# STAT242 Introduction to Data Science with R

(Formerly STAT490 Introduction to Statistical Computing with R)

Dan Yorgov

## 1. Introduction

What is Data Science? No universally agreed upon definition.

Attempt: *Data Science is the study of extracting meaningful insights from data using programming skills, mathematical and statistical knowledge, and domain expertise.*

*"... All sciences are, in the abstract, mathematics.*
*All judgements are, in their rationale, statistics."*
*C. R. Rao (1920-2023)*

In 18th century, Gauss came up with the least squares method for linear regression. Logistic regression is an extension of it.

*"When you're fundraising, it's AI. When you're hiring, it's Machine Learning. When you're implementing, it's logistic regression." Reddit User, 2024*

In the past 4-5 years, the demand for Data Science (DS) professionals has evolved.

- Previously, a biology major with a couple of DS bootcamps would secure a "Data Scientist" job. Now, they might be able to find a "Data Analyst" role.
- Currently, around 90% of applicants in specialist DS companies have degrees in DS, Statistics, Mathematics, or Computer Science.

- "Technical Skills" are a must but with AI propagation, perhaps not the major emphasis anymore.
  - You don't need to be a great programmer (but it helps).
    - You do need to be able to read and understand code and do coding alone or with some help from AI.
    - Great programmers are surely still in high demand.
- Your Math and Statistics skills are your major plus now.
  - Your understanding of major statistical concepts like overfitting, bias-variance tradeoff, statistical significance, etc. becomes crucial.
  - Strong statistical knowledge for choosing appropriate models and interpreting results accurately.
- Critical thinking and problem-solving skills are essential.

## Traditional Statistics:

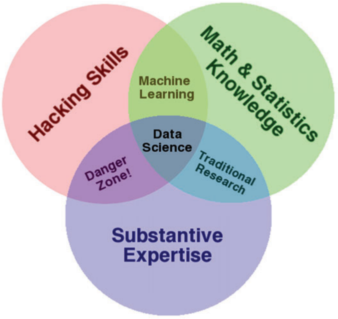
- What specific question are we trying to answer?
- Can this question be measured quantitatively?
- How can we collect the data appropriately?
- What statistical methods should we use to analyze it?
- How can we use the analysis to make decisions, draw conclusions, or make predictions?

## Data Science:

- What data sources can we access or extract?
- What machine learning models or algorithms are suitable for this data? What patterns, trends, or insights can we extract from the data?
- How can we use these models to improve decision-making or automate processes? Do these insights apply to future datasets or scenarios?
- How do we ensure the reliability and ethical use of our data-driven insights?

## Data Science:

- For the most part, old theoretical groundwork.
- Statisticians and computer scientists developing software for machine learning and stochastic modeling.



“I am a data scientist,” you said.

Before:

“I am a data scientist,” you said.

Before:

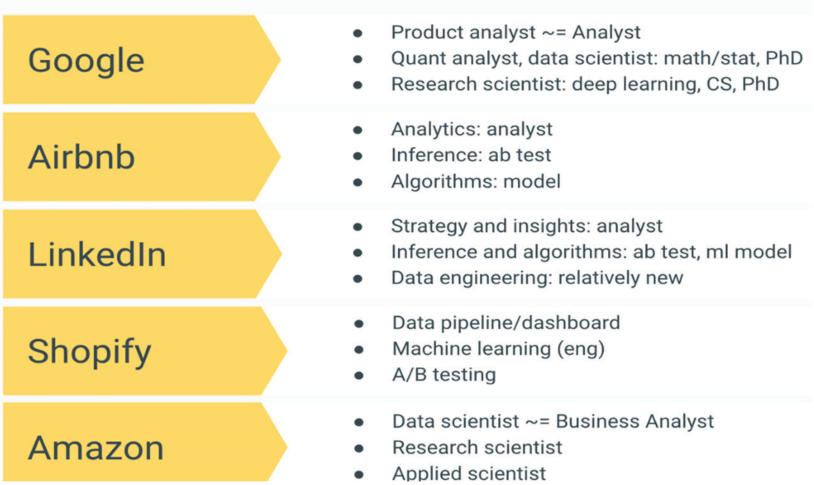
Now:

## Titles in/Related to Data Science

Many Midwest companies aim to build data science teams in the next years. Interesting jobs and well paid too.

