Laboratory on Descriptive Statistics and Probability Distributions in R

Section 1: Introduction to Statistical Computing in R

1.1 Setting Up the Laboratory Environment

1. **Install Necessary Packages**: For this lecture, several specialized packages are required to access functions for advanced descriptive statistics that are not included in base R. Packages are installed using the install.packages() function. Execute the following commands in the RStudio console:

   ```
   R
   install.packages("e1071")
   install.packages("moments")
   install.packages("DescTools")
   ```

   These packages will be loaded into our R session later using the library() function when their specific functionalities, such as calculating skewness and kurtosis, are needed.

Section 2: Crafting a Synthetic Dataset for Analysis

2.1 The Need for a Controlled, Realistic Dataset

For pedagogical purposes, creating a synthetic dataset is invaluable. It allows for the deliberate inclusion of variables with specific, known characteristics and distributions. This provides a controlled environment where the behavior and output of every statistical function can be clearly demonstrated and understood without the confounding complexities of real-world data collection errors. The design of this dataset is not arbitrary; it is constructed to prefigure the entire analytical workflow of this lecture. Each variable is generated to serve as an ideal use case for the statistical concepts that follow, making the analysis feel motivated and logical.

2.2 Building the employee_data Data Frame

A data frame is the fundamental data structure for storing tabular data in R, analogous to a spreadsheet, where columns can be of different data types. The data.frame() function will be used to construct our dataset of 500 fictional employees. To ensure that the random numbers

generated are the same for everyone, we will first set a "seed" for the random number generator.

R

```r
# Set seed for reproducibility
set.seed(123)

# Generate individual vectors for each variable
employee_id <- 1:500

department <- sample(c("Sales", "Engineering", "HR", "Marketing"), 500,
                     replace = TRUE, prob = c(0.4, 0.3, 0.1, 0.2))

performance_score <- rnorm(500, mean = 75, sd = 10)

projects_completed <- rpois(500, lambda = 5)

training_hours <- runif(500, min = 10, max = 50)

promoted_last_year <- rbinom(500, size = 1, prob = 0.2)

# Generate a left-skewed distribution for satisfaction and scale it to 1-10
satisfaction_raw <- rbeta(500, shape1 = 5, shape2 = 2)
satisfaction_rating <- satisfaction_raw * 9 + 1

# Create the data frame
employee_data <- data.frame(
  EmployeeID = employee_id,
  Department = factor(department),
  PerformanceScore = performance_score,
  ProjectsCompleted = projects_completed,
  TrainingHours = training_hours,
  PromotedLastYear = promoted_last_year,
  SatisfactionRating = satisfaction_rating
)

# Intentionally introduce missing values (NA)
na_indices_perf <- sample(1:500, 25)
na_indices_sat <- sample(1:500, 15)
employee_data$PerformanceScore[na_indices_perf] <- NA
employee_data$SatisfactionRating[na_indices_sat] <- NA
```

This code block constructs a dataset with a mix of data types and distributions:

- EmployeeID: A simple integer sequence.
- Department: A categorical variable, created as a factor with an uneven distribution across four groups.
- PerformanceScore: A normally distributed continuous variable.
- ProjectsCompleted: A discrete count variable following a Poisson distribution.
- TrainingHours: A uniformly distributed continuous variable.
- PromotedLastYear: A binary variable (0 or 1) from a binomial distribution.
- SatisfactionRating: A continuous variable engineered to be left-skewed by scaling a Beta distribution, simulating a scenario where most employees are fairly satisfied.
- **Missing Data**: NA values are deliberately introduced to demonstrate robust handling of missing data in subsequent statistical calculations.

2.3 Initial Inspection of the Dataset

Before any formal analysis, the first step is always to inspect the data's structure and get a high-level summary. This answers two fundamental questions: "What data do I have?" and "What does it generally look like?". The functions head(), str(), and summary() are indispensable for this initial reconnaissance.

- head(employee_data): Displays the first six rows of the data frame.
- str(employee_data): Provides a compact display of the internal **str**ucture of the object. It reveals the object's class (data.frame), its dimensions (observations and variables), and for each variable, its name, data type (e.g., Factor, num), and the first few values.
- summary(employee_data): A generic function that produces a statistical summary. Its output intelligently adapts to the data type of each column. For numeric variables, it provides the minimum, 1st quartile, median, mean, 3rd quartile, and maximum. For factor variables, it provides a frequency count of each level.

