

# How can I avoid the R warning “reached getOption(‘max.print’)” in my code?

Authored by  
**stats writer**

June 28, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I avoid the R warning “reached getOption(‘max.print’)” in my code?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=156627>

To avoid the R warning "reached getOption('max.print')" in your code, you can increase the value of the "max.print" option or use the "message=FALSE" parameter when printing large data sets. Alternatively, you can save the data to a file or use other methods to reduce the size of the output. This will prevent the warning from being triggered and ensure that your code runs smoothly without any interruptions.

## **Avoid R Warning: reached getOption("max.print")**

**One warning message you may encounter in R is:**

**This message appears when you attempt to print more than 1000 values at once in RStudio.**

**By default, RStudio only lets you print 1000 values at once. However, you can easily increase this limit by using one of the following methods:**

### **Method 1: Increase Limit to Specific Value**

```
#increase print limit to 2000 values  
options(max.print=2000)
```

### **Method 2: Increase Limit to Max Amount Allowed by Machine**

```
#increase print limit to max allowed by your machine
```

```
options(max.print = .Machine$integer.max)
```

The following example shows how to use these methods in practice.

Example: Increase Print Limit in R

Suppose we create a data frame in R with 1,002 rows and 2 columns:

```
#make this example reproducible  
set.seed(0)
```

```
#create data frame  
df <- data.frame(x=runif(1002),  
y=runif(1002))
```

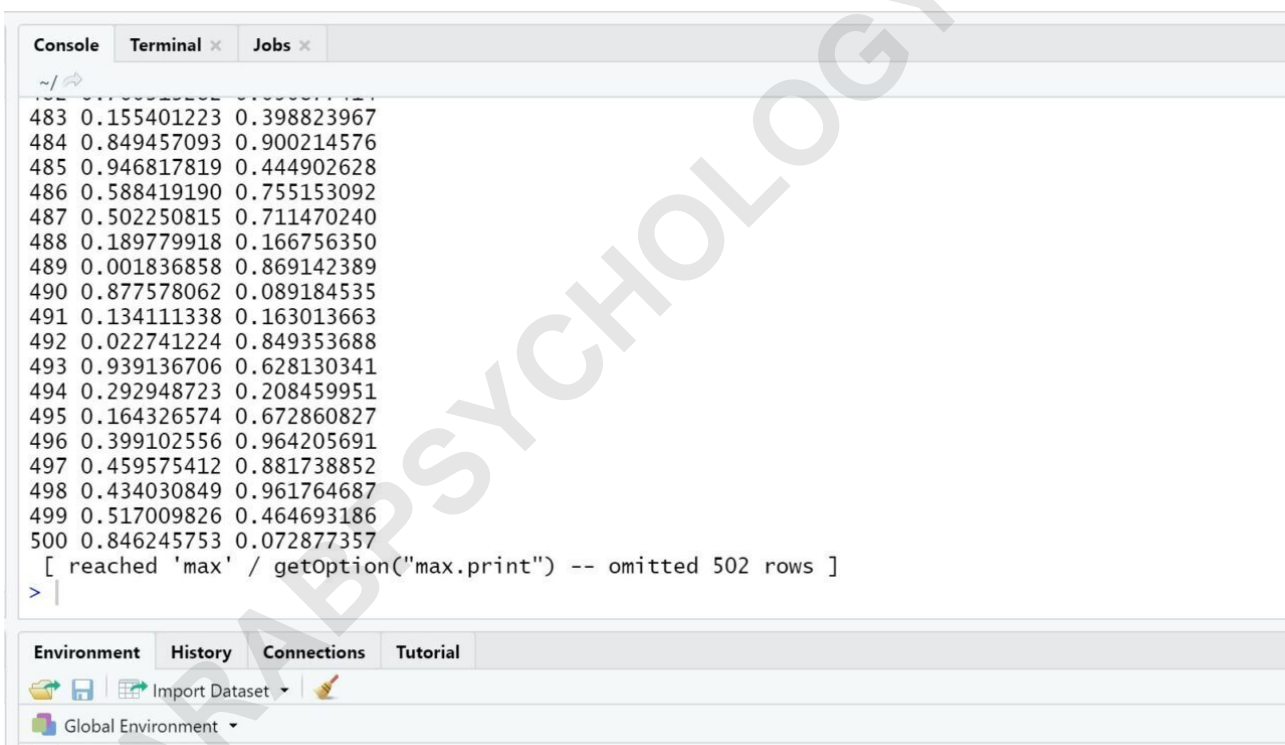
```
#view head of data frame  
head(df)
```

```
x y  
1 0.8966972 0.68486090  
2 0.2655087 0.38328339  
3 0.3721239 0.95498800  
4 0.5728534 0.11835658  
5 0.9082078 0.03910006
```

**6 0.2016819 0.50450503**

**Then suppose we attempt to print the entire data frame in RStudio:**

**#attempt to print entire data frame  
df**



The screenshot shows the RStudio interface with the Console pane open. The console displays the following output:

```
483 0.155401223 0.398823967
484 0.849457093 0.900214576
485 0.946817819 0.444902628
486 0.588419190 0.755153092
487 0.502250815 0.711470240
488 0.189779918 0.166756350
489 0.001836858 0.869142389
490 0.877578062 0.089184535
491 0.134111338 0.163013663
492 0.022741224 0.849353688
493 0.939136706 0.628130341
494 0.292948723 0.208459951
495 0.164326574 0.672860827
496 0.399102556 0.964205691
497 0.459575412 0.881738852
498 0.434030849 0.961764687
499 0.517009826 0.464693186
500 0.846245753 0.072877357
 [ reached 'max' / getOption("max.print") -- omitted 502 rows ]
> |
```

The bottom of the console shows the RStudio environment pane with tabs for Environment, History, Connections, and Tutorial. The Environment pane shows a Global Environment.

