Visual Model Summary

Parmeshvar

2025-07-06

Table of contents

1 Introduction

Introduction

DR.Harsh Pradhan, [Phone: +91-9930034241 , Email: harsh.231284@gmail.com], Institute of Management Studies, Banaras Hindu University, Address: 18-GF, Jaipuria Enclave, Kaushambhi, Ghaziabad, India, 2010
Interest: Goal Orientation Job Performance Consumer Behavior Behavioral Finance Bibiliometric Analysis Options as Derivatives Statistics Indian Knowledge System,

Orcid ID
Google Scholar
Youtube ID

Academic Profile

Courses offered:

1. Free online course, four weeks (MOOC), enrollments open: Introduction to Bayesian Data Analysis
2. Short (four-hour) tutorial on Bayesian statistics, taught at EMLAR 2022: here
3. Introduction to (frequentist) statistics
4. Introduction to Bayesian data analysis for cognitive science
5. BDA cover

1.1 Lecture notes

Download from here.

1.2 Moodle website

All communications with students in Potsdam will be done through this website. # Schedule

Week	Lecture	Main Topic	Subtopic	Video	PDF Resource
Week 2	1	Descriptive Statistics	Central Tendency	Video	Week 2.pdf
	2	Descriptive Statistics	Measure of Variability	Video	Same as above
	3	Descriptive Statistics	Describing Data	Video	Same as above
	4	Descriptive Statistics	Probability	Video	Same as above
	5	Descriptive Statistics	Distribution	Video	Same as above
Week 3	1	Descriptive Statistics	Z Table (Normal Distribution)	Video	Week 3.pdf
	2	Descriptive Statistics	Measuring Divergence	Video	Same as above
	3	Inferential Statistics	Sample and Population	Video	Same as above
	4	Inferential Statistics	Model Fit	Video	Same as above
	5	Inferential Statistics	Hypothesis and Error	Video	Same as above
Week 4	1	Terms of Statistics	Terms of Statistics	Video	Week 4.pdf
	2	Terms of Statistics	T-Test	Video	Same as above
	3	Terms of Statistics	T-Test in Detail	Video	Same as above
	4	ANOVA	ANOVA	Video	Same as above
Week 5	1	ANOVA	Example of ANOVA	Video	Week 5.pdf
	2	ANOVA	Types of ANOVA	Video	Same as above

Week	Lecture	Main Topic	Subtopic	Video	PDF Resource
	3	Correlation	Introduction to Correlation	Video	Same as above
	4	Correlation	Regression (Part 1)	Video	Same as above
	5	Correlation	Regression (Part 2)	Video	Same as above
Week 6	1	Correlation	R Script for Regression	Video	Week 6.pdf
	2	Chi Square	Chi Square	Video	Same as above
	3	Chi Square	Chi Square Test	Video	Same as above
	4	Logistic Function	Regression Function	Video	Same as above
	5	Logistic Function	Distribution	Video	Same as above
Week 7	1	Time Series	Intro to Time Series	Video	Week 7.pdf
	2	Time Series	Conditional Probability	Video	Same as above
	3	Time Series	Additional Concepts	Video	Same as above
	4	Time Series	Distribution	Video	Same as above
	5	Time Series	Poisson Distribution	Video	Same as above
	6	Index Numbers	Price & Quantity Index	Video	Same as above
	7	Decision Environments	Risk/Uncertainty, Bayes, Trees	Video	Same as above
	8	Time Series Analysis	Components, Trend, Seasonality	Video	Same as above
	9	Time Series Analysis	Least Squares Method	Video	Same as above
Week 8	1	Effect Size & Documentation	Package/Library	Video	Week 8.pdf

Week	Lecture	Main Topic	Subtopic	Video	PDF Resource
	2	Effect Size & Documentation	RStudio vs RKward	Video	Same as above
	3	Effect Size & Documentation	Flexplot	Video	Same as above
	4	Effect Size & Documentation	Functions	Video	Same as above
	5	Effect Size & Documentation	R Shiny & R Markdown	Video	Same as above
	6	Effect Size & Documentation	Application with Real Datasets	Video	Same as above
	7	Effect Size & Interpretation	Importance in Testing	Video	Same as above
	8	Effect Size & Interpretation	Installing dplyr, ggplot2	Video	Same as above
	9	Effect Size & Interpretation	Visual Model Interpretation	Video	Same as above
	10	Effect Size & Interpretation	Creating/Using Functions	Video	Same as above
	11	Effect Size & Interpretation	Report, Dashboard, Interactivity	Video	Same as above

2 Week 1

2.1 Module 1: Introduction to Statistics

2.1.1 Pre-Requisites

- Just an open and eager mind
- Basic understanding of Mathematics or Statistics

2.1.2 Agenda

- Meaning of Statistics
- Nature and Scope
- Uses of Statistics
- Limitations
- Fallacies and Misuse
- Math vs Statistics
- GUI Tools & Transition to Software-based Stats

2.1.3 Meaning of Statistics

Statistics is a science which provides tools for **analysis and interpretation** of raw data collected for decision-making in diverse fields.

It includes four core concepts:

- **Population** – Complete data or total group
- **Sample** – Subset of population
- **Parameter** – Numerical summary from population
- **Statistic** – Numerical summary from sample

2.1.4 Nature of Statistics

- Deals with **numerical facts**
- Focused on **social phenomena** and real-world data
- Organizes, classifies, and analyzes data
- Facilitates **prediction**, **interpretation**, and **decision-making**

2.1.5 Uses of Statistics

- Drawing representative samples
- Summarizing collected data
- Tabulation and systematic arrangement
- Group comparisons
- Determining behavioral relationships
- Estimating chance vs causation
- Application in:
  - Psychology
  - Education
  - Employment surveys
  - Market Research
  - Industrial and Organizational studies

2.1.6 Limitations of Statistics

- Cannot study **qualitative phenomena** without quantification
- Not applicable to individuals
- **Statistical laws are not exact**
- Does not guarantee **causal relationships**
- Vulnerable to misuse

2.2 Misuse of Statistics

- Use of extremely **small or biased** samples
- **Misleading graphs** or visual misrepresentation
- Illogical or **unexpected comparisons**

Fallacies in Statistics

Fallacies may arise from:

- Poor data collection methods
- Vague or manipulated term definitions
- Improper unit selection

- Faulty classification or grouping
- Inappropriate statistical methods

2.3 Module 2: Mathematics vs Statistics

Aspect	Mathematics	Statistics
Nature	Abstract, symbolic reasoning	Applied, data-based reasoning
Focus	Pure logic, proofs	Real-world data, decision-making
Techniques	Algebra, Calculus, Geometry	Probability, Hypothesis testing, Regression
Output	Theorems, functions, formulas	Inferences, predictions, summaries
Tools	Equations, graphs	Charts, tables, models

2.4 Module 3: Software-Based Statistical Revolution

From Paper to Code

Why shift to software?

- **Faster analysis** of massive data
- **Error-free calculations**
- **Anywhere-anytime** access
- **Cloud-based integration**
- Supports **ML/AI**, automation, and deep visualization

2.4.1 Popular Statistical Software

Software	Type	Use Case
R	Script	Core for academic and professional stats
RKWard	GUI	GUI wrapper for R
R Commander	GUI	Menu-based GUI for R
Rattle	GUI	Data mining toolkit in R
Excel	GUI	Basic stats with plugins
Python (pandas)	Script	Modern data science + ML

2.4.2 GUI vs CLI

Feature	GUI (e.g., RKWard)	Command Line (e.g., R Console)
Accessibility	User-friendly	Requires learning syntax
Speed	Slower for heavy tasks	High performance
Learning Curve	Minimal	Moderate to High

Feature	GUI (e.g., RKWard)	Command Line (e.g., R Console)
Customization	Limited	Fully scriptable
Teaching Utility	Good for beginners	Good for understanding logic

2.4.3 Recommended GUI Tools for R

- **RKWard**
- **Rattle**
- **R Commander**
- **R AnalyticFlow**

  https://rkward.kde.org

2.4.4 Installing RKWard on Ubuntu

bash sudo apt install kbibtex kate libcurl4-openssl-dev libssl-dev libxml2-dev cmake sudo add-apt-repository ppa:rkward-devel/rkward-stable echo "deb https://ppa.launchpad.net/rkward-devel/rkward-stable/ubuntu jammy main" | sudo tee /etc/apt/sources.list.d/rkward.list sudo apt update sudo apt-get install rkward Awesome. Here's Part 2 of the full markdown, Lines 251–600, continuing the structured content from your Week 1 lecture.

2.5 Module 4: Understanding Variables

2.5.1 What is a Variable?

A **variable** is a characteristic or attribute that can assume different values across individuals or items.

In statistics, variables are categorized for analysis and measurement.

2.5.2 R Definition:

In R, variables are containers for data, created by assignment:

x <- 10 name <- "Harsh" flag <- TRUE

Classification of Variables

A. Qualitative (Categorical)

Type Description Example

Nominal Categories without order Gender (Male, Female) Ordinal Categories with a meaningful order Education Level (UG, PG)

B. Quantitative (Numerical)

Type Description Example

Discrete Countable numbers No. of students Continuous Infinite values in a range Height, Weight

Statistical Data Types (Scale of Measurement)

Data Type Description Examples

Nominal Categories with no order Blood group (A, B, AB, O) Ordinal Ranked categories Satisfaction (Low, Med, High) Interval Numeric scale with no true zero Temperature in Celsius Ratio Numeric scale with true zero Income, Weight, Age

Data Types in R

R Type Description Example Code

Numeric Real numbers x <- 15.3 Integer Whole numbers y <- as.integer(10) Complex Real + imaginary z <- 2+3i Character Text strings c <- "hello" Logical Boolean values b <- TRUE Factor Categorical encoding factor(c("yes", "no", "yes"))

2.6 Examples in R

x <- 15.6 y <- as.integer(18) z <- 7 + 5i c <- "I am OK" b <- TRUE

Module 5: Data Structures in R

Vectors

A vector is a one-dimensional array of elements.

vec1 <- c(5, 2, 3, 7, 8, 9, 1, 4, 10, 15)

Matrices

Two-dimensional arrays of rows and columns.

mat <- matrix(1:9, nrow=3, ncol=3)

Arrays

Multidimensional generalization of matrices.

arr <- array(1:24, dim=c(3,4,2))

Lists

Collection of different types of elements.

mylist <- list(name="Alice", age=30, scores=c(89,90))

Data Frames

Tabular data (like a spreadsheet), each column can have a different type.

df <- data.frame(ID=1:3, Name=c("A", "B", "C"), Score=c(85, 90, 95))

Factors

Used for categorical variables.

gender <- factor(c("Male", "Female", "Male"))

Module 6: Descriptive Statistics

Descriptive statistics summarize and simplify data.

Central Tendency

Measure Formula Meaning

Mean $\bar{x} = \frac{\sum x_i}{n}$ Average Median Middle value in sorted data Central observation Mode Most frequent value Most common observation

Dispersion Measures

Measure Formula Purpose

Range $Range = Max - Min$ Spread of data Variance $s^2 = \frac{\sum(x_i - \bar{x})^2}{n-1}$ Spread from mean Standard Deviation $s = \sqrt{Variance}$ Average distance from mean

Example in R

x <- c(10, 20, 30, 40, 50) mean(x) median(x) var(x) sd(x)

Module 7: Inferential Statistics

Inferential stats allow us to make conclusions about populations using samples.

Key Concepts

Hypothesis Testing: Assesses assumptions about a population.

Confidence Intervals: Estimate population parameters within a range.

Significance Levels ( ): Commonly 0.05 or 5%

P-Value: Probability of observing the data assuming the null is true.

Hypothesis Types

Type Description

Null Hypothesis No difference / no effect Alternative There is a difference / effect

R Examples

t.test(x) # One-sample t-test t.test(x, y) # Two-sample t-test

Module 8: Visualizing Data

Data visualization helps uncover patterns and insights.

Boxplot

Shows 5-number summary

Identifies outliers

boxplot(x)

Histogram

Frequency distribution of continuous data

hist(x)

Pie Chart

Shows proportion in categories

slices <- c(10, 12, 4, 16, 8) labels <- c("A", "B", "C", "D", "E") pie(slices, labels=labels)

Scatter Plot

Relationship between two variables

plot(x, y)

Ogive (Cumulative Frequency)

2.7 Create cumulative frequency table manually

Module 9: Spreadsheet Basics

Spreadsheets like Excel or Google Sheets are entry points for data work.

Key Features:

Rows → Observations

Columns → Variables

Supports sorting, filtering

Built-in formulas: =SUM(), =AVERAGE(), etc.

Spreadsheets vs R

Feature Spreadsheet (Excel, GSheets) R / RKWard

Cost Usually licensed Free and open source Flexibility Limited to GUI formulas Full programming capability Graphics Basic Advanced (ggplot2) Reproducibility Low High (script-based)

Module 10: Command Line vs GUI

Command Line (R Console)

2.8 Windows Command Line

cd .. mkdir new_folder dir

R Console Commands

getwd() setwd("path") install.packages("ggplot2") library(ggplot2)

GUI (RKWard)

Point-and-click interface

No coding needed

View script history and console

Menu for graphs, models, tables

Learning Resources:

Books

Mohanty, B., & Misra, S. (2016). Statistics for Behavioural and Social Sciences

Pandya et al. (2018). Statistical Analysis in Simple Steps using R

Field, A. P. et al. (2012). Discovering Statistics using R

Harris, J. K. (2019) . Statistics with R: Solving Problems using Real-World Data

2.9 Utilizing Statistical Methods for Decision Making

- Use statistical evidence to guide business strategies.
- Make informed policy decisions based on empirical data.
- Report findings clearly for transparency and comprehension.

2.10 Summary

The "Basic Statistics Using GUI-R (RK Ward)" course equips learners with the foundational and practical skills needed for statistical analysis using R. Students will understand theoretical concepts, grasp practical applications, and use RKWard effectively to analyze real-world data.

2.11 Key Takeaways

- Proficiency in defining and using variables and data types.
- Capability to import and manipulate data in RKWard.
- Understanding of basic statistical practices and their applications.
- Skill in visualizing data for effective communication of results.

2.12 Websites

https://rkward.kde.org https://r4stats.com https://cran.r-project.org

3 Week 2

3.1 Introduction

3.1.1 Purpose of the eBook

This eBook is designed as a complete beginner-to-intermediate guide for understanding the foundational concepts of statistics. It aims to bridge theoretical knowledge and practical application using RKWard (a GUI for R). Readers will be introduced to descriptive and inferential statistics, probability theory, and probability distributions with ample examples and exercises.

3.1.2 Who Should Read This?

- Undergraduate students
- MBA and management students
- Data analysis beginners
- Professionals dealing with data

3.1.3 What You'll Learn

- Data classification and types
- Descriptive statistics: central tendency and variability
- Basic probability and events
- Probability distributions: Bernoulli, Binomial, and Normal
- Use of RKWard in statistical analysis

3.2 1. Fundamentals of Statistics

3.2.1 1.1 What is Statistics?

