PLS 120: Applied Statistics in Agricultural Sciences

Probability and Sampling

Week 4 Tutorial Guide

Mohammadreza Narimani
Department of Biological and Agricultural Engineering
University of California, Davis

mnarimani@ucdavis.edu

October 2025

Contents

Important Links

Essential Course Resources

Course Website

All course materials available at:

Course Website Link

Interactive Binder Environment

Access Week 4 lab materials:

Week 4 Binder Link

Welcome to Week 4: Probability and Sampling

This week, we explore **probability theory and sampling techniques** - essential foundations for statistical inference in agricultural research. You'll learn to simulate probability experiments, work with distributions, and understand randomness!

Logical Variables and Data Types

Understanding Logical Variables

Logical variables in R can only take the values TRUE, FALSE, or NA (missing value). They are fundamental in decision-making processes in programming.

3.1.1 Basic Logical Operations

Comparison Operators:
`==` - Equal to
`!=` - Not equal to
`<` - Less than
`>` - Greater than
`<=` - Less than or equal to
`>=` - Greater than or equal to

Logical Operators:
`&` - AND (element-wise)
`|` - OR (element-wise)
`!` - NOT (negation)

Data Type Conversion

3.2.1 Essential Conversion Functions

Type Conversion Functions:
`as.numeric(x)` - Convert to numeric
`as.character(x)` - Convert to character
`as.factor(x)` - Convert to factor
`as.logical(x)` - Convert to logical
`data.frame(x)` - Convert to data frame

Example:
```
numeric_vector <- c(0, 1, 2)
logical_vector <- as.logical(numeric_vector)
# Result:  FALSE, TRUE, TRUE
```

Random Sampling Techniques

The sample() Function

Understanding how to perform random sampling from a population is essential for statistical analysis.

4.1.1 Function Parameters

Syntax: `sample(x, size, replace)`

Parameters:
`x` - Population (dataset) to sample from
`size` - Number of samples to draw
`replace` - TRUE (with replacement) or FALSE (without replacement)

Examples:
`sample(1:6, 10, replace = TRUE)` - Roll die 10 times
`sample(nrow(data), 30, replace = FALSE)` - Select 30 unique rows

4.1.2 Reproducible Results

set.seed() Function:
Use `set.seed(number)` before random sampling to ensure reproducible results

Example:
```
set.seed(123)
sample(c("H", "T"), 10, replace = TRUE)
# Will always produce the same sequence
```

Probability Simulation

Coin Toss Experiments

5.1.1 Basic Coin Simulation

Create a Coin: `coin <- c("H", "T")`

Simulate Tosses: `tosses <- sample(coin, size = 50, replace = TRUE)`

Count Outcomes:
```
heads <- sum(tosses == "H")
tails <- sum(tosses == "T")
```

Calculate Probabilities:
```
prob_heads <- heads / length(tosses)
prob_tails <- tails / length(tosses)
```

5.1.2 Frequency Analysis

Create Frequency Table: `toss_table <- table(tosses)`

Convert to Probabilities:
```
toss_probabilities <- toss_table / sum(toss_table)
```

Theoretical vs Experimental:
Theoretical probability for fair coin: $P(H) = P(T) = 0.5$
Experimental probability varies with sample size

Dice Roll Simulation

5.2.1 Single Die Experiments

Create a Die: `dice <- c(1:6)` or `dice <- seq(1, 6, 1)`

Simulate Rolls: `rolls <- sample(dice, size = 100, replace = TRUE)`

Analyze Results:
```
roll_counts <- table(rolls)
roll_probabilities <- roll_counts / sum(roll_counts)
```

Theoretical: Each face should have probability $= 1/6 \approx 0.167$

5.2.2 Two Dice and Central Limit Theorem

Sum of Two Dice:

```
die1 <- sample(1:6, 1000, replace = TRUE)
die2 <- sample(1:6, 1000, replace = TRUE)
sums <- die1 + die2
```

Observe Distribution:
As sample size increases, the distribution of sums approaches normal distribution (Central Limit Theorem)

Theoretical Mean: $E(\text{sum}) = 7$
Theoretical SD: $\sigma(\text{sum}) \approx 2.42$

Normal Distribution Functions

Generating Random Normal Data

6.1.1 rnorm() Function

Purpose: Generate random numbers from normal distribution

Syntax: `rnorm(n, mean = 0, sd = 1)`

Parameters:
`n` - Number of random values to generate
`mean` - Mean of the distribution (default: 0)
`sd` - Standard deviation (default: 1)

Example: `normal_data <- rnorm(100, mean = 50, sd = 15)`

Probability Calculations

6.2.1 pnorm() Function

Purpose: Calculate cumulative probability (area under curve)

