Lecture 2: Project Design & Data Visualization

Bill Perry

# Review of Lecture 1

Covered

* Inductive vs deductive reasoning
* Formulating research questions
* Accuracy vs precision
* Data types and classifications
* Setting up R projects
* Installing and loading libraries
* Reading files into R
* Creating basic graphs

# Lecture 2: Project Design & Data Visualization

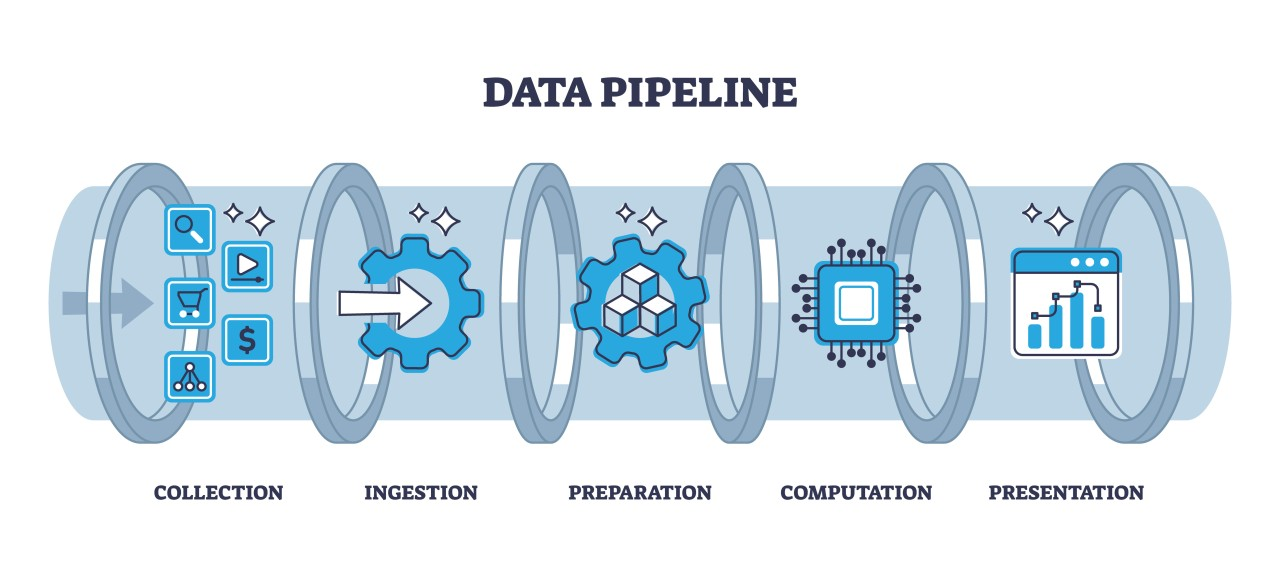
## The objectives:

1. Design a well-organized project
2. Implement good naming conventions
   * Controlled vocabulary
   * Including units in names
3. Create and use metadata effectively
4. Build tidy, well-structured spreadsheets
5. Understand data repositories
6. Create effective visualizations with ggplot2

<>

# Project Design: Step 1

* Data: the raw material of science
* Wide variety of formats, sizes, complexity
* Data management and curation often under emphasized
* Good data management: owe it to our funding agencies, colleagues, supervisors, and study systems

<>

# **Lecture 2:** Project Design: Step 1

1. **Determine data types** you’ll collect
2. **Establish controlled vocabulary**
   * Example: do\_mgl for dissolved oxygen in mg/L
   * Example: drp\_ugl for dissolved reactive phosphorus in μg/L
3. **Plan your data flow** from collection to analysis
4. **Organize your project structure** (folders, files)
5. **Enter data promptly** after collection
6. **Save in multiple formats** (Excel and CSV)
7. **Ensure tidy data principles** from the start

|  |
| --- |
| Note |
| See Hadley Wickham’s [Tidy Data principles](https://r4ds.hadley.nz/data-tidy.html) for best practices |

<>

# Project Design: Step 2

Create a **Metadata Sheet** that includes:

* Variable descriptions
* Units of measurement
* Collection methods
* Instrument details
* Dates and locations
* Any other relevant contextual information