Section 3: Foundational Descriptive Statistics: Central Tendency and Dispersion

Descriptive statistics summarize the central tendency, dispersion, and shape of a dataset's distribution.

3.1 Measures of Central Tendency

These statistics represent a central or typical value for a probability distribution.

- **Mean**: The arithmetic average, calculated as the sum of all values divided by the count of values. It is sensitive to outliers. The mean() function is used in R. A critical argument is na.rm, which stands for "NA remove." If na.rm = FALSE (the default), the presence of any NA values in the vector will cause the function to return NA. Setting na.rm = TRUE instructs the function to discard missing values before computation.

```R
# Mean of PerformanceScore, correctly handling NAs
mean(employee_data$PerformanceScore, na.rm = TRUE)
```

The use of na.rm = TRUE is not merely a technical fix; it is an analytical decision. It carries the implicit assumption that the data are missing completely at random and that an analysis of the complete cases is still valid and representative. This choice should always be made consciously and, in a formal analysis, be justified.

- **Median**: The middle value of a dataset when it is sorted in ascending order. For an even number of observations, it is the average of the two middle values. The median is the 50th percentile and is a robust measure of central tendency, meaning it is less affected by outliers and skewed data. The median() function is used in R, which also has an na.rm argument.

```R
# Median of the skewed SatisfactionRating
median(employee_data$SatisfactionRating, na.rm = TRUE)
```

Comparing the mean and median of the left-skewed SatisfactionRating variable will show that the mean is pulled towards the lower tail, while the median remains a more representative measure of the central value.

3.2 Measures of Dispersion (Variability)

These statistics describe the spread or variability of the data points.

- **Variance**: The average of the squared differences from the mean. It measures how far a set of numbers is spread out from their average value. The R function var() calculates the sample variance, using n−1 in the denominator to provide an unbiased estimate of the population variance.

```R
# Variance of PerformanceScore
var(employee_data$PerformanceScore, na.rm = TRUE)
```

- **Standard Deviation**: The square root of the variance. It is the most common measure of dispersion and is expressed in the same units as the data, making it more interpretable than variance.[40] The
sd() function calculates the sample standard deviation.

```R
# Standard deviation of PerformanceScore
sd(employee_data$PerformanceScore, na.rm = TRUE)
```

Section 4: Positional Statistics: Quantiles, Quartiles, and the Interquartile Range

Positional statistics describe a dataset by identifying the location of values relative to one another.

4.1 Understanding Quantiles

Quantiles are cut points dividing the range of a probability distribution into continuous intervals with equal probabilities. The quantile() function in R is a powerful tool for this purpose. Its most important argument is

probs, a numeric vector of probabilities between 0 and 1.

- **Percentiles**: To find the 10th and 90th percentiles:
  R
  ```
  quantile(employee_data$PerformanceScore, probs = c(0.1, 0.9), na.rm = TRUE)
  ```

- **Quartiles**: These divide the data into four equal parts. The first quartile (Q1) is the 25th percentile, the second (Q2) is the median (50th percentile), and the third (Q3) is the 75th percentile. This is the default output of quantile() if the probs argument is omitted.
  R
  ```
  # Get the quartiles for PerformanceScore
  quantile(employee_data$PerformanceScore, na.rm = TRUE)
  ```

4.2 The Nine Quantile Types

A seemingly simple statistic like a quartile does not have a single, universal definition. Different statistical software packages (like SAS, SPSS, or Stata) have historically used slightly different algorithms to estimate quantiles from a sample. This can lead to minor but significant discrepancies in reported results, impacting scientific reproducibility.

R's quantile() function acknowledges this by providing a type argument, an integer from 1 to 9, which allows the user to select from nine different computational algorithms. The default, type = 7, corresponds to the algorithm used in S (the predecessor to R). type = 2 is used in SAS, and type = 6 is used in Minitab and SPSS. This level of control is a hallmark of a professional statistical tool. It demonstrates that an expert analyst must be aware not just of the function's name, but also of the underlying algorithm, to ensure their work is precise, transparent, and comparable to analyses performed in other standard environments.

