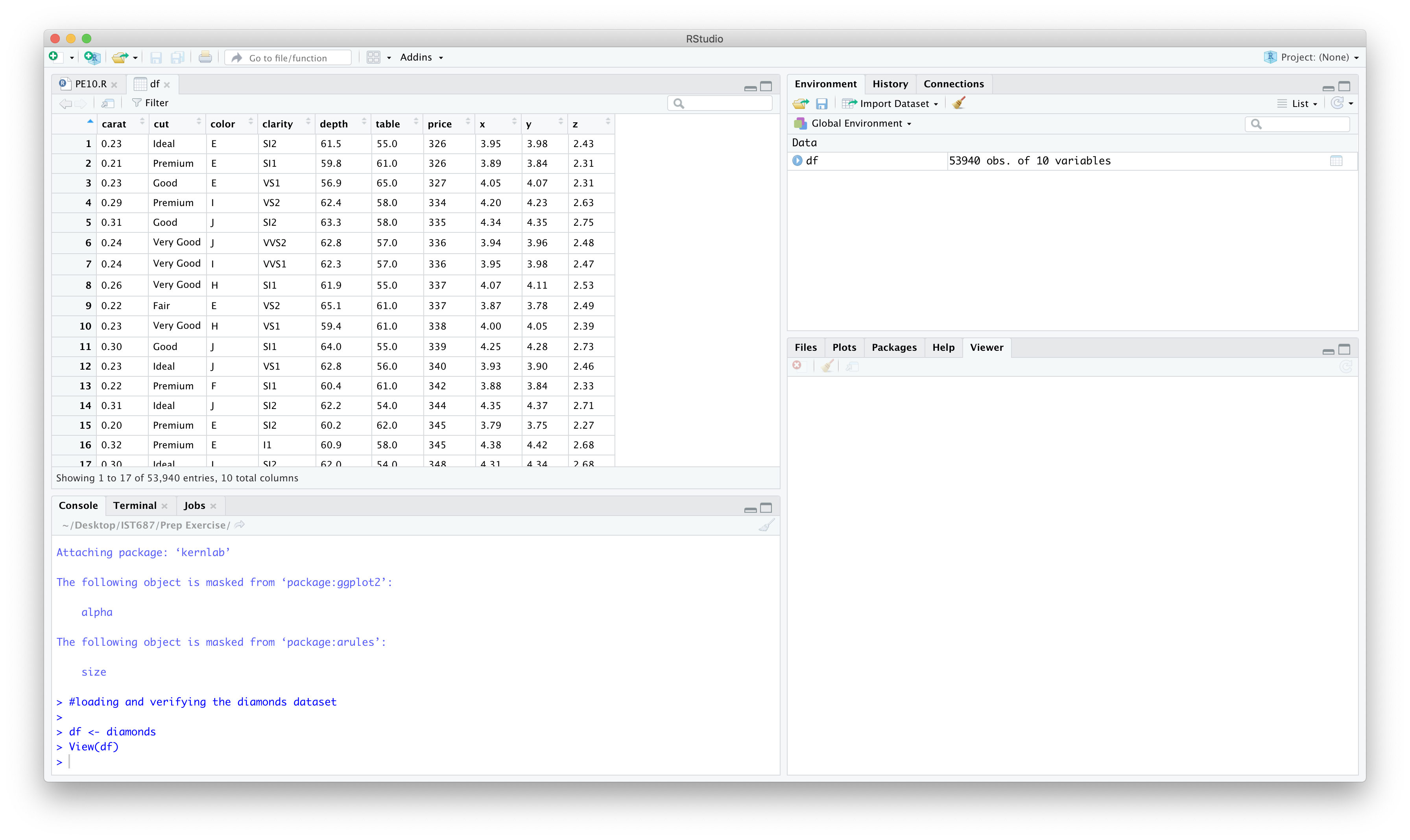
# Prep Exercise (PE10) Support Vector Machines (SVM)

