Module 4. Analytics in R

- Unidad 1. Analytics in R
- References
- Download

Unidad 1. Analytics in R

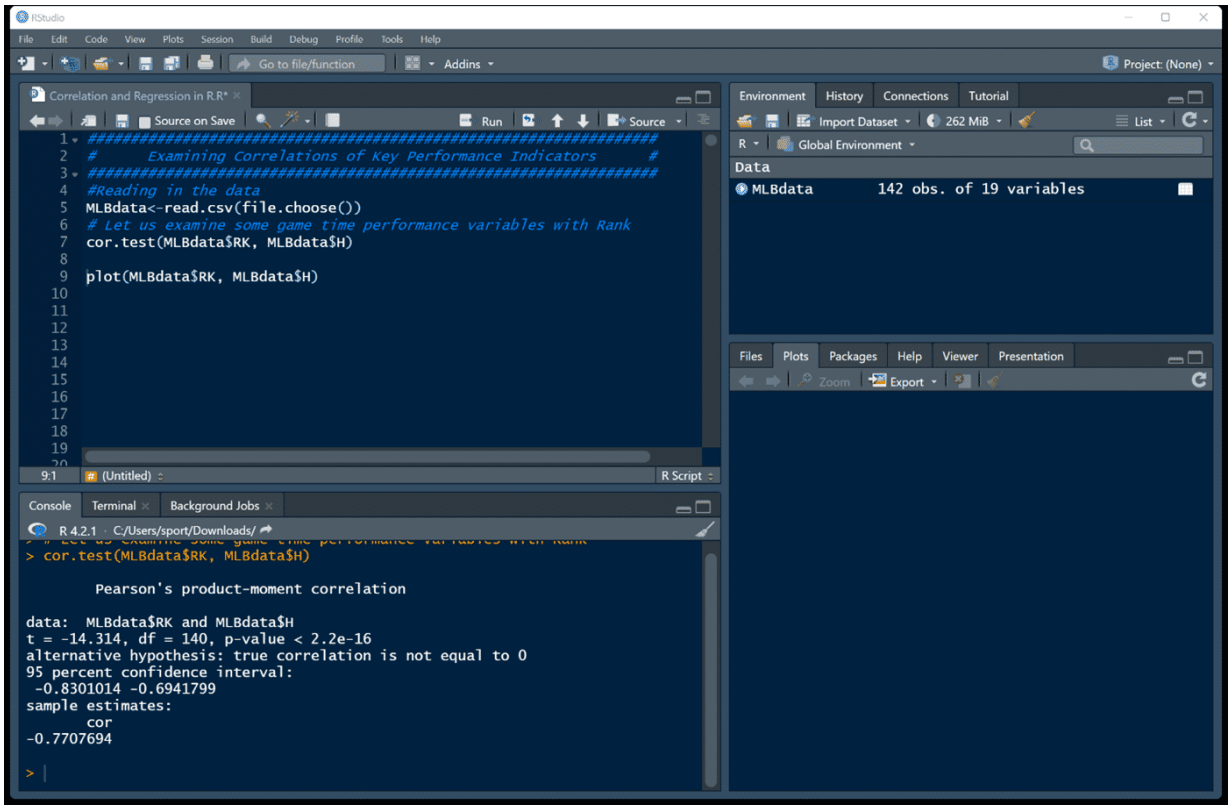
Simple linear regression

This module is designed to help you examine relationships and how certain factors influence another factor; for example, how specific key performance indicators (KPIs) influence performance. We will cover the most popular statistical model and most commonly implemented analysis (called regression), in particular, the ordinary least squares regression model. Learning about regression, we will set you up to learn how to train and test data, leading you to successfully implement supervised machine learning models. Although machine learning is not taught in this module, what you learn in this module will tee you up.

First, let us begin with examining associations between two numeric variables. This is called a correlational analysis, more commonly known as Pearson's product correlation coefficient.

In R, we are going to import the accompanying MLB dataset and type the code for the correlation function as shown below in the figure.

Figure 1. Correlation analysis in R examining the association between rank and hits



Source: screenshot of RStudio [Software], 2011.