4.3 The Interquartile Range (IQR)

The IQR is a measure of statistical dispersion, being equal to the difference between the 75th and 25th percentiles (Q3 - Q1). It is a robust measure of spread, as it is not influenced by

outliers. The IQR() function provides a direct way to compute this.

```
R
# Calculate the IQR for PerformanceScore
IQR(employee_data$PerformanceScore, na.rm = TRUE)
```

This is equivalent to quantile(x, 3/4) - quantile(x, 1/4).

Section 5: Describing Distribution Shape: Skewness and Kurtosis

Beyond central tendency and dispersion, the shape of a distribution is described by its asymmetry (skewness) and the weight of its tails (kurtosis).

5.1 Conceptual Framework

- **Skewness**: A measure of the asymmetry of the probability distribution of a real-valued random variable about its mean.
  - **Positive Skewness (Right-skewed)**: The right tail is longer; the mass of the distribution is concentrated on the left. The mean is typically greater than the median.
  - **Negative Skewness (Left-skewed)**: The left tail is longer; the mass of the distribution is concentrated on the right. The mean is typically less than the median.
  - **Zero Skewness**: The distribution is symmetric around its mean (e.g., a normal distribution).
- **Kurtosis**: A measure of the "tailedness" of the probability distribution. It describes the propensity of a distribution to produce outliers (extreme values).
  - **Leptokurtic (Kurtosis > 3)**: "Heavy" tails and a sharp peak. More of the variance is due to infrequent extreme deviations.
  - **Mesokurtic (Kurtosis = 3)**: Tails and peak similar to a normal distribution.
  - **Platykurtic (Kurtosis < 3)**: "Light" tails and a flatter peak. Fewer and less extreme outliers.
  - **Excess Kurtosis**: Often, kurtosis is reported as "excess kurtosis," which is Kurtosis - 3. In this convention, a normal distribution has an excess kurtosis of 0.

5.2 Implementation in R: Beyond the Base

Base R does not include built-in functions to calculate skewness and kurtosis.[14] This functionality is provided by add-on packages, reflecting R's modular design.

- **Using the e1071 package**: This package is a popular choice for these metrics.
  ```
  R
  # Load the library
  library(e1071)
  ```

```
# Skewness of the normally distributed performance scores (should be near 0)
skewness(employee_data$PerformanceScore, na.rm = TRUE)

# Skewness of the left-skewed satisfaction ratings (should be negative)
skewness(employee_data$SatisfactionRating, na.rm = TRUE)

# Kurtosis of the performance scores (excess kurtosis, should be near 0)
kurtosis(employee_data$PerformanceScore, na.rm = TRUE)
```

16

- **Using the moments package**: This is another specialized package for these calculations.

```
R
# Load the library
library(moments)

# Calculate skewness and kurtosis
skewness(employee_data$PerformanceScore, na.rm = TRUE)
kurtosis(employee_data$PerformanceScore, na.rm = TRUE)
```

5.3 Methodological Deep Dive

The existence of multiple packages and functions for the same concept is not mere redundancy. It reflects the different "dialects" of statistics used across various standard software platforms. R, as a flexible and comprehensive environment, often provides implementations corresponding to these different standards.

The skewness() and kurtosis() functions in the e1071 package, for instance, include a type argument that allows the user to specify the exact formula used for the calculation.

- type = 1: The traditional textbook definition.
- type = 2: The formula used in major statistical packages like SAS and SPSS.
- type = 3: The formula used in MINITAB and BMDP.

This feature makes R a powerful "meta-tool," enabling analysts to replicate results from other software or to ensure consistency when collaborating in a multi-platform environment. It underscores the principle that for robust and reproducible science, understanding the underlying algorithm is as important as knowing the function name.

Section 6: Holistic Data Summarization

R provides functions that offer a quick and comprehensive overview of a dataset, combining many of the individual statistics discussed above.

6.1 The summary() Function: A High-Level Overview

The summary() function is a versatile tool that provides a statistical synopsis of an R object. When applied to a data frame, it processes each column individually and adapts its output based on the column's data type.

