Lesson 1, part 1

Let's learn some R!

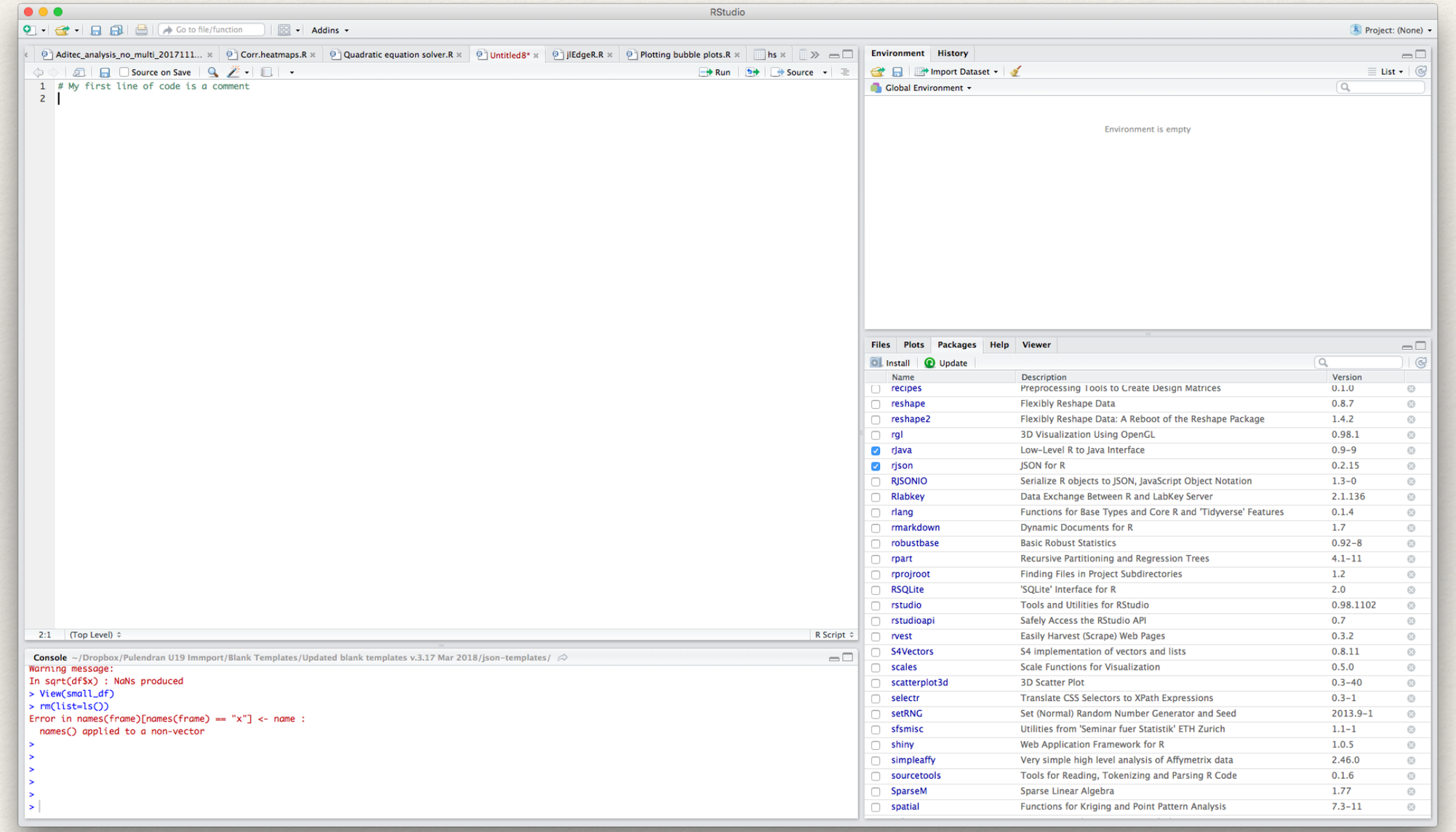
Variables
Functions
Indexing

The first line of code

This is the single most important line that you will ever type…

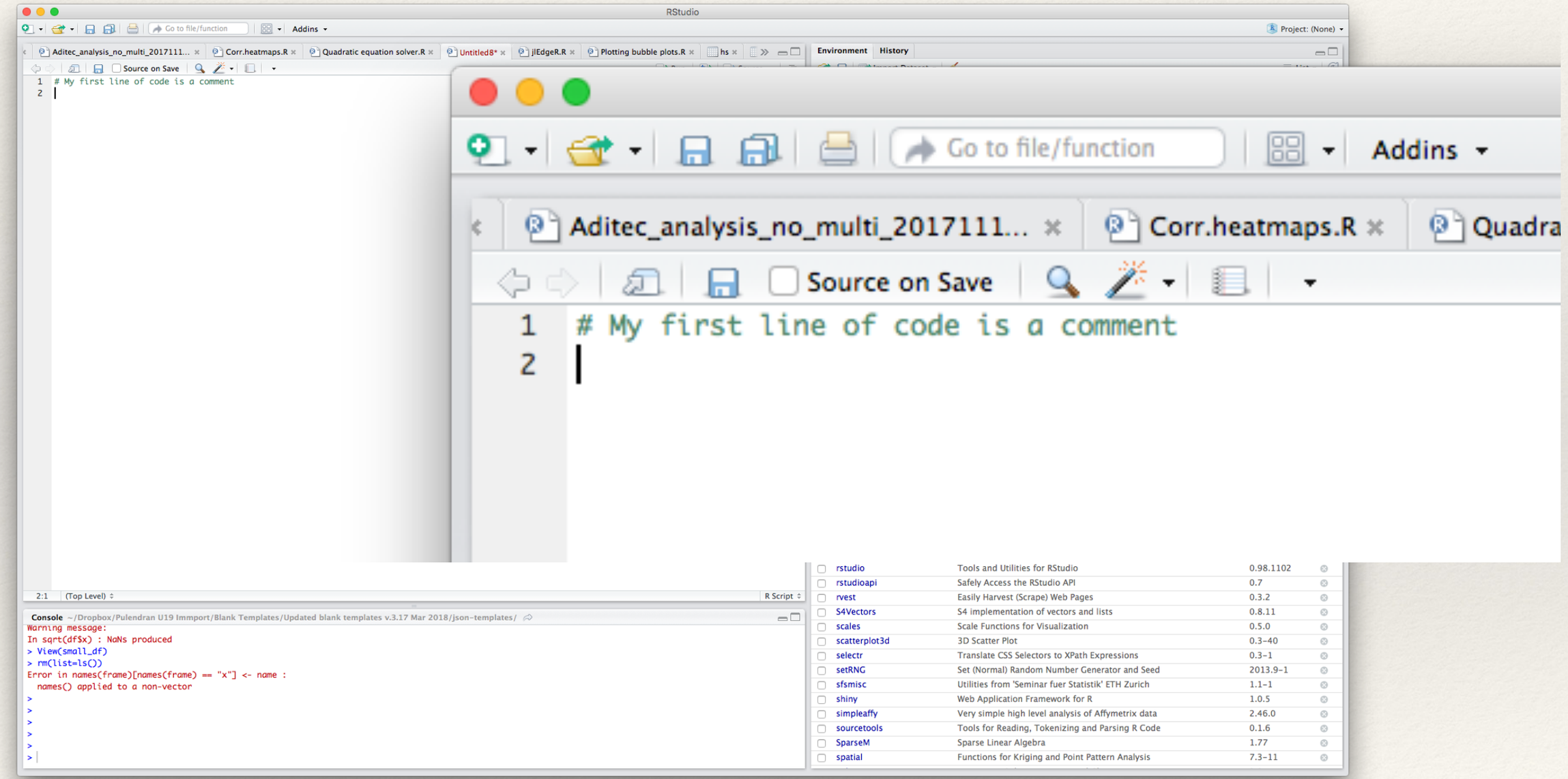
The first line of code

This is the single most important line that you will ever type…

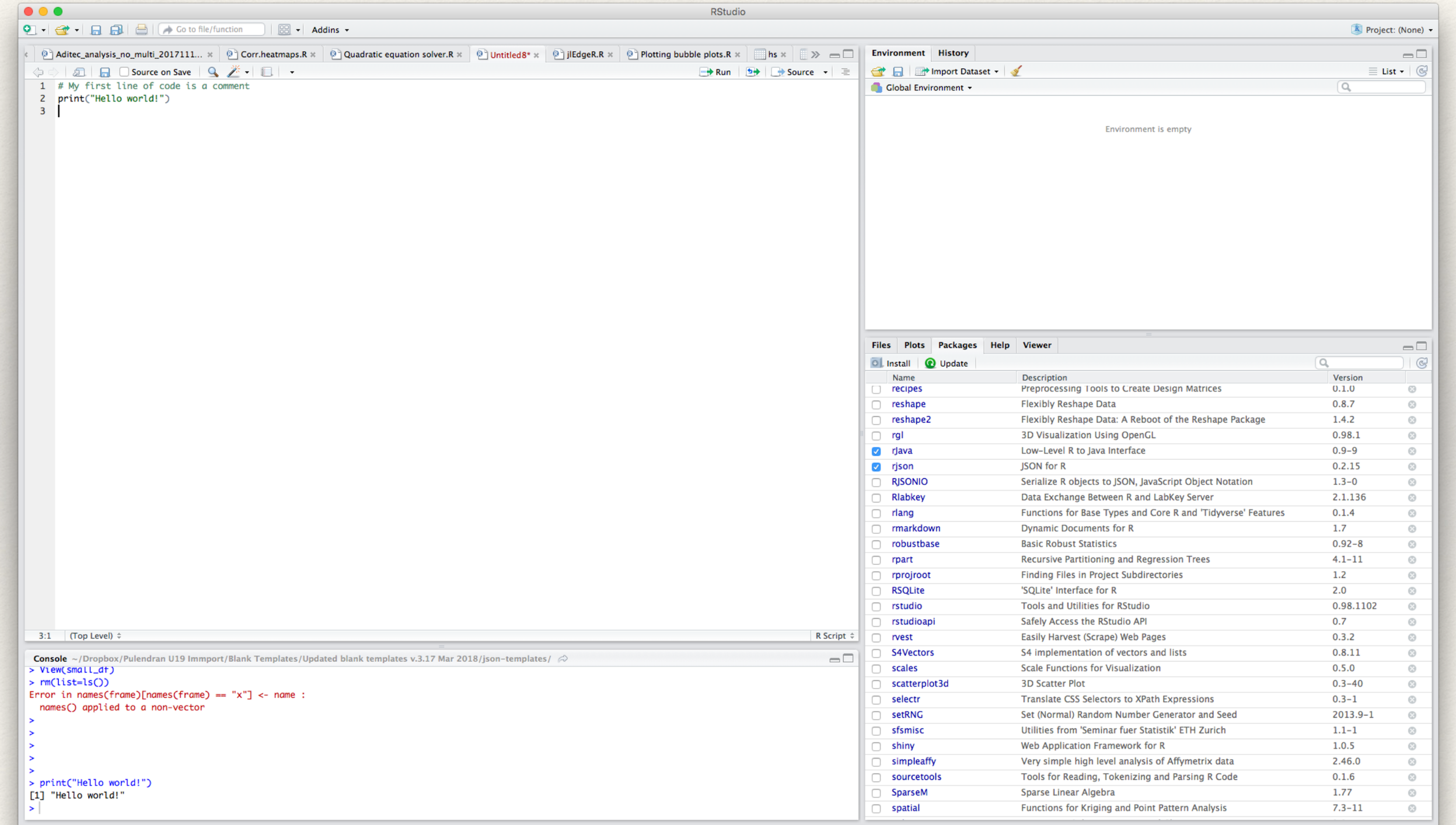


The first line of code

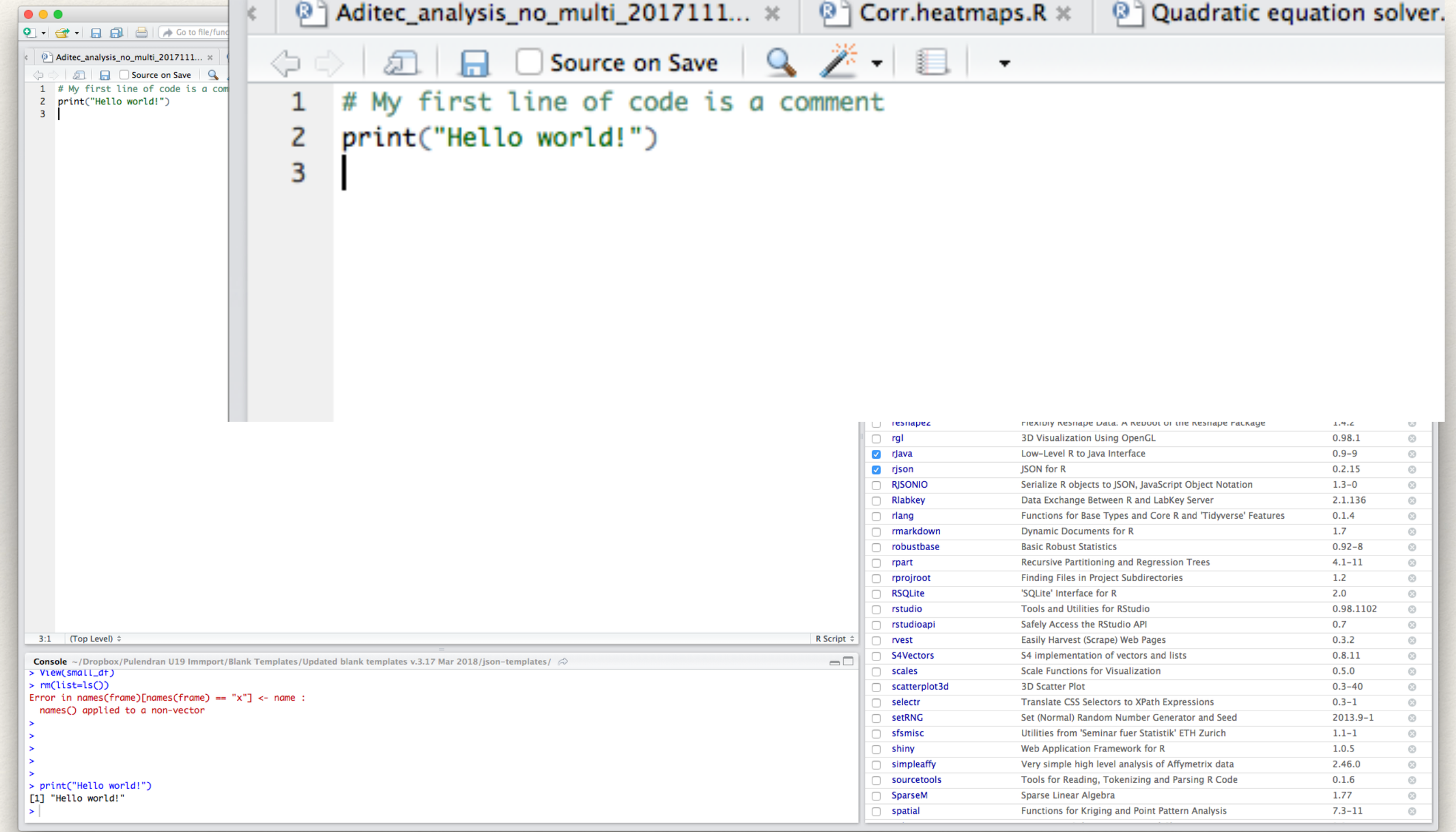
This is the single most important line that you will ever type…