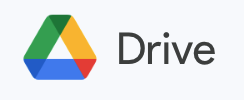
<>

|  |
| --- |
| Practice Exercise 1: Pine Data Organization |
| Let’s examine our pine needle data: - What naming conventions did you choose? - How did you organize the data? - How can you verify data formats (numeric vs categorical)? - What’s your plan for organizing outputs and figures?  # Code to read and examine data library(tidyverse) library(patchwork) library(flextable)  pine\_df <- read\_csv("data/pine\_needles.csv") pine\_df  # A tibble: 48 × 6  date group n\_s wind tree\_no length\_mm  <chr> <chr> <chr> <chr> <dbl> <dbl>  1 3/20/25 cephalopods n lee 1 20  2 3/20/25 cephalopods n lee 1 21  3 3/20/25 cephalopods n lee 1 23  4 3/20/25 cephalopods n lee 1 25  5 3/20/25 cephalopods n lee 1 21  6 3/20/25 cephalopods n lee 1 16  7 3/20/25 cephalopods s wind 1 15  8 3/20/25 cephalopods s wind 1 16  9 3/20/25 cephalopods s wind 1 14 10 3/20/25 cephalopods s wind 1 17 # ℹ 38 more rows |

# **Lecture 2:** Data Management: Step 3

**Storage and Backup Strategy**:

1. Store raw data and metadata securely
   * Save in both Excel and CSV formats
   * Consider write-protecting raw data files
2. Implement the 3-2-1 backup rule:
   * 3 total copies of data
   * 2 different storage media
   * 1 offsite location (cloud storage)
3. Establish a regular backup schedule

