⚡ R Programming Ultimate Guide ⚡

🔷 Data Types

- 🔢 **Numeric** : Real or decimal numbers
- 📊 **Integer** : Whole numbers
- 📝 **Character** : Text strings
- 📦 **Factor** : Categorical variables with levels
- 📅 **Date** : Dates (YYYY-MM-DD format)
- 📅 **POSIXct** : Dates and times (seconds since 1970-01-01 00:00:00 UTC)
- 📈 **Logical** : True or false values
- 📁 **List** : Collection of objects
- 📁 **Data frame** : Tabular data with columns of different types

🔷 Basic Operations

- ➕ **x + y** : Addition
- ➖ **x - y** : Subtraction
- ✖️ **x * y** : Multiplication
- ➗ **x / y** : Division
- ➗ **x %/% y** : Integer division
- 📈 **x ^ y** : Exponentiation
- 📈 **sqrt(x)** : Square root
- 📈 **abs(x)** : Absolute value
- 📈 **ceiling(x)** : Smallest integer greater than or equal to x
- 📈 **floor(x)** : Largest integer less than or equal to x
- 📈 **round(x, digits)** : Round x to the specified number of digits
- 📈 **trunc(x)** : Truncate x to an integer
- 📈 **min(x, y)** : Minimum value of x and y
- 📈 **max(x, y)** : Maximum value of x and y

By: Waleed Mousa

⚡ R Programming Ultimate Guide ⚡

🔷 Data Manipulation and Visualization

- 📥 *install.packages("package_name")* : **Install a package**
- 📥 *library(package_name)* : **Load a package**
- 📖 *read.csv("filename.csv")* : **Read a CSV file**
- 📖 *read.table("filename.txt")* : **Read a text file**
- 📤 *write.csv(data, "filename.csv")* : **Write data to a CSV file**
- 📤 *write.table(data, "filename.txt")* : **Write data to a text file**
- 📊 *dim(data)* : **Get the dimensions of a data frame**
- 📊 *head(data)* : **View the first few rows of a data frame**
- 📊 *tail(data)* : **View the last few rows of a data frame**
- 📊 *summary(data)* : **View summary statistics of a data frame**
- 📊 *subset(data, subset)* : **Subset a data frame based on a condition**
- 📊 *merge(data1, data2)* : **Merge two data frames**
- 📊 *aggregate(data, by)* : **Aggregate data by a specified variable**
- 📊 *tapply(data, by)* : **Apply a function to subsets of data based on a specified variable**
- 📊 *transform(data, new_var = f(var))* : **Add a new variable to a data frame based**
- 📊 *sort(data, decreasing = FALSE)* : **Sort a data frame by one or more variables**
- 📊 *order(data, decreasing = FALSE)* : **Order a data frame by one or more variables**
- 📊 *unique(data)* : **Remove duplicate rows from a data frame**
- 📊 *duplicated(data)* : **Identify duplicate rows in a data frame**
- 📊 *which(data > value)* : **Find the indices of values that meet a condition**
- 📊 *is.na(data)* : **Identify missing values in a data frame**
- 📊 *na.omit(data)* : **Remove missing values from a data frame**
- 📊 *complete.cases(data)* : **Identify complete cases in a data frame**

By: Waleed Mousa

⚡ R Programming Ultimate Guide ⚡

├ 📊 *reshape(data, idvar, timevar, direction)* **: Reshape a data frame from long to wide or vice versa**

├ 📊 *melt(data, id.vars, measure.vars)* : **Reshape a data frame from wide to long**

└ 📊 *dcast(data, formula, fun.aggregate)* **: Reshape a data frame using an aggregation function**

├ 📥 *library(dplyr)* **: Load the dplyr package for data manipulation**

├ 📥 *library(tidyr)* : **Load the tidyr package for data manipulation**

├ 📥 *library(ggplot2)* **: Load the ggplot2 package for graphics**

├ 📥 *library(lubridate)* **: Load the lubridate package for working with dates and times**

├ 📥 *library(stringr)* **: Load the stringr package for string manipulation**

├ 📊 *select(data, variables)* **: Select variables from a data frame**

├ 📊 *filter(data, condition)* **: Filter rows from a data frame based on a condition**

├ 📊 *group_by(data, variable)* **: Group a data frame by a variable**

├ 📊 *summarise(data, variable = function)* **: Apply a function to a variable in a grouped data frame**

