Statistical Machine Learning For Data Science

Subject code: **BAD702** IA Marks: **50**

Hour/week: **02** Total Hours: **24**

Course objectives:

- Understand Exploratory Data Analysis
- Explain Data and Sampling Distributions
- To Analyze Statistical experiments and perform significance testing
- To demonstrate how to perform regression analysis on the data
- Explain Discriminant Analysis on the data.

8	A researcher conducts an experiment with a sample of 20 participants to determine if a new drug affects heart rate. The sample has a mean heart rate increase of 8 beats per minute and a standard deviation of 2 beats per minute. Perform a hypothesis test using the t-distribution to determine if the mean heart rate increase is significantly different from zero at the 5% significance level	25-28
9	A company is testing two versions of a webpage (A and B) to determine which version leads to more sales. Version A was shown to 1,000 users and resulted in 120 sales. Version B was shown to 1,200 users and resulted in 150 sales. Perform an A/B test to determine if there is a statistically significant difference in the conversion rates between the two versions. Use a 5% significance level.	29-31
10	You are comparing the average daily sales between two stores. Store A has a mean daily sales value of $1,000 with a standard deviation of $100 over 30 days, and Store B has a mean daily sales value of $950 with a standard deviation of $120 over 30 days. Conduct a two-sample t-test to determine if there is a significant difference between the average sales of the two stores at the 5% significance level.	32-34
11	A company collects data on employees' salaries and records their education level as a categorical variable with three levels: "High School", "Bachelor's", and "Master's". Fit a multiple linear regression model to predict salary using education level (as a factor variable) and years of experience. Interpret the coefficients for the education levels in the regression model.	35-37
12	You have data on housing prices and square footage and notice that the relationship between square footage and price is nonlinear. Fit a spline regression model to allow the relationship between square footage and price to change at 2,000 square feet. Explain how spline regression can capture different behaviours of the relationship before and after 2,000 square feet.	38-40
13	A hospital is using a Poisson regression model (a type of GLM) to predict the number of emergency room visits per week based on patient age and medical history. The model is given by: $\text{Log}(\lambda) = 2.5 - 0.03*\text{Age} + 0.5*\text{condition}$ where λ is the expected number of visits per week, Age is the patient's age, and condition is a binary variable (1 if the patient has a chronic condition, 0 otherwise). Interpret the coefficients of Age and condition. What is the expected number of visits per week for a 60-year-old patient with a chronic condition? How would the expected number of visits change if the patient did not have a chronic condition?	41-44
14	A bakery claims that its new cookie recipe is lower in calories compared to the old recipe, which had a mean calorie count of 200. You sample 40 new cookies and find a mean of 190 calories with a standard deviation of 15 calories. Perform a one-tailed t-test to determine if the new recipe has significantly fewer calories at a 5% significance level.	45-47

1. INTRODUCTION

Introduction to R programming:

R is a programming language and free software developed by Ross Ihaka and Robert Gentleman in 1993. R possesses an extensive catalog of statistical and graphical methods. It includes machine learning algorithms, linear regression, time series, statistical inference to name a few.

Most of the R libraries are written in R, but for heavy computational tasks, C, C++ and Fortran codes are preferred. R is not only entrusted by academic, but many large companies also use R programming language, including Uber, Google, Airbnb, Facebook and so on.

Data analysis with R is done in a series of steps; programming, transforming, discovering, modeling and communicate the results.

Program: R is a clear and accessible programming tool
Transform: R is made up of a collection of libraries designed specifically for data science
Discover: Investigate the data, refine your hypothesis and analyze them
Model: R provides a wide array of tools to capture the right model for your data
Communicate: Integrate codes, graphs, and outputs to a report with R Markdown or build Shiny apps to share with the world.

What is R used for?

- Statistical inference
- Data analysis
- Machine learning algorithm

2. Download and install R-Programming environment :

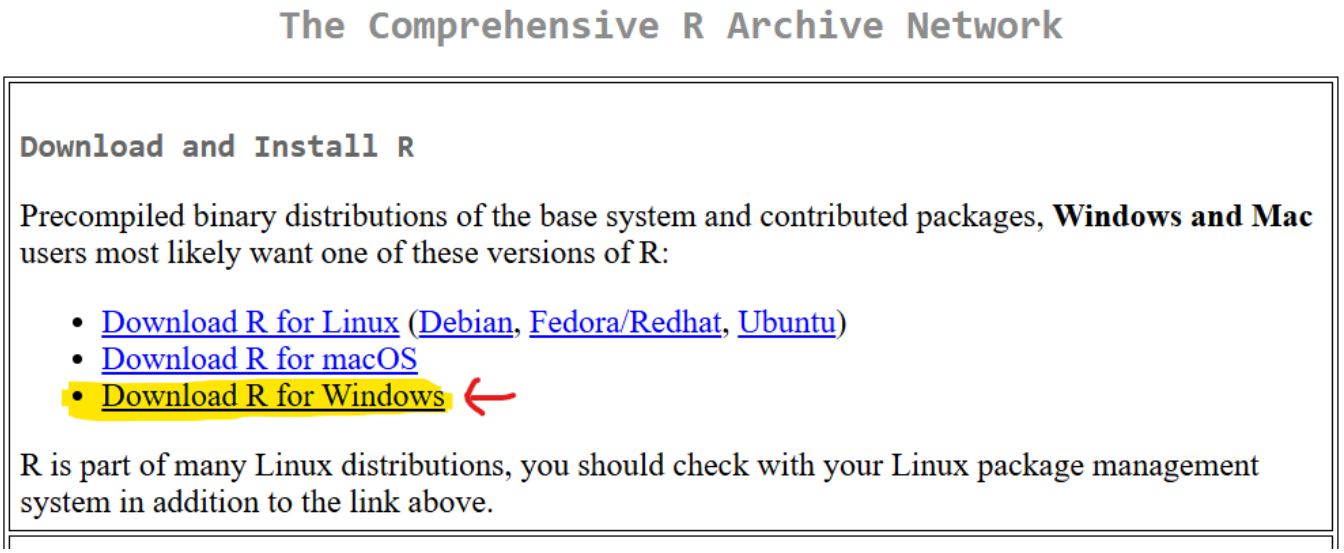
R programming is a very popular language and to work on that we have to install RGui.

Installing R to the local computer is very easy. First, we must know which operating system we are using so that we can download it accordingly. The official site https://cloud.r-project.org provides installer files for major operating systems including Windows, Linux, and Mac OS.

Install R in Windows:
To install R on your Windows, just follow these steps:

1. Download the R Installer File
2. Run the Installer
3. Install R
4. Add R to the PATH Environment Variable Manually
5. Verify the Installation

Step 1: Download the R Installer File
Go to the official R Project website (https://cloud.r-project.org) and download the latest version of R for Windows.



To proceed, please click the link labelled “Download R for Windows”.

R for Windows

Subdirectories:

base	Binaries for base distribution. This is what you want to **install R for the first time**.
contrib	Binaries of contributed CRAN packages (for R >= 4.0.x).
old contrib	Binaries of contributed CRAN packages for outdated versions of R (for R < 4.0.x).
Rtools	Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the R FAQ and R for Windows FAQ.

To proceed, please click the link labelled "install R for the first time."

R-4.4.2 for Windows

Download R-4.4.2 for Windows (83 megabytes, 64 bit)

README on the Windows binary distribution
New features in this version

This build requires UCRT, which is part of Windows since Windows 10 and Windows Server 2016. On older systems, UCRT has to be installed manually from here.

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the md5sum of the .exe to the fingerprint on the master server.

To proceed, click the link labelled Download R-4.4.2 for Windows

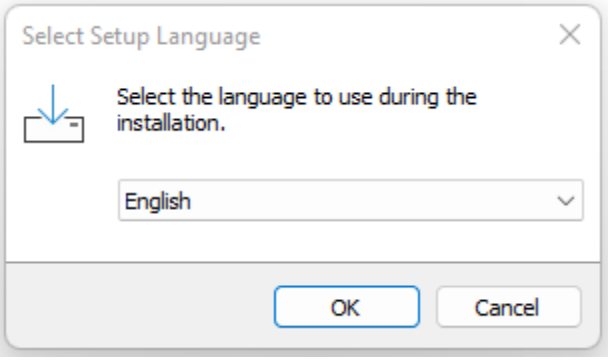
Step 2: Run the Installer

Open your downloads folder and double-click on the R installer file that you just downloaded.

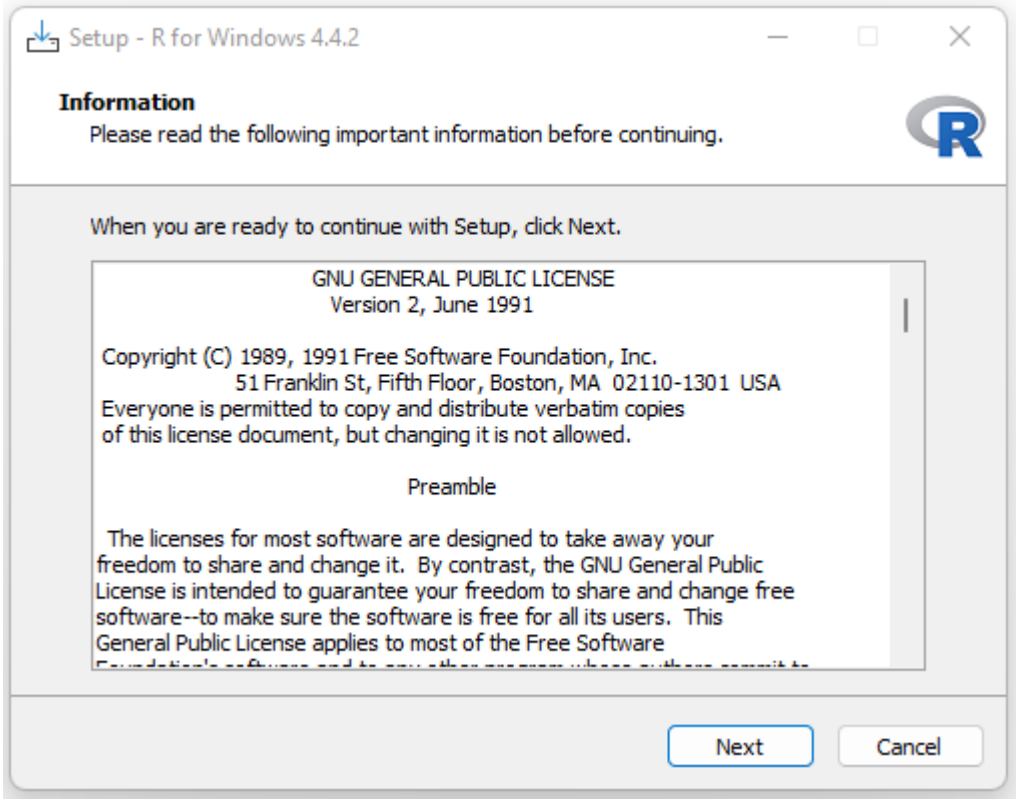
You may be required to grant access based on your security configurations. Please permit it and continue.

