How do we show "communication skills"?*

- "Attention and commitment to creating **clean reproducible code**"

  🤔 *What makes code "clean"? What makes code "reproducible"?*

Data Scientist - **Inference**

Commonly listed Needs:

- "**Define metrics** to accurately **measure** our progress"
  - 🤔 *What is the difference between metric and measure?*
- "Ensure our **understanding** of product changes is rigorous and accurate"
  - 🤔 *Why is understanding important?*
- "**Find anomalies** in transactions"
  - 🤔 *How do companies find anomalies?*
- "**Evolve our statistical models** of user lifetime value"
  - 🤔 *What are statistical models?*

Data Scientist –**Algorithms**

Commonly listed Needs:

- "Building **machine learning models** to detect high risk activities"

  🤔 *How are machine learning models different?*

- "leverage **active learning** and other human-in-the-loop techniques to improve ML efficiency"

  🤔 *How do machines "learn?*

- "Know best practices, (e.g. **skew** minimization, A/B **testing**, **feature engineering**, **model selection**"

  🤔 *What are features? How do we test?*

Computational Statistics

Traditional Statistical Methods

Central Tendency, Dispersion

Hypothesis Testing

Relationships between Variables

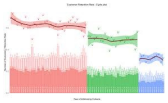
Regression Analysis

Power and Error

Dimensionality and PCA

Path Modeling

New Abilities as Analysts

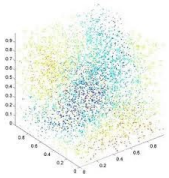


Basic Programming

Iterative Algorithms

Advanced Visualization

New Perspectives on Data



Data Simulation

Data Resampling

Geometric Interpretation

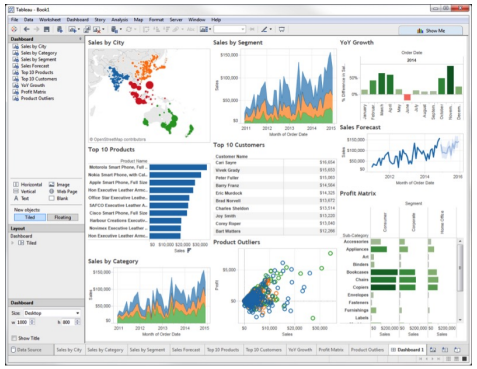
New Tools in Your Toolbelt

R programming

RStudio

Traditional Business Analytics Tools

Graphical User Interface



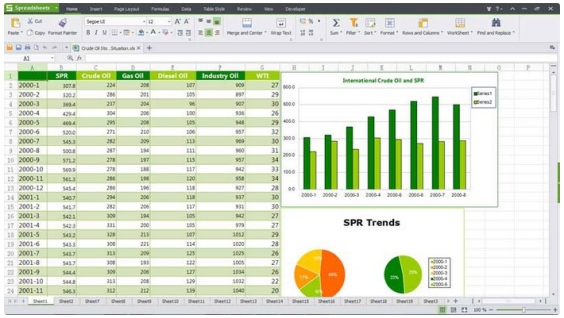
Easy to Learn and Use

Easy to Analyze Data

Easy to Visualize and Communicate

*What are the **downsides** of being limited to GUI and spreadsheet tools?*

Spreadsheets



Familiar Metaphor (Balance Books)