Statistics is the science of collecting, organizing, analyzing, and interpreting data to make informed decisions. It involves both **theoretical** (mathematical) and **applied** approaches to understanding uncertainty and variability in real-world phenomena.

3.2.2 1.2 Key Objectives

- Summarizing large datasets effectively
- Estimating population parameters
- Testing hypotheses
- Making predictions and decisions under uncertainty

3.2.3 1.3 Types of Statistics

- **Descriptive Statistics:** Deals with the presentation and summarization of data.
- **Inferential Statistics:** Draws conclusions about populations based on sample data.

3.3 2. Types of Data

3.3.1 2.1 Classification of Data

Type	Example	Description
Qualitative	Gender, Nationality	Non-numeric labels
Quantitative	Height, Age	Numeric values
Discrete	No. of Children	Countable numbers
Continuous	Temperature, Weight	Infinite values in a range

3.3.1.1 Qualitative (Categorical) Data

- **Nominal:** No inherent order (e.g., religion, marital status).
- **Ordinal:** Natural order (e.g., customer satisfaction: Poor, Average, Good).

3.3.1.2 Quantitative (Numerical) Data

- **Discrete:** Integers; e.g., number of books.
- **Continuous:** Measurable; e.g., weight in kilograms.

3.4 3. Descriptive Statistics

3.4.1 3.1 Measures of Central Tendency

3.4.1.1 What is Central Tendency?

Central tendency refers to the center or middle of a dataset. It's the value that best represents the entire distribution.

3.4.1.2 Characteristics of a Good Measure

- Rigidly defined
- Easy to understand
- Takes all data into account
- Amenable to algebraic treatment
- Stable under sampling
- Minimally affected by outliers (except mean)

3.4.2 3.2 The Mean

3.4.2.1 Definition

The arithmetic mean is the sum of all values divided by the number of values.

3.4.2.2 Formula

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

3.4.2.3 Properties of Mean

- Uses all data values
- Affected by extreme values
- The sum of deviations from the mean is zero

3.4.2.4 Example

Data: 10, 15, 20, 25, 30
Mean $= (10 + 15 + 20 + 25 + 30)/5 = 20$

3.4.3 3.3 The Median

3.4.3.1 Definition

The median is the value separating the higher half from the lower half of a data sample.

3.4.3.2 Calculation

- Odd number of items: Middle value
- Even number of items: Average of the two middle values

3.4.3.3 Properties

- Not influenced by extreme values
- Best for skewed data

3.4.3.4 Example

Data: 4, 6, 9, 12, 15, 21, 33
Median = 12 (middle value)

3.4.4 3.4 The Mode

3.4.4.1 Definition

The mode is the value that appears most frequently in a dataset.

3.4.4.2 Characteristics

- Can be used for categorical data
- Dataset can be unimodal, bimodal, or multimodal
- May not exist if all values are unique

3.4.4.3 Example

Data: 4, 4, 6, 8, 9, 10, 4
Mode = 4

3.4.5 3.5 Comparison Table

Measure	Use Case	Affected by Outliers	Mathematical Use
Mean	Symmetric distributions	Yes	High
Median	Skewed distributions	No	Moderate
Mode	Categorical variables	No	Low

3.5 4. Measures of Variability

3.5.1 4.1 Why Measure Variability?

While central tendency summarizes data, variability tells us how spread out the data is. It's essential in determining consistency and reliability.

3.5.2 4.2 Range

3.5.2.1 Definition

The difference between the maximum and minimum values.

$$\text{Range} = x_{\text{max}} - x_{\text{min}}$$

3.5.2.2 Example

Data: 12, 14, 17, 19, 23
Range = 23 - 12 = 11

3.5.2.3 Limitations

- Ignores distribution shape
- Extremely sensitive to outliers

3.5.3 4.3 Quartiles and Interquartile Range

3.5.3.1 Quartiles

- Q1 (25th percentile): Lower quartile
- Q2 (50th percentile): Median
- Q3 (75th percentile): Upper quartile

3.5.3.2 Formula for Position

$$Q_k = \frac{k(n+1)}{4}$$

3.5.3.3 IQR Formula

$$IQR = Q3 - Q1$$

3.5.3.4 Example

Data: 12, 30, 45, 57, 70
Q1 = 30, Q3 = 57 → IQR = 27

3.5.4 4.4 Variance

3.5.4.1 Concept

Variance is the average of the squared differences from the Mean.

3.5.4.2 Formulas

Population Variance:

$$\sigma^2 = \frac{1}{N} \sum (x_i - \mu)^2$$

Sample Variance:

$$s^2 = \frac{1}{n-1} \sum (x_i - \bar{x})^2$$

3.5.5 4.5 Standard Deviation

3.5.5.1 Concept

Standard deviation is the square root of variance. It provides a measure of spread in the same units as the data.

$$s = \sqrt{s^2}$$

3.5.5.2 Properties

- Same unit as original data
- Measures how far values deviate from the mean
- Widely used in most statistical computations

3.5.6 4.6 Coefficient of Variation (CV)

3.5.6.1 Definition

The ratio of the standard deviation to the mean, expressed as a percentage. Used to compare variability between datasets with different units.

$$CV = \left(\frac{s}{\bar{x}}\right) \times 100\%$$

3.5.6.2 Example

Dataset A: Mean = 100, SD = 10 → CV = 10%
Dataset B: Mean = 50, SD = 5 → CV = 10%

3.5.7 4.7 Moment-Based Measures

- First Moment (about mean): 0 (since $\sum(x - \bar{x}) = 0$)
- Second Moment: Variance
- Third Moment: Skewness
- Fourth Moment: Kurtosis

3.6 5. Probability Fundamentals

3.6.1 5.1 Introduction to Probability

Probability is the mathematical framework for quantifying uncertainty. It helps us estimate how likely an event is to occur.

3.6.2 5.2 Key Definitions

- **Experiment:** A process that leads to an outcome.
- **Outcome:** The result of an experiment.
- **Sample Space (Ω):** All possible outcomes.
- **Event:** A subset of the sample space.

3.6.3 5.3 Types of Events

Event Type	Description
Independent	Occurrence of one does not affect the other
Dependent	One affects the outcome of another
Mutually Exclusive	Cannot occur together
Exhaustive	Includes all possible outcomes

3.6.4 5.4 Classical Probability

Used when all outcomes are equally likely.

Formula:

$$P(A) = \frac{\text{Number of favorable outcomes}}{\text{Total outcomes in } \Omega}$$

Example: Rolling a fair die
P(rolling a 3) = 1/6

3.6.5 5.5 Probability Rules

3.6.5.1 Rule 1: Non-Negativity

$$0 \leq P(A) \leq 1$$

3.6.5.2 Rule 2: Total Probability

$$P(\Omega) = 1$$

3.6.5.3 Rule 3: Complement Rule

$$P(A^c) = 1 - P(A)$$

3.6.5.4 Rule 4: Addition Rule

If A and B are mutually exclusive:

$$P(A \cup B) = P(A) + P(B)$$

Otherwise:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

3.6.5.5 Rule 5: Multiplication Rule

- For independent events:

$$P(A \cap B) = P(A) \cdot P(B)$$

3.6.6 5.6 Conditional Probability

Formula:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

3.7 6. Discrete Probability Distributions

3.7.1 6.1 Bernoulli Distribution

- One trial, two outcomes (success/failure).
- Success = 1, Failure = 0

$$P(X = x) = p^x(1-p)^{1-x}, \quad x \in \{0, 1\}$$

- Mean = p
- Variance = $p(1-p)$

3.7.1.1 Example:

Flip a fair coin → p = 0.5
Mean = 0.5, Variance = 0.25

3.7.2 6.2 Binomial Distribution

- Series of n independent Bernoulli trials
- Number of successes x out of n trials

Formula:

$$P(X = x) = \binom{n}{x} p^x(1-p)^{n-x}$$

- Mean: $\mu = np$
- Variance: $\sigma^2 = np(1-p)$

3.7.2.1 Example:

Flip a coin 5 times (p = 0.5)
P(X = 3) = $\binom{5}{3}(0.5)^3(0.5)^2 = 10 \cdot 0.125 \cdot 0.25 = 0.3125$

3.8 7. Continuous Distributions

3.8.1 7.1 Normal Distribution

The most important continuous distribution in statistics.

Properties:

- Bell-shaped and symmetric
- Defined by mean ( ) and variance (^{2})
- Total area under the curve = 1

Probability Density Function (PDF):

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

3.8.1.1 Empirical Rule:

- 68% of values lie within ±1
- 95% within ±2
- 99.7% within ±3

3.8.2 7.2 Standard Normal Distribution

A normal distribution with:

- Mean = 0
- Standard deviation = 1

Z-score Formula:

$$Z = \frac{X-\mu}{\sigma}$$

3.8.2.1 Example:

If $\mu = 100$, $\sigma = 15$, and $X = 130$
Then $Z = \frac{130-100}{15} = 2$

3.9 8. Visualizing Data

3.9.1 8.1 Frequency Distribution

Class Interval	Frequency
0–10	3
11–20	7
21–30	9
31–40	6

3.9.2 8.2 Histogram

A bar chart representing the frequency distribution of numerical data.

Use Case: Visualize shape (e.g., normal, skewed)

3.9.3 8.3 Boxplot (Box-and-Whisker Plot)

Shows:

- Minimum
- Q1
- Median
- Q3
- Maximum
- Outliers (as dots)

Helps identify skewness and outliers quickly.

3.9.4 8.4 Scatter Plot

Used to study the relationship between two quantitative variables.

3.10 9. Practical Applications

3.10.1 9.1 Business Use Cases

- Retail: Analyze sales patterns
- Healthcare: Patient outcome probabilities
- Finance: Stock volatility (using SD, CV)

3.10.2 9.2 Education and Research

- Student test scores: Use mean, SD, and percentile ranking
- Experiment analysis: Use Z-scores and Normal Distribution

3.11 10. Using RKWard

3.11.1 10.1 What is RKWard?

A graphical frontend for the R programming language designed for statistical analysis and data visualization.

3.11.2 10.2 Installation Guide

1. Download R from CRAN
2. Install RKWard from rkward.kde.org
3. Start RKWard and begin with menu-driven tasks

3.11.3 10.3 Sample RKWard Activities

3.11.3.1 Calculate Mean and SD

- Load dataset
- Click *Statistics* → *Descriptive Statistics*
- Choose variables and click *OK*

3.11.3.2 Visualize Histogram

- Click *Graphics* → *Histogram*
- Select variable and customize bins

3.11.4 10.4 Using R Code in RKWard

data <- c(12, 15, 17, 18, 21) mean(data) sd(data) hist(data)

3.12 Summary

This eBook provided a deep dive into basic statistics including:

Data types and classification Central tendency and variability Probability theory and rules Discrete and continuous distributions Visual interpretation and real-world applications GUI-based statistical analysis using RKWard

4 Week 3

4.1 Introduction

4.1.1 Importance of Statistics

Statistics is a powerful tool used across disciplines — from economics and psychology to biology, data science, and machine learning. It enables:

- Interpretation of data
- Generalization from samples to populations
- Hypothesis testing and decision-making
- Prediction and modeling

Understanding statistics is essential for anyone involved in **empirical research**, **policy making**, **data-driven decision-making**, or **scientific inquiry**.

4.1.2 Overview of Topics

This book covers:

- Population vs Sample
- Hypotheses and Errors
- Descriptive vs Inferential Statistics
- Data Types (R + Theoretical)
- Sampling Techniques
- Normal Distribution
- Linear and Logistic Regression
- GUI-based R interfaces: RKWard, Rcmdr, Rattle
- Fallacies and misuse in statistics
- Graphical Methods
- R programming constructs for statistics

4.2 Understanding Populations and Samples

4.2.1 Population

The complete set of all units of interest. Examples:

- All students in India
- All electric cars in the U.S.

4.2.2 Sample

A **subset of the population**, selected for analysis. Goal: represent the population accurately.

4.2.3 Why Use Samples?

- More practical and cost-efficient
- Enables faster analysis
- Allows estimation and inference

4.2.4 Relation Between Population & Sample

Population → Sample → Statistic → Inference → Population Parameter

---

4.3 Hypotheses and Errors

4.3.1 Hypothesis Defined

A hypothesis is a testable assumption about a population.

4.3.1.1 Null Hypothesis (H_0)

- No difference or effect
- Example: H_0: " = 100"

4.3.1.2 Alternative Hypothesis (H_A)

- A difference or effect exists
- Example: H_A: " 100"

4.3.2 Types of Errors

Error Type	Description
Type I Error	Rejecting H_0 when it's true (false positive)
Type II Error	Failing to reject H_0 when it's false (false neg)

4.3.3 Significance Level ( )

The probability of making a Type I error — commonly set to **0.05 (5%)**

4.4 Inferential Statistics

4.4.1 Purpose

- Estimate unknown population parameters
- Test hypotheses
- Predict outcomes

4.4.2 Common Techniques

- t-test
- z-test
- ANOVA
- Chi-square
- Regression

4.4.3 Sampling Techniques

4.4.3.1 1. Simple Random Sampling

Every unit has equal probability.

4.4.3.2 2. Systematic Sampling

Pick every kth element.

4.4.3.3 3. Stratified Sampling

Subdivide population into strata (e.g. age groups), then sample from each.

4.4.3.4 4. Cluster Sampling

Randomly choose entire groups (e.g. schools, cities).

4.4.4 Central Limit Theorem (CLT)

If $n > 30$, the distribution of sample means approximates a **normal distribution** even if the original population is not normal.

Formula:

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$$

4.5 Descriptive Statistics

4.5.1 Measures of Central Tendency

4.5.1.1 Mean

$$\bar{x} = \frac{\sum x_i}{n}$$

4.5.1.2 Median

Middle value in an ordered dataset.

4.5.1.3 Mode

Most frequent value.

4.5.2 Measures of Dispersion

4.5.2.1 Range

$$Range = Max - Min$$

4.5.2.2 Variance

$$s^2 = \frac{\sum(x_i - \bar{x})^2}{n-1}$$

4.5.2.3 Standard Deviation

$$s = \sqrt{s^2}$$

4.5.3 Measures of Shape

- **Skewness**: Degree of asymmetry
- **Kurtosis**: Peakedness of distribution

4.6 Graphical Methods

4.6.1 Histogram

r hist(data$height, col="blue", main="Height Distribution") Boxplot

boxplot(data$score \ data$group) Scatter Plot

plot(data$x, data$y, col="red") Ogive (Cumulative Frequency Plot)

Built using cumulative frequency of class intervals.