Step 3: Install R

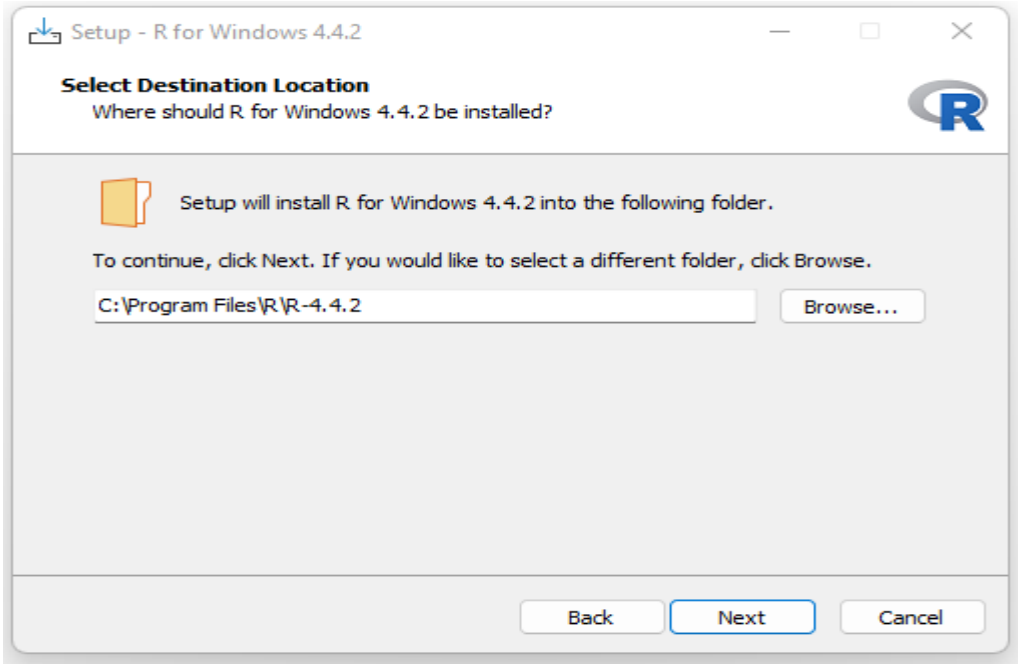
After executing the installer, you will encounter this screen.



Select your desired language. (Here English) and click on OK button.

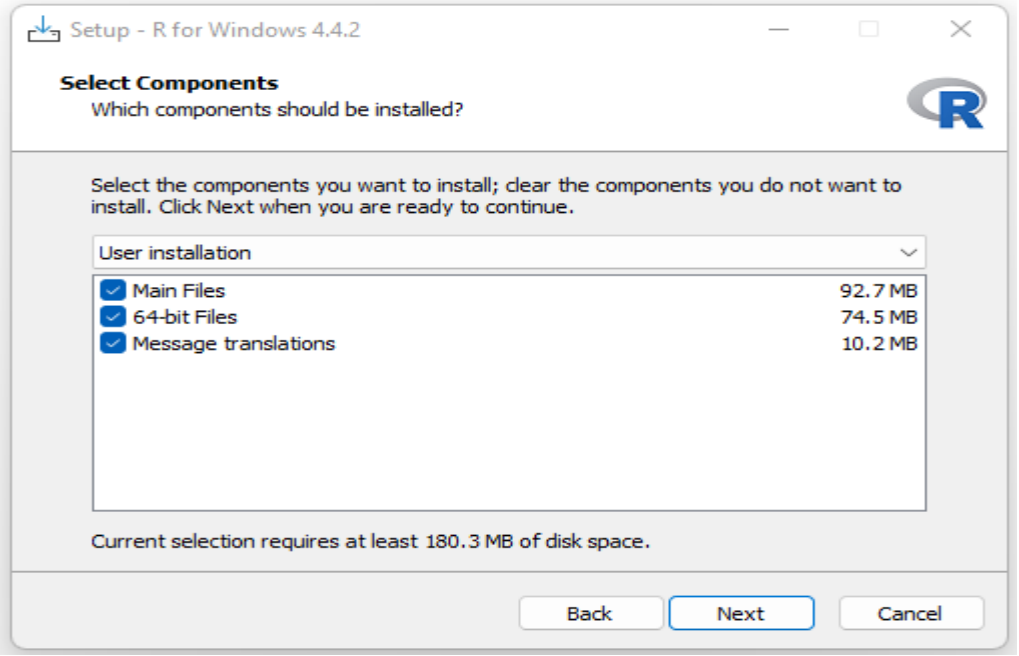


And then click on Next button to accept licenses agreements,

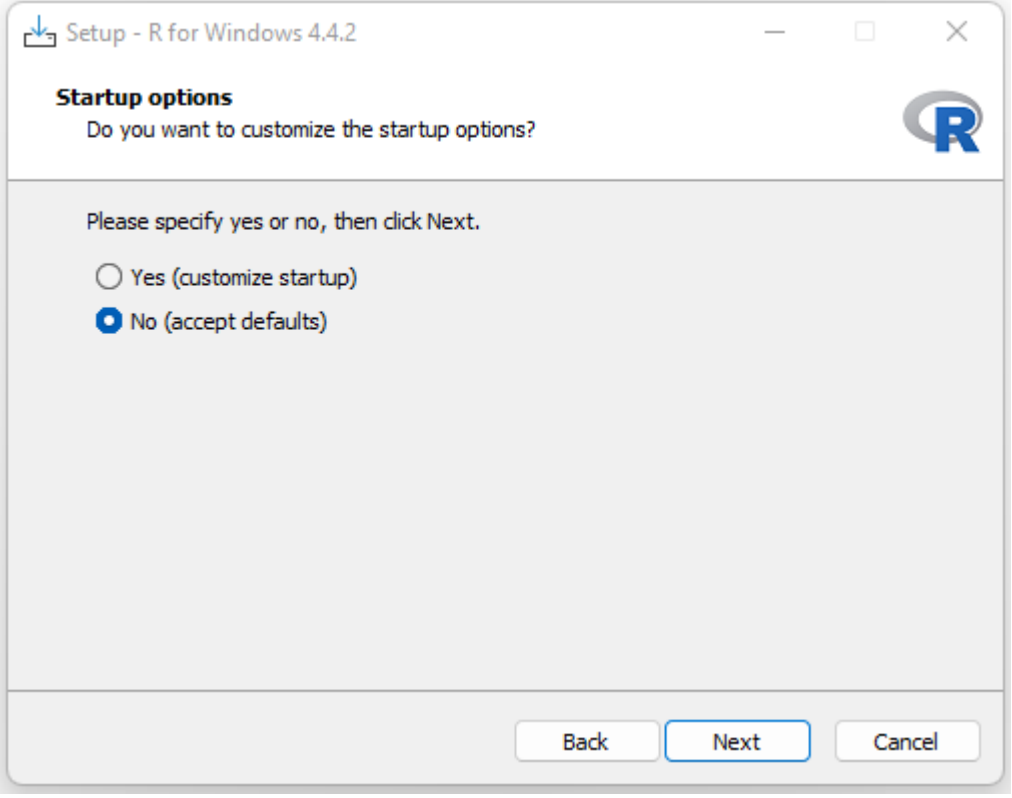


You will be prompted to specify the installation directory. The default location is `C:\Program Files\R\`.

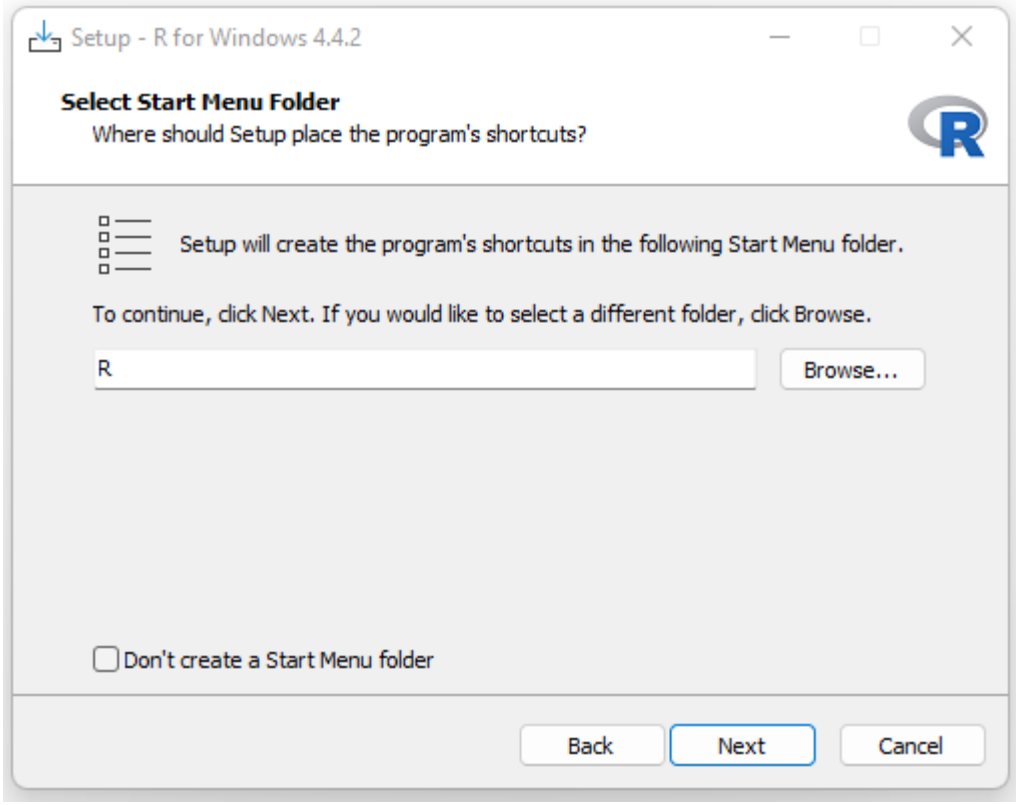
You may retain this setting and continue with the installation by clicking on Next button.