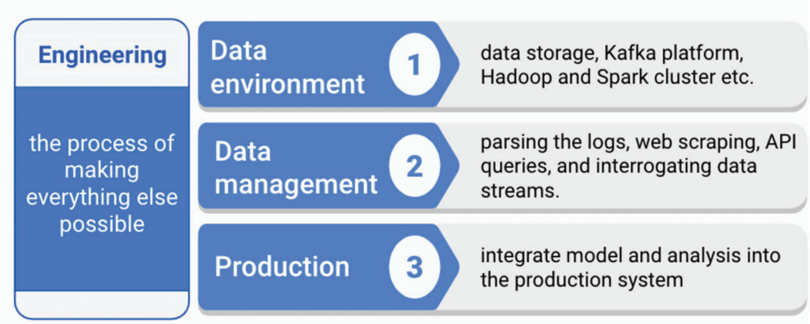
Data Scientist
Data Analyst
Machine Learning Engineer
Data Engineer
Data Infrastructure Engineer
Analytics Engineer
Business Intelligence (BI) Engineer
Business Analyst / Economist
Research Scientist / Applied Scientist
Quant Researcher (UX, Market, Finance) or Analyst

No unification of job titles and expectations yet.

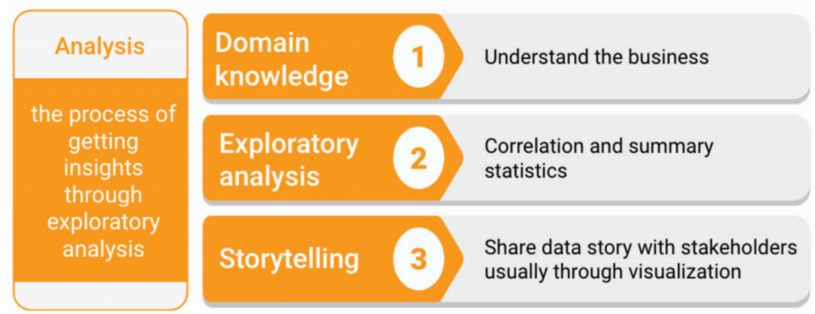


| | |
|---|---|
| Google | • Product analyst ~= Analyst<br>• Quant analyst, data scientist: math/stat, PhD<br>• Research scientist: deep learning, CS, PhD |
| Airbnb | • Analytics: analyst<br>• Inference: ab test<br>• Algorithms: model |
| LinkedIn | • Strategy and insights: analyst<br>• Inference and algorithms: ab test, ml model<br>• Data engineering: relatively new |
| Shopify | • Data pipeline/dashboard<br>• Machine learning (eng)<br>• A/B testing |
| Amazon | • Data scientist ~= Business Analyst<br>• Research scientist<br>• Applied scientist |

Some researchers claim three concentrations.


Engineering
the process of making everything else possible
Data environment
1
data storage, Kafka platform, Hadoop and Spark cluster etc.
Data management
2
parsing the logs, web scraping, API queries, and interrogating data streams.
Production
3
integrate model and analysis into the production system

