

# RESEARCH TOOLS 2011

## LECTURE 25

2011-Nov-29

Lecture by Rob Braswell

Kurt Schwehr

<http://schwehr.org>

UNH CCOM/JHC

R for Statistics



```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
c:b4:75:ce:9b:6c:48:94:56:a1:fe not verified (check hostfingerprints or web.c
rts config setting)
adding changesets
adding manifests
adding file changes
added 1 changesets with 1 changes to 1 files
(run 'hg update' to get a working copy)
researchtools@ubuntu:~/projects/researchtools$ fix-ntp
[sudo] password for researchtools:
29 Nov 11:10:31 ntpdate[4604]: step time server 91.189.94.4 offset 1437.71403
ec
researchtools@ubuntu:~/projects/researchtools$ cd ~/class/25
researchtools@ubuntu:~/class/25$ ls -l
total 220
-rw-r--r-- 1 researchtools researchtools 74929 2011-11-29 09:31 25-R-lab1-Int
pdf
-rw-r--r-- 1 researchtools researchtools 97334 2011-11-29 09:31 25-R-lab2-C02
f
-rw-r--r-- 1 researchtools researchtools 41263 2011-11-29 09:31 25-R-lab3-ANO
pdf
lrwxrwxrwx 1 researchtools researchtools 68 2011-11-29 10:17 25-r-statisti
org -> /home/researchtools/projects/researchtools/class/25-r-statistics.org
researchtools@ubuntu:~/class/25$ xdg-open 25-R-lab1-Intro.pdf
researchtools@ubuntu:~/class/25$

```

```

emacs23@ubuntu
File Edit Options Buffers Tools Operate Mark Regexp Immediate Subdir Help
/home/researchtools/class/25:
total used in directory 228 available 10177724
drwxr-xr-x 2 researchtools researchtools 4096 2011-11-29 10:17 .
drwxr-xr-x 20 researchtools researchtools 4096 2011-11-29 10:17 ..
-rw-r--r-- 1 researchtools researchtools 74929 2011-11-29 09:31 25-R-lab1-Intro.pdf
-rw-r--r-- 1 researchtools researchtools 97334 2011-11-29 09:31 25-R-lab2-C02.pdf
-rw-r--r-- 1 researchtools researchtools 41263 2011-11-29 09:31 25-R-lab3-ANOVA.pdf
lrwxrwxrwx 1 researchtools researchtools 68 2011-11-29 10:17 25-r-statistics.org -> /home/
researchtools/projects/researchtools/class/25-r-statistics.org

```

R is an "interpreted language", which means it executes statements (i.e., commands) one at a time. You will typically perform analyses with several statements, which are (1) either typed directly on the command line, or (2) entered into a text file that is executed all at once. For this introduction we will focus on the former.

**The Basics.** The most simple usage of R is as a calculator. Type the following statements at the R prompt (">"), e.g., for the first one, just type "3" and press "return".

```

> 3
[1] 3
> 3 + 5
[1] 8
> 3.4 + 6
[1] 9.4
> x = (3.4 + 6)/4.
> x
[1] 2.35
> y = sin(pi*x)
> y
[1] 0.6910045

```

For the statements that return a value, the values are numbered, but in the examples above there is only one returned item, indicated by "[1]".

If you are new to programming, we refer to statements like `x=10` as an "assignment". The equals sign indicates that the value to its right is being inserted into a variable called `x`. Also, it is important to remember that the names `x` and `y` above are chosen by the user, and

```

-U:%%- 25 All L5 (Dired by name)-----
kurtvm on #unhresearchtools (+,lag:0)
<sthein> Hi ! [11:11]
*** bwelton (-chatzill@lab6.ccom.nh) has joined channel #unhresearchtools
<bwelton> hello
<kurtvm> make sure that you have installed R [11:12]
<kurtvm> sudo apt-get install r-base
*** hminami (-chatzill@192.168.8.245) has joined channel #unhresearchtools
<hminami> hi
ERC>
-U:***- #unhresearchtools@Unknown Bot L28 (ERC)-----

```

```
researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
researchtools@ubuntu:~/class/25$ xdg-open 25-R-lab1-Intro.pdf
```

```
emacs23@ubuntu
File Edit Options Buffers Tools Minibuf Help
* A place for notes on class 25
* Introduction
* Why R?
- Free
- Tons of addons
- Lots of developers and users
Why not?
- R syntax is a little awkward
- Complex heirarchy of objects - a little opaque
- It's not python
```

R is an "interpreted language", which means it executes statements (i.e., commands) one at a time. You will typically perform analyses with several statements, which are (1) either typed directly on the command line, or (2) entered into a text file that is executed all at once. For this introduction we will focus on the former.

**The Basics.** The most simple usage of R is as a calculator. Type the following statements at the R prompt (>), e.g., for the first one, just type "3" and press "return".

```
> 3
[1] 3
> 3 + 5
[1] 8
> 3.4 + 6
[1] 9.4
> x = (3.4 + 6)/4.
> x
[1] 2.35
> y = sin(4*x)
> y
[1] 0.8910045
```

For the statements that return a value, the values are numbered, but in the examples above there is only one returned item, indicated by "[1]".

If you are new to programming, we refer to statements like `x=10` as an "assignment". The equals sign indicates that the value to its right is being inserted into a variable called `x`. Also, it is important to remember that the names `x` and `y` above are chosen by the user, and

```
-U:**- 25-notes.org All L14 (Org)-----
kurtvm on #unhresearchtools (+,lag:0)
<sthein> Hi ! [11:11]
*** bwelton (~chatzill@lab6.ccom.nh) has joined channel #unhresearchtools
<bwelton> hello
<kurtvm> make sure that you have installed R [11:12]
<kurtvm> sudo apt-get install r-base
*** hminami (~chatzill@192.168.8.245) has joined channel #unhresearchtools
<hminami> hi
<mluke> hoody hoo [11:13]
ERC>
```

```
-U:**- #unhresearchtools@Unknown Bot L29 (ERC)-----
M-x flyspell-mode
```

# Index of /~braswell/eos864

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>	-	-	-
<a href="#">assignments/</a>	28-Nov-2011 09:41	-	-
<a href="#">data/</a>	28-Nov-2011 09:41	-	-
<a href="#">lectures/</a>	28-Nov-2011 09:41	-	-
<a href="#">materials/</a>	28-Nov-2011 09:41	-	-
<a href="#">scripts/</a>	28-Nov-2011 09:41	-	-
<a href="#">spring09/</a>	28-Nov-2011 09:41	-	-

Apache/2.2.12 (Ubuntu) Server at waage.sr.unh.edu Port 80

either typed directly on the command line, or (2) entered into a text file that is executed all at once. For this introduction we will focus on the former.

**The Basics.** The most simple usage of R is as a calculator. Type the following statements at the R prompt (>), e.g., for the first one, just type "3" and press "return".

```
> 3
[1] 3
> 3 + 5
[1] 8
> 3 - 4 + 6
[1] 5
> x = (3 - 4 + 6) / 4
> x
[1] 2.25
> y = sin(pi * x)
> y
[1] 0.8910045
```

For the statements that return a value, the values are numbered, but in the examples above there is only one returned item, indicated by "[1]".

If you are new to programming, we refer to statements like `x=10` as an "assignment". The equals sign indicates that the value to its right is being inserted into a variable called `x`. Also, it is important to remember that the names `x` and `y` above are chosen by the user, and

\* A place for notes on class 25

\* Introduction

\* Why R?

- Free
- Tons of addons
- Lots of developers and users

Why not?

- R syntax is a little awkward
- Complex heirarchy of objects - a little opaque
- It's not python

\* Links / See also

- [usingR.pdf](#)
- <http://r-project.org>
- [R-intro.pdf](#)
- [E0895 Course material http://waage/~braswell/eos864/](#)
- <http://waage.sr.unh.edu/~braswell/eos864/>
- <http://r-i>

kurtvm on #unhresearchtools (+,lag:0)

```
*** #unhresearchtools modes: +
<kurtvm> http://waage.sr.unh.edu/~braswell/eos864/
ERC> |
```

researchtools@ubuntu: ~/class/25

R Graph Gallery :: thumbnails gallery - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Index of /~braswell/eos864 R-bloggers | R news & tutori... R Graph G

addictedtor.free.fr/graphiques/thumbs.php

Graph Gallery : Enhance your data visualization with R

Home List View Thumbnails View

Transferring data from addictedtor.free.fr...

emacs23@ubuntu

File Edit Options Buffers Tools Org Tbl Help

- R syntax is a little awkward
- Complex heirarchy of objects - a little opaque
- It's not python

• Links / See also

- [usingR.pdf](#)
- <http://r-project.org>
- [R-intro.pdf](#)
- [E0895 Course material http://waage/~braswell/eos864/](#)
- <http://waage.sr.unh.edu/~braswell/eos864/>
- <http://r-bloggers.com>
- [graphics gallery http://addictedtor.free.fr/graphiques/thumbs.php](http://addictedtor.free.fr/graphiques/thumbs.php)

1

either typed directly on the command line, or (2) entered into a text file that is executed all at once. For this introduction we will focus on the former.

**The Basics.** The most simple usage of R is as a calculator. Type the following statements at the R prompt (>), e.g., for the first one, just type "3" and press "return".

```
> 3
[1] 3
> 3 + 5
[1] 8
> 3.4 + 6
[1] 9.4
> x = (3.4 + 6)/4.
> x
[1] 2.35
> y = sin(pi*x)
> y
[1] 0.6910045
```

2

For the statements that return a value, the values are numbered, but in the examples above there is only one returned item, indicated by "[1]".

If you are new to programming, we refer to statements like `x=10` as an "assignment". The equals sign indicates that the value to its right is being inserted into a variable called `x`. Also, it is important to remember that the names `x` and `y` above are chosen by the user, and

-U:\*\*\*- 25-notes.org Bot L25 (Org Fly)-----

kurtvm on #unhresearchtools (+,lag:0)

\*\*\* #unhresearchtools modes: +

<kurtvm> <http://waage.sr.unh.edu/~braswell/eos864/>

<matt w> hi

ERC> [ ]

-U:\*\*\*- #unhresearchtools@Unknown Bot L38 (ERC)-----

researchtools@ubuntu: ~/class/25

emacs23@ubuntu

R Graph Gallery (158) Image lag plot matrix - Mozilla Firefox

File Edit View History Bookmarks Tools Help

addictedtor.free.fr/graphiques/RGraphGallery.php?graph=158

RELATED SITES: R-project | CRAN | Bioconductor | R Wiki

Graph Gallery: Enhance your data visualization with R

Home List View Thumbnails View

lag 1 lag 2 lag 3 lag 4

max 0

### Image lag plot matrix

#### Description

Produces an image lag plot matrix of large timeseries where the colors encode the density of the points in the lag plots. In this example, the auto correlations with lag 1-4 of O3 concentrations in Aldorf (Switzerland) in 2004 can be compared.

Requirements

Author(s)  
René Locher

Publicly recommend on Google. ?

