

# Intro to R for Ecologists Course

Dr. Erin Witkop  
District Fish Biologist Willapa Bay  
June 5<sup>th</sup>-June 7<sup>th</sup>  
48 Devonshire Rd. Montesano WA, 98563



# Course Information

- 3-day course 8-12pm
- Curriculum closely based on Data Analysis and visualization in R for Ecologists by Data Carpentry
- See [github](#) repo for course info and documents
- [Etherpad](#) document for collaborative notetaking
- Use sticky notes to signal when help is needed



# Introductions

- Course instructor – Dr. Erin Witkop
- Course helpers – Dr. Matthew Siskey, Marlene Wagner, Dr. Matthew George
- Special Thanks to
  - James P. Losee
  - Jennifer Allan
  - Ryan Hilleary
- Participant intros



# Workshop goals

- Empower your data analysis goals
- Enable reproducibility of repetitive analysis tasks
- Encourage culture of collaboration and feedback



# Agency R resources

- DFW-R Users Teams Channel
  - List of sharepoint resources:  
[https://stateofwa.sharepoint.com/:f:/r/sites/DFW-TeamWDFWRWorkgroup/Shared%20Documents/General/R sharepoint resources?csf=1&web=1&e=geDLKx](https://stateofwa.sharepoint.com/:f:/r/sites/DFW-TeamWDFWRWorkgroup/Shared%20Documents/General/R%20sharepoint%20resources?csf=1&web=1&e=geDLKx)
- R studio online server
- Jason Neuswanger ChatGPT presentation: <https://youtu.be/o3-oK1GDQkw?si=9wHXIHHSBP-5hjWc&t=2591>
- [Washington Department of Fish & Wildlife, Fish Program \(github.com\)](https://github.com/WashingtonDepartmentofFishandWildlife/FishProgram)



# Day 1 Agenda

Time	Topic	Data Carpentry Curriculum
7:30	Set-up/software help	
8:00	Introduction	
8:10:00 - 8:45	Before we Start	<a href="https://datacarpentry.org/Data-Analysis-in-R-for-Ecologists/01-before-we-start/">Data Analysis and Visualisation in R for Ecologists: Before we start (datacarpentry.org)</a>
8:45-9:30	Introduction to R	<a href="https://datacarpentry.org/Data-Analysis-in-R-for-Ecologists/02-introduction-to-r/">Data Analysis and Visualisation in R for Ecologists: Introduction to R (datacarpentry.org)</a>
9:30-9:45	BREAK	
9:45-10:00	Introduction to R cont.	<a href="https://datacarpentry.org/Data-Analysis-in-R-for-Ecologists/02-introduction-to-r/">Data Analysis and Visualisation in R for Ecologists: Introduction to R (datacarpentry.org)</a>
10:00-11:00	Starting with Data in R	<a href="https://datacarpentry.org/Data-Analysis-in-R-for-Ecologists/03-starting-with-data/">Data Analysis and Visualisation in R for Ecologists: Starting with data (datacarpentry.org)</a>
11:00-11:15	BREAK	
11:15-12:00	Manipulating, analyzing and exporting data with tidyverse	<a href="https://datacarpentry.org/Data-Analysis-in-R-for-Ecologists/04-manipulating-analyzing-and-exporting-data-with-tidyverse/">Data Analysis and Visualisation in R for Ecologists: Manipulating, analyzing and exporting data with tidyverse (datacarpentry.org)</a>



# Before we start

## OVERVIEW

### Questions

---

- What is R and RStudio?
- What is a working directory?
- How should files be set up to import into R?
- How can I look for help with R functions?

### Objectives

---

- Explain what R and RStudio are, what they are used for, and how they relate to each other.
- Describe the purpose of the RStudio Script, Console, Environment, and Plots panes.
- Organize files and directories for a set of analyses as an R Project, and understand the purpose of the working directory.
- Use the built-in RStudio help interface to search for more information on R functions.
- Demonstrate how to provide sufficient information for troubleshooting with the R user community.



# Introduction to R

## OVERVIEW

### Questions

---

- How do you create objects in R?
- How do you save R code for later use?
- How do you manipulate data in R?

### Objectives

---

- Define the following terms as they relate to R: object, assign, call, function, arguments, options.
- Create objects and assign values to them in R.
- Learn how to *name* objects.
- Save a script file for later use.
- Use comments to inform script.
- Solve simple arithmetic operations in R.
- Call functions and use arguments to change their default options.
- Inspect the content of vectors and manipulate their content.
- Subset and extract values from vectors.
- Analyze vectors with missing data.