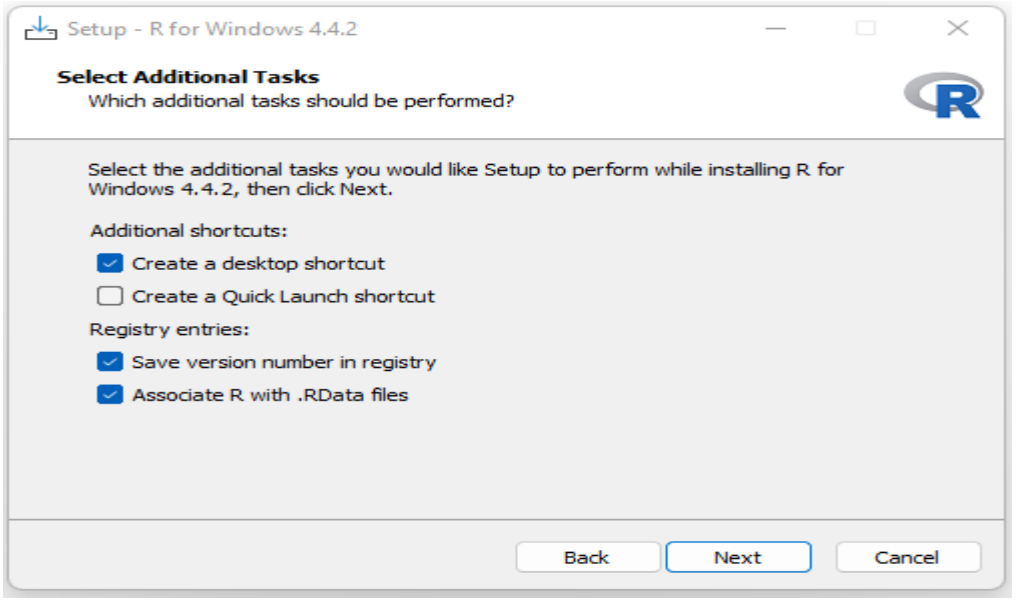
The installer will then prompt you to choose components and where to place shortcuts. Click Next button to continue installation.



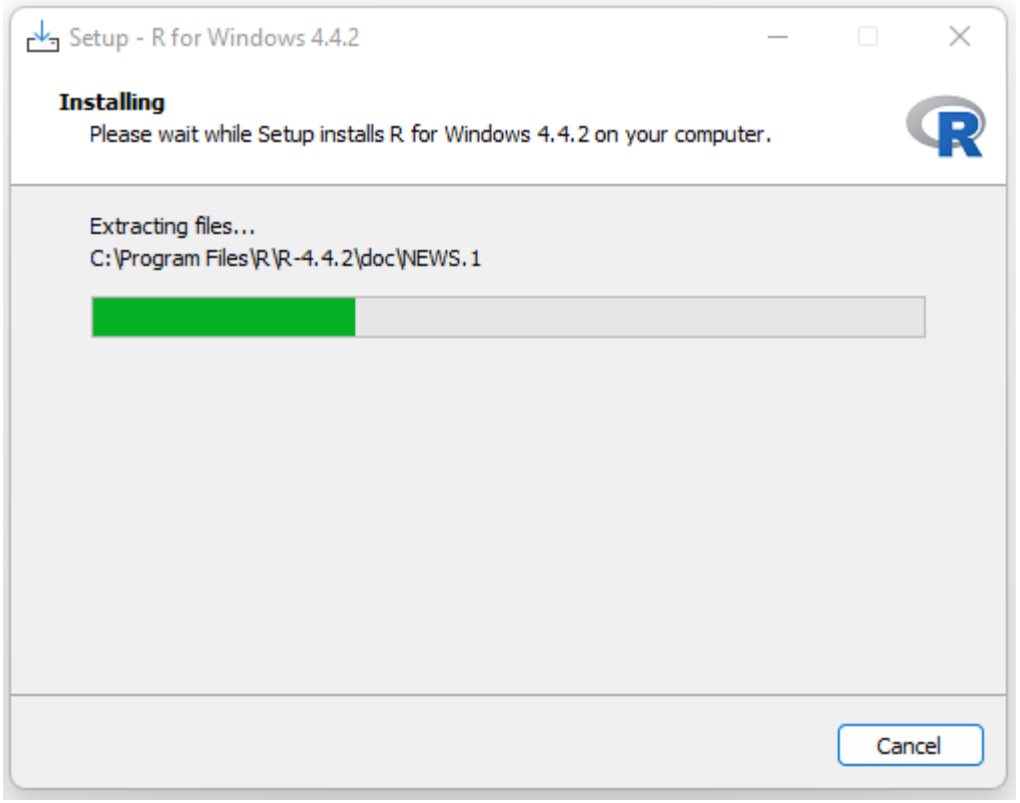
Select Start-up options, then click Next button.

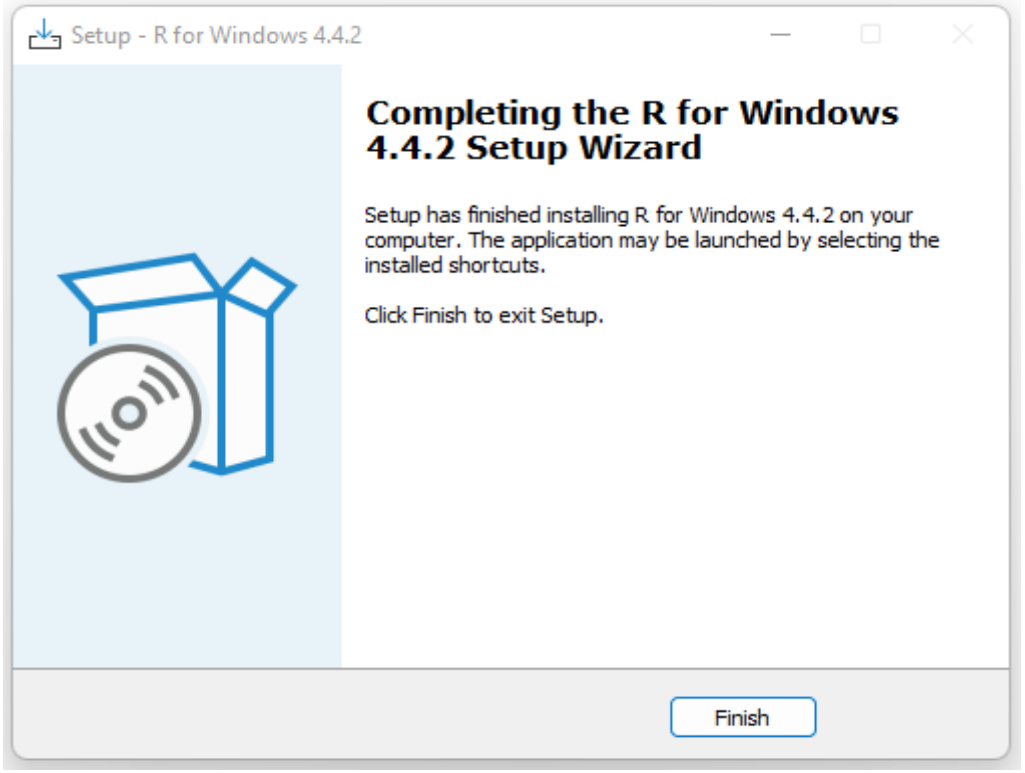


Select Start Menu Folder, then click Next button.



Select Additional Tasks, then click Next button.





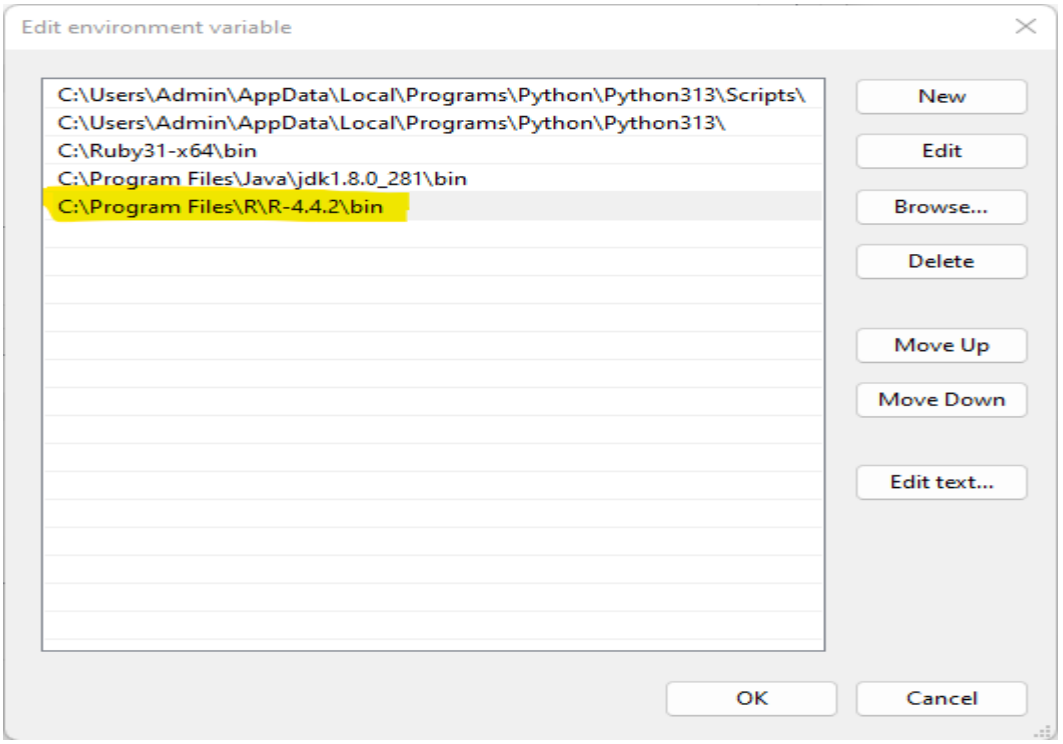
Setup has finished installing R for Windows 4.4.2. Click Finish to exit the installer.