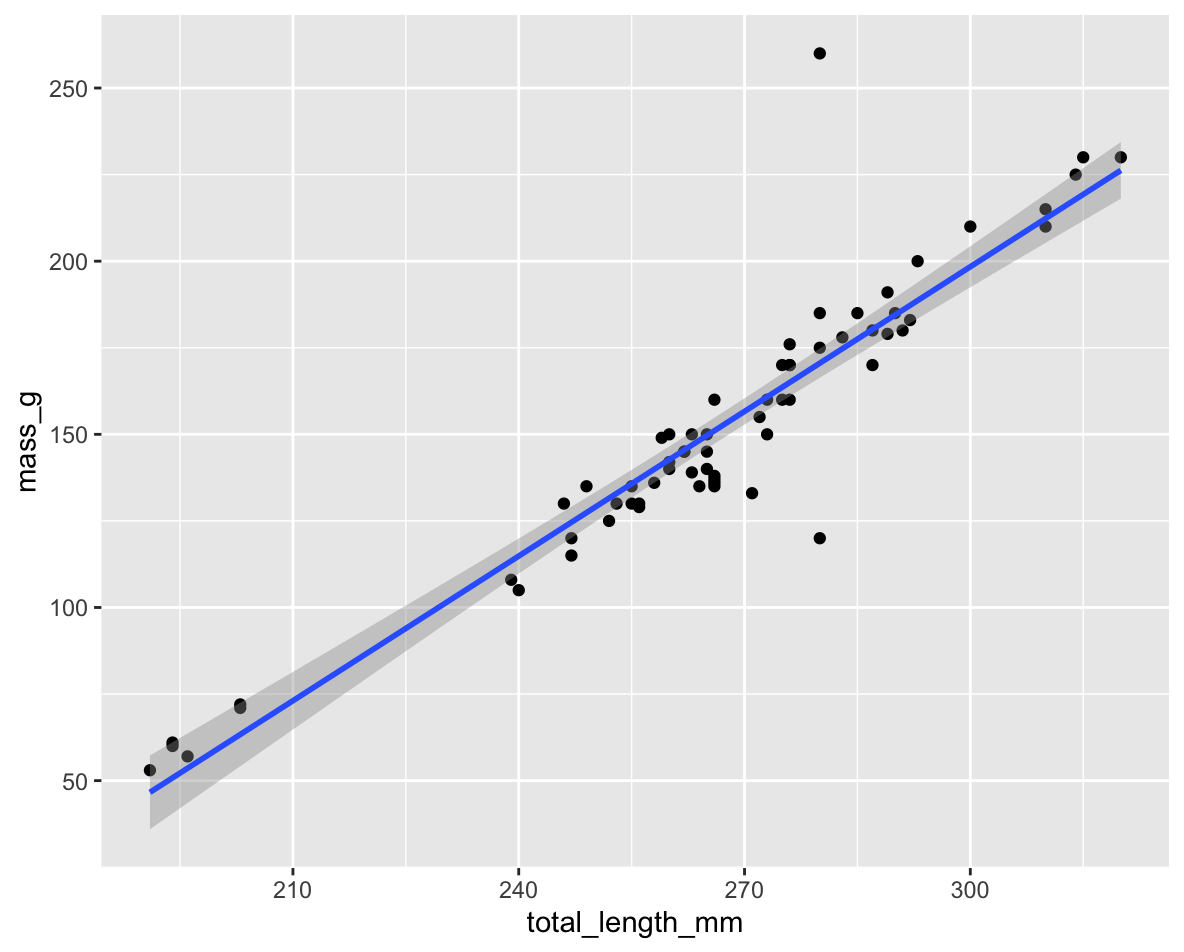
< 

# **Lecture 2:** Data Management: Step 4

**Initial Data Inspection**:

1. Examine data in the Environment tab
2. Run summary() and glimpse() functions
3. Create exploratory visualizations
4. Check for outliers, errors, and missing data

# Specify how to handle missing values during import  
pine\_df <- read\_csv("data/pine\_needles.csv",   
 na = c("", "NA", "N/A", "missing", "null"))  
  