+1 1

source code

javascript:void(0);

assignments: the equals sign indicates that the value to its right is being inserted into a variable called x. Also, it is important to remember that the names x and y above are chosen by the user, and

Help

le opaque

braswell/eos864/  
864/

ee.fr/graphiques/thumbs.php

ll/eos864/

```
-U: ** - #unhresearchtools@Unknown Bot L38 (ERC)
Wrote /home/researchtools/class/25/25-notes.org
```

researchtools@ubuntu:~/class/25\$



- R syntax is a little awkward
- Complex heirarchy of objects - a little opaque
- It's not python

\* Links / See also

- [usingR.pdf](#)
- <http://r-project.org>
- [R-intro.pdf](#)
- [E0895 Course material](#) <http://waage/~braswell/eos864/>
- <http://waage.sr.unh.edu/~braswell/eos864/>
- <http://r-bloggers.com>
- [graphics gallery](http://addictedtor.free.fr/graphiques/thumbs.php) <http://addictedtor.free.fr/graphiques/thumbs.php>

\* Data types

- vector - all the same type
- array - can have two or more dimensions - ragged possible
- matrix - stricter than arrays. 2d and rectangular
- list - similar to python list
- factor - describes discrete classes
- data frames

Previous Next 1 (1 of 9) Fit Page Width

Thumbnails



### Worksheet 1: INTRODUCTION TO R

R is an "interpreted language", which means it executes statements (i.e., commands) one at a time. You will typically perform analyses with several statements, which are (1) either typed directly on the command line, or (2) entered into a text file that is executed all at once. For this introduction we will focus on the former.

**The Basics.** The most simple usage of R is as a calculator. Type the following statements at the R prompt (>), e.g., for the first one, just type "3" and press "return".

```
> 3
[1] 3
> 3 + 5
[1] 8
> 3.4 + 6
[1] 9.4
> x = (3.4 + 6)/4.
> x
[1] 2.35
> y = sin(pi*x)
> y
[1] 0.6910045
```

For the statements that return a value, the values are numbered, but in the examples above there is only one returned item, indicated by "[1]".

If you are new to programming, we refer to statements like `x=10` as an "assignment". The equals sign indicates that the value to its right is being inserted into a variable called `x`. Also, it is important to remember that the names `x` and `y` above are chosen by the user, and

-U:\*\*- 25-notes.org Bot L28 (Org Fly)

kurtvm on #unhresearchtools (+,lag:0)

```
*** #unhresearchtools modes: +
<kurtvm> http://waage.sr.unh.edu/~braswell/eos864/
<matt w> hi
ERC>
```

-U:\*\*- #unhresearchtools@Unknown Bot L38 (ERC)

M-x replace-string

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> letters
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
[20] "t" "u" "v" "w" "x" "y" "z"
> sunspots
  Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1749 58.0 62.6 70.0 55.7 85.0 83.5 94.8 66.3 75.9 75.5 158.6 85.2
1750 73.3 75.9 89.2 88.3 90.0 100.0 85.4 103.0 91.2 65.7 63.3 75.4
1751 70.0 43.5 45.3 56.4 60.7 50.7 66.3 59.8 23.5 23.2 28.5 44.0
1752 35.0 50.0 71.0 59.3 59.7 39.6 78.4 29.3 27.1 46.6 37.6 40.0
1753 44.0 32.0 45.7 38.0 36.0 31.7 22.2 39.0 28.0 25.0 20.0 6.7

```

```

emacs23@ubuntu
lit Options Buffers Tools Org Tbl Help

graphics gallery http://addictedtor.free.fr/graphiques/thumbs.php

data types
vector | all the same type
array  | can have two or more dimensions, can be ragged
matrix | stricter than arrays. 2d and rectangular
list   | similar to python list
factor | describes discrete classes
data frames |

R is a functional language

R comes with build in datasets

```

**1**

R is an "interpreted language", which means it executes statements (i.e., commands) one at a time. You will typically perform analyses with several statements, which are (1) either typed directly on the command line, or (2) entered into a text file that is executed all at once. For this introduction we will focus on the former.

**The Basics.** The most simple usage of R is as a calculator. Type the following statements at the R prompt (>), e.g., for the first one, just type "3" and press "return".

```

> 3
[1] 3
> 3 + 5
[1] 8
> 3.4 + 6
[1] 9.4
> x = (3.4 + 6)/4.
> x
[1] 2.35
> y = sin(pi*x)
> y
[1] 0.6910045

```

For the statements that return a value, the values are numbered, but in the examples above there is only one returned item, indicated by "[1]".

If you are new to programming, we refer to statements like `x=10` as an "assignment". The equals sign indicates that the value to its right is being inserted into a variable called `x`. Also, it is important to remember that the names `x` and `y` above are chosen by the user, and

**2**

```

-U:***- 25-notes.org Bot L39 (Org Fly)-----
kurtvm on #unhresearchtools (+,lag:0)
*** #unhresearchtools modes: +
<kurtvm> http://waage.sr.unh.edu/~braswell/eos864/
<matt w> hi
ERC>

-U:***- #unhresearchtools@Unknown Bot L38 (ERC)-----

```



```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
1955 23.1 20.8 4.9 11.3 28.9 31.7 26.7 40.7 42.7 58.5 89.2 76.9
1956 73.6 124.0 118.4 110.7 136.6 116.6 129.1 169.6 173.2 155.3 201.3 192.1
1957 165.0 130.2 157.4 175.2 164.6 200.7 187.2 158.0 235.8 253.8 210.9 239.4
1958 202.5 164.9 190.7 196.0 175.3 171.5 191.4 200.2 201.2 181.5 152.3 187.6
1959 217.4 143.1 185.7 163.3 172.0 168.7 149.6 199.6 145.2 111.4 124.0 125.0
1960 146.3 106.0 102.2 122.0 119.6 110.2 121.7 134.1 127.2 82.8 89.6 85.6
1961 57.9 46.1 53.0 61.4 51.0 77.4 70.2 55.9 63.6 37.7 32.6 40.0
1962 38.7 50.3 45.6 46.4 43.7 42.0 21.8 21.8 51.3 39.5 26.9 23.2
1963 19.8 24.4 17.1 29.3 43.0 35.9 19.6 33.2 38.8 35.3 23.4 14.9
1964 15.3 17.7 16.5 8.6 9.5 9.1 3.1 9.3 4.7 6.1 7.4 15.1
1965 17.5 14.2 11.7 6.8 24.1 15.9 11.9 8.9 16.8 20.1 15.8 17.0
1966 28.2 24.4 25.3 48.7 45.3 47.7 56.7 51.2 50.2 57.2 57.2 70.4
1967 110.9 93.6 111.8 69.5 86.5 67.3 91.5 107.2 76.8 88.2 94.3 126.4
1968 121.8 111.9 92.2 81.2 127.2 110.3 96.1 109.3 117.2 107.7 86.0 109.8
1969 104.4 120.5 135.8 106.8 120.0 106.0 96.8 98.0 91.3 95.7 93.5 97.9
1970 111.5 127.8 102.9 109.5 127.5 106.8 112.5 93.0 99.5 86.6 95.2 83.5
1971 91.3 79.0 60.7 71.8 57.5 49.8 81.0 61.4 50.2 51.7 63.2 82.2
1972 61.5 88.4 80.1 63.2 80.5 88.0 76.5 76.8 64.0 61.3 41.6 45.3
1973 43.4 42.9 46.0 57.7 42.4 39.5 23.1 25.6 59.3 30.7 23.9 23.3
1974 27.6 26.0 21.3 40.3 39.5 36.0 55.8 33.6 40.2 47.1 25.0 20.5
1975 18.9 11.5 11.5 5.1 9.0 11.4 28.2 39.7 13.9 9.1 19.4 7.8
1976 8.1 4.3 21.9 18.8 12.4 12.2 1.9 16.4 13.5 20.6 5.2 15.3
1977 16.4 23.1 8.7 12.9 18.6 38.5 21.4 30.1 44.0 43.8 29.1 43.2
1978 51.9 93.6 76.5 99.7 82.7 95.1 70.4 58.1 138.2 125.1 97.9 122.7
1979 166.6 137.5 138.0 101.5 134.4 149.5 159.4 142.2 188.4 186.2 183.3 176.3
1980 159.6 155.0 126.2 164.1 179.9 157.3 136.3 135.4 155.0 164.7 147.9 174.4
1981 114.0 141.3 135.5 156.4 127.5 90.0 143.8 158.7 167.3 162.4 137.5 150.1
1982 111.2 163.6 153.8 122.0 82.2 110.4 106.1 107.6 118.8 94.7 98.1 127.0
1983 84.3 51.0 66.5 80.7 99.2 91.1 82.2 71.8 50.3 55.8 33.3 33.4
>
Save workspace image? [y/n/c]: n
researchtools@ubuntu:~/class/25$ # Just pressed Control d to quit
researchtools@ubuntu:~/class/25$

```

```

emacs23@ubuntu
lit Options Buffers Tools Org Tbl Help
graphics gallery http://addictedtor.free.fr/graphiques/thumbs.php
ca types
ctor | all the same type |
ray | can have two or more dimensions, can be ragged |
trix | stricter than arrays. 2d and rectangular |
st | similar to python list |
ctor | describes discrete classes |
ca frames |
is a functional language
comes with build in datasets
roduction to R
25-notes.org Bot L42 (Org Fly)-----
on #unhresearchtools (+,lag:0)
/unhresearchtools modes: +
vm> http://waage.sr.unh.edu/~braswell/eos864/
w> hi

```

```

> 3 + 5
[1] 8
> 3.4 + 6
[1] 9.4
> x = (3.4 + 6)/4.
> x
[1] 2.35
> y = sin(4*x)
> y
[1] 0.8910045

```

For the statements that return a value, the values are numbered, but in the examples above there is only one returned item, indicated by "[1]".

If you are new to programming, we refer to statements like `x=10` as an "assignment". The equals sign indicates that the value to its right is being inserted into a variable called `x`. Also, it is important to remember that the names `x` and `y` above are chosen by the user, and

```

-U: *- #unhresearchtools@Unknown Bot L38 (ERC)-----
Wrote /home/researchtools/class/25/25-notes.org

```

researchtools@ubuntu: ~/class/25

File Edit View Search Terminal Help

Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.