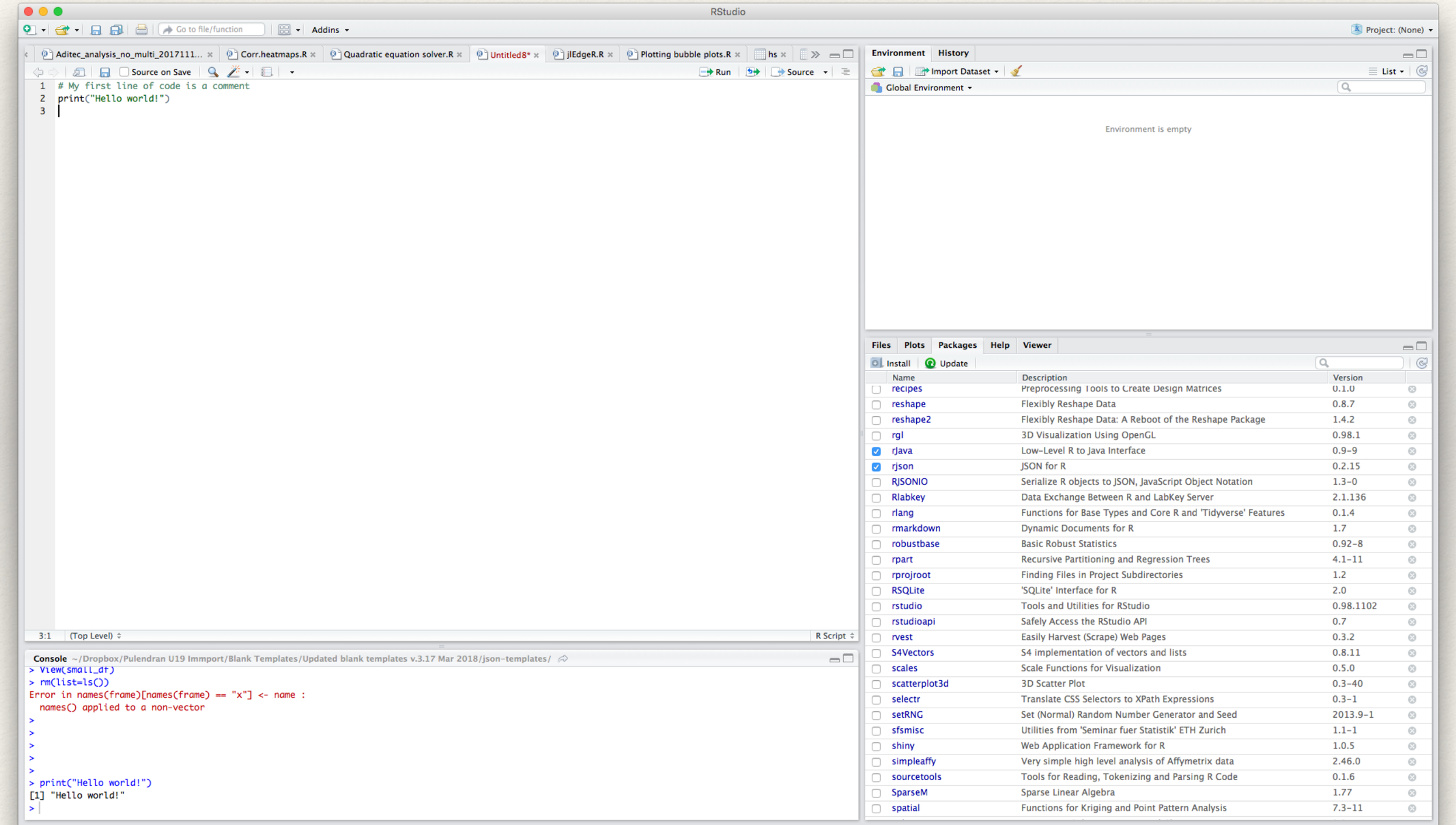
My second line of code



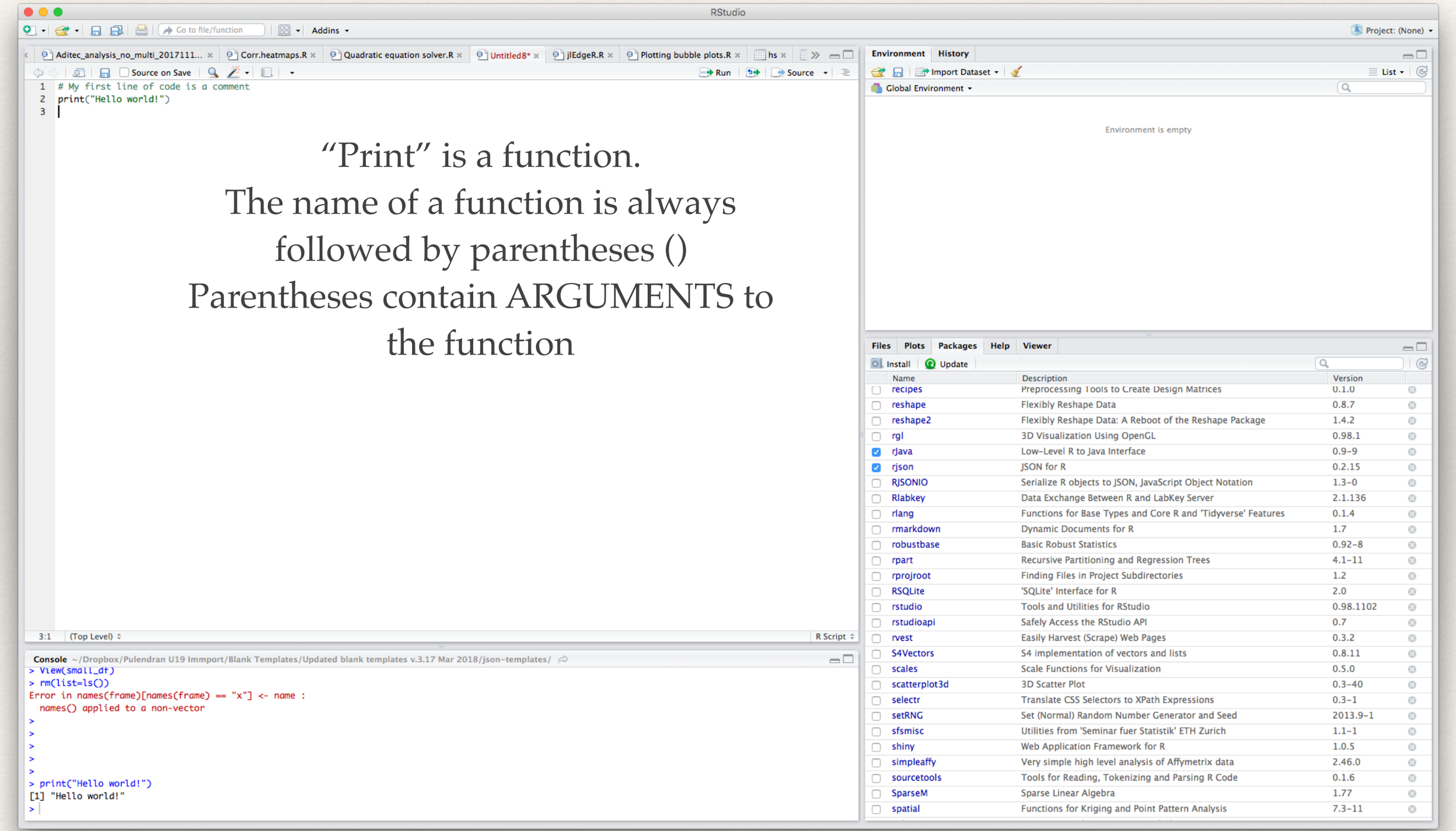
My first line of code is a comment



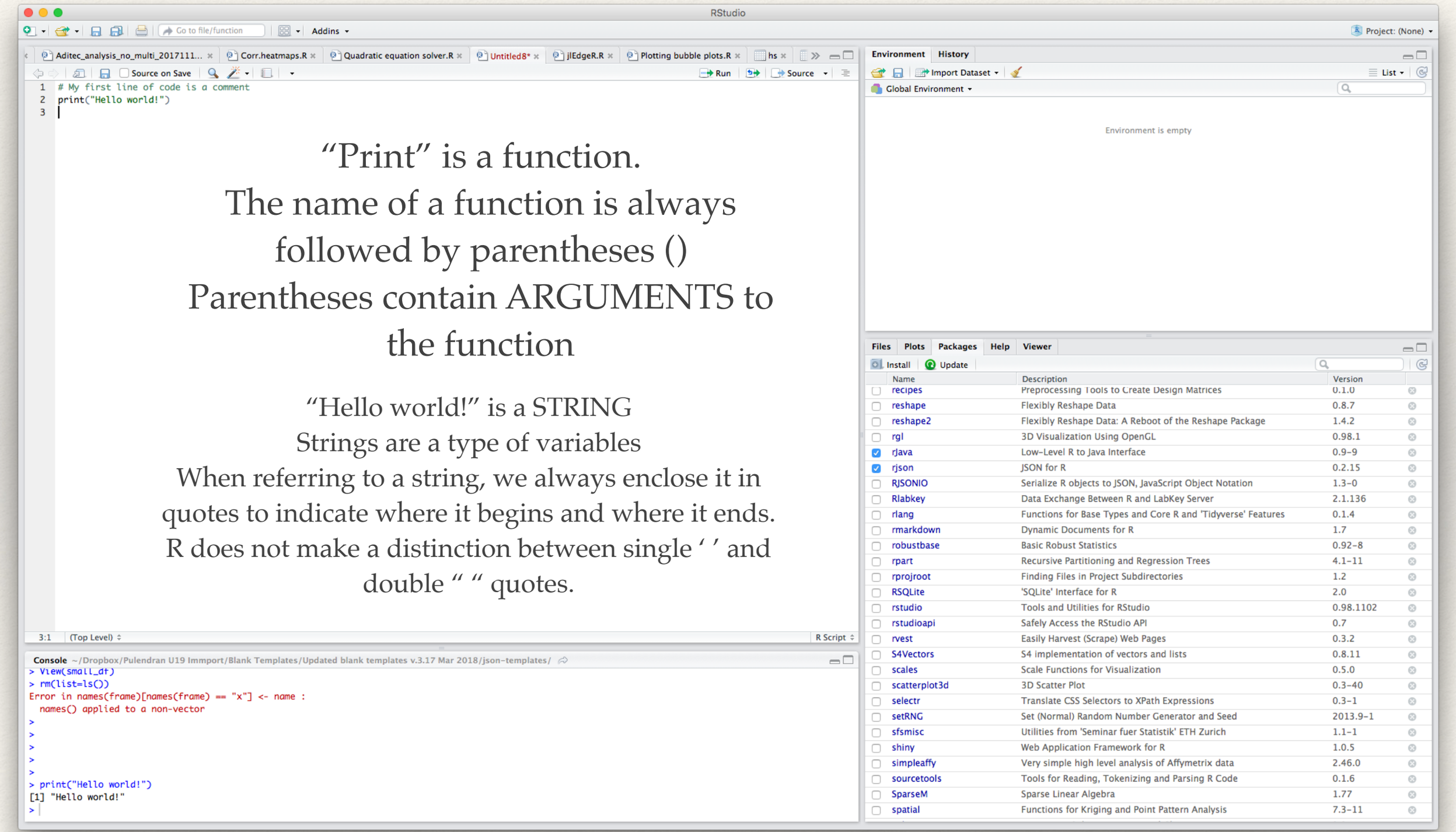
My second line of code



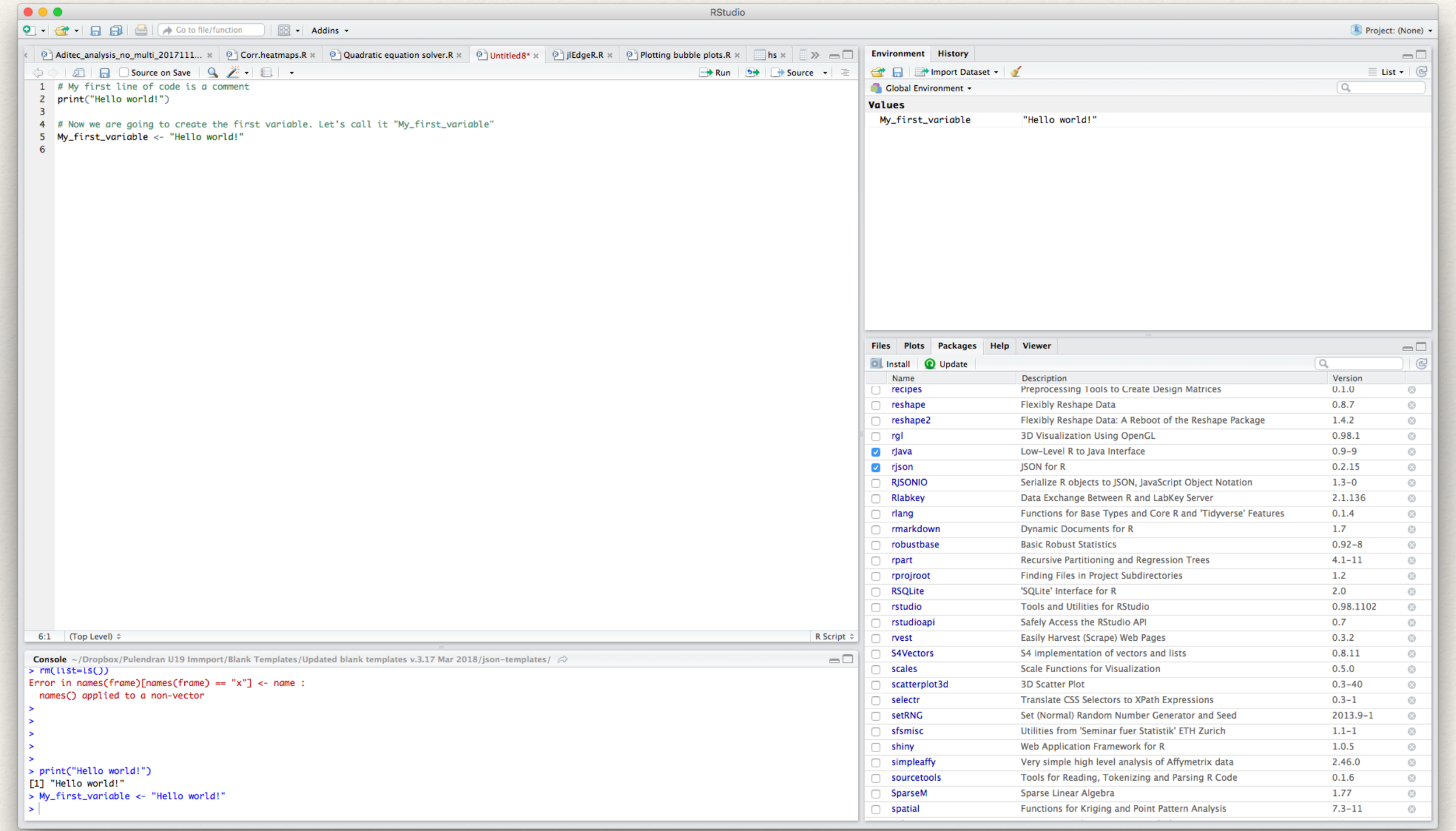
My second line of code



My second line of code



My first variable

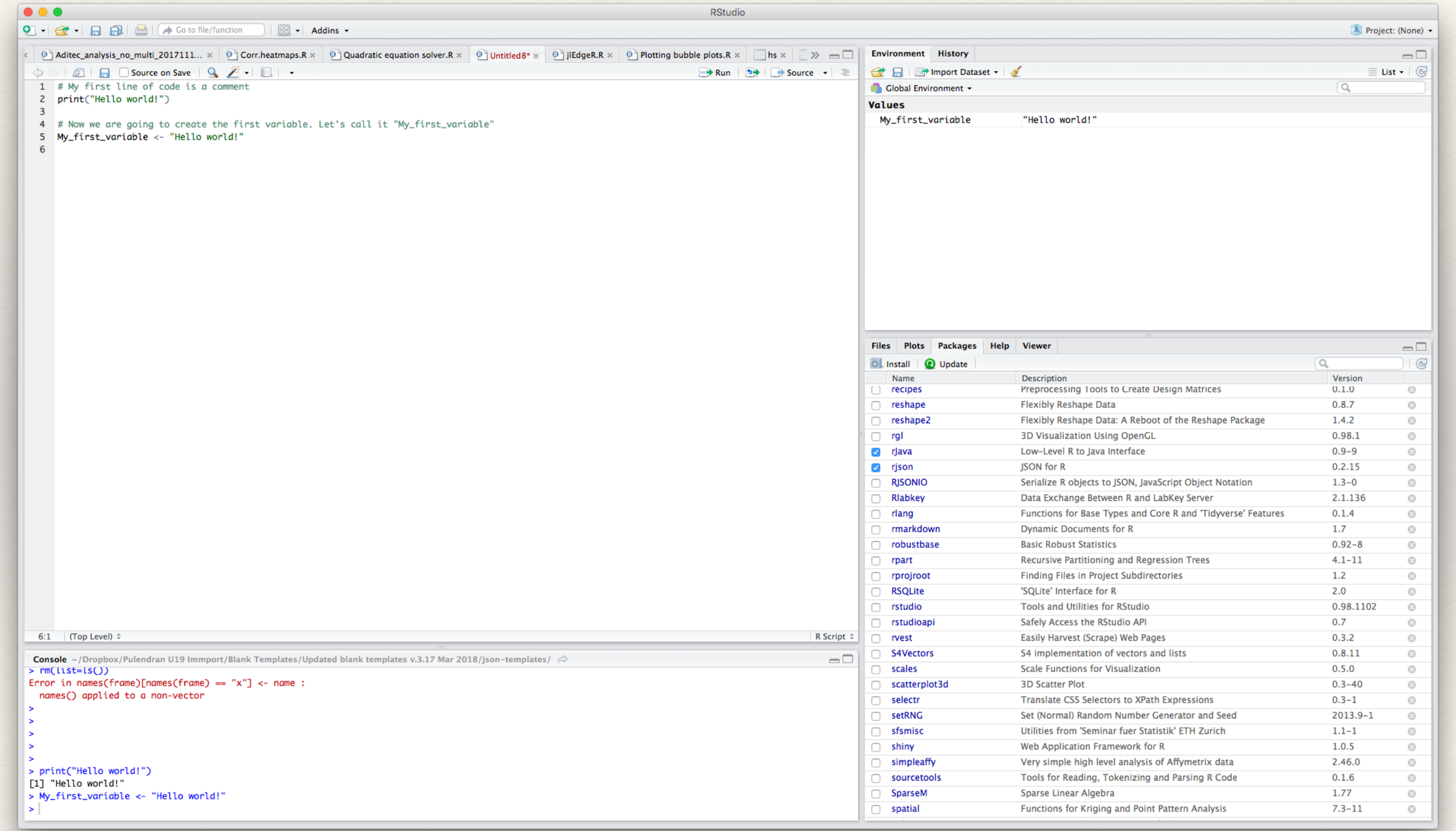


My first variable

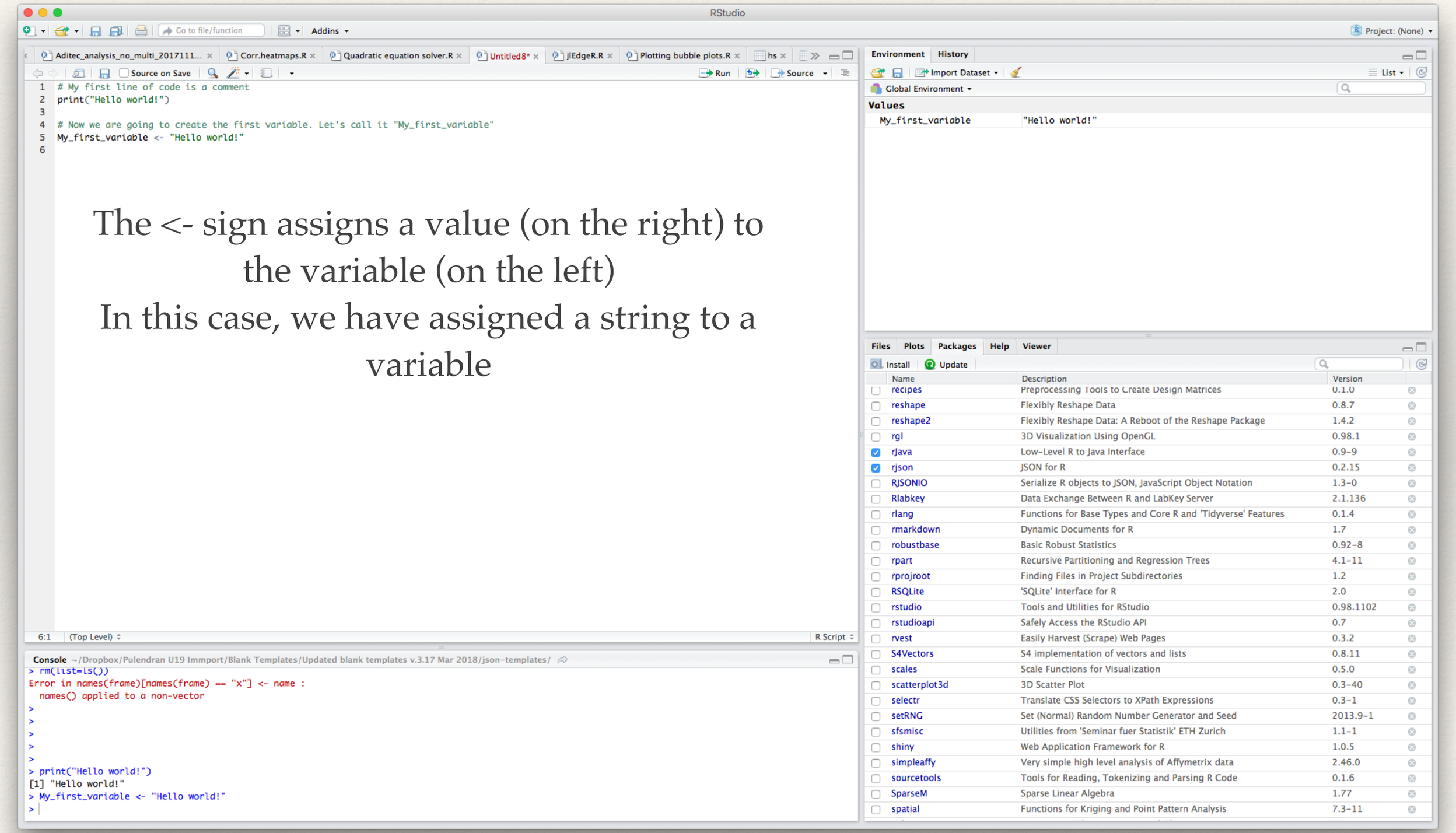
```
# My first line of code is a comment
print("Hello world!")

# Now we are going to create the first variable. Let's call it "My_first_variable"
My_first_variable <- "Hello world!"

```

```
> rm(list=ls())
Error in names(frame)[names(frame) == "x"] <- name :
  names() applied to a non-vector
>
>
>
>
>
> print("Hello world!")
[1] "Hello world!"
> My_first_variable <- "Hello world!"
>
```