├ 📊 *mutate(data, variable = function)* **: Create a new variable in a data frame**

├ 📊 *arrange(data, variables)* **: Arrange rows in a data frame by variables**

├ 📊 *gather(data, key, value, columns)* **: Convert multiple columns into key-value pairs**

├ 📊 *spread(data, key, value)* **: Convert key-value pairs into multiple columns**

├ 📊 *separate(data, column, into = c("new1", "new2", ...))* **: Separate a column into multiple columns**

├ 📊 *unite(data, new_column, columns, sep = "")* **: Combine multiple columns into a single column**

├ 📊 *merge(data1, data2, by = "variable")* **: Merge two data frames by a variable**

├ 📊 *ggplot(data, aes(x, y)) + geom_*(...)* **: Create a plot using ggplot2**

├ 📥 *library(ggmap)* **: Load the ggmap package for working with maps**

By: Waleed Mousa

⚡ R Programming Ultimate Guide ⚡

├ 📊 *get_map(location)* : **Get a map for a location**

├ 📊 *ggmap(map) + geom_*(...)* : **Create a map plot using ggmap**

├ 📥 *library(ggvis)* : **Load the ggvis package for interactive graphics**

├ 📊 *ggvis(data, ~x, ~y) %>% layer_*(...) %>% ...* : **Create an interactive plot using ggvis**

├ 📥 *library(plotly)* : **Load the plotly package for interactive graphics**

├ 📊 *plot_ly(data, x = ~x, y = ~y, ...) %>% add_*() %>% ...* : **Create an interactive plot using plotly**

└ 📥 *library(leaflet)* : **Load the leaflet package for interactive maps**

🔷 Control Structures

├ 📚 *if (condition) {expression}* : **Execute an expression if a condition is true**

├ 📚 *ifelse(condition, true_expression, false_expression)* : **Execute one expression if a condition is true and another if it is false**

├ 📚 *for (variable in sequence) {expression}* : **Execute an expression for each element in a sequence**

├ 📚 *while (condition) {expression}* : **Execute an expression while a condition is true**

├ 📚 *repeat {expression}* : **Execute an expression indefinitely until a break statement is encountered**

├ 📚 *break* : **Exit a loop**

├ 📚 *next* : **Skip an iteration in a loop**

├ 📚 *return(value)* : **Return a value from a function**

├ 📥 *source("filename.R")* : **Run R code from a file**

├ 📥 *setwd("path")* : **Set the working directory**

└ 📥 *getwd()* : **Get the current working directory**

By: Waleed Mousa

⚡ R Programming Ultimate Guide ⚡

🔷 Data Import and Export

├ 📥 *library(readr)* : **Load the readr package for data import**

├ 📥 *library(readxl)* : **Load the readxl package for Excel files**

├ 📥 *library(haven)* : **Load the haven package for SPSS, SAS, and Stata files**

├ 📥 *library(dplyr)* : **Load the dplyr package for data manipulation**

├ 📥 *library(tidyr)* : **Load the tidyr package for data manipulation**

├ 📥 *library(tidyverse)* : **Load the tidyverse package for data manipulation and visualization**

├ 📤 *write.csv(data, file)* : **Export data as a CSV file**

├ 📤 *write_excel_csv2(data, file)* : **Export data as an Excel file**

├ 📤 *write.table(data, file)* : **Export data as a text file**

├ 📥 *read.csv(file)* : **Import data from a CSV file**

├ 📥 *read_excel(file)* : **Import data from an Excel file**

├ 📥 *read_spss(file)* : **Import data from an SPSS file**

├ 📥 *read_sas(file)* : **Import data from a SAS file**

├ 📥 *read_stata(file)* : **Import data from a Stata file**

├ 📊 *separate(data, column, into = c("new1", "new2", ...))* : **Separate a column into multiple columns**

├ 📊 *gather(data, key, value, columns)* : **Convert multiple columns into key-value pairs**

└ 📊 *spread(data, key, value)* : **Convert key-value pairs into multiple columns**