Step 4: Add R to the PATH Environment Variable Manually

It is essential to manually set that the PATH environment variable of your system is configured properly.

To add the PATH environment variable, please follow these steps.

- Copy the path where R is installed to the bin folder.
- (Usually, it is in C:\Program Files\R\R-4.4.2\bin)
- Right-click on **This PC** and select **Properties**.
- Click on **Advanced System Settings** and then **Environment Variables**.
- Under User and System Variables sections, find and select Path, then click Edit.
- Click New and paste the path.



Click OK to close all dialogs and apply the changes.

Step 5: Verify the Installation

After the installation is finished, you can confirm that R has been installed correctly by opening a command prompt (cmd) and entering the following command:

R --version

```
C:\Users\Admin>R --version
R version 4.4.2 (2024-10-31 ucrt) -- "Pile of Leaves"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under the terms of the
GNU General Public License versions 2 or 3.
For more information about these matters see
https://www.gnu.org/licenses/.
```

3. Install RStudio Desktop

- Go to the Posit (formerly RStudio) website: https://posit.co/download/rstudio-desktop/.
- Scroll down and locate the "RStudio Desktop" section.
- Click on the "DOWNLOAD" button under the "RStudio Open Source License" (which is free).
- Download the installer (.exe file for Windows) recommended for your operating system.
- Run the downloaded .exe file to launch the RStudio Setup Wizard.
- Follow the prompts of the installer, accepting the default options as recommended.
- Once the installation is complete, you can click "Finish" to exit the wizard.

Launching and using RStudio

- You can launch RStudio by searching for it in the Windows search bar or by locating the shortcut that might have been created during the installation.
- When RStudio opens, you'll see a multi-pane interface with the R console and other features.
- You can start using RStudio for data analysis, statistical modeling, and visualizations.

4 . **A dataset contains the prices of houses in a city. Find the 25th and 75th percentiles and calculate the interquartile range (IQR). How does the IQR help in understanding the price variability?**

Solu - calculate the 25th percentile (Q1), 75th percentile (Q3), and the Interquartile Range (IQR) in R using the built in quantile() and IQR() functions.

code

```
# Create a sample dataset of house prices
house_prices <- c(250000, 300000, 280000, 450000, 320000, 380000, 500000, 420000, 290000, 350000)
# Calculate the 25th and 75th percentiles (quartiles)
quartiles <- quantile(house_prices, probs = c(0.25, 0.75))
# Extract Q1 and Q3
Q1 <- quartiles[1]
Q3 <- quartiles[2]
# Calculate the Interquartile Range (IQR)

iqr_value <- IQR(house_prices)
# Print the results
cat("25th percentile (Q1):", Q1, "\n")
cat("75th percentile (Q3):", Q3, "\n")
cat("Interquartile Range (IQR):", iqr_value, "\n")
```

output

```
25th percentile (Q1): 292500
75th percentile (Q3): 410000
Interquartile Range (IQR): 117500
```

Explanation

Percentiles:

- Divide a dataset into 100 equal parts.
- The nth percentile indicates the value below which n% of the data falls.
- For example, the 25th percentile (Q1) is the value below which 25% of the data lies, and the 75th percentile (Q3) is the value below which 75% of the data lies.

Interquartile Range (IQR):

- Measures the spread of the middle 50% of the data.
- It's calculated as the difference between the 75th percentile (Q3) and the 25th percentile (Q1):
  IQR = Q3 - Q1

- A smaller IQR indicates data clustered tightly around the median, meaning low variability, while a larger IQR suggests data that is more spread out, indicating higher variability.
- The IQR is useful because it is less affected by outliers than the range and provides a clearer picture of data distribution, especially in skewed datasets.
- It is used in outlier detection and box plots for data visualization.

5. You are given a dataset with categorical variables about customer satisfaction levels (Low, Medium, High) and whether customers made repeat purchases (Yes/No). Create visualizations such as bar plots or stacked bar charts to explore the relationship between satisfaction level and repeat purchases. What can you infer from the data?

(**note**-We can install packages into this library using the install.packages() function in the R console:
install.packages("ggplot2"))

code

```
# Load the ggplot2 library
library(ggplot2)

# Create a data frame with customer satisfaction and repeat purchase data
customer_data <- data.frame( satisfaction = factor(c("Low", "Medium", "High", "Low",
"Medium", "High", "Low", "Medium", "High", "High", "Medium", "Low", "Medium"),
levels = c("Low", "Medium", "High")),
  repeat_purchase = factor(c("No", "No", "Yes", "No", "Yes", "Yes", "No", "No", "Yes",
"Yes", "No", "No", "Yes"))
)
# Create the stacked bar chart
ggplot(customer_data, aes(x = satisfaction, fill = repeat_purchase)) +
  geom_bar(position = "stack") + labs(title = "Customer Satisfaction vs. Repeat Purchases",
       x = "Satisfaction Level", y = "Number of Customers", fill = "Repeat Purchase")  +
theme_minimal()
```

Output

Explanation

This stacked bar chart will display the distribution of repeat purchase behavior (Yes/No) across different levels of customer satisfaction (Low, Medium, High).

- Each bar represents a level of customer satisfaction (Low, Medium, or High).
- The total height of each bar indicates the total number of customers within that particular satisfaction level.
- The segments within each bar show the proportion of customers in that satisfaction level who either made a repeat purchase ("Yes") or did not ("No").

By visually comparing the segments across the satisfaction levels, you can observe the following:

- How customer satisfaction might influence repeat purchase behavior: For example, if the "Yes" (repeat purchase) segment is noticeably larger in the "High" satisfaction bar compared to the "Low" satisfaction bar, it suggests a positive relationship between satisfaction and repeat purchases.
- The overall distribution of repeat purchases: You can see which satisfaction level has the highest or lowest number of repeat purchases.
- Comparison of "Yes" and "No" across satisfaction levels: This helps understand whether satisfied customers are more likely to repurchase compared to dissatisfied customers, and also how unsatisfied customers behave.

6. A dataset contains information about car models, including the engine size (in Liters), fuel efficiency (miles per gallon), and car price. Use a pair plot or correlation matrix to

explore the relationships between these variables. Which variables seem to have the strongest relationships, and what might be the practical significance of these findings?

(note-We can install packages into this library using the install.packages() function in the R console: install.packages("GGally"))

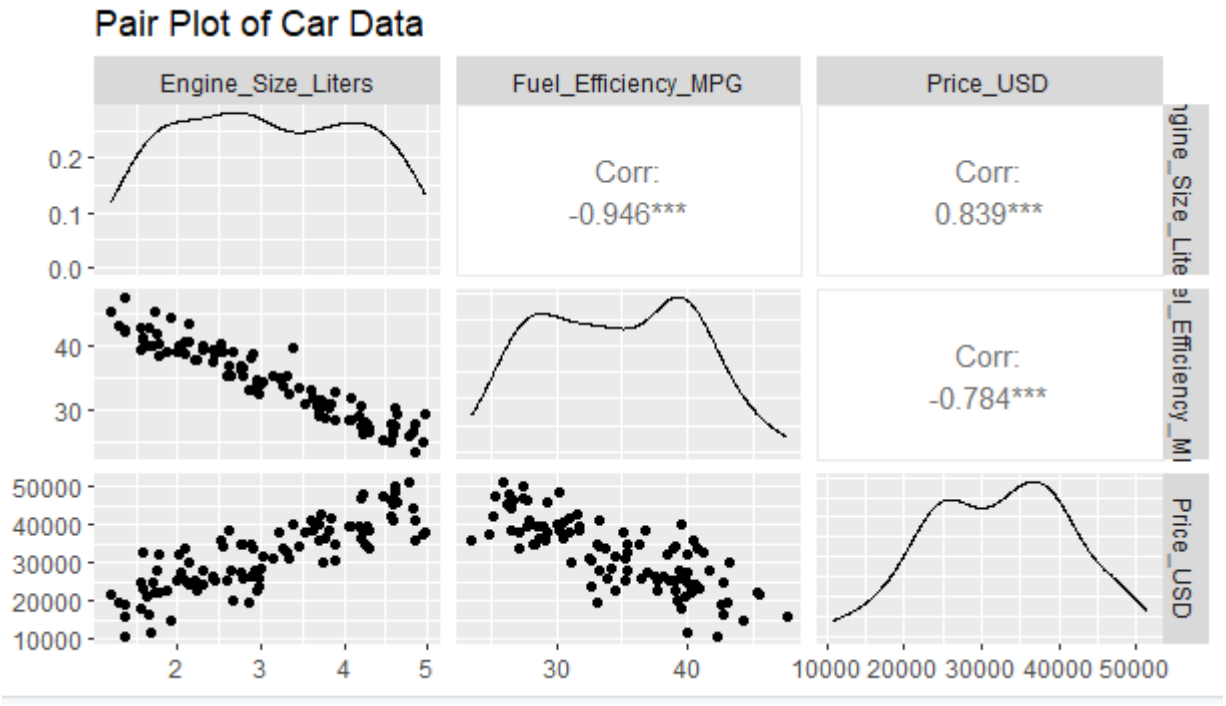
Code

```
library(GGally)
library(ggplot2)

# Simulate car dataset
set.seed(123)
car_data <- data.frame(
  Engine_Size_Liters = round(runif(100, 1.2, 5.0), 2),
  Fuel_Efficiency_MPG = round(runif(100, 15, 40), 1),
  Price_USD = round(runif(100, 15000, 80000), 0)
)

# Introduce some realistic relationships
car_data$Fuel_Efficiency_MPG <- 50 - 5 * car_data$Engine_Size_Liters +
rnorm(100, 0, 2)
car_data$Price_USD <- 10000 + 7000 * car_data$Engine_Size_Liters +
rnorm(100, 0, 5000)

# View first few rows
head(car_data)

# Generate Pair Plot
ggpairs(car_data, title = "Pair Plot of Car Data")

# Correlation Matrix
cor_matrix <- cor(car_data)
print("Correlation Matrix:")
print(round(cor_matrix, 2))
```

output

Correlation Matrix:

```
                  Engine_Size_Liters Fuel_Efficiency_MPG Price_USD
Engine_Size_Liters               1.00                -0.95      0.84
Fuel_Efficiency_MPG             -0.95                 1.00     -0.78
Price_USD                        0.84                -0.78      1.00
```



Explaination

The R code provided simulates a dataset of car characteristics and then analyzes the relationships between these variables using two primary methods: a pair plot and a correlation matrix.

