Introduction to STATS 406

Mark M. Fredrickson (`mfredric@umich.edu`)

2021-05-10

Computational Methods in Statistics and Data Science (STATS4060J, Summer 2021)

Course Overview

Course Goals

At the end of this course you will be able to

- Trade **computation time** for **analytical effort**

Course Goals

At the end of this course you will be able to

- Trade **computation time** for **analytical effort**
- Connect **mathematical theory** to **computational applications**

Course Goals

At the end of this course you will be able to

- Trade **computation time** for **analytical effort**
- Connect **mathematical theory** to **computational applications**
- Understand and explain **costs, benefits, and limits** of computational techniques

Course Goals

At the end of this course you will be able to

- Trade **computation time** for **analytical effort**
- Connect **mathematical theory** to **computational applications**
- Understand and explain **costs, benefits, and limits** of computational techniques
- Implement key **statistical algorithms** in R

Course Goals

At the end of this course you will be able to

- Trade **computation time** for **analytical effort**
- Connect **mathematical theory** to **computational applications**
- Understand and explain **costs, benefits, and limits** of computational techniques
- Implement key **statistical algorithms** in R
- Generate **useful visualizations** of your data

Course Goals

At the end of this course you will be able to

- Trade **computation time** for **analytical effort**
- Connect **mathematical theory** to **computational applications**
- Understand and explain **costs, benefits, and limits** of computational techniques
- Implement key **statistical algorithms** in R
- Generate **useful visualizations** of your data
- Create your own **statistical models**

Course Goals

At the end of this course you will be able to

- Trade **computation time** for **analytical effort**
- Connect **mathematical theory** to **computational applications**
- Understand and explain **costs, benefits, and limits** of computational techniques
- Implement key **statistical algorithms** in R
- Generate **useful visualizations** of your data
- Create your own **statistical models**
- **Analyze data** using computational techniques

Course Goals

At the end of this course you will be able to

- Trade **computation time** for **analytical effort**
- Connect **mathematical theory** to **computational applications**
- Understand and explain **costs, benefits, and limits** of computational techniques
- Implement key **statistical algorithms** in R
- Generate **useful visualizations** of your data
- Create your own **statistical models**
- **Analyze data** using computational techniques
- Frame **research questions** and find **relevant data**

What is Computational Statistics?

Statistics: the application of **probability theory** to real **data**.

What is Computational Statistics?

Statistics: the application of **probability theory** to real **data**.

Goals:

- Specifying **models of data generation**

What is Computational Statistics?

Statistics: the application of **probability theory** to real **data**.

Goals:

- Specifying **models of data generation**
- Performing **inference** to connect real data to model

What is Computational Statistics?

Statistics: the application of **probability theory** to real **data**.

Goals:

- Specifying **models of data generation**
- Performing **inference** to connect real data to model
  - **Estimate** model parameters

What is Computational Statistics?

Statistics: the application of **probability theory** to real **data**.

Goals:

- Specifying **models of data generation**
- Performing **inference** to connect real data to model
    - **Estimate** model parameters
    - **Test** hypotheses

What is Computational Statistics?

Statistics: the application of **probability theory** to real **data**.

Goals:

- Specifying **models of data generation**
- Performing **inference** to connect real data to model
  - **Estimate** model parameters
  - **Test** hypotheses
- Quantify **uncertainty** in the inference process.

What is Computational Statistics?

Statistics: the application of **probability theory** to real **data**.

Goals:

- Specifying **models of data generation**
- Performing **inference** to connect real data to model
  - **Estimate** model parameters
  - **Test** hypotheses
- Quantify **uncertainty** in the inference process.
- Understand **operating characteristics** of tools.

What is Computational Statistics?

Statistics: the application of **probability theory** to real **data**.

Goals:

- Specifying **models of data generation**
- Performing **inference** to connect real data to model
  - **Estimate** model parameters
  - **Test** hypotheses
- Quantify **uncertainty** in the inference process.
- Understand **operating characteristics** of tools.