By: Waleed Mousa

⚡ R Programming Ultimate Guide ⚡

🔷 Functions

├ 📚 *function(arguments) {expression}* : **Define a function**

├ 📚 *return(value)* : **Return a value from a function**

├ 📚 *formals(function)* : **Get the formal arguments of a function**

├ 📚 *missing(argument)* : **Check if an argument was supplied to a function**

├ 📥 *source("filename.R")* : **Run R code from a file**

├ 📥 *setwd("path")* : **Set the working directory**

└ 📥 *getwd()* : **Get the current working directory**

├ 📊 *args(function)* : **View the arguments of a function**

├ 📊 *exists("var")* : **Check if a variable exists in the current environment**

├ 📊 *ls()* : **View the objects in the current environment**

├ 📊 *rm("var")* : **Remove a variable from the current environment**

├ 📊 *source("file.R")* : **Run code from a file**

└ 📊 *sink("file.txt")* : **Redirect output to a file**

🔷 Graphics

├ 📥 *library(ggplot2)* : **Load the ggplot2 package**

├ 🎨 *ggplot(data, aes(x = x_var, y = y_var)) + geom_point()* : **Create a scatter plot**

├ 🎨 *ggplot(data, aes(x = x_var, y = y_var)) + geom_line()* : **Create a line plot**

├ 🎨 *ggplot(data, aes(x = x_var, y = y_var)) + geom_bar(stat = "identity")* : **Create a bar plot**

├ 🎨 *ggplot(data, aes(x = x_var, fill = fill_var)) + geom_bar()* : **Create a stacked bar plot**

├ 🎨 *ggplot(data, aes(x = x_var, fill = fill_var)) + geom_histogram()* : **Create a histogram**

By: Waleed Mousa

⚡ R Programming Ultimate Guide ⚡

├ 🎨 *ggplot(data, aes(x = x_var, y = y_var, fill = fill_var)) + geom_boxplot()* : ***Create a box plot***

├ 🎨 *ggplot(data, aes(x = x_var, y = y_var, fill = fill_var)) + geom_errorbar()* : **Create an error bar plot**

├ 🎨 *ggplot(data, aes(x = x_var, y = y_var, fill = fill_var)) + geom_smooth()* : **Create a smooth line plot**

├ 🎨 *ggplot(data, aes(x = x_var, y = y_var, fill = fill_var)) + facet_grid(facets)* : **Create a grid of plots**

├ 🎨 *ggplot(data, aes(x = x_var, y = y_var, fill = fill_var)) + labs(x = "X axis label", y = "Y axis label", title = "Plot title")* : **Add labels and a title to a plot**

├ 🎨 *ggplot(data, aes(x = x_var, y = y_var, fill = fill_var)) + theme_bw()* : **Set the plot theme to black and white**

├ 🎨 *ggplot(data, aes(x = x_var, y = y_var, fill = fill_var)) + theme_classic()* : **Set the plot theme to a classic style**

├ 🎨 *ggplot(data, aes(x = x_var, y = y_var, fill = fill_var)) + scale_x_continuous(limits = c(min_value, max_value), breaks = seq(min_value, max_value, step))* : **Set the limits and tick marks for the x-axis**

├ 🎨 *ggplot(data, aes(x = x_var, y = y_var, fill = fill_var)) + scale_y_continuous(limits = c(min_value, max_value), breaks = seq(min_value, max_value, step))* : **Set the limits and tick marks for the y-axis**

├ 🎨 *ggplot(data, aes(x = x_var, y = y_var, fill = fill_var)) + scale_fill_manual(values = c("color1", "color2", ...))* : **Set the color scale for the fill variable**

├ 🎨 *ggplot(data, aes(x = x_var, y = y_var, fill = fill_var)) + scale_color_manual(values = c("color1", "color2", ...))* : **Set the color scale for the color variable**

└ 🎨 *ggplot(data, aes(x = x_var, y = y_var, fill = fill_var)) + scale_size_continuous(range = c(min_size, max_size))* : **Set the size scale for the size variable**

By: Waleed Mousa

⚡ R Programming Ultimate Guide ⚡

🔷 Statistics

├ 📥 *library(stats)* : **Load the stats package**

├ 📊 *mean(data)* : **Calculate the mean of a vector or column in a data**

├ 📊 *sd(data)* : **Calculate the standard deviation of a vector or column in a data frame**

├ 📊 *var(data)* : **Calculate the variance of a vector or column in a data frame**