**We're only able to view the first 500 rows (i.e. the first 1,000 values) and we receive the warning that 502 rows were omitted.**

**However, if we use the max.print function then we can**

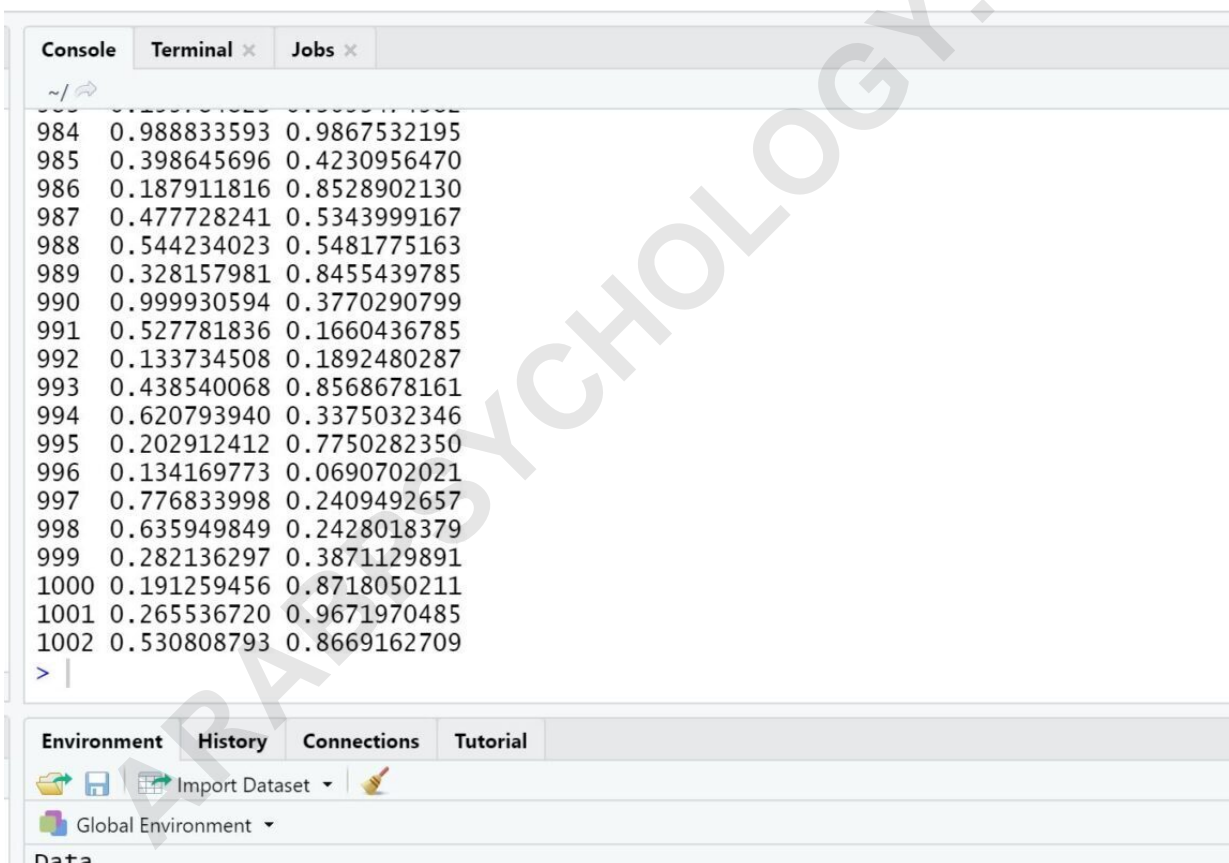
**increase the print limit to 2,500 values:**

**#increase print limit to 2500 values**

**options(max.print=2500)**

**#attempt to print entire data frame again**

**df**



```
~/>
984 0.988833593 0.9867532195
985 0.398645696 0.4230956470
986 0.187911816 0.8528902130
987 0.477728241 0.5343999167
988 0.544234023 0.5481775163
989 0.328157981 0.8455439785
990 0.999930594 0.3770290799
991 0.527781836 0.1660436785
992 0.133734508 0.1892480287
993 0.438540068 0.8568678161
994 0.620793940 0.3375032346
995 0.202912412 0.7750282350
996 0.134169773 0.0690702021
997 0.776833998 0.2409492657
998 0.635949849 0.2428018379
999 0.282136297 0.3871129891
1000 0.191259456 0.8718050211
1001 0.265536720 0.9671970485
1002 0.530808793 0.8669162709
> |
```

**This time we're able to print all 1,002 rows in the data frame and we receive no warning message since we increased the print limit.**

If we'd like to take it to the extreme and set the print limit to the max number of values allowed by our machine, we can use the following syntax:

```
#increase print limit to max allowed by your machine  
options(max.print = .Machine$integer.max)
```

However, only use this option if you absolutely need to be able to view every line in your data frame because this can take a long time to run if the data that you're working with is extremely large.

Additional Resources

The following tutorials explain how to troubleshoot other common errors in R:

**[How to Fix in R: longer object length is not a multiple of shorter object length](#)**