What is Computational Statistics?

Statistics: the application of **probability theory** to real **data**.

Goals:

- Specifying **models of data generation**
- Performing **inference** to connect real data to model
  - **Estimate** model parameters
  - **Test** hypotheses
- Quantify **uncertainty** in the inference process.
- Understand **operating characteristics** of tools.

Computational statistics uses **substantial amounts of computation** to achieve these goals.

Distributions

We often model data using the **Normal distribution** because it is well understood and we have **mathematically analyzed** many of its properties.

Distributions

We often model data using the **Normal distribution** because it is well understood and we have **mathematically analyzed** many of its properties.

$$X_1, X_2, \ldots, X_n \sim N(0,1) \quad \text{(independent)}$$

Distributions

We often model data using the **Normal distribution** because it is well understood and we have **mathematically analyzed** many of its properties.

$$X_1, X_2, \ldots, X_n \sim N(0,1) \quad \text{(independent)}$$
$$\frac{1}{n}\sum_{i=1}^{n} X_i \sim N\left(0, \frac{1}{n}\right)$$

Distributions

We often model data using the **Normal distribution** because it is well understood and we have **mathematically analyzed** many of its properties.

$$X_1, X_2, \ldots, X_n \sim N(0, 1) \quad \text{(independent)}$$

$$\frac{1}{n} \sum_{i=1}^{n} X_i \sim N\left(0, \frac{1}{n}\right)$$

For example,

$$X_1, X_2, X_3 \overset{\text{iid}}{\sim} N(0, 1)$$

then

$$\frac{1}{3} \sum_{i=1}^{3} X_i \sim N(0, 1/3)$$

Distribution of the Sample Maximum Magnitude

What is the distribution of the **sample maximum magnitude**?

$$\max_{i=1,2,3} |X_i|$$

Distribution of the Sample Maximum Magnitude

What is the distribution of the **sample maximum magnitude**?

$$\max_{i=1,2,3} |X_i|$$

Generate many replicates of (X_1, X_2, X_3) and find the **empirical distribution**.

Distribution of the Sample Maximum Magnitude

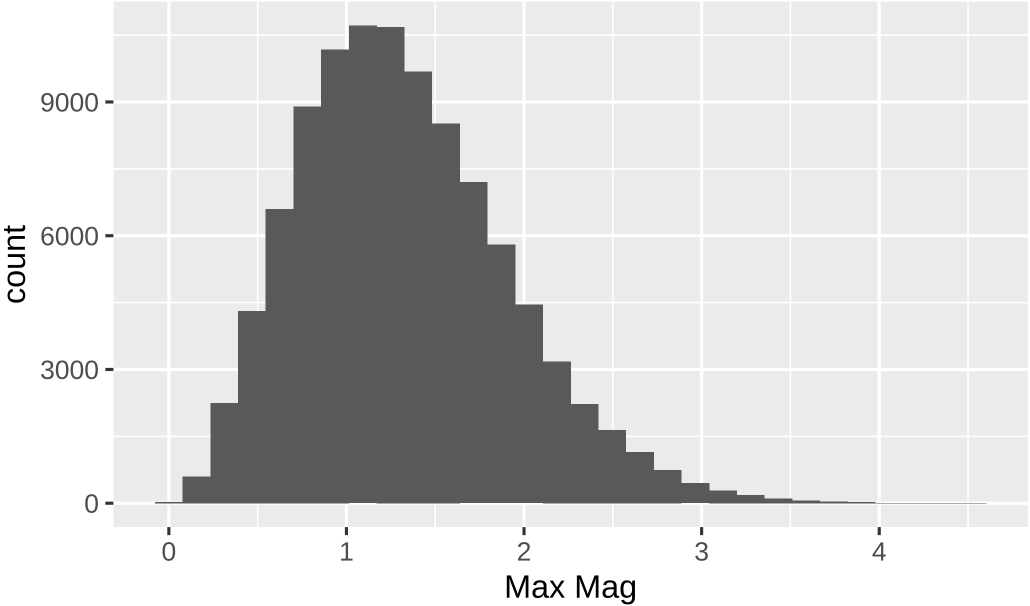
What is the distribution of the **sample maximum magnitude**?

$$\max_{i=1,2,3} |X_i|$$

Generate many replicates of (X_1, X_2, X_3) and find the **empirical distribution**.