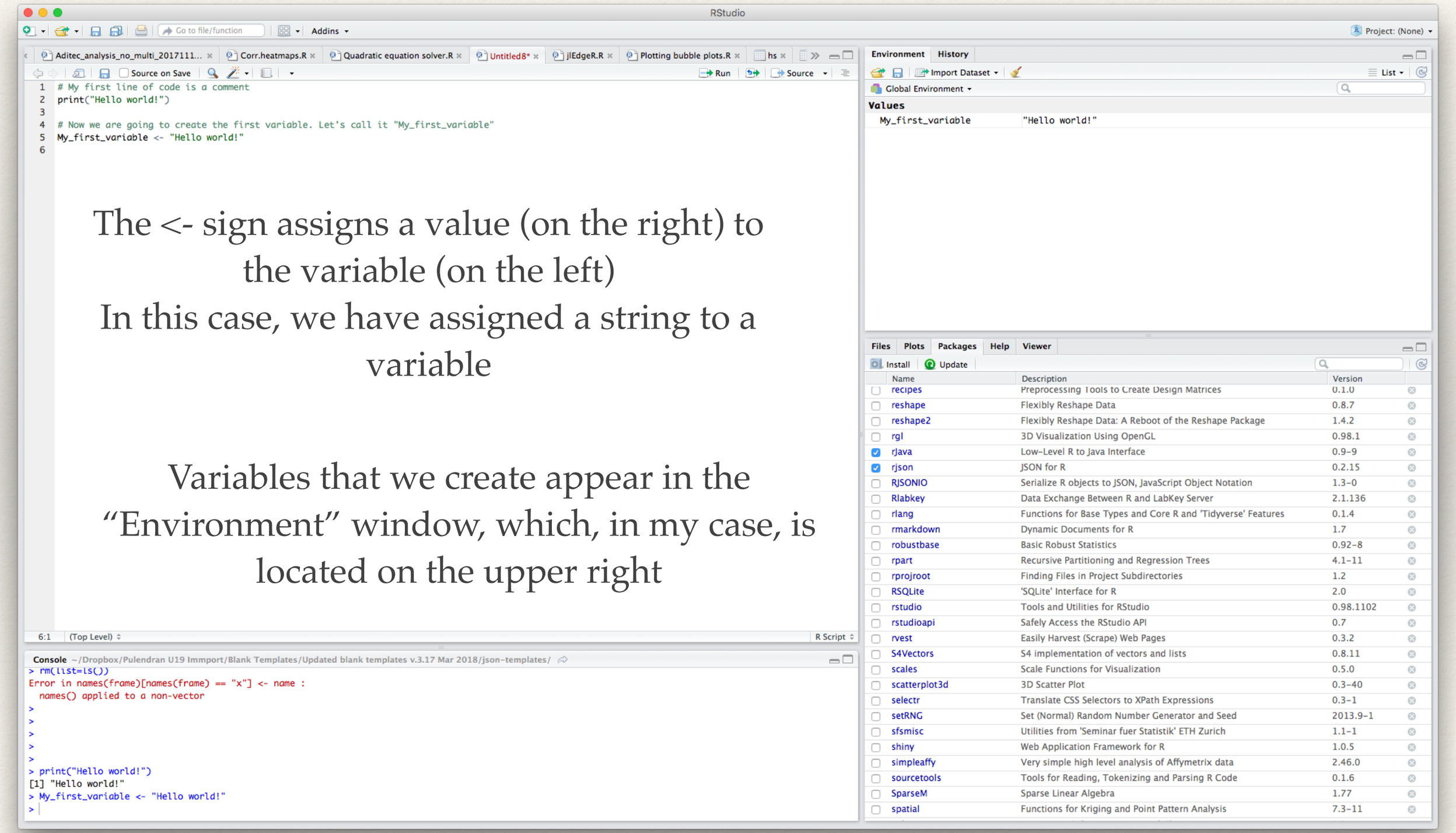
My first variable



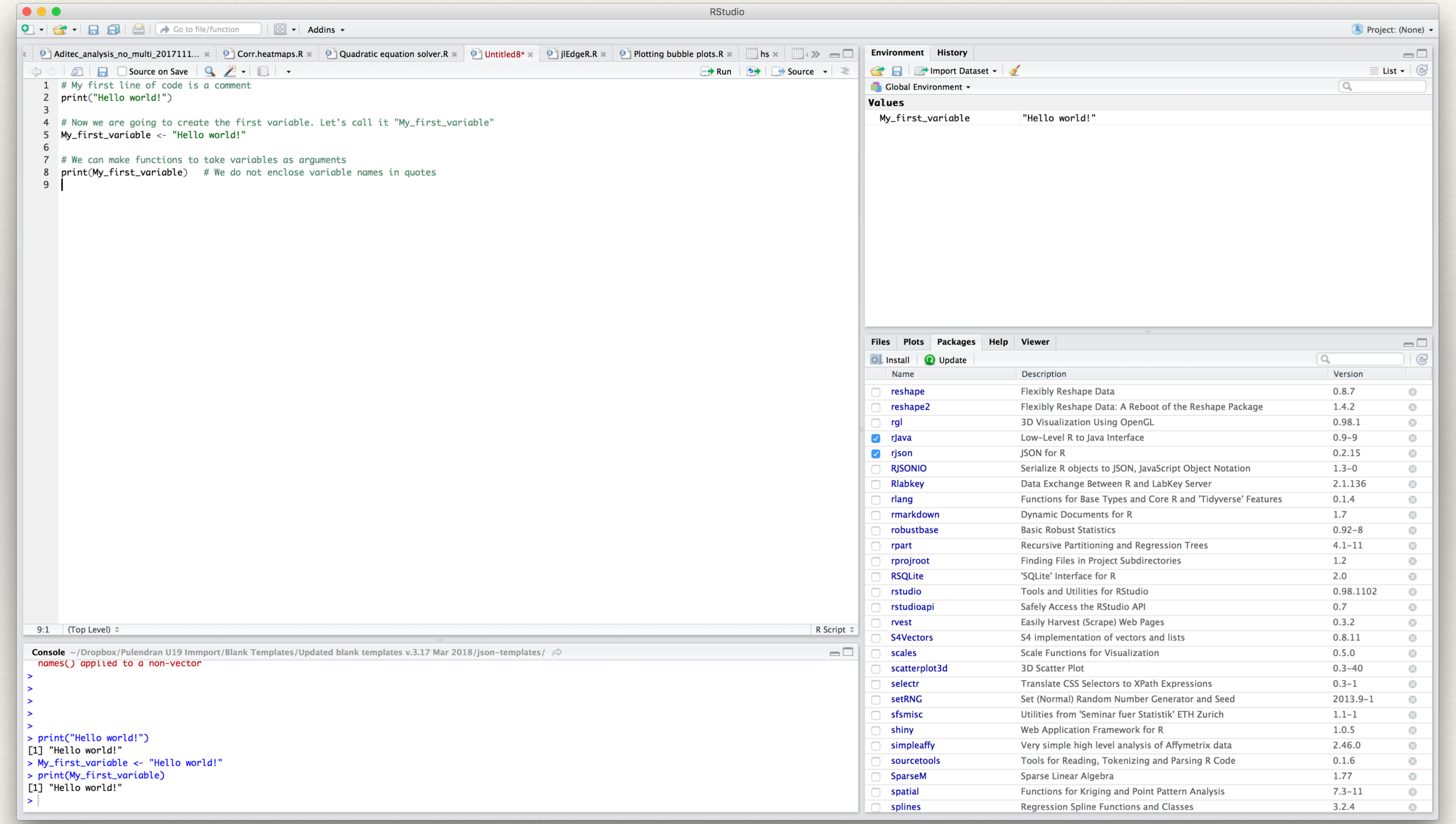
My first variable



My first variable



Referring to variables by name



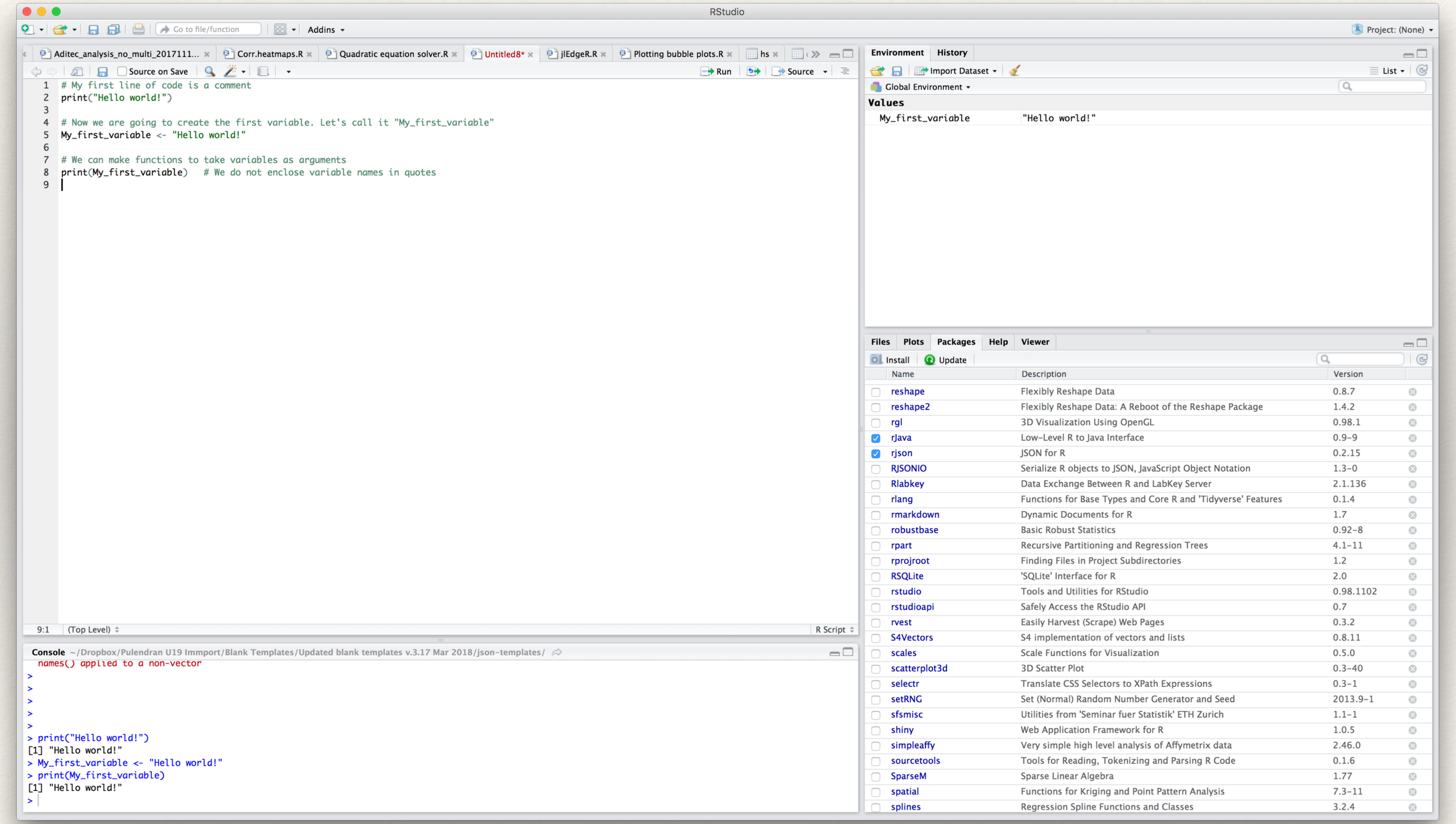
Referring to variables by name

```
# My first line of code is a comment
print("Hello world!")

# Now we are going to create the first variable. Let's call it "My_first_variable"
My_first_variable <- "Hello world!"

# We can make functions to take variables as arguments
print(My_first_variable)   # We do not enclose variable names in quotes
```

```
  names() applied to a non-vector
>
>
>
>
>
> print("Hello world!")
[1] "Hello world!"
> My_first_variable <- "Hello world!"
> print(My_first_variable)
[1] "Hello world!"
>
```

Name	Description	Version
reshape	Flexibly Reshape Data	0.8.7
reshape2	Flexibly Reshape Data: A Reboot of the Reshape Package	1.4.2
rgl	3D Visualization Using OpenGL	0.98.1
rJava	Low-Level R to Java Interface	0.9-9
rjson	JSON for R	0.2.15
RJSONIO	Serialize R objects to JSON, JavaScript Object Notation	1.3-0
Rlabkey	Data Exchange Between R and LabKey Server	2.1.136
rlang	Functions for Base Types and Core R and 'Tidyverse' Features	0.1.4
rmarkdown	Dynamic Documents for R	1.7
robustbase	Basic Robust Statistics	0.92-8
rpart	Recursive Partitioning and Regression Trees	4.1-11
rprojroot	Finding Files in Project Subdirectories	1.2
RSQLite	'SQLite' Interface for R	2.0
rstudio	Tools and Utilities for RStudio	0.98.1102
rstudioapi	Safely Access the RStudio API	0.7
rvest	Easily Harvest (Scrape) Web Pages	0.3.2
S4Vectors	S4 implementation of vectors and lists	0.8.11
scales	Scale Functions for Visualization	0.5.0
scatterplot3d	3D Scatter Plot	0.3-40
selectr	Translate CSS Selectors to XPath Expressions	0.3-1
setRNG	Set (Normal) Random Number Generator and Seed	2013.9-1
sfsmisc	Utilities from 'Seminar fuer Statistik' ETH Zurich	1.1-1
shiny	Web Application Framework for R	1.0.5
simpleaffy	Very simple high level analysis of Affymetrix data	2.46.0
sourcetools	Tools for Reading, Tokenizing and Parsing R Code	0.1.6
SparseM	Sparse Linear Algebra	1.77
spatial	Functions for Kriging and Point Pattern Analysis	7.3-11
splines	Regression Spline Functions and Classes	3.2.4

Referring to variables by name



Referring to variables by name

RStudio

```
1  # My first line of code is a comment
2  print("Hello world!")
3
4  # Now we are going to create the first variable. Let's call it "My_first_variable"
5  My_first_variable <- "Hello world!"
6
7  # We can make functions to take variables as arguments
8  print(My_first_variable)   # We do not enclose variable names in quotes
9
```

Here we have passed a variable as an argument to the "print" function.

The result is the printing of the contents of the variable.

```
Console ~/Dropbox/Pulendran U19 Immport/Blank Templates/Updated blank templates v.3.17 Mar 2018/json-templates/
  names() applied to a non-vector
>
>
>
>
>
> print("Hello world!")
[1] "Hello world!"
> My_first_variable <- "Hello world!"
> print(My_first_variable)
[1] "Hello world!"
>
```

Environment | History

Global Environment

Values

My_first_variable	"Hello world!"

Files | Plots | Packages | Help | Viewer

Install | Update

Name	Description	Version
reshape	Flexibly Reshape Data	0.8.7
reshape2	Flexibly Reshape Data: A Reboot of the Reshape Package	1.4.2
rgl	3D Visualization Using OpenGL	0.98.1
rJava	Low-Level R to Java Interface	0.9-9
rjson	JSON for R	0.2.15
RJSONIO	Serialize R objects to JSON, JavaScript Object Notation	1.3-0
Rlabkey	Data Exchange Between R and LabKey Server	2.1.136
rlang	Functions for Base Types and Core R and 'Tidyverse' Features	0.1.4
rmarkdown	Dynamic Documents for R	1.7
robustbase	Basic Robust Statistics	0.92-8
rpart	Recursive Partitioning and Regression Trees	4.1-11
rprojroot	Finding Files in Project Subdirectories	1.2
RSQLite	'SQLite' Interface for R	2.0
rstudio	Tools and Utilities for RStudio	0.98.1102
rstudioapi	Safely Access the RStudio API	0.7
rvest	Easily Harvest (Scrape) Web Pages	0.3.2
S4Vectors	S4 implementation of vectors and lists	0.8.11
scales	Scale Functions for Visualization	0.5.0
scatterplot3d	3D Scatter Plot	0.3-40
selectr	Translate CSS Selectors to XPath Expressions	0.3-1
setRNG	Set (Normal) Random Number Generator and Seed	2013.9-1
sfsmisc	Utilities from 'Seminar fuer Statistik' ETH Zurich	1.1-1
shiny	Web Application Framework for R	1.0.5
simpleaffy	Very simple high level analysis of Affymetrix data	2.46.0
sourcetools	Tools for Reading, Tokenizing and Parsing R Code	0.1.6
SparseM	Sparse Linear Algebra	1.77
spatial	Functions for Kriging and Point Pattern Analysis	7.3-11
splines	Regression Spline Functions and Classes	3.2.4

Classes of variables

There are many different classes of variables: character arrays, numeric arrays, matrices, data frames, factors.

