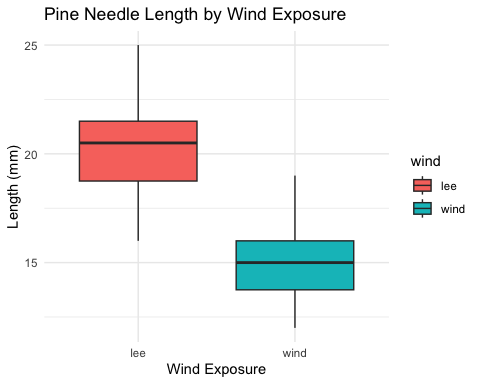
Lecture 9

Bill Perry

# Lecture 8: Review

Covered

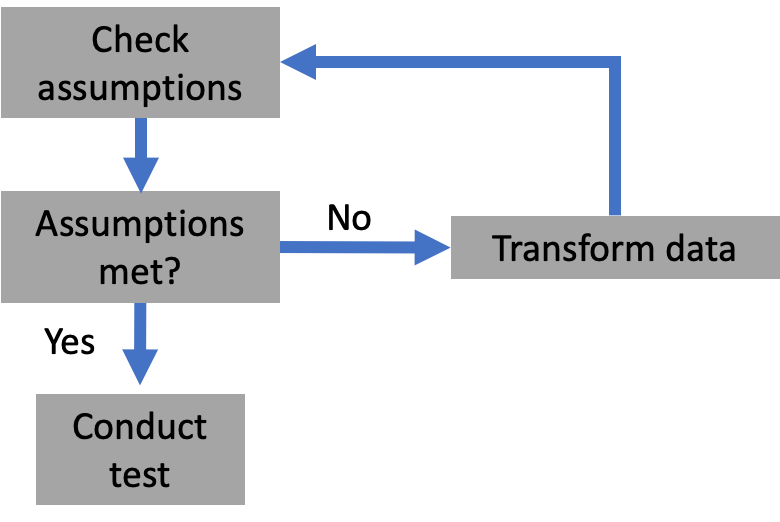
* Decision errors
* Data exploration and transformation
  + Exploratory graphical data analysis
  + Graphical testing of assumptions
  + Data transformation and standardization
  + Outliers
* R practice: robust tests, basic graphics



# **Lecture 9:** Overview

## The objectives:

* Note on test assumptions
* Multiple testing
* Graphics:
  + Why graphics?
  + Rules of good graphics
  + Some bad graphics



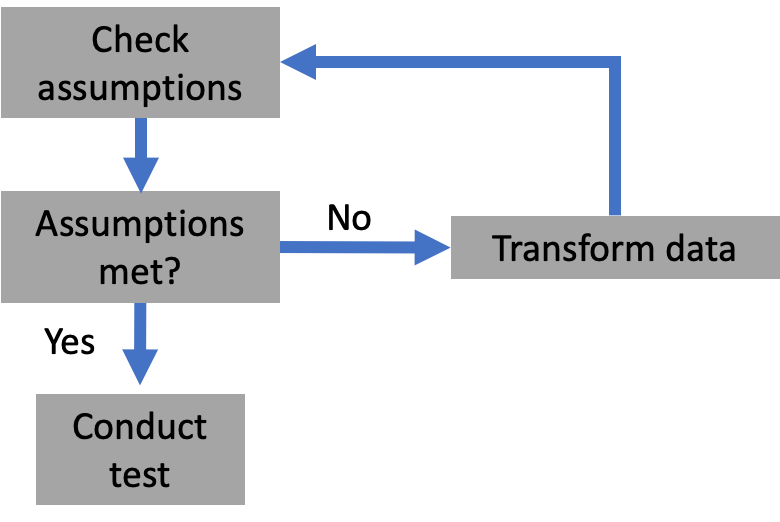
# **Lecture 9:** Overview

Assumption testing: iterative process

If unable to transform: non-parametric approach

When assumptions are violated, we can:

1. 1. Transform data
2. 2. Use robust methods
3. 3. Use non-parametric tests



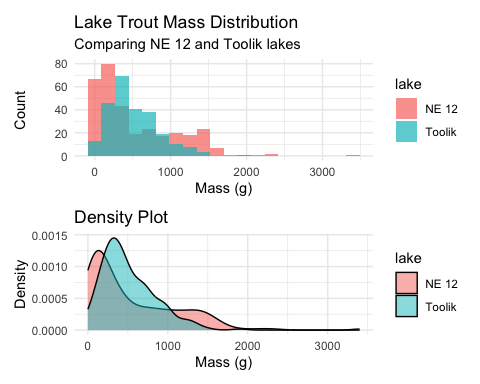
# **Lecture 9:** Overview

Assumption testing: iterative process

If unable to transform: non-parametric approach

When assumptions are violated, we can:

1. 1. Transform data
2. 2. Use robust methods
3. 3. Use non-parametric tests



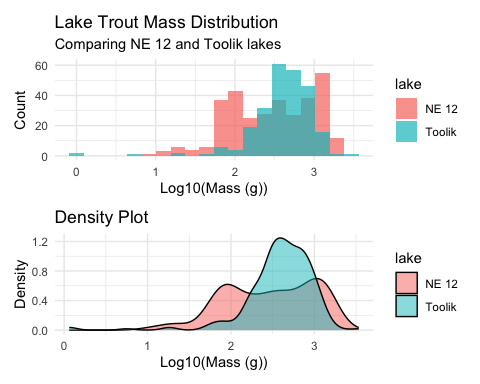
# **Testing assumptions**

Assumption testing: iterative process

If unable to transform: non-parametric approach

When assumptions are violated, we can:

1. 1. Transform data
2. 2. Use robust methods
3. 3. Use non-parametric tests



# **Testing assumptions**

* Assumption testing: iterative process
* If unable to transform: non-parametric approach

# Testing normality assumption on mice weights  
shapiro.test(trout\_data$mass\_g)

Shapiro-Wilk normality test  
  
data: trout\_data$mass\_g  
W = 0.87436, p-value < 2.2e-16

# Testing equality of variances across sampling sites  
# First create a model  
mice\_model <- lm(mass\_g ~ sampling\_site, data = trout\_data)  
# Then test for homogeneity of variances  
car::leveneTest(mice\_model)

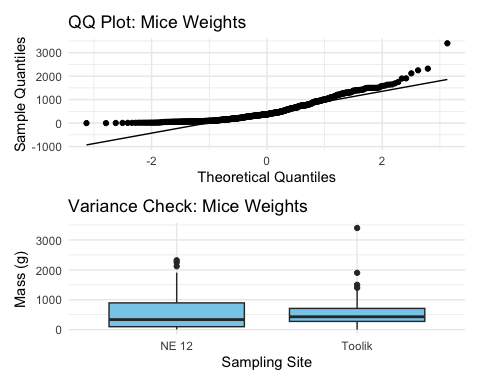
Levene's Test for Homogeneity of Variance (center = median)  
 Df F value Pr(>F)   
group 1 26.352 3.911e-07 \*\*\*  
 569   
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

When assumptions are violated, we can:

1. Transform data

2. Use robust methods

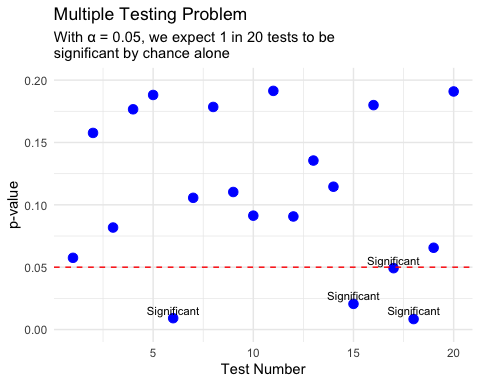
3. Use non-parametric tests



# **Lecture 9:** Overview

## The objectives:

* Note on test assumptions
* Multiple testing
* Graphics:
  + Why graphics?
  + Rules of good graphics
  + Some bad graphics