Easy to Manipulate Data

Quick Results

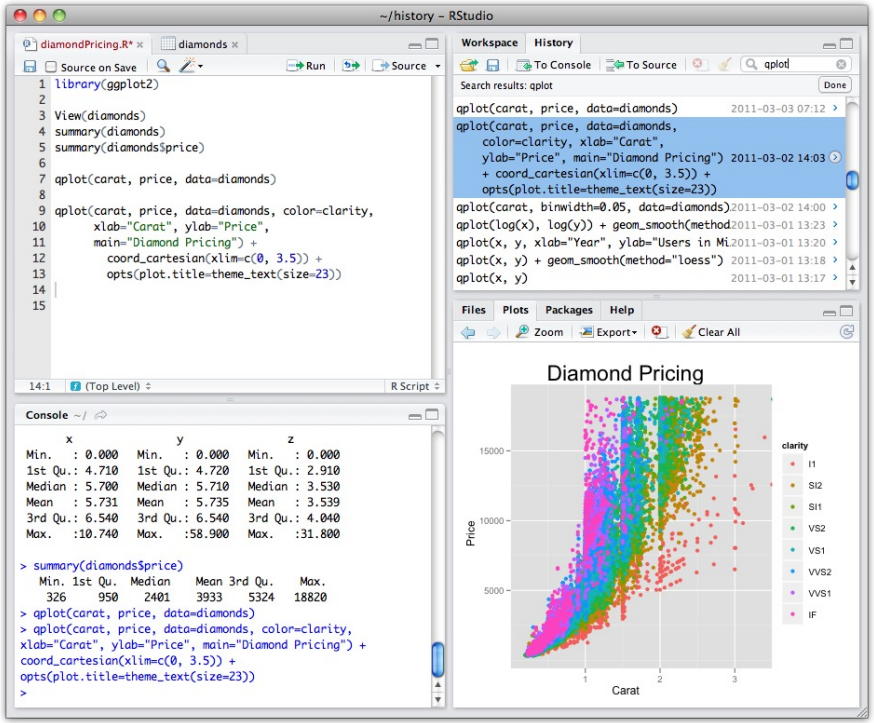
Computational Tools

Analytics Advantage

Run simulations/scenarios
See what happens when assumptions change

Create novel solutions
Make new ways of solving problems

Generate custom visualizations
Allow others to see an understand your solution



Social Analytics Advantage

Sharable
Give your solution to others

Repeatable
Others can run your code with same results

Testable
Ensure your code produces the right results

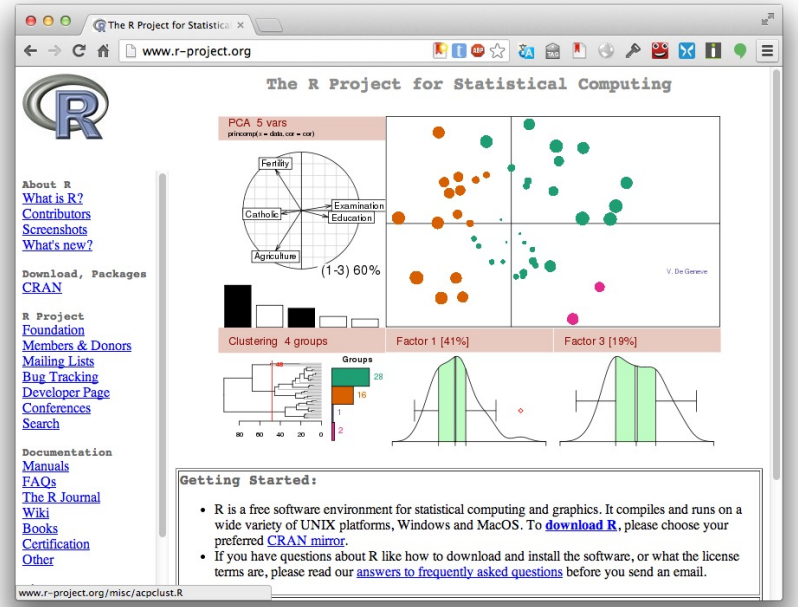
Deployable
Run your code as part of a product platform