A pair plot, also known as a scatterplot matrix, provides a visual representation of the relationships between each pair of variables in a dataset.

What a pair plot tells you

- Visual Relationships: You can visually inspect scatter plots to see if relationships are positive (points trend upwards), negative (points trend downwards), linear, non-linear, or if there's no clear pattern.
- Distributions: The diagonal plots help understand the distribution shape of each variable (e.g., normal, skewed).
- Outliers: Unusual data points that deviate from the general pattern can be identified, which might indicate data entry errors or unique cases.
- Clusters: Groupings of data points might suggest subpopulations within the data.
- Feature Selection: Visualizing relationships can help in identifying which variables might be strong predictors for a model.

A correlation matrix is a table that displays the correlation coefficients between all possible

pairs of variables in a dataset.

- Table Format: It is typically a square table where each variable is listed in both the rows and the columns.
- Correlation Coefficient: Each cell in the matrix contains a numerical value (correlation coefficient), usually Pearson's r, representing the strength and direction of the *linear* relationship between the two corresponding variables.
    - Value Range: The correlation coefficient ranges from -1 to +1.
        - +1: Indicates a perfect positive linear relationship (as one variable increases, the other increases proportionally).
        - -1: Indicates a perfect negative linear relationship (as one variable increases, the other decreases proportionally).
        - 0: Suggests no linear relationship between the variables.
    - Diagonal: The diagonal elements are always 1 because a variable is perfectly correlated with itself.
- Symmetry: The matrix is symmetrical, meaning the correlation between variable A and B is the same as between B and A.

What a correlation matrix tells you

- Strength and Direction: It quantifies the strength and direction of linear relationships between pairs of variables.
- Numerical Summary: Provides a concise numerical summary of relationships for the entire dataset.
- Multicollinearity Detection: High correlation coefficients between independent variables can signal multicollinearity, a potential issue in regression analysis.

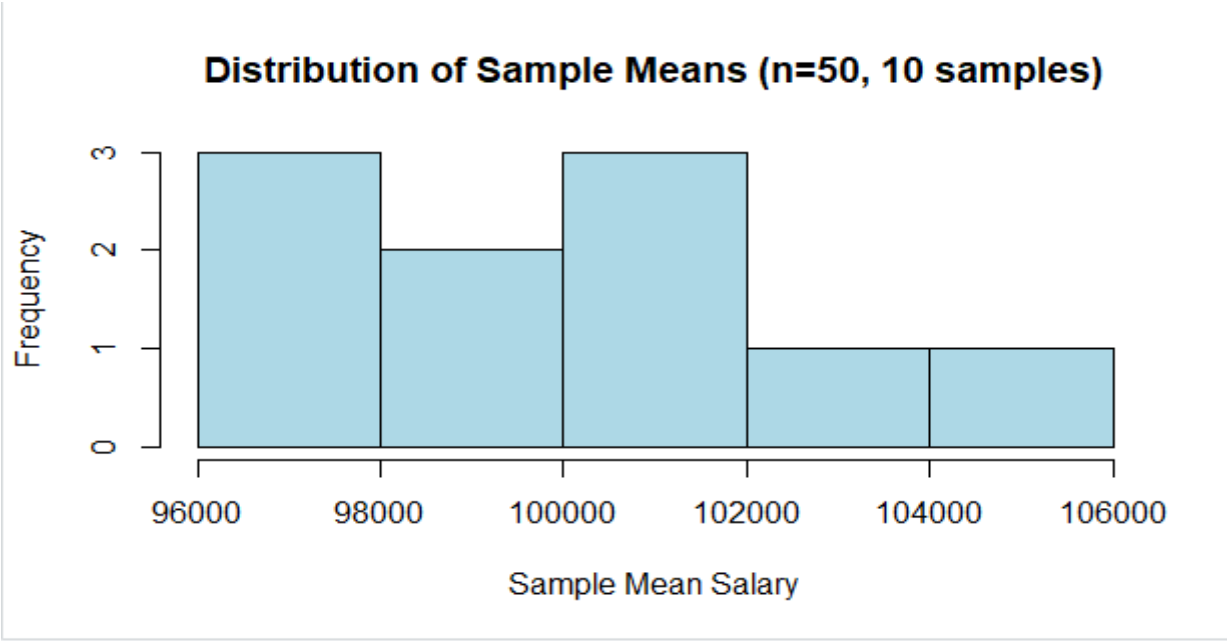
7. You want to estimate the mean salary of software engineers in a country. You take 10

different random samples, each containing 50 engineers, and calculate the sample mean for each. Plot the distribution of these sample means. How does the Central Limit Theorem explain the shape of this sampling distribution, even if the underlying salary distribution is skewed?

Code

```
# Setting parameters for the simulation
pop_mean <- 100000 # Population mean salary (hypothetical)
pop_sd <- 30000 # Population standard deviation (hypothetical)

# Create a skewed population distribution (e.g., log-normal)
# This simulates a situation where salaries are not normally distributed
set.seed(123) # for reproducibility
skewed_population <- rlnorm(100000, meanlog = log(pop_mean) - 0.5 *
log((pop_sd/pop_mean)^2 + 1), sdlog = sqrt(log((pop_sd/pop_mean)^2 + 1)))

num_samples <- 10 # Number of random samples to take
sample_size <- 50 # Size of each sample

# Create a vector to store the sample means
sample_means <- numeric(num_samples)

# Loop to take random samples and calculate sample means
for (i in 1:num_samples) {
  sample_data <- sample(skewed_population, sample_size, replace = TRUE) # Take a
random sample from the skewed population
  sample_means[i] <- mean(sample_data) # Calculate the mean of the sample
}

# Plotting the distribution of the sample means
hist(sample_means,
     main = "Distribution of Sample Means (n=50, 10 samples)",
     xlab = "Sample Mean Salary",
     ylab = "Frequency",
     col = "lightblue",
     border = "black")
```

Output



Explanation

The Central Limit Theorem states that as the sample size increases, the sampling distribution of the sample mean will approach a normal distribution, regardless of the original population distribution shape.
Here's how the CLT applies:

- Skewed Underlying Distribution: Even if the salary distribution is not normal, the distribution of the sample means will still tend to be approximately normal.
- Sample Size and Normality: A sample size of 50 is considered large enough for the CLT to apply, so the distribution of the 10 sample means will start to resemble a normal distribution. Taking more samples would make the distribution even closer to normal.
- Mean and Standard Deviation of the Sampling Distribution:
    - The mean of the sampling distribution of sample means will be approximately equal to the population mean salary.
    - The standard deviation of the sampling distribution (also called the standard error) will be smaller than the population standard deviation, calculated as the population standard deviation divided by the square root of the sample size ($\sigma/\sqrt{n}$). The smaller standard error means the sample means cluster more tightly around the population mean, leading to more precise estimates.

Even without knowing the exact shape of the salary distribution, the CLT allows the prediction that the distribution of sample means will be approximately normal. This is important because it allows the use of statistical tools that rely on normality to make inferences about the average salary.

8. A researcher conducts an experiment with a sample of 20 participants to determine if a

new drug affects heart rate. The sample has a mean heart rate increase of 8 beats per minute and a standard deviation of 2 beats per minute. Perform a hypothesis test using the t-distribution to determine if the mean heart rate increase is significantly different from zero at the 5% significance level.

Code

```
# Define the known parameters
sample_mean <- 8       # Sample mean heart rate increase
sample_sd <- 2         # Sample standard deviation
sample_size <- 20      # Number of participants in the sample
hypothesized_mean <- 0 # The null hypothesis is that the true mean is 0
alpha <- 0.05          # Significance level

# Calculate the t-statistic
t_statistic <- (sample_mean - hypothesized_mean) / (sample_sd / sqrt(sample_size))

# Calculate the degrees of freedom
degrees_of_freedom <- sample_size - 1

# Calculate the p-value (two-tailed test)
# The pt() function calculates the cumulative distribution function (CDF) of the t-
distribution.
# Multiply by 2 for a two-tailed test.
p_value <- 2 * pt(abs(t_statistic), df = degrees_of_freedom, lower.tail = FALSE)

# Print the results
cat("Sample Mean:", sample_mean, "\n")
cat("Sample Standard Deviation:", sample_sd, "\n")
cat("Sample Size:", sample_size, "\n")
cat("Hypothesized Mean (Null Hypothesis):", hypothesized_mean, "\n")
cat("Significance Level (alpha):", alpha, "\n")
cat("\n")
cat("Calculated t-statistic:", t_statistic, "\n")
cat("Degrees of Freedom:", degrees_of_freedom, "\n")
cat("P-value:", p_value, "\n")

# Make a decision based on the p-value
if (p_value < alpha) {  cat("Conclusion: Reject the null hypothesis. The mean heart rate
increase is significantly different from zero.\n")
}else {
  cat("Conclusion: Fail to reject the null hypothesis. There is no significant evidence that the
mean heart rate increase is different from zero.\n")
}

# Alternatively, the built-in t.test function can be used if you have raw data
```

```
# For demonstration, create some dummy data that would yield similar results:
# (Note: This is a simulation; the actual raw data would be the 20 heart rate increase measurements)
dummy_data <- rnorm(sample_size, mean = sample_mean, sd = sample_sd)
t_test_result <- t.test(dummy_data, mu = hypothesized_mean, alternative = "two.sided")

cat("\n--- Results using built-in t.test function (with dummy data for demonstration) ---\n")
print(t_test_result)
```

output

```
Sample Mean: 8
Sample Standard Deviation: 2
Sample Size: 20
Hypothesized Mean (Null Hypothesis): 0
Significance Level (alpha): 0.05

Calculated t-statistic: 17.88854
Degrees of Freedom: 19
P-value: 2.395621e-13
Conclusion: Reject the null hypothesis. The mean heart rate increase is
significantly different from zero.

--- Results using built-in t.test function (with dummy data for demonstration)
---
    One Sample t-test

data:  dummy_data
t = 16.07, df = 19, p-value = 1.629e-12
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 6.562517 8.527998
sample estimates:
mean of x
 7.545258
```