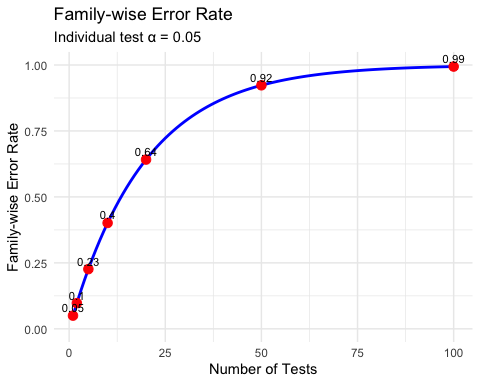


# **Multiple testing**

* Multiple tests: simultaneous tests of related hypotheses on single dataset
  + e.g., 5 pops of snails, are mean sizes different among all groups (1-2, 1-3, 1-5, 2-3, 2-4…)?
* Multiple testing increases possibility of type I error
* If 5% chance of falsely rejecting Ho in 1 test, with each additional test your **“family-wise”** type I error rate increases:
  + 1 test = 0.05
  + 2 tests = 0.098
  + 5 tests = 0.23
  + 20 tests = 0.64

# Function to calculate family-wise error rate  
family\_wise\_error <- function(alpha\_per\_test, num\_tests) {  
 1 - (1 - alpha\_per\_test)^num\_tests  
}  
  
# Create a data frame of family-wise error rates  
error\_rates <- tibble(  
 num\_tests = c(1, 2, 5, 10, 20, 50, 100),  
 error\_rate = family\_wise\_error(0.05, num\_tests)  
)  
  
error\_rates

# A tibble: 7 × 2  
 num\_tests error\_rate  
 <dbl> <dbl>  
1 1 0.0500  
2 2 0.0975  
3 5 0.226   
4 10 0.401   
5 20 0.642   
6 50 0.923   
7 100 0.994

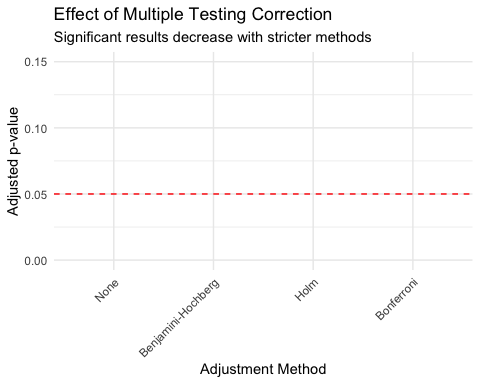


# **Multiple testing adjustments**

* Adjust **family-wise** rate using lower pair-wise rate (α=0.01), but increase type II error rate…
* Common correction methods:
  + **Bonferroni correction**: pairwise α₍ₚₑₚ = α₍fwₑₚ/c
    - For 20 tests, desired α₍fwₑₚ=0.05, α₍ₚₑₚ = 0.0025
  + **Holm-Sidak**: 1 - (1 - α₍fwₑₚ)1/c
    - For 20 tests α₍ₚₑₚ = 0.0026
  + **Sequential Holm**: p-values ranked, smallest tested at α₍fwₑₚ/c (0.005), second at α₍fwₑₚ/(c-1) (0.0055), etc.