*What are the **downsides** of being limited to programmatic tools?*

R and RStudio

Install **R**

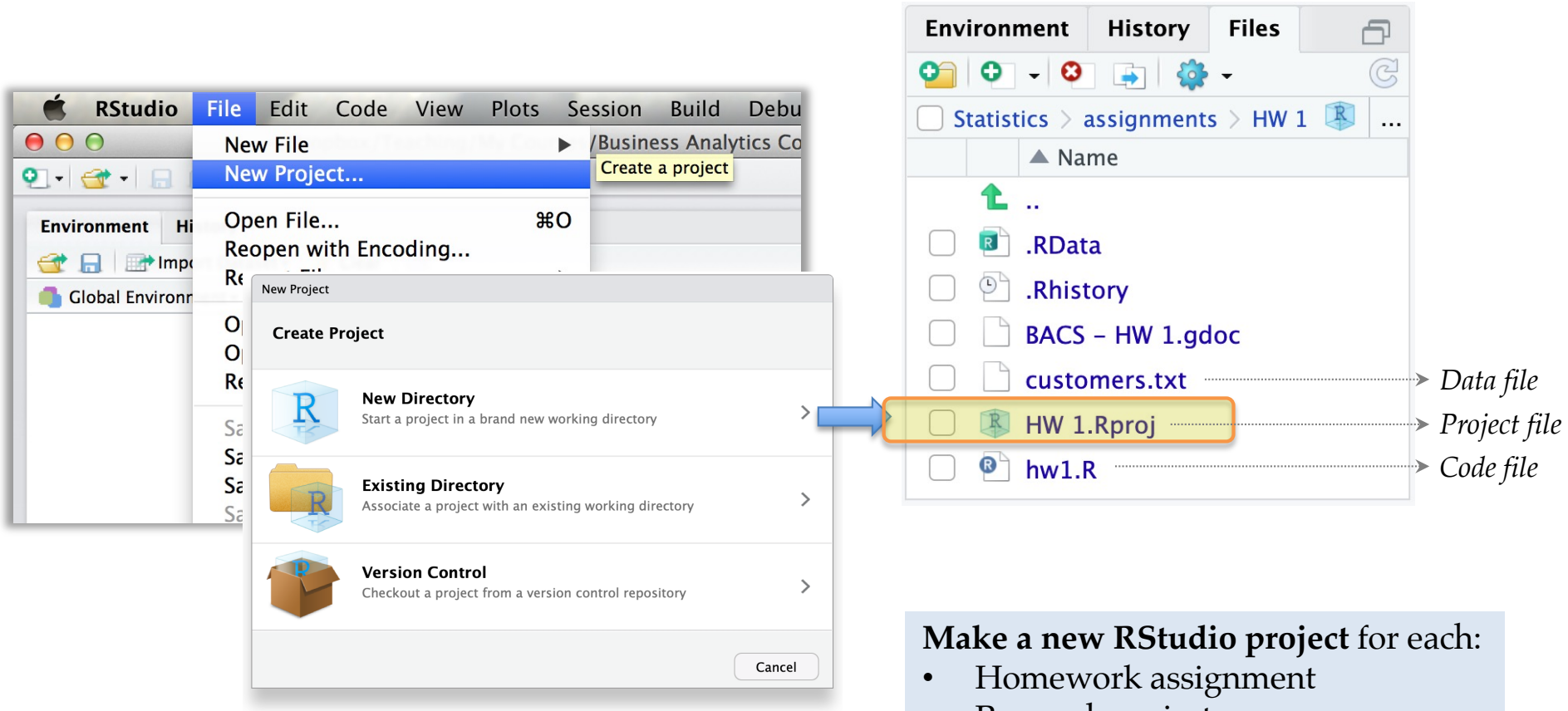
http://www.r-project.org/



Install **RStudio**

http://www.rstudio.com/

