

Introduction to R Workshop

**Session 1
Intro to R and R
Studio
May 6, 2019**

Goals

1. Get oriented to R and RStudio
2. Learn some fundamentals of coding

Objectives

1. Log in and tour RStudio Cloud
2. Execute code at the console
3. Define and use functions
4. Define and create objects in the environment
5. Interact with a dataframe

Getting Oriented to R

What is R?

- R is a statistical programming language.
- Using R you can load, analyze, and visualize data.
- R also provides an environment in which we can conduct reproducible data analysis.
 - Documented
 - Revisable
 - Shareable



RStudio: The Portal to R

- RStudio is an integrated development environment (IDE)
- Using RStudio we can interact with the R programming language to:
 - Write and execute code interactively
 - View data
 - Debug and fix errors
 - Author our code



RStudio: In the Cloud... In Your Home

- RStudio Cloud: An online hosted version of RStudio that we will use for today's course
- RStudio Desktop: A locally installed version of RStudio that you will use when you get home to continue your learning



Your Turn #1

Navigate to: <https://bit.ly/2DaTFwj>

Enter your log in credentials

Select “API R Workshop 2019”

Make a copy of the workspace for yourself



Spaces

Your Workspace

API R Workshop 2019

+ New Space

Learn

Guide

What's New

Primers

DataCamp Courses

Cheat Sheets

Feedback and Questions

Info

Terms and Conditions

System Status

Welcome to API R Workshop 2019

If you did not intend to join this space, or you later decide you don't want to be a member, just go to the [Members](#) area and click "Leave Space".

[Terms and Conditions](#) [System Status](#)

© 2019 RStudio Inc.



All Projects

New Project

Options



API R Project



Amrom Obstfeld

Created Apr 5, 2019 3:59 PM

 Copy

Search Projects



List Projects

 All Shared with everyone Yours

Sort Projects

 By name By date created

Spaces

Your Workspace

API R Workshop 2019

+ New Space

Learn

Guide

What's New

Primers

DataCamp Courses

Cheat Sheets

Feedback and Questions

Info

Terms and Conditions

System Status

File Edit Code View Plots Session Build Debug Profile Tools Help

A row of small icons for file operations: Create, Open, Save, Print, etc.

Go to file/function



Addins

R 3.5.2

Console Terminal Jobs

/cloud/project/

```
R version 3.5.2 (2018-12-20) -- "Eggshell Igloo"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
```

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
```

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

> |

Environment History Connections Git

A row of small icons for environment management: File, Import Dataset, etc.

Global Environment

Environment is empty

Files Plots Packages Help Viewer

A row of small icons for file operations: New Folder, Upload, Delete, Rename, More.

Cloud > project

	Name	Size	Modified
	..		
	.gitignore	581 B	Apr 5, 2019, 4:00 PM
	.Rhistory	0 B	Apr 5, 2019, 4:30 PM
	03 - Transform.pptx	2.1 MB	Apr 5, 2019, 4:04 PM
	03 - transform.Rmd	7.3 KB	Apr 5, 2019, 4:00 PM
	data		
	LICENSE	1 KB	Apr 5, 2019, 4:00 PM
	project.Rproj	205 B	Apr 5, 2019, 4:30 PM
	README.md	11 B	Apr 5, 2019, 4:00 PM

Console Terminal x Jobs x

/cloud/project/ ↵

```
R version 3.5.2 (2018-12-20) -- "Eggshell Igloo"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
```