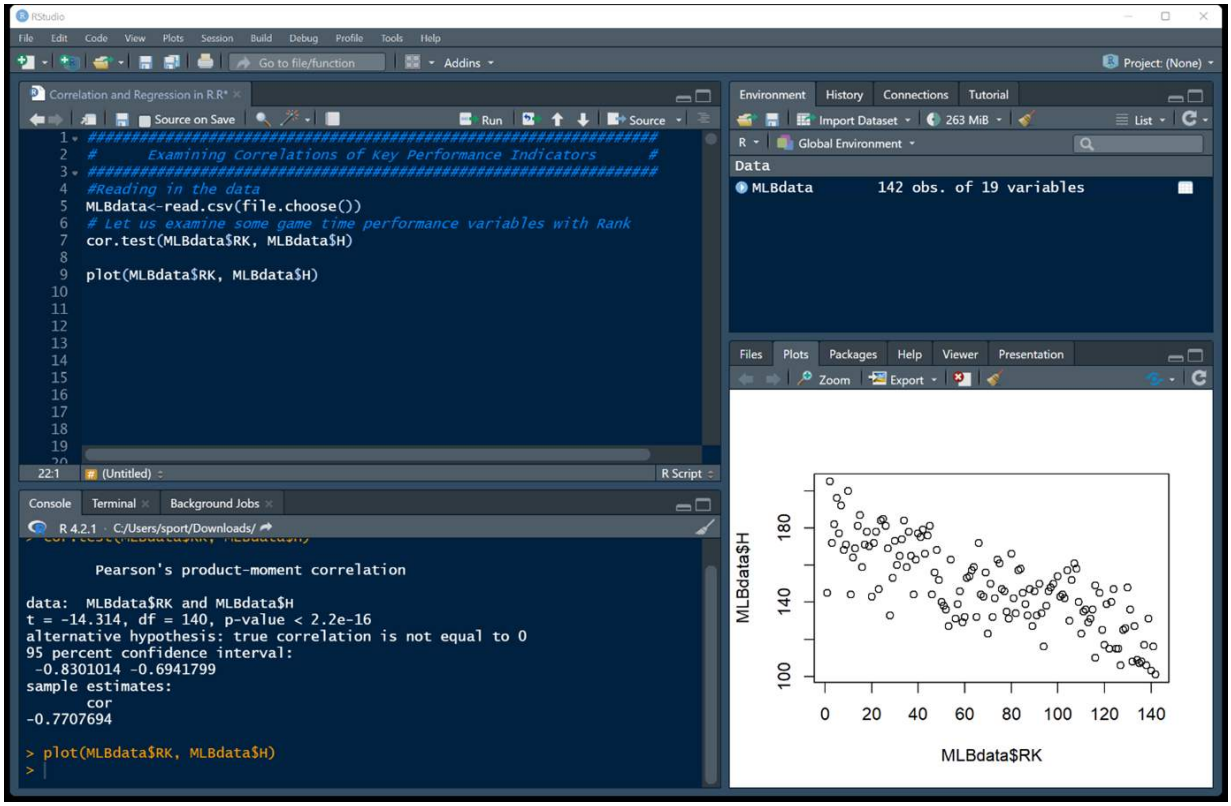
The figure above displays the correlation of -0.77 in the console, indicative of a negative correlation. This implies that, as there is an increase in hits, the ranking goes lower; thereby, there is a better rank (since there is lower ranking, a number closer to 1 is better).

However, to check that this correlation is accurate, we must also check for the assumption of linearity. If there is linearity, then the value of the correlation

is correct; otherwise, we should rethink another type of analysis that could take into account nonlinear associations.

Based on the figure below displaying a plot command, we can safely assume linearity and that the correlation coefficient of -0.77 is an accurate representation of the relationship between rank and hits.