### General Instructions

1. For this exercise you will answer all of the questions in this document and turn it in to Blackboard.
2. Before you get started make sure to read Chapter 18 of *An Introduction to Data Science* and execute the code throughout the chapter to gain familiarity.
3. Getting Started:
   1. Further exploring data mining techniques, this week we branch off from unsupervised learning techniques such as associated rules mining where there were no particular criteria we tried to predict, to a supervised learning technique called Support Vector Machines (SVM). Support vector machines (SVM) are a highly flexible and powerful method of doing supervised machine learning. Supervised learning means that there is a criterion one is trying to predict. The typical strategy is to divide data into a training set and a test set (for example, two-thirds training and one-third test), train the model on the training set, and then see how well the model does on the test set. For this week’s prep ex and homework, we will be using a chunk of the “diamonds” dataset from the ggplot2 package to do some classification with an SVM. For the sake of simplicity, we will see if we can correctly classify the “cut” of a diamond into one of the two categories, “premium” or “ideal”.
   2. As usual we will use this Prep Ex to set you up for the homework exercises and test your knowledge of materials within the chapter reading. Let’s begin…

### Prep Exercise

1. **Getting Ready: Loading and Verifying the Diamonds Dataset.**
   1. This week’s “diamonds” dataset comes from the ggplot2 package while the SVM function are located within a package called kernlab. Therefore, you will need to install and library ggplot2 and kernlab.
   2. Assign the “diamonds” dataset to a dataframe and view the dataset. You will notice that there are five cuts of diamonds: Fair, Good, Very Good, Premium and Ideal.
   3. View the dataframe and place a screenshot below. It is not important to fit the entire dataset in the screenshot, just enough to verify that you have completed the above steps.



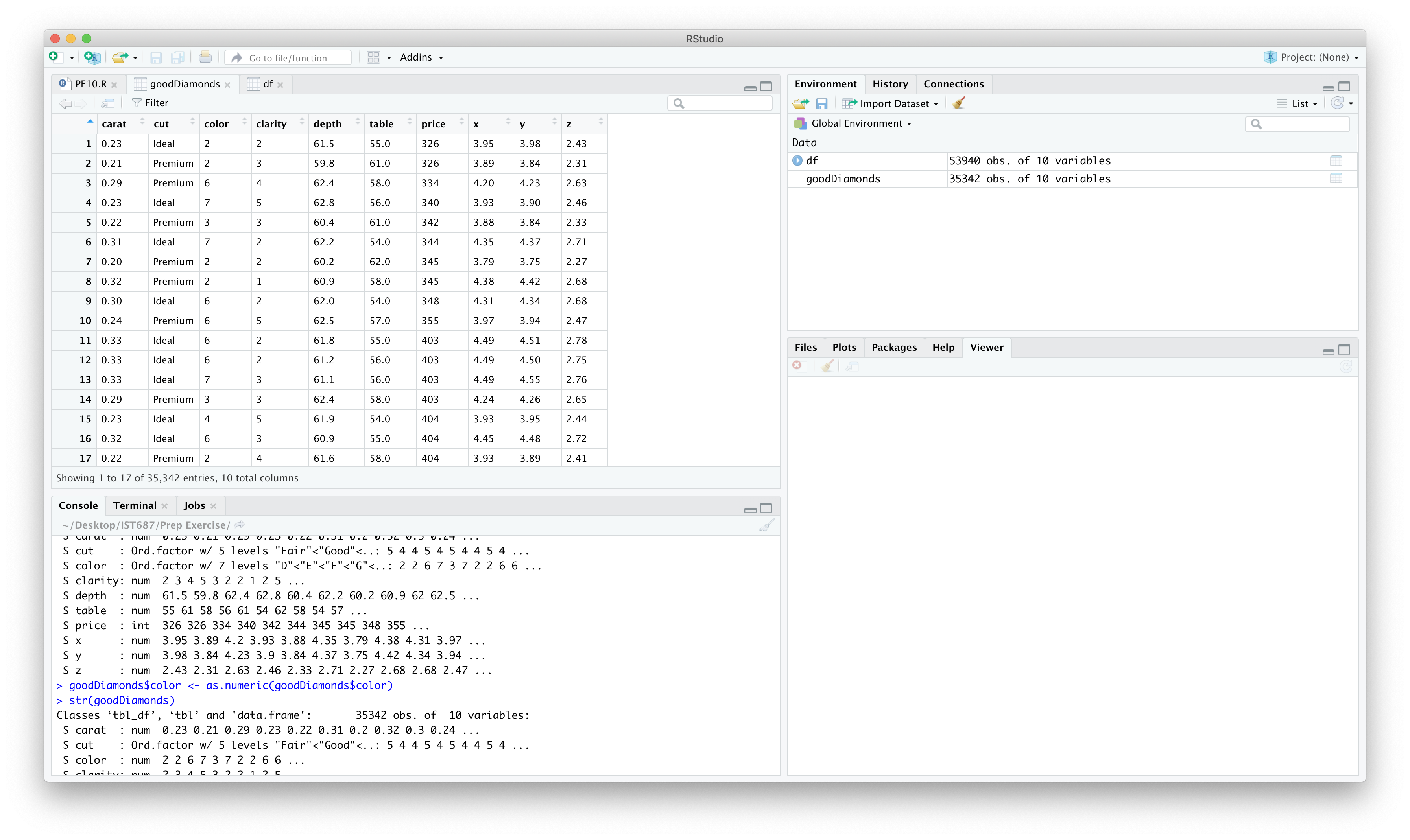
1. **Cleaning the data.**
   1. To simplify our classification task, we will focus only on Premium and Ideal cuts of diamonds. Below, there are two different ways to create the subset of the dataframe that fits the two “cut” categories. Create a new dataframe, called ‘goodDiamonds’ that only has the “Primium” and “Ideal” cut of diamons.
   2. The clarity and color variables in the dataframe are “ordered factors.” This means that for analytical purposes such as this, you can convert the factor level directly into a number and it will make sense. Convert the clarity and color variables into numbers.