```
> 3
[1] 3
> x <- 3
> x = 3
> # comment
> # Both = and <- are the same
> sin(.1)
[1] 0.09983342
> y = sin(.1)
> 4/3
[1] 1.333333
> 4/3.
[1] 1.333333
> stuff
Error: object 'stuff' not found
> ?
+
+
+
+
+ q
>
> 3+
+ 4
[1] 7
> ?
+
+ # Control C or escape to get out of the line continuation
> █
```

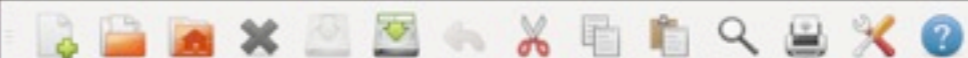
```
+ 4.5 + log(4.5 ) /
+ 44.9)
> z
[1] 0.8534074
```

which is good to know, but as you can see this is  
want to avoid.

2

emacs23@ubuntu

File Edit Options Buffers Tools ERC Help



\* R is a functional language

Also comes with build in [datasets](#)

\* Introduction to R

```
#+BEGIN_SRC r
```

```
3
x <- 3
x = 3
# comment
# Both = and <- are the same
#+END_SRC
```

```
█
```

-U:--- 25-notes.org Bot L49 (Org Fly)-----

kurtvm on #unhresearchtools (+,lag:0)

```
*** #unhresearchtools modes: +
<kurtvm> http://waage.sr.unh.edu/~braswell/eos864/
<matt w> hi
ERC> █
```

-U:\*\*- #unhresearchtools@Unknown Bot L38 (ERC)-----

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
1973 43.4 42.9 46.0 57.7 42.4 39.5 23.1 25.6 59.3 30.7 23.9 23
1974 27.6 26.0 21.3 40.3 39.5 36.0 55.8 33.6 40.2 47.1 25.0 20
1975 18.9 11.5 11.5 5.1 9.0 11.4 28.2 39.7 13.9 9.1 19.4 7
1976 8.1 4.3 21.9 18.8 12.4 12.2 1.9 16.4 13.5 20.6 5.2 15
1977 16.4 23.1 8.7 12.9 18.6 38.5 21.4 30.1 44.0 43.8 29.1 43
1978 51.9 93.6 76.5 99.7 82.7 95.1 70.4 58.1 138.2 125.1 97.9 122
1979 166.6 137.5 138.0 101.5 134.4 149.5 159.4 142.2 188.4 186.2 183.3 176
1980 159.6 155.0 126.2 164.1 179.9 157.3 136.3 135.4 155.0 164.7 147.9 174
1981 114.0 141.3 135.5 156.4 127.5 90.0 143.8 158.7 167.3 162.4 137.5 150
1982 111.2 163.6 153.8 122.0 82.2 110.4 106.1 107.6 118.8 94.7 98.1 127
1983 84.3 51.0 66.5 80.7 99.2 91.1 82.2 71.8 50.3 55.8 33.3 33
> z = x+sin(pi/4)* (x -
+ 4.5 + log(4.5) / 44.9)
> z
[1] 1.963027
> # the + was added by R saying that it wants more input
> letters
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r"
[20] "t" "u" "v" "w" "x" "y" "z"
> x = letters
> x[0]
character(0)
> x[1]
[1] "a"
> x[2]
[1] "b"
> x[-1]
[1] "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r"
[20] "u" "v" "w" "x" "y" "z"
> x = c(1,2,3,4,5)
> x[-2]
[1] 1 3 4 5
>

```

```

emacs23@ubuntu
File Edit Options Buffers Tools Org Tbl Help
* R is a functional language
* Introduction to R
#+BEGIN_SRC r
3
x <- 3
x = 3
# comment
# Both = and <- are the same
#+END_SRC

--+= is the continuation mark. R is asking you for more stuff

*WARNING:* R starts counting from 1, *not* zero.

Also comes with build in datasets

#+BEGIN_SRC r
letters
sunspots
#+END_SRC

```

```

-U:--- 25-notes.org Bot L56 (Org Fly)-----
kurtvm on #unhresearchtools (+,lag:0)
*** #unhresearchtools modes: +
<kurtvm> http://waage.sr.unh.edu/~braswell/eos864/
<matt w> hi
ERC>

-U:***- #unhresearchtools@Unknown Bot L38 (ERC)-----
Wrote /home/researchtools/class/25/25-notes.org

```

$$\frac{+ 4.5 + \log(4.5)}{+ 44.9}$$

$$> z$$

$$[1] 0.8534074$$

which is good to know, but as you can want to avoid.

researchtools@ubuntu: ~/class/25

File Edit View Search Terminal Help

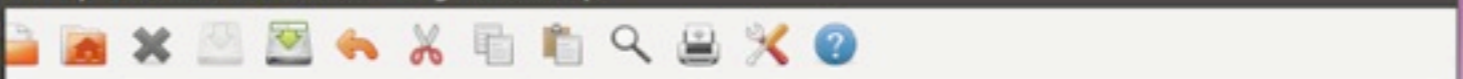
```

1982 111.2 163.6 153.8 122.0 82.2 110.4 106.1 107.6 118.8 94.7 98.1 127.0
1983 84.3 51.0 66.5 80.7 99.2 91.1 82.2 71.8 50.3 55.8 33.3 33.4
> z = x+sin(pi/4)* (x -
+ 4.5 + log(4.5) / 44.9)
> z
[1] 1.963027
> # the + was added by R saying that it wants more input
> letters
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
[20] "t" "u" "v" "w" "x" "y" "z"
> x = letters
> x[0]
character(0)
> x[1]
[1] "a"
> x[2]
[1] "b"
> x[-1]
[1] "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t"
[20] "u" "v" "w" "x" "y" "z"
> x = c(1,2,3,4,5)
> x[-2]
[1] 1 3 4 5
> x = c(2,4,5,60)
> x
[1] 2 4 5 60
> month.abb
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"
> month
Error: object 'month' not found
> subspot.month
Error: object 'subspot.month' not found
> sunspot.mo

```

emacs23@ubuntu

lit Options Buffers Tools Org Tbl Help



is the continuation mark. R is asking you for more stuff

WARNING:\* R starts counting from 1, \*not\* zero.

comes with build in [datasets](#)

GIN\_SRC r

ers

oots

0\_SRC

```

[19] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o"
> month.abb
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct"
"Nov" "Dec"
There are also built-in data sets for testing, but we'll get to
all these things in more detail later.
> sunspot.month
      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct
Nov Dec
1749 58.0 62.6 70.0 55.7 85.0 83.5 94.8 66.3 75.9 75.5
158.6 85.2
1750 73.3 75.9 89.2 88.3 90.0 100.0 85.4 103.0 91.2 65.7
63.3 75.4
1751 70.0 43.5 45.3 56.4 60.7 50.7 66.3 59.8 23.5 23.2
28.5 44.0

```

researchtools@ubuntu: ~/class/25

File Edit View Search Terminal Help

```
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"
> month
Error: object 'month' not found
> subspot.month
Error: object 'subspot.month' not found
> sunspot.month
  Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
1749 58.0 62.6 70.0 55.7 85.0 83.5 94.8 66.3 75.9 75.5 158.6 85.2
1750 73.3 75.9 89.2 88.3 90.0 100.0 85.4 103.0 91.2 65.7 63.3 75.4
1751 70.0 43.5 45.3 56.4 60.7 50.7 66.3 59.8 23.5 23.2 28.5 44.0
1752 35.0 50.0 71.0 59.3 59.7 39.6 78.4 29.3 27.1 46.6 37.6 40.0
1753 44.0 32.0 45.7 38.0 36.0 31.7 22.0 39.0 28.0 25.0 20.0 6.7
1754 0.0 3.0 1.7 13.7 20.7 26.7 18.8 12.3 8.2 24.1 13.2 4.2
1755 10.2 11.2 6.8 6.5 0.0 0.0 8.6 3.2 17.8 23.7 6.8 20.0
1756 12.5 7.1 5.4 9.4 12.5 12.9 3.6 6.4 11.8 14.3 17.0 9.4
1757 14.1 21.2 26.2 30.0 38.1 12.8 25.0 51.3 39.7 32.5 64.7 33.5
1758 37.6 52.0 49.0 72.3 46.4 45.0 44.0 38.7 62.5 37.7 43.0 43.0
1759 48.3 44.0 46.8 47.0 49.0 50.0 51.0 71.3 77.2 59.7 46.3 57.0
1760 67.3 59.5 74.7 58.3 72.0 48.3 66.0 75.6 61.3 50.6 59.7 61.0
1761 70.0 91.0 80.7 71.7 107.2 99.3 94.1 91.1 100.7 88.7 89.7 46.0
1762 43.8 72.8 45.7 60.2 39.9 77.1 33.8 67.7 68.5 69.3 77.8 77.2
1763 56.5 31.9 34.2 32.9 32.7 35.8 54.2 26.5 68.1 46.3 60.9 61.4
1764 59.7 59.7 40.2 34.4 44.3 30.0 30.0 30.0 28.2 28.0 26.0 25.7
1765 24.0 26.0 25.0 22.0 20.2 20.0 27.0 29.7 16.0 14.0 14.0 13.0
1766 12.0 11.0 36.6 6.0 26.8 3.0 3.3 4.0 4.3 5.0 5.7 19.2
1767 27.4 30.0 43.0 32.9 29.8 33.3 21.9 40.8 42.7 44.1 54.7 53.3
1768 53.5 66.1 46.3 42.7 77.7 77.4 52.6 66.8 74.8 77.8 90.6 111.8
1769 73.9 64.2 64.3 96.7 73.6 94.4 118.6 120.3 148.8 158.2 148.1 112.0
1770 104.0 142.5 80.1 51.0 70.1 83.3 109.8 126.3 104.4 103.6 132.2 102.3
1771 36.0 46.2 46.7 64.9 152.7 119.5 67.7 58.5 101.4 90.0 99.7 95.7
1772 100.9 90.8 31.1 92.2 38.0 57.0 77.3 56.2 50.5 78.6 61.3 64.0
1773 54.6 29.0 51.2 32.9 41.1 28.4 27.7 12.7 29.3 26.3 40.9 43.2
1774 46.8 65.4 55.7 43.8 51.3 28.5 17.5 6.6 7.9 14.0 17.7 12.2
```

emacs23@ubuntu

lit Options Buffers Tools Org Tbl Help

```
is the continuation mark. R is asking you for more stuff
WARNING:* R starts counting from 1, *not* zero.
comes with build in datasets
GIN_SRC r
ers
ots
SRC
actices", but the priority is almost
constants (e.g., pi in the
54/
-a" "j" "k" "l" "m" "n" "o"
RC)-----
```

```
> month.abb
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct"
"Nov" "Dec"
There are also built-in data sets for testing, but we'll get to
all these things in more detail later.
> sunspot.month
  Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct
Nov  Dec
1749 58.0 62.6 70.0 55.7 85.0 83.5 94.8 66.3 75.9 75.5
158.6 85.2
1750 73.3 75.9 89.2 88.3 90.0 100.0 85.4 103.0 91.2 65.7
63.3 75.4
1751 70.0 43.5 45.3 56.4 60.7 50.7 66.3 59.8 23.5 23.2
28.5 44.0
```

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
1997 5.7 7.6 8.7 15.5 18.5 12.7 10.4 24.4 51.3 22.8 39.0 41.2
> y = seq(1,4)
> y
[1] 1 2 3 4
> y/3
[1] 0.3333333 0.6666667 1.0000000 1.3333333
> seq(1,3)
[1] 1 2 3
> seq(1,3,by=0.1)
[1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8
[20] 2.9 3.0
> q = 99
> q_vector = c(99)
> q
[1] 99
> q_vector
[1] 99
> q == q_vector
[1] TRUE
> n = c(3,4,100)
> m = c(4,4,4)
> n < m
[1] TRUE FALSE FALSE
> n[2]
[1] 4
> n[2]*m[3]
[1] 16
> n[1:2]
[1] 3 4
> n[1:2] + c(m[3], 5)
[1] 7 9
> help(":")
>

```

emacs23@ubuntu

lit Options Buffers Tools ERC Help

is the continuation mark. R is asking you for more stuff

WARNING:\* R starts counting from 1, \*not\* zero.

comes with build in [datasets](#)

GIN\_SRC r

ers

ots

D\_SRC

:", ">", "<=", ">=", "!=", yield a

54/

ough the examples? [11:38]

on things! [11:39]

RC)



2

```

[1] TRUE TRUE FALSE
So logical tests, like arithmetic operations, are performed
element-wise on vectors.
Vectors, like most R data objects, can be subsetted ("sliced"),
using the [] notation.
> n[2]
[1] 4
> n[2]*m[3]
[1] 16
> n[1:2] + c(m[3], 5) # help(":")
[1] 7 9

```

3

What was that last thing? Don't move on until it makes sense. All R data objects have at least two intrinsic attributes: length and mode.