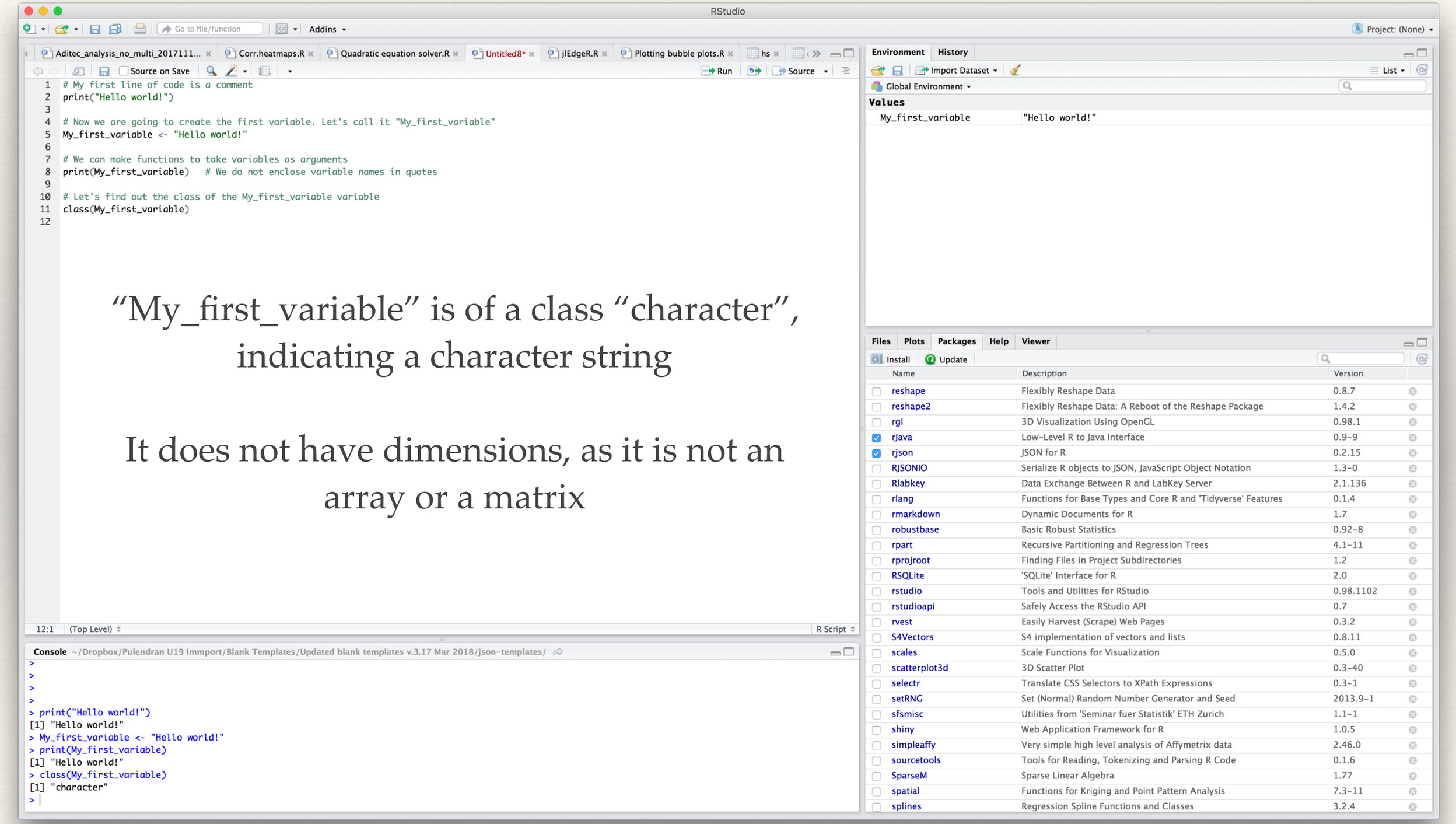
Class is often times also referred as data type.

There are the following data types in base R:

- Matrices
- Vectors
- Arrays
- Data frames
- Lists
- Factors

To find out the data type, we can use the **class()** function

Classes of variables



"My_first_variable" is of a class "character", indicating a character string

It does not have dimensions, as it is not an array or a matrix

Classes of variables

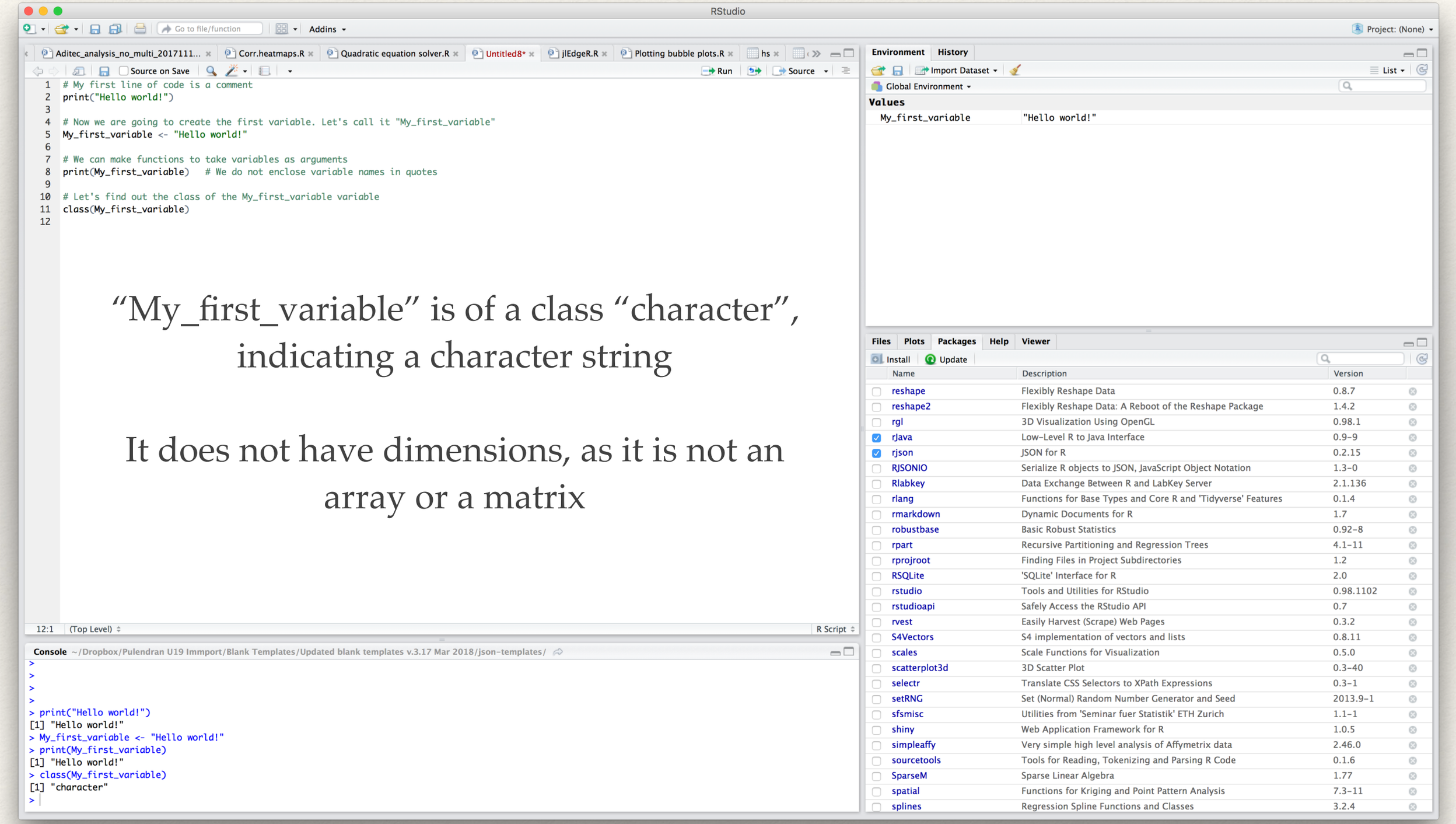
```
# My first line of code is a comment
print("Hello world!")

# Now we are going to create the first variable. Let's call it "My_first_variable"
My_first_variable <- "Hello world!"

# We can make functions to take variables as arguments
print(My_first_variable)   # We do not enclose variable names in quotes

# Let's find out the class of the My_first_variable variable
class(My_first_variable)
```

```
> print("Hello world!")
[1] "Hello world!"
> My_first_variable <- "Hello world!"
> print(My_first_variable)
[1] "Hello world!"
> class(My_first_variable)
[1] "character"
>
```

"My_first_variable" is of a class "character", indicating a character string

It does not have dimensions, as it is not an array or a matrix

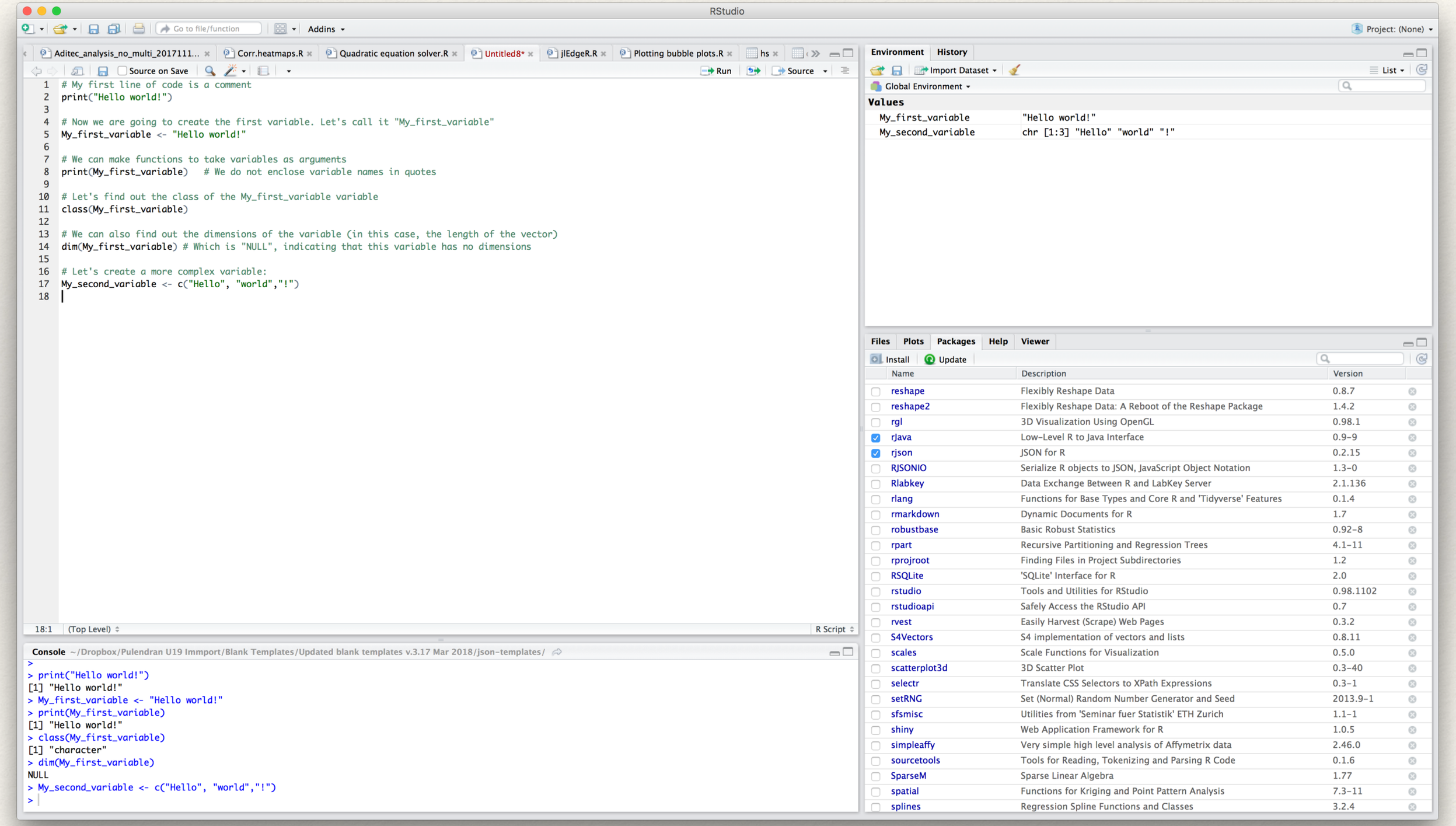
Classes of variables



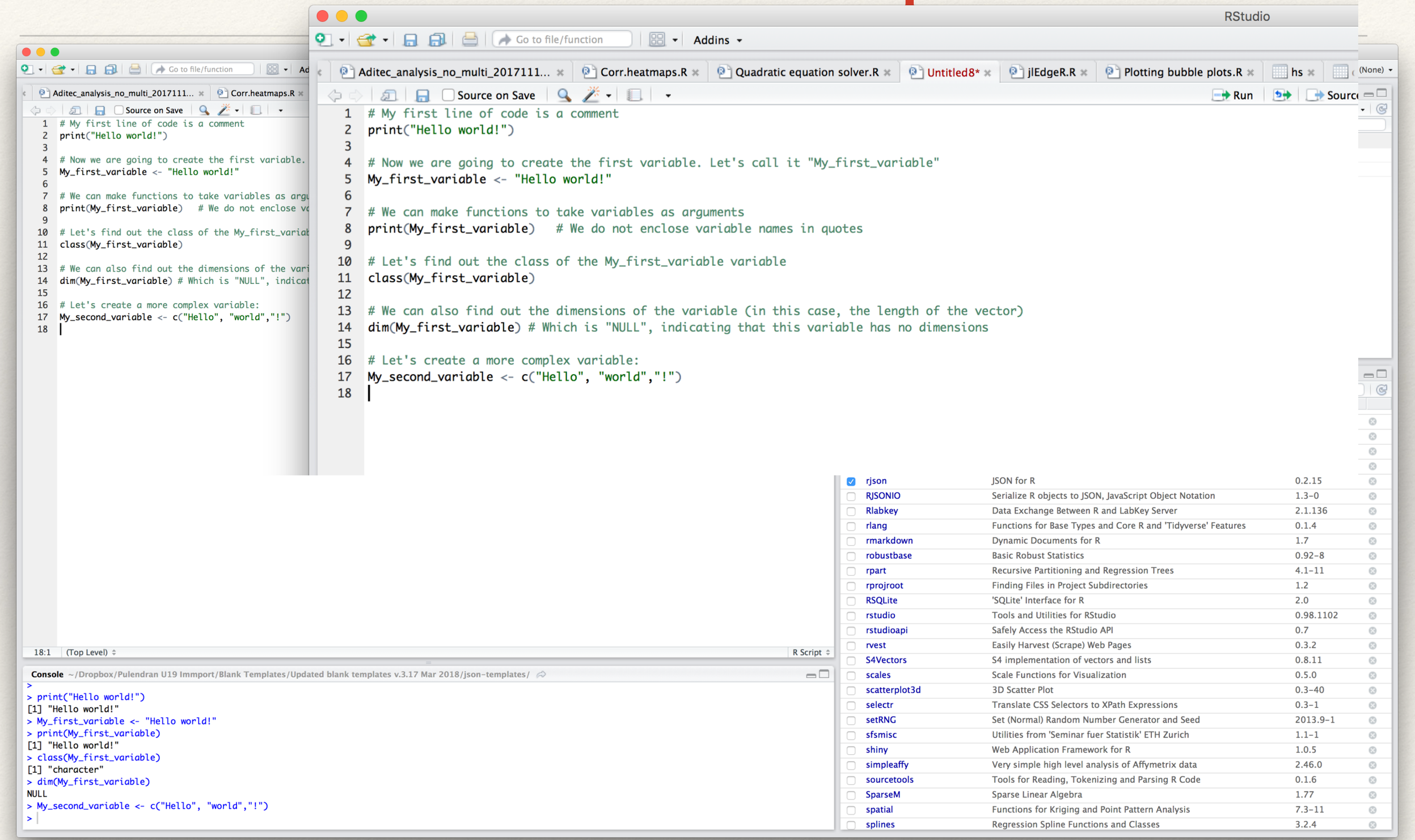
"My_first_variable" is of a class "character", indicating a character string