Explanation

1. State the Hypotheses

- **Null Hypothesis (H0):** The mean heart rate increase (μ) is equal to zero. This means the drug has no effect on heart rate.
  - H0: $\mu = 0$
- **Alternative Hypothesis (Ha):** The mean heart rate increase (μ) is not equal to zero. This means the drug *does* have an effect on heart rate (it could be an increase or a decrease). This will be a two-tailed test.
  - Ha: $\mu \neq 0$

2. Choose the significance level (α)

The problem specifies a 5% significance level, so:

- $\alpha = 0.05$

3. Calculate the test statistic

The formula for the t-statistic for a one-sample t-test is:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

Where:

- $\bar{x}$ = sample mean = 8 beats per minute
- μ_0 = hypothesized population mean (from the null hypothesis) = 0 beats per minute
- s = sample standard deviation = 2 beats per minute
- n = sample size = 20 participants

Plugging in the values:

$$t = \frac{8 - 0}{2/\sqrt{20}} = \frac{8}{2/4.472} \approx \frac{8}{0.4472} \approx 17.889$$

4. Determine the degrees of freedom (df)

The degrees of freedom for a one-sample t-test is n - 1.

- df = 20 - 1 = 19

5. Find the Critical Values

Since this is a two-tailed test with α = 0.05 and df = 19, you need to find the t-values that leave 0.025 in each tail of the t-distribution. Looking up a t-distribution table (or using statistical software), the critical values are approximately **±2.093**. According to Newcastle University, the t-distribution table gives $t_{\alpha,v} = t_{0.05,24} = 1.7109$.

6. Make a Decision

- Compare the calculated t-statistic (17.889) to the critical values (**±2.093**).
- Since **17.889 > 2.093**, the calculated t-statistic falls into the rejection region.

7. Conclusion

Because the calculated t-statistic (17.889) exceeds the critical value (2.093), the null hypothesis is rejected.

9. A company is testing two versions of a webpage (A and B) to determine which version leads to more sales. Version A was shown to 1,000 users and resulted in 120 sales. Version B was shown to 1,200 users and resulted in 150 sales. Perform an A/B test to determine if there is a statistically significant difference in the conversion rates between the two versions. Use a 5% significance level

Code

```
# Data given in the problem
users_A <- 1000
sales_A <- 120

users_B <- 1200
sales_B <- 150

# Create a contingency table
# This table will summarize the observed frequencies (sales and non-sales) for each
    version.
contingency_table <- matrix(c(sales_A, users_A - sales_A, sales_B, users_B - sales_B),
                            nrow = 2, byrow = TRUE, dimnames = list(Version = c("A", "B"),
                                                                    Outcome = c("Sales", "No Sales")))

cat("Observed Contingency Table:\n")
print(contingency_table)

# Perform the Chi-square test of independence
# The chisq.test() function from base R performs the Chi-square test.
chi_square_test_result <- chisq.test(contingency_table, correct = TRUE)

# Print the results of the Chi-square test
cat("\nChi-Square Test Results:\n")
print(chi_square_test_result)

# Interpret the results
alpha <- 0.05
cat("\nSignificance Level (alpha):", alpha, "\n")

if (chi_square_test_result$p.value < alpha) {
  cat("Conclusion: Reject the null hypothesis.\n")
  cat("There is a statistically significant difference in conversion rates between Version A
     and Version B.\n")
} else {
  cat("Conclusion: Fail to reject the null hypothesis.\n")
```

```
cat("There is no statistically significant difference in conversion rates between Version A
    and Version B at the", alpha * 100, "% significance level.\n")
}

# Optional: Calculate conversion rates and difference
cr_A <- sales_A / users_A
cr_B <- sales_B / users_B
diff_cr <- cr_B - cr_A

cat("\nConversion Rate for Version A:", round(cr_A * 100, 2), "%\n")
cat("Conversion Rate for Version B:", round(cr_B * 100, 2), "%\n")
cat("Difference in Conversion Rates (B - A):", round(diff_cr * 100, 2), "%\n")
```

output

Observed Contingency Table:

```
          Outcome
Version    Sales No Sales
      A      120    880
      B      150   1050
```

Chi-Square Test Results:

```
        Pearson's Chi-squared test with Yates' continuity correction

data:  contingency_table
X-squared = 0.057392, df = 1, p-value = 0.8105
```

```
Significance Level (alpha): 0.05
Conclusion: Fail to reject the null hypothesis.
There is no statistically significant difference in conversion rates between Versi
```

```
Conversion Rate for Version A: 12.0 %
Conversion Rate for Version B: 12.5 %
Difference in Conversion Rates (B - A): 0.5 %
```

Explanation

An **A/B test** to determine whether there's a **statistically significant difference in conversion rates** (sales) between two versions (A and B) of something—likely a webpage, ad, or product.

1.

- **Version A**: 1000 users, 120 sales → **Conversion Rate = 12%**
- **Version B**: 1200 users, 150 sales → **Conversion Rate = 12.5%**

You're testing if this **0.5% difference is significant**.

2 . Contingency Table

This creates a **2×2 matrix** showing how many people made a purchase and how many didn't:

Version	**Sales**	**No Sales**
A	120	880
B	150	1050

You're now ready to test whether the difference between these two rows is **statistically significant**.

3 . Chi-Square Test of Independence

This test checks if **"conversion is independent of version"**—in other words, does being in Group A or B affect the chance of conversion?

- **Null Hypothesis (H_0)**: Conversion is independent of version (A or B).
- **Alternative Hypothesis (H_1)**: Conversion is dependent on the version (A or B).

correct = TRUE applies **Yates' continuity correction**, which slightly adjusts the chi-square result for small sample sizes in 2×2 tables.

4 . Interpret the p-value

- You compare the **p-value** with your **significance level** $\alpha = 0.05$.
- If **$p < 0.05$**, there's a significant difference.
- If **$p \geq 0.05$**, there's **no** statistically significant difference.

5 . Conversion Rates and Difference

You calculate the actual **conversion rate** for each version and their **difference**.
Even though Version B looks slightly better (0.5% higher), the **Chi-square test** says this difference **could easily be due to chance**.

10. You are comparing the average daily sales between two stores. Store A has a mean daily sales value of $1,000 with a standard deviation of $100 over 30 days, and Store B has a mean daily sales value of $950 with a standard deviation of $120 over 30 days. Conduct a two-sample t-test to determine if there is a significant difference between the average sales of the two stores at the 5% significance level.

Problem Summary:

You're comparing the **average daily sales** between **Store A** and **Store B** over 30 days using a **two-sample t-test**.

Given:

Metric	Store A	Store B
Mean ($\bar{x}$)	$1,000	$950
Std. Deviation (s)	$100	$120
Sample Size (n)	30	30

Code

```
# Given data
mean_A <- 1000
sd_A <- 100
n_A <- 30

mean_B <- 950
sd_B <- 120
n_B <- 30

# Calculate the t-statistic
se <- sqrt((sd_A^2 / n_A) + (sd_B^2 / n_B))
t_stat <- (mean_A - mean_B) / se

# Calculate degrees of freedom (Welch-Satterthwaite)
df <- ((sd_A^2 / n_A + sd_B^2 / n_B)^2) /
     (((sd_A^2 / n_A)^2) / (n_A - 1) + ((sd_B^2 / n_B)^2) / (n_B - 1))

# Two-tailed p-value
p_value <- 2 * pt(-abs(t_stat), df)

# Significance level
alpha <- 0.05
```

```
# Print results
cat("Two-Sample t-Test (Welch's t-test)\n")
cat("==================================\n")
cat("t-statistic:", round(t_stat, 4), "\n")
cat("Degrees of freedom:", round(df, 2), "\n")
cat("p-value:", round(p_value, 4), "\n")
cat("Significance Level (alpha):", alpha, "\n\n")

if (p_value < alpha) {
  cat("Conclusion: Reject the null hypothesis.\n")
  cat("There is a statistically significant difference in mean sales between Store A and Store
B.\n")
} else {
  cat("Conclusion: Fail to reject the null hypothesis.\n")
  cat("There is no statistically significant difference in mean sales between Store A and Store
B.\n")
}
```

Output

```
Two-Sample t-Test (Welch's t-test)
==================================
t-statistic: 1.753
Degrees of freedom: 56.17
p-value: 0.0847
Significance Level (alpha): 0.05

Conclusion: Fail to reject the null hypothesis.
There is no statistically significant difference in mean sales between Store A and Store B.
```

(Note: **t ≈ 1.753**, **df ≈ 56.17**, **p ≈ 0.0847**
Since **0.0847 > 0.05**, we **fail to reject the null hypothesis)**

Explanation

The **two-sample t-test** is used to compare the **means of two independent groups** to determine if the difference between them is **statistically significant**.
Here to know if Store A's average daily sales ($1,000) are significantly different from Store B's ($950).

Steps of the t-Test

1. Set Hypotheses

- **Null Hypothesis (H_0):** There is **no difference** in average sales. ($\mu_1 = \mu_2$)
- **Alternative Hypothesis (H_1):** There **is a difference.** ($\mu_1 \neq \mu_2$)

2. Calculate the t-statistic

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

This tells you **how many standard errors apart** the two sample means are.