```
researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
Colon package:base R Documentation
Colon Operator
Description:
  Generate regular sequences.
Usage:
  from:to
  a:b
Arguments:
  from: starting value of sequence.
  to: (maximal) end value of the sequence.
  a, b: 'factor's of the same length.
Details:
  The binary operator ':' has two meanings: for factors 'a:b' is
  equivalent to 'interaction(a, b)' (but the levels are ordered and
  labelled differently).
  For other arguments 'from:to' is equivalent to 'seq(from, to)',
  and generates a sequence from 'from' to 'to' in steps of '1' or
  '-1'. Value 'to' will be included if it differs from 'from' by an
  integer up to a numeric fuzz of about '1e-7'. Non-numeric
  arguments are coerced internally (hence without dispatching
```

```
emacs23@ubuntu
lit Options Buffers Tools ERC Help
is the continuation mark. R is asking you for more stuff
WARNING:* R starts counting from 1, *not* zero.
comes with build in datasets
GIN_SRC r
ers
pots
D_SRC
, ">", "<=", ">=", "!=", yield a
64/
ough the examples? [11:38]
on things! [11:39]
RC)
```



```
[1] TRUE TRUE FALSE
So logical tests, like arithmetic operations, are performed
element-wise on vectors.
Vectors, like most R data objects, can be subsetted ("sliced"),
using the [] notation.
> n[2]
[1] 4
> n[2]*m[3]
[1] 16
> n[1:2] + c(m[3], 5) # help(":")
[1] 7 9
What was that last thing? Don't move on until it makes sense. All R data objects
have at least two intrinsic attributes: length and mode.
```

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
[1] 0.3333333 0.6666667 1.0000000 1.3333333
> seq(1,3)
[1] 1 2 3
> seq(1,3,by=0.1)
[1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8
[20] 2.9 3.0
> q = 99
> q_vector = c(99)
> q
[1] 99
> q_vector
[1] 99
> q == q_vector
[1] TRUE
> n = c(3,4,100)
> m = c(4,4,4)
> n < m
[1] TRUE FALSE FALSE
> n[2]
[1] 4
> n[2]*m[3]
[1] 16
> n[1:2]
[1] 3 4
> n[1:2] + c(m[3], 5)
[1] 7 9
> help(":")
> help(":")
> length(n) # len then press tab and it will complete to length
[1] 3
> mode(m)
[1] "numeric"
> help(m

```

```

emacs23@ubuntu
File Edit Options Buffers Tools ERC Help
[1] 0.3333333 0.6666667 1.0000000 1.3333333
> seq(1,3)
[1] 1 2 3
> seq(1,3,by=0.1)
[1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8
[20] 2.9 3.0
> q = 99
> q_vector = c(99)
> q
[1] 99
> q_vector
[1] 99
> q == q_vector
[1] TRUE
> n = c(3,4,100)
> m = c(4,4,4)
> n < m
[1] TRUE FALSE FALSE
> n[2]
[1] 4
> n[2]*m[3]
[1] 16
> n[1:2]
[1] 3 4
> n[1:2] + c(m[3], 5)
[1] 7 9
> help(":")
> help(":")
> length(n) # len then press tab and it will complete to length
[1] 3
> mode(m)
[1] "numeric"
> help(m

```



data type (numeric, character, logical, or factor). This stuff becomes more important later when the objects and functions we use are more complicated. Briefly, what about other data types?

```

> pals = c('mike', 'steve', 'maria', 'maggie')
> length(pals)
[1] 4
> mode(pals)
[1] "character"
> pals == 'steve'
[1] FALSE TRUE FALSE FALSE
> pals.named.steve = (pals == 'steve')

```

ough the examples? [11:38]  
 on things! [11:39]



```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
[1] 3 4
> n[1:2] + c(m[3], 5)
[1] 7 9
> help(":")
> help(":")
> length(n) # len then press tab and it will complete to length
[1] 3
> mode(m)
[1] "numeric"
> help(mode)
> pals = c('mike', 'steve', 'maria', 'maggie')
> length(pals)
[1] 4
> mode(pals)
[1] "character"
> pals == 'steve'
[1] FALSE TRUE FALSE FALSE
> pals.named.steve = (pals == 'steve')
> pals.named.steve
[1] FALSE TRUE FALSE FALSE
> mode(pals.named.steve)
[1] "logical"
> aww = c(23, 44.45, 2002)
> names(aww) = c('count', 'value', 'year')
> aww
  count  value  year
23.00  44.45 2002.00
> attributes(aww)
$names
[1] "count" "value" "year"

> x = runif(100)
>

```

```

emacs23@ubuntu
File Edit Options Buffers Tools ERC Help
[1] 3 4
> n[1:2] + c(m[3], 5)
[1] 7 9
> help(":")
> help(":")
> length(n) # len then press tab and it will complete to length
[1] 3
> mode(m)
[1] "numeric"
> help(mode)
> pals = c('mike', 'steve', 'maria', 'maggie')
> length(pals)
[1] 4
> mode(pals)
[1] "character"
> pals == 'steve'
[1] FALSE TRUE FALSE FALSE
> pals.named.steve = (pals == 'steve')
> pals.named.steve
[1] FALSE TRUE FALSE FALSE
> mode(pals.named.steve)
[1] "logical"
> aww = c(23, 44.45, 2002)
> names(aww) = c('count', 'value', 'year')
> aww
  count  value  year
23.00  44.45 2002.00
> attributes(aww)
$names
[1] "count" "value" "year"

> x = runif(100)
>

```



worry about that too much right now. The plot() function invokes the simplest, but most commonly used graphics tool in R.

```

> x = runif(100)
> y = x + 0.2*rnorm(100)
> plot(x, y)
> plot(x, y, type='b')
> plot(x, y, type='l', xlab='Width', ylab='Height')

```

This is a good time to introduce the "help" utility. All R functions have a help page, e.g.,

54/

ough the examples? [11:38]

on things! [11:39]

researchtools@ubuntu: ~/class/25

File Edit View Search Terminal Help

```

> mode(pals.named.steve)
[1] "logical"
> aww = c(23, 44.45, 2002)
> names(aww) = c('count', 'value', 'year')
> aww
  count  value  year
23.00  44.45 2002.00
> attributes(aww)
$names
[1] "count" "value" "year"

> x = runif(100)
> x
 [1] 0.677041235 0.686875008 0.038526787 0.956271859 0.257130058 0.789944801
 [7] 0.053271472 0.715333289 0.970328137 0.769217671 0.259168015 0.598175012
[13] 0.090016661 0.991640435 0.509402109 0.655267025 0.098075978 0.352907787
[19] 0.412179362 0.215590553 0.073950448 0.246369214 0.190084386 0.946313610
[25] 0.140764915 0.759393041 0.276611121 0.150390834 0.600697354 0.103166463
[31] 0.494385719 0.033503166 0.343685571 0.823396574 0.076112306 0.928730217
[37] 0.946362088 0.635631301 0.639020732 0.483137194 0.212308027 0.954313842
[43] 0.273770388 0.970644746 0.741593034 0.497131838 0.507275213 0.203678575
[49] 0.993763270 0.723100799 0.303982950 0.857592044 0.581204009 0.375495009
[55] 0.678705782 0.346730035 0.248351591 0.747436913 0.689679162 0.397098238
[61] 0.067711796 0.329052310 0.333775117 0.902886950 0.830327766 0.815373196
[67] 0.817578924 0.756773087 0.422317602 0.121025509 0.799123618 0.256204150
[73] 0.236042877 0.492069317 0.112634549 0.938668118 0.768768107 0.860058185
[79] 0.122262271 0.265091039 0.790928464 0.945571603 0.933986403 0.501144309
[85] 0.233966948 0.127889631 0.032096690 0.990114507 0.480860709 0.961403433
[91] 0.552351892 0.439863029 0.506831459 0.065126597 0.803020234 0.628605130
[97] 0.915688577 0.558593410 0.487703286 0.000202948

> help runif
Error: unexpected symbol in "help runif"
> help (runif

```

emacs23@ubuntu

lit Options Buffers Tools ERC Help

is the continuation mark. R is asking you for more stuff

WARNING: R starts counting from 1, \*not\* zero.

comes with build in **datasets**

SIN\_SRC r

ers

oots

D\_SRC

ates a vector of uniformly

ult settings that can be

54/

ough the examples?

[11:38]

on things!

[11:39]

worry about that too much right now. The plot() function invokes the simplest, but most commonly used graphics tool in R.

```

> x = runif(100)
> y = x + 0.2*rnorm(100)
> plot(x, y)
> plot(x, y, type='b')
> plot(x, y, type='l', xlab='Width', ylab='Height')

```

This is a good time to introduce the "help" utility. All R functions have a help page, e.g.,

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
Uniform package:stats R Documentation
The Uniform Distribution
Description:
These functions provide information about the uniform distribution
on the interval from 'min' to 'max'. 'dunif' gives the density,
'punif' gives the distribution function 'qunif' gives the quantile
function and 'runif' generates random deviates.
Usage:
dunif(x, min=0, max=1, log = FALSE)
punif(q, min=0, max=1, lower.tail = TRUE, log.p = FALSE)
qunif(p, min=0, max=1, lower.tail = TRUE, log.p = FALSE)
runif(n, min=0, max=1)
Arguments:
x,q: vector of quantiles.
p: vector of probabilities.
n: number of observations. If 'length(n) > 1', the length is
taken to be the number required.
min,max: lower and upper limits of the distribution. Must be finite.
log, log.p: logical; if TRUE, probabilities p are given as log(p).
lower.tail: logical; if TRUE (default), probabilities are P[X <= x],
:

```

```

emacs23@ubuntu
lit Options Buffers Tools ERC Help
is the continuation mark. R is asking you for more stuff
WARNING:* R starts counting from 1, *not* zero.
comes with build in datasets
BIN_SRC r
ers
ots
D_SRC
ates a vector of uniformly
ult settings that can be
54/
ough the examples? [11:38]
on things! [11:39]
RC)

```



```

worry about that too much right now. The plot() function invokes the simplest,
but most
commonly used graphics tool in R.
> x = runif(100)
> y = x + 0.2*rnorm(100)
> plot(x, y)
> plot(x, y, type='b')
> plot(x, y, type='l', xlab='Width', ylab='Height')
This is a good time to introduce the "help" utility. All R functions have a help
page, e.g.,

```

researchtools@ubuntu: ~/class/25

File Edit View Search Terminal Help

