Lesson 1: Introduction to R, RStudio, and Quarto

AUTHOR
John Fariss

PUBLISHED
April 3, 2025

Introduction

Welcome to Lesson 1! In this lesson, you will: - Set up an RStudio project - Learn how Quarto works - Render a simple Quarto document. But first, let's talk about R and RStudio.

R is a computer language designed for data analysis, and RStudio is a user-friendly interface that makes working with R much easier. For appraisers, RStudio offers a powerful way to manage, analyze, and visualize real estate data without needing to be a programmer. It's particularly well-suited for tasks like exploring market trends, calculating adjustments, and producing consistent, reproducible results, a key concept in Evidence Based Valuation.

One of the most valuable features for appraisers is RStudio's Project system. Projects function like digital workfiles, keeping all related files—your raw data, analysis scripts, visuals, and reports—organized in one place. This means everything tied to a specific assignment or market study is easy to access, review, or update later. Just like a paper workfile, but cleaner, searchable, and built for today's data-driven workflow.

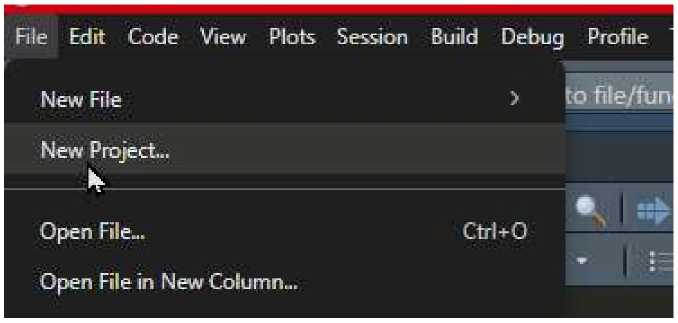
Quarto is ideal for appraisers because it allows you to combine your analysis, narrative, and visuals into a single, professional report that updates automatically when your data changes. It supports plain text writing alongside live R code, so you can calculate adjustments, insert charts, and summarize findings—all in the same document. This makes your work more transparent, reproducible, and efficient.

For appraisers moving beyond static spreadsheets and into evidence-based reporting, Quarto becomes an all-in-one tool for clear, data-backed communication. For now, we'll use it for our analysis to supplement our reporting. Don't expect to start out creating finished reports in Quarto…it takes time.

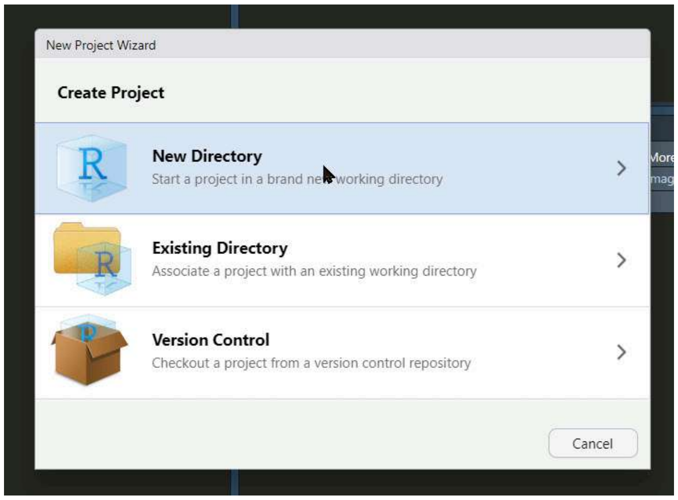
So, let's start by creating a project!