It does not have dimensions, as it is not an array or a matrix

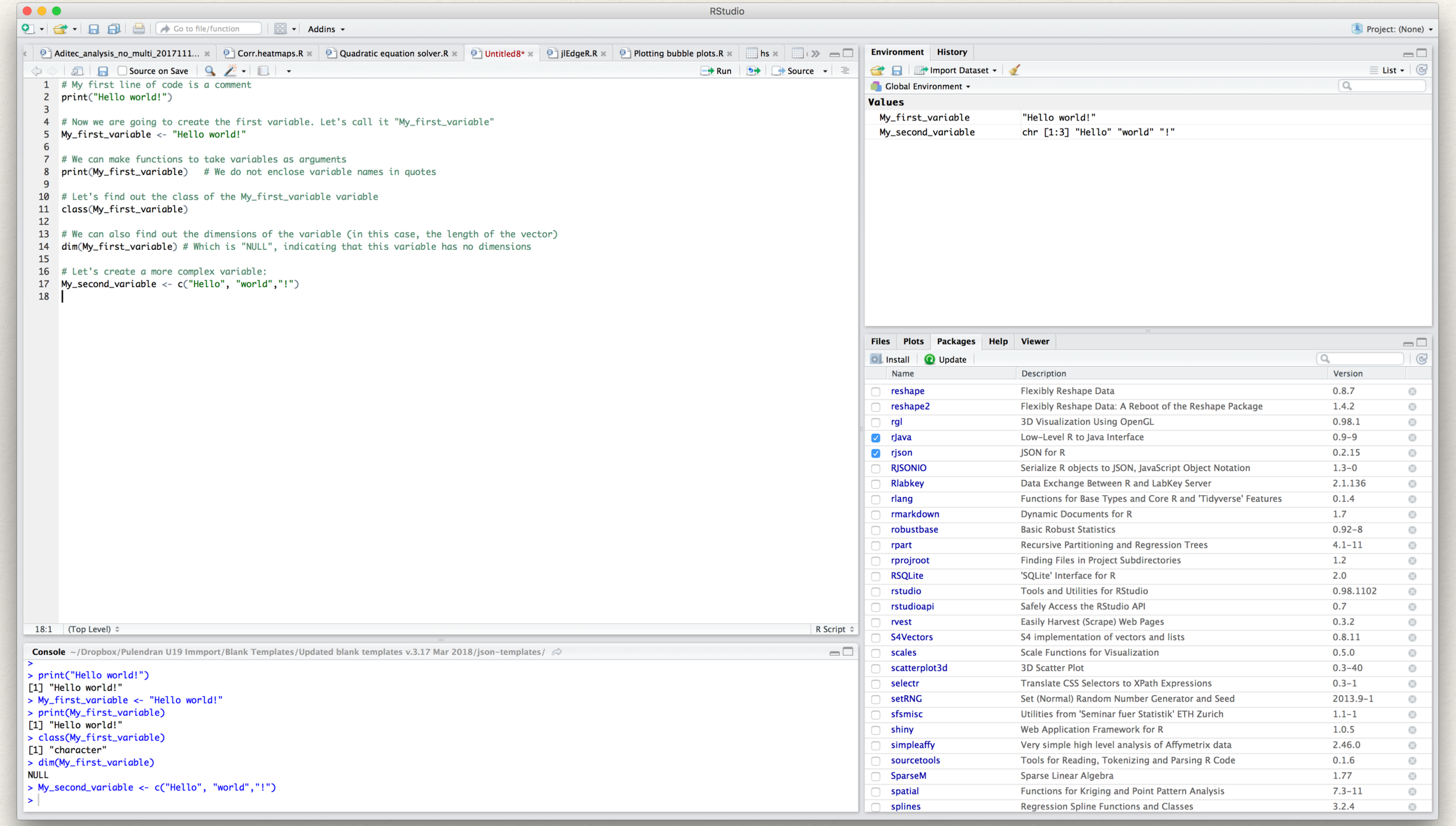
Let's create a more complex variable



Let's create a more complex variable



Let's create a more complex variable



Let's create a more complex variable

RStudio

```
1  # My first line of code is a comment
2  print("Hello world!")
3
4  # Now we are going to create the first variable. Let's call it "My_first_variable"
5  My_first_variable <- "Hello world!"
6
7  # We can make functions to take variables as arguments
8  print(My_first_variable)   # We do not enclose variable names in quotes
9
10 # Let's find out the class of the My_first_variable variable
11 class(My_first_variable)
12
13 # We can also find out the dimensions of the variable (in this case, the length of the vector)
14 dim(My_first_variable) # Which is "NULL", indicating that this variable has no dimensions
15
16 # Let's create a more complex variable:
17 My_second_variable <- c("Hello", "world","!")
18
```

Here we have created a character VECTOR, consisting of three elements

The accessory function c() is used co concatenate its arguments into a single array

```
> print("Hello world!")
[1] "Hello world!"
> My_first_variable <- "Hello world!"
> print(My_first_variable)
[1] "Hello world!"
> class(My_first_variable)
[1] "character"
> dim(My_first_variable)
NULL
> My_second_variable <- c("Hello", "world","!")
>
```

Values	
My_first_variable	"Hello world!"
My_second_variable	chr [1:3] "Hello" "world" "!"

Name	Description	Version
reshape	Flexibly Reshape Data	0.8.7
reshape2	Flexibly Reshape Data: A Reboot of the Reshape Package	1.4.2
rgl	3D Visualization Using OpenGL	0.98.1
rJava	Low-Level R to Java Interface	0.9-9
rjson	JSON for R	0.2.15
RJSONIO	Serialize R objects to JSON, JavaScript Object Notation	1.3-0
Rlabkey	Data Exchange Between R and LabKey Server	2.1.136
rlang	Functions for Base Types and Core R and 'Tidyverse' Features	0.1.4
rmarkdown	Dynamic Documents for R	1.7
robustbase	Basic Robust Statistics	0.92-8
rpart	Recursive Partitioning and Regression Trees	4.1-11
rprojroot	Finding Files in Project Subdirectories	1.2
RSQLite	'SQLite' Interface for R	2.0
rstudio	Tools and Utilities for RStudio	0.98.1102
rstudioapi	Safely Access the RStudio API	0.7
rvest	Easily Harvest (Scrape) Web Pages	0.3.2
S4Vectors	S4 implementation of vectors and lists	0.8.11
scales	Scale Functions for Visualization	0.5.0
scatterplot3d	3D Scatter Plot	0.3-40
selectr	Translate CSS Selectors to XPath Expressions	0.3-1
setRNG	Set (Normal) Random Number Generator and Seed	2013.9-1
sfsmisc	Utilities from 'Seminar fuer Statistik' ETH Zurich	1.1-1
shiny	Web Application Framework for R	1.0.5
simpleaffy	Very simple high level analysis of Affymetrix data	2.46.0
sourcetools	Tools for Reading, Tokenizing and Parsing R Code	0.1.6
SparseM	Sparse Linear Algebra	1.77
spatial	Functions for Kriging and Point Pattern Analysis	7.3-11
splines	Regression Spline Functions and Classes	3.2.4

Dimensions and length of a vector

Vectors do not have dimensions:
dim(My_second_variable) produces NULL output

However, vectors have length:
length(My_second_variable) produces output:
3

Dimensions and length of a vector

RStudio

Aditec_analysis_no_multi_2017111... | Corr.heatmaps.R | Quadratic equation solver.R | Untitled8* | jlEdgeR.R | Plotting bubble plots.R | hs

```
1  # My first line of code is a comment
2  print("Hello world!")
3
4  # Now we are going to create the first variable. Let's call it "My_first_variable"
5  My_first_variable <- "Hello world!"
6
7  # We can make functions to take variables as arguments
8  print(My_first_variable)   # We do not enclose variable names in quotes
9
10 # Let's find out the class of the My_first_variable variable
11 class(My_first_variable)
12
13 # We can also find out the dimensions of the variable (in this case, the length of the vector)
14 dim(My_first_variable) # Which is "NULL", indicating that this variable has no dimensions
15
16 # Let's create a more complex variable:
17 My_second_variable <- c("Hello", "world","!")
18
19 # We can find out the dimensions of the new variable: However it wil not be 1 by 3, as vetors do not have dimensions
20 dim(My_second_variable)
21
22 # However, vectors have length, which can be found out by using the length() function
23 length(My_second_variable)
24
```

24:1 (Top Level) R Script

Console ~/Dropbox/Pulendran U19 Immport/Blank Templates/Updated blank templates v.3.17 Mar 2018/json-templates/

```
> print(My_first_variable)
[1] "Hello world!"
> class(My_first_variable)
[1] "character"
> dim(My_first_variable)
NULL
> My_second_variable <- c("Hello", "world","!")
> dim(My_second_variable)
NULL
> length(My_second_variable)
[1] 3
>
```

Environment | History

Import Dataset | List

Global Environment

Values

My_first_variable	"Hello world!"
My_second_variable	chr [1:3] "Hello" "world" "!"

Files | Plots | Packages | Help | Viewer

Install | Update

Name	Description	Version
reshape	Flexibly Reshape Data	0.8.7
reshape2	Flexibly Reshape Data: A Reboot of the Reshape Package	1.4.2
rgl	3D Visualization Using OpenGL	0.98.1
rJava	Low-Level R to Java Interface	0.9-9
rjson	JSON for R	0.2.15
RJSONIO	Serialize R objects to JSON, JavaScript Object Notation	1.3-0
Rlabkey	Data Exchange Between R and LabKey Server	2.1.136
rlang	Functions for Base Types and Core R and 'Tidyverse' Features	0.1.4
rmarkdown	Dynamic Documents for R	1.7
robustbase	Basic Robust Statistics	0.92-8
rpart	Recursive Partitioning and Regression Trees	4.1-11
rprojroot	Finding Files in Project Subdirectories	1.2
RSQLite	'SQLite' Interface for R	2.0
rstudio	Tools and Utilities for RStudio	0.98.1102
rstudioapi	Safely Access the RStudio API	0.7
rvest	Easily Harvest (Scrape) Web Pages	0.3.2
S4Vectors	S4 implementation of vectors and lists	0.8.11
scales	Scale Functions for Visualization	0.5.0
scatterplot3d	3D Scatter Plot	0.3-40
selectr	Translate CSS Selectors to XPath Expressions	0.3-1
setRNG	Set (Normal) Random Number Generator and Seed	2013.9-1
sfsmisc	Utilities from 'Seminar fuer Statistik' ETH Zurich	1.1-1
shiny	Web Application Framework for R	1.0.5
simpleaffy	Very simple high level analysis of Affymetrix data	2.46.0
sourcetools	Tools for Reading, Tokenizing and Parsing R Code	0.1.6
SparseM	Sparse Linear Algebra	1.77
spatial	Functions for Kriging and Point Pattern Analysis	7.3-11
splines	Regression Spline Functions and Classes	3.2.4

Dimensions and length of a vector

Vectors do not have dimensions:
dim(My_second_variable) produces NULL output

However, vectors have length:
length(My_second_variable) produces output:
3

Referring to the elements of a vector

RStudio

Aditec_analysis_no_multi_2017111... | Corr.heatmaps.R | Quadratic equation solver.R | Untitled8* | jlEdgeR.R | Plotting bubble plots.R | hs

```
1  # My first line of code is a comment
2  print("Hello world!")
3
4  # Now we are going to create the first variable. Let's call it "My_first_variable"
5  My_first_variable <- "Hello world!"
6
7  # We can make functions to take variables as arguments
8  print(My_first_variable)   # We do not enclose variable names in quotes
9
10 # Let's find out the class of the My_first_variable variable
11 class(My_first_variable)
12
13 # We can also find out the dimensions of the variable (in this case, the length of the vector)
14 dim(My_first_variable) # Which is "NULL", indicating that this variable has no dimensions
15
16 # Let's create a more complex variable:
17 My_second_variable <- c("Hello", "world","!")
18
19 # We can find out the dimensions of the new variable: However it wil not be 1 by 3, as vetors do not have dimensions
20 dim(My_second_variable)
21
22 # However, vectors have length, which can be found out by using the length() function
23 length(My_second_variable)
24
25 # Now let's refer to the individual elements of the character vector:
26 My_second_variable[1]     # This will display the contents of the first element in the command line window
27 one <- My_second_variable[1]    # This will assign the contents of the first element of the My_second_variable to the new variable, "one"
28
29 # And likewise we can do for the second and the third elements:
30 My_first_variable[2]
31 two <- My_second_variable[2]
32 My_second_variable[3]
33 three <- My_second_variable[3]
```