```

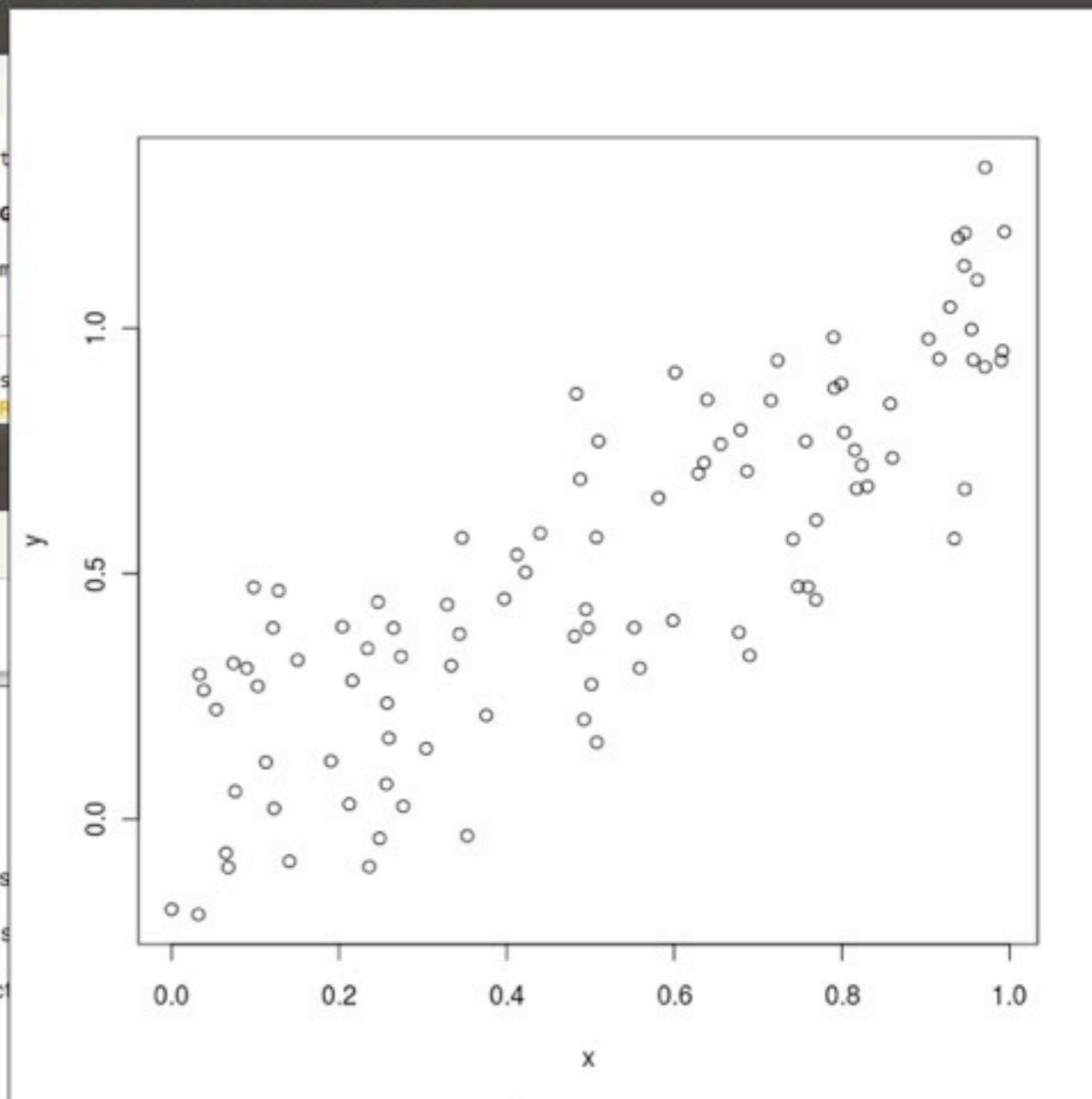
> names(aww) = c('count','value','year')
> aww
  count  value  year
23.00  44.45 2002.00
> attributes(aww)
$names
[1] "count" "value" "year"

> x = runif(100)
> x
 [1] 0.677041235 0.686875008 0.038526787 0.956271859 0.257130058 0.789944801
 [7] 0.053271472 0.715333289 0.970328137 0.769217671 0.259168015 0.598175012
[13] 0.090016661 0.991640435 0.509402109 0.655267025 0.098075978 0.352907787
[19] 0.412179362 0.215590553 0.073950448 0.246369214 0.190084386 0.946313610
[25] 0.140764915 0.759393041 0.276611121 0.150390834 0.600697354 0.103166463
[31] 0.494385719 0.033503166 0.343685571 0.823396574 0.076112306 0.928730217
[37] 0.946362088 0.635631301 0.639020732 0.483137194 0.212308027 0.954313842
[43] 0.273770388 0.970644746 0.741593034 0.497131838 0.507275213 0.203678575
[49] 0.993763270 0.723100799 0.303982950 0.857592044 0.581204009 0.375495009
[55] 0.678705782 0.346730035 0.248351591 0.747436913 0.689679162 0.397098238
[61] 0.067711796 0.329052310 0.333775117 0.902886950 0.830327766 0.815373196
[67] 0.817578924 0.756773087 0.422317602 0.121025509 0.799123618 0.256204150
[73] 0.236042877 0.492069317 0.112634549 0.938668118 0.768768107 0.860058185
[79] 0.122262271 0.265091039 0.790928464 0.945571603 0.933986403 0.501144309
[85] 0.233966948 0.127889631 0.032096690 0.990114507 0.480860709 0.961403433
[91] 0.552351892 0.439863029 0.506831459 0.065126597 0.803020234 0.628605130
[97] 0.915688577 0.558593410 0.487703286 0.000202948

> help runif
Error: unexpected symbol in "help runif"
> help (runif)
> y = x + 0.2*rnorm(100)
> plot(x,y)
> plot(x,y, type='b')

```

R Graphics: Device 2 (ACTIVE)



worry about that too much right now. The plot() function is the most commonly used graphics tool in R.

```

> x = runif(100)
> y = x + 0.2*rnorm(100)
> plot(x, y)
> plot(x, y, type='b')
> plot(x, y, type='l', xlab='Width', ylab='Height')

```

This is a good time to introduce the "help" utility. All R functions have a help page, e.g.,

researchtools@ubuntu: ~/class/25

File Edit View Search Terminal Help

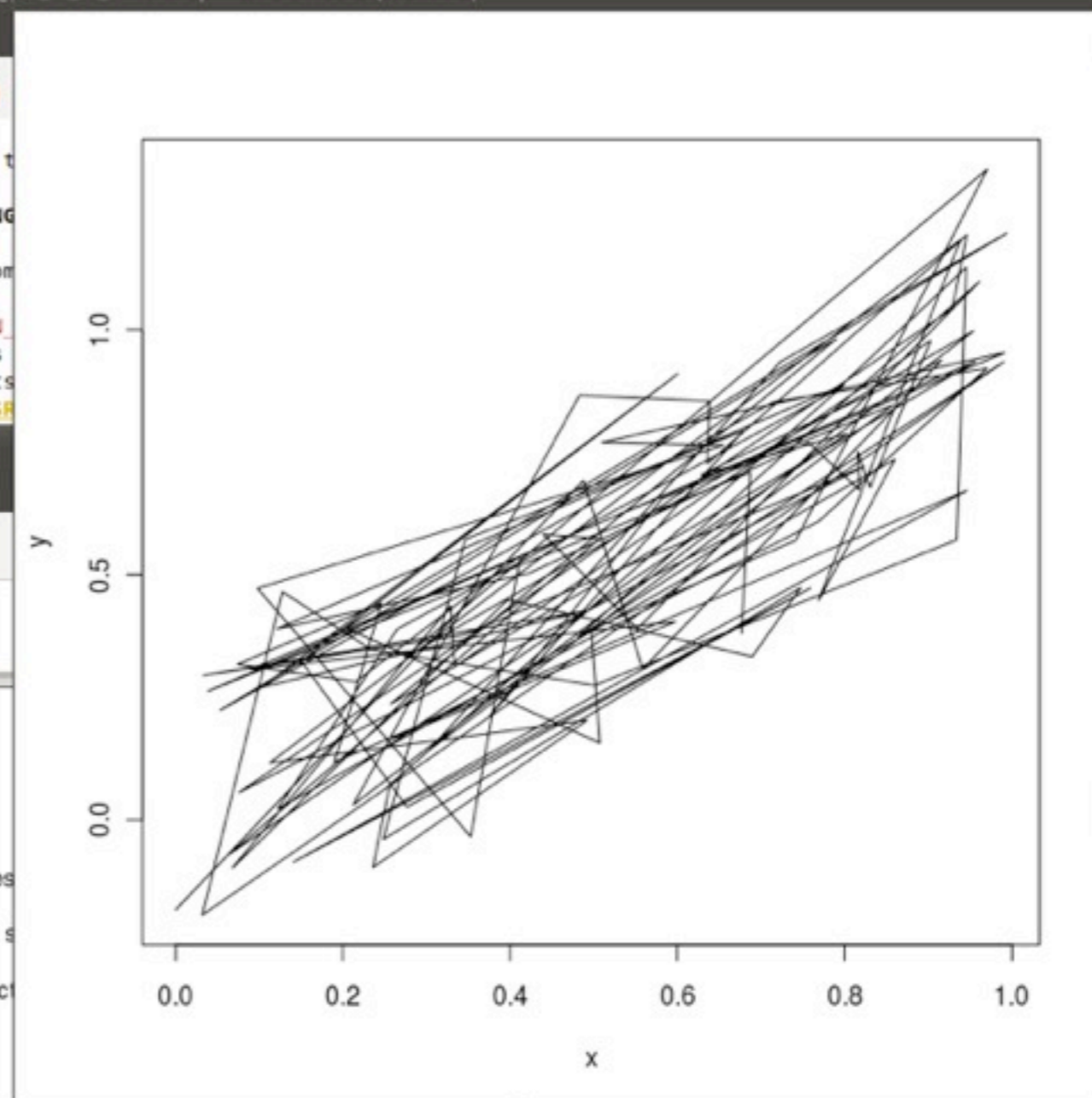
```
count value year
23.00 44.45 2002.00
> attributes(aww)
$names
[1] "count" "value" "year"
> x = runif(100)
> x
 [1] 0.677041235 0.686875008 0.038526787 0.956271859 0.257130058 0.789944801
 [7] 0.053271472 0.715333289 0.970328137 0.769217671 0.259168015 0.598175012
[13] 0.090016661 0.991640435 0.509402109 0.655267025 0.098075978 0.352907787
[19] 0.412179362 0.215590553 0.073950448 0.246369214 0.190084386 0.946313610
[25] 0.140764915 0.759393041 0.276611121 0.150390834 0.600697354 0.103166463
[31] 0.494385719 0.033503166 0.343685571 0.823396574 0.076112306 0.928730217
[37] 0.946362088 0.635631301 0.639020732 0.483137194 0.212308027 0.954313842
[43] 0.273770388 0.970644746 0.741593034 0.497131838 0.507275213 0.203678575
[49] 0.993763270 0.723100799 0.303982950 0.857592044 0.581204009 0.375495009
[55] 0.678705782 0.346730035 0.248351591 0.747436913 0.689679162 0.397098238
[61] 0.067711796 0.329052310 0.333775117 0.902886950 0.830327766 0.815373196
[67] 0.817578924 0.756773087 0.422317602 0.121025509 0.799123618 0.256204150
[73] 0.236042877 0.492069317 0.112634549 0.938668118 0.768768107 0.860058185
[79] 0.122262271 0.265091039 0.790928464 0.945571603 0.933986403 0.501144309
[85] 0.233966948 0.127809631 0.032096690 0.990114507 0.480860709 0.961403433
[91] 0.552351892 0.439863029 0.506831459 0.065126597 0.803020234 0.628605130
[97] 0.915688577 0.558593410 0.487703286 0.000202948
> help runif
Error: unexpected symbol in "help runif"
> help (runif)
> y = x + 0.2*rnorm(100)
> plot(x,y)
> plot(x,y, type='b')
> plot(x,y, type='l')
> plot(x,y, type='l', xlab='width', ylab='height')
```

worry about that too much right now. The plot() function is the most commonly used graphics tool in R.