RStudio: Using Projects

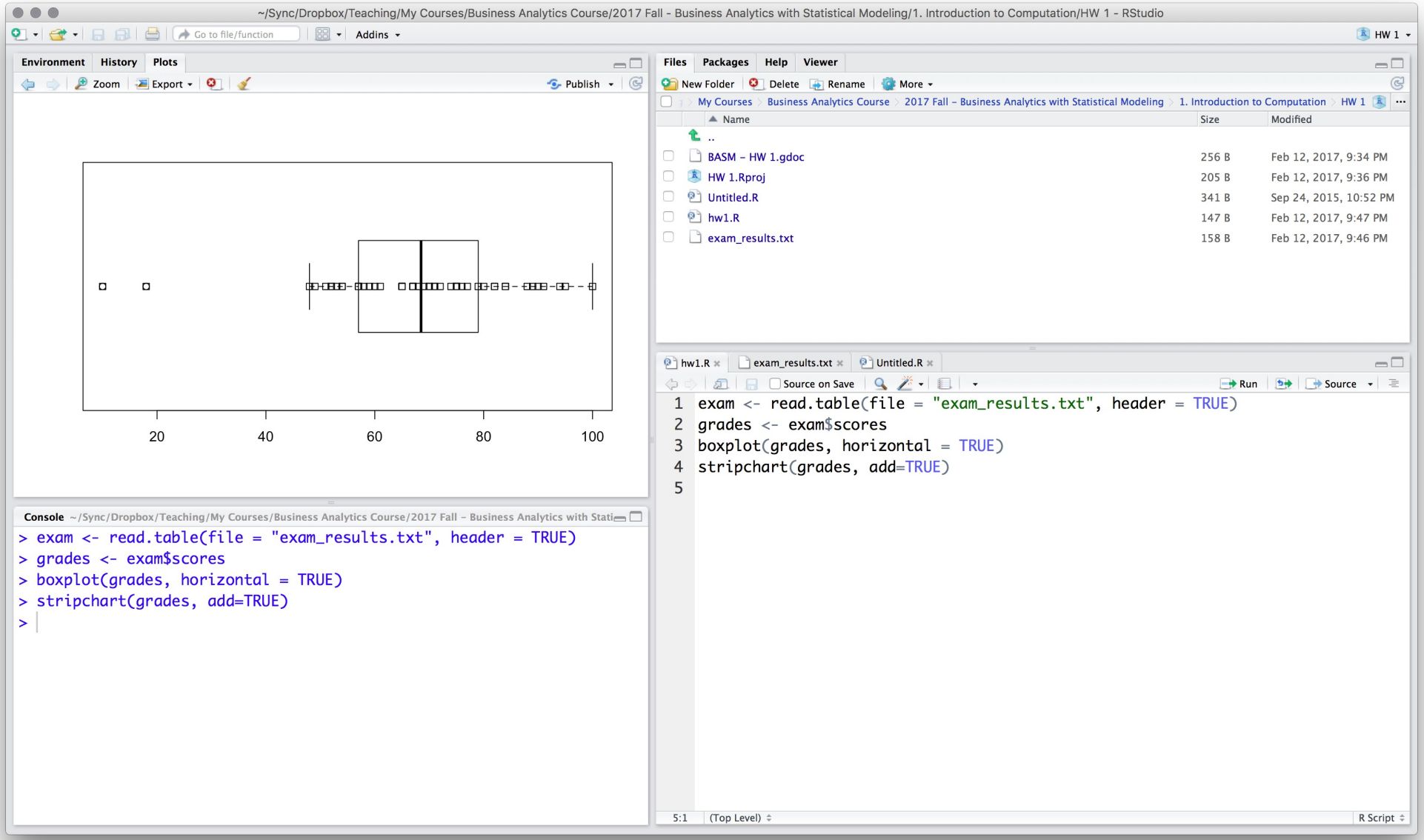


Make a new RStudio project for each:

- Homework assignment
- Research project

Read More: https://support.posit.co/hc/en-us/articles/200526207-Using-RStudio-Projects