(**Hint:** use the *as.numeric()* command to accomplish this)

* 1. The cut attribute now has two level (just premium and idea), but the dataframe still has the initial five factors. You can fix this by using the following code:

as.factor(as.character(goodDiamonds$cut))

* 1. View the adjusted dataframe and place a screenshot below.

****

* 1. In a block comment, describe the meaning of each variable within the dataframe.

**#the dataframe has 10 attributes**

**#carat: weight of diamond**

**#cut: quality of cut**

**#color: diamond color**

**#clarity: deterimes how clear the diamond is**

**#depth: total percentage depth**

**#table: width of top of diamond**

**#price: cost of diamond in USD**

**#x, y, z: length, width and depth in mm**

1. **Understanding Terminology that will be used in this PE and HW.**
   1. In a paragraph or two explain the concept of a confusion matrix and the theoretical process behind creating one in RStudio.

**Confusion matrix is a table that is used to determine the performance of a classifier on a set of test data for which the actual outcome is known. It has four main values**

* **True Positive (TP): when predicted value is positive and actual value is positive**
* **True Negative (TN): when predicted value is negative and actual value is negative**
* **False Positive (FP): when predicted value is positive and actual value is negative**
* **False Negative (TN): when predicted value is negative and actual value is positive**

**Confusion matrix can be created in R using confusionMatrix() function present in caret package. It mainly takes two parameters, data- a factor of predicted classes and reference- a factor of classes to be used as the true results. It displays a table for prediction and correct values.**

1. **List any additional resources that you used here.**

<https://www.rdocumentation.org/packages/caret/versions/3.45/topics/confusionMatrix>

1. **Be sure to save your R file as this will become the starting code for your homework.**

***You must submit all Prep Exercises to blackboard prior to the deadline specified for each assignment.*** PE assignments are due on the evening prior to the lecture class. Late PE assignments will not be accepted for credit.

*Materials provided by the instructors of this course are copyrighted 2018 by Jeffrey Saltz and Jeffrey Stanton, or by the respective instructor that produced them. These materials are provided to enrolled students for their exclusive use during the semester of instruction. Uploading instructor-produced materials to Internet sites (e.g., Coursehero) without permission is considered a breach of professional ethics and will be reported as a possible academic integrity violation.*