```
> x = runif(100)
> y = x + 0.2*rnorm(100)
> plot(x, y)
> plot(x, y, type='b')
> plot(x, y, type='l', xlab='width', ylab='height')
```

This is a good time to introduce the "help" utility. All R functions have a help page, e.g.,

R Graphics: Device 2 (ACTIVE)



RC)

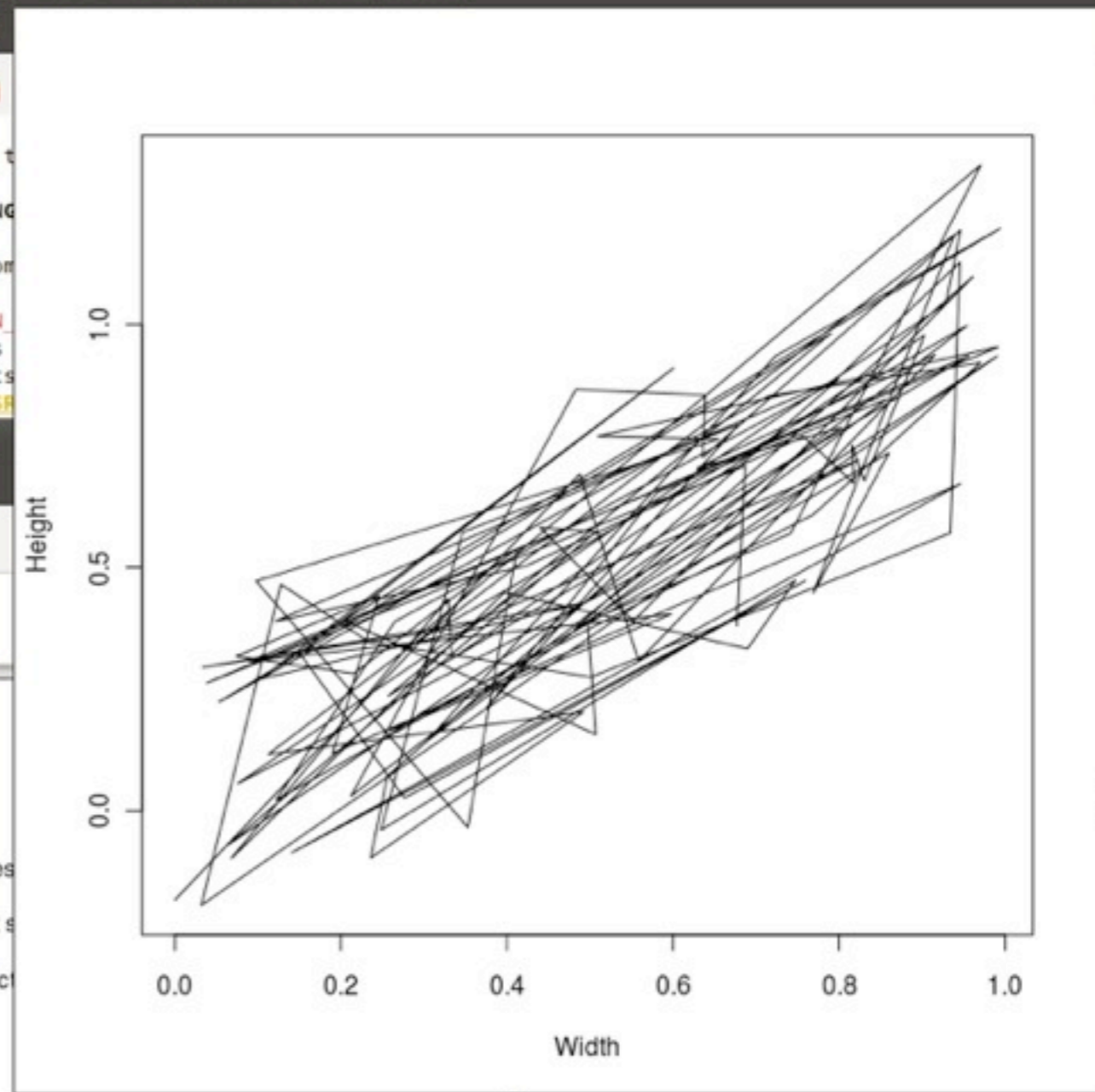
```

23.00 44.45 2002.00
> attributes(aww)
$names
[1] "count" "value" "year"

> x = runif(100)
> x
 [1] 0.677041235 0.686875008 0.038526787 0.956271859 0.257130058 0.789944801
 [7] 0.053271472 0.715333289 0.970328137 0.769217671 0.259168015 0.598175012
[13] 0.090016661 0.991640435 0.509402109 0.655267025 0.098075978 0.352907787
[19] 0.412179362 0.215590553 0.073950448 0.246369214 0.190084386 0.946313610
[25] 0.140764915 0.759393041 0.276611121 0.150390834 0.600697354 0.103166463
[31] 0.494385719 0.033503166 0.343685571 0.823396574 0.076112306 0.928730217
[37] 0.946362088 0.635631301 0.639020732 0.483137194 0.212308027 0.954313842
[43] 0.273770388 0.970644746 0.741593034 0.497131838 0.507275213 0.203678575
[49] 0.993763270 0.723100799 0.303982950 0.857592044 0.581204009 0.375495009
[55] 0.678705782 0.346730035 0.248351591 0.747436913 0.689679162 0.397098238
[61] 0.067711796 0.329052310 0.333775117 0.902886950 0.830327766 0.815373196
[67] 0.817578924 0.756773087 0.422317602 0.121025509 0.799123618 0.256204150
[73] 0.236042877 0.492069317 0.112634549 0.938668118 0.768768107 0.860058185
[79] 0.122262271 0.265091039 0.790928464 0.945571603 0.933986403 0.501144309
[85] 0.233966948 0.127889631 0.032096690 0.990114507 0.480860709 0.961403433
[91] 0.552351892 0.439863029 0.506831459 0.065126597 0.803020234 0.628605130
[97] 0.915688577 0.558593410 0.487703286 0.000202948

> help runif
Error: unexpected symbol in "help runif"
> help (runif)
> y = x + 0.2*rnorm(100)
> plot(x,y)
> plot(x,y, type='b')
> plot(x,y, type='l')
> plot(x,y, type='l', xlab='Width', ylab='Height')

```



worry about that too much right now. The plot() function is the most commonly used graphics tool in R.

```

> x = runif(100)
> y = x + 0.2*rnorm(100)
> plot(x, y)
> plot(x, y, type='b')
> plot(x, y, type='l', xlab='Width', ylab='Height')

```

This is a good time to introduce the "help" utility. All R functions have a help page, e.g.,



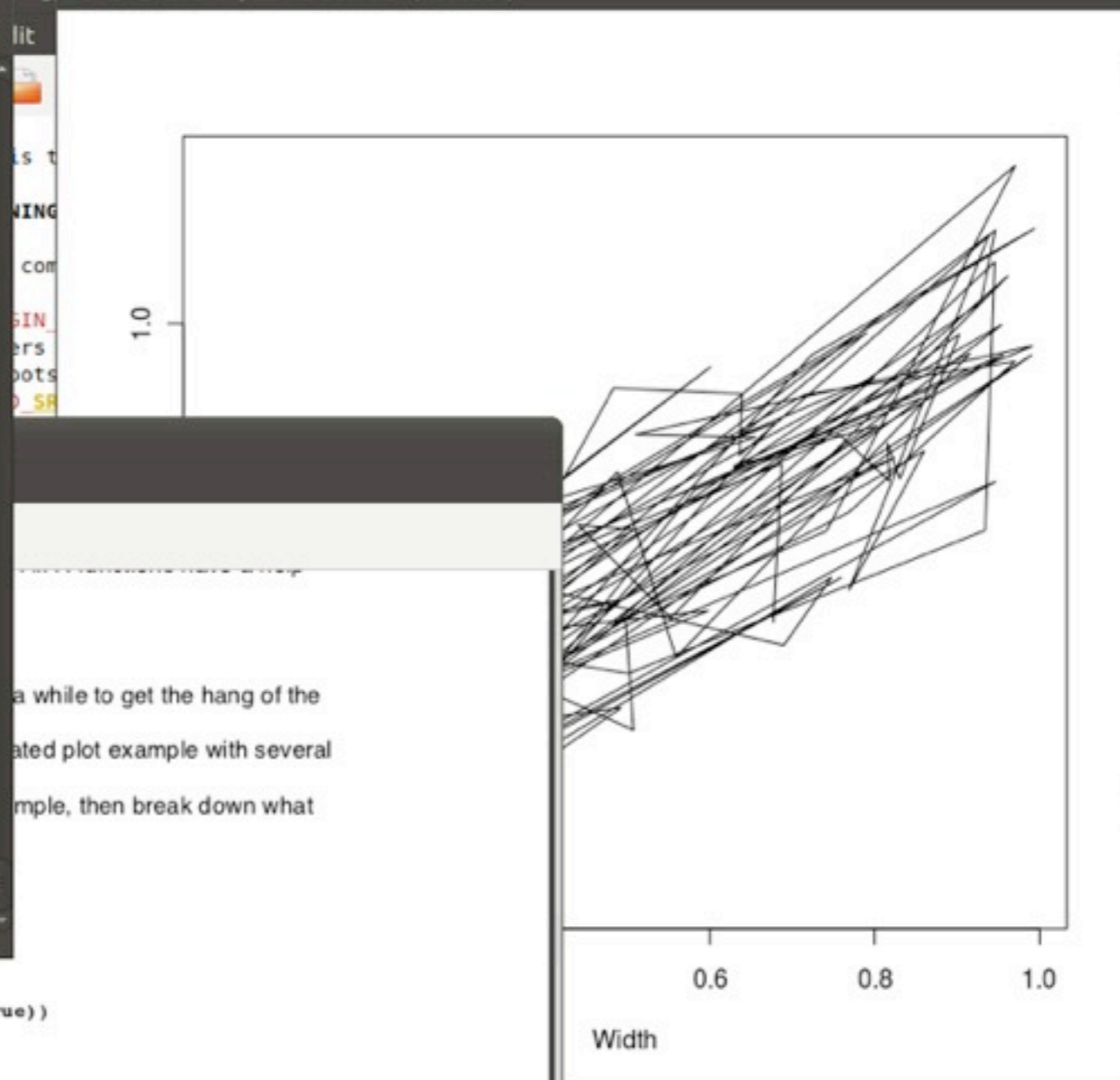
researchtools@ubuntu: ~/class/25

File Edit View Search Terminal Help