In R, the **random number generator** for standard Normal variables is `rnorm`:

```
> sample_max_mag <- function(n) {
+     x <- rnorm(n)
+     max(abs(x))
+ }
> empirical_distrib <- replicate(100000, sample_max_mag(3))
```



9000
6000
3000
0
count
0
1
2
3
4
Max Mag

Some Properties

Mean and **variance**:

```
> mean(empirical_distrib); var(empirical_distrib)

[1] 1.327

[1] 0.3454
```

Some Properties

Mean and **variance**:

```
> mean(empirical_distrib); var(empirical_distrib)
[1] 1.327
[1] 0.3454
```

Median and other **quantiles**:

```
> quantile(empirical_distrib, c(0.25, 0.5, 0.75))
   25%    50%    75%
0.8957 1.2646 1.6911
```

Some Properties

Mean and **variance**:

```
> mean(empirical_distrib); var(empirical_distrib)
[1] 1.327
[1] 0.3454
```

Median and other **quantiles**:

```
> quantile(empirical_distrib, c(0.25, 0.5, 0.75))
   25%    50%    75%
0.8957 1.2646 1.6911
```

Note: these are **estimates**, which we'll discuss more later.

In the previous example, the function

$$f(X_1, X_2, X_3) = \max(|X_1|, |X_2|, |X_3|)$$

is an example of a **statistic**, a function of random data.

Distributions of Statistics

In the previous example, the function

$$f(X_1, X_2, X_3) = \max(|X_1|, |X_2|, |X_3|)$$

is an example of a **statistic**, a function of random data.

We use statistics (functions of random data) to

- Estimate population parameters
- Test hypotheses about populations
- Perform prediction for new observations

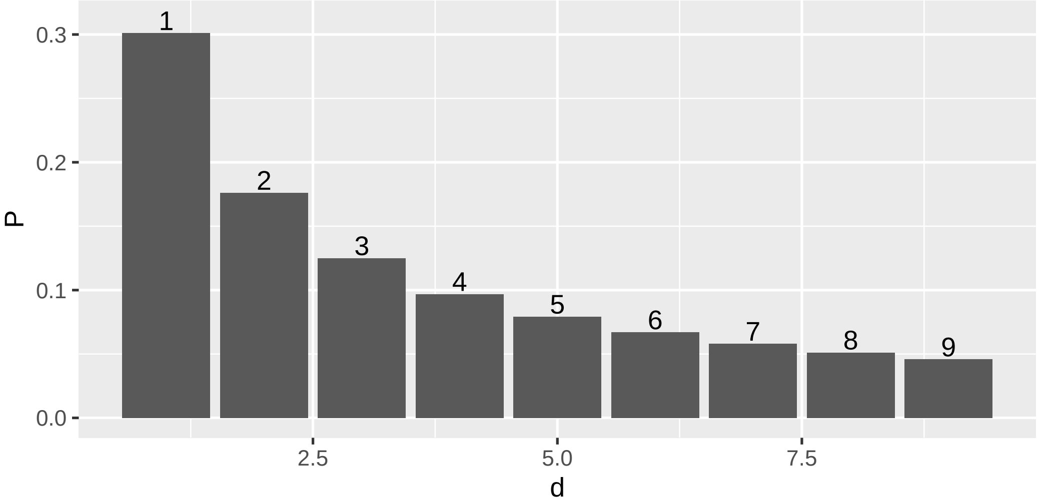
Example: Test statistics for Benford's Law