```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
```

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

> |

CONSOLE

Environment History Connections

Import Dataset Global Environment

Environment is empty

ENVIRONMENT

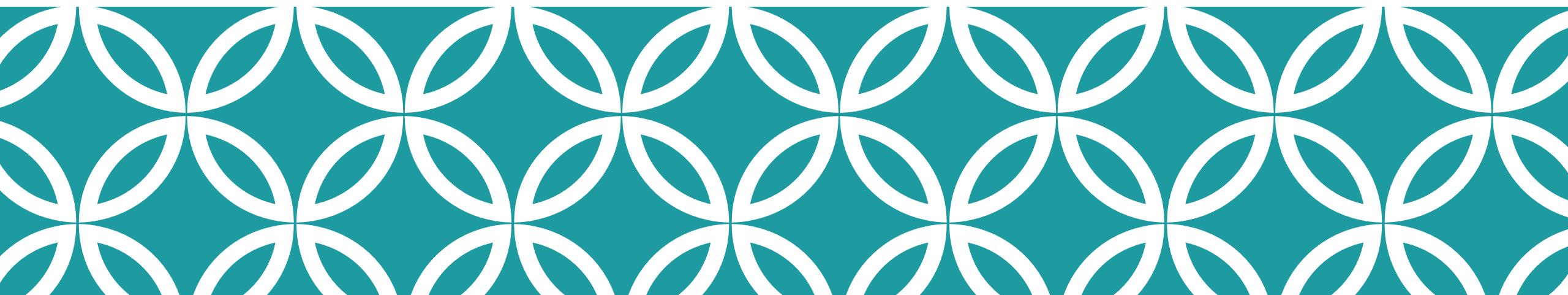
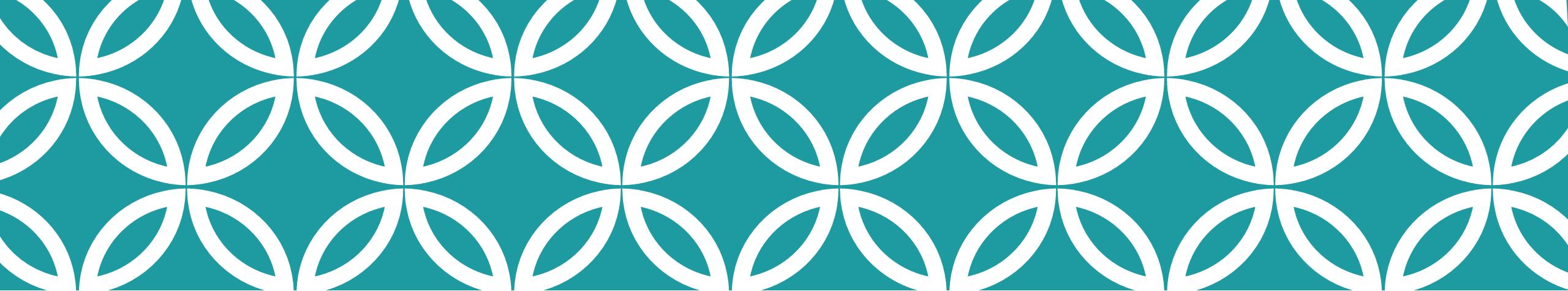
Files Plots Packages Help Viewer

New Folder Upload Delete Rename More

Cloud > project

	Name	Size	Modified
	..		
	.Rhistory	0 B	Apr 20, 2019, 1:39 PM
	project.Rproj	205 B	Apr 20, 2019, 1:39 PM

OUTPUT



The Basics of Coding

The Basics of Coding: Calculation

- R is a calculator!

```
> 2 + 3 + 2  
[1] 7  
>  
>  
> 4 * 20  
[1] 80  
>  
>  
> 6 ^ 8  
[1] 1679616  
>
```

enter/return to
execute code

answer returned
here

Your Turn #2

Place your cursor at the console and click to enter the console.

Complete the following calculation:

- For the date 12-29-1974
- Take the four digit year
- Subtract the month then multiply by the day



What did you get?

- A four digit number? A five digit number?

```
> 1974 - 12 * 29
```

```
[1] 1626
```

```
>
```

```
>
```

```
> (1974 - 12) * 29
```

```
[1] 56898
```

- Order of operations matters!

The Basics of Coding: Functions

- Code that extends our reach beyond the basic operators

```
> abs(-77)  
[1] 77  
>
```

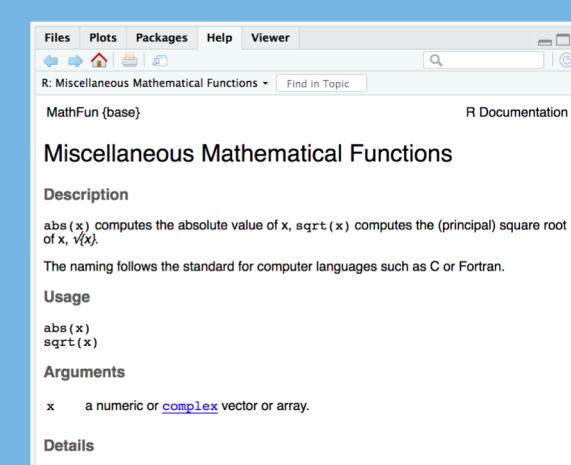
- What I don't know what a function does?