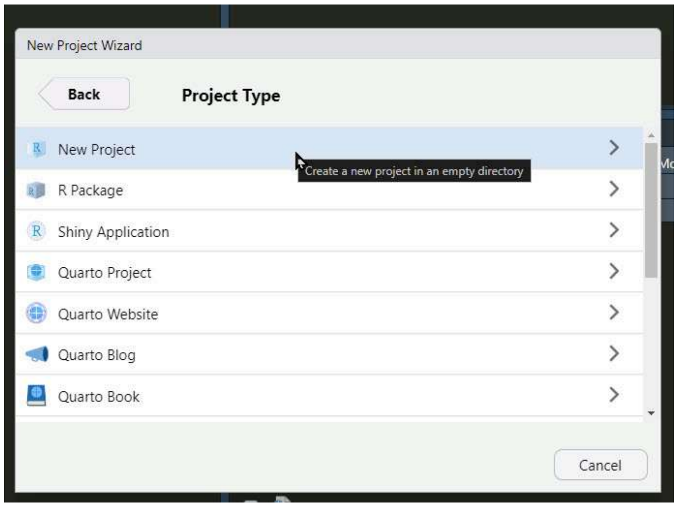
Step 1: Setting Up Your Project

1. Open **RStudio**.
2. Click **File > New Project**.

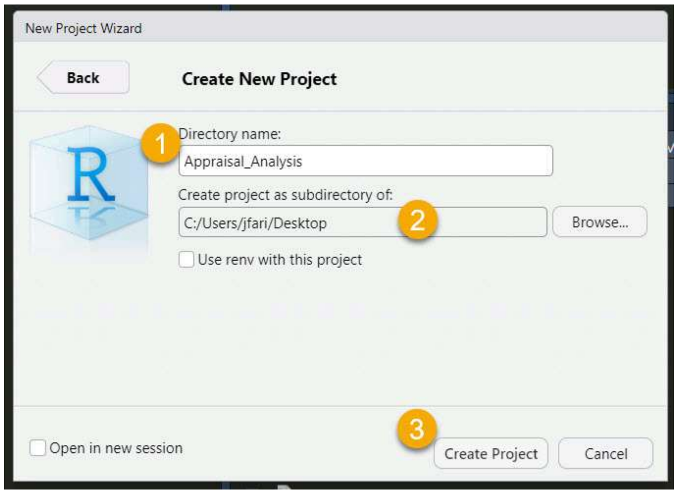


3. Choose **New Directory > New Project**.

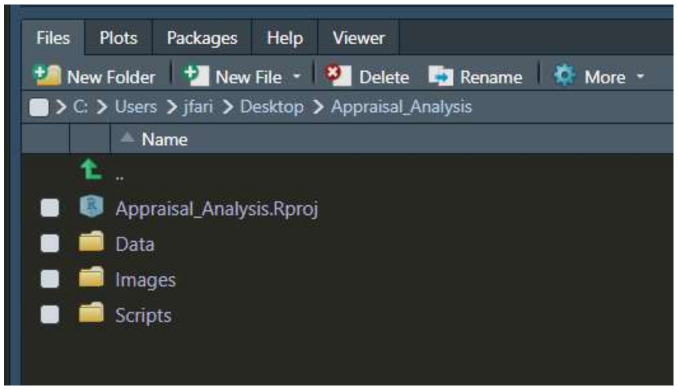

New Project Wizard
Create Project
New Directory
Start a project in a brand new working directory
Existing Directory
Associate a project with an existing working directory
Version Control
Checkout a project from a version control repository
Cancel



4. Name your project (e.g., `"Appraisal_Analysis"`) (1), select a folder (2) and click **Create Project** (3).



5. Inside your project, create **three folders**: *Data*, *Scripts*, and *Images*. A sample is shown below.



📁 This keeps all your files organized in one place. Create a new project for each appraisal assignment.

Step 2: Creating a New Quarto File

1. Go to **File > New File > Quarto Document...**
2. A window will pop up:
   - **Title**: Enter a title (e.g., `My First Report`)
   - **Author**: Enter your name
   - **Format**: Keep as HTML for now
3. Click **Create**

> 📄 This opens a new file with `.qmd` extension within your project folder. IT STILL NEEDS TO BE SAVED!

4. Go to **File > Save**
5. Save the file as something like `FirstQuarto.qmd` inside the Scripts folder of your project folder

That's It! 🎉

You've created **first Quarto document**! You can now:

- Add code chunks
- Write text in markdown
- Import data and create visualizations

Let's learn a bit more about quarto before moving on.