*Source: ScientistCafe, 2022 ASA Course*



Analysis

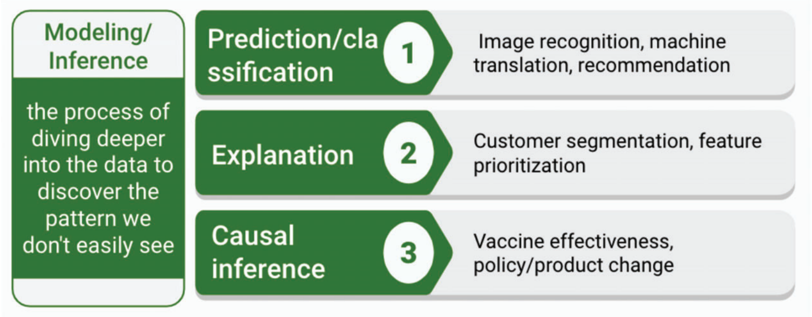
the process of getting insights through exploratory analysis

Domain knowledge 1 Understand the business

Exploratory analysis 2 Correlation and summary statistics

Storytelling 3 Share data story with stakeholders usually through visualization

*Source: ScientistCafe, 2022 ASA Course*



Modeling/ Inference

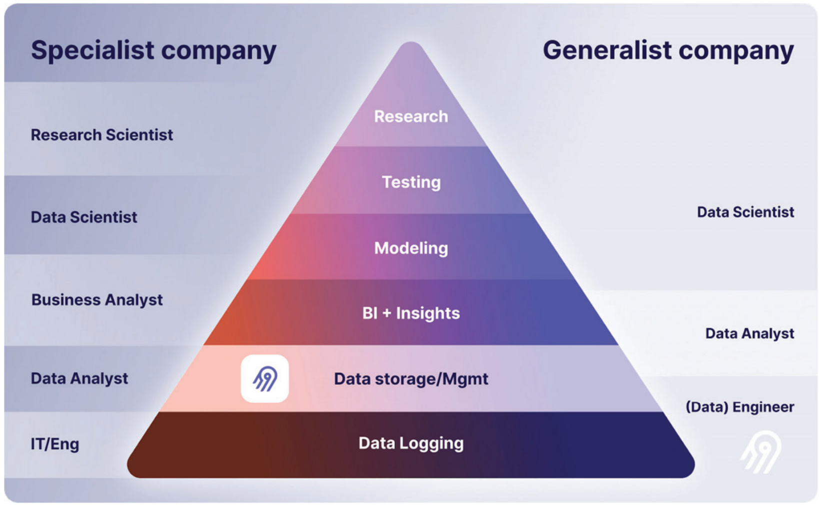
the process of diving deeper into the data to discover the pattern we don't easily see

Prediction/classification 1 Image recognition, machine translation, recommendation

Explanation 2 Customer segmentation, feature prioritization

Causal inference 3 Vaccine effectiveness, policy/product change

*Source: ScientistCafe, 2022 ASA Course*



Specialist company

Research Scientist

Data Scientist

Business Analyst

Data Analyst

IT/Eng

Research

Testing

Modeling

BI + Insights

Data storage/Mgmt

Data Logging

Generalist company

Data Scientist

Data Analyst

(Data) Engineer

*Source: Airbyte Blog, 2024*

## Why Data Science Developed?

John Tukey **in 1962** regarding forces driving data analysis:

- The formal theories of math and statistics
- Acceleration of developments in computers and display devices
- The challenge, in many fields, of more and ever larger bodies of data
- The emphasis on quantification in an ever-wider variety of disciplines

All of the above is still relevant, just the scale of the data and the computing power are both immensely larger…

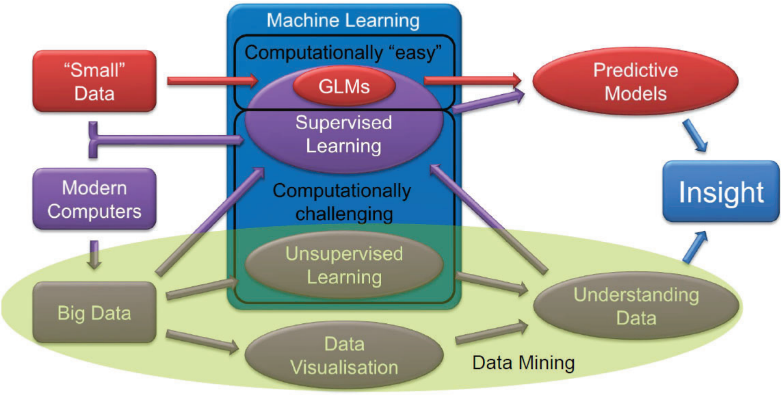
  - ○ Also, we don't use the term "display device" that much.