Benford's Law holds that the distribution of **leading digits** in a collection of numbers spanning several orders of magnitudes will follow the following distribution:

$$\Pr(D = d) = \log_{10}\left(\frac{d+1}{d}\right), \quad d = 1, \ldots, 9$$

```
> dbenford <- function(x) {
+     ifelse(x >= 1 & x <= 9, log((x + 1)/ x, base = 10), 0)
+ }
```

$\Pr(D = d)$ under Benford's Law



Using random *D*s

Tam Cho and Gaines (2007) investigated **political contributions between political committees** as reported by the FEC. Here are the digit frequencies for 8,396 contributions in 2004 (Table 1):

```
> pol_digits <- c(23.3, 21.1, 8.5, 11.7, 9.5, 4.2, 3.7, 4.0, 14.1) / 100
```

Tam Cho and Gaines (2007) investigated **political contributions between political committees** as reported by the FEC. Here are the digit frequencies for 8,396 contributions in 2004 (Table 1):

```
> pol_digits <- c(23.3, 21.1, 8.5, 11.7, 9.5, 4.2, 3.7, 4.0, 14.1) / 100
```

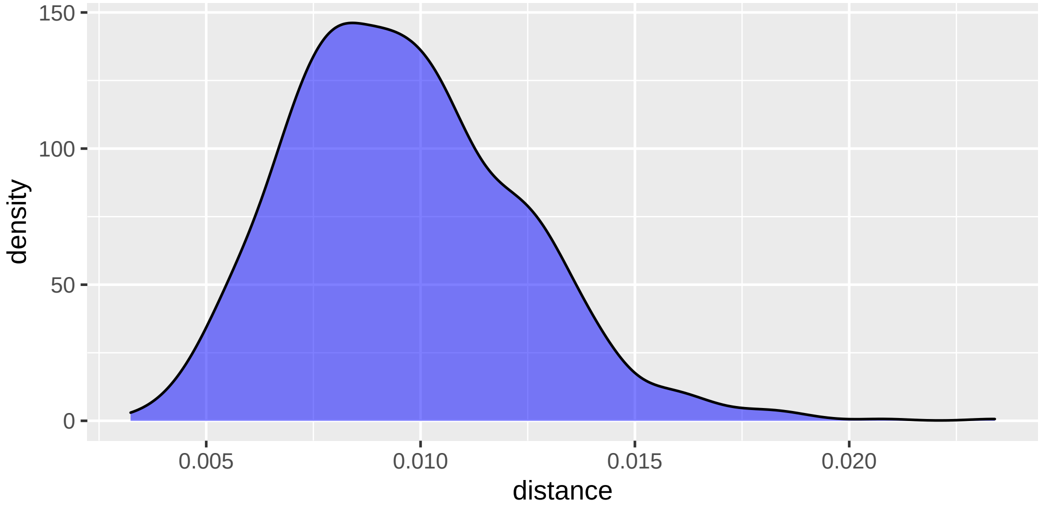
A typical way to analyze these data would be to use a χ^2 test comparing the **expected** to the **observed counts**. Alternatively, Tam Cho and Gaines suggest the statistic:

```
> distance <- function(v)  { sqrt(sum((v - dbenford(1:9))^2)) }
```

Distribution of the Test Statistic

```
> rbenford <- function(n) {
+   sample(1:9, size = n, prob = dbenford(1:9), replace = TRUE)
+ }
> n <- 8396
> compute_test_statistic <- function(ds) {
+     probs <- map_dbl(0:9, ~ mean(ds == .x))
+     distance(probs)
+ }
> null_distances <- replicate(1000,
+                              compute_test_statistic(rbenford(n)))
```

Null Distribution



p-value for the hypothesis test

```
> (p_value <- mean(null_distances >= observed_dist)) # P(T > t)
[1] 0
```

The observed test statistic was larger than any sample we generated (so the p-value was zero) and is 47 standard deviations from the mean of the null distribution.

p-value for the hypothesis test

```
> (p_value <- mean(null_distances >= observed_dist)) # P(T > t)
[1] 0
```

The observed test statistic was larger than any sample we generated (so the *p*-value was zero) and is 47 standard deviations from the mean of the null distribution.