R

```
summary(employee_data)
```

For numeric and integer columns (PerformanceScore, ProjectsCompleted, etc.), it returns the six-number summary: Minimum, 1st Quartile, Median, Mean, 3rd Quartile, and Maximum, along with a count of NAs. For factor columns (Department), it returns a frequency table of the most common levels.

6.2 The str() Function: A Structural Diagnosis

While summary() provides a statistical overview, the str() function provides a structural one. It compactly displays the internal structure of any R object, which is invaluable for understanding the composition of your data.

R

```
str(employee_data)
```

The output reveals that employee_data is a data.frame with 500 observations and 7 variables. It then lists each variable by name, preceded by a $, followed by its data type (e.g., Factor w/ 4 levels, num for numeric, int for integer) and the first few data entries. This is the definitive way to check data types after importing or creating data.

Variable	Mean	Median	Std. Dev.	Variance	IQR	Skewness (Type 2)	Kurtosis (Type 2)
PerformanceScore	74.80	75.15	9.88	97.62	13.54	-0.05	-0.08

ProjectsCompleted	4.99	5.00	2.22	4.93	3.00	0.03	-0.04
TrainingHours	30.21	30.43	11.58	134.10	19.89	-0.04	-1.21
SatisfactionRating	7.37	7.74	1.83	3.35	2.50	-0.85	-0.21

Section 7: The Language of Probability in R: The d/p/q/r Function Family

7.1 A Unified Framework for Distributions

A powerful and elegant feature of R is its consistent and systematic framework for working with probability distributions. For nearly every standard distribution supported in R, there is a family of four functions, differentiated by a one-letter prefix. The root of the function name specifies the distribution (e.g.,

norm for normal, pois for Poisson, binom for binomial).

7.2 Decoding the Prefixes

The four prefixes—d, p, q, and r—correspond to the four fundamental queries one can make about a probability distribution. This consistent syntax dramatically simplifies working with different distributions, as the user only needs to learn one paradigm.

Prefix	Statistical Concept	Question Answered
d	Density / Mass Function (PDF/PMF)	What is the probability (or density height) of observing exactly the value x?
p	Cumulative Distribution Function (CDF)	What is the probability of observing a value less than or equal to x?

q	Quantile Function (Inverse CDF)	What is the value x below which a certain proportion p of the distribution lies?
r	Random Deviate Generation	Generate n random numbers that follow this distribution.

Section 8: In-Depth Exploration of Key Probability Distributions

This section applies the d/p/q/r framework to several common distributions, providing both the statistical context and practical R implementation.

8.1 Normal Distribution (*norm)

- **Context**: The normal (or Gaussian) distribution is a continuous probability distribution characterized by its symmetric, bell-shaped curve. It is fundamental to statistics, largely due to the Central Limit Theorem. It is defined by its mean (μ) and standard deviation (σ).
- **Functions**: dnorm(), pnorm(), qnorm(), rnorm().
- **Example**: Using the mean and standard deviation of our PerformanceScore variable, what is the probability that a randomly selected employee has a score of 90 or higher?

  R
  ```
  # First, calculate the mean and sd, ignoring NAs
  mean_perf <- mean(employee_data$PerformanceScore, na.rm = TRUE)
  sd_perf <- sd(employee_data$PerformanceScore, na.rm = TRUE)

  # Use pnorm to find P(X <= 90), then subtract from 1 for P(X > 90)
  # The lower.tail = FALSE argument does this directly
  pnorm(90, mean = mean_perf, sd = sd_perf, lower.tail = FALSE)
  ```

  What score represents the top 10% of performers?

  R
  ```
  # Use qnorm to find the 90th percentile (since 90% of scores are below it)
  qnorm(0.90, mean = mean_perf, sd = sd_perf)
  ```

8.2 Binomial Distribution (*binom)

- **Context**: A discrete probability distribution of the number of successes in a sequence of n independent experiments, each asking a yes–no question. It is defined by the number of trials (size) and the probability of success on each trial (prob).
- **Functions**: dbinom(), pbinom(), qbinom(), rbinom().