# Let's perform multiple t-tests on our mice data  
# Compare mass between each pair of sampling sites  
  
# Get unique sampling sites  
sites <- unique(trout\_data$lake)  
num\_sites <- length(sites)  
num\_comparisons <- num\_sites \* (num\_sites - 1) / 2  
  
# Matrix to store results  
results <- data.frame(  
 comparison = character(num\_comparisons),  
 p\_value = numeric(num\_comparisons),  
 stringsAsFactors = FALSE  
)  
  
# Perform pairwise t-tests  
counter <- 1  
for (i in 1:(num\_sites-1)) {  
 for (j in (i+1):num\_sites) {  
 site\_i\_data <- trout\_data$mass\_g[trout\_data$lake == sites[i]]  
 site\_j\_data <- trout\_data$mass\_g[trout\_data$lake == sites[j]]  
   
 test\_result <- t.test(site\_i\_data, site\_j\_data)  
   
 results$comparison[counter] <- paste(sites[i], "vs", sites[j])  
 results$p\_value[counter] <- test\_result$p.value  
   
 counter <- counter + 1  
 }  
}  
  
# Apply different p-value adjustments  
results$bonferroni <- p.adjust(results$p\_value, method = "bonferroni")  
results$holm <- p.adjust(results$p\_value, method = "holm")  
results$BH <- p.adjust(results$p\_value, method = "BH") # Benjamini-Hochberg  
  
# Display results  
results %>%  
 arrange(p\_value) %>%  
 mutate(across(where(is.numeric), round, 4))

comparison p\_value bonferroni holm BH  
1 NE 12 vs Toolik 0.6718 0.6718 0.6718 0.6718

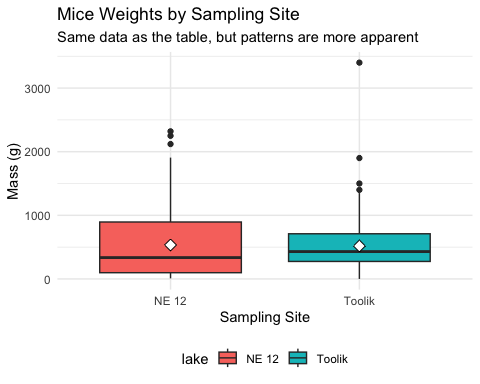


# **Graphics: Why use them?**

* Graphics are visual metaphors for data
* Closest to actual data: table
* But graphics can:
  + Summarize data (means, CVs, R²)
  + Make patterns more apparent
  + Communicate results efficiently
  + Tell a story with the data

# First, let's look at the data as a table  
mice\_summary <- trout\_data %>%  
 group\_by(lake) %>%  
 summarize(  
 n = n(),  
 mean\_mass = mean(mass\_g),  
 sd\_mass = sd(mass\_g),  
 min\_mass = min(mass\_g),  
 max\_mass = max(mass\_g)  
 )  
  
mice\_summary

# A tibble: 2 × 6  
 lake n mean\_mass sd\_mass min\_mass max\_mass  
 <chr> <int> <dbl> <dbl> <dbl> <dbl>  
1 NE 12 322 534. 520. 9 2320  
2 Toolik 249 518. 373. 0.15 3400



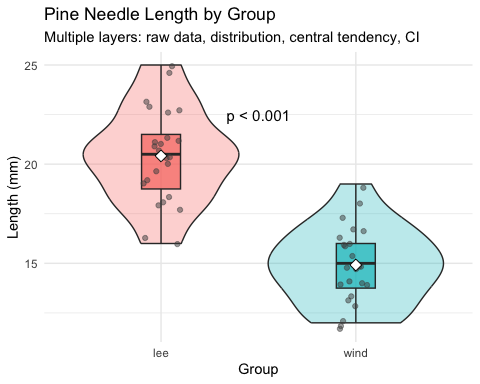
# **Good scientific graphics**

According to Tufte (2001), good scientific graphics:

* Show the data
* Are efficient: show many numbers in small space
* Make large datasets coherent by using appropriate graphic methods
* Encourage comparison
* Reveal several layers of information (e.g., averages, relationships, variability)
* Serve clear purpose: important to telling the main story
* Integrated with statistical methods (e.g., boxplots with t-tests, scatter plots with regression)