RStudio: Workspace



Learning R

At Home

http://swirlstats.com

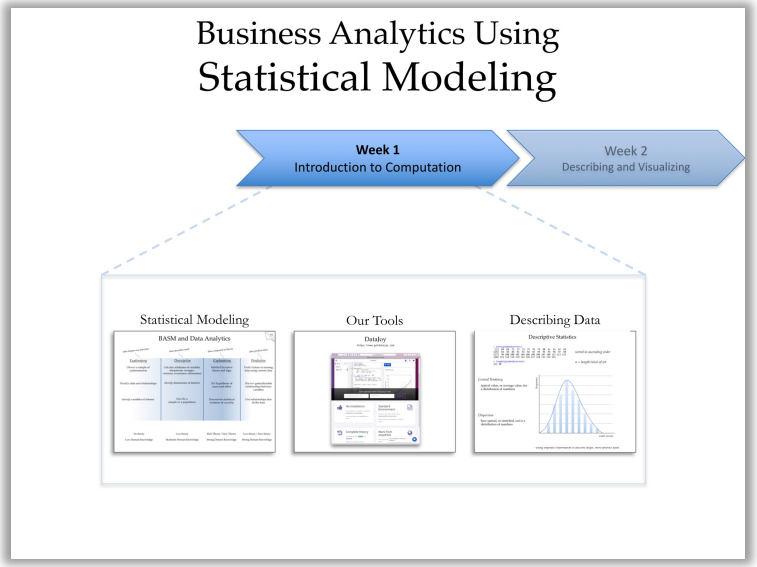
Data structures

Functions

Packages

Loops and Iterations

In Class



Different paradigms in analytic computing

Benchmarking code for performance

Writing re-readable, reusable code

Packaging your code for others

R: Basic Commands

Creating a *vector* of data

```
numbers <- c(12, 14, 14, 25, 33, 35, 38, 38, 41, 43, 45, 50, 58, 59)
numbers[5]
[1] 33

sum(numbers)
[1] 505

seq(3,7)
[1] 3 4 5 6 7
```

Vector: *a sequence of data elements of the same type*

Function: *a sequence of code that can be called to perform an action*

Loading data into a *data frame*

Data Frame: *collection of named vectors*

```
customers <- read.table(file = "customers.txt", header = TRUE)
customers$age
```

`$` *extract named vectors out of a data frame*

```
  [1] 49 69 41 73 45 71 50 43 70 32 47 77 64 50 50 45 49 47 62 50 47 72 47 63 21
 [26] 49 50 48 35 77 48 48 50 47 29 42 42 85 45 49 45 43 49 68 42 48 72 79 48 50
  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
[376] 48 18 45 62 41 71 19 73 26 75 41 46 49 49 23 74 53 23 51 71 50 50 67 74
```

```
ages <- customers$age
length(ages)
 [1] 399
```

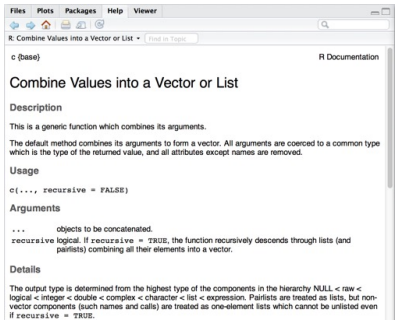
`length()` *: size of a data frame*

Data frame variable

```
customers
     age
1     49
2     69
3     41
4     73
5     45
.    ...

398   67
399   74
```

Help function

```
help(c)
```



1. A reference website for R:
http://cran.r-project.org/doc/manuals/R-intro.html

2. R Introduction
http://www.r-tutor.com/r-introduction

2. What is: c(…)? It makes a vector (combination of values)
http://stat.ethz.ch/R-manual/R-devel/library/base/html/c.html