4.6.2 R Data Types and Structures

Basic Data Types

x <- 12.5 # numeric y <- as.integer(5) # integer z <- 4 + 3i # complex name <- "Ravi" # character flag <- TRUE # logical Vectors

v <- c(1, 2, 3) Matrices

m <- matrix(1:9, nrow=3, byrow=TRUE) Data Frame

df <- data.frame(Name=c("A", "B"), Score=c(89, 94)) Lists

lst <- list(id=101, name="John", marks=c(78, 82)) Factors

gender <- factor(c("Male", "Female", "Male")) Statistical Fallacies

What are Fallacies?

Fallacies occur when conclusions are drawn based on flawed statistical reasoning.

Common Fallacies

Improper Sampling Misleading Graphs Ambiguous Term Definitions Ignoring Confounding Variables Assuming Correlation Implies Causation Misuse of Statistics

Examples of Misuse

Using biased samples Cherry-picking data Using 3D pie charts to exaggerate results Misrepresenting scale in graphs

4.6.3 Comparing R vs Excel vs GUI-R (RKWard)

Feature	R (Script)	Excel	RKWard GUI
Usability	Medium	Easy	Easy
Flexibility	High	Low-Medium	Medium
Statistical Power	Very High	Low	High
Graphics	ggplot2	Basic	ggplot2 supported
Reproducibility	High	Low	High

4.6.4 Installing RKWard (Ubuntu)

$sudoaptinstallkbibtexkatelibcurl4-openssl-devlibssl-devlibxml2-devcmakesudoadd-apt-repositoryppa : rku$

4.6.5 Teaching Tools in RKWard

$install.packages(c("R2HTML", "car", "e1071", "Hmisc", "plyr", "ggplot2", "prob", "ez", "multcomp", "remotes"), de$

4.6.6 GUI-Based Statistical Tools

RKWard – KDE interface for R Rcmdr – Classic R Commander GUI Rattle – Data mining GUI in R R AnalyticFlow – Flow-based programming for statistics

4.7 Linear Regression in R

4.7.1 What is Linear Regression?

Linear regression models the relationship between a **dependent variable (Y)** and one or more **independent variables (X)**.

4.7.1.1 Simple Linear Regression Equation:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Where:

- Y is the dependent variable
- X is the independent variable
- β_0 is the intercept
- β_1 is the slope
- ϵ is the error term

4.7.2 Code Example

r ## Load data data(mtcars)

4.8 Fit model

model <- lm(mpg ~ wt, data=mtcars)

4.9 Summary

summary(model)

4.9.1 Adjusted R-squared

Penalizes the number of predictors to avoid overfitting.

AIC & BIC

AIC: Akaike Information Criterion BIC: Bayesian Information Criterion Lower values of AIC/BIC $\rightarrow$ better model fit (with penalty for complexity).

4.9.2 Normal Distribution

Key Properties

Symmetrical, bell-shaped curve Mean = Median = Mode Total area under curve = 1 Empirical Rule: 68% within ±1 SD 95% within ±2 SD 99.7% within ±3 SD

Example: Given: Mean = 70, SD = 5, X = 75

z <- (75 - 70) / 5 # Result: 1.0 Z-Table Usage

Find the area under the curve to the left of the z-score Useful for probability and percentile ranking

4.9.3 Data Import Techniques

CSV Import in R

df <- read.csv("data.csv", header=TRUE) head(df) Excel Import (using readxl)

install.packages("readxl") library(readxl)

df <- read_excel("data.xlsx")

4.9.4 Working with the RKWard Interface

Sections: Console – Run R code Script Editor – Write reusable code Workspace – View loaded variables Teaching Tab – Education-focused modules

4.9.5 Spreadsheet Concepts

Structure

Component |Description Rows |Individual observations Columns |Variables Cells |Data points Header Row| Variable names

4.9.6 Advantages

Easy data entry Visual inspection Good for small datasets

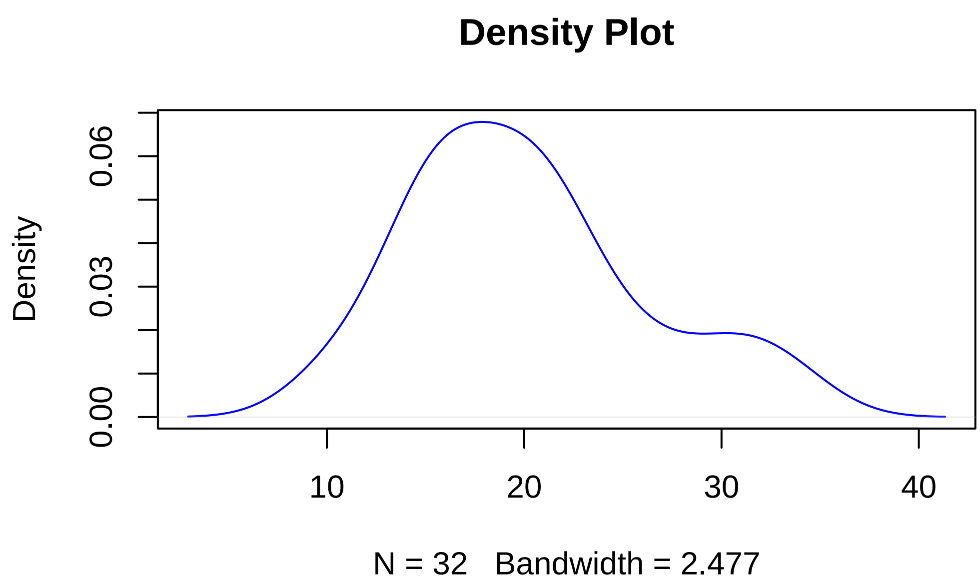
4.9.7 Limitations

Limited statistical functionality Hard to reproduce Error-prone for large datasets

4.9.8 Advanced Plots and Techniques

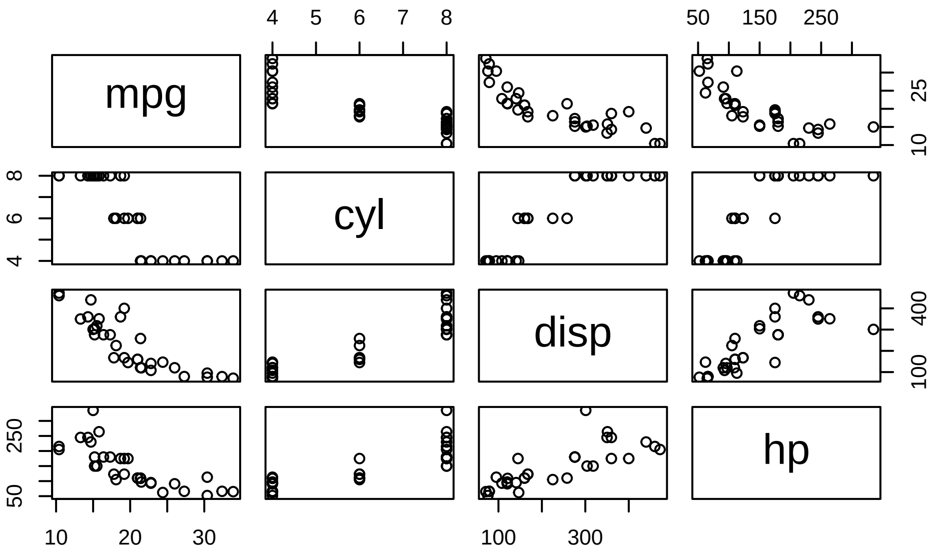
Density Plot

```
plot(density(mtcars$mpg), main="Density Plot", col="blue")
```



Pair Plot

```
pairs(mtcars[, 1:4])
```



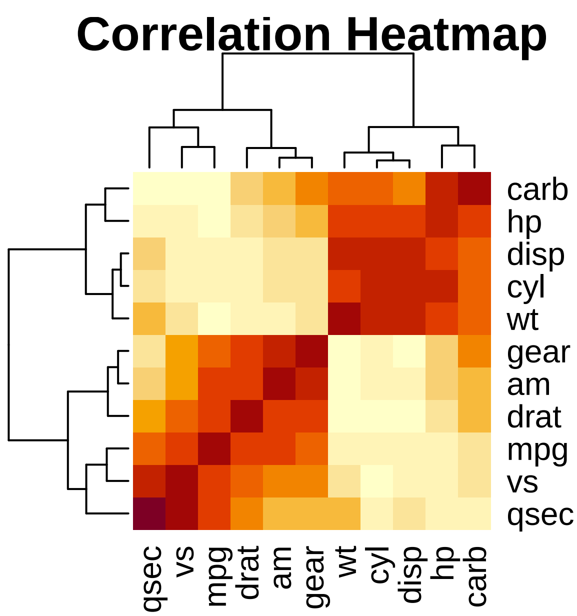
Correlation Matrix

```r
cor(mtcars)
```

```
            mpg        cyl       disp         hp        drat         wt
mpg   1.0000000 -0.8521620 -0.8475514 -0.7761684  0.68117191 -0.8676594
cyl  -0.8521620  1.0000000  0.9020329  0.8324475 -0.69993811  0.7824958
disp -0.8475514  0.9020329  1.0000000  0.7909486 -0.71021393  0.8879799
hp   -0.7761684  0.8324475  0.7909486  1.0000000 -0.44875912  0.6587479
drat  0.6811719 -0.6999381 -0.7102139 -0.4487591  1.00000000 -0.7124406
wt   -0.8676594  0.7824958  0.8879799  0.6587479 -0.71244065  1.0000000
qsec  0.4186840 -0.5912421 -0.4336979 -0.7082234  0.09120476 -0.1747159
vs    0.6640389 -0.8108118 -0.7104159 -0.7230967  0.44027846 -0.5549157
am    0.5998324 -0.5226070 -0.5912270 -0.2432043  0.71271113 -0.6924953
gear  0.4802848 -0.4926866 -0.5555692 -0.1257043  0.69961013 -0.5832870
carb -0.5509251  0.5269883  0.3949769  0.7498125 -0.09078980  0.4276059
            qsec         vs          am       gear        carb
mpg   0.41868403  0.6640389  0.59983243  0.4802848 -0.55092507
cyl  -0.59124207 -0.8108118 -0.52260705 -0.4926866  0.52698829
disp -0.43369788 -0.7104159 -0.59122704 -0.5555692  0.39497686
hp   -0.70822339 -0.7230967 -0.24320426 -0.1257043  0.74981247
drat  0.09120476  0.4402785  0.71271113  0.6996101 -0.09078980
wt   -0.17471588 -0.5549157 -0.69249526 -0.5832870  0.42760594
qsec  1.00000000  0.7445354 -0.22986086 -0.2126822 -0.65624923
vs    0.74453544  1.0000000  0.16834512  0.2060233 -0.56960714
am   -0.22986086  0.1683451  1.00000000  0.7940588  0.05753435
gear -0.21268223  0.2060233  0.79405876  1.0000000  0.27407284
carb -0.65624923 -0.5696071  0.05753435  0.2740728  1.00000000
```

Heatmap

```
heatmap(cor(mtcars), main="Correlation Heatmap")
```



4.9.9 Common R Packages for Statistics

Package |Purpose ggplot2 |Data visualization dplyr |Data manipulation tidyr |Data tidying Hmisc |Misc stats functions car |Regression diagnostics e1071 |Skewness/kurtosis, ML tools psych |Psychological statistics shiny |Interactive apps caret |Classification and regression

4.9.10 Introduction to Command Line

4.10 Windows Terminal

$$cd..mkdirmy_projectdir$$

Linux Terminal

$$cd\ mkdirstats_projectls - l$$

4.10.1 Git + R Project Example

$$gitinitgitclonehttps : //github.com/username/project.git$$

4.10.2 Fallacies and Bias: Real-World Cautions

Examples of Statistical Abuse

Cherry-picking data Data dredging (p-hacking) Using relative risk without absolute context Non-random sampling Ethics in Data Analysis

Be transparent Document sources Disclose methodology Avoid overstating conclusions

4.10.3 Future Applications of Statistics

Real-World Domains

Healthcare: Drug effectiveness, diagnostics Economics: Forecasting, policy evaluation Sociology: Survey analysis Sports: Performance analytics AI/ML: Predictive modeling, optimization Next Steps

Learn tidyverse ecosystem Explore machine learning in R Build Shiny dashboards Get familiar with reproducible research using Quarto

4.10.4 Practice Challenges

1. Load and summarize data

Load mtcars or your own dataset Use summary(), mean(), sd() 2. Create 3 different plots

Histogram Boxplot by group Scatter plot with trend line 3. Build a regression model

Identify predictor and outcome Use lm() and summary() 4. Explore a GUI like RKWard or Rcmdr

4.10.5 Key Takeaways

Statistics supports informed decision-making. R and its GUI frontends offer flexibility + power. Understand theory → then automate with code. Avoid fallacies by following robust methods. Visuals are crucial: plot early, plot often.

5 Week 4

5.1 Introduction

This eBook is a comprehensive companion to the course *Basic Statistics using GUI-R (RKWard)*. It includes foundational theory, practical examples, and step-by-step explanations, with integrated GUI-R usage.

5.2 Course Overview

5.2.1 Course Name

Basic Statistics using GUI-R (RKWard)

5.2.2 Instructor Profile

Dr. Harsh Pradhan is Assistant Professor at the Institute of Management Studies, Banaras Hindu University.
Faculty Profile

5.2.3 Learning Objectives

- Understand core concepts in statistics
- Apply t-tests and ANOVA using real data
- Compute confidence intervals and test statistics
- Use GUI-R (RKWard) for statistical analysis

5.3 Chapter 1: Fundamental Concepts

5.3.1 Descriptive Statistics

5.3.1.1 Central Tendency

- **Mean**
- **Median**

- **Mode**

5.3.1.2 Dispersion

- **Range**
- **Variance**
- **Standard Deviation**

5.3.1.3 Example:

```r
data <- c(4, 8, 6, 5, 3)
mean(data)
```

```
[1] 5.2
```

```r
median(data)
```

```
[1] 5
```

```r
sd(data)
```

```
[1] 1.923538
```

5.3.2 Standard Error

$$SE = \frac{s}{\sqrt{n}}$$

Small SE = sample mean is a good estimate of the population mean.

5.3.3 Central Limit Theorem

For $n > 30$, sampling distribution of the mean approximates normal:

$$\bar{X} \sim \mathcal{N}(\mu, \frac{\sigma}{\sqrt{n}})$$

5.3.4 Confidence Intervals

$$CI = \bar{x} \pm Z \cdot \frac{s}{\sqrt{n}}$$

Interpret 95% CI as: 95 of 100 such intervals would contain the true mean.