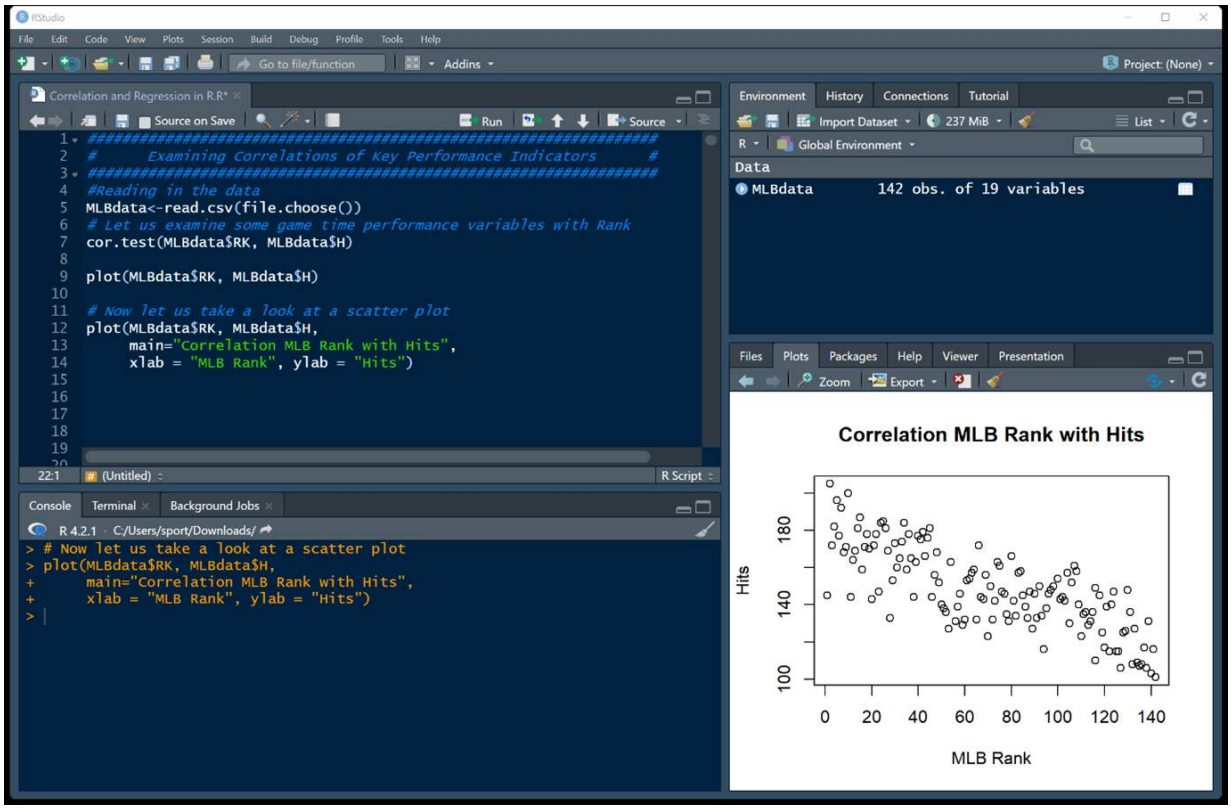
Figure 2. Basic plot of correlation between rank and hits



Source: screenshot of RStudio [Software], 2011.

Although we can embellish this plot with regular commands for the axis titles, as shown below, we strongly recommend that the "Grammar of Graphics" package, ggplot, be implemented as there are customisations that only that package is able to generate. Still, if in need of a quick plot, as is sometimes the case working in pro sports, a graph can quickly be generated with the plot command as shown in the figure below.

Figure 3. Basic R plot command with title and axis labels



Source: screenshot of RStudio [Software], 2011.

Since we have begun plotting, we will take a quick detour as we generated a basic graph using the plot(). This is a function that comes with the base R package, but graphing in R can be an art form if packages like ggplot are installed. Ggplot is a package named as an acronym for "grammar of graphics".