```
> plot(x,y, type='l', xlab='Width', ylab='Height')
> help(plot)
> x = seq(0, 2*pi, by=pi/100)
Error in seq.default(0, 2 * pi, by = pi/100) : object 'pi' not found
> x = seq(0, 2*pi, by=py/100)
Error in seq.default(0, 2 * pi, by = py/100) : object 'py' not found
> x = seq(0, 2*pi, by=pi/100)
> y.true = sin(x)
> y.obs = y.true + rnorm(length(y.true))
Error in rnorm(length(y.true)) : could not find function "length"
> y.obs = y.true + rnorm(length(y.true))
> y
 [1] 0.38070405 0.70888962 0.26294848 0.93555879 0.23653209 0.98141344
 [7] 0.22387453 0.85294810 0.92143502 0.60954259 0.16528519 0.40504211
[13] 0.30759486 0.95356759 0.77035950 0.76423660 0.47279298 -0.03375378
[19] 0.53919154 0.28237485 0.31779874 0.44245689 0.11865432 0.67254863
[25] -0.08576078 0.47330848 0.02666198 0.32440023 0.91024358 0.27136509
[31] 0.42767611 0.29522879 0.37724780 0.72155036 0.05652227 1.04286504
[37] 1.19401811 0.72647059 0.85447750 0.86676835 0.03062600 0.99766974
[43] 0.33133839 1.32746554 0.57030640 0.38954883 0.15711842 0.39196118
[49] 1.19686059 0.93438992 0.14405303 0.84631178 0.65502037 0.21170073
[55] 0.79344456 0.57308377 -0.03905062 0.47399737 0.33357678 0.44934856
[61] -0.09815956 0.43713004 0.31309569 0.97812095 0.67841770 0.75187395
[67] 0.67360936 0.77008910 0.50299565 0.38971149 0.88743474 0.07199642
[73] -0.09656263 0.20341761 0.11653821 1.18388344 0.44747044 0.73582272
[79] 0.02241823 0.38938948 0.87819312 1.12761336 0.57130928 0.27501661
[85] 0.34760132 0.46575411 -0.19376734 0.93431750 0.37287705 1.09851433
[91] 0.39029975 0.58280824 0.57384136 -0.06849502 0.78833032 0.70441729
[97] 0.93772624 0.30803622 0.69246134 -0.18310708
> y.smoo = smooth.split(x,y.obs)
Error: could not find function "smooth.split"
> y.smoo = smooth.spline(x,y.obs)
> y.low = lowess(x, y.obs)
```

```
> y.true = sin(x)
# 2
> y.obs = y.true + rnorm(length(y.true))
# 3
> y.smoo = smooth.spline(x, y.obs)
# 4
> y.low = lowess(x, y.obs)
# 5
> plot(x, y.obs, pch=16, cex=0.6, col='darkgray', ylab='y')
# 6
> lines(x, y.true, col='pink', lwd=3)
# 7
> lines(y.smoo$x, y.smoo$y, lwd=2, col='purple')
# 8
> lines(x, y.low, lwd=2, col='green')
```

R Graphics: Device 2 (ACTIVE)

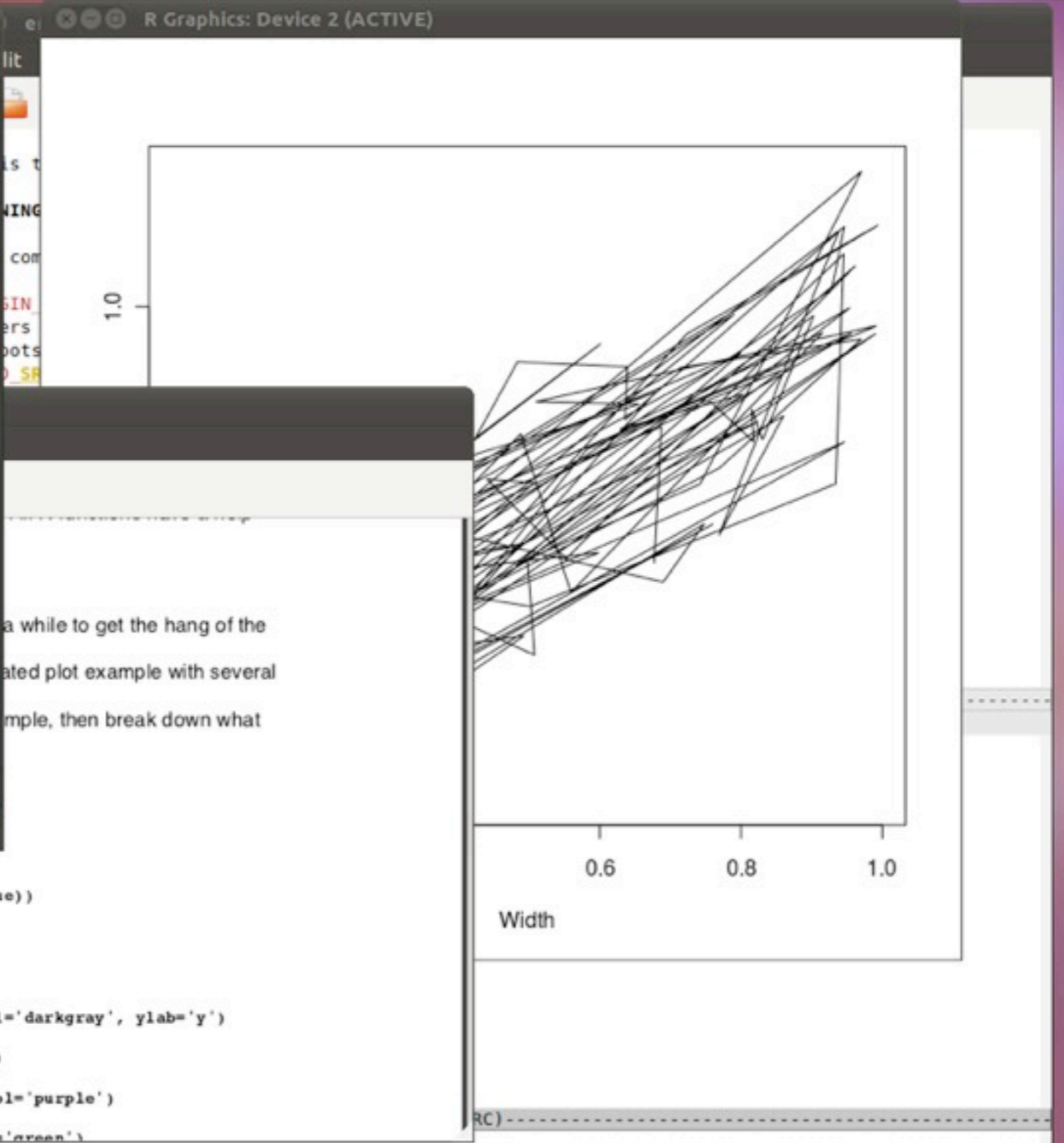


a while to get the hang of the  
 ated plot example with several  
 mple, then break down what

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
Error in seq.default(0, 2 * pi, by = pi/100) : object 'pi' not found
> x = seq(0, 2*pi, by=pi/100)
Error in seq.default(0, 2 * pi, by = pi/100) : object 'pi' not found
> x = seq(0, 2*pi, by=pi/100)
> y.true = sin(x)
> y.obs = y.true + rnorm(length(y.true))
Error in rnorm(length(y.true)) : could not find function "length"
> y.obs = y.true + rnorm(length(y.true))
> y
 [1] 0.38070405 0.70888962 0.26294848 0.93555879 0.23653209 0.98141344
 [7] 0.22387453 0.85294810 0.92143502 0.60954259 0.16528519 0.40504211
[13] 0.30759486 0.95356759 0.77035950 0.76423660 0.47279298 -0.03375378
[19] 0.53919154 0.28237485 0.31779874 0.44245689 0.11865432 0.67254863
[25] -0.08576078 0.47330848 0.02666198 0.32440023 0.91024358 0.27136509
[31] 0.42767611 0.29522879 0.37724780 0.72155036 0.05652227 1.04286504
[37] 1.19401811 0.72647059 0.85447750 0.86676835 0.03062600 0.99766974
[43] 0.33133839 1.32746554 0.57030640 0.38954883 0.15711842 0.39196118
[49] 1.19686059 0.93438992 0.14405303 0.84631178 0.65502037 0.21170073
[55] 0.79344456 0.57308377 -0.03905062 0.47399737 0.33357678 0.44934856
[61] -0.09815956 0.43713004 0.31309569 0.97812095 0.67841770 0.75187395
[67] 0.67360936 0.77008910 0.50299565 0.38971149 0.88743474 0.07199642
[73] -0.09656263 0.20341761 0.11653821 1.18388344 0.44747044 0.73582272
[79] 0.02241823 0.38938948 0.87819312 1.12761336 0.57130928 0.27501661
[85] 0.34760132 0.46575411 -0.19376734 0.93431750 0.37287705 1.09851433
[91] 0.39029975 0.58280824 0.57384136 -0.06849502 0.78833032 0.70441729
[97] 0.93772624 0.30803622 0.69246134 -0.18310708
> y.smoo = smooth.split(x,y.obs)
Error: could not find function "smooth.split"
> y.smoo = smooth.spline(x,y.obs)
> y.low = lowesss(x, y.obs)
Error: could not find function "lowesss"
> y.low = lowess(x, y.obs)
> help (lowess)