33:31 (Top Level) R Script

Console ~/Dropbox/Pulendran U19 Immport/Blank Templates/Updated blank templates v.3.17 Mar 2018/json-templates/

```
> My_second_variable[1]     # This will display the contents of the first element in the command line window
[1] "Hello"
> one <- My_second_variable[1]    # This will assign the contents of the first element of the My_second_variable to the new variable, "one"
>
> # And likewise we can do for the second and the third elements:
> My_first_variable[2]
[1] NA
> two <- My_second_variable[2]
> My_second_variable[3]
[1] "!"
> three <- My_second_variable[3]
>
```

Environment | History

Global Environment

Values

My_first_variable	"Hello world!"
My_second_variable	chr [1:3] "Hello" "world" "!"
one	"Hello"
three	"!"
two	"world"

Files | Plots | Packages | Help | Viewer

Install | Update

	Name	Description	Version
☐	reshape	Flexibly Reshape Data	0.8.7
☐	reshape2	Flexibly Reshape Data: A Reboot of the Reshape Package	1.4.2
☐	rgl	3D Visualization Using OpenGL	0.98.1
☑	rJava	Low-Level R to Java Interface	0.9-9
☑	rjson	JSON for R	0.2.15
☐	RJSONIO	Serialize R objects to JSON, JavaScript Object Notation	1.3-0
☐	Rlabkey	Data Exchange Between R and LabKey Server	2.1.136
☐	rlang	Functions for Base Types and Core R and 'Tidyverse' Features	0.1.4
☐	rmarkdown	Dynamic Documents for R	1.7
☐	robustbase	Basic Robust Statistics	0.92-8
☐	rpart	Recursive Partitioning and Regression Trees	4.1-11
☐	rprojroot	Finding Files in Project Subdirectories	1.2
☐	RSQLite	'SQLite' Interface for R	2.0
☐	rstudio	Tools and Utilities for RStudio	0.98.1102
☐	rstudioapi	Safely Access the RStudio API	0.7
☐	rvest	Easily Harvest (Scrape) Web Pages	0.3.2
☐	S4Vectors	S4 implementation of vectors and lists	0.8.11
☐	scales	Scale Functions for Visualization	0.5.0
☐	scatterplot3d	3D Scatter Plot	0.3-40
☐	selectr	Translate CSS Selectors to XPath Expressions	0.3-1
☐	setRNG	Set (Normal) Random Number Generator and Seed	2013.9-1
☐	sfsmisc	Utilities from 'Seminar fuer Statistik' ETH Zurich	1.1-1
☐	shiny	Web Application Framework for R	1.0.5
☐	simpleaffy	Very simple high level analysis of Affymetrix data	2.46.0
☐	sourcetools	Tools for Reading, Tokenizing and Parsing R Code	0.1.6
☐	SparseM	Sparse Linear Algebra	1.77
☐	spatial	Functions for Kriging and Point Pattern Analysis	7.3-11
☐	splines	Regression Spline Functions and Classes	3.2.4

Putting the elements back together

We can do the inverse trick:
put the three variables back together into a character vector, and even add something to it:

Putting the elements back together

RStudio

Aditec_analysis_no_multi_2017111... | Corr.heatmaps.R | Quadratic equation solver.R | R lesson 1.R* | jlEdgeR.R | Plotting bubble plots.R | hs

```
1  # My first line of code is a comment
2  print("Hello world!")
3
4  # Now we are going to create the first variable. Let's call it "My_first_variable"
5  My_first_variable <- "Hello world!"
6
7  # We can make functions to take variables as arguments
8  print(My_first_variable)   # We do not enclose variable names in quotes
9
10 # Let's find out the class of the My_first_variable variable
11 class(My_first_variable)
12
13 # We can also find out the dimensions of the variable (in this case, the length of the vector)
14 dim(My_first_variable) # Which is "NULL", indicating that this variable has no dimensions
15
16 # Let's create a more complex variable:
17 My_second_variable <- c("Hello", "world","!")
18
19 # We can find out the dimensions of the new variable: However it wil not be 1 by 3, as vetors do not have dimensions
20 dim(My_second_variable)
21
22 # However, vectors have length, which can be found out by using the length() function
23 length(My_second_variable)
24
25 # Now let's refer to the individual elements of the character vector:
26 My_second_variable[1]     # This will display the contents of the first element in the command line window
27 one <- My_second_variable[1]    # This will assign the contents of the first element of the My_second_variable to the new variable, "one"
28
29 # And likewise we can do for the second and the third elements:
30 My_first_variable[2]
31 two <- My_second_variable[2]
32 My_second_variable[3]
33 three <- My_second_variable[3]
34
35 # We can put the three variables back together into a character variable, and can also add something to it:
36 My_third_variable <- c(one, two, three, "My name is Vanya", "I am", 38, "years", "old")
```

36:88 (Top Level) R Script

Console ~/Dropbox/Pulendran U19 Immport/Blank Templates/Updated blank templates v.3.17 Mar 2018/json-templates/

```
> My_third_variable
[1] "Hello"           "world"           "!"               "My name is Vanya" "I am"            "38"              "years"
[8] "old"
```

Environment | History

Global Environment

Values

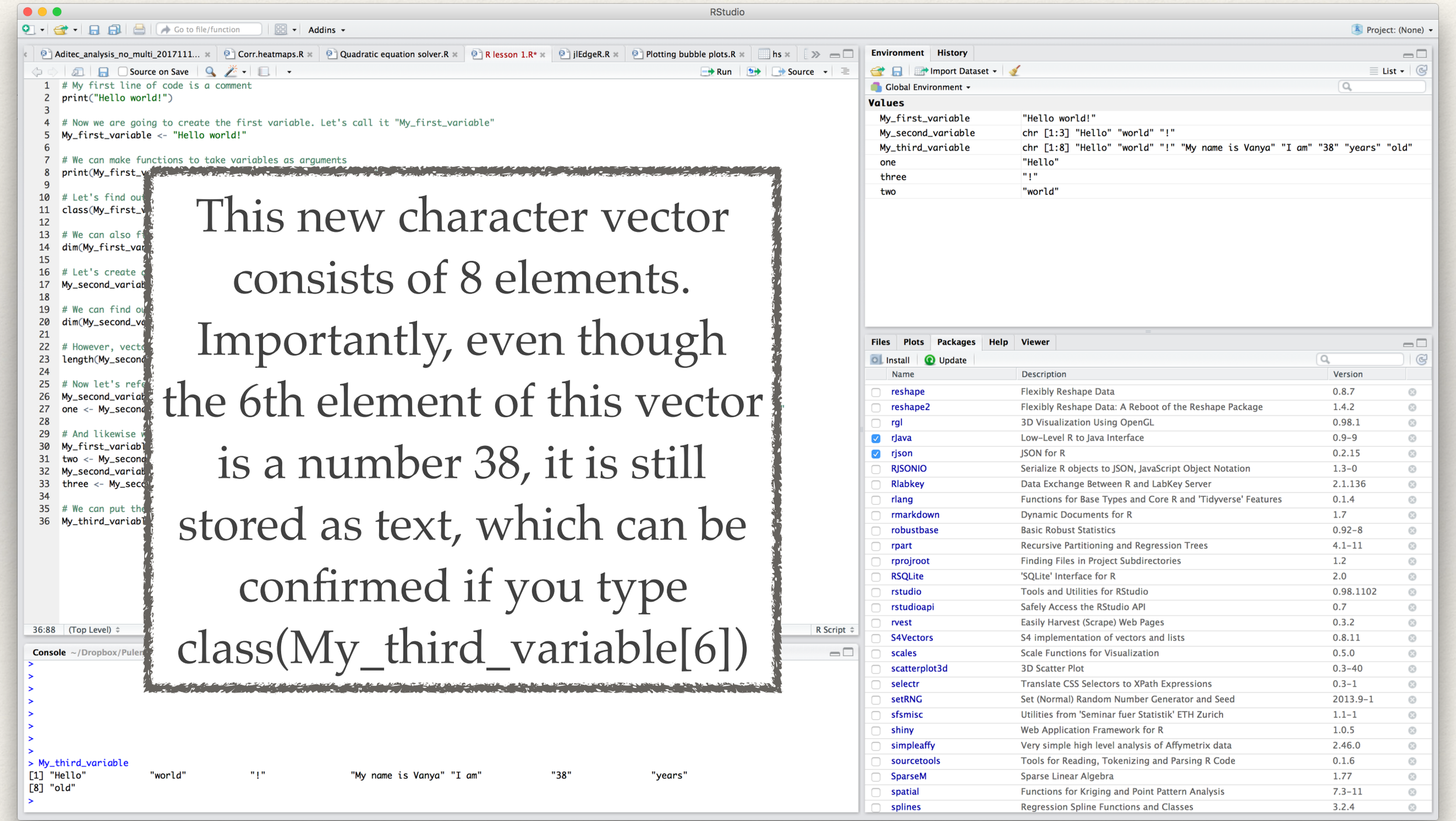
My_first_variable	"Hello world!"
My_second_variable	chr [1:3] "Hello" "world" "!"
My_third_variable	chr [1:8] "Hello" "world" "!" "My name is Vanya" "I am" "38" "years" "old"
one	"Hello"
three	"!"
two	"world"

Files | Plots | Packages | Help | Viewer

Install | Update

	Name	Description	Version
☐	reshape	Flexibly Reshape Data	0.8.7
☐	reshape2	Flexibly Reshape Data: A Reboot of the Reshape Package	1.4.2
☐	rgl	3D Visualization Using OpenGL	0.98.1
☑	rJava	Low-Level R to Java Interface	0.9-9
☑	rjson	JSON for R	0.2.15
☐	RJSONIO	Serialize R objects to JSON, JavaScript Object Notation	1.3-0
☐	Rlabkey	Data Exchange Between R and LabKey Server	2.1.136
☐	rlang	Functions for Base Types and Core R and 'Tidyverse' Features	0.1.4
☐	rmarkdown	Dynamic Documents for R	1.7
☐	robustbase	Basic Robust Statistics	0.92-8
☐	rpart	Recursive Partitioning and Regression Trees	4.1-11
☐	rprojroot	Finding Files in Project Subdirectories	1.2
☐	RSQLite	'SQLite' Interface for R	2.0
☐	rstudio	Tools and Utilities for RStudio	0.98.1102
☐	rstudioapi	Safely Access the RStudio API	0.7
☐	rvest	Easily Harvest (Scrape) Web Pages	0.3.2
☐	S4Vectors	S4 implementation of vectors and lists	0.8.11
☐	scales	Scale Functions for Visualization	0.5.0
☐	scatterplot3d	3D Scatter Plot	0.3-40
☐	selectr	Translate CSS Selectors to XPath Expressions	0.3-1
☐	setRNG	Set (Normal) Random Number Generator and Seed	2013.9-1
☐	sfsmisc	Utilities from 'Seminar fuer Statistik' ETH Zurich	1.1-1
☐	shiny	Web Application Framework for R	1.0.5
☐	simpleaffy	Very simple high level analysis of Affymetrix data	2.46.0
☐	sourcetools	Tools for Reading, Tokenizing and Parsing R Code	0.1.6
☐	SparseM	Sparse Linear Algebra	1.77
☐	spatial	Functions for Kriging and Point Pattern Analysis	7.3-11
☐	splines	Regression Spline Functions and Classes	3.2.4

Putting the elements back together



This new character vector consists of 8 elements. Importantly, even though the 6th element of this vector is a number 38, it is still stored as text, which can be confirmed if you type class(My_third_variable[6])

Can we output the contents of a character vector as a printed line?