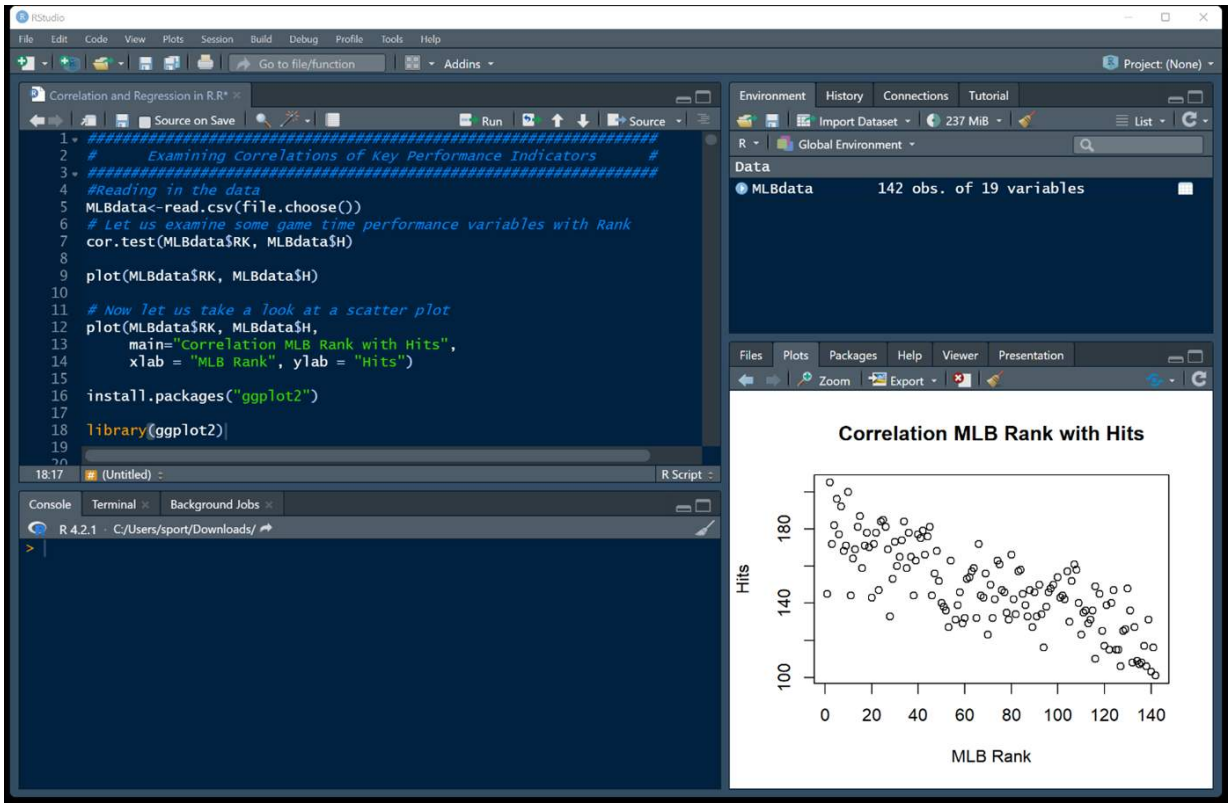
To install the Grammar of Graphics package, please use the following line of code:

```
Install.packages("ggplot2")

library(ggplot2)
```

Ensure that within the install.packages command, the ggplot2 function is within quotations and within the library function it is not as shown in the figure below:

Figure 4. Installing and calling the library of ggplot2, "Grammar of Graphics"




```
#################################################################
#        Examining Correlations of Key Performance Indicators   #
#################################################################
#Reading in the data
MLBdata<-read.csv(file.choose())
# Let us examine some game time performance variables with Rank
cor.test(MLBdata$RK, MLBdata$H)

plot(MLBdata$RK, MLBdata$H)

# Now let us take a look at a scatter plot
plot(MLBdata$RK, MLBdata$H,
     main="Correlation MLB Rank with Hits",
     xlab = "MLB Rank", ylab = "Hits")

install.packages("ggplot2")

library(ggplot2)
```

Source: screenshot of RStudio [Software], 2011.

It is important to identify the ggplot works on the base of layers. Included in the first line is the dataset followed by the aesthetic function, which includes the variable that you want to place on the x axis and then the variable that you want to take up the y axis followed by the geom_–which represents the geometric figure, whether it be a scatterplot, column, a boxplot, or a line chart.

Data visualisations are one of the key ways to communicate data and information. In this module, we are going to go over in detail which type of visuals are most effective based on the data type, the analyses, and the audience you wish to convey the information to.

In this algorithm, the line of best fit is identified by reducing the sum of squared errors, the errors more commonly referred to as residuals.

Regression equation for the population

Estimated regression equation for samples

Multiple linear regression is an extension of simple linear regression where multiple predictors are included in the model for evaluation of their contributed influence on the dependent variable.

Cross validation concepts

K fold

Holdout method

Inputting other values to

Training and testing datasets are commonly used in Machine Learning Algorithm Evaluation for the purpose of obtaining an assessment of the performance of the algorithm within the same dataset.

Logistic regression is a type of regression that is implemented when the outcome is binary.

Regular R base plots can be implemented, however, using the following command line: plot().

- **Ggplot package**

Whereas numeric variables may be better suited to be represented in some of the other following options:

- Scatterplots
- Barplots (column and bar charts)
- Boxplots

When discussing dispersion and distribution of the data, it is strongly recommended to become well versed in quartiles, quantiles, deciles, and percentiles, as well as in the most common visual representation of these summary statistics, which is known to be the boxplot.