# Introduction to R: key points



## KEY POINTS

- `<-` is used to assign values on the right to objects on the left
- Code should be saved within the Source pane in RStudio to help you return to your code later.
- `#` can be used to add comments to your code.
- Functions can automate more complicated sets of commands, and require arguments as inputs.
- Vectors are composed by a series of values and can take many forms.
- Data structures in R include 'vector', 'list', 'matrix', 'data.frame', 'factor', and 'array'.
- Vectors can be subset by indexing or through logical vectors.
- Many functions exist to remove missing data from data structures.



# Starting with Data

## OVERVIEW

### Questions

---

- What is a `data.frame`?
- How can I read a complete csv file into R?
- How can I get basic summary information about my dataset?
- How can extract specific information from a `dataframe`?
- What are factors, and how are they different from other datatypes?
- How can I rename factors?
- How are dates represented in R and how can I change the format?

### Objectives

---

- Load external data from a .csv file into a data frame.
- Install and load packages.
- Describe what a data frame is.
- Summarize the contents of a data frame.
- Use indexing to subset specific portions of data frames.
- Describe what a factor is.
- Convert between strings and factors.
- Reorder and rename factors.
- Change how character strings are handled in a data frame.
- Format dates.



# Dataframes

## Wide vs. Long Format

data frame

1	"S"	TRUE
7	"A"	FALSE
3	"U"	TRUE
numeric	character	logical

id	x	y	z
1	a	c	e
2	b	d	f

 github.io

id	key	val
1	x	a
2	x	b
1	y	c
2	y	d
1	z	e
2	z	f



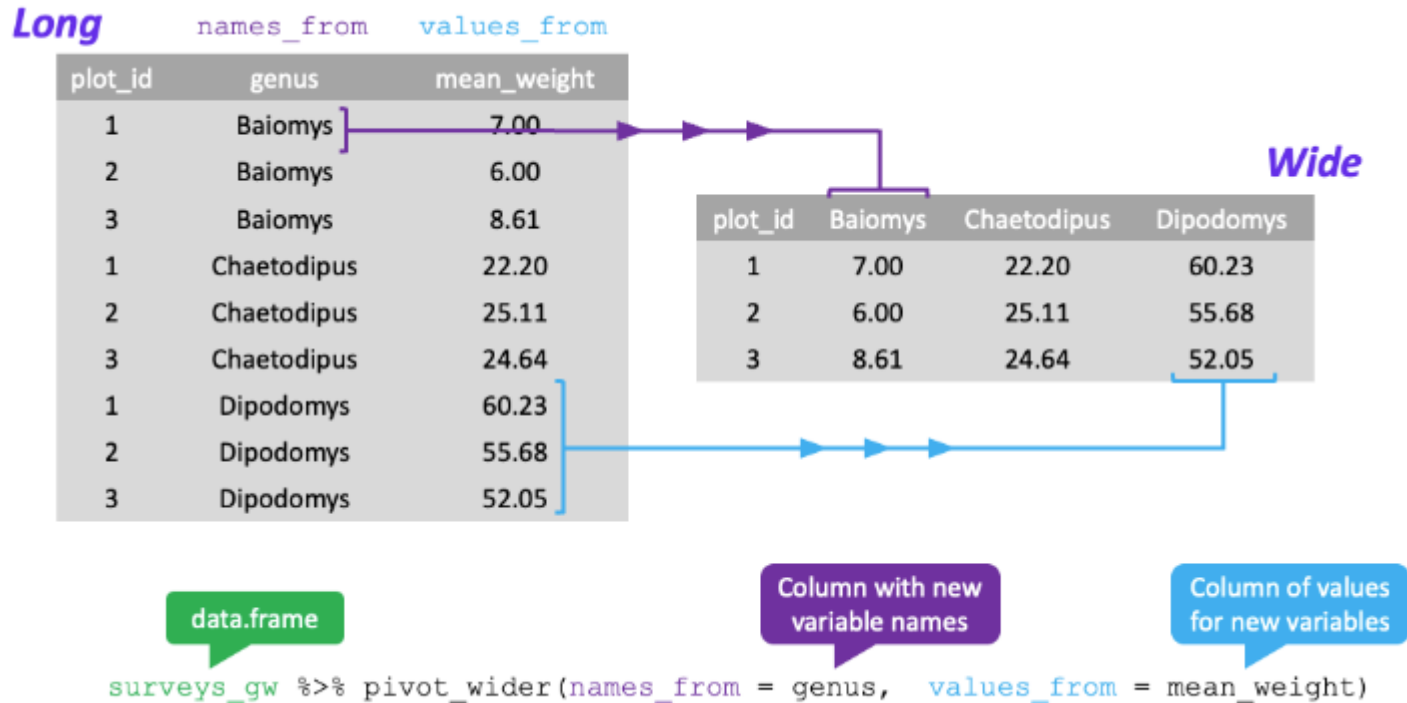
# Rules for tidy data

1. Each variable has its own column
2. Each observation has its own row
3. Each value must have its own cell
4. Each type of observational unit forms a table

id	key	val
1	x	a
2	x	b
1	y	c
2	y	d
1	z	e
2	z	f



# Long vs wide data



# Starting with Data: key points



## KEY POINTS

- Use `read.csv` to read tabular data in R.
- A data frame is the representation of data in the format of a table where the columns are vectors that all have the same length.
- `dplyr` provides many methods for inspecting and summarizing data in data frames.
- Use factors to represent categorical data in R.
- The **lubridate** package has many useful functions for working with dates.



# End of Day 1

- Please provide anonymous comments on your way out
- See you tomorrow!





## Day 2



# Day 2 Agenda

Time	Topic	Data Carpentry Curriculum
8:00-8:30	Starting with Data in R	<a href="https://datacarpentry.org/R-ecology-lesson/01-starting-with-data.html">Data Analysis and Visualisation in R for Ecologists: Starting with data (datacarpentry.org)</a>