5.4 Chapter 2: Estimation

5.4.1 Types of Estimates

Type	Description	Example
Point Estimate	Single value	Sample mean
Interval Estimate	Range + confidence	Confidence Int

5.4.2 Parameter vs Statistic

Term	Description
Parameter	Value from population (e.g., μ)
Statistic	Value from sample (e.g., $\bar{x}$)

5.5 Chapter 3: Hypothesis Testing

- **Null Hypothesis** (H_0): No effect
- **Alternative Hypothesis** (H_1): Some effect
- **Type I Error**: Reject H_0 when true
- **Type II Error**: Fail to reject H_0 when false

5.6 Chapter 4: Student's T-Test

5.6.1 Types

Test Type	Description
One-Sample	Compare sample to fixed value
Independent	Compare two unrelated groups
Paired	Compare two related groups

5.6.2 One-Sample T-Test Example

```r
data <- c(22, 24, 27, 26, 28, 23, 25, 29, 21, 26, 24, 27)
t.test(data, mu = 25)
```

```

	One Sample t-test

data:  data
t = 0.2363, df = 11, p-value = 0.8175
alternative hypothesis: true mean is not equal to 25
95 percent confidence interval:
 23.61427 26.71906
sample estimates:
mean of x
 25.16667
```

5.6.3 Test Statistic

$$t = \frac{\bar{x} - \mu}{SE}$$

5.6.4 Degrees of Freedom

$$df = n - 1$$

5.6.5 Decision Rule

Compare calculated t to table value. If $|t| > t_{critical}$, reject H_0.

5.6.6 T-Test in GUI-R

1. Import data
2. Choose T-Test
3. Define groups
4. Run & interpret output

5.7 Chapter 5: ANOVA

5.7.1 Purpose

Used when comparing means across 3+ groups.

5.7.1.1 One-Way ANOVA Formula

$$F = \frac{MS_{between}}{MS_{within}}$$

Where:

- $MS_{between} = \frac{SS_{between}}{df_{between}}$
- $MS_{within} = \frac{SS_{within}}{df_{within}}$

5.7.1.2 Assumptions

- Normality
- Homogeneity of variance
- Independence

5.7.1.3 Example Table

Group	Mean	Var	n
A	5.5	1.5	30
B	7.1	2.0	30
C	6.8	1.8	30

5.7.2 Post-Hoc Tests

Run if ANOVA is significant to locate pairwise differences.

5.7.3 ANOVA in GUI-R

1. Load data
2. Choose "One-Way ANOVA"
3. Define groups
4. Interpret output

5.8 Chapter 6: GUI-R Workflow

1. **Import Data** (CSV, Excel)
2. **Choose Test** (T-Test, ANOVA, etc.)
3. **Run** the analysis
4. **Interpret** the output
5. **Export** the results or visualizations

5.9 Chapter 7: Advanced Concepts

5.9.1 Variance Partitioning

$$\text{Total Variance} = \text{Explained Variance} + \text{Unexplained Variance}$$

Explained Terms	Unexplained Terms
Systematic	Random
Predictive	Error
Deterministic	Noise

5.9.2 Degrees of Freedom

For equation $x + y + z = 3$, if 2 values are known, third is fixed.
Hence, $df = n - k$ where n = total variables, k = constraints.

5.9.3 Chi-Square and F Distribution

- **Chi-Square**: Categorical variable comparison
- **F-Distribution**: Used in ANOVA, variance testing

5.9.4 Univariate, Bivariate, Multivariate

Type	Variables	Example
Univariate	1	Height
Bivariate	2	Height vs Weight
Multivariate	>2	Study w/ Age, Gender, Income

5.9.5 Parametric Test Assumptions

- Interval/Ratio DV
- Random Sampling
- Normality
- Equal Variances

If assumptions violated → use non-parametric test.

5.9.6 Effect Size

$$\text{Effect Size} = \frac{|\mu_1 - \mu_2|}{\sigma}$$

Used for comparison across studies.

5.9.7 Power of a Test

$$\text{Power} = 1 - \beta$$

Higher power → lower chance of Type II error
Power increases with sample size, effect size

5.10 Conclusion

Statistics is the language of data. GUI-R makes statistical tools accessible for everyone. This book empowers you to analyze data effectively using t-tests, ANOVA, and confidence intervals in a GUI environment.

5.11 References

- Pradhan, H. (2023). *Basic Statistics using GUI-R (RKWard)*
- Field, A. (2013). *Discovering Statistics Using R.*
- https://methods.sagepub.com

5.12 Chapter 8: Advanced T-Test Applications

5.12.1 Paired Sample T-Test

Used when the same group is measured twice (e.g., before and after).

5.12.1.1 Example:

```r
before <- c(80, 82, 79, 84, 88)
after <- c(78, 81, 76, 83, 86)
t.test(before, after, paired = TRUE)
```

```
    Paired t-test

data:  before and after
t = 4.8107, df = 4, p-value = 0.008581
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 0.7611494 2.8388506
sample estimates:
mean difference
            1.8
```

5.12.2 Independent Samples T-Test

Compare means of two unrelated groups.

```r
group1 <- c(85, 90, 88, 92, 87)
group2 <- c(80, 83, 85, 84, 82)
t.test(group1, group2)
```

```
    Welch Two Sample t-test

data:  group1 and group2
t = 3.7755, df = 7.226, p-value = 0.006537
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 2.114814 9.085186
sample estimates:
mean of x mean of y 
     88.4      82.8 
```

5.12.3 One-Sample T-Test with GUI-R

- Import dataset
- Use 'Descriptive Statistics' to check mean
- Navigate to 'T-Test' → 'One Sample'
- Input hypothesized mean and run

5.13 Chapter 9: More on Confidence Intervals

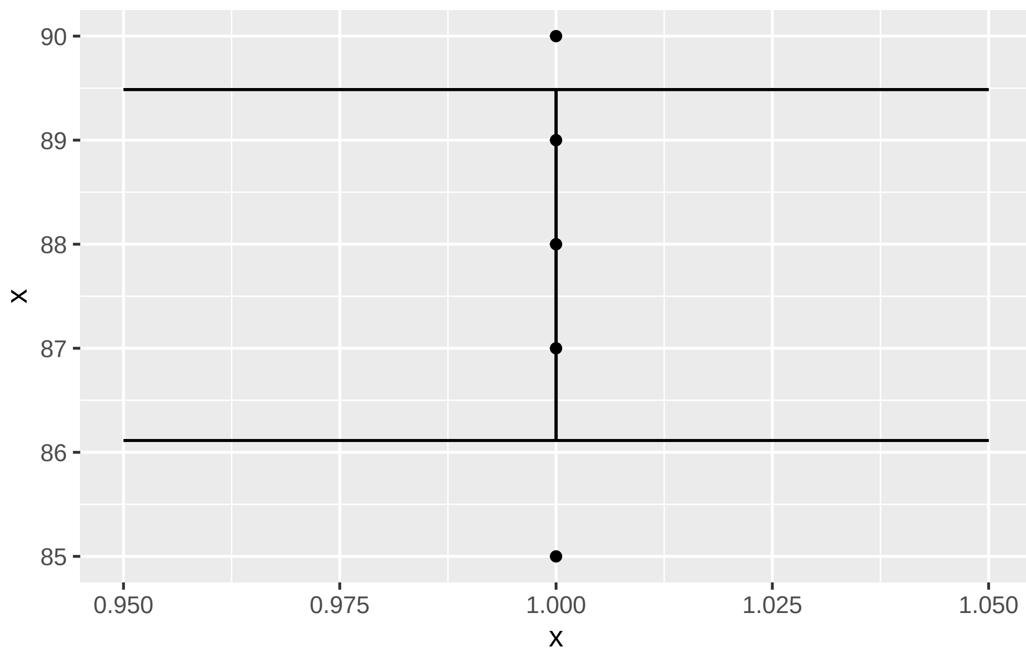
5.13.1 Visualizing Confidence Intervals in R

```
x <- c(88, 90, 85, 87, 89)
mean_x <- mean(x)
se <- sd(x) / sqrt(length(x))
ci_lower <- mean_x - 1.96 * se
ci_upper <- mean_x + 1.96 * se
c(ci_lower, ci_upper)
```

```
[1] 86.11394 89.48606
```

Plot using `ggplot2`:

```
library(ggplot2)
df <- data.frame(x = x)
ggplot(df, aes(y = x, x = 1)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_lower, ymax = ci_upper), width = 0.1)
```



5.14 Chapter 10: Robust ANOVA Models

5.14.1 Two-Way ANOVA

Examines the effect of two categorical independent variables on a continuous dependent variable.

```
# Sample dataset for demonstration
dataset <- data.frame(
  score = c(85, 90, 88, 92, 87, 80, 83, 85, 84, 82),
  gender = rep(c("Male", "Female"), each = 5),
  teaching_method = rep(c("A", "B"), times = 5)
)
aov_result <- aov(score ~ gender * teaching_method, data = dataset)
summary(aov_result)
```

```
                       Df Sum Sq Mean Sq F value  Pr(>F)   
gender                  1  78.40   78.40  23.718 0.00279 **
teaching_method         1   6.02    6.02   1.820 0.22598   
gender:teaching_method  1  18.15   18.15   5.491 0.05759 . 
Residuals               6  19.83    3.31                   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.14.2 Repeated Measures ANOVA

Use when the same subjects are used for each treatment.

```
# Sample repeated measures data in long format
data_long <- data.frame(
  id = rep(1:5, each = 3),
  condition = rep(c("A", "B", "C"), times = 5),
  score = c(85, 88, 90, 80, 82, 85, 78, 80, 83, 90, 92, 95, 88, 90, 91)
)
library(ez)
ezANOVA(data = data_long, dv = .(score), wid = .(id), within = .(condition))
```

```
Warning: Converting "id" to factor for ANOVA.
```

```
Warning: Converting "condition" to factor for ANOVA.
```

```
$ANOVA
     Effect DFn DFd        F            p p<.05       ges
2 condition   2   8 88.22222 3.539139e-06     * 0.1479687

$`Mauchly's Test for Sphericity`
     Effect         W         p p<.05
2 condition 0.5555556 0.4140867

$`Sphericity Corrections`
     Effect       GGe        p[GG] p[GG]<.05       HFe        p[HF] p[HF]<.05
2 condition 0.6923077 9.135419e-05         * 0.9411765 6.568851e-06         *
```

5.15 Chapter 11: Effect Size Measures

5.15.1 Cohen's d

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s_p}$$

Where s_p is the pooled standard deviation.

5.15.1.1 R Example

```
library(effsize)
cohen.d(group1, group2)
```

```
Cohen's d

d estimate: 2.387848 (large)
95 percent confidence interval:
    lower     upper
0.4791634 4.2965327
```

5.15.2 Eta-Squared (η^2)

Used for ANOVA:

$$\eta^2 = \frac{SS_{between}}{SS_{total}}$$

5.16 Chapter 12: Statistical Assumptions Checking

5.16.1 Normality

Use Shapiro-Wilk test:

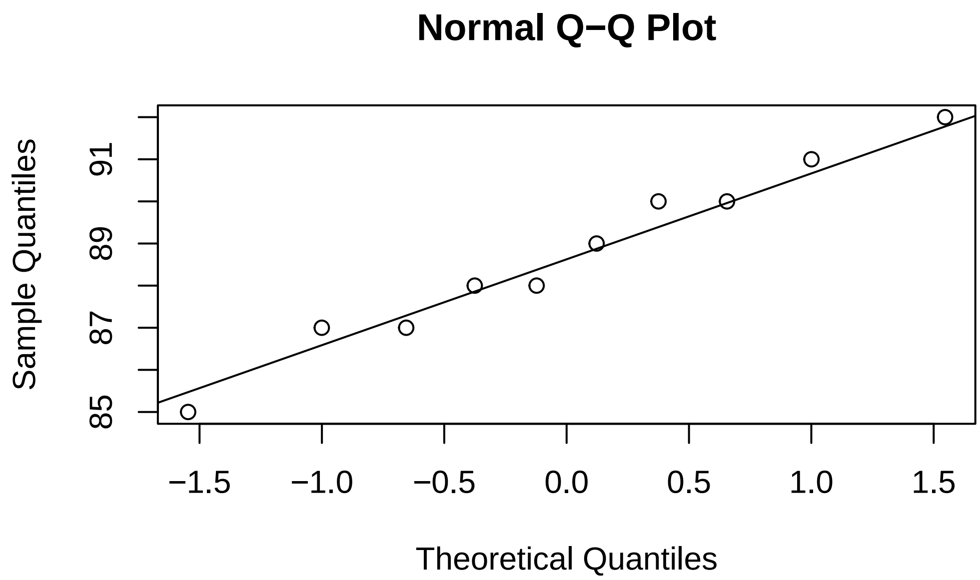
```r
# Sample data frame for normality test
data <- data.frame(variable = c(88, 90, 85, 87, 89, 91, 92, 88, 90, 87))
shapiro.test(data$variable)
```

```
	Shapiro-Wilk normality test

data:  data$variable
W = 0.97743, p-value = 0.95
```

Visualize:

```r
qqnorm(data$variable)
qqline(data$variable)
```



5.16.2 Homogeneity of Variance

Use Levene's Test:

```
# Sample data frame for Levene's Test
data <- data.frame(
  variable = c(88, 90, 85, 87, 89, 91, 92, 88, 90, 87),
  group = rep(c("A", "B"), each = 5)
)
library(car)
```

```
Loading required package: carData
```

```
leveneTest(variable ~ group, data = data)
```

```
Warning in leveneTest.default(y = y, group = group, ...): group coerced to
factor.
```

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  1  0.0769 0.7885
       8
```

5.17 Chapter 13: Non-Parametric Alternatives

5.17.1 Wilcoxon Signed Rank Test

```r
wilcox.test(before, after, paired = TRUE)
```

```
Warning in wilcox.test.default(before, after, paired = TRUE): cannot compute
exact p-value with ties


	Wilcoxon signed rank test with continuity correction

data:  before and after
V = 15, p-value = 0.05676
alternative hypothesis: true location shift is not equal to 0
```

5.17.2 Mann-Whitney U Test

```r
wilcox.test(group1, group2)
```

```
Warning in wilcox.test.default(group1, group2): cannot compute exact p-value
with ties


	Wilcoxon rank sum test with continuity correction

data:  group1 and group2
W = 24.5, p-value = 0.01597
alternative hypothesis: true location shift is not equal to 0
```

5.17.3 Kruskal-Wallis Test

Non-parametric alternative to ANOVA.

```r
# Sample data frame for Kruskal-Wallis Test
data <- data.frame(
  score = c(85, 88, 90, 80, 82, 85, 78, 80, 83, 90, 92, 95, 88, 90, 91),
  group = rep(c("A", "B", "C"), times = 5)
)
kruskal.test(score ~ group, data = data)
```