## Why Data Science Developed?

- The digital revolution is creating vast amounts of data.
- Advances in statistical and machine learning methods via theoretical work and a lot of computer experimentation.
- The Major Impetus: vast advances in computing power, use of GPUs for scientific computing.

- ○ One iPhone 15 Pro is about 100x more powerful than the supercomputer array that rendered Jurassic Park.
- ○ iPhone 15 Pro is perhaps 1,000,000x more powerful than the UNIVAC 1219 supercomputer that run USS Midway Aircraft carrier in the Fifties.

## Tradditional Data Science



Source: "An introduction to the data science universe" by Mark Lee https://www.actuaries.org.uk/practice-areas/general-insurance/research-working-parties/modelling-analytics-and-insights-data

## Main Recent Development is Generative AI

### Large Language Models (LLMs)

- **Technology**: Utilize transformer models with self-attention mechanisms to process and generate text.
- **Applications**: Employed for Natural Language Processing, tasks like text generation, translation, summarization, etc.

### Generative Adversarial Networks (GANs)

- **Components**: Consist of two neural networks - a generator that creates fake data and a discriminator that distinguishes between real and fake data.
- **Applications**: Used for creating realistic images, videos, etc.

### Both LLMs and GANs are Deep Neural Networks

## Why R?

- No matter where you aim to be career-wise in the field of Data Science, you need to be able to read and understand code and do coding alone or perhaps with some help of AI.

- Python is becoming the most popular language in data science, largely because many industry professionals have a background in computer science.
  - ○ This is especially true for advanced stuff like LLMs.
  - ○ Better for certain tasks (e.g., reading large files).
  - ○ Mature machine learning environment (Keras, PyTorch).
  - ○ Easier to parallelize code.

- R is much used in Academia and Research.
  - Efficient and more intuitive for statistical tasks.
  - Well-documented with detailed CRAN packages.
  - Widely used in statistics textbooks.
  - Easier and (arguably) better visualization.
  - Perhaps, better to start with R if you have no programming experience.

- The goal of this class is to introduce you to Data Science through R, along with some basic statistics and computing concepts.
  - Several classes in your letter academic curriculum will be using R too.

## 2. Overview and Installation

What is R?

- R is free statistical software.
- R is high level programing language.
  - It is an open-source implementation of the S programming language.
  - Official description is:
    *R is a programming language and a free software environment for statistical computing and graphics.*

I am grateful to Dr. J. French and Dr. S. Santorico, CU Denver, and
Dr. Y. Deng, PFW, for sharing their introductory R materials and advice

Why use R?

- It's free!
- Widely used, especially for research.
- You can produce publication-quality graphics.
- R is *interactive*: you type what you want and get out results.
  - It is much easier to get started programming.
    You get immediate feedback and you can immediately address syntax and other errors.

- R contains advanced statistical routines not yet available in most other software.
  - Users write functions and easily add software libraries, a.k.a. *packages*, to R.
  - More than ten thousand packages are available from the official R depositary. Even more at GitHub.
- R is highly vectorized and can run relatively fast.
  - You dive deep into programming, e.g., by calling C and C++ routines if you need even more speed.
  - You can do Machine Learning and Artificial Intelligence relatively easily from within R, e.g. by indirectly calling Google's TensorFlow AI libraries without the need to learn Python, to optimize your code for GPU, etc.

Although we could use the R interface directly, we will be using RStudio IDE instead.