```
>  
> ?abs()  
>
```

function
(does stuff)

argument
(input)

`abs(-77)`



Your Turn #4

Launch the help function for seq() using help (?)

Read about the seq() function in the output window

Can you describe what the seq() function does?



Putting Functions to Work

- We can use functions to do more than simple math, we can make things!
- We can create a series of integers (a vector) using the `seq()` function

```
>  
> seq(from=5, to=150, by=10)  
[1] 5 15 25 35 45 55 65 75 85 95 105 115 125 135 145
```

The Basics of Coding: Objects

- Objects are the vessel for your output

object

(stores output)

function

(does stuff)

arguments

(input)

```
sequence_of_10s <- seq(from=5, to=150, by=10)
```

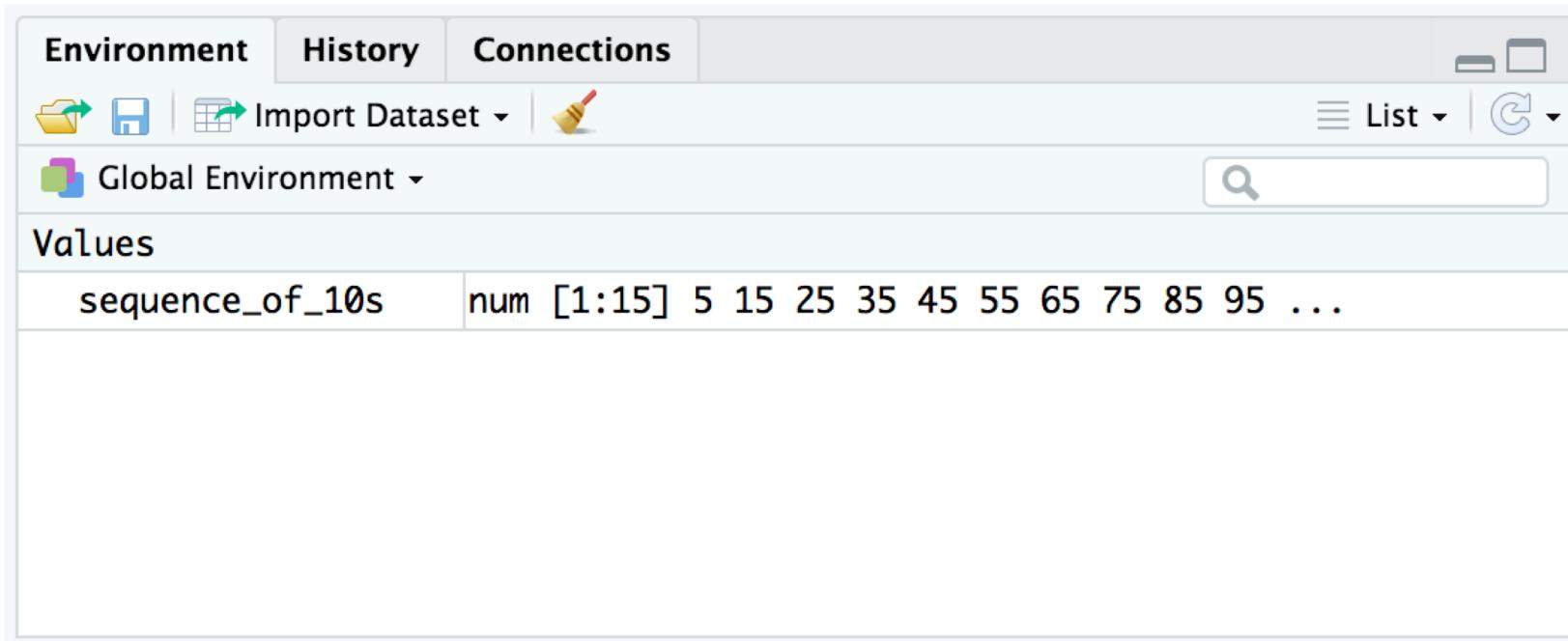
Checking the Contents of an Object

- Entering the object name at the console allows us to output the contents of an object.

