# Prep Exercise (PE11) Sentiment Analysis

### 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 Chapters 14 and 15 of *An Introduction to Data Science* and execute the code throughout the chapter to gain familiarity.
3. Getting Started:
   1. The Internet contains billions of unstructured documents that consist mainly of natural language text. For example, presidency.ucsb.edu contains hundreds of speeches made by presidents and presidential nominees. Locating and parsing text documents, restructuring them, filtering them, counting particular words or phrases, and creating visualizations – these are all critical skills for data scientists. The tm package provides an essential toolbox for manipulating text data.
   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…

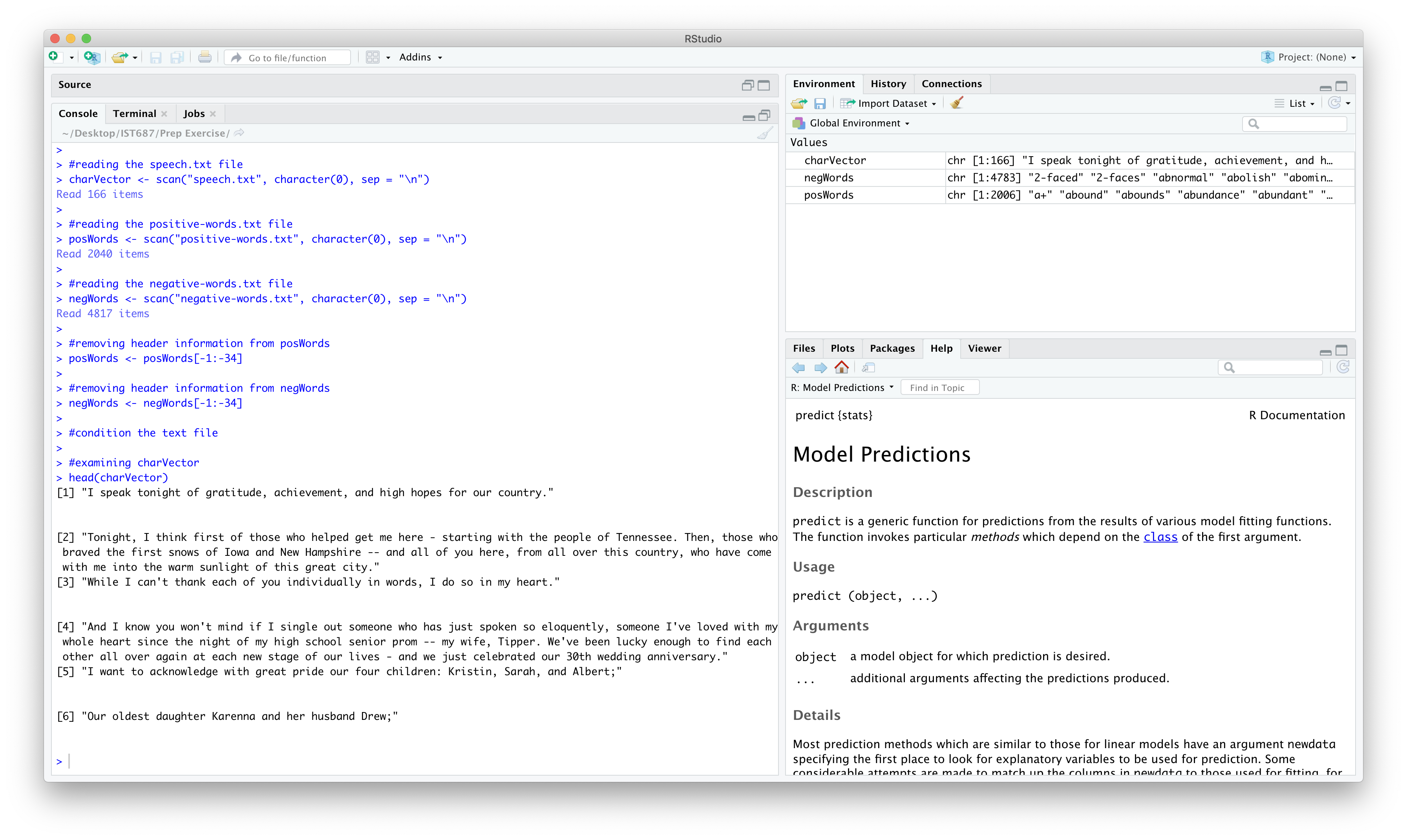
### Prep Exercise

1. **Getting Ready: Loading the data.**
   1. Install the “tm” package and library it.
   2. On Blackboard download the speech.txt, positive-words.txt, and negative-words.txt files. Set the working directory to point to the folder that contains the downloaded files.
   3. Read in the speech.txt file using the *scan()* function, the code is provided below.