With *extremely high confidence*, we can reject the null hypothesis that these data were a sample from a population that follows Benford's Law.

Joint Relationships

The Benford's Law example we

- only considered a **single variable** (leading digit)
- **assumed** Benford's digit distribution for data

Joint Relationships

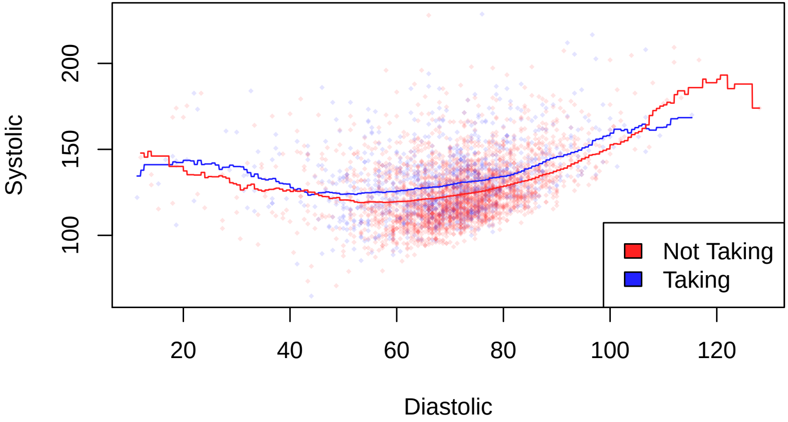
The Benford's Law example we

- only considered a **single variable** (leading digit)
- **assumed** Benford's digit distribution for data

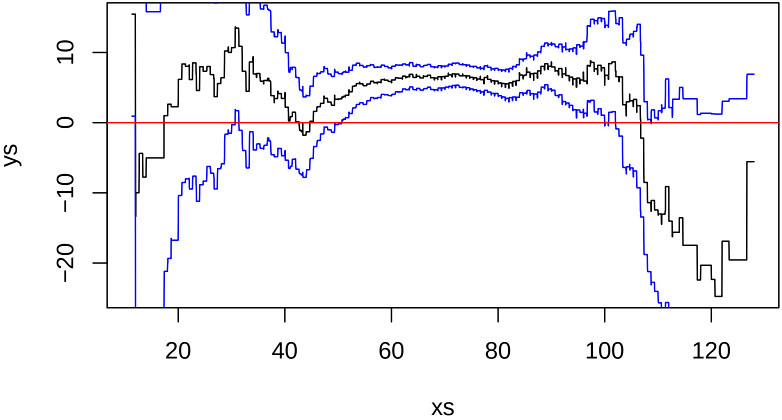
More often than not, we care about **the joint distribution** of two or more variables.

- How can we relate variables to each other?
- How can we make uncertainty statements about our estimated relationships?

Smoothed Mean Function Estimation



Bootstrapped difference of smoothed mean estimators



Simulated Gene Expression Data

	individual	group	value1	value2	value3
1	RZTYXH	A	43.223	88.63	29.782
2	JVXDCH	A	9.352	51.47	44.470
11	JOGSAH	B	115.369	28.35	27.778
12	ZLHDVP	B	113.624	45.37	5.159
55	RKWUXM	D	9.900	24.53	121.841
56	GNJYQC	D	46.668	31.88	131.449