3. Calculate degrees of freedom (df)

Use Welch's approximation, because the standard deviations are not equal:

$$df = \text{complex formula based on variances and sample sizes}$$

4. Get p-value from t-distribution

This tells you the **probability** of observing such a difference (or more extreme) **if the null hypothesis is true.**

5. Compare p-value to alpha (0.05)

- If **p-value < 0.05**, difference is statistically significant → Reject H_0.
- If **p-value ≥ 0.05**, difference is **not** statistically significant → Fail to reject H_0.

11. A company collects data on employees' salaries and records their education level as a categorical variable with three levels: "High School", "Bachelor's", and "Master's". Fit a multiple linear regression model to predict salary using education level (as a factor variable) and years of experience. Interpret the coefficients for the education levels in the regression model

```
# Simulate Sample Data
set.seed(123)  # for reproducibility

# Create categorical variable: Education
education <- factor(  sample(c("High School", "Bachelor", "Master"), 100, replace = TRUE),
  levels = c("High School", "Bachelor", "Master")  # set reference level
)

# Create numerical variable: Experience
experience <- round(runif(100, 1, 20), 1)

# Simulate Salary based on education and experience
salary <- 30000 +
 ifelse(education == "Bachelor", 8000,
        ifelse(education == "Master", 15000, 0)) +
  2000 * experience +
  rnorm(100, mean = 0, sd = 5000)  # add noise

# Combine into a data frame
data <- data.frame(Salary = salary, Education = education, Experience = experience)

#  Fit Multiple Linear Regression Model
model <- lm(Salary ~ Education + Experience, data = data)

#  View Summary of the Model
summary(model)

#  Interpret Coefficients
cat("\nInterpretation of Coefficients:\n")
cat("Intercept:", round(coef(model)[1]), "\n")
cat("Effect of Bachelor's vs High School:", round(coef(model)[2]), "\n")
cat("Effect of Master's vs High School:", round(coef(model)[3]), "\n")
cat("Salary increase per year of experience:", round(coef(model)[4]), "\n")
```

Output

Interpretation of Coefficients:
Intercept: 30991
Effect of Bachelor's vs High School: 6700
Effect of Master's vs High School: 12578
Salary increase per year of experience: 2010

Explanation

This shows how to **fit and interpret a multiple linear regression model** when one of the predictors is a **categorical variable** (education level), and the other is a **numerical variable** (years of experience).

Building a **multiple linear regression model** to **predict salary** using:

1. **Education level** (categorical: High School, Bachelor's, Master's)
2. **Years of experience** (numeric)

In R:

```
r

lm(Salary ~ Education + Experience, data = data)
```

This fits a **multiple linear regression model**:

$$\text{Salary} = \beta_0 + \beta_1(\text{Bachelor}) + \beta_2(\text{Master}) + \beta_3(\text{Experience}) + \varepsilon$$

A **multiple linear regression model** is a statistical technique used to model the relationship between **one continuous dependent variable** and **two or more independent variables** (which may be continuous or categorical).

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \varepsilon$$

- Y: Dependent variable (response)
- $X_1, X_2, ..., X_k$: Independent variables (predictors)
- β_0: Intercept
- $\beta_1, ..., \beta_k$: Coefficients (effect of each predictor)
- ε: Error term

- Let's say the estimated model is:

$$\text{Salary} = 30321 + 8343 \cdot \text{Bachelor} + 14952 \cdot \text{Master} + 1968 \cdot \text{Experience}$$

Then:

- A **High School** graduate with **5 years experience**:

$$Salary = 30321 + 0 + 0 + 1968 \times 5 = 40161$$

- A **Bachelor's** graduate with 5 years:

$$Salary = 30321 + 8343 + 1968 \times 5 = 48504$$

- A **Master's** graduate with 5 years:

$$Salary = 30321 + 14952 + 1968 \times 5 = 55113$$

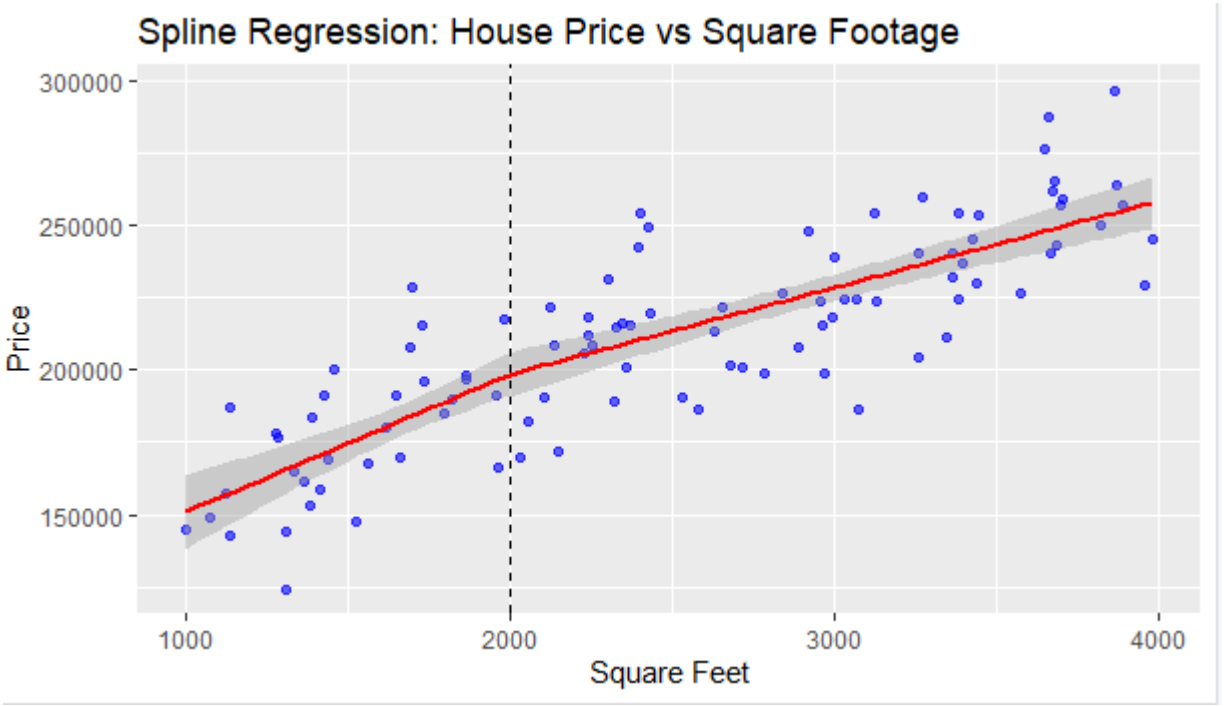
12. You have data on housing prices and square footage and notice that the relationship between square footage and price is nonlinear. Fit a spline regression model to allow the relationship between square footage and price to change at 2,000 square feet. Explain how spline regression can capture different behaviours of the relationship before and after 2,000 square feet.

code

```
# Load required libraries
library(ggplot2)

#  Generate sample data
set.seed(123)
square_feet <- round(runif(100, 1000, 4000), 0)

# Simulate price: linear until 2000 sqft, slower growth after
price <- 100000 +50 * square_feet +  ifelse(square_feet > 2000, -20 * (square_feet - 2000),
0) +rnorm(100, 0, 20000)  # Add random noise

# Create dataframe
housing <- data.frame(Price = price, SquareFeet = square_feet)

#  Create spline term: captures difference after 2000 sqft
housing$After2000 <- pmax(0, housing$SquareFeet - 2000)

# Fit spline regression model
model_spline <- lm(Price ~ SquareFeet + After2000, data = housing)

# Summary of the model
summary(model_spline)

#  Visualize the data and spline regression line
ggplot(housing, aes(x = SquareFeet, y = Price)) +
  geom_point(color = "blue", alpha = 0.6) +  geom_smooth(method = "lm", formula = y ~ x +
pmax(0, x - 2000), color = "red") +  geom_vline(xintercept = 2000, linetype = "dashed",
color = "black") +  labs(title = "Spline Regression: House Price vs Square Footage",
       x = "Square Feet", y = "Price")
```

output

```
Call:
lm(formula = Price ~ SquareFeet + After2000, data = housing)

Residuals:
   Min     1Q Median     3Q    Max
-44726 -12325   -214  11860  44485

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 103336.176  14733.544   7.014 3.13e-10 ***
SquareFeet      47.602      8.601   5.534 2.66e-07 ***
After2000      -17.599     11.062  -1.591    0.115
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 19480 on 97 degrees of freedom
Multiple R-squared:  0.7018,    Adjusted R-squared:  0.6957
F-statistic: 114.2 on 2 and 97 DF,  p-value: < 2.2e-16
```



Explanation

Spline regression is a method used in regression analysis when the relationship between the independent and dependent variables is **nonlinear** but can be broken into **piecewise linear or smooth segments**.

In simple terms, instead of fitting a **single straight line** through the entire data, spline regression fits **different lines** (or curves) for different parts of the data, joined at specific points called **knots**. (A **knot** is a specific value of the **independent variable (x)** — like square footage — **where the slope of the regression line is allowed to change**.)

Why Use Spline Regression?
Because real-world relationships between variables are **rarely perfectly linear**. For example:

- **House price** might increase fast up to a certain square footage, and then level off.
- **Growth curves**, **income vs. age**, **biological measurements**, etc., often show **non-constant slopes**.

How it Works (Mathematically):

Suppose `x` is the square footage, and `k = 2000` is the knot.

```ini
After2000 = max(0, x - 2000)
```

Then the model:

```ini
Price = β0 + β1 * x + β2 * After2000 + error
```

- For x ≤ 2000: `After2000 = 0`, so the model is `Price = β0 + β1 * x`
- For x > 2000: `After2000 = x - 2000`, so the model becomes
  `Price = β0 + β1 * x + β2 * (x - 2000) = (β1 + β2) * x + constant`

So the **slope changes after the knot**, allowing the curve to bend.