Step 3: Understanding Quarto

Quarto is a tool that allows you to render text, code, and output in one document. There are 3 primary components of a Quarto document: YAML header, text, and code chunks. Let's take a look at each.

YAML Header: Defines the document's format (see the `---` section at the top of each lesson and below). The YAML below contains several elements including metadata, output format, rendering options, editing options and params. The YAMLs in the lessons will be similar, but not exact. Each document can have its own YAML.

```
---
title: "Market Analysis"
author: "John Fariss"
date: today

format:
  html:
    embed-resources: true

execute:
  warning: false
  echo: false

editor: visual

params:
  SubAddress: 1234 Main St
  SubGLA: 1500
  SubNHood: SilverCreek

---
```

- This gives instructions on how the document should be edited and rendered.
- Params give us values to use in formulas throughout the document.
- In the lesson1.qmd document, we specify that the output format will be a single html file by specifying:

```
format:
  html:
    embed-resources: true
```

- Additionally, we specify that we want the document to open in **Visual** mode, which works more like a standard word processor.

```
editor: visual
```

Text: Written in Markdown (like this section). However, as we are in visual mode, this looks more like a standard word processor, and instead of having to know Markdown syntax, you can use the tool bars above.

Code Chunks: Used to run instructions, calculations and create data visualizations.

- Code chunks are inserted by typing / and selecting “R Code Chunk” or pressing ‘enter’ as “R Code Chunk” is the first option selected. They look like the box below.

```
{r Code_Chunk}

# Comments (text) must have a # at the front or else r will think it is a formula/code.
# Each code chunk can have a name as shown above. Enter a space after r and enter a name.
```

- Code is written similar to formulas in Excel. Call the function name, specify the data to use, and enter the formula options.

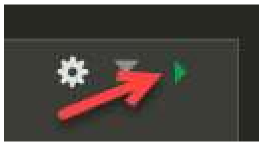
- Each function can have its own syntax, but most will show a prompt with the syntax, or you can highlight the function name and press F1 for help.

Let’s try using some of the features of Quarto.

> *Use the steps outlined above to create the chunks in Step 4 in your new Quarto file, or open “lesson1.qmd” and continue with Step 4 in it.*

Step 4: Using a Simple Quarto Document

Let’s test Quarto by running a simple calculation. Click the green arrow at the top right of the code chunk to run all code inside the chunk.



```
# A simple addition. Results will be displayed below this box.
5 + 5
```

```
[1] 10
```

The green arrow runs all the code inside the chunk. Results are typically output below the code chunk, but sometimes plots may appear in the Viewer Pane in the bottom right quadrant of RStudio.

In the instance above, our result should be 10. The [1] simply indicates there is 1 result from the calculation.