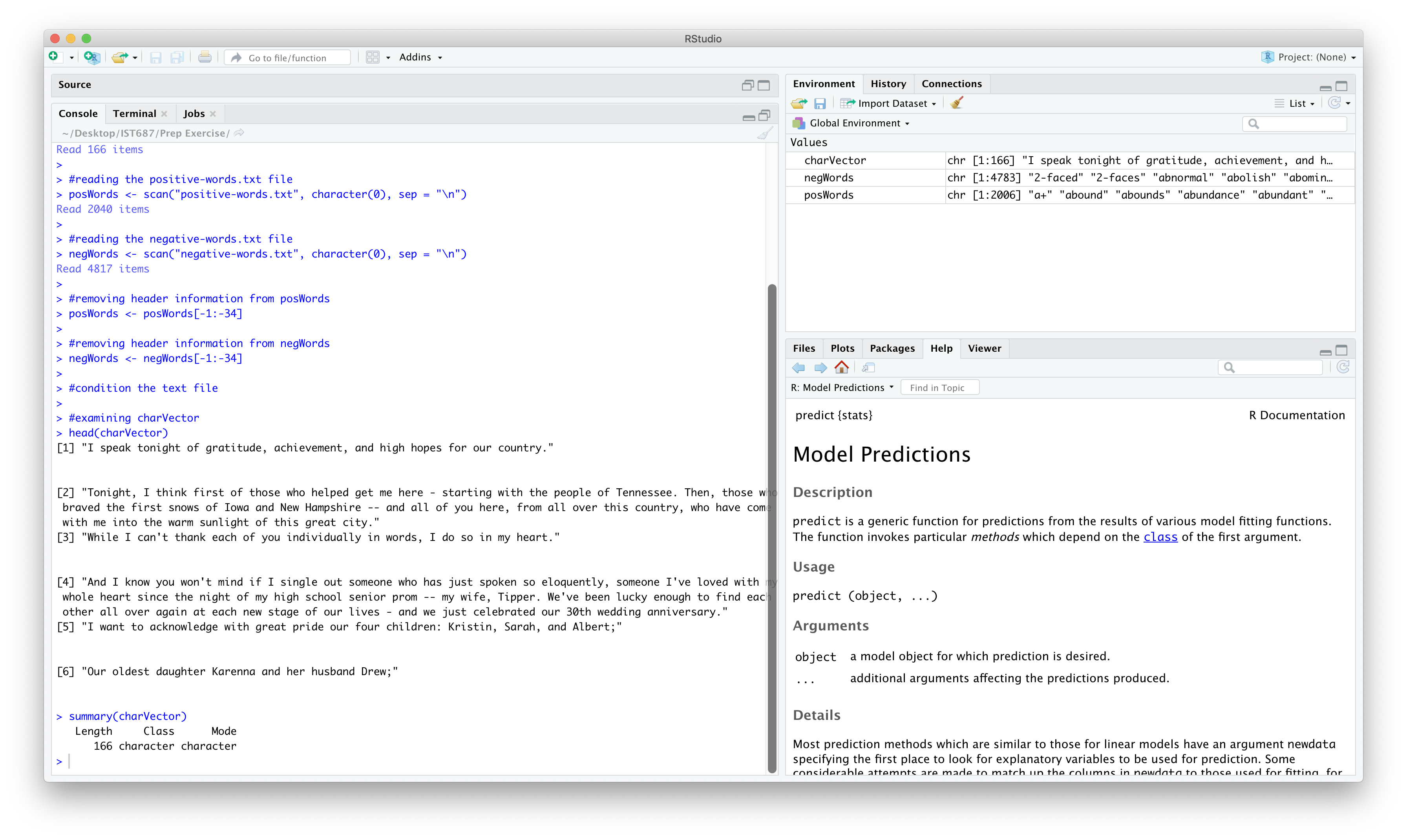
**charVector <- scan("speech.txt", character(0), sep = "\n")**

* 1. Using similar code read in the positive-words.txt and negative-words.txt files as “posWords” and “negWords” respectively. Be sure to remove any header information from those files (if you need help, review chapter 15 in the text book).

1. **Condition the text file.** 
   1. Examine charVector using the *head()* and *summary()* commands.
   2. Describe the items returned from the previous step and place a screenshot of each command’s output below (hint: There should be 166 total items in the charVector).