# Get a quick summary  
summary(pine\_df)

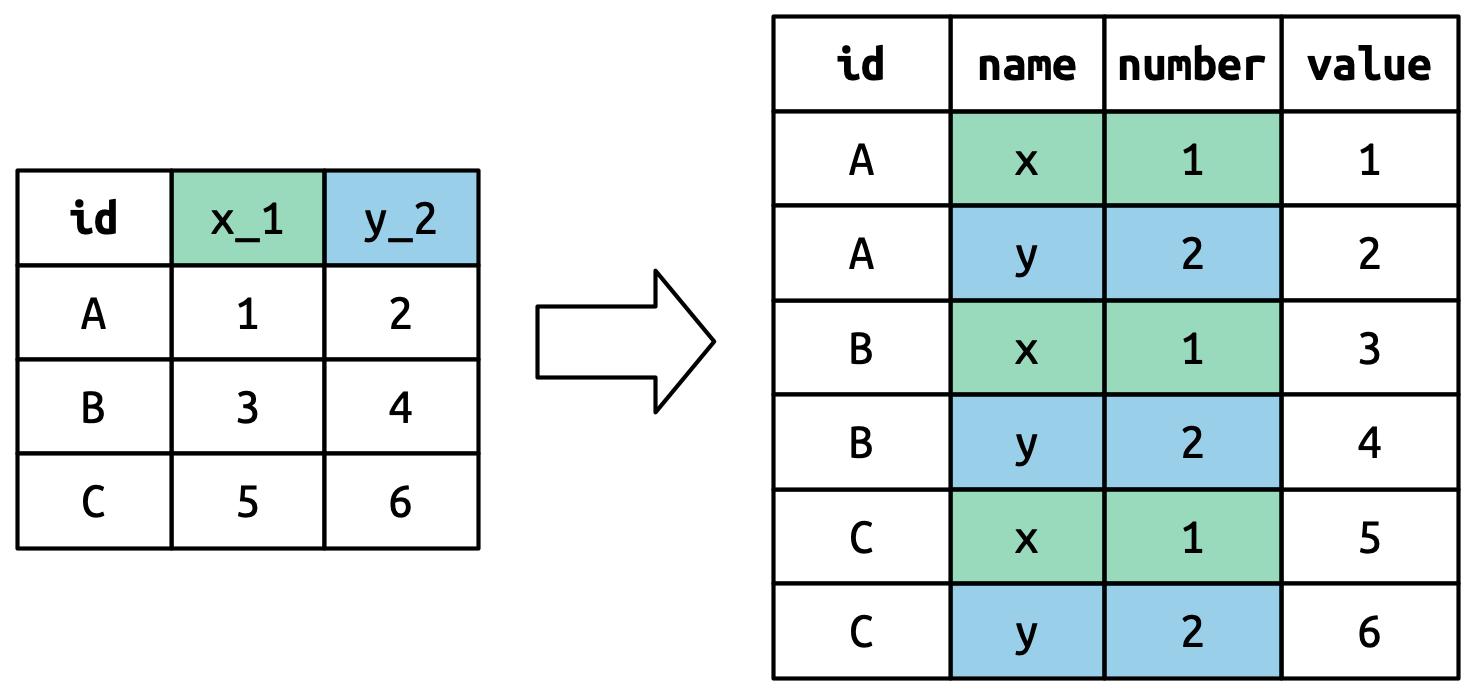


|  |
| --- |
| Practice Exercise 2: Try plotting a histogram of your data |
| Create a histogram of pine needle lengths to check the distribution:  # Write your code here to make a plot # How do you examine the data - what are the ways you think and lets try it! |

# Lecture 2: Data Management: Step 5

**Data Cleaning**:

1. Correct errors and inconsistencies
2. Replace missing values with proper NA codes
3. Document all changes made to raw data
4. Save a clean, master version (consider making read-only)
5. Keep notes on data cleaning procedures

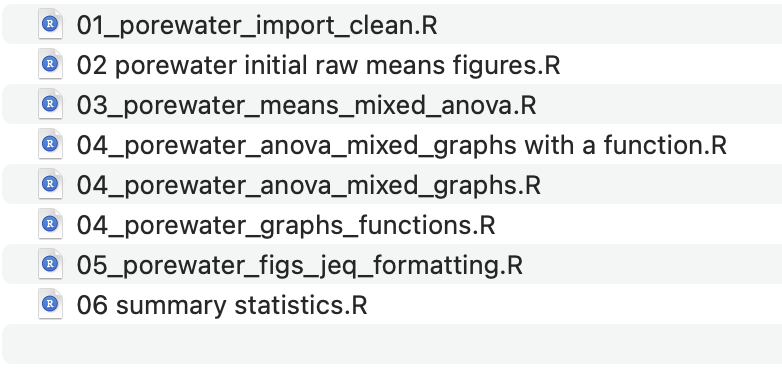


# Lecture 2: Data Management: Step 6

**Analysis and Visualization Workflow**:

1. Create exploratory visualizations
2. Summarize and transform data as needed
3. Document all analysis steps
4. Save outputs systematically

A good way to organize script files is number them in the order they get run.



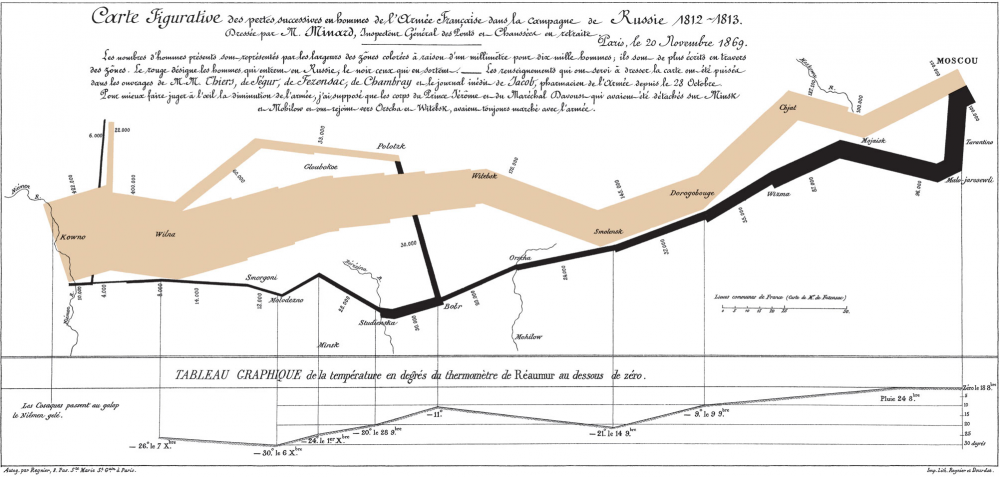
# Lecture 2: Effective Data Visualization

## Why make plots?

## Get in a group and discuss

* What is the purpose of a data visualization?
* What elements are essential in an effective plot?
* What characteristics define a “good” plot?
* What common mistakes make plots ineffective?

