Q1. R is an open-source statistical programming language and environment. On the other hand, R Studio is an integrated development environment designed for use with R software. R Studio allows users to work with R within a clean, accessible graphical user interface (GUI), streamlining user experience with R software.

Q2. One major advantage of using R software for data analysis is the ease with which analyses can be reproduced relative to point and click programs. Against the backdrop of the replication crisis, open practices with respect to data processing and analysis are very important. Conducting data analyses in R allows for creation of projects that allow others to exactly reproduce the processing and analysis of data, step by step. Accordingly, peers can easily check these analyses for errors and ensure that reported results are accurate. On the other hand, analysis using point and click programs does not necessarily preserve each step that was taken to arrive at a given outcome. Others that review the data might thereby have to process the data and set up analyses themselves, creating the possibility of errors due to discrepancies in processing and analysis. Furthermore, conducting data analysis in R allows you, the scientist, to easily conduct and re-conduct your own analyses without the need to import data into a point and click program and set up the analyses each time.

Another major advantage of using R software over point and click programs is the versatility of the former. Point and click programs allow you to conduct only a finite number of different types of statistical analyses. On the other hand, R allows you to install thousands of different packages to employ different statistical approaches. Further, R allows you to write your functions and create your own packages if no package currently exists to suit the needs of your statistical approach. Although programs like jamovi allow users to install different modules (which mirror R packages and use R software under the hood), available modules number only a few dozen. With R software, on the other hand, access to package repositories and the ability to create your own functions allow for an essentially infinite number of ways to process data.

Q3. Within an R script, a comment is a chunk of text that is not executed as code. Comments are denoted by placing a “#” symbol at the beginning of each line of the comment. Sections allow code (and comments) to be broken up into discrete chunks, which improves ease of navigation through a script. Sections can be delineated by navigating to Code -> Insert Section or using the keyboard shortcut CTRL + SHIFT + R within the R Studio GUI.

Comments allow for description of code chunks and may delineate what the chunks do or why they are being executed. Using comments is advantageous for both you and others who may attempt to reproduce your analyses. This is because comments can tell you exactly what a line of code is doing, allowing you to easily repurpose this code to solve similar problems elsewhere. Further, comments may tell you *why* a line of code is in place, allowing you to keep track of why you took the steps you did. For example, if you were to transform a dataset or apply a correction to an analysis, including a comment that specifies why this change was made will ensure that you can justify the change to someone who is reviewing or attempting to reproduce your analyses.

Making use of sections is also advantageous. Processing and analysis of data can span thousands of lines of code depending on the degree to which the data needs to be cleaned and the nature of the analyses performed. Using sections can allow you to easily navigate to relevant parts of the code depending on what you are attempting to do. Broadly, using comments and sections makes your code clearer, more readable, and easier to navigate for you and others.

Q4. Refer to “Question 4” section in the R script for commented definition of the function and proof that it works.

> evenProd(2, 6, 8)

[1] 96

> evenProd(2, 6, 7)

[1] 15

Q5.

# Create variable to store the data and define this variable as data that is read in from a file.

# Clean up the data

# Select the relevant columns in the data to remove irrelevant variables

# Rename the relevant columns for ease of use

# Recode the condition column as a factor with 2 levels for clarity; name levels

# Remove the old and now redundant condition column

# Store the output of this cleaning as a new dataframe

# Add columns for log-transformed response times (RTs) and accuracy to dataframe

# Create new column that contains log-transformed RT values for each row

# Create new column that contains logical values corresponding to whether or not a value for RT exists in that row

# Use not operator to reverse these values so that rows containing a value for RT are TRUE and rows missing a value are FALSE

# Convert logical values to zeroes and ones

# Store the modified dataframe as a new variable

# Use write function to write the modified dataframe to an RDS file

# Group modified dataframe by subject ID and condition

# Summarize the data, creating new columns containing the mean RT and log RT by participant and condition

# Store the output as a new dataframe in a new variable

# Pass mean RT values from the summarized dataframe to the base histogram function to create histogram

# Pass mean log-transformed RT values from the summarized dataframe to the base histogram function to create histogram

# Pass mean RT values from the summarized dataframe to ggplot’s histogram function to create histogram

# Pass mean log-transformed RT values from the summarized dataframe to ggplot’s histogram function to create histogram

# Filter the summarized dataframe based on the condition column such that only congruent means are retained

# Save the output as a new dataframe

# Use the summary function to inspect mean raw RTs and log-transformed RTs

# Filter the summarized dataframe based on the condition column such that only incongruent means are retained

# Save the output as a new dataframe

# Use the summary function to inspect mean raw RTs and log-transformed RTs

# Call ANOVA function using mean RTs as a dependent variable, subject ID as a subject identifier, and condition (congruent versus incongruent) as a within-subject factor