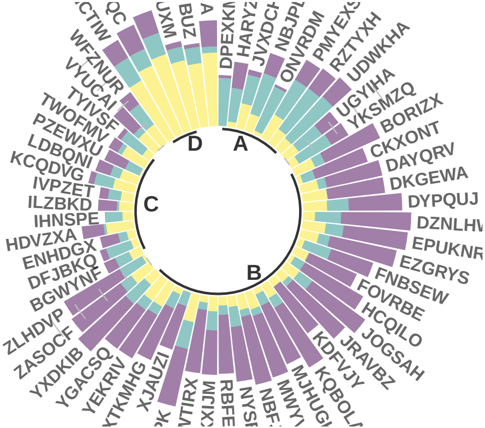


Figure 1: R Graph Gallery

Visualizations II

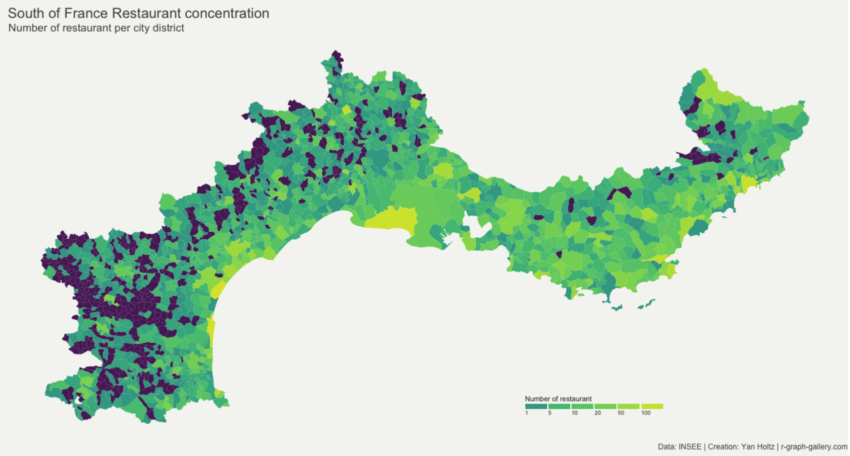


Figure 2: R Graph Gallery

Visualizations III

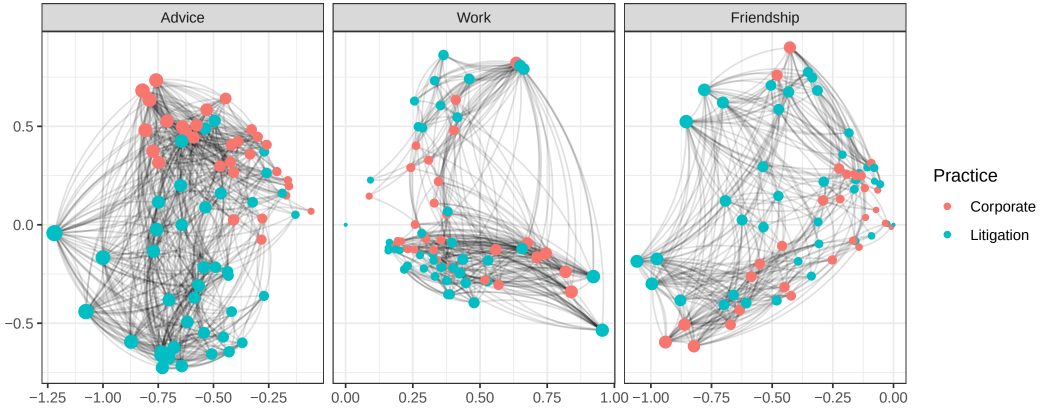


Figure 3: Fredrickson and Levin

Course Logistics

People and Resources

Professor	Mark Fredrickson, `mfredric@umich.edu`
Office Hours	Tuesday 9am – 10am (UTC+8), Thursday 8pm – 9pm (UTC+8) by appointment.
TAs	Jiaming Kang (`Jiaming.Kang@sjtu.edu.cn`), second TA
TA OH	TBA
Assignments	Distributed and turned in through Canvas.
Online	Canvas and Piazza

Course Pre-requisites

- Some programming experience (R, Python, C/C++, Java)
- An understanding of the following terms: Distributions and random variables, random sampling and sampling distributions, population parameters and estimation, hypothesis tests, basic calculus (integrals, derivatives).