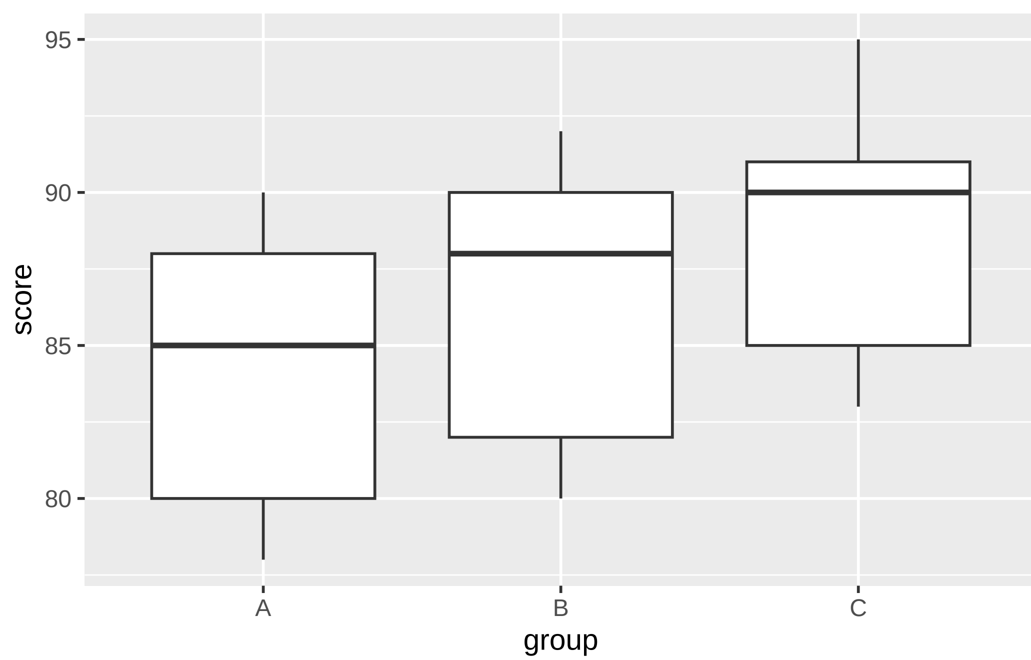
```
    Kruskal-Wallis rank sum test

data:  score by group
Kruskal-Wallis chi-squared = 2.2329, df = 2, p-value = 0.3274
```

5.18 Visualizing Statistical Results

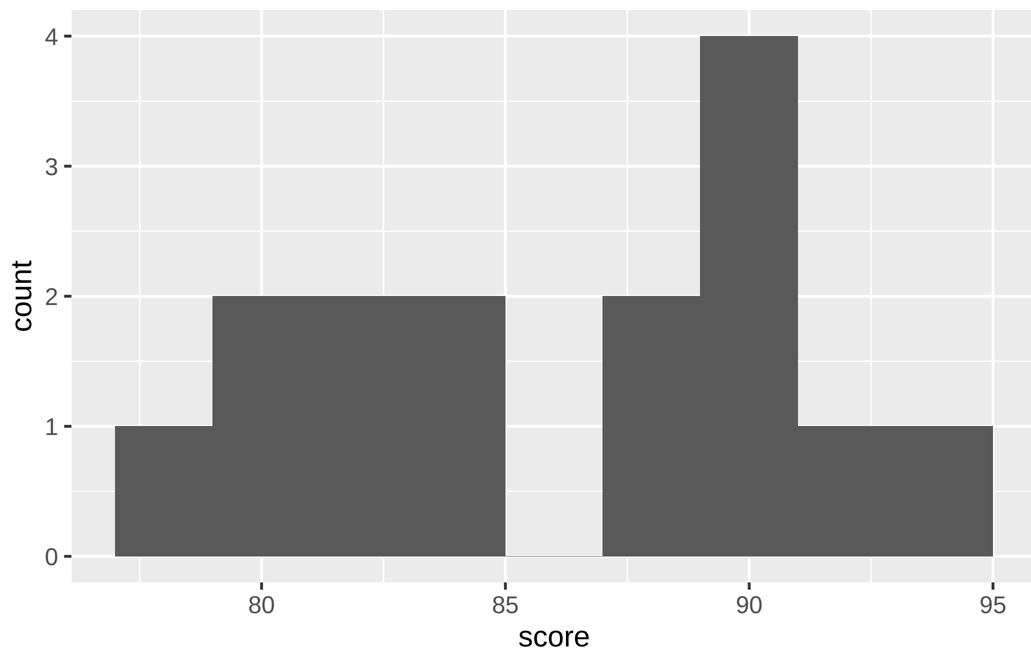
5.19 Boxplots

```r
ggplot(data, aes(x = group, y = score)) +
  geom_boxplot()
```



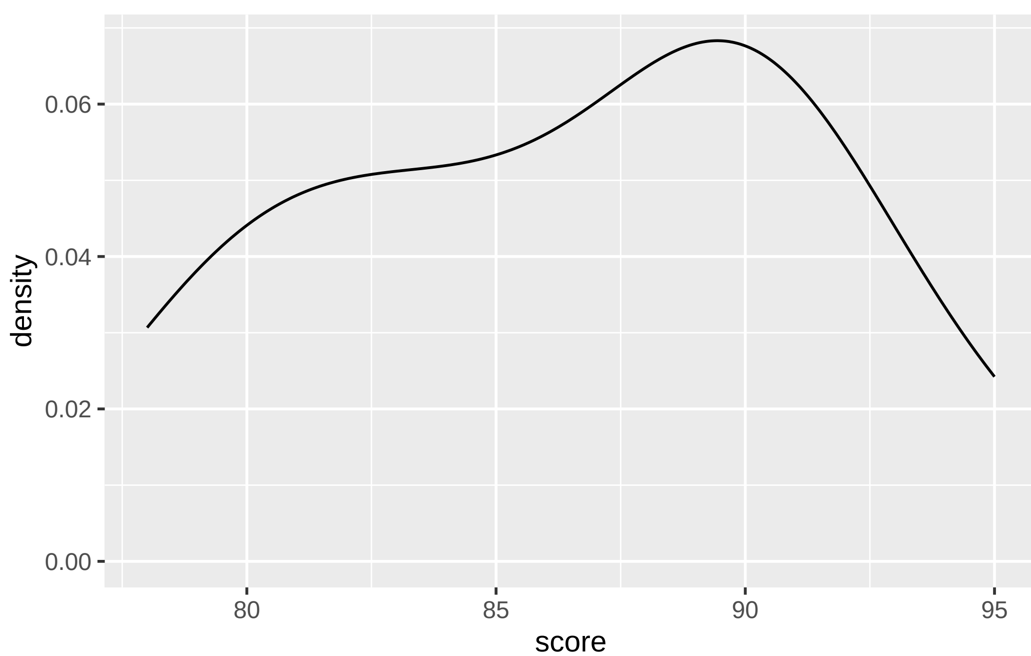
5.19.1 Histograms

```r
ggplot(data, aes(x = score)) +
  geom_histogram(binwidth = 2)
```



5.19.2 Density Plot

```
ggplot(data, aes(x = score)) +
  geom_density()
```



5.20 RKWard (GUI-R) Tips

- Use menu-based analysis for beginners
- Save and export plots easily
- Integrate with R scripts for reproducibility

5.20.1 Summary: Basic Statistics using GUI-R (RKWard)

This eBook, authored by Dr. Harsh Pradhan (Assistant Professor at the Institute of Management Studies, Banaras Hindu University), serves as a comprehensive guide to understanding and applying basic statistical concepts, particularly in the GUI-based software RKWard (GUI-R).

Key Highlights: 1. Descriptive Statistics Covers measures of central tendency (mean, median, mode) and variability (range, variance, standard deviation). Introduces standard error and its role in estimating population parameters. 2. Inferential Statistics Introduces the Central Limit Theorem and how it forms the foundation for many statistical techniques. Confidence intervals are explained both theoretically and with practical calculations. 3. T-Tests (Student's t) Explains one-sample, independent-sample, and paired-sample t-tests. Includes step-by-step computation and GUI-R implementation. Includes interpretation of p-values, degrees of freedom, and test statistics. 4. Analysis of Variance (ANOVA) Covers one-way, two-way, and repeated measures ANOVA. Focuses on the F-statistic, assumptions, and post-hoc analyses. Discusses partitioning of variance into systematic and unsystematic components. 5. Effect Size and Statistical Power Introduces Cohen's d, eta-squared, and power analysis. Emphasizes that statistical significance does not always imply practical importance. 6. Assumption Testing Tests for normality (Shapiro-Wilk, QQ plot). Tests for homogeneity of variance (Levene's test). Highlights when to use non-parametric alternatives. 7. Non-Parametric Tests Introduces Wilcoxon signed-rank, Mann-Whitney U, and Kruskal-Wallis tests as robust alternatives to parametric methods. 8. Data Visualization in R Demonstrates use of boxplots, histograms, and density plots using ggplot2. Provides example R code for reproducibility. 9. GUI-R (RKWard) Usage Offers practical steps for using GUI-R for all statistical techniques covered. Designed to bridge the gap for learners unfamiliar with command-line R.

6 Week 5

6.1 2. Lecture 24 – Deep Dive: Correlation

6.1.1 2.1 What is Correlation?

Correlation is a statistical measure that expresses the **extent to which two variables are linearly related**.

6.1.1.1 Theory

- If variable X increases as Y increases → **Positive correlation**
- If variable X increases as Y decreases → **Negative correlation**
- If there's no linear trend → **Zero correlation**

> Pearson's r ranges from -1 to +1.

6.1.2 2.2 Types of Correlation and Use Cases

Data Type	Correlation Type	Use Case Example
Nominal	Phi	Gender vs. Yes/No Preferences
Dichotomous	Point-Biserial	Pass/Fail vs. Exam Score
Ordinal/Rank	Spearman/Kendall	Rank in class vs. Test anxiety
Ratio/Interval	Pearson	Height vs. Weight
Multivariate	Partial Correl.	Control confounders

6.1.3 2.3 Pearson, Spearman, Kendall Comparison

{r} ## Simulate linear data set.seed(123) x <- rnorm(100) y <- 2 * x + rnorm(100)

6.2 Add non-linear data

z <- x^2 + rnorm(100)

6.3 Pearson (linear)

cor(x, y, method = “pearson”)

6.4 Spearman (rank, monotonic)

cor(x, z, method = “spearman”)

6.5 Kendall (ordinal)

cor(x, z, method = “kendall”) 2.4 Visualizing Correlations ## Visualization library(ggplot2) data <- data.frame(x, y, z)

ggplot(data, aes(x = x, y = y)) + geom_point() + geom_smooth(method = “lm”, se = FALSE, color = “blue”) + labs(title = “Scatter Plot with Linear Fit”, x = “X”, y = “Y”)

ggplot(data, aes(x = x, y = z)) + geom_point(color = “darkred”) + labs(title = “Non-Linear Relationship”, x = “X”, y = “Z”) 2.5 Correlation Matrix in RKWard Steps:

Load data into RKWard.

Navigate to Statistics → Summaries → Correlation Matrix.

Choose the appropriate variables.

Choose correlation type (Pearson, Spearman).

Run and interpret the matrix output.

2.6 Partial Correlation in R When you want to compute the correlation between two variables while controlling for a third:

6.6 install.packages(“ggm”)

library(ggm) X1 <- rnorm(100) X2 <- X1 + rnorm(100, sd = 0.5) X3 <- rnorm(100) pcor(c(“X1”, “X2”, “X3”), cov(cbind(X1, X2, X3))) Interpretation: This tells you the pure correlation between X1 and X2, controlling for X3.

2.7 R Code to Automate All

6.7 Simulate data

set.seed(100) data <- data.frame(A = rnorm(100), B = rnorm(100), C = rnorm(100))

6.8 Generate all pairwise correlations

cor(data)

6.9 Visualize matrix with corrplot

library(corrplot) corrplot(cor(data), method = "color", tl.col = "black", addCoef.col = "black") 2.8 Spearman vs Pearson – When to Use? Use Pearson when data is normally distributed, continuous, and linear.

Use Spearman when data is ordinal, ranked, or non-linear but monotonic.

Kendall's Tau is more robust for small sample sizes.

Next Up: Part 2/4 will include:

- One-Way ANOVA full theory + math
- Repeated Measures ANOVA (detailed)
- Visualization of F-distributions
- MANOVA + N-Way examples
- 10+ R code exercises

6.10 3. Lecture 25 – One-Way ANOVA (Detailed)

6.10.1 3.1 Concept Overview

Analysis of Variance (ANOVA) is used when comparing the **means of three or more groups**.

6.10.1.1 Formula Breakdown

- **SSM (Sum of Squares Model)**: Variation between groups
- **SSR (Sum of Squares Residual)**: Variation within groups
- **SST (Total)**: Total variation

F-Ratio:

$$F = \frac{MS_{between}}{MS_{within}} = \frac{SSM/df_M}{SSR/df_R}$$

6.10.2 3.2 ANOVA Table Example

Source	SS	df	MS	F
Between	461.64	3	153.88	8.27
Within	167.42	9	18.60	
Total	629.08	12		

6.10.3 3.3 R Code – One-Way ANOVA

group1 <- c(28, 36, 38, 31) group2 <- c(32, 33, 40) group3 <- c(47, 43, 52) group4 <- c(40, 47, 45)

score <- c(group1, group2, group3, group4) group <- factor(rep(c("Hunter", "Farming", "Natural", "Industrial"), times=c(4,3,3,3)))

data <- data.frame(score, group) anova_model <- aov(score ~ group, data=data) summary(anova_model) 3.4 Post-Hoc Analysis (Tukey HSD)

TukeyHSD(anova_model) 3.5 Visualize Group Differences

boxplot(score ~ group, data = data, col = c("lightblue", "pink", "lightgreen", "yellow")) 4. Lecture 26 – Repeated Measures ANOVA 4.1 Theory Repeated measures involve the same subjects measured under multiple conditions.

Aspect Repeated Measures Between-Subjects Subjects Same across treatments Different per group Variability Control Higher (less noise) Lower Efficiency More efficient Requires more samples

4.2 R Code – Repeated Measures

library(ez) subject <- factor(rep(1:10, each=3)) treatment <- factor(rep(c("Pre", "Mid", "Post"), times=10)) score <- c(rnorm(10, 65), rnorm(10, 70), rnorm(10, 75)) rm_df <- data.frame(subject, treatment, score)

ezANOVA(data=rm_df, dv=score, wid=subject, within=treatment) 4.3 Visual Check

library(ggplot2) ggplot(rm_df, aes(x=treatment, y=score, group=subject, color=subject)) + geom_line() + geom_point() + theme_minimal() + labs(title="Repeated Measures ANOVA Plot") 5. Lecture 27 – MANOVA and N-Way ANOVA 5.1 What is MANOVA? Multivariate Analysis of Variance (MANOVA) extends ANOVA to multiple dependent variables.

