1. 1. Language name, origin/history, and design goals; and development tool availability
   1. R first appeared in August 1993 and has been used by statisticians and data miners for general data analysis ever since. Since its’ introduction, R has been slowly gaining in popularity due to an increase for demand in data analysis and machine learning models (citation: <https://en.wikipedia.org/wiki/R_(programming_language)> )
   2. R is most used in conjunction with RStudio development environment, both of which are free for use (for open source purposes). The software is regularly updated and currently available for Windows, Mac, and most common Linux distributions (source: <https://rstudio.com/products/rstudio/download/#download> ).
   3. R is procedural and dynamically typed. R programs, called as scripts, are executed on a line-by-line basis, generating global variables which can be used throughout a larger R project. This language and environment design makes it ideal for data analysists who need to reference multiple data sets in the context of the same statistical analysis.
2. 2. Programming paradigm(s) you used in your project. Is the language statically typed, dynamically typed? What data representations are available? Is the language compiled or interpreted?
   1. R is dynamically typed and interpreted. Most common “flow” of an R project includes importing the data from a CSV or an SQL format in the form of a data object. Each column can be accessed directly from the object and can be cast into common primitive data types. Packages are commonly used within R scripts to perform any meaningful data representation. Packages are installed directly by running a library function, and package functions can be invoked directly after.
3. 3. Discussion of language design criteria based on the material in Chapter 2 of the text. (Pearson)
   1. R is a functional language.
4. 4. Lessons learned while using the language on your project
   1. Most raw data sets need to be “trimmed” or “normalized” before performing data analysis. Most commonly this is done by recasting the data object with fewer columns.
   2. Individual data column types often need to be recast to an appropriate data type. For example, a column full of numbers is often imported as a “character” type, preventing numerical analysis from being performing. Recasting can be performing by invoking the “class” function.
   3. Global variables remain throughout the life of the project, regardless of the script being used. Renaming an object simply creates a new object with a different name. The “remove” function must be used regularly to keep the global environment clean of duplicate objects.