[Napoleon’s Disastrous Invasion of Russia Detailed in an 1869 Data Visualization: It’s Been Called “the Best Statistical Graphic Ever Drawn”](https://www.openculture.com/2019/07/napoleons-disastrous-invasion-of-russia-explained-in-an-1869-data-visualization.html)



# **Lecture 2:** Tables vs. Visualizations

**How readable are tables?**

We will get to what these number mean and how to make them in the next lecture.

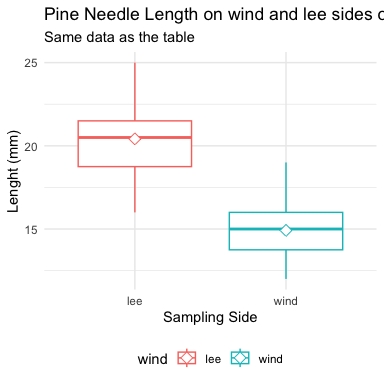
* Tables
  + are they useful in a presentation?

| wind | n | mean\_mm | sd\_mm | min\_mms | max\_mm |
| --- | --- | --- | --- | --- | --- |
| lee | 24 | 20 | 2.45 | 16 | 25 |
| wind | 24 | 15 | 1.91 | 12 | 19 |

# **Lecture 2:** Displaying data

* **how does a table compare to a plot?**
* Does this help?
* What is this plot?
  + if you don’t explain does the audience know?

<



# **Lecture 2:** Principles of Effective Graphics

According to [Tufte (2001)](https://www.edwardtufte.com/book/the-visual-display-of-quantitative-information/), good scientific graphics:

1. **Show the data** without distortion
2. **Maximize data-ink ratio** (minimize non-data elements)
3. **Make large datasets coherent** and understandable
4. **Encourage comparison** between elements
5. **Reveal multiple layers** of information
6. **Serve a clear purpose** in telling your story
7. **Integrate with statistical methods** appropriately

| group | mean\_mm | sd\_mm | n | se\_mm | conf\_low | conf\_high |
| --- | --- | --- | --- | --- | --- | --- |
| cephalopods | 18.00000 | 3.861229 | 12 | 1.1146409 | 15.54669 | 20.45331 |
| crayfish | 18.00000 | 3.861229 | 12 | 1.1146409 | 15.54669 | 20.45331 |
| salmon | 16.33333 | 3.938928 | 12 | 1.1370705 | 13.83066 | 18.83601 |
| snail | 18.33333 | 2.269695 | 12 | 0.6552045 | 16.89124 | 19.77543 |

# **Lecture 2:** Creating Effective Graphics

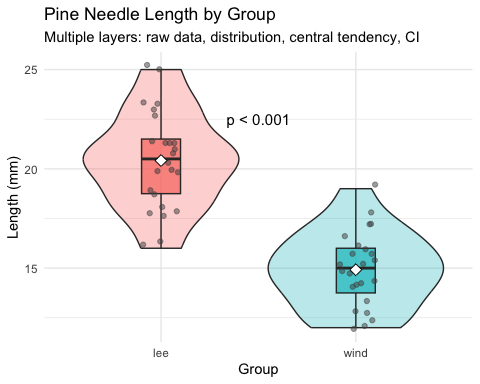
According to [Tufte (2001)](https://www.edwardtufte.com/book/the-visual-display-of-quantitative-information/), good scientific graphics:

* To implement these principles:
  + Focus on the data, not decorative elements
  + Ensure proportional representation of numbers
  + Provide clear and informative labels
  + Remove unnecessary elements (“chart junk”)
  + Revise and refine visualizations iteratively

# **Lecture 2:** Displaying data

To make good graphics:

* Above all, focus on data
* Do not distort data
* Graphical representation of numbers → directly proportional to numbers
* Strive for clarity through labeling
* Maximize data-ink ratio
  + Remove non-data ink
  + Reduce redundant data ink
* Revise and redraw



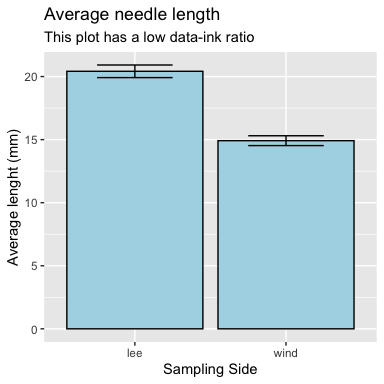
# **Lecture 2:** Displaying data

What do you think?

Does this -

* Focus on data
* Distort data
* Is it directly proportional to numbers
* Is labeling clear
* Maximize data-ink ratio
  + Remove non-data ink
  + Reduce redundant data ink
* Revise and redraw

<



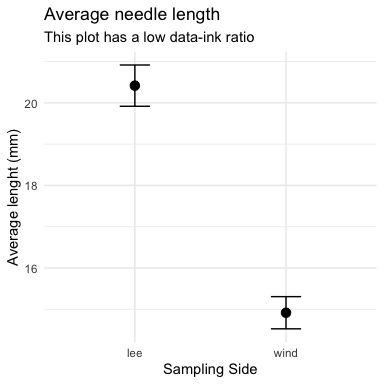
# **Lecture 2:** Displaying data

What do you think?

Does this -

* Focus on data
* Distort data
* Is it directly proportional to numbers
* Is labeling clear
* Maximize data-ink ratio
  + Remove non-data ink
  + Reduce redundant data ink
* Revise and redraw

What is one of the most common plots you make all the time?



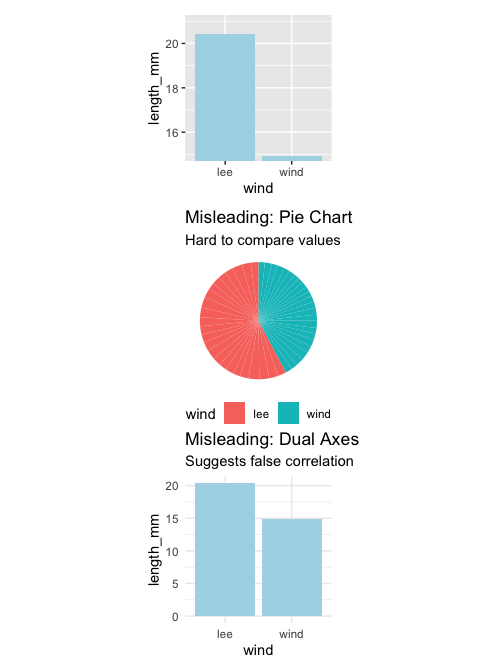
# **Lecture 2:** Displaying data -the bad

## Common Visualization Problems

1. **Data distortion**:
   * Non-zero baselines on bar charts
   * 3D effects that skew perspective
   * Inappropriate scales
2. **Excessive “chart junk”**:
   * Too many gridlines
   * Unnecessary decorative elements
   * Redundant information
3. **Poor color choices**:
   * Too many colors
   * Non-colorblind-friendly palettes
   * Colors that don’t print well in grayscale
4. **Misleading representations**:
   * Pie charts with too many categories
   * Dual y-axes with different scales
   * Truncated axes without clear indication

<

NULL



|  |
| --- |
| Practice Exercise 3: Lets try some plots with pine data first |
| Lets try to make some basic plots  # Write your code here to make a dot plot or X y plot # How do you examine the data - what are the ways you think and lets try it! # what is missing - hwo do you tell the effect of wind? |

|  |
| --- |
| Practice Exercise 4: OK we are closer but what about colors or shape or fills |
| Lets try to make some more basic plots  This is free time - we will free code this….  Below are some examples of code you will need for the future  # Write your code here to make a dot plot or X y plot # How do you examine the data - what are the ways you think and lets try it! # what is missing - hwo do you tell the effect of wind? |