├ 📊 *cor(data)* : **Calculate the correlation matrix of a data frame**

├ 📊 *cov(data)* : **Calculate the covariance matrix of a data frame**

├ 📊 *lm(y ~ x, data)* : **Fit a linear regression model to the data**

├ 📊 *summary(lm_model)* : **View a summary of the linear regression model**

├ 📊 *predict(lm_model, new_data)* : **Make predictions using the linear regression model**

├ 📊 *glm(y ~ x1 + x2 + ..., data, family = "binomial")* : **Fit a generalized linear model to the data**

├ 📊 *summary(glm_model)* : **View a summary of the generalized linear model**

├ 📊 *predict(glm_model, new_data, type = "response")* : **Make predictions using the generalized linear model**

├ 📥 *library(lme4)* : **Load the lme4 package for mixed effects models**

├ 📊 *lmer(y ~ x1 + x2 + ... + (1|group), data)* : **Fit a linear mixed effects model to the data**

├ 📊 *summary(lmer_model)* : **View a summary of the linear mixed effects model**

└ 📊 *predict(lmer_model, new_data)* : **Make predictions using the linear mixed effects model**

├ 📥 *library(forecast) : Load the forecast package for time series analysis*

├ 📊 *auto.arima(data)* : **Fit an ARIMA model to the data**

├ 📊 *ets(data)* : **Fit an ETS model to the data**

By: Waleed Mousa

⚡ R Programming Ultimate Guide ⚡

├ 📊 *stl(data)* : **Decompose a time series into trend, seasonal, and remainder components**

├ 📊 *acf(data)* : **Calculate the autocorrelation function of a time series**

├ 📊 *pacf(data)* : **Calculate the partial autocorrelation function of a time series**

├ 📥 *library(nnet)* : **Load the nnet package for neural networks**

🔷 Machine Learning

├ 📥 *library(caret)* : **Load the caret package for machine learning**

├ 📥 *library(randomForest)* : **Load the randomForest package for random forests**

├ 📥 *library(gbm)* : **Load the gbm package for gradient boosting**

├ 📥 *library(xgboost)* : **Load the xgboost package for extreme gradient boosting**

├ 📊 *train(formula, data, method)* : **Train a machine learning model using cross-validation**

├ 📊 *predict(model, new_data)* : **Make predictions using a machine learning model**

├ 📊 *confusionMatrix(predictions, actuals)* : **Calculate the confusion matrix for a classification model**

├ 📊 *varImp(model)* : **Calculate the variable importance for a machine learning model**

├ 📊 *partialPlot(model, data, variable)* : **Create a partial dependence plot for a machine learning model**

├ 📥 *library(mlr)* : **Load the mlr package for machine learning**

├ 📊 *makeLearner("classif.ksvm")* : **Create a support vector machine classifier**

├ 📊 *makeLearner("regr.ranger")* : **Create a random forest regression model**

├ 📊 *train(learner, task)* : **Train a machine learning model using cross-validation**

By: Waleed Mousa

⚡ R Programming Ultimate Guide ⚡

├ 📥 *library(h2o)* : **Load the h2o package for machine learning**

├ 📊 *h2o.init()* : **Initialize the h2o package**

├ 📊 *h2o.importFile(path)* : **Import a file into h2o**

├ 📊 *h2o.glm(y = "response", x = c("predictor1", "predictor2", ...), training_frame = data)* : **Fit a generalized linear model in h2o**

├ 📊 *h2o.randomForest(y = "response", x = c("predictor1", "predictor2", ...), training_frame = data)* : **Fit a random forest model in h2o**

└ 📊 *h2o.deepLearning(y = "response", x = c("predictor1", "predictor2", ...), training_frame = data)* : **Fit a deep learning model in h2o**

🔷 Web Scraping

├ 📥 *library(rvest)* : **Load the rvest package for web scraping**

├ 📊 *html_text(html_nodes(page, css))* : **Extract text from HTML nodes**

├ 📊 *html_attr(html_nodes(page, css), "attribute")* : **Extract attributes from HTML nodes**

├ 📊 *html_table(html_nodes(page, css))* : **Extract tables from HTML nodes**

├ 📊 *html_form(page)* : **Extract forms from an HTML page**

├ 📊 *html_submit(form, "button")* : **Click a button in an HTML form**

├ 📥 *library(httr)* : **Load the httr package for HTTP requests**

├ 📊 *GET(url)* : **Send a GET request to a URL**

├ 📊 *POST(url, body)* : **Send a POST request to a URL with a request body**

└ 📥 *library(RSelenium)* : **Load the RSelenium package for automated web scraping**