8:30-9:30	Manipulating, analyzing and exporting data with tidyverse	<a href="https://datacarpentry.org/R-ecology-lesson/02-manipulating-analyzing-and-exporting-data-with-tidyverse.html">Data Analysis and Visualisation in R for Ecologists: Manipulating, analyzing and exporting data with tidyverse (datacarpentry.org)</a>
9:30-9:45	BREAK	
9:45-11:00	Manipulating, analyzing and exporting data with tidyverse	<a href="https://datacarpentry.org/R-ecology-lesson/02-manipulating-analyzing-and-exporting-data-with-tidyverse.html">Data Analysis and Visualisation in R for Ecologists: Manipulating, analyzing and exporting data with tidyverse (datacarpentry.org)</a>
11:00-11:15	BREAK	
11:15-12:00	Visualizing Data (ggplot)	<a href="https://datacarpentry.org/R-ecology-lesson/04-visualization-ggplot2.html">https://datacarpentry.org/R-ecology-lesson/04-visualization-ggplot2.html</a>



# Starting with Data

## OVERVIEW

### Questions

---

- What is a data.frame?
- How can I read a complete csv file into R?
- How can I get basic summary information about my dataset?
- How can extract specific information from a dataframe?
- What are factors, and how are they different from other datatypes?
- How can I rename factors?
- How are dates represented in R and how can I change the format?

### Objectives

---

- Load external data from a .csv file into a data frame.
- Install and load packages.
- Describe what a data frame is.
- Summarize the contents of a data frame.
- Use indexing to subset specific portions of data frames.
- Describe what a factor is.
- Convert between strings and factors.
- Reorder and rename factors.
- Change how character strings are handled in a data frame.
- Format dates.



# Manipulating, analyzing and exporting data with tidyverse

## OVERVIEW

### Questions

- What are dplyr and tidyr?
- How can I select specific rows and/or columns from a dataframe?
- How can I combine multiple commands into a single command?
- How can I create new columns or remove existing columns from a dataframe?

### Objectives

- Describe the purpose of the **dplyr** and **tidyr** packages.
- Select certain columns in a data frame with the **dplyr** function **select**.
- Extract certain rows in a data frame according to logical (boolean) conditions with the **dplyr** function **filter**.
- Link the output of one **dplyr** function to the input of another function with the 'pipe' operator **%>%**.
- Add new columns to a data frame that are functions of existing columns with **mutate**.
- Use the split-apply-combine concept for data analysis.
- Use **summarize**, **group\_by**, and **count** to split a data frame into groups of observations, apply summary statistics for each group, and then combine the results.
- Describe the concept of a wide and a long table format and for which [↑](#) purpose those formats are useful.
- Describe what key-value pairs are.
- Reshape a data frame from long to wide format and back with the **pivot\_wider** and **pivot\_longer** commands from the **tidyr** package.
- Export a data frame to a .csv file.