```
>  
> sequence_of_10s  
[1] 5 15 25 35 45 55 65 75 85 95 105 115 125 135 145
```

Checking the contents of an object

- The environment tab shows us the objects we have created.



Bending objects to your will

- Once we have created an object we can start to interact with it.
- This includes passing our objects to other functions... Whoa!

```
>  
> min(sequence_of_10s)  
[1] 5  
>  
> max(sequence_of_10s)  
[1] 145  
>
```

Your Turn #5

Generate a sequence, store it to an object, and ply your object*

-use the `seq()` function to create a vector from 0 to 500 by 25.

-assign your sequence to an object called sequence of 25s

-check the contents of your object (at the console and in the environment tab)

-use the `median()` function to find the median value of your vector

*Use the help (?) feature or your neighbor if you get stuck



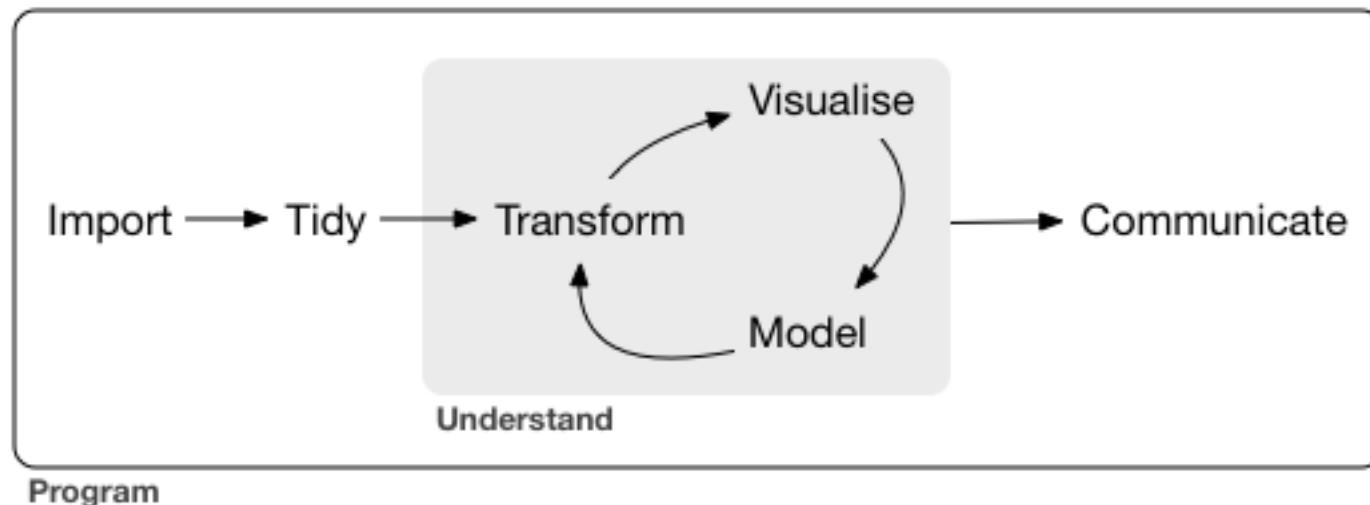
The Basics of Coding: Packages

- A package is a collection of functions.
- Packages extend the capabilities of the base R programming language.
- The **tidyverse** includes functions for reading data into the R environment, cleaning and manipulating data, and plotting our results.



Getting Tidy in the Verse

- Being "Tidy" with your data is important!



Installing and Loading Packages

- Installing a package

function
(does stuff)

arguments
(input)

```
install.packages("tidyverse", dependencies = TRUE)
```

- Loading into your environment