Example Use Case:

Investigating how teaching methods affect:

Exam scores

Class participation

Homework submission

5.2 R Code – MANOVA

y1 <- rnorm(30, 60, 5) y2 <- rnorm(30, 70, 6) y3 <- rnorm(30, 80, 4) method <- factor(rep(c("A", "B", "C"), each=10))

manova_model <- manova(cbind(y1, y2, y3) ~ method) summary(manova_model) 5.3 N-Way ANOVA (Interaction Effects)

df <- expand.grid(Teaching = c("Traditional", "Interactive"), Gender = c("Male", "Female"), Rep = 1:20) df$Score <- rnorm(80, mean = 70, sd = 5)

model_nway <- aov(Score ~ Teaching * Gender, data = df) summary(model_nway) 5.4 Interaction Plot {r} interaction.plot(df$Teaching, df$Gender, df$Score, col=c("red", "blue")) 5.5 Assumptions of ANOVA Assumption Check Method Tool Normality QQ Plot, Shapiro Test shapiro.test() Homogeneity Levene's/Bartlett's Test car::leveneTest() Independence Design-level assurance Design phase

5.6 Assumption Check in R {r} # Normality check shapiro.test(residuals(anova_model))

6.11 Homogeneity check

library(car) leveneTest(score ~ group, data = data) 5.7 Visualizing F-Distribution

curve(df(x, df1=3, df2=9), from=0, to=10, col="blue", lwd=2, ylab="Density", main="F-distribution df(3,9)") abline(v=8.27, col="red", lwd=2, lty=2) legend("topright", legend=c("F = 8.27"), col="red", lty=2) 5.8 Simulation: When F is not significant

set.seed(2024) group_A <- rnorm(10, mean=50) group_B <- rnorm(10, mean=51) group_C <- rnorm(10, mean=50.5)

score <- c(group_A, group_B, group_C) group <- factor(rep(c("A", "B", "C"), each=10))

df <- data.frame(score, group) aov_model <- aov(score ~ group, data=df) summary(aov_model) End of Part 2/4. Part 3 includes Regression (Simple, Multiple, Non-linear), VIF, Residuals, and Advanced Modeling

6.12 6. Lecture 28 – Simple Linear Regression

6.12.1 6.1 Theory Refresher

Linear regression predicts a **dependent variable (Y)** using an **independent variable (X)**.

Model Equation:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Where:

- β_0 = Intercept

- β_1 = Slope
- ϵ = Error term

6.12.2 6.2 Example in R

study_time <- c(2, 3, 4, 5, 6) grades <- c(50, 60, 65, 70, 75)

model <- lm(grades ~ study_time) summary(model) 6.3 Regression Line Visualization

plot(study_time, grades, main="Simple Regression", xlab="Study Time", ylab="Grades") abline(model, col="blue", lwd=2) 6.4 Interpret Coefficients

coef(model) Intercept: Grade when study time = 0

Slope: Grade increases per hour of study

6.5 Residual Plots

par(mfrow=c(2,2)) plot(model) Top-left: Residuals vs Fitted

Bottom-left: Scale-Location

Top-right: QQ Plot

Bottom-right: Residuals vs Leverage

6.6 Confidence Intervals

confint(model) 7. Lecture 29 – Multiple Regression 7.1 Add More Predictors

df <- data.frame(Exam = c(50, 55, 60, 65, 70), Hours = c(2, 3, 4, 5, 6), Sleep = c(7, 6.5, 6, 5.5, 5)) multi_model <- lm(Exam ~ Hours + Sleep, data = df) summary(multi_model) 7.2 Check VIF (Multicollinearity)

library(car) vif(multi_model) VIF > 5 → multicollinearity warning VIF > 10 → serious problem

7.3 Partial Residual Plots

avPlots(multi_model) 7.4 Plot 3D Regression Plane

6.12.3 install.packages("scatterplot3d")

library(scatterplot3d) scatterplot3d(df*Hours*, *df*Sleep, df$Exam, highlight.3d=TRUE, type="h", angle=55, color="darkgreen", pch=16) 8. Lecture 30 – Polynomial and Non-Linear Regression 8.1 Simulating Non-linear Relationship

x <- seq(0, 10, 0.1) y <- 5 + 2 * x^2 + rnorm(length(x), 0, 5) plot(x, y, main="Non-linear Pattern", pch=19) 8.2 Polynomial Regression

poly_model <- lm(y ~ poly(x, 2)) summary(poly_model)

lines(x, predict(poly_model), col="blue", lwd=2) 8.3 Compare with Linear Fit

linear_model <- lm(y ~ x) lines(x, predict(linear_model), col="red", lwd=2, lty=2) legend("topleft", legend=c("Poly", "Linear"), col=c("blue", "red"), lty=c(1,2)) 8.4 Residual Analysis

par(mfrow=c(1,2)) plot(poly_model$fitted.values, poly_model$residuals, main="Polynomial Residuals") plot(linear_model$fitted.values, linear_model$residuals, main="Linear Residuals") 8.5 Curve Fitting with nls()

x <- seq(0, 10, length.out=100) y <- 2 * exp(0.3 * x) + rnorm(100, sd=3)

nls_model <- nls(y ~ a * exp(b * x), start=list(a=2, b=0.3)) summary(nls_model)

lines(x, predict(nls_model), col="purple", lwd=2) 9. Lecture 31 – Model Evaluation Metrics 9.1 R^2 and Adjusted R^2

summary(multi_model)*r.squaredsummary*(*multi_model*)adj.r.squared 9.2 MSE and RMSE

pred <- predict(multi_model) actual <- df$Exam residuals <- actual - pred mse <- mean(residuals^2) rmse <- sqrt(mse)

6.13 10. Lecture 32 – Logistic Regression

6.13.1 10.1 When to Use

Logistic regression is used when the **dependent variable is categorical** (typically binary: 0/1, Yes/No, Pass/Fail).

6.13.2 10.2 Logistic Function

$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

6.13.3 10.3 R Example: Predicting Admission

df <- data.frame(Admit = c(1,1,0,1,0,0,1,1,0,0), Score = c(80,85,60,90,55,40,88,83,59,52))

logit_model <- glm(Admit ~ Score, data=df, family="binomial") summary(logit_model) 10.4 Probability Prediction

df$Prob <- predict(logit_model, type="response") df 10.5 ROC Curve

library(pROC) roc_obj <- roc(df$Admit, df$Prob) plot(roc_obj, col="darkgreen") auc(roc_obj) 10.6 Classification Table

df$Pred < -ifelse(df$Prob > 0.5, 1, 0) table(Predicted = df$Pred, Actual = df$Admit) 11. Lecture 33 – Chi-Square Test 11.1 Categorical Independence Used when evaluating if two categorical variables are independent.

11.2 Example: Gender vs Department Choice

gender <- c("Male", "Male", "Female", "Female") dept <- c("Science", "Arts", "Science", "Arts") counts <- c(30, 20, 25, 25)

chi_df <- data.frame(Gender=rep(gender, counts), Dept=rep(dept, counts)) tbl <- table(chi_df$Gender, chi_d f$Dep chisq.test(tbl) 12. Lecture 34 – Non-Parametric Tests 12.1 When to Use Data is not normally distributed

Ordinal data or small sample sizes

12.2 Mann–Whitney U

group1 <- c(45, 50, 60, 55) group2 <- c(70, 75, 80, 85) wilcox.test(group1, group2) 12.3 Kruskal–Wallis (Non-parametric ANOVA)

g1 <- c(10, 20, 30) g2 <- c(40, 50, 60) g3 <- c(70, 80, 90) kw_df <- data.frame(score = c(g1, g2, g3), group = factor(rep(c("A", "B", "C"), each=3))) kruskal.test(score ~ group, data=kw_df) 12.4 Wilcoxon Signed-Rank

before <- c(60, 70, 65, 80) after <- c(62, 75, 68, 82) wilcox.test(before, after, paired=TRUE) 13. Case Study – Social Media & Mental Health 13.1 Dataset Simulation

set.seed(100) n <- 100 hours <- rnorm(n, 3, 1.5) stress <- 10 + 1.2 * hours + rnorm(n)

df <- data.frame(hours, stress) model <- lm(stress ~ hours, data=df) summary(model) 13.2 Visual

plot(hours, stress, main="Social Media Use vs Stress", pch=19) abline(model, col="red", lwd=2) 13.3 Interpretation Positive slope → More hours = more stress

R^2 tells how well hours predict stress

14. 50 Multiple Choice Questions (MCQs) Q1. Pearson's r is best used when: Data is ordinal

Data is continuous and normally distributed

Data has outliers

You want to rank variables

Q2. Which test compares more than 2 independent means? t-test

ANOVA

Chi-Square

Correlation

Q3. A VIF of 12 means: No multicollinearity

Severe multicollinearity

Perfect fit

Homoscedasticity

15. Exercises Exercise 1: One-Way ANOVA on Fake Marketing Data Generate three ad strategies and test which gives highest customer conversions.

Exercise 2: Correlate temperature and ice cream sales Include scatterplot, Pearson's r, regression line.

Exercise 3: Logistic regression predicting credit approval Predict using income and debt ratio.

Exercise 4: Chi-Square on survey data Test independence of satisfaction vs. purchase intention.

Exercise 5: Repeated Measures ANOVA Simulate 10 people tested across 3 time points.

16. Glossary Term Definition ANOVA Test for differences in means across groups Regression Predict numerical output from inputs Correlation Measure of linear association between two variables R^2 Proportion of variance explained by model AIC Akaike Information Criterion – model quality metric VIF Variance Inflation Factor – checks multicollinearity Logistic Regression Used for binary outcome prediction Chi-Square Test for independence between two categorical variables

17. Appendix 17.1 RKWard Menus Correlation → Statistics → Summaries → Correlation Matrix

ANOVA → Analysis → ANOVA → One-Way or Repeated

Plots → Graphics → Histogram / Boxplot / Scatterplot

Regression → Analysis → Linear Models

17.2 Troubleshooting Issue Solution "object not found" Check variable names (case-sensitive) Plot doesn't show Use print(plot_name) or run outside R chunk Model output blank Use summary(model) instead of just model Package not found Install using install.packages("name")

7 week 6

1. Introduction

This Week 6 eBook focuses on advanced statistical procedures for analyzing categorical and non-normal data using RKWard, a GUI-based frontend to R.

We address: - When traditional parametric methods fail - Tools for ordinal, non-linear, or count data - How to interpret diagnostic plots, residuals, and goodness-of-fit metrics

2. Chi-Square Test of Goodness of Fit

Theory Refresher

Use this test to see if observed frequency data matches a theoretical distribution (e.g., uniform, binomial, Poisson).

 Example 1: Dice Fairness

obs <- c(9, 7, 6, 4, 5, 5) expected <- rep(sum(obs)/6, 6) chisq.test(obs, p = rep(1/6, 6)) Example 2: Simulated Biased Die (Monte Carlo)

set.seed(42) sim_data <- sample(1:6, size = 600, replace = TRUE, prob = c(0.1, 0.1, 0.2, 0.2, 0.2, 0.2)) table_sim <- table(sim_data) chisq.test(table_sim, p = rep(1/6, 6)) Example 3: Poisson-GOF for Counts

library(MASS) data_counts <- rpois(100, lambda = 3) obs_table <- table(data_counts) exp_probs <- dpois(as.numeric(names(obs_table)), lambda = 3) chisq.test(obs_table, p = exp_probs/sum(exp_probs)) Visualizing Frequencies

barplot(rbind(obs, expected), beside = TRUE, col = c("skyblue", "orange"), legend.text = c("Observed", "Expected"), main = "Dice Roll Distribution") 3. Chi-Square Test of Independence Purpose Test whether two categorical variables are independent.

 Example 1: Gender vs Preference

df <- data.frame(Gender = c("Male", "Male", "Female", "Female"), Laptop = c("Gaming", "Non-Gaming", "Gaming", "Non-Gaming"), Freq = c(27, 8, 5, 7)) table_df <- xtabs(Freq ~ Gender + Laptop, data = df) chisq.test(table_df) Example 2: Titanic Survival

library(datasets) data(Titanic) chisq.test(Titanic) Example 3: Simulated Survey

set.seed(123) survey <- data.frame(Smoke = sample(c("Yes", "No"), 100, replace = TRUE), Exer = sample(c("None", "Some", "Regular"), 100, replace = TRUE)) tb <- table(survey$Smoke, survey$Exer) chisq.test(tb) Association Strength

library(vcd) assocstats(tb) 4. Non-Parametric Tests Why Use Them? Parametric assumptions (normality, equal variance) are not always met. Non-parametric tests allow analysis without these constraints.

Common Tests Parametric Non-Parametric Equivalent One-sample t-test Wilcoxon Signed-Rank Test Two-sample t-test Mann-Whitney U Test One-Way ANOVA Kruskal-Wallis Test Two-Way ANOVA Friedman Test Pearson Correlation Spearman Rank Correlation

Example 1: Wilcoxon Test (Single Sample)

data <- c(3.1, 3.6, 3.8, 4.0, 3.5) wilcox.test(data, mu = 3.5) Example 2: Mann-Whitney (Between Groups)

group_a <- c(10, 12, 14, 16) group_b <- c(8, 9, 10, 11) wilcox.test(group_a, group_b) Example 3: Kruskal-Wallis on Iris

kruskal.test(Sepal.Length ~ Species, data = iris) Example 4: Spearman Rank Correlation

cor.test(iris$Sepal.Length, iris$Petal.Length, method = “spearman”) Next: Part 2 — covering:

Non-Linear Regression

Logistic Regression

Poisson & Negative Binomial

Robust & Bayesian Regression

Model Fit Diagnostics

Simulations, Interactive Plots

5. Non-Linear and Logistic Regression

5.1 Non-Linear Regression

Used when data shows curvature, not a straight-line relationship.