For the given experiment

```r
# Fit the spline regression model
model <- lm(price ~ square_footage + sqft_after_2000, data = housing_data)
summary(model)
```

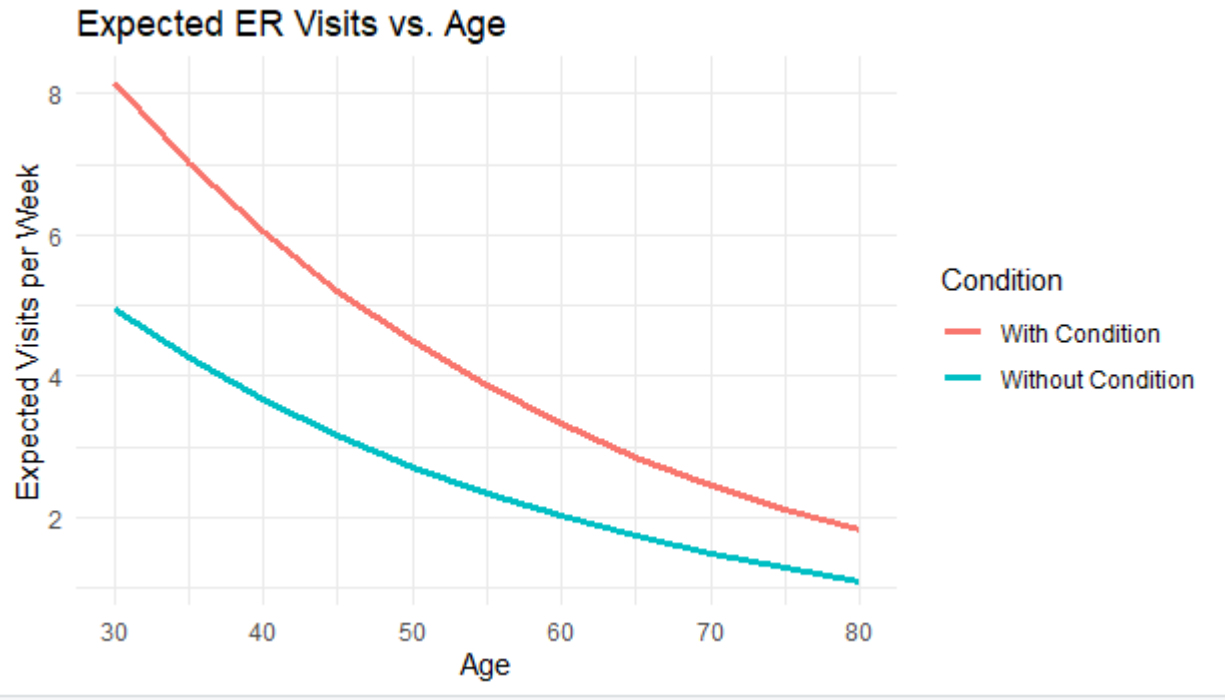
- **What it does:**
- Fits a **linear model with a spline term**:
  - One slope up to 2000 sqft (`square_footage`)
  - One additional slope **after 2000** (`sqft_after_2000`)
- `summary(model)` prints coefficients and p-values.

13. A hospital is using a Poisson regression model (a type of GLM) to predict the number of emergency room visits per week based on patient age and medical history. The model is given by: Log(λ) =2.5-0.03*Age+0.5*condition where λ is the expected number of visits per week, Age is the patient's age, and condition is a binary variable (1 if the patient has a chronic condition, 0 otherwise). Interpret the coefficients of Age and condition. What is the expected number of visits per week for a 60-year-old patient with a chronic condition? How would the expected number of visits change if the patient did not have a chronic condition?

Code

```
# Define the coefficients from the model
intercept <- 2.5
age_coeff <- -0.03
condition_coeff <- 0.5

# Function to compute expected number of visits
expected_visits <- function(age, condition) {
  log_lambda <- intercept + age_coeff * age + condition_coeff * condition
  lambda <- exp(log_lambda)
  return(lambda)
}

#  60-year-old with chronic condition
visits_with_condition <- expected_visits(age = 60, condition = 1)
cat("Expected visits (with chronic condition):", visits_with_condition, "\n")

#  60-year-old without chronic condition
visits_without_condition <- expected_visits(age = 60, condition = 0)
cat("Expected visits (without chronic condition):", visits_without_condition, "\n")

# Optional: Try for other ages
ages <- seq(30, 80, by = 5)
visits_plot <- data.frame(
  Age = ages,
  With_Condition = sapply(ages, function(a) expected_visits(a, 1)),
  Without_Condition = sapply(ages, function(a) expected_visits(a, 0))
)
# Plot the results
library(ggplot2)
ggplot(visits_plot, aes(x = Age)) +
  geom_line(aes(y = With_Condition, color = "With Condition"), size = 1.2) +
  geom_line(aes(y = Without_Condition, color = "Without Condition"), size = 1.2) +
  labs(title = "Expected ER Visits vs. Age", y = "Expected Visits per Week",
color = "Condition") +
  theme_minimal()
```

Output



Explanation

Poisson regression is a type of **Generalized Linear Model (GLM)** used for modeling **count data**, i.e., data representing the number of times an event occurs in a fixed interval (time, space, etc.).

The model assumes that the **response variable** Y (e.g., number of ER visits) follows a **Poisson distribution**:

$$Y \sim \text{Poisson}(\lambda)$$

where:

- λ = expected number of events (must be > 0)

The model uses a **log link function** to relate predictors to the expected count:

$$\log(\lambda) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

This ensures that $\lambda = e^{(\beta_0 + \beta_1 X_1 + \ldots)}$ is always positive.

For the given experiment

Model Equation (Log-link function):

$$\log(\lambda) = 2.5 - 0.03 \cdot \text{Age} + 0.5 \cdot \text{Condition}$$

Where:

- λ: expected number of ER visits per week
- **Age**: patient's age
- **Condition**: 1 if the patient has a chronic condition, 0 otherwise

📖 Interpretation of Coefficients

1. **Age coefficient: -0.03**
   - For every **1-year increase in age**, the **log of expected visits decreases by 0.03**.
   - In terms of λ:
     Each year older → multiply expected visits by $e^{-0.03} \approx 0.9704$
     ⇒ a **3% decrease in expected visits per year of age**.
2. **Condition coefficient: +0.5**
   - Having a **chronic condition increases the log expected visits by 0.5**.
   - In terms of λ:
     Chronic condition present ⇒ multiply visits by $e^{0.5} \approx 1.65$
     ⇒ **65% higher expected visits** than someone without the condition.

🧮 Expected Visits for 60-Year-Old with Chronic Condition

Plug in values:

- Age = 60
- Condition = 1

$$\log(\lambda) = 2.5 - 0.03 \cdot 60 + 0.5 \cdot 1 = 2.5 - 1.8 + 0.5 = 1.2$$

Now exponentiate:

$$\lambda = e^{1.2} \approx 3.32$$

✅ **Expected visits = ~3.32 per week**

🔁 If Same Patient Has No Chronic Condition

Same age (60), but condition = 0:

$$\log(\lambda) = 2.5 - 1.8 + 0 = 0.7$$

$$\lambda = e^{0.7} \approx 2.01$$

✅ **Expected visits = ~2.01 per week**

14. A bakery claims that its new cookie recipe is lower in calories compared to the old recipe, which had a mean calorie count of 200. You sample 40 new cookies and find a mean of 190 calories with a standard deviation of 15 calories. Perform a one-tailed t-test to determine if the new recipe has significantly fewer calories at a 5% significance level.

Code

```
# Inputs
sample_mean <- 190
sample_sd <- 15
n <- 40
mu <- 200  # mean of old recipe

# Manually compute the t-statistic
t_statistic <- (sample_mean - mu) / (sample_sd / sqrt(n))

# Degrees of freedom
df <- n - 1

# Compute p-value (left-tailed test)
p_value <- pt(t_statistic, df)

# Critical value for 5% significance level (one-tailed)
t_critical <- qt(0.05, df)

# Display results
cat("t-statistic:", round(t_statistic, 3), "\n")
cat("Degrees of freedom:", df, "\n")
cat("Critical t-value:", round(t_critical, 3), "\n")
cat("p-value:", round(p_value, 4), "\n")

# Conclusion
if (p_value < 0.05) {
  cat("Conclusion: Reject the null hypothesis. The new recipe has significantly fewer calories.\n")
} else {
  cat("Conclusion: Fail to reject the null hypothesis. No significant reduction in calories.\n")
}
```

Output

t-statistic: -4.216
Degrees of freedom: 39
Critical t-value: -1.685
p-value: 1e-04
Conclusion: Reject the null hypothesis. The new recipe has significantly fewer calories.

Explanation

Here perform o**ne-tailed t-test** to determine whether the new cookie recipe has significantly fewer calories than the old one.
A one-tailed t-test is a statistical test used when you want to check if the mean of a sample is significantly less than or greater than a known value (but only in one direction).

A bakery claims its **new cookie recipe** has **fewer calories** than the old one (which had **200 calories** on average).
You sampled **40 cookies**, and found:

- Sample mean = 190
- Standard deviation = 15
- Significance level = 0.05 (5%)
- We want to **prove that the new recipe has *fewer* calories**, not just "different"

Hypotheses:
This is a **left-tailed** (lower-tailed) test:

- **Null hypothesis (H_0):**
  Mean calories ≥ 200
  (No improvement or even worse)
- **Alternative hypothesis (H_1):**
  Mean calories < 200
  (New recipe has fewer calories)

Test Statistic:

Use **t-statistic**:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

Where:

- $\bar{x}$ = sample mean = 190
- μ_0 = hypothesized mean = 200
- s = sample standard deviation = 15
- n = sample size = 40

🔍 In Your Example:

Given:

- $\bar{x} = 190$
- $\mu_0 = 200$
- $s = 15$
- $n = 40$

Let's plug into the formula:

Step 1: Compute standard error:

$$SE = \frac{15}{\sqrt{40}} \approx \frac{15}{6.3246} \approx 2.37$$

Step 2: Compute t-statistic:

$$t = \frac{190 - 200}{2.37} \downarrow \frac{-10}{2.37} \approx -4.22$$

Decision Rule:

- If **p-value < 0.05**, reject the null hypothesis → there's **significant evidence** that the new recipe has **fewer** calories.
-

Interpretation:

If the result is significant:

"At the 5% significance level, we conclude that the new cookie recipe has significantly fewer calories than the old one."

If not significant:

"There is **not enough evidence** to conclude that the new recipe has fewer calories."