- *RStudio* is free, open-source Integrated Development Environment (IDE) for R.
  - ○ Commonly used as it makes life easier.
  - ○ Allows for:
    - ▪ *syntax highlighting*
    - ▪ *interactive autocomplete*
    - ▪ *easier data/object/graph viewing, etc.*

- I arranged for both RStudio and R to be able to install via Purdue Fort Wayne (PFW) software deployment system on *classroom and lab* computers.
  - ○ Double click at the *Purdue FW Software Center* icon at your Desktop and search for *RStudio*. Press “Install”. Both R and RStudio will install automatically.
  - ○ To start, type RStudio in the taskbar.

*Installation on a Non-PFW Computer*

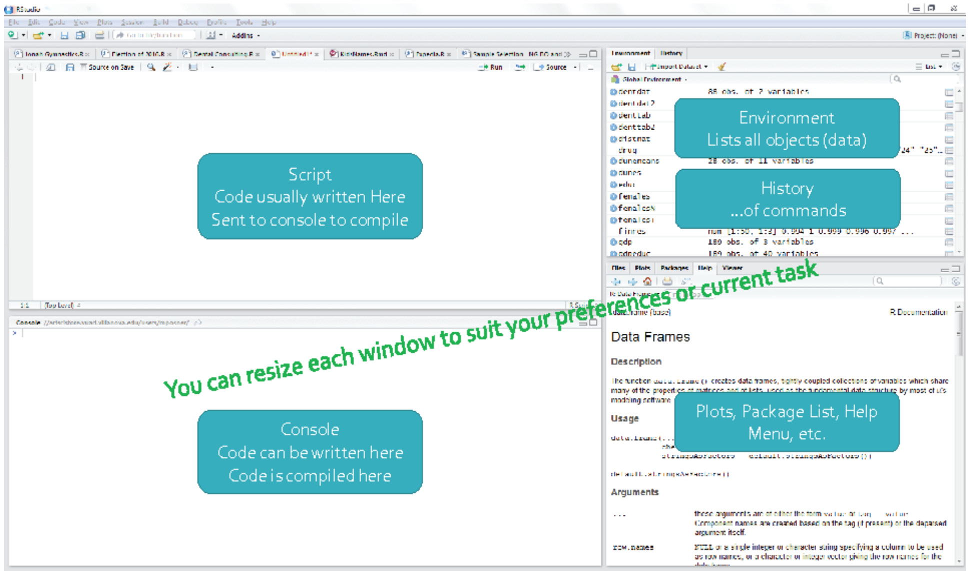
- ○ Download from `https://cloud.r-project.org/` (or simply google "download R" and click on the first link).
- ○ Save the executable file and remember the location.
- ○ (Double-)click the file and follow the instructions.
- ○ Once the installation is complete, start R (typically the icon is on your Desktop).
- ○ Test R by computing `2+2` (or something else :).
- ○ To quit R, type `q()` and hit enter.
- ○ Next, download and install the free version of RStudio Desktop from

`https://www.rstudio.com/products/rstudio/download/`

- RStudio is free for personal or academic use.
- Please **install R first**. RStudio will then automatically detect the location of the R installation.
- You can also **quickly run something in R** without installing R on the computer you are using.
  - ○ You can use an online emulator in a web browser.
  - ○ For example, if I need to do a quick computation, I use the first “USEFUL LINK” from my Quick Course in R webpage.

  `https://users.pfw.edu/yorgovd/R-Tutorial/`

  I keep an updated link to an online emulator at this webpage.


Script
Code usually written Here
Sent to console to compile
Environment
Lists all objects (data)
History
...of commands
You can resize each window to suit your preferences or current task
Console
Code can be written here
Code is compiled here
Plots, Package List, Help
Menu, etc.