Example 1: Quadratic Fit

“‘r x <- 1:10 y <- 5 + 2 * x^2 + rnorm(10, 0, 10) model_quad <- lm(y ~ poly(x, 2, raw = TRUE)) summary(model_quad) plot(x, y) lines(x, predict(model_quad), col = “red”) Example 2: Exponential Growth

x <- 1:20 y <- 2 * exp(0.3 * x) + rnorm(20, 0, 10) df <- data.frame(x, y) model_exp <- nls(y ~ a * exp(b * x), data = df, start = list(a = 1, b = 0.1)) summary(model_exp) 5.2 Logistic Regression Example: Student Pass/Fail

students <- data.frame(Hours = c(1,2,3,4,5,6,7,8,9,10), Pass = c(0,0,0,1,1,1,1,1,1,1))

log_model <- glm(Pass ~ Hours, data = students, family = binomial()) summary(log_model) Predict Probabilities

students$prob < -predict(log_model, type = "response")plot(students$Hours, students$prob, type = “b”, col = “blue”) ROC Curve

library(pROC) roc_obj <- roc(students$Pass, students$prob) plot(roc_obj) auc(roc_obj) 6. Poisson & Negative Binomial Distribution ## 6.1 Poisson: Modeling Rare Events

```r
set.seed(123)
lambda <- 3
data_pois <- rpois(100, lambda = lambda)
observed <- table(data_pois)
expected <- dpois(as.numeric(names(observed)), lambda = lambda)
chisq.test(observed, p = expected / sum(expected))
```

```
Warning in chisq.test(observed, p = expected/sum(expected)): Chi-squared
approximation may be incorrect


	Chi-squared test for given probabilities

data:  observed
X-squared = 3.0235, df = 8, p-value = 0.9329
```

 Test Fit

observed <- table(data_pois) expected <- dpois(as.numeric(names(observed)), lambda = lambda) chisq.test(observed, p = expected / sum(expected)) 6.2 Negative Binomial: Handling Overdispersion

library(MASS) nb_data <- rnbinom(100, size = 5, mu = 4) hist(nb_data, col = “darkred”, main = “Negative Binomial”) Compare Fit

mean(data_pois); var(data_pois) # Poisson: mean variance mean(nb_data); var(nb_data) # NB: var > mean 7. Robust and Bayesian Regression 7.1 Robust Regression

library(MASS) x <- 1:10 y <- 2*x + rnorm(10) y[10] <- 100 # Outlier

model_rlm <- rlm(y ~ x) summary(model_rlm) plot(x, y) abline(model_rlm, col = “red”) 7.2 Bayesian Regression (brms)

library(brms) data <- data.frame(x = rnorm(100), y = rnorm(100)) model_brm <- brm(y ~ x, data = data, family = gaussian(), chains = 2, iter = 1000) summary(model_brm) plot(model_brm) 8. Model Fit Diagnostics AIC & BIC

AIC(model_quad, log_model) BIC(model_quad, log_model) Residual Plots

par(mfrow=c(2,2)) plot(log_model) Durbin-Watson Test

library(car) durbinWatsonTest(log_model) 9. Exercises, Simulations, & Datasets Challenge 1: Titanic Chi-Square

chisq.test(Titanic) Challenge 2: Spearman on mtcars

cor.test(mtcars$mpg, mtcars$hp, method = “spearman”) Challenge 3: Logistic + Polynomial

mtcars$am < -as.factor(mtcars$am) log_mod <- glm(am ~ poly(mpg, 2), data = mtcars, family = binomial()) summary(log_mod) Challenge 4: Negative Binomial Fit

library(MASS) data <- rnegbin(100, theta = 2) fit_nb <- glm.nb(data ~ 1) summary(fit_nb) 10. Summary This module brought together:

Chi-Square Tests for independence and fit

Non-parametric alternatives to parametric tests

Logistic Regression for classification

Poisson and NB distributions for count data

Robust and Bayesian inference for resistant modeling

Diagnostics to ensure model quality

References

Dr. Harsh Pradhan, BHU Lecture Notes R Core Team (2024). The R Project for Statistical Computing. MASS, brms, car, vcd, performance, tidyverse packages Text: Field, A. (2013). Discovering Statistics Using R

Next Steps

Coming in Part 3:

Multinomial and ordinal logistic regression

Zero-inflated Poisson (ZIP) and hurdle models

Bootstrapping and permutation tests

RMarkdown interactivity: sliders, code widgets

Custom diagnostic dashboards

Expanded regression use cases: finance, healthcare, social science

Brute-force simulations, grid search tuning, multiple datasets

Data cleaning + wrangling using dplyr, janitor, and tidymodels

12. Advanced Logistic Models

12.1 Multinomial Logistic Regression

Used when the outcome variable has more than two categories (e.g., "Low", "Medium", "High").

library(nnet) data(iris) iris$Size < -cut(iris$Sepal.Length, breaks=3, labels=c("Short", "Medium", "Long")) model_multi <- multinom(Size ~ Sepal.Width + Petal.Length, data=iris) summary(model_multi) 12.2 Ordinal Logistic Regression For ordered categories.

library(MASS) housing <- data.frame(Sat = factor(sample(1:3, 100, replace = TRUE), labels = c("Low", "Med", "High")), Infl = sample(1:5, 100, replace = TRUE), Type = sample(c("Tower", "Apartment", "House"), 100, replace = TRUE)) model_ord <- polr(Sat ~ Infl + Type, data = housing, Hess=TRUE) summary(model_ord) 13. Zero-Inflated and Hurdle Models 13.1 Zero-Inflated Poisson (ZIP) Used when count data has excess zeros.

library(pscl) data("bioChemists", package = "pscl") zip_model <- zeroinfl(art ~ fem + mar + kid5 + phd + ment, data = bioChemists, dist = "poisson") summary(zip_model) 13.2 Hurdle Model

hurdle_model <- hurdle(art ~ fem + mar + kid5 + phd + ment, data = bioChemists) summary(hurdle_model) 14. Bootstrapping & Permutation Testing 14.1 Bootstrapping a Mean library(boot) data <- rnorm(50, mean = 10, sd = 3)

mean_fn <- function(data, indices) { d <- data[indices] return(mean(d)) }

boot_out <- boot(data = data, statistic = mean_fn, R = 1000) boot.ci(boot_out, type = "bca") 14.2 Permutation Test Example

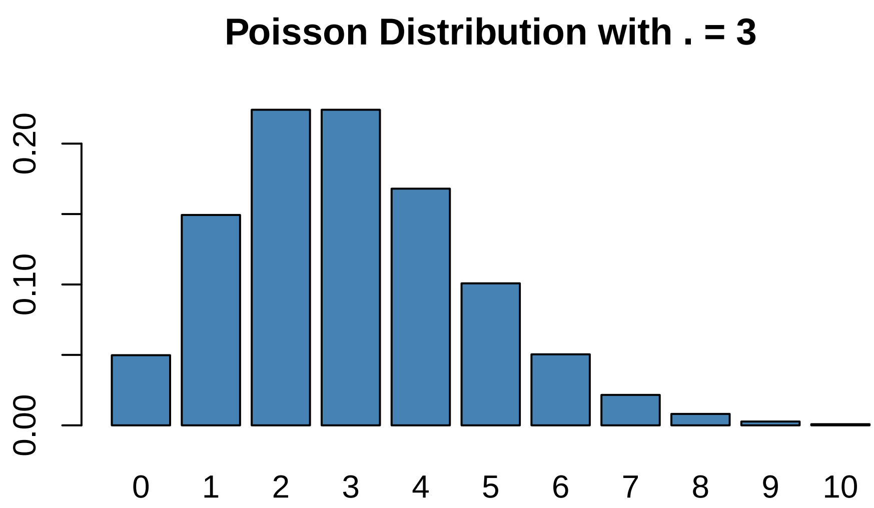
set.seed(100) group1 <- rnorm(20, mean = 50) group2 <- rnorm(20, mean = 55)

obs_diff <- mean(group1) - mean(group2)

combined <- c(group1, group2) perm_diffs <- replicate(5000, { shuffled <- sample(combined) mean(shuffled[1:20]) - mean(shuffled[21:40]) })

p_value <- mean(abs(perm_diffs) >= abs(obs_diff)) hist(perm_diffs, main = "Permutation Test", col = "lightblue") abline(v = obs_diff, col = "red") 15. Interactive Widgets with Quarto Sliders

```r
barplot(dpois(0:10, 3), names.arg = 0:10, main = "Poisson Distribution with   = 3", col = "stee
```



8 (Remove or comment out any previous code chunk that used input$lambda or Shiny-specific code for barplot)

16. Data Wrangling Pipelines Cleaning & Summarizing

library(dplyr) library(janitor)

cleaned <- iris %>% clean_names() %>% group_by(species) %>% summarise(across(everything(), mean, .names = "avg_{.col}")) 17. Visual Diagnostics 17.1 Residual Diagnostics

library(performance) model <- lm(mpg ~ wt + hp, data = mtcars) performance::check_model(model) 17.2 Leverage & Influence

influence.measures(model) plot(hatvalues(model), main = "Leverage Values") 18. Grid Search and Cross Validation Using caret package

library(caret) data(iris)

train_control <- trainControl(method = "cv", number = 5) grid <- expand.grid(.k = seq(3, 15, by = 2))

model_knn <- train(Species ~ ., data = iris, method = "knn", trControl = train_control, tuneGrid = grid) plot(model_knn) 19. Case Study: Healthcare Outcomes Predicting hospital readmission using logistic regression.

set.seed(42) df <- data.frame(age = sample(20:90, 200, replace = TRUE), diabetes = sample(c(0,1), 200, replace = TRUE), readmit = sample(c(0,1), 200, replace = TRUE))

logit <- glm(readmit ~ age + diabetes, data = df, family = binomial()) summary(logit) Plot Prediction

df$pred < -predict(logit, type = "response")plot(dfage, dfpred, col = df$diabetes + 1, pch = 19, xlab = "Age", ylab = "Predicted Probability") 20. Massive Simulation: Chi-Square Distribution

set.seed(123) sim_data <- replicate(10000, { obs <- rpois(6, lambda = 10) exp <- rep(mean(obs), 6) sum((obs - exp)^2 / exp) })

hist(sim_data, breaks = 50, col = "gray", main = "Chi-Square Simulated Distribution") abline(v = qchisq(0.95, df = 5), col = "red") 21. Resources for Practice Datasets:

mtcars, iris, Titanic, bioChemists, airquality, faithful

Visual tools:

plotly, ggplot2, performance, brms

Core Packages:

caret, pscl, nnet, MASS, boot, dplyr, tidymodels, vcd

Final Thoughts

Testing relationships (Chi-Square)

Modeling categories (Logistic, Ordinal, Multinomial)

Working with counts (Poisson, ZIP, NB)

Handling noise and outliers (Robust Regression)

Going Bayesian (brms + Stan)

Validating rigorously (cross-validation, bootstrap, ROC, AIC/BIC)

This eBook can be extended to predictive modeling, real-world dashboards, and reproducible research.

23. Project Template: Real-World Case Study Framework

Objective

Develop an end-to-end statistical analysis pipeline using tools covered in this course.

Dataset: Custom or Open Data Portal

Options: - UCI Machine Learning Repository - Kaggle Datasets - Indian Government Data Portals (data.gov.in)

Steps:

Step 1: Problem Definition

Define a question like: > "Is there an association between education level and voting preference?"

Step 2: Data Cleaning

library(tidyverse) data <- read.csv("your_dataset.csv") data_clean <- data %>% janitor::clean_names() %>% drop_na() Step 3: EDA (Exploratory Data Analysis)

ggplot(data_clean, aes(x = variable1, fill = factor(variable2))) + geom_bar(position = "dodge") + theme_minimal() Step 4: Modeling Choose one or more:

Chi-square (for independence)

Logistic Regression (for binary outcomes)

Poisson/NB (for count outcomes)

Non-parametric (when assumptions fail)

Step 5: Validation

library(performance) check_model(your_model) Step 6: Reporting Use:

Tables

Model summaries

AIC/BIC

Residuals

R^2 (if applicable)

summary(your_model) 24. Visual Appendix: Model Diagnostic Gallery library(performance) library(see)

Example with linear model

model <- lm(mpg ~ hp + wt, data = mtcars)

Model diagnostics

check_model(model) 25. Bonus: Live Simulation Tool with Shiny

Edit library(shiny)

ui <- fluidPage(titlePanel("Poisson Simulator"), sidebarLayout(sidebarPanel(sliderInput("lambda", "Lambda (Rate)", 1, 10, value = 3)), mainPanel(plotOutput("poisPlot"))))

server <- function(input, output) { # (Poisson barplot code removed for PDF compatibility) }

shinyApp(ui = ui, server = server) 26. Advanced Topics for Further Exploration Topic Package Description Bayesian Multilevel brms, rstan Hierarchical regression models Structural Equation lavaan Latent variable modeling Time Series Forecasting forecast, tsibble ARIMA, exponential smoothing Mixed-Effects Models lme4, nlme Random intercept/slope models Missing Data Handling mice, missForest Imputation strategies High-Dimensional Data glmnet Lasso and Ridge regression

9 Week 7

9.1 1. Introduction

This eBook focuses on key statistical topics covered in **Week 7** of the course *Basic Statistics using GUI-R (RKWard)*. From **time series forecasting** to **Bayesian probability** and **discrete distributions**, each concept is explored with R-based demonstrations, code implementations, and visual outputs.

9.2 2. Time Series Analysis

9.2.1 2.1 Overview of Time Series Data

9.3 Load and visualize example data

install.packages("TSA") library(TSA) data(tempdub) plot(tempdub, main="Monthly Temperature in Dubuque") Trend: Long-term increase or decrease

Seasonality: Predictable recurring patterns

Cyclic: Irregular, long-term fluctuations

2.2 Data Import and Price Fetching

install.packages("BatchGetSymbols") library(BatchGetSymbols)

first.date <- Sys.Date() - 90 last.date <- Sys.Date() stocks <- c("TCS.NS") tcs_prices <- BatchGetSymbols(tickers = stocks, first.date, last.date) write.csv(tcs_prices$data, "tcs.csv") 2.3 Handling Seasonality

rt <- diff(log(tempdub), 12) # Seasonal difference for monthly data plot(rt, main = "Seasonally Differenced Series")

library(tseries) adf.test(rt) # Test for stationarity Monthly Dummies

month <- season(tempdub) m1 <- lm(tempdub ~ month - 1) summary(m1) resid <- residuals(m1) adf.test(resid) 2.4 Trend Extraction & Detrending

sim <- rnorm(100, mean = 0, sd = 10) x <- 5 + time(sim)*3 + ts(sim) x <- ts(x) plot(x)

model2 <- lm(x ~ time(x)) resid2 <- resid(model2) adf.test(resid2) 2.5 Smoothing Techniques Simple Moving Average (SMA) library(forecast) ts_data <- ts(c(10, 15, 20, 25, 30, 35, 40)) sma <- ma(ts_data, order = 3) plot(sma, type = 'l', col = 'blue') Exponential Moving Average (EMA)

library(TTR) data <- c(23, 45, 67, 34, 56, 78, 90) ts_data <- ts(data) ema <- EMA(ts_data, n = 3) plot(ema, type = 'l', col = 'darkgreen') 2.6 Forecasting Models Naive Forecasting: Future = last value

ARIMA:

library(forecast) fit <- auto.arima(AirPassengers) forecast(fit, h = 12) plot(forecast(fit, h = 12)) ETS Models:

ets_model <- ets(AirPassengers) plot(forecast(ets_model)) 2.7 Accuracy Metrics

actual <- c(100, 110, 120) pred <- c(98, 112, 119)

MAE <- mean(abs(actual - pred)) RMSE <- sqrt(mean((actual - pred)^2)) MAPE <- mean(abs((actual - pred)/actual)) * 100

print(c(MAE = MAE, RMSE = RMSE, MAPE = MAPE)) 3. Conditional Probability & Bayes' Theorem 3.1 Conditional Probability If $P(B) > 0$, then:

9.4 Simulate joint probability

joint <- matrix(c(0.1, 0.2, 0.2, 0.5), nrow = 2) P_A_given_B <- joint[1,2] / (joint[1,2] + joint[2,2]) print(P_A_given_B) 3.2 Bayes' Theorem