****

**head command returns the first six character elements in the charVector vector. Each element in this vector is a document (statement made by some individual).**

****

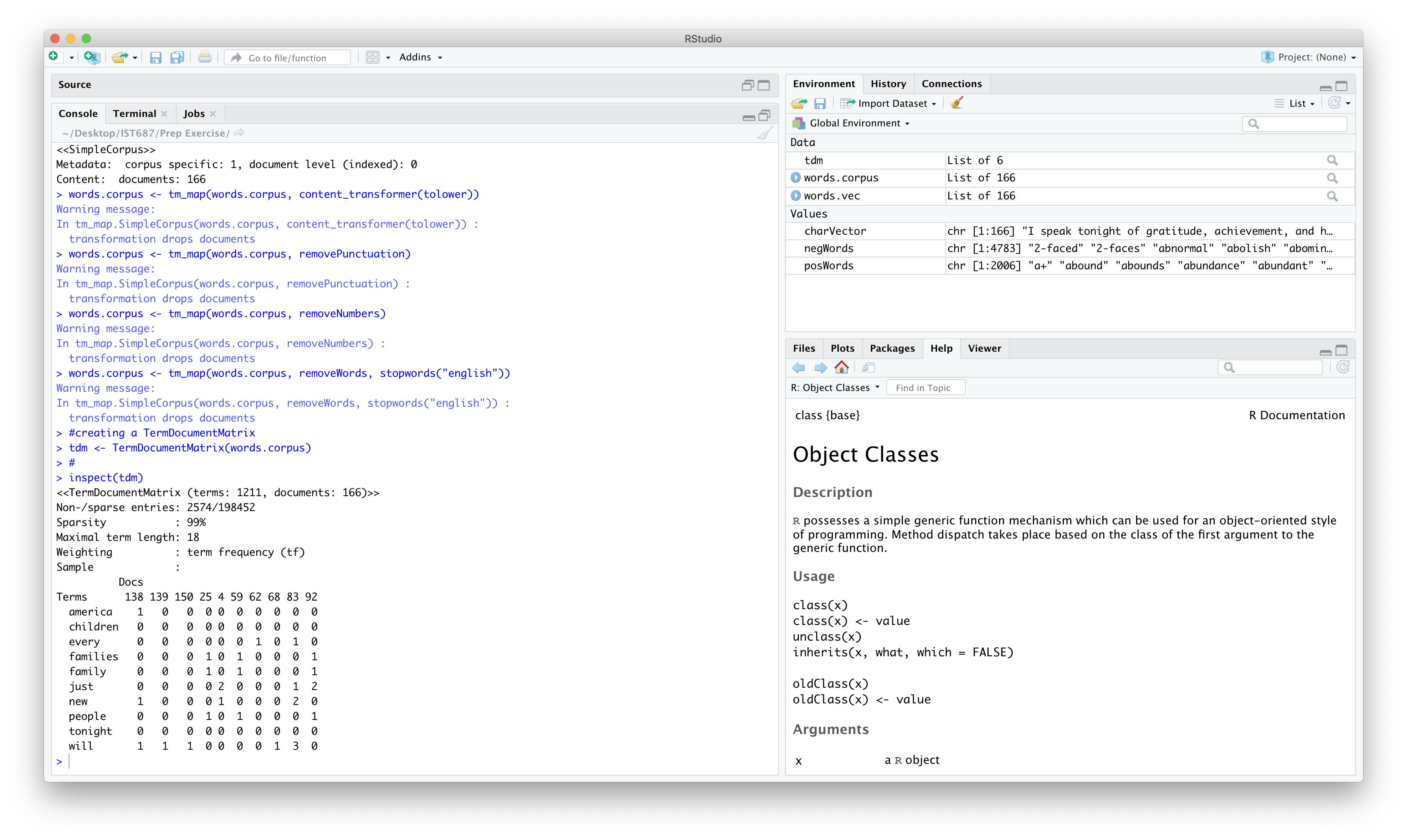
**summary command returns the length, class and mode of the charVector vector. This vector is of class character and has 166 elements.**

1. **Transform charVector into a term document matrix.**
   1. In a few sentences define/explain a term document matrix.

**Term document matrix is a rectangular data structure with terms as the rows and documents as the columns. A term can be a single word or a composite word. The matrix element is zero when the corresponding term is not present in the document. Non-zero elements determine the number of times the corresponding term appeared in the document.**

* 1. Create a word corpus (called words.corpus). Then, make sure everyting is lower case, remove punctuation, remove numbers and then, finally, remove English stopwords. If you need help, review pages 180-181 in the text book (**Note:** Ignore any warning messages that come from *tm\_map()*)
  2. Create a TermDoumentMatrix variable called ‘tdm’, from the words.corpus variable.

1. **Understanding the term document matrix.**
   1. Using *inspect(tdm*) function create a summary of the term document matrix along with a sample of some of the terms and documents.
   2. In a few sentences, explain the output from the previous step. Place a screenshot of the output from *inspect(tdm)* below.

****

**The term document matrix contains 1211 rows(terms) and 166 columns(documents). There are 2574 non-zero entries and the rest (195898) are zeros. Here the sparsity is 99%, which is the proportion of zero elements in the matrix. Maximal term length is the number of characters in the longest term in the matrix, here it is 18. Weighting indicates what type of values is stored in the matrix, here it stores the frequency of terms in the document.**

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

<https://stackoverflow.com/questions/48501599/maximal-term-length-in-document-term-matrix?rq=1>

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.*