RStudio

Aditec_analysis_no_multi_2017111... | Corr.heatmaps.R | Quadratic equation solver.R | R lesson 1.R* | jlEdgeR.R | Plotting bubble plots.R | hs

Source on Save | Run | Source

```
1  # My first line of code is a comment
2  print("Hello world!")
3
4  # Now we are going to create the first variable. Let's call it "My_first_variable"
5  My_first_variable <- "Hello world!"
6
7  # We can make functions to take variables as arguments
8  print(My_first_variable)   # We do not enclose variable names in quotes
9
10 # Let's find out the class of the My_first_variable variable
11 class(My_first_variable)
12
13 # We can also find out the dimensions of the variable (in this case, the length of the vector)
14 dim(My_first_variable) # Which is "NULL", indicating that this variable has no dimensions
15
16 # Let's create a more complex variable:
17 My_second_variable <- c("Hello", "world","!")
18
19 # We can find out the dimensions of the new variable: However it wil not be 1 by 3, as vetors do not have dimensions
20 dim(My_second_variable)
21
22 # However, vectors have length, which can be found out by using the length() function
23 length(My_second_variable)
24
25 # Now let's refer to the individual elements of the character vector:
26 My_second_variable[1]     # This will display the contents of the first element in the command line window
27 one <- My_second_variable[1]    # This will assign the contents of the first element of the My_second_variable to the new variable, "one"
28
29 # And likewise we can do for the second and the third elements:
30 My_first_variable[2]
31 two <- My_second_variable[2]
32 My_second_variable[3]
33 three <- My_second_variable[3]
34
35 # We can put the three variables back together into a character variable, and can also add something to it:
36 My_third_variable <- c(one, two, three, "My name is Vanya", "I am", 38, "years", "old")
37
38 # Can we print the contents of a character array as a readable text?
39 print(My_third_variable)  # Ugh... ths is not what we want
```

39:59 (Top Level) R Script

Console ~/Dropbox/Pulendran U19 Immport/Blank Templates/Updated blank templates v.3.17 Mar 2018/json-templates/

```
>
>
>
> My_third_variable
[1] "Hello"           "world"           "!"               "My name is Vanya" "I am"            "38"              "years"
[8] "old"
> class(My_third_variable[6])
[1] "character"
> print(My_third_variable)
[1] "Hello"           "world"           "!"               "My name is Vanya" "I am"            "38"              "years"
[8] "old"
>
```

Environment | History

Import Dataset | List

Global Environment

Values

My_first_variable	"Hello world!"
My_second_variable	chr [1:3] "Hello" "world" "!"
My_third_variable	chr [1:8] "Hello" "world" "!" "My name is Vanya" "I am" "38" "years" "old"
one	"Hello"
three	"!"
two	"world"

Files | Plots | Packages | Help | Viewer

Install | Update

	Name	Description	Version
☐	reshape	Flexibly Reshape Data	0.8.7
☐	reshape2	Flexibly Reshape Data: A Reboot of the Reshape Package	1.4.2
☐	rgl	3D Visualization Using OpenGL	0.98.1
☑	rJava	Low-Level R to Java Interface	0.9-9
☑	rjson	JSON for R	0.2.15
☐	RJSONIO	Serialize R objects to JSON, JavaScript Object Notation	1.3-0
☐	Rlabkey	Data Exchange Between R and LabKey Server	2.1.136
☐	rlang	Functions for Base Types and Core R and 'Tidyverse' Features	0.1.4
☐	rmarkdown	Dynamic Documents for R	1.7
☐	robustbase	Basic Robust Statistics	0.92-8
☐	rpart	Recursive Partitioning and Regression Trees	4.1-11
☐	rprojroot	Finding Files in Project Subdirectories	1.2
☐	RSQLite	'SQLite' Interface for R	2.0
☐	rstudio	Tools and Utilities for RStudio	0.98.1102
☐	rstudioapi	Safely Access the RStudio API	0.7
☐	rvest	Easily Harvest (Scrape) Web Pages	0.3.2
☐	S4Vectors	S4 implementation of vectors and lists	0.8.11
☐	scales	Scale Functions for Visualization	0.5.0
☐	scatterplot3d	3D Scatter Plot	0.3-40
☐	selectr	Translate CSS Selectors to XPath Expressions	0.3-1
☐	setRNG	Set (Normal) Random Number Generator and Seed	2013.9-1
☐	sfsmisc	Utilities from 'Seminar fuer Statistik' ETH Zurich	1.1-1
☐	shiny	Web Application Framework for R	1.0.5
☐	simpleaffy	Very simple high level analysis of Affymetrix data	2.46.0
☐	sourcetools	Tools for Reading, Tokenizing and Parsing R Code	0.1.6
☐	SparseM	Sparse Linear Algebra	1.77
☐	spatial	Functions for Kriging and Point Pattern Analysis	7.3-11
☐	splines	Regression Spline Functions and Classes	3.2.4

Using cat() function, and the first case of function modifiers

RStudio

Aditec_analysis_no_multi_2017111... | Corr.heatmaps.R | Quadratic equation solver.R | R lesson 1.R* | jlEdgeR.R | Plotting bubble plots.R | hs

```
1  # My first line of code is a comment
2  print("Hello world!")
3
4  # Now we are going to create the first variable. Let's call it "My_first_variable"
5  My_first_variable <- "Hello world!"
6
7  # We can make functions to take variables as arguments
8  print(My_first_variable)   # We do not enclose variable names in quotes
9
10 # Let's find out the class of the My_first_variable variable
11 class(My_first_variable)
12
13 # We can also find out the dimensions of the variable (in this case, the length of the vector)
14 dim(My_first_variable) # Which is "NULL", indicating that this variable has no dimensions
15
16 # Let's create a more complex variable:
17 My_second_variable <- c("Hello", "world","!")
18
19 # We can find out the dimensions of the new variable: However it wil not be 1 by 3, as vetors do not have dimensions
20 dim(My_second_variable)
21
22 # However, vectors have length, which can be found out by using the length() function
23 length(My_second_variable)
24
25 # Now let's refer to the individual elements of the character vector:
26 My_second_variable[1]     # This will display the contents of the first element in the command line window
27 one <- My_second_variable[1]    # This will assign the contents of the first element of the My_second_variable to the new variable, "one"
28
29 # And likewise we can do for the second and the third elements:
30 My_first_variable[2]
31 two <- My_second_variable[2]
32 My_second_variable[3]
33 three <- My_second_variable[3]
34
35 # We can put the three variables back together into a character variable, and can also add something to it:
36 My_third_variable <- c(one, two, three, "My name is Vanya", "I am", 38, "years", "old")
37
38 # Can we print the contents of a character array as a readable text?
39 print(My_third_variable)  # Ugh... ths is not what we want
40 # but what if we try it with cat()? (Anotehr concatenation function. The same function also exists in Bash)
41 cat(My_third_variable, "\n", sep=" ")
42
43
```

42:1 (Top Level) R Script

Console ~/Dropbox/Pulendran U19 Immport/Blank Templates/Updated blank templates v.3.17 Mar 2018/json-templates/

```
> cat(My_third_variable, "\n", sep=" ")
Hello world ! My name is Vanya I am 38 years old
>
```

Environment | History

Import Dataset | List

Global Environment

Values

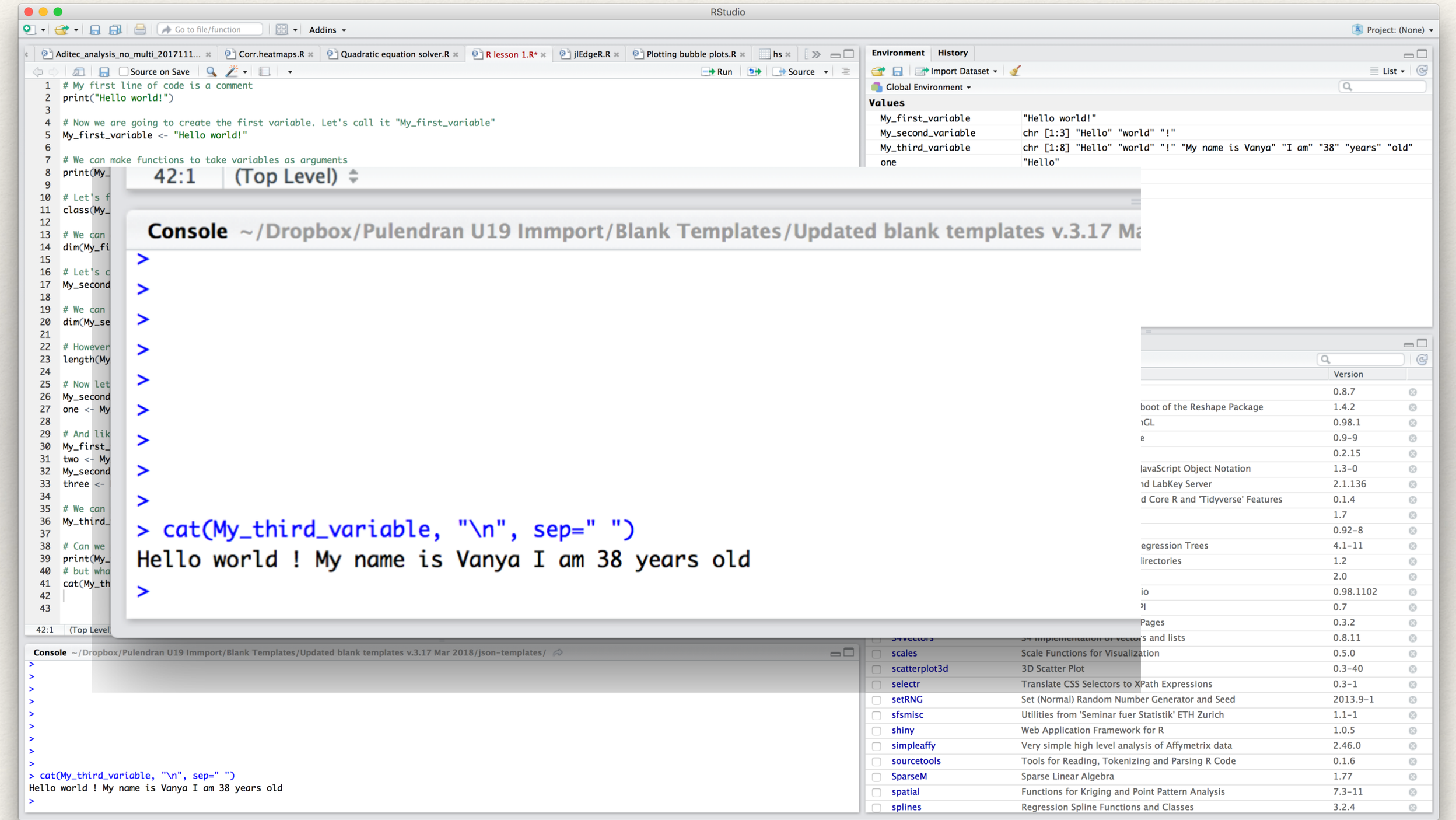
My_first_variable	"Hello world!"
My_second_variable	chr [1:3] "Hello" "world" "!"
My_third_variable	chr [1:8] "Hello" "world" "!" "My name is Vanya" "I am" "38" "years" "old"
one	"Hello"
three	"!"
two	"world"

Files | Plots | Packages | Help | Viewer

Install | Update

	Name	Description	Version
☐	reshape	Flexibly Reshape Data	0.8.7
☐	reshape2	Flexibly Reshape Data: A Reboot of the Reshape Package	1.4.2
☐	rgl	3D Visualization Using OpenGL	0.98.1
☑	rJava	Low-Level R to Java Interface	0.9-9
☑	rjson	JSON for R	0.2.15
☐	RJSONIO	Serialize R objects to JSON, JavaScript Object Notation	1.3-0
☐	Rlabkey	Data Exchange Between R and LabKey Server	2.1.136
☐	rlang	Functions for Base Types and Core R and 'Tidyverse' Features	0.1.4
☐	rmarkdown	Dynamic Documents for R	1.7
☐	robustbase	Basic Robust Statistics	0.92-8
☐	rpart	Recursive Partitioning and Regression Trees	4.1-11
☐	rprojroot	Finding Files in Project Subdirectories	1.2
☐	RSQLite	'SQLite' Interface for R	2.0
☐	rstudio	Tools and Utilities for RStudio	0.98.1102
☐	rstudioapi	Safely Access the RStudio API	0.7
☐	rvest	Easily Harvest (Scrape) Web Pages	0.3.2
☐	S4Vectors	S4 implementation of vectors and lists	0.8.11
☐	scales	Scale Functions for Visualization	0.5.0
☐	scatterplot3d	3D Scatter Plot	0.3-40
☐	selectr	Translate CSS Selectors to XPath Expressions	0.3-1
☐	setRNG	Set (Normal) Random Number Generator and Seed	2013.9-1
☐	sfsmisc	Utilities from 'Seminar fuer Statistik' ETH Zurich	1.1-1
☐	shiny	Web Application Framework for R	1.0.5
☐	simpleaffy	Very simple high level analysis of Affymetrix data	2.46.0
☐	sourcetools	Tools for Reading, Tokenizing and Parsing R Code	0.1.6
☐	SparseM	Sparse Linear Algebra	1.77
☐	spatial	Functions for Kriging and Point Pattern Analysis	7.3-11
☐	splines	Regression Spline Functions and Classes	3.2.4

Using cat() function, and the first case of function modifiers




```
> cat(My_third_variable, "\n", sep=" ")
Hello world ! My name is Vanya I am 38 years old
>
```

Using cat() function, and the first case of function modifiers

Cat() function concatenates all the elements listed before the modifiers, and separated by commas.
In our case, we have the first element “My_third_variable” variable, which itself consists of 8 elements, and the newline character, \n, enclosed in quotes, to indicate that it is a string, and not a variable.

A backslash in front of “n” is a very special symbol called “an escape character”.
It tells the program to treat whatever follows as a “special character”, and not as a regular letter.
For instance, “n” is a just a letter n, but \n is a command to insert a new line (like pressing “enter” on a computer). Likewise, “t” is just a letter t, but “\t” becomes a command to insert a tab (like pressing a tab button on a computer).

Finally, the list of things to concatenate is followed by a function MODIFIER.
In this case, it says sep=“ “, which indicates that the elements in the resulting string should separated from each other by a white space.
We could make this separator to be anything we want, for instance, an underscore, or a word “cooties”:

Using cat() function, and the first case of function modifiers

Cat() function concatenates all the elements listed before the modifiers, and separated by commas.

In our case, we have the first element “My_third_variable” variable, which itself consists of 8 elements, and the newline character, \n, enclosed in quotes, to indicate that it is a string, and not a variable.

```
>
> cat(My_third_variable, "\n", sep=" ")
Hello world ! My name is Vanya I am 38 years old
> cat(My_third_variable, "\n", sep="COOTIES")
HelloCOOTIESworldCOOTIES!COOTIESMy name is VanyaCOOTIESI amCOOTIES38COOTIESyearsCOOTIESoldCOOTIES
>
```

For instance, “n” is a just a letter n, but “\n” is a command to insert a new line (like pressing “enter” on a computer). Likewise, “t” is just a letter t, but “\t” becomes a command to insert a tab (like pressing a tab button on a computer).

Finally, the list of things to concatenate is followed by a function MODIFIER.

In this case, it says sep=“ “, which indicates that the elements in the resulting string should separated from each other by a white space.

We could make this separator to be anything we want, for instance, an underscore, or a word “cooties”:

Using paste() function to assign concatenated strings to variables

Paste() works in a way similar to cat(), but they serve different purposes.
Cat() is used to output the string to the screen, while paste() works to output the resulting string to a variable.

For instance:

```
>
> mitia <- paste("Dmitri", "Kazmin", sep=" ")
> mitia
[1] "Dmitri Kazmin"
>
```

Summary

- Today we have gone through a lot of introductory material
- We have learned about functions, arguments to functions, and function modifiers
- We have learned about variables and saw an example of one kind of a variable - a character vector
- We learned about addressing elements of a vector by their index
- We learned about functions print(), c(), dim(), length(), cat() and paste().