# Specify call to stats function using the same arguments

# Call ANOVA function using mean log-transformed RTs as a dependent variable, subject ID as a subject identifier, and condition (congruent versus incongruent) as a within-subject factor

# Specify call to stats function using the same arguments

# Pass the output of the stats function to ggplot’s bar plot function, with mean RTs as y-axis values, condition labels as x-axis values, standard deviations as error bars

# Pass the output of the stats function to ggplot’s bar plot function, with mean log-transformed RTs as y-axis values, condition labels as x-axis values, standard deviations as error bars

Q6 – Q10. Refer to corresponding “Question X” sections in the R script.

Q11.

Base Graphics:

A graph of a graph

Description automatically generated

A graph of a graph

Description automatically generated

GGplot versions:

A graph of a graph

Description automatically generated

A graph of a graph

Description automatically generated

Raw response times appear to be positively skewed, whereas log-transformed response times appear to resemble a normal distribution to some degree. Based on visual inspection of the histograms, log-transformed response times appear to be better suited to ANOVA relative to raw response times due to the former’s relative normality.

Q12. Refer to “Question 12” section in the R script for creation and summary of the dataframes.

Based on informal inspection of means for congruent and incongruent trials, it appears that mean raw and log-transformed response times are greater for incongruent trials relative to congruent trials. This implies that on average, participants took longer to respond to incongruent trials relative to congruent trials. Accordingly, I predict a significant effect of trial condition on both raw and log-transformed response times, such that response times are significantly greater for incongruent trials relative to congruent trials.

Q13. Refer also to the “Question 13” section in the R script. Raw output:

> rt\_anova

$ANOVA

Effect DFn DFd SSn SSd F p p<.05 ges

1 (Intercept) 1 35 29948574 3575342.0 293.1748 1.378011e-18 \* 0.8749648

2 condition 1 35 1430293 704403.4 71.0676 6.048430e-10 \* 0.2504875

> rt\_stats

condition N Mean SD FLSD

1 Congruent 36 503.9998 174.3039 67.88277

2 Incongruent 36 785.8876 303.1445 67.88277

> log\_rt\_anova

$ANOVA

Effect DFn DFd SSn SSd F p p<.05 ges

1 (Intercept) 1 35 2824.013269 8.663617 11408.68403 1.329602e-45 \* 0.9965022

2 condition 1 35 3.391479 1.248758 95.05584 1.638976e-11 \* 0.2549245

> log\_rt\_stats

condition N Mean SD FLSD

1 Congruent 36 6.045745 0.341010 0.09038326

2 Incongruent 36 6.479814 0.408562 0.09038326

Q14. Not sure if you wanted both rt\_stats and log\_rt\_stats and forgot to specify; included both.

A graph of a bar graph

Description automatically generated with medium confidence

A graph of a graph

Description automatically generated with medium confidence

*Note*. Error bars show standard deviation of the mean.

Q15.

Mean raw response times in milliseconds were analyzed using a one-way analysis of variance (ANOVA) with condition (congruent versus incongruent) as a within-subject factor. The ANOVA revealed a significant effect of condition on raw response times, such that participants responded slower to incongruent trials (*M* = 785.89, *SD* = 303.14) relative to congruent trials (*M* = 504.00, *SD* = 174.30), *F*(1, 35) = 71.10, *p* < .001, η­­2­G = 0.25.

Mean log-transformed response times in were analyzed using a separate one-way ANOVA with condition as a within-subject factor. The ANOVA revealed a significant effect of condition on log-transformed response times, such that participants responded slower to incongruent trials (*M* = 6.48, *SD* = 0.41) relative to congruent trials (*M* = 6.05, *SD* = 0.34), *F*(1, 35) = 95.10, *p* < .001, η­­2­G = 0.25.

Taken together, the results of these analyses suggest that participants in this sample exhibited a typical Stroop effect (i.e., slower responses for incongruent relative to congruent trials). This pattern persisted regardless of whether analyses were conducted on raw response times – which are positively skewed, because participants cannot exhibit response times faster than 0 ms – or the more normally distributed log-transformed response times.