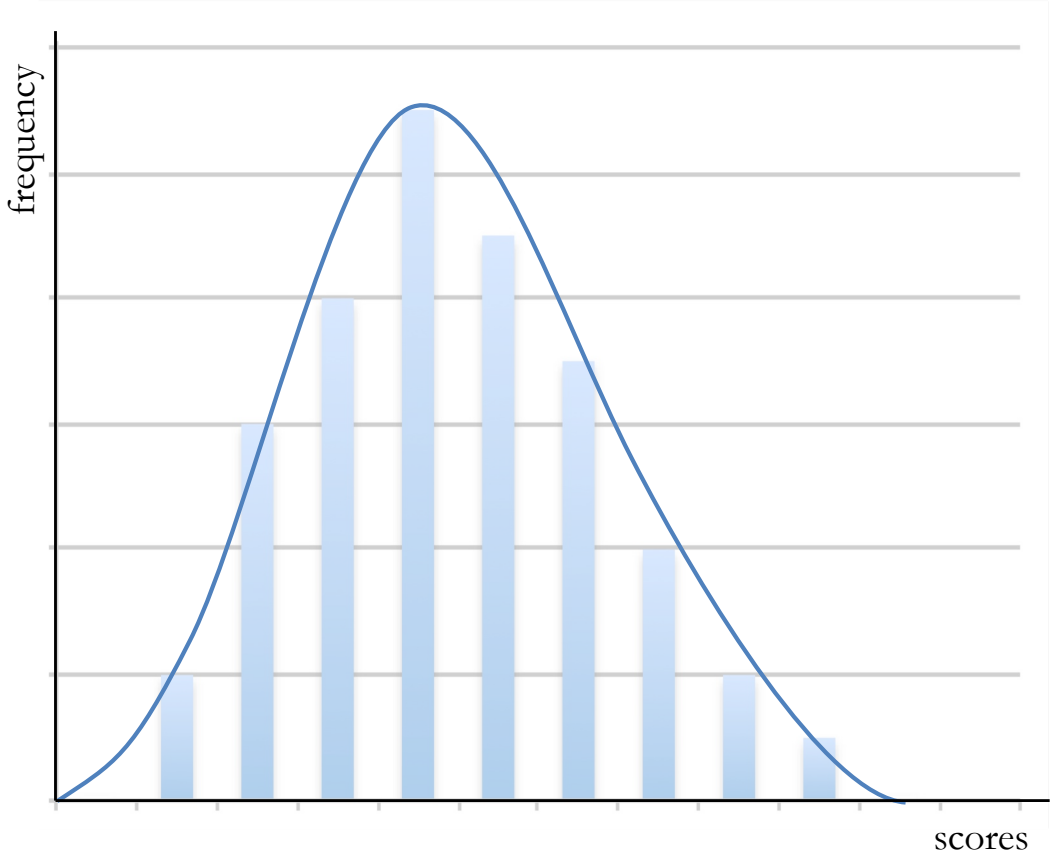
Descriptive Statistics

Central Tendency

typical value, or average value, for a distribution of numbers

Dispersion

how spread, or stretched, out is a distribution of numbers



Central Tendency

```
ages <- customers$age
ages
```

`<-` : *assign value to a variable*

```
  [1] 49 69 41 73 45 71 50 43 70 32 47 77 64 50 50 45 49 47 62 50 47 72 47 63 21
 [26] 49 50 48 35 77 48 48 50 47 29 42 42 85 45 49 45 43 49 68 42 48 72 79 48 50
  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
[376] 48 18 45 62 41 71 19 73 26 75 41 46 49 49 23 74 53 23 51 71 50 50 67 74
```

Population Mean : *the average of your population*

$$\bar{x} = \frac{\sum x_i}{n}$$

```
sum(ages) / length(ages)
[1] 46.80702

mean(ages)
[1] 46.80702
```

***Sample** elements from fixed indeces*

Sampling : *getting a subset of the full population of data*

```
  [1] 49 69 41 73 45 71 50 43 70 32 47 77 64 50 50 45 49 47 62 50 47 72 47 63 21
 [26] 49 50 48 35 77 48 48 50 47 29 42 42 85 45 49 45 43 49 68 42 48 72 79 48 50
  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
[376] 48 18 45 62 41 71 19 73 26 75 41 46 49 49 23 74 53 23 51 71 50 50 67 74
```

```
pick_ages <- ages[c(2, 15, 28, 385)]

[1] 69 50 48 75
```

Pick specific elements with fixed indeces

Random sample : *getting a random subset of the full population of data*

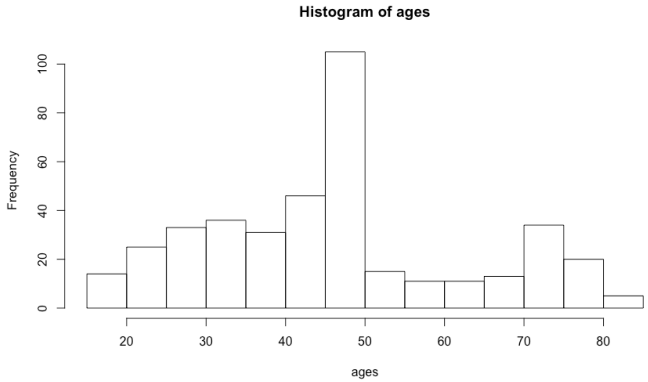
```
random_ages <- sample(ages, 4)
[1] 74 43 47 34

mean(random_ages)
 [1] 49.5
```