Quartiles, by 25 percent of the data, are cut into chunks, thereby delineating the bottom 25% of the data typically denoted by a “whisker” in the boxplot or the “box and whisker plot”, which is termed the first quartile. Then, the second and third quartiles are commonly referred to as the box in the visual of the boxplot, which represent the middle chunk of the data where the

majority of the data is. A key pointer to keep in mind when reading a boxplot or box and whisker plot is that the second and the third quartile are split by a thin black line which many mistake for the average, but is, in fact, visible to represent the median, which is the fiftieth percentile cut point. Finally, the upper twenty-fifth percentile is the top whisker or typically seen to the furthest right and is representative of the top 25 data points in the distribution of the data.

Quantiles

Deciles

Percentiles

- Distributions
- Normal, uniform, standard normal, poisson
- PDF, CDF

A random variable is anything that we do not know that we would like to know. When we attempt to assign a probability to the random variable, then we can generate a probability distribution function that consists of the different values that random variable can take on along with the respective probability. This is called probability distribution function (PDF).

The probability distribution function is commonly used when we work with discrete random variables.

It is important to comprehend that when we work with continuous random variables such as time in a game or match, there is no exact probability for any particular value, meaning that the exact probability of any given value is actually 0. When we work with continuous random variables, we are examining the area, from 0 to that certain value or a range of values. Therefore, the cumulative distribution function is implemented (CDF).

If you are interested in Bayes' theorem and Bayesian analytics, this will provide the basics for obtaining probabilities for discrete random variables and probabilities for areas for continuous random variables.

The boxplot is a great visual when you want to display the distribution clearly to your audience. It does a great job when automatically showing the minimum, quartile 1, the median, quartile 3, and the max, as well as any outliers that may reside outside the minimum and maximums. In sum, the boxplot typically displays the well-known five number summary that is referred to as the basic descriptive statistics (more on boxplots in module 4).

Interestingly, in R and RStudio, when you implement summary() function, R yields a six number summary rather than a five number summary, which consists of the same statistics as the boxplot with the mean included as well.

Another function that is commonly run in RStudio to obtain descriptive statistics is the describe() function from the psych package, which provides a multitude of exploratory descriptives of the data.

General concept of statistics

Specific statistic

Difference between statistics and analytics

It is rare to find such a great display of statistical summary in a visual chart. Typically, the boxplot is depicted by a box and whiskers, with the box representing the interquartile range (IQR). The IQR can be calculated by subtracting Q1 from Q3, therefore:

IQR = Q3 – Q1

Figure of box and whisker plot including the five to six number summary.

- Boxplots
- Quartiles
- Summary
- Describe by

- Data visualisations

Another type of data visualisation that is commonly used is the scatterplot. What exactly is a scatterplot? It is a visual where you plot dots of your x and y variables. Bringing it back to basics, think of the x-y basic structure as depicted below:

Figure of an x-y plot

It is the mapping of x y coordinates on a 2D plane.

When should you use a scatterplot?

The main function of a scatterplot is showing the relationship between two variables.

If both the variables of interest are numerical, then it may be of interest to examine the correlation between them.

Now, what about when you want to display data over time and trends?

- Line chart (is the go to)

The line chart is the most commonly used visual for examining a variable over a given time period. The function to implement is geom_line() when working with the ggplot2 package in R and RStudio, as it is a powerful data visualisation package.

In this particular case, the dataset has a variable that is some sort of time component, whether it be minutes played, duration of the game, days training, months in the season, etc. The time variable is typically assigned to the x variable, and the y variable in the ggplot argument within R will take on the actual variable of interest. For example, if you want to track number of assists over time, then number of assists would go on the y variable and time on x.

How to troubleshoot when your data is inherently categorical and you want to generate a line chart?

In this scenario, you may have certain singular data points for each day of performance; however, in order to visualise this as a line, you literally will have to connect the dots. If you were to run the regular ggplot with geom_line() function, an error will yield, stating that there is a single data point for each time. Therefore, you have to connect them together using the group = argument within ggplot.

See the example figure and R code below: basic R syntax, better retain R knowledge by trying to solve a problem.

R basics, functions and data types

Vectors and sorting

Indexing data manipulation and plots

Programming basics, ifelse and forloops to commands

CONTINUE

References

Allaire, J. J. (2011). *RStudio* [Software]. Posit. https://posit.co/downloads/.

CONTINUE

Download

PDF

Module 4. Analytics in R.pdf

2.4 MB