# Let's create a plot showing several layers of information  
pine\_summary <- pine\_data %>%  
 group\_by(group) %>%  
 summarize(  
 mean\_length = mean(len\_mm),  
 sd\_length = sd(len\_mm),  
 n = n()  
 ) %>%  
 mutate(se\_length = sd\_length / sqrt(n),  
 conf\_low = mean\_length - qt(0.975, n-1) \* se\_length,  
 conf\_high = mean\_length + qt(0.975, n-1) \* se\_length)  
  
pine\_summary

# A tibble: 4 × 7  
 group mean\_length sd\_length n se\_length conf\_low conf\_high  
 <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl>  
1 cephalopods 18 3.86 12 1.11 15.5 20.5  
2 crayfish 18 3.86 12 1.11 15.5 20.5  
3 salmon 16.3 3.94 12 1.14 13.8 18.8  
4 snail 18.3 2.27 12 0.655 16.9 19.8

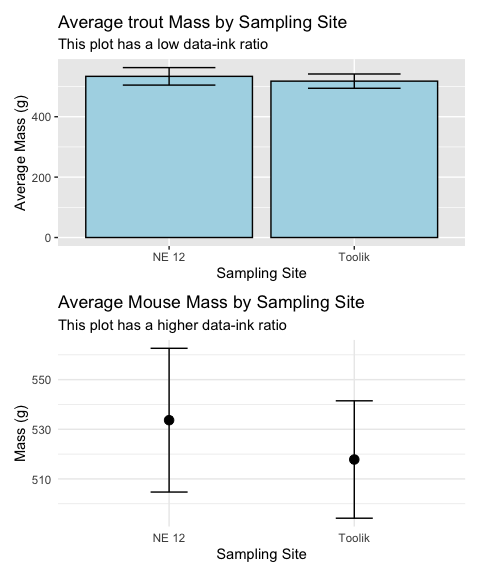


# **Principles of good graphics**

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 labelling
* Maximize data-ink ratio
  + Remove non-data ink
  + Reduce redundant data ink
* Revise and redraw

# Let's create two versions of the same plot  
# First, a "poor" version with low data-ink ratio  
library(ggthemes)  
p1 <- ggplot(trout\_data, aes(x = lake, y = mass\_g)) +  
 geom\_bar(stat = "summary", fun = "mean", fill = "lightblue",   
 color = "black") +  
 geom\_errorbar(stat = "summary", fun.data = "mean\_se", width = 0.5) +  
 # theme\_excel() +  
 labs(title = "Average trout Mass by Sampling Site",  
 subtitle = "This plot has a low data-ink ratio",  
 x = "Sampling Site", y = "Average Mass (g)")



# **Bad graphics examples**

Common problems in graphics:

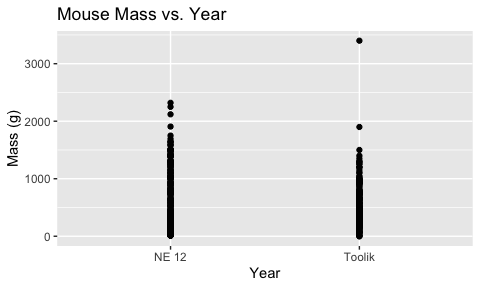
1. **Distorting the data**:
   * Using non-zero baselines for bar charts
   * Using 3D effects that distort perspective
   * Using inappropriate scales
2. **Chart junk**:
   * Excessive gridlines
   * Unnecessary legends
   * Decorative elements that don’t add information
3. **Poor color choices**:
   * Too many colors
   * Non-color-blind friendly palettes
   * Colors that don’t print well in grayscale
4. **Misleading representations**:
   * Pie charts for many categories
   * Dual y-axes with different scales
   * Truncated axes without clear indication