Let’s try using a function. A function is similar to a formula name in Excel. In the chunk below we will use the **head()** function to view the first few rows of a data frame. Remember: Use the green arrow to run all the code in the chunk.

```
# Assign the built-in data frame mtcars to df
df <- mtcars # <- is called the assignment operator. It assigns mtcars to df.
```

```r
# Now call the head function on df
head(df)
```

```
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

Some function names have the same name and algorithm as in Excel. Use the median function below to get the median horsepower (hp) from our dt data frame.

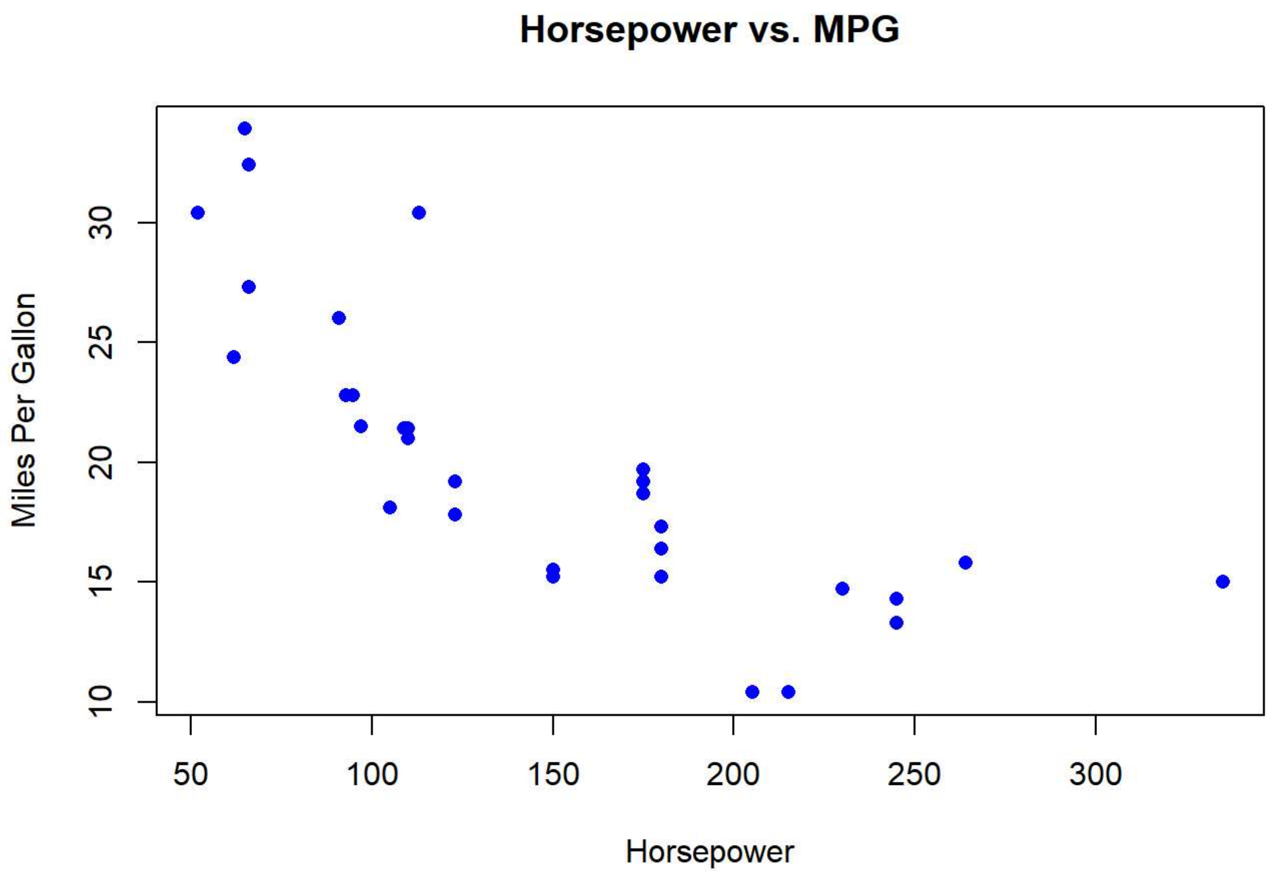
```r
median(df$hp) # specify the data frame and the variable
```

```
[1] 123
```

You should get a result of 123 for horsepower. Try changing the variable to another and running the chunk to get a new result.

Let's create a simple **scatter plot** showing horsepower vs. miles per gallon using base r commands.

```r
# create a scatter plot using base r functions
plot(df$hp, df$mpg, # Specify x and y variables to use
     main = "Horsepower vs. MPG", # Create a title
     xlab = "Horsepower", # Add nicer x label
     ylab = "Miles Per Gallon", # Add nicer y label
     pch = 16, # Make the points solid circles
     col = "blue") # Make points blue
```



Step 4: Rendering the Document

To generate the ouput document, which we've instructed should be an html document, click **Render** in the tool bar.

> *Tip! When rendering, you must import your data with code or use a built-in data set. Rendering does not work from the environment, so not all Quarto docs can be rendered.*

In the next lesson, we will **import and clean real estate market data**.