***Randomly sample** elements from a list (default: without replacement)*

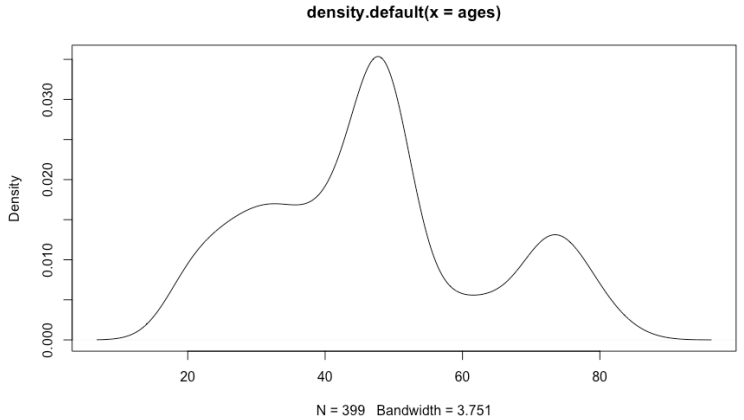
🤔 *Is the mean of my sample close to the population mean?*

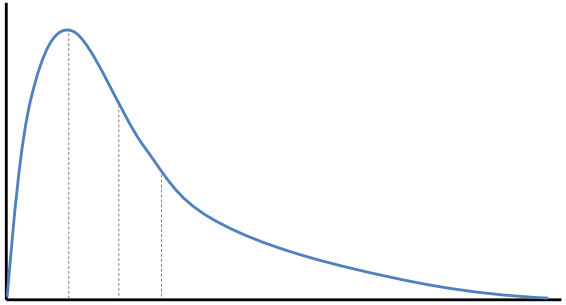
Dispersion: How the Data Deviates From Center

```
hist(ages)
```

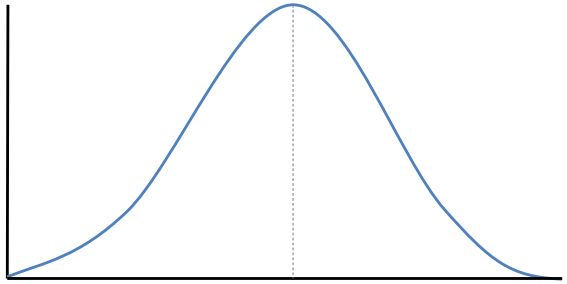



```
plot(density(ages))
```

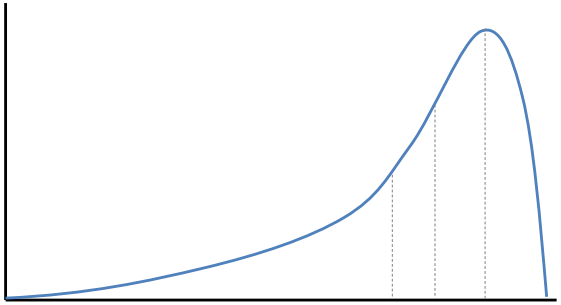




positive skew
right skew



symmetrical



negative skew
left skew

source: http://en.wikipedia.org/wiki/Kernel_density_estimation

```
sorted_ages <- sort(ages)
  [1] 18 19 19 19 19 19 19 19 19 20 20 20 20 20 21 21 21 21 21 21 21 22 22 23 23
 [26] 23 23 23 23 24 24 24 25 25 25 25 25 25 25 26 26 26 26 26 26 26 26 26 27 27
   .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

[376] 76 76 76 76 76 77 77 77 77 78 78 78 78 79 79 79 79 80 80 81 82 82 83 85
```

***sort**ed in ascending order*

Percentile = value at (p/100)n

10th Percentile (p=10):

```
quantile(sorted_ages, 0.10)
[1] 26
```

25th Percentile (p=25):

```
quantile(sorted_ages, 0.25)
[1] 34
```

Quartile = value at i(N+1)/4

1st Quantile (i=1):