Course Materials

No required text, but several **recommended** (particularly first 2):

- Rizzo, Maria L. *Statistical Computing with R*
- Wickham, H. & Grolemund, G. *R for Data Science*
- Robert, C. & Casella, G. *Introducing Monte Carlo Methods with R*
- Agresti, A. *Foundations of Linear and Generalized Linear Models*
- Gentle, J. E. *Computational Statistics*
- *Handbook of Computational Statistics*. Härdle, Gentle, and Mori, eds.

All slides will be posted to **Canvas**.

Grading

Your grade will be made up of 200 points:

- **90** 10 weekly assignments due at 10:00pm on Sundays via Canvas. **No late submissions.** 10 points each. Lowest assignment dropped.
- **20** 5 quizzes (administered in Friday sessions), 5 points each, lowest dropped.
- **90** Final Project (approximately 10 to 15 pages)
  - **10** First Draft, Due July 18
  - **80** Final Draft, Due August 4
- **+6** Extra credit for watching and summarizing computational statistics, data science, or applied research talk (up to 3 times).

Homework

Distributed by **9am Monday**, due the following **Sunday at 10:00pm**.

Homework

Distributed by **9am Monday**, due the following **Sunday at 10:00pm**.

First homework currently available on Canvas, due 2021-05-23 at 10pm UTC+8.

Homework

Distributed by **9am Monday**, due the following **Sunday at 10:00pm**.

First homework currently available on Canvas, due 2021-05-23 at 10pm UTC+8.

No late submissions will be accepted, but **lowest homework will be dropped**.

Final Project

To showcase your knowledge and skills, in **groups of 3**, you will prepare a **final project** (approximately 10 to 15 pages) in which you

- Propose a research question (three example projects available)
- Select and describe a data set
- Analyze it using multiple techniques from class
- Interpret your results for a broader audience

Final Project

To showcase your knowledge and skills, in **groups of 3**, you will prepare a **final project** (approximately 10 to 15 pages) in which you

- Propose a research question (three example projects available)
- Select and describe a data set
- Analyze it using multiple techniques from class
- Interpret your results for a broader audience

Some topics from past semesters:

- Analyzing drug overdose by types of drug and age group
- Comparing the emotional content of art over time
- Comparing hurricane strength on the Atlantic and Gulf coasts
- Building option pricing models

Extra Credit Talks

You **may** view up to **three** research seminars to earn extra credit (1%/2 pts each).

- Lectures **must be approved** in advance.
- There is a list of **approved videos** on Canvas.
- You may email me other lectures for approval.

Extra Credit Talks

You **may** view up to **three** research seminars to earn extra credit (1%/2 pts each).

- Lectures **must be approved** in advance.
- There is a list of **approved videos** on Canvas.
- You may email me other lectures for approval.

Academic Misconduct

At all times, ask yourself the question, “Am I avoiding learning something by my choices?”

Academic Misconduct

At all times, ask yourself the question, “Am I avoiding learning something by my choices?”

You may **freely discuss general approaches** to all work. Each student must **write up/implement solutions individually**. Use of code/packages (not shown in class) is generally discouraged. Feel free to ask.

Inclusivity, Sexual Misconduct, and Students with Disabilities

This classroom strives to be a welcoming space for **all students** of all ages, backgrounds, beliefs, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, ability, and other visible and non-visible differences.

Inclusivity, Sexual Misconduct, and Students with Disabilities

This classroom strives to be a welcoming space for **all students** of all ages, backgrounds, beliefs, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, ability, and other visible and non-visible differences.