```
library(tidyverse)
```

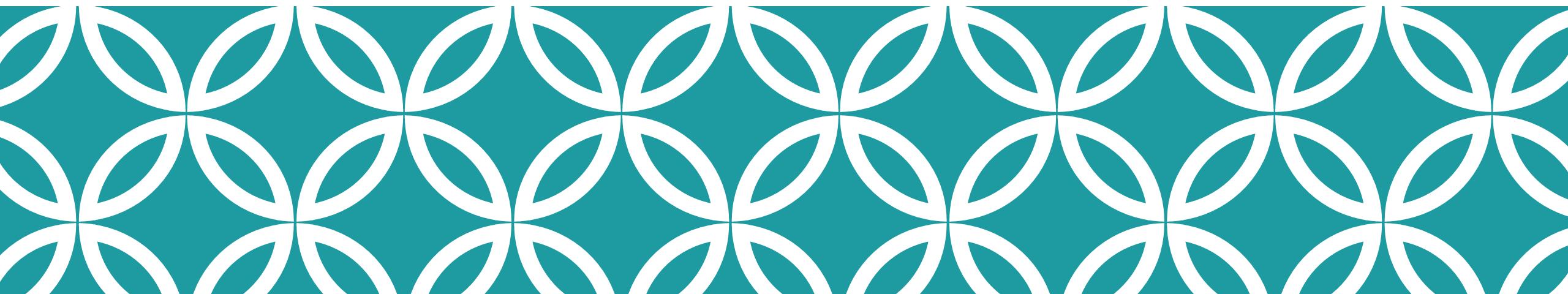
Your Turn #5

Install a package and load it into the environment

-Leverage the `install.packages()` function to install a package called "janitor"

-Load the package into your environment using the `library()` function.

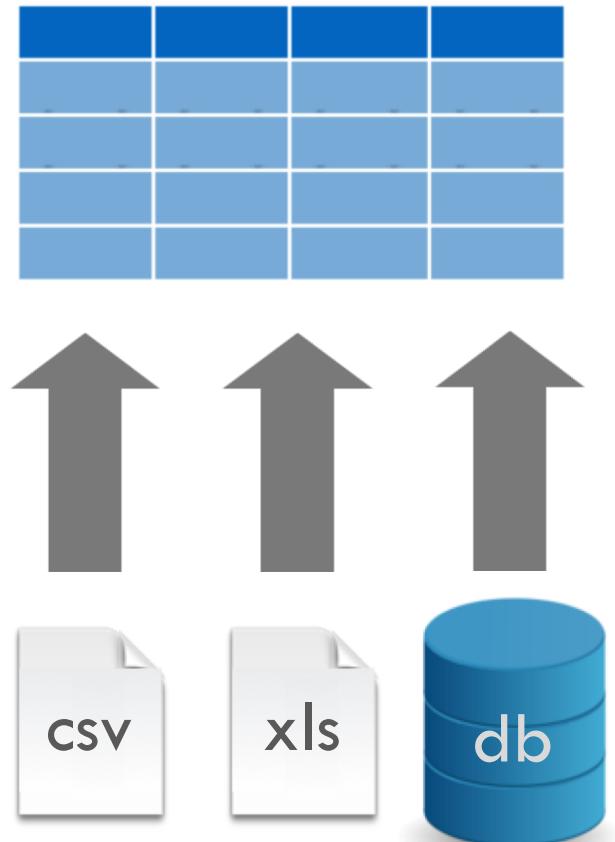




Working with Dataframes (aka Useful Data)

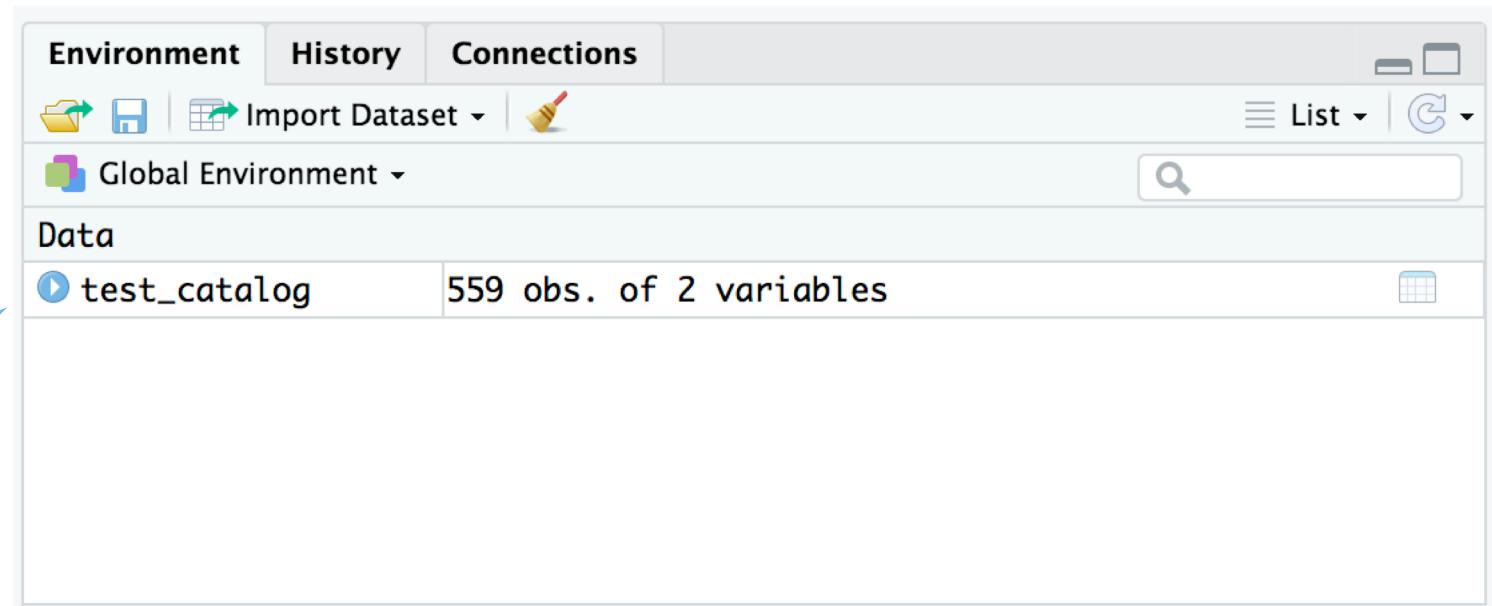
Dataframes: Beyond the Vector

- Dataframe is the term for a table
- Dataframes are composed:
Columns (Variables)
Rows (Observations)
- Dataframes are objects and can be acted on like other objects



Viewing the Contents of a Dataframe

- We have already loaded a data frame called **test_catalog**.



Viewing the Contents of a Dataframe

559
Observations
(Rows)

	description	proc_code
1	1,25 DIHYDROXY VITAMIN D	VTD125
2	17-HYDROXYPROGESTERONE	OHPORG
3	1H GLUCOSE TOL TEST, NON-PREGNANT	GLUPP1
4	1ST EXTRA BLUE TOP	XBL
5	2H GLUCOSE TOL TEST, NONPREG	GLUPP2
6	5_HYDROXYINDOLACETIC ACID, URN	RUHIA
7	6_THIOGUANINE & 6_MMP	R6TGG
8	A1C RAPID, ONSITE	83036
9	ACANTHAMOEBA CULT & WETMOUNT	ACANC
10	ACETAMINOPHEN (TYLENOL)	TYL
11	ACH RECEPTOR (MUSCLE) BIND AB	RACHRM

2 Attributes
(Columns)

Working with Dataframes at the Console

- The `summary()` function provides you with some abbreviated metadata about your dataframe.