└ 📊 *remDr$findElement(using = "css selector", "selector")$clickElement()* : **Click an element on a web page using RSelenium**

By: Waleed Mousa

⚡ R Programming Ultimate Guide ⚡

🔷 Text Analysis

├ 📥 *library(tm)* : **Load the tm package for text mining**

├ 📊 *Corpus(VectorSource(text))* : **Create a corpus from a character vector**

├ 📊 *tm_map(corpus, function)* : **Apply a function to a corpus**

├ 📊 *DocumentTermMatrix(corpus)* : **Create a document-term matrix from a corpus**

├ 📊 *findAssocs(dtm, term)* : **Find terms associated with a given term in a document-term matrix**

├ 📥 *library(quanteda)* : **Load the quanteda package for text analysis**

├ 📊 *corpus(text, docnames = c("doc1", "doc2", ...))* : **Create a corpus from a character vector with document names**

├ 📊 *tokens(corpus)* : **Split a corpus into tokens**

├ 📊 *dfm(tokens)* : **Create a document-feature matrix from tokens**

├ 📊 *textstat_frequency(dfm)* : **Calculate term frequencies from a document-feature matrix**

├ 📊 *textstat_collocations(dfm)* : **Find collocations from a document-feature matrix**

├ 📥 *library(quanteda.textmodels)* : **Load the quanteda.textmodels package for text classification**

├ 📊 *textmodel_nb(dfm, y)* : **Train a naive Bayes model on a document-feature matrix and target variable**

├ 📊 *textmodel_svm(dfm, y)* : **Train a support vector machine model on a document-feature matrix and target variable**

├ 📥 *library(wordcloud)* : **Load the wordcloud package for creating word clouds**

├ 📊 *wordcloud(words, freq, max.words = 100, colors = brewer.pal(8, "Dark2"))* : **Create a word cloud from a vector of words and their frequencies**

├ 📥 *library(ggplot2)* : **Load the ggplot2 package for creating visualizations**

└ 📊 *ggplot(data, aes(x, y)) + geom_*(...)* : **Create a plot using ggplot2**

By: Waleed Mousa

⚡ R Programming Ultimate Guide ⚡

💠 Time Series Analysis

├ 📥 *library(lubridate)* : **Load the lubridate package for working with dates and times**

├ 📊 *as_date(time)* : **Convert a character vector to a date**

├ 📊 *year(time)* : **Extract the year from a date**

├ 📊 *month(time)* : **Extract the month from a date**

├ 📊 *day(time)* : **Extract the day from a date**

├ 📊 *hour(time)* : **Extract the hour from a date-time**

├ 📊 *minute(time)* : **Extract the minute from a date-time**

├ 📊 *second(time)* : **Extract the second from a date-time**

├ 📥 *library(zoo)* : **Load the zoo package for working with time series**

├ 📊 *as.zoo(data)* : **Convert a data frame to a zoo object**

├ 📊 *rollmean(zoo, k)* : **Calculate the rolling mean of a time series with a window of k**

├ 📊 *rollapply(zoo, k, function)* : **Apply a function to a rolling window of size k in a time series**

├ 📥 *library(forecast)* : **Load the forecast package for time series forecasting**

├ 📊 *auto.arima(ts)* : **Automatically fit an ARIMA model to a time series**

├ 📊 *ets(ts)* : **Fit an exponential smoothing model to a time series**

├ 📊 *stl(ts)* : **Seasonal decomposition of a time series**

└ 📥 *library(tidyquant)* : **Load the tidyquant package for financial time series analysis**

└ 📊 *tq_get(symbol)* : **Get financial data for a symbol**

└ 📊 *tq_transmute(data, mutate_fun, select_fun, fill_fun)* : **Manipulate financial data using tidyverse-style syntax.**

By: Waleed Mousa

⚡ R Programming Ultimate Guide ⚡

🔷 Other Useful Functions

├ 📊 *which.max(data)* : **Return the index of the maximum value in a vector**

├ 📊 *which.min(data)* : **Return the index of the minimum value in a vector**

├ 📊 *table(data)* : **Create a frequency table of the values in a vector**

├ 📊 *summarize(data, variable = function)* : **Apply a function to a variable in a data frame**