## 3. Help

There are many ways to get help for R:

- On the Internet at `www.r-project.org`
- `help.start()` will start a local browser with many locally accessible manuals, etc. for these rare moments without an internet access.
- If you know the command you need help for, from the R command line type:

  `help(command)`, e.g., `help(lm)`

  `?command`, e.g., `?lm`, `?"+"`

- If you only know the topic you need help for, from the R command line type:

  `??topic`, e.g., `??logarithm`

- Examples are available for many functions:

  `example(lm)`

  ○ Or just scroll to the end of the help

Great overview at `https://www.r-project.org/help.html`

Many more resources are available online.

I personally often use Quick-R if I need a quick external reference for something that I forgot: `https://www.statmethods.net/about/sitemap.html`

RStudio (called Posit now) Cheatsheets - RStudio are great: `https://www.rstudio.com/resources/cheatsheets/`.
Other resources about R that may be helpful:

- A rather thorough "Short Reference Card":

`https://cran.r-project.org/doc/contrib/Baggott-refcard-v2.pdf`

- Free "official" R notes from R-Project.org:

*Notes on R: A Programming Environment for Data Analysis and Graphics*, by W. N. Venables, D. M. Smith and the R Core Team

`https://cran.r-project.org/doc/manuals/R-intro.pdf`

- *R for Beginners, by E. Paradis*
`https://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf`

Also, many books have examples that you can download and modify.

For example:

- *Linear Models with R, Second Edition (LMR)*, by Julian J. Faraway (a lot of R code; one of our examples follows his book and uses his R Package)
- *R Cookbook*, by Paul Teetor
- *The Art of R Programming*, by Norman Matloff (Very good reviews)
- *An Introduction to Statistical Learning*, by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani (modern machine learning topics), PDF file free at https://trevorhastie.github.io/ISLR/

- R for Data Science by Hadley Wickham and Garrett Grolemund, free at https://r4ds.had.co.nz/
- Modern Dive: Statistical Inference via Data Science. A Modern Dive into R and the Tidyverse, by Chester Ismay and Albert Y. Kim at https://moderndive.com/

Many more free and paid tutorials, courses, and videos are available online (including on YouTube); for instance:

http://cyclismo.org/tutorial/R/

We will be using DataCamp, arguably, "the best way to ease into data analysis online".

This resource is with some free and many paid online courses with hundreds of hours of videos on R and related to R topics. There are very good automatic exercises in the courses too.

- You can continue your Data Science learning journey selecting the online classes that you take based on your current needs and interests.
- There are also <u>many</u> courses on Python and SQL and anything Data Science related.
  - ○ Number of courses keeps growing fast too, currently at more than 350.

As part of this class, you have **Unlimited Free Access to All Courses in DataCamp** through the invitation provided.

datacamp

- Please use the link here or in the syllabus to register.

https://www.datacamp.com/groups/shared_links/f34cd4ffd0a2e05705e422459898ecf0dfc5693d34ecb4b9c9124542fdbcdb2e

- Please, do not share this link as I have limited number of complimentary subscription slots secured.
  - ○ We will partially flip some of the class and we will assign several topics to do online. I fully expect you to complete these. Details to come.

- For instance, here is a link if you would like to review some of the topics that we will cover in our first class sessions: https://www.datacamp.com/courses/free-introduction-to-r
- What to do after this class? Which courses to take?
  - This will depend on your interests and needs.

If you want to continue with DataCamp…

- There are numerous Skill Tracks: https://learn.datacamp.com/skill-tracks
- There are also Career Tracks: https://learn.datacamp.com/career-tracks

Let's explore some!

## Help from AI

- A Large Language Model (LLM) is a type of artificial intelligence (AI) algorithm that applies neural network techniques to process and "understand" human languages or text including code.
- LLMs can recognize, summarize, translate, predict and generate text and other forms of content based on knowledge gained from massive datasets. Notable examples of LLMs include OpenAI's GPT 4.0 model, Google's PaLM, and Meta's LLaMa, as well as BLOOM, Ernie 3.0 Titan, and Claude.