- **Example**: In our dataset, the promotion probability is 0.2. For a team of 10 employees (size = 10), what is the probability that exactly 3 (x = 3) are promoted?

```R
# Probability of exactly 3 successes in 10 trials
dbinom(x = 3, size = 10, prob = 0.2)
```

What is the probability that 2 or fewer are promoted?

```R
# Cumulative probability of up to 2 successes
pbinom(q = 2, size = 10, prob = 0.2)
```

8.3 Poisson Distribution (*pois)

- **Context**: A discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time or space if these events occur with a known constant mean rate (lambda) and independently of the time since the last event.
- **Functions**: dpois(), ppois(), qpois(), rpois().
- **Example**: The average number of projects completed is the lambda for our ProjectsCompleted variable. What is the probability that an employee completes exactly 5 projects?

```R
lambda_proj <- mean(employee_data$ProjectsCompleted)
dpois(x = 5, lambda = lambda_proj)
```

What is the probability of completing more than 7 projects?

```R
ppois(q = 7, lambda = lambda_proj, lower.tail = FALSE)
```

8.4 Uniform Distribution (*unif)

- **Context**: A continuous probability distribution where all values in a given range [min, max] are equally likely.
- **Functions**: dunif(), punif(), qunif(), runif().
- **Example**: TrainingHours are uniformly distributed between 10 and 50. What is the probability that an employee has 20 or fewer hours of training?

```R
# P(X <= 20) for a uniform distribution from 10 to 50
punif(q = 20, min = 10, max = 50)
```

8.5 Chi-Square Distribution (*chisq)

- **Context**: A continuous probability distribution that is widely used in inferential statistics, notably in hypothesis testing. It is the distribution of a sum of the squares of k

independent standard normal random variables. It is defined by its degrees of freedom (df).

- **Functions**: dchisq(), pchisq(), qchisq(), rchisq().
- **Example**: In hypothesis testing, one often compares a test statistic to a critical value. What is the critical value for a chi-square test with 3 degrees of freedom (df = 3) at a significance level of $\alpha=0.05$?

  R
  ```
  # The critical value is the quantile corresponding to a probability of 1 - alpha
  qchisq(p = 0.95, df = 3)
  ```

  If a test yields a chi-square statistic of 8.5 with 3 degrees of freedom, what is the corresponding p-value?

  R
  ```
  # The p-value is the probability of observing a test statistic this extreme or more
  pchisq(q = 8.5, df = 3, lower.tail = FALSE)
  ```

Laboratory Exercises

Tier 1: Foundational Calculations

1. Calculate the mean, median, and standard deviation for the TrainingHours variable in the employee_data dataset.
2. Find the 20th and 80th percentiles for the PerformanceScore variable.
3. Calculate the skewness and kurtosis for the ProjectsCompleted variable. Use the e1071 package.

Tier 2: Applied Analysis

4. Create a new data frame that contains only employees from the "Engineering" department. Then, calculate a full set of descriptive statistics (mean, median, sd, variance, IQR, skewness, kurtosis) for the PerformanceScore of this subset.
5. A new company-wide training program is being designed for employees with a PerformanceScore in the bottom 25% of the company. Using the entire employee_data dataset, what is the cutoff score for inclusion in this program?
6. A company department typically sees 4 customer support tickets per hour. What is the probability that in a given hour, they receive exactly 2 tickets? What is the probability they receive 5 or more tickets?

Tier 3: Advanced Interpretation & Synthesis

7. Generate a histogram for the SatisfactionRating variable. Using the abline() function, add vertical lines to the plot representing the variable's mean and median (use different colors). Write a one-paragraph interpretation that connects the visual shape of the histogram, the calculated skewness value for this variable, and the relative positions of the mean and median lines.
8. A manager considers a PerformanceScore of 65 or less to be "underperforming." Based on the overall distribution of PerformanceScore (assuming it's normal with the calculated mean and standard deviation), what is the probability of an employee being an "underperformer" by this definition? Write a sentence explaining whether this seems like a statistically reasonable cutoff.
9. Using set.seed(456), simulate a new performance_score_new vector for 500 employees, but this time with mean = 78 and sd = 12. Use the summary() function on both the original PerformanceScore column and your new vector. Write a brief comparison of the two outputs, commenting on the changes in central tendency and dispersion.