├ 📊 *arrange(data, variable)* : **Sort a data frame by a variable**

├ 📊 *mutate(data, new_variable = function)* : **Create a new variable in a data frame based on an existing variable using a user-defined function or a built-in function.**

├ 📊 *select(data, variables)* : **Select variables from a data frame**

├ 📊 *filter(data, condition)* : **Filter rows from a data frame based on a condition**

├ 📊 *group_by(data, variable)* : **Group a data frame by a variable**

├ 📊 *summarise(data, variable = function)* : **Apply a function to a variable in a grouped data frame**

├ 📊 *ungroup(data)* : **Remove grouping from a data frame**

├ 📊 *merge(data1, data2, by = "variable")* : **Merge two data frames by a variable**

├ 📥 *library(dplyr)* : **Load the dplyr package for data manipulation**

├ 📥 *library(tidyr)* : **Load the tidyr package for data manipulation**

├ 📥 *library(ggplot2)* : **Load the ggplot2 package for graphics**

├ 📥 *library(stats)* : **Load the stats package for statistical analysis**

└ 📥 *library(caret)* : **Load the caret package for machine learning**

By: Waleed Mousa

⚡ R Programming Ultimate Guide ⚡

🔷 Resources

├ 🌐 **R documentation:** https://www.rdocumentation.org/

├ 🌐 **RStudio cheat sheets**: https://www.rstudio.com/resources/cheatsheets/

├ 📚 The R Graphics Cookbook by Winston Chang

├ 📚 R Graphics Cookbook, 2nd edition by Winston Chang

├ 📚 An Introduction to Statistical Learning with Applications in R by Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani

├ 📚 Machine Learning with R, 2nd edition by Brett Lantz

└ 📚 R for Data Science by Hadley Wickham and Garrett Grolemund

├ 📚 R Programming for Data Science by Hadley Wickham

├ 📚 Advanced R by Hadley Wickham

├ 📚 Text Mining with R by Julia Silge and David Robinson

├ 📚 Applied Time Series Analysis with R by Wayne A. Woodward and Henry L. Gray

└ 📚 Tidyquant: A Tidy Approach to Financial Analysis by Matt Dancho

🔷 Useful Packages

├ 📦 **dplyr** : Data manipulation

├ 📦 **tidyr** : Data manipulation

├ 📦 **ggplot2** : Data visualization

├ 📦 **rvest** : Web scraping

├ 📦 **quanteda** : Text analysis

├ 📦 **lubridate** : Working with dates and times

├ 📦 **zoo** : Working with time series

└ 📦 **forecast** : Time series forecasting

By: Waleed Mousa

⚡ R Programming Ultimate Guide ⚡

💠 Useful Websites

├ 📙 **RStudio :** https://www.rstudio.com/

├ 📙 **CRAN :** https://cran.r-project.org/

├ 📙 **Stack Overflow :** https://stackoverflow.com/questions/tagged/r

├ 📙 **R-bloggers :** https://www.r-bloggers.com/

└ 📙 **DataCamp :** https://www.datacamp.com/

├ 📙 **R for Data Science:** https://r4ds.had.co.nz/

├ 📙 **Advanced R:** https://adv-r.hadley.nz/

├ 📙 **R Graphics Cookbook:** https://r-graphics.org/

├ 📙 **Machine Learning with R:** https://www.tidyverse.org/

├ 📙 **Time Series Analysis and Its Applications:** https://www.stat.pitt.edu/stoffer/tsa4/

├ 📙 **Text Mining with R:** https://www.tidytextmining.com/

├ 📙 **Web Scraping with R:** https://www.r-bloggers.com/web-scraping-with-r-tutorial/

├ 📙 **R Studio Cheat Sheets:** https://www.rstudio.com/resources/cheatsheets/

└ 📙 **R Project for Statistical Computing:** https://www.r-project.org/

💠 Keyboard Shortcuts

├ 🎮 **Ctrl+Enter :** Run current line or selection

├ 🎮 **Ctrl+Shift+Enter :** Run current block

├ 🎮 **Ctrl+Alt+I :** Insert a new chunk

├ 🎮 **Ctrl+Alt+L :** Reformat current document

├ 🎮 **Ctrl+Shift+M :** Insert a markdown cell

├ 🎮 **Ctrl+Shift+K :** Insert a code cell above

└ 🎮 **Ctrl+Shift+J :** Insert a code cell below

By: Waleed Mousa