- LLMs are very good at coding in any language and can even translate between programming languages.
  - However, it's important to review their output including code and understand their limitations.
  - The code you submit here (and in real life) is your responsibility.
- Some potential issues to consider include:
  - They may not always follow the exact instructions.
  - They may generate code that does not work.
  - They may hallucinate code, meaning they generate code that is not relevant or does not make sense.
- To use AI effectively, you need to have a good understanding of what you are doing.

- If you are proficient in coding and know what you are doing, using AI can boost your productivity.
- Currently, I recommend trying Claude, which is available for free, and GPT 4.0, which is also available for free. Microsoft Co-Pilot is also GPT4.0 is my understanding but a version that is overly cautious it seems.

## 4. Data and Data Structures

R actually operates on **data structures**.

A data structure is an object, some sort of "container" that holds certain kinds of information.

Common R data structures:

- Vector (a sequence of *numerical*, *character*, *factor/categorical*, or *logical* elements of the *same type*)
  - Even a single number is still stored internally as a vector.
- Lists (collection of other objects, e.g., vectors of *any type* and *any length*).

- Matrices/Arrays (multi-dimensional collection of vectors of the *same type* and *same length*)
- Data Frame (list of vectors of *equal length* but possibly *different data types*).
  - Think of a Data Frame as a cross between a matrix and a list:
    - Columns of variables of different types but the same length.

A **vector** is a sequence of values of the same data type.

The c function (concatenate) can be used to join data from end to end to create vectors.

The calls below will create a numeric, a character, and a Boolean vector.

```
c(1, 2, 5.3, 6, -2, 4)
c("one", "two", "three")
c(TRUE, FALSE, TRUE)
```

The seq function (from sequence) can be used to create an equidistant series of values.

- A sequence of numbers from 1 to 10 in increments of 1.
  ```
  seq(1, 10)
  1:10
  ```
- A sequence of numbers from 1 to 20 in increments of 2.
  ```
  seq(1, 20, by = 2)
  ```
- A sequence of numbers from 10 to 20 of length 100
  ```
  seq(10, 20, len = 100)
  ```

The `rep` function (from replicate) can be used to create a vector by replicating values.

- Repeat the sequence 1, 2, 3 three times in a row.
```
rep(1:3, times = 3)
```
- Repeat "trt1" once, "trt2" twice, and "trt3" three times.
```
rep(c("trt1", "trt2", "trt3"), times =
1:3) # repeat trt1-once, trt2-twice,
trt3 – tree times
```
- Anything on a line after a “#” character is a *comment* (R will ignore all comments).
    - Comments are great to tell others (and often to remind yourself) what you did in your code.

- Repeat each element of the sequence 1, 2, 3 four times
```
rep(1:3, each = 4)
```
- To access the last output in R, type
```
.Last.value  #last output
```
- To store a data structure in the computer’s memory we must assign it to an object (name). Names are <u>case sensitive</u>.
- Data structures can be stored using the assignment operator "<-" or ("=")
    - For example, store the sequence from 1 through 5 in an object named `v1`.
```
v1 <- 1:5
```

- To access the data stored in an object, we simply type the variable name into R and hit enter.
```
v1
```
- Alternatively, the `print()` function can display the entire object.
- Vectors can be combined and stored in a single vector using the `c` function and the assignment operator.
```
v2 <- c(1, 10, 11)
new <- c(v1, v2); new
```
    - Note that the semicolon above allowed you to type more than one command in a row.

A **matrix** can be created with the `matrix()` function. For example:

```
A<-matrix(data=1:6, nrow=3,ncol=2); A
B<-matrix(data=1:6, nrow=5, ncol=10,
byrow=T); B
```

Notice that for B above the matrix function automatically repeats the “data” `1,2,3,4,5,6` as needed in order to fill-in all the entries.

```
dim(B) #gives dimensions of the matrix B
```

For all ways to create a matrix, type
```
?matrix
```