Syntax: `pnorm(q, mean = 0, sd = 1)`

Returns: $P(X \leq q)$ for normal distribution

Example:

```
prob_less_than_60 <- pnorm(60, mean = 50, sd = 10)
# Returns probability that X < 60
```

6.2.2 qnorm() Function

Purpose: Find quantiles (inverse of pnorm)

Syntax: `qnorm(p, mean = 0, sd = 1)`

Returns: Value x such that $P(X \leq x) = p$

Example:
```
value_at_90th <- qnorm(0.90, mean = 50, sd = 10)
# Returns value below which 90% of data falls
```

Visual Probability Functions

6.3.1 tigerstats Package

Enhanced Visualization:
```
library(tigerstats)

pnormGC(60, mean = 50, sd = 10, graph = TRUE)
# Shows probability with visual graph

qnormGC(0.90, mean = 50, sd = 10, graph = TRUE)
# Shows quantile with visual graph
```

Data Visualization for Probability

Creating Probability Plots

7.1.1 Bar Plots for Discrete Distributions

Base R Approach:
```
barplot(probability_table, main = "Probability Distribution")
```

ggplot2 Approach:
```
ggplot(data_frame, aes(x = outcome, y = probability)) +
geom_bar(stat = "identity")
```

Note: Use `stat = "identity"` to plot actual probability values

7.1.2 Histograms for Continuous Distributions

For Normal Data:

```
hist(normal_data, breaks = 15)
```

With ggplot2:

```
ggplot(data.frame(x = normal_data), aes(x = x)) +
geom_histogram(bins = 15)
```

Density Plots:

```
ggplot(data.frame(x = normal_data), aes(x = x)) +
geom_density()
```

Assignment 4 Overview

Assignment Structure (20 points total)

1. **Part 1: Simulation (6 points)**
   - Simulate 50 coin flips (3 points)
   - Simulate 50 dice rolls (3 points)
2. **Part 2: Probability Calculation (6 points)**
   - Calculate experimental probabilities for coin outcomes (2 points)
   - Calculate experimental probabilities for dice outcomes (3 points)
   - Compare experimental vs theoretical probabilities (1 point)
3. **Part 3: Data Frames and Visualization (8 points)**
   - Create coin probability data frame (2 points)
   - Create dice probability data frame (2 points)
   - Generate coin flip bar plot (2 points)
   - Generate dice roll bar plot (2 points)

Agricultural Applications

Real-World Applications:

- **Seed Germination Studies** - Model probability of germination success under different conditions
- **Weather Risk Assessment** - Simulate probability of drought, frost, or extreme weather events
- **Quality Control Sampling** - Random sampling of agricultural products for testing
- **Field Trial Design** - Understanding sampling variability in experimental plots
- **Pest Management** - Modeling probability distributions of pest occurrence
- **Crop Insurance** - Calculating risk probabilities for insurance premium determination

Key Concepts Summary

Probability Fundamentals

Basic Probability Rules:
- Probability ranges from 0 to 1
- P(Event) = Favorable outcomes / Total outcomes
- Sum of all probabilities = 1
- P(not A) = 1 - P(A)

Law of Large Numbers:
As sample size increases, experimental probability approaches theoretical probability

Sampling Concepts

Sampling with Replacement:
Each item can be selected multiple times (like rolling dice)

Sampling without Replacement:
Each item can only be selected once (like drawing cards without putting back)

Population vs Sample:
Population = entire group; Sample = subset of population

Getting Started

1. Launch Week 4 Binder environment
2. Navigate to `class_activity` folder
3. Open `Week4_Probability_Sampling.ipynb`

4. Work through interactive exercises
5. Complete Assignment 4 in `assignment` folder

Learning Objectives

By the end of this week, you will be able to:

- Understand logical variables and data type conversions
- Perform random sampling with and without replacement
- Simulate probability experiments (coins, dice)
- Work with normal distribution functions (rnorm, pnorm, qnorm)
- Compare experimental and theoretical probabilities
- Visualize probability distributions with bar plots and histograms
- Apply probability concepts to agricultural research scenarios

Tips for Success

Best Practices:

- Use `set.seed()` for reproducible random results
- Start with small sample sizes to understand concepts
- Always verify that probabilities sum to 1
- Compare experimental results to theoretical expectations
- Use visualization to understand probability distributions

Need Help?

Mohammadreza Narimani
Email: mnarimani@ucdavis.edu
Department of Biological and Agricultural Engineering, UC Davis
Office Hours: Thursdays 10 AM - 12 PM (Zoom)