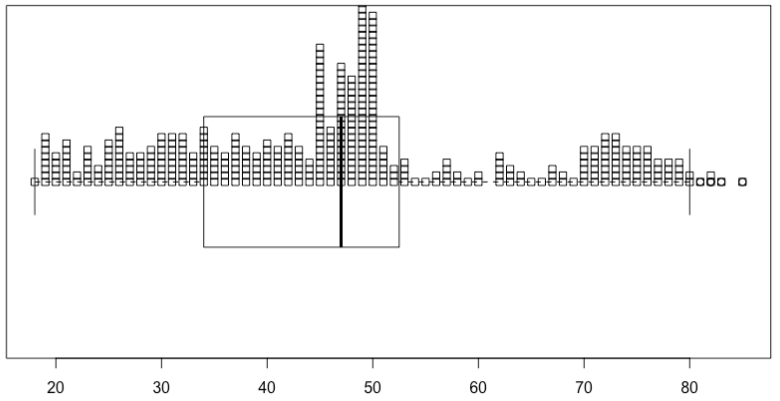
```
quantile(sorted_ages, 0.25)
[1] 26
```

2nd Quantile (i=2):

```
quantile(sorted_ages, 0.50)
[1] 47
```

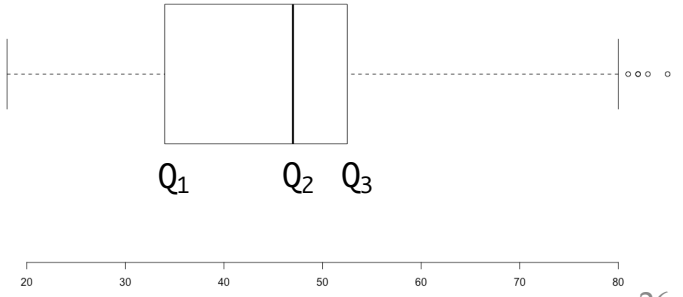
3rd Quantile (i=3):

```
quantile(sorted_ages, 0.75)
[1] 52.5
```




```
boxplot(ages, horizontal = TRUE)
stripchart(ages, method = "stack", add = TRUE)
```

```
summary(ages)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  18.00   34.00   47.00   46.81   52.50   85.00
```



Limits of Computation

Differences in Arithmetic

```
0.1 == 0.1                  TRUE or FALSE?
0.1 + 0.1 == 0.2            TRUE or FALSE?
0.1 + 0.1 + 0.1 == 0.3      TRUE or FALSE?
```

```
(0.1 * 3) - 0.3
[1] 5.551115e-17
```

$-5.551115 \times 10^{-17}$

$-0.00000000000000005551115$

```
0.1 * 4  ==  0.4 ?  TRUE or FALSE
0.1 * 5  ==  0.5 ?  TRUE or FALSE
0.1 * 6  ==  0.6 ?  TRUE or FALSE
0.1 * 7  ==  0.7 ?  TRUE or FALSE
0.1 * 8  ==  0.8 ?  TRUE or FALSE
0.1 * 9  ==  0.9 ?  TRUE or FALSE
0.1 * 10 ==  1.0 ?  TRUE or FALSE
```

When Math Fails You
https://dev.to/jdsteinhauser/when-math-fails-you-2if8

Why 0.1 Does Not Exist In Floating-Point
https://www.exploringbinary.com/why-0-point-1-does-not-exist-in-floating-point/

Finding Zero

```
ages - mean(ages)
 [1]   2.1929825  22.1929825  -5.8070175  26.1929825  -1.8070175  24.1929825   3.1929825
 [8]  -3.8070175  23.1929825 -14.8070175   0.1929825  30.1929825  17.1929825   3.1929825
```

```
mean(ages - mean(ages))
 [1] -1.623275e-15
```

$-1.623275 \times 10^{-15}$

-0.00000000000001623275

It make a difference:
- *which **language** we use*
- *which **operating system** libraries we use*
- *which **hardware (CPU)** we use*
- *and more…*