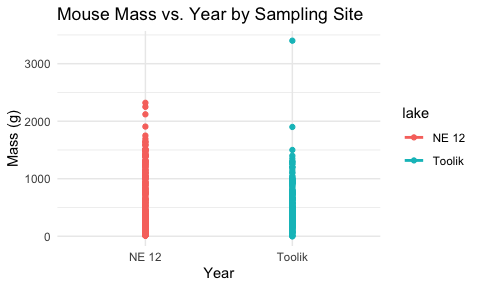
# **R practice: ggplot2**

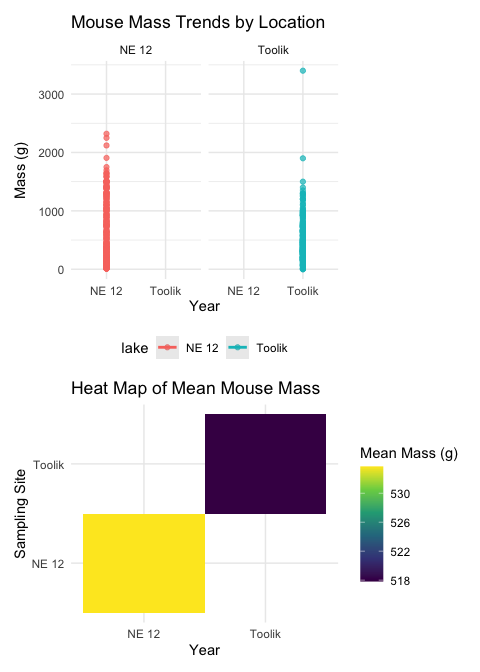
Let’s create some plots with ggplot2 using the mice data:

# Basic scatter plot  
ggplot(trout\_data, aes(x = lake, y = mass\_g)) +  
 geom\_point() +  
 labs(title = "Mouse Mass vs. Year",  
 x = "Year", y = "Mass (g)")



# Scatter plot with grouping and trend line  
ggplot(trout\_data, aes(x = lake, y = mass\_g, color = lake)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", se = FALSE) +  
 labs(title = "Mouse Mass vs. Year by Sampling Site",  
 x = "Year", y = "Mass (g)") +  
 theme\_minimal()





# **Final Activity:** Take home messages

**Key points about multiple testing:**

1. Running multiple tests increases the family-wise error rate
2. Various correction methods exist (Bonferroni, Holm, Benjamini-Hochberg)
3. Choose the appropriate correction based on your research question
4. Report both uncorrected and corrected p-values for transparency

**Principles of good graphics:**

1. Focus on the data, not decoration
2. Maximize data-ink ratio
3. Ensure proportional representation
4. Clear labeling and annotation
5. Choose appropriate visualization for your data type

When applying multiple testing corrections:

* Bonferroni: Most conservative, controls family-wise error rate
* Holm: Less conservative than Bonferroni, still controls FWER
* Benjamini-Hochberg: Controls false discovery rate instead of FWER
* No correction: Highest power, but highest type I error rate

Alternative to multiple pairwise tests: - ANOVA with post-hoc tests - Planned comparisons - Multilevel models

# **Summary and Conclusions**

In this lecture, we’ve:

1. Explored the problem of multiple testing and why it increases type I error rates
2. Learned various methods for correcting p-values in multiple test scenarios
3. Discussed principles of good scientific graphics based on Tufte’s work
4. Identified common pitfalls in data visualization
5. Practiced creating effective visualizations using ggplot2

**Key takeaways:**

* Be cautious when conducting multiple tests on the same dataset
* Apply appropriate corrections to control error rates
* Focus on clear, efficient data visualization that emphasizes the data
* Remove chart junk and maximize the data-ink ratio
* Choose visualization methods that match your research question and data type
* Consider both statistical significance and visual presentation when communicating results

# **What do you see as the key points?**

Things that stood out

1. The dramatic increase in type I error rate with multiple testing
2. The trade-off between type I error control and statistical power
3. The importance of choosing appropriate graphics to communicate your findings
4. How poor visualization choices can mislead readers even when the statistics are correct

# **What are the muddy points?**

What does not make sense or what questions do you have…

What makes you nervous?

1. When to choose which multiple testing correction method
2. How to balance aesthetic appeal with statistical accuracy in graphics
3. Deciding between different visualization types for the same data