# **Lecture 2:** Introduction to the Grammar of Graphics - ggPLOT

We will learn the anatomy of a GGplot is layers

* ggplot2 uses a **layered grammar of graphics** approach:
  1. **Data**: The dataset you’re visualizing
  2. **Aesthetics**: Mapping variables to visual properties
  3. **Geometries**: The visual elements representing data
  4. **Facets**: Splitting visualization into subplots
  5. **Statistics**: Statistical transformations of the data
  6. **Coordinates**: The space in which data is plotted
  7. **Themes**: Overall visual style of the plotWe have aesthetics

# **Lecture 2:** Building a ggplot Visualization

### Key Components:

1. **Aesthetics (aes)** map variables to visual properties:
   * x and y positions
   * color, fill, shape, size, alpha
   * group, linetype
2. \*\*Geometries (geom\_\*)\*\* determine how data is displayed:
   * geom\_point(): Scatter plots
   * geom\_line(): Line graphs
   * geom\_boxplot(): Box-and-whisker plots
   * geom\_violin(): Violin plots
   * geom\_histogram(): Histograms
   * geom\_bar(): Bar charts
3. **Position adjustments** control how elements are arranged:
   * position\_dodge(): Side-by-side elements
   * position\_jitter(): Add random noise to points
   * position\_stack(): Stack elements on top of each other
4. **Labels and annotations** provide context:
   * labs(): Title, subtitle, caption, axis labels
   * annotate(): Add text, shapes, etc.

# **Lecture 2:** Fine-tuning your visualizations:

1. **Colors, fills, and shapes**:

* scale\_color\_manual(  
   values = c("wind" = "darkblue", "lee" = "darkred"),  
   labels = c("wind" = "Windward", "lee" = "Leeward")  
  )

1. **Coordinate systems**:

* coord\_cartesian(ylim = c(10, 30)) # Zoom in without dropping data

1. **Themes**:

* theme\_minimal() +  
  theme(  
   axis.title = element\_text(size = 14),  
   legend.position = "bottom"  
  )

1. **Combining plots with patchwork**:

* plot1 + plot2 + plot\_layout(ncol = 2)

|  |
| --- |
| Practice Exercise 4: Creating a Publication-Quality Plot |
| Create a fully customized plot that would be suitable for publication:  # Create a publication-quality plot pine\_df %>%  ggplot(aes(x = wind, y = length\_mm, fill = wind)) +  geom\_violin(alpha = 0.4) +  geom\_boxplot(width = 0.2, alpha = 0.7, outlier.shape = NA) +  geom\_jitter(width = 0.1, alpha = 0.5, color = "gray30", size = 2) +  stat\_summary(fun = mean, geom = "point", shape = 23, size = 3, fill = "white") +  labs(  title = "Pine Needle Length Varies with Wind Exposure",  subtitle = "Needles on the leeward side tend to be longer",  x = "Tree Side",   y = "Needle Length (mm)",  caption = "Data collected Spring 2023") +  scale\_fill\_manual(  values = c("wind" = "#1b9e77", "lee" = "#d95f02"),  labels = c("wind" = "Windward", "lee" = "Leeward")) +  theme\_minimal() +  theme(  plot.title = element\_text(face = "bold", size = 16),  plot.subtitle = element\_text(size = 12, color = "gray30"),  axis.title = element\_text(face = "bold"),  legend.title = element\_blank(),  legend.position = "bottom") |

## Key Takeaways

1. **Plan your data management** from the beginning
   * Consistent naming conventions
   * Good organization
   * Regular backups
2. **Make your data tidy** from the start
   * One observation per row
   * One variable per column
   * One value per cell
3. **Create effective visualizations** by:
   * Focusing on data, not decoration
   * Using appropriate plot types
   * Following good design principles
   * Customizing for clear communication
4. **Master the grammar of graphics** to:
   * Build plots layer by layer
   * Communicate patterns clearly
   * Tell compelling stories with data

## Next Steps

* Practice creating different types of plots
* Learn to combine multiple plots effectively
* Explore statistical transformations in ggplot2
* Develop a consistent visualization style