[To Top](#)



# Manipulating, analyzing and exporting data with tidyverse

## OVERVIEW

### Questions

- What are dplyr and tidyr?
- How can I select specific rows and/or columns from a dataframe?
- How can I combine multiple commands into a single command?
- How can I create new columns or remove existing columns from a dataframe?

### Objectives

- Describe the purpose of the **dplyr** and **tidyr** packages.
- Select certain columns in a data frame with the **dplyr** function **select**.
- Extract certain rows in a data frame according to logical (boolean) conditions with the **dplyr** function **filter**.
- Link the output of one **dplyr** function to the input of another function with the 'pipe' operator **%>%**.
- Add new columns to a data frame that are functions of existing columns with **mutate**.
- Use the split-apply-combine concept for data analysis.
- Use **summarize**, **group\_by**, and **count** to split a data frame into groups of observations, apply summary statistics for each group, and then combine the results.
- Describe the concept of a wide and a long table format and for which [↑](#) purpose those formats are useful. [To Top](#)
- Describe what key-value pairs are.
- Reshape a data frame from long to wide format and back with the **pivot\_wider** and **pivot\_longer** commands from the **tidyr** package.
- Export a data frame to a .csv file.



# Manipulating Data: Key Points



## KEY POINTS

- Use the `dplyr` package to manipulate data frames.
- Use `select()` to choose variables from a data frame.
- Use `filter()` to choose data based on values.
- Use `mutate()` to create new variables.
- Use `group_by()` and `summarize()` to work with subsets of data.



# Data visualization with ggplot

## OVERVIEW

### Questions

---

- How do you make plots using R?
- How do you customize and modify plots?

### Objectives

---

- Produce scatter plots, boxplots, and time series plots using ggplot.
- Set universal plot settings.
- Describe what faceting is and apply faceting in ggplot.
- Modify the aesthetics of an existing ggplot plot (including axis labels and color).
- Build complex and customized plots from data in a data frame.



# Data visualization with ggplot: Key points



## KEY POINTS

- start simple and build your plots iteratively
- the `ggplot()` function initiates a plot, and `geom_` functions add representations of your data
- use `aes()` when mapping a variable from the data to a part of the plot
- use `facet_` to partition a plot into multiple plots based on a factor included in the dataset
- use premade `theme_` functions to broadly change appearance, and the `theme()` function to fine-tune
- the `patchwork` library can combine separate plots into a single figure
- use `ggsave()` to save plots in your favorite format and dimensions





# Day 3



# Day 3 Agenda

Time	Topic	Data Carpentry Curriculum
7:30-8:00	Set Up and Challenge Review	
8:00-9:30	Visualizing Data Cont. (customization, arranging, exporting)	<a href="https://datacarpentry.org/R-ecology-lesson/04-visualization-ggplot2.html">https://datacarpentry.org/R-ecology-lesson/04-visualization-ggplot2.html</a>
9:30-9:45	BREAK	
9:45-11:00	Visualizing Data Cont. (customization, arranging, exporting)	<a href="https://datacarpentry.org/R-ecology-lesson/04-visualization-ggplot2.html">https://datacarpentry.org/R-ecology-lesson/04-visualization-ggplot2.html</a>
11:00-11:15	BREAK	
11:00-11:45	Statistical Analysis and ggplot	
11:45-12:00	Wrap up and next steps	



# Data visualization with ggplot

## OVERVIEW

### Questions

---

- How do you make plots using R?
- How do you customize and modify plots?

### Objectives

---

- Produce scatter plots, boxplots, and time series plots using ggplot.
- Set universal plot settings.
- Describe what faceting is and apply faceting in ggplot.
- Modify the aesthetics of an existing ggplot plot (including axis labels and color).
- Build complex and customized plots from data in a data frame.



# Data visualization with ggplot: Key points



## KEY POINTS

- start simple and build your plots iteratively
- the `ggplot()` function initiates a plot, and `geom_` functions add representations of your data
- use `aes()` when mapping a variable from the data to a part of the plot
- use `facet_` to partition a plot into multiple plots based on a factor included in the dataset
- use premade `theme_` functions to broadly change appearance, and the `theme()` function to fine-tune
- the `patchwork` library can combine separate plots into a single figure
- use `ggsave()` to save plots in your favorite format and dimensions



# Next steps

- What is the best way to keep momentum?
  - Follow up courses?
  - Collaboration on small projects?
  - Local meetings with your team?
- Follow up survey
- Lunch!