```
summary(object_name)
```

Number of
Observations
(Rows)

Data Type

```
>  
> summary(test_catalog)  
description      proc_code  
Length:559       Length:559  
Class :character Class :character  
Mode  :character Mode  :character  
>  
>
```

Attributes
(Columns)

Working with Dataframes at the Console

- The `head()` function is helpful for displaying a snippet of your dataframe

```
head(object_name, n=number of rows to view)
```

```
>  
> head(test_catalog, n=5)  
# A tibble: 5 x 2  
  description                      proc_code  
  <chr>                            <chr>  
1 1,25 DIHYDROXY VITAMIN D        VTD125  
2 17-HYDROXYPROGESTERONE          OHPROG  
3 1H GLUCOSE TOL TEST, NON-PREGNANT GLUPP1  
4 1ST EXTRA BLUE TOP              XBL  
5 2H GLUCOSE TOL TEST, NONPREG    GLUPP2  
>
```

Sample of Data
in Your Object

Your Turn #6

Summarizing Dataframes and Understanding Object Contents

- Use the **summary()** function on our object **test_catalog**
- Leverage the **tail()** to review the **last 10 rows** of the data frame.
- What is the 10th test from the end of the table?





Taking the Leap to Visualization

Even Simple Dataframes Can Be Interesting

- With a few lines of code we can turn our test catalog...

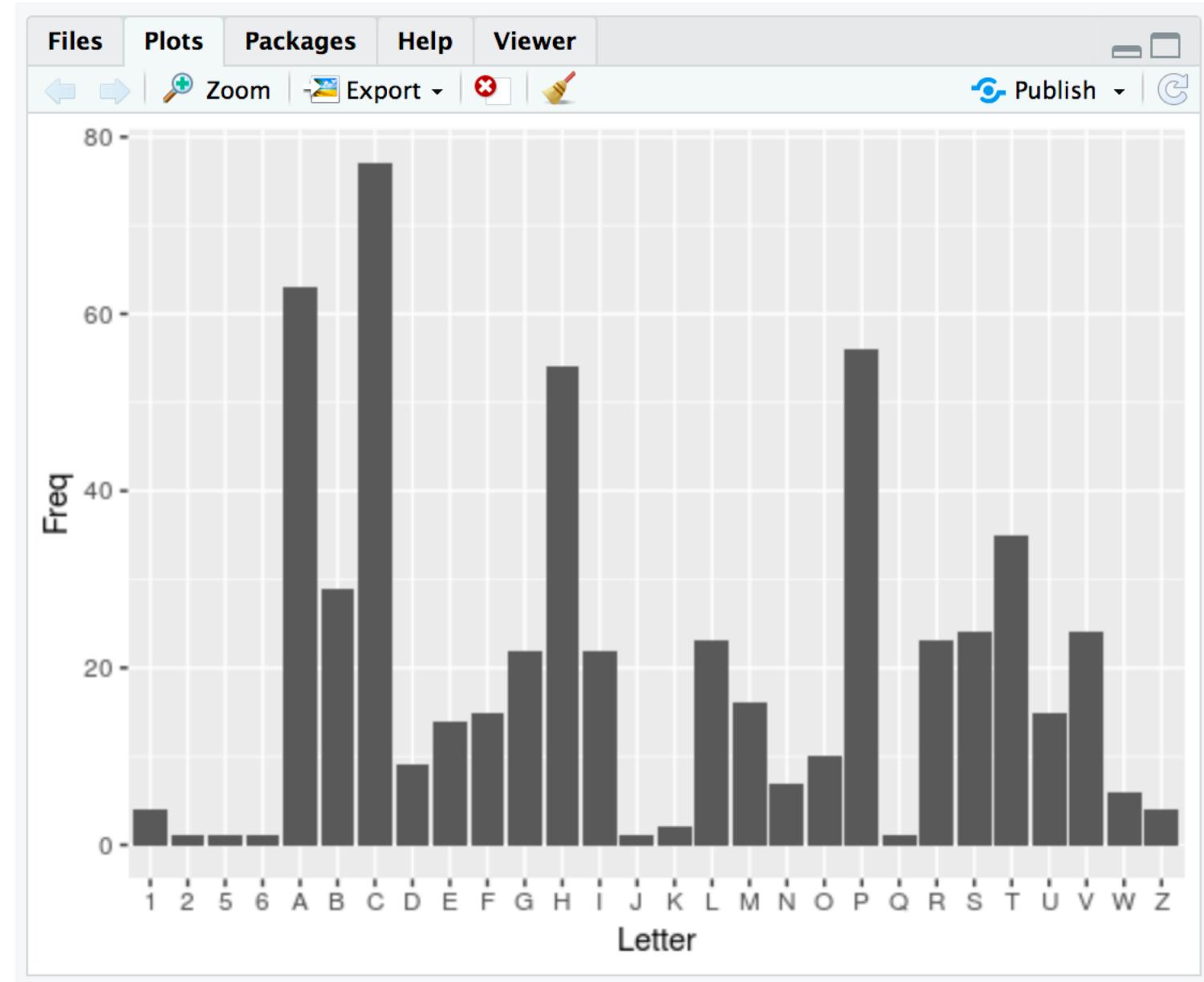
Identify
First Letter
in the Test
Name

```
>  
> alphabet_analysis <- test_catalog  
> alphabet_analysis$first_letter <- substr(alphabet_analysis$description, 1, 1)  
> alphabet_analysis <- data.frame(table(alphabet_analysis$first_letter))  
> alphabet_analysis <- rename(alphabet_analysis, Letter = Var1)  
> ggplot(data=alphabet_analysis, aes(x=Letter, y=Freq)) +  
+   geom_bar(stat="identity")  
>
```

Make a
bar plot of
the Data

Count the
Test
Names by
First Letter

- Into an insightful visualization



Console... I only just knew you

- As the complexity of our coding increases, the ability to iterate our lines of code as we write becomes apparent
- We find it increasing difficult to manage the names and states of objects in our environment
- And the console shows us its limitations

A Hero is Born



05 : 00