We have a **zero tolerance policy** for disrespect, violence, and sexual misconduct. Please see the syllabus for additional details.

Inclusivity, Sexual Misconduct, and Students with Disabilities

This classroom strives to be a welcoming space for **all students** of all ages, backgrounds, beliefs, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, ability, and other visible and non-visible differences.

We have a **zero tolerance policy** for disrespect, violence, and sexual misconduct. Please see the syllabus for additional details.

Students requiring accommodations for a disability should **speak with me as early as possible**.

Tour of Canvas

R, RStudio, RMarkdown

What is R?

When we say, “R” we are referring to three interrelated things:

- A language
- A community
- An implementation or environment

R: The language

R is a **statistical programming language**:

R: The language

R is a **statistical programming language**:

- R is specifically design to load, manipulate, and analyze tabular data (versus Python, Java, C++)

R: The language

R is a **statistical programming language**:

- R is specifically design to load, manipulate, and analyze tabular data (versus Python, Java, C++)
- We can use R to easily code up new algorithms, methods (versus Stata, SAS)

R: The language

R is a **statistical programming language**:

- R is specifically design to load, manipulate, and analyze tabular data (versus Python, Java, C++)
- We can use R to easily code up new algorithms, methods (versus Stata, SAS)
- We interact with R via scripts containing textual input (versus Minitab, Excel)

R: The language

R is a **statistical programming language**:

- R is specifically design to load, manipulate, and analyze tabular data (versus Python, Java, C++)
- We can use R to easily code up new algorithms, methods (versus Stata, SAS)
- We interact with R via scripts containing textual input (versus Minitab, Excel)

Key concepts:

- Store data in variables, usually **vectors**, **matrices**, and **data frames**.
- Manipulate data using **functions**, **iteration**, and **high level declarations**.
- Process data using **scripts** and **RMarkdown documents**.

R: The community

The **Comprehensive R Archive Network** is a collection of **user submitted packages**. As of this writing, there **17,548** packages available.

R: The community

The **Comprehensive R Archive Network** is a collection of **user submitted packages**. As of this writing, there **17,548** packages available.

R is supported via: textbooks, official mailing lists, StackOverflow, R Bloggers, YoutTube, etc (though it can be hard to Google for sometimes).

R: The community

The **Comprehensive R Archive Network** is a collection of **user submitted packages**. As of this writing, there **17,548** packages available.

R is supported via: textbooks, official mailing lists, StackOverflow, R Bloggers, YoutTube, etc (though it can be hard to Google for sometimes).

R is being adopted by Fortune 500 companies, government, start ups, applied academic disciplines, many others.

R: The environment

The official R implementation consists of an **command line interface** for entering R commands, a **batch file processor** for handling scripts, and a basic **graphical user interface** for handling plots.

R: The environment

The official R implementation consists of an **command line interface** for entering R commands, a **batch file processor** for handling scripts, and a basic **graphical user interface** for handling plots.

We will be using **RStudio** which adds:

- Projects to handle multiple R files, data files.
- More complete file editor with syntax completion
- Help system and graph tab
- Integration with external software development tools
- RMarkdown to PDF support
- Desktop and server instances

RMarkdown is a **plain text file** that contains **structured text** and **R snippets**. It can be **processed** into a PDF or HTML file. The R is **evaluated** and the **results added to the file**, including plots.

RMarkdown

RMarkdown is a **plain text file** that contains **structured text** and **R snippets**. It can be **processed** into a PDF or HTML file. The R is **evaluated** and the **results added to the file**, including plots.

Some great features:

- Put the description and the implementation in one place.
- Inline R code allows printing out values – no more copy and paste errors.
- Easy to supply starter code for home works.
- Includes a math language for writing up analytical results.

RStudio and RMarkdown

Final notes

Before next class:

- Install R and RStudio
- Download HW1 and confirm that you can "knit" it.
- Sign up for course Piazza.

Next topic: Statistical Review