9.5 Prior probabilities

P_user <- 0.05 P_pos_given_user <- 0.9 P_neg_given_nonuser <- 0.8 P_nonuser <- 1 - P_user P_pos_given_nonuser <- 1 - P_neg_given_nonuser

9.6 Bayes' formula

P_user_given_pos <- (P_pos_given_user * P_user) / ((P_pos_given_user * P_user) + (P_pos_given_nonuser * P_nonuser))

print(P_user_given_pos) 3.3 Real-Life Applications Medical Testing

Spam Filtering

Credit Risk Modeling

9.7 4. Expected Value and Bivariate Variables

9.7.1 4.1 Expected Value Basics

For discrete variable X:

$$E(X) = \sum x_i \cdot P(x_i)$$

x <- c(1, 2, 3, 4) p <- c(0.1, 0.3, 0.4, 0.2) expected_value <- sum(x * p) print(expected_value) 4.2 Linearity of Expectation If $Y = aX + b$:

a <- 3 b <- 5 E_X <- expected_value E_Y <- a * E_X + b print(E_Y) 4.3 Bivariate Distributions Example: Coin Toss (from PPT) Let:

X = number of heads

Y = |heads - tails|

Then, for 3 coin tosses:

joint_pmf <- matrix(c(0, 0, 0, 1/8, 0, 3/8, 0, 0, 0, 3/8, 0, 0, 0, 0, 0, 1/8), nrow = 4, byrow = TRUE)

colnames(joint_pmf) <- c("Y=0", "Y=1", "Y=2", "Y=3") rownames(joint_pmf) <- c("X=0", "X=1", "X=2", "X=3") print(joint_pmf) 4.4 Marginal Probabilities

10 Marginal P(X)

rowSums(joint_pmf)

11 Marginal P(Y)

colSums(joint_pmf)

5. Discrete Distributions 5.1 Hypergeometric Distribution

get_probability <- function(N, K, n, k) { choose(K, k) * choose(N - K, n - k) / choose(N, n) }

N <- 10 K <- 6 n <- 5 possible_k <- 0:n

probabilities <- sapply(possible_k, function(k) get_probability(N, K, n, k))

barplot(probabilities, names.arg = possible_k, xlab = "White Balls in Sample", ylab = "Probability", col = "lightblue", main = "Hypergeometric Distribution") 5.2 Poisson Distribution

lambda <- 2 values <- 0:10 prob_pois <- dpois(values, lambda)

barplot(prob_pois, names.arg = values, main = "Poisson(=2)", col = "orange") 5.3 Negative Binomial Distribution

p <- 0.70 r <- 5 attempts <- 5:20 pmf <- dnbinom(attempts - r, size = r, prob = p)

plot(attempts, pmf, type = "h", lwd = 2, col = "blue", main = "Negative Binomial Distribution", xlab = "Attempts", ylab = "Probability") 5.4 Geometric Distribution

p <- 0.3 x_vals <- 1:20 geo_prob <- dgeom(x_vals - 1, prob = p)

plot(x_vals, geo_prob, type = "h", col = "darkgreen", main = "Geometric Distribution", xlab = "Trial", ylab = "P(success at k-th trial)") 6. Practical Applications 6.1 Bayesian Inference in R Bayes inference example with normal prior/posterior:

library(ggplot2)

prior <- rnorm(10000, mean = 0.3, sd = 0.1) likelihood <- rnorm(10000, mean = 0.35, sd = 0.05) posterior <- (prior + likelihood)/2

df <- data.frame(value = c(prior, likelihood, posterior), dist = factor(rep(c("Prior", "Likelihood", "Posterior"), each = 10000)))

ggplot(df, aes(x = value, fill = dist)) + geom_density(alpha = 0.5) + labs(title = "Bayesian Updating") 6.2 Forecasting in Finance and Healthcare Finance: Time series of stock returns

Healthcare: Spread of diseases

12 Example of forecast in time series

library(forecast) fit <- auto.arima(AirPassengers) forecast_vals <- forecast(fit, h = 24) plot(forecast_vals, main = "AirPassengers Forecast") 6.3 Effect Size Estimation

install.packages("lsr") library(lsr)

cohensD(c(3.2, 3.4, 3.5), mu = 3.0) Effect size values:

Small: 0.2

Medium: 0.5

Large: 0.8

This section wraps up with:

ARIMA modeling

Stationarity & Unit Root tests (ADF)

Residual analysis

Advanced diagnostics

Summary & references

12.1 7. Advanced Statistical Concepts

12.1.1 7.1 Stationarity and Unit Root Testing

A **stationary time series** has constant mean and variance over time. Its essential for: - Forecasting - Valid modeling - Avoiding spurious regression

12.1.1.1 Unit Root: Augmented Dickey-Fuller (ADF) Test

library(tseries) set.seed(42) x <- cumsum(rnorm(100)) # non-stationary random walk plot.ts(x, main = "Simulated Random Walk")

adf.test(x) # Likely non-stationary (p > 0.05) 7.2 Detrending Time Series

t <- time(x) trend_model <- lm(x ~ t) resid_trend <- resid(trend_model) plot(resid_trend, main = "Detrended Series") adf.test(resid_trend) # Residuals should now be stationary 7.3 ARIMA Modeling Autoregressive Integrated Moving Average AR(p): Autoregression

I(d): Differencing

MA(q): Moving average

library(forecast) auto.arima(AirPassengers) Full Workflow

tsdata <- AirPassengers plot(tsdata)

13 Step 1: Stationarity check

adf.test(tsdata) # May need differencing

14 Step 2: Model Selection

fit <- auto.arima(tsdata) summary(fit)

15 Step 3: Forecasting

fc <- forecast(fit, h = 12) plot(fc) 7.4 ACF & PACF Plots Used for identifying model orders:

acf(diff(log(AirPassengers))) pacf(diff(log(AirPassengers))) 7.5 Residual Diagnostics

checkresiduals(fit) # From forecast package Box.test(residuals(fit), lag = 20, type = "Ljung-Box") 7.6 Forecast Accuracy

actuals <- window(AirPassengers, start = c(1960,1)) preds <- forecast(fit, h = 12)$mean accuracy(preds, actuals) 8. Summary This eBook covered advanced Week 7 content with practical R implementation:

Topic Key Concepts & Tools Time Series Analysis TSA, decomposition, ADF test, ARIMA Conditional Probability Bayes theorem, real-life problems Expected Value Joint PMFs, linearity of expectation Discrete Distributions Poisson, Hypergeometric, Negative Binomial Forecasting Techniques SMA, EMA, ETS, ARIMA Bayesian Applications Posterior inference, medical testing Model Evaluation AIC, BIC, RMSE, MAPE, residuals

16 Week 8

16.1 1. Introduction

This module explores the powerful integration of visual analytics and statistical reasoning. While traditional models often rely on tabular outputs, the **flexplot** package and similar tools highlight the importance of **graphical modeling**, especially in response to the **replication crisis**. The week also emphasizes how GUIs like **RKWard** and **RStudio** serve different user bases for statistical analysis.

16.2 2. Effect Size and Cohen's d

Effect size quantifies the **magnitude** of the difference, independent of sample size. One of the most common effect size measures is **Cohen's d**, which compares two means.

16.2.1 Interpretation of d:

d	Meaning
0.2	Small effect
0.5	Medium effect
0.8	Large effect

16.2.2 R Code Example (Cohen's d)

16.3 Load required package

install.packages("lsr") library(lsr)

16.4 Load your data (CSV format)

my.csv.data <- read.csv("yourdata.csv")

16.5 Independent groups Cohen's d

lsr::cohensD(my.csv.data[["CSE_1"]], my.csv.data[["CSE_2"]])

16.6 One-sample mean vs population mean

lsr::cohensD(my.csv.data[["CSE_1"]], mu = 3.9)   Practical Use: Effect size helps you understand practical significance, especially in behavioral research where p-values alone are insufficient.

Note: Effect sizes should always accompany inferential statistics to avoid overreliance on significance testing.

3. Understanding flexplot: Graphical Statistical Modeling The flexplot package allows for intuitive, formula-driven visual modeling. It uses GLM-style formulas like y ~ x1 + x2, bringing clarity between statistical models and their graphical representations.

Key Features: Visualize univariate, bivariate, and multivariate models

Supports linear, logistic, and mixed models

Matches graphical output directly with statistical models

Requires only one line of code for most use cases

Installation and Setup:

install.packages("flexplot") library(flexplot) library(cowplot) # For arranging multiple plots   More Coming in Part 2: Univariate & Bivariate flexplot() demos

GLM integration

Paneling, Ghost Lines, Beeswarm Visuals

Overlap handling and jitter control

16.7 4. Using flexplot: Examples and Best Practices

16.7.1 4.1 Univariate Visualization

flexplot(CSE_1 ~ 1, data = my.csv.data) Plots raw data (jittered) with mean overlay

Useful for outlier detection and distribution shape

4.2 Bivariate Continuous vs Categorical

16.8 Visualizing continuous DV vs categorical IV

flexplot(CSE_1 ~ Gender, data = my.csv.data) Automatically creates beeswarm or violin plots

Overlay: mean ± error bars

Jitter is used to prevent overlap of points

4.3 Continuous DV vs Continuous IV

flexplot(CSE_1 ~ Age, data = my.csv.data) Shows scatterplot + best-fit line

Adds error ribbons

Outliers stand out visually

4.4 Multiple Predictors (Additive Models)

flexplot(CSE_1 ~ Age + Gender, data = my.csv.data) Panels by Gender

Linear fits across Age

Helps uncover interaction

4.5 Logistic Regression Visualization

16.9 Convert pass/fail variable to factor

my.csv.data$Pass < -as.factor(my.csv.data$Pass)

16.10 Logistic visualization

flexplot(Pass ~ Hours, data = my.csv.data, family = "binomial") 5. RKWard vs RStudio: Interface & Functionality Feature RKWard RStudio Target Users Beginners, GUI-centric Coders, devs, advanced users Data Handling Spreadsheet-like Tidyverse-friendly Plots Auto-generated via dialogs ggplot2 required manually Statistical Models GUI for t-tests, ANOVA Syntax for all models

Conclusion: RKWard is ideal for non-programmers, while RStudio is better for reproducible analysis via code and markdown.

6. Cognitive Fit and Visual Communication Flexplot builds upon Cognitive Fit Theory — visual representations should match the task and viewer's expectation.

Key Graph Types in flexplot Type Best For Beeswarm Small-to-medium samples Violin Density + mean overlay Ghost Lines Slope visualization across panels Panels 2–3 categorical moderators

7. Advanced Flexplot Controls 7.1 Ghost Lines for Slope Tracking

flexplot(mpg ~ wt + cyl, data = mtcars) Panels by cyl

Gray reference slope: overall trend

Colored slope: panel-specific

7.2 Model Overlays

flexplot(CSE_1 ~ CSE_2 + Gender, data = my.csv.data) Adds regression lines

Includes model summaries in plot captions

7.3 Added Plot (Influence Visualization)

model <- lm(CSE_1 ~ CSE_2 + Age, data = my.csv.data) added.plot(model) Visualizes the unique contribution of predictors

Residual scatter by regressor

8. Association, AVPs, and Repeated Measures 8.1 Visualizing Correlation

flexplot(mpg ~ hp, data = mtcars) Adds correlation line

Includes Pearson's r

8.2 Repeated Measures (Paneling)

flexplot(score ~ time + condition, data = repeated_df) Each condition as panel

Time as predictor

Fits separate lines

8.3 Binned Paneling (Continuous Moderators)

flexplot(CSE_1 ~ Age + Income, data = my.csv.data) Age: X-axis

Income: Panel bins (equal-width)

Visualizes moderation effects

8.4 Jitter, Transparency, Point Customization

flexplot(CSE_1 ~ Age + Gender, data = my.csv.data, jitter = 0.3, alpha = 0.5, point.size = 2) 9. Interactive Plots and R Markdown Integration

install.packages("plotly") library(plotly) p <- flexplot(mpg ~ wt + cyl, data = mtcars) ggplotly(p) # Adds interactivity Quarto Embedding markdown

flexplot(CSE_1 ~ Gender, data = my.csv.data)

16.11 What's Next in Part 3?

- Full-scale simulation for effect size
- Reproducible workflows
- Custom function design
- Summary + export instructions

This final section includes:

Simulation for effect size

Custom model visuals

Reproducible workflows

Summary + rendering/export notes

16.12 10. Simulation: Effect Size and Visual Inference

16.12.1 10.1 Simulate Cohen's d with Flexplot

set.seed(123) group1 <- rnorm(50, mean = 5, sd = 1) group2 <- rnorm(50, mean = 6.2, sd = 1)

group <- factor(rep(c("A", "B"), each = 50)) score <- c(group1, group2)

sim_df <- data.frame(group, score)

library(lsr) cohensD(score ~ group, data = sim_df) # Should return d 1.2

flexplot(score ~ group, data = sim_df) 10.2 Power and Confidence Visualization

library(pwr) pwr.t.test(d = 0.8, power = 0.8, sig.level = 0.05, type = "two.sample") 10.3 Monte Carlo Effect Size Estimation

sim_d <- replicate(1000, { g1 <- rnorm(30, 5, 1) g2 <- rnorm(30, 6, 1) cohensD(g1, g2) })

hist(sim_d, breaks = 50, col = "lightblue", main = "Simulated Cohen's d Distribution") abline(v = mean(sim_d), col = "red") 11. Visual Inference in Teaching Overlay Raw + Model Together

flexplot(mpg ~ wt + cyl, data = mtcars) Cyl = Panel

Gray slope = overall

Color slope = per panel

R^2 and p-values appear below

12. Workflow: Reproducible Visual Analytics in R 12.1 Data Import

df <- read.csv("CSE_scores.csv") str(df) 12.2 Visualization Plan Start with flexplot()

Panel by categorical moderators

Add continuous predictors

Use added.plot() to show incremental effect

Report both visualization + model summary

12.3 R Markdown Report markdown

library(flexplot) flexplot(score ~ gender + age, data = df)

16.13 13. Model Summary with Visual + Numeric Layers

model <- lm(CSE_1 ~ CSE_2 + Age + Gender, data = my.csv.data) summary(model)

added.plot(model) # Visual version of unique effect 14. Combining flexplot with ggplot2

p1 <- flexplot(CSE_1 ~ CSE_2 + Gender, data = my.csv.data) p2 <- ggplot(my.csv.data, aes(CSE_2, CSE_1)) + geom_point() + geom_smooth(method = "lm")

cowplot::plot_grid(p1, p2, labels = c("Flexplot", "GGplot")) 15. Summary Concept Tool Used Effect Size cohensD() from lsr Graphical Modeling flexplot() Simulation Monte Carlo Association Plot Slope Panels Influence Plot added.plot() Interactive Graphs ggplotly()