```



```

> y.true = sin(x)
# 2
> y.obs = y.true + rnorm(length(y.true))
# 3
> y.smoo = smooth.spline(x, y.obs)
# 4
> y.low = lowess(x, y.obs)
# 5
> plot(x, y.obs, pch=16, cex=0.6, col='darkgray', ylab='y')
# 6
> lines(x, y.true, col='pink', lwd=3)
# 7
> lines(y.smoo$x, y.smoo$y, lwd=2, col='purple')
# 8
> lines(x, y.low, lwd=2, col='green')

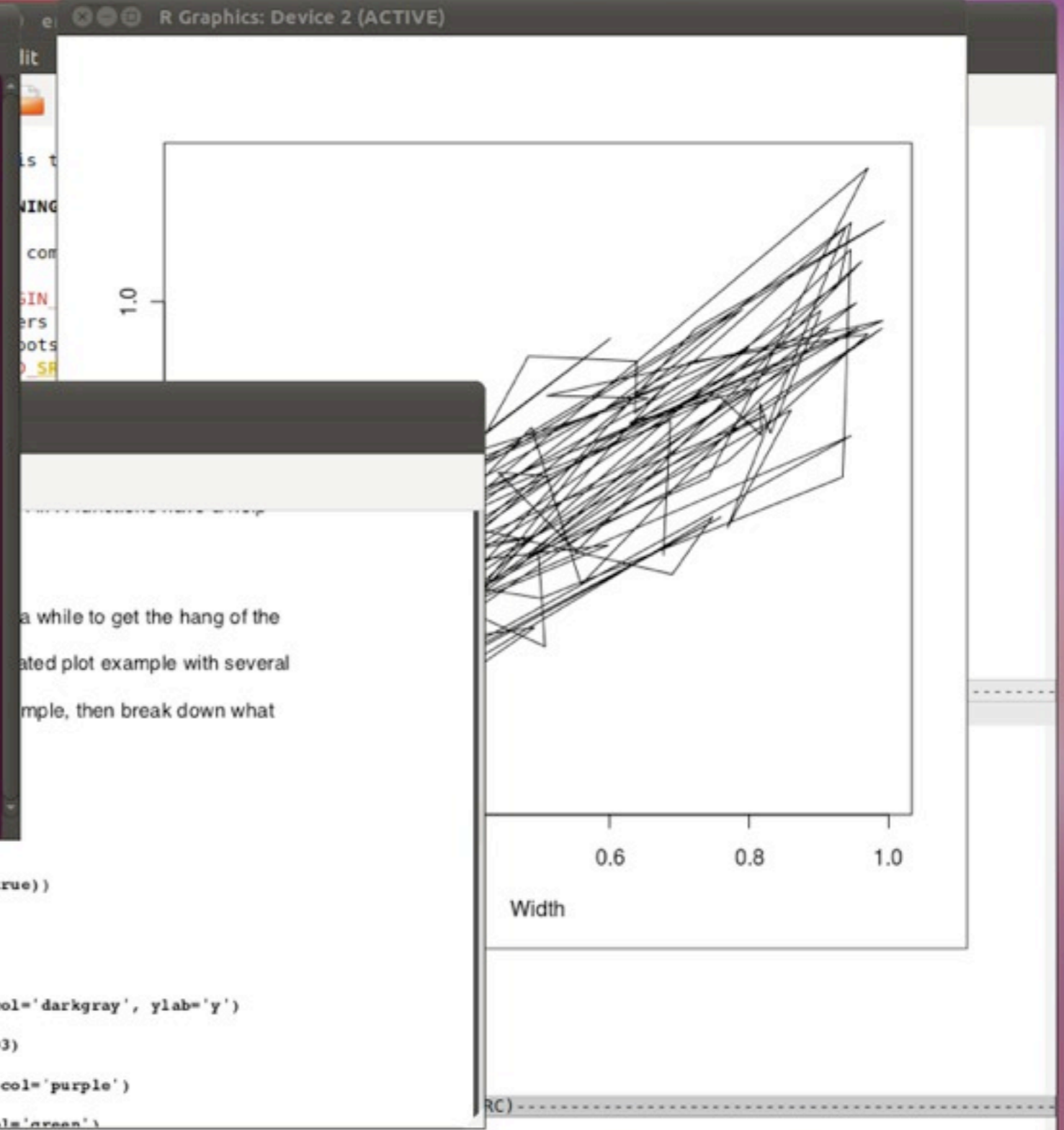
```



```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
lowess package:stats R Documentation
Scatter Plot Smoothing
Description:
This function performs the computations for the LOWESS smoother
which uses locally-weighted polynomial regression (see the
references).
Usage:
lowess(x, y = NULL, f = 2/3, iter = 3,
       delta = 0.01 * diff(range(xy$x[o])))
Arguments:
x, y: vectors giving the coordinates of the points in the scatter
plot. Alternatively a single plotting structure can be
specified - see 'xy.coords'.
f: the smoother span. This gives the proportion of points in the
plot which influence the smooth at each value. Larger values
give more smoothness.
iter: the number of 'robustifying' iterations which should be
performed. Using smaller values of 'iter' will make 'lowess'
run faster.
delta: See 'Details'. Defaults to 1/100th of the range of 'x'.
Details:
:

```



```

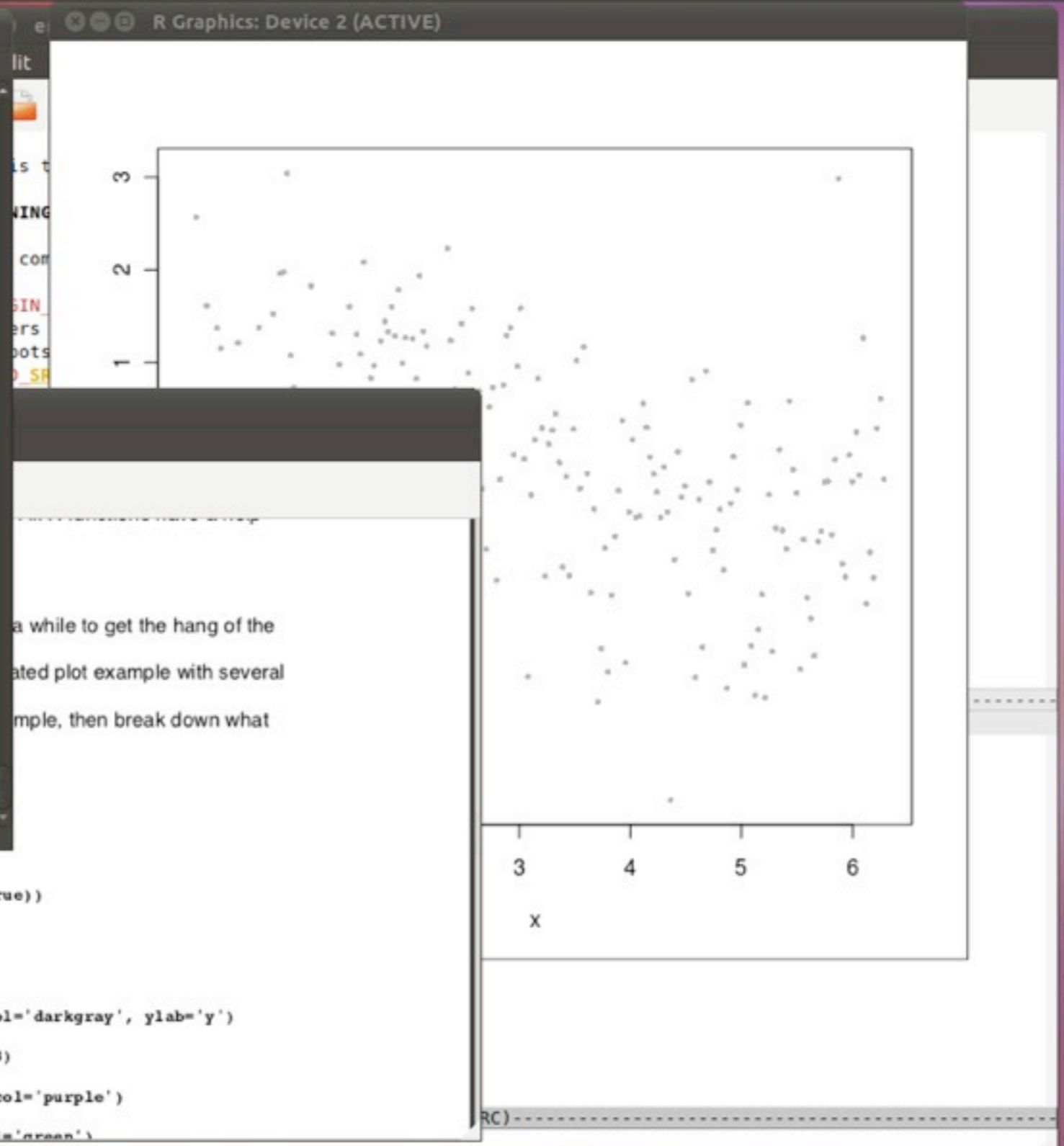
> y.true = sin(x)
# 2
> y.obs = y.true + rnorm(length(y.true))
# 3
> y.smoo = smooth.spline(x, y.obs)
# 4
> y.low = lowess(x, y.obs)
# 5
> plot(x, y.obs, pch=16, cex=0.6, col='darkgray', ylab='y')
# 6
> lines(x, y.true, col='pink', lwd=3)
# 7
> lines(y.smoo$x, y.smoo$y, lwd=2, col='purple')
# 8
> lines(x, y.low$x, y.low$y, lwd=2, col='green')

```

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
Error in seq.default(0, 2 * pi, by = pi/100) : object 'pi' not found
> x = seq(0, 2*pi, by=pi/100)
> y.true = sin(x)
> y.obs = y.true + rnorm(length(y.true))
Error in rnorm(length(y.true)) : could not find function "length"
> y.obs = y.true + rnorm(length(y.true))
> y
 [1] 0.38070405 0.70888962 0.26294848 0.93555879 0.23653209 0.98141344
 [7] 0.22387453 0.85294810 0.92143502 0.60954259 0.16528519 0.40504211
[13] 0.30759486 0.95356759 0.77035950 0.76423660 0.47279298 -0.03375378
[19] 0.53919154 0.28237485 0.31779874 0.44245689 0.11865432 0.67254863
[25] -0.08576078 0.47330848 0.02666198 0.32440023 0.91024358 0.27136509
[31] 0.42767611 0.29522879 0.37724780 0.72155036 0.05652227 1.04286504
[37] 1.19401811 0.72647059 0.85447750 0.86676835 0.03062600 0.99766974
[43] 0.33133839 1.32746554 0.57030640 0.38954883 0.15711842 0.39196118
[49] 1.19686059 0.93438992 0.14405303 0.84631178 0.65502037 0.21170073
[55] 0.79344456 0.57308377 -0.03905062 0.47399737 0.33357678 0.44934856
[61] -0.09815956 0.43713004 0.31309569 0.97812095 0.67841770 0.75187395
[67] 0.67360936 0.77008910 0.50299565 0.38971149 0.88743474 0.07199642
[73] -0.09656263 0.20341761 0.11653821 1.18388344 0.44747044 0.73582272
[79] 0.02241823 0.38938948 0.87819312 1.12761336 0.57130928 0.27501661
[85] 0.34760132 0.46575411 -0.19376734 0.93431750 0.37287705 1.09851433
[91] 0.39029975 0.58280824 0.57384136 -0.06849502 0.78833032 0.70441729
[97] 0.93772624 0.30803622 0.69246134 -0.18310708
> y.smoo = smooth.split(x,y.obs)
Error: could not find function "smooth.split"
> y.smoo = smooth.spline(x,y.obs)
> y.low = lowesss(x, y.obs)
Error: could not find function "lowesss"
> y.low = lowess(x, y.obs)
> help(lowess)
> plot(x,y.obs, pch=16, cex=0.6, col='darkgray', ylab='y')
> lines

```



```

> y.true = sin(x)
# 2
> y.obs = y.true + rnorm(length(y.true))
# 3
> y.smoo = smooth.spline(x, y.obs)
# 4
> y.low = lowess(x, y.obs)
# 5
> plot(x, y.obs, pch=16, cex=0.6, col='darkgray', ylab='y')
# 6
> lines(x, y.true, col='pink', lwd=3)
# 7
> lines(y.smoo$x, y.smoo$y, lwd=2, col='purple')
# 8
> lines(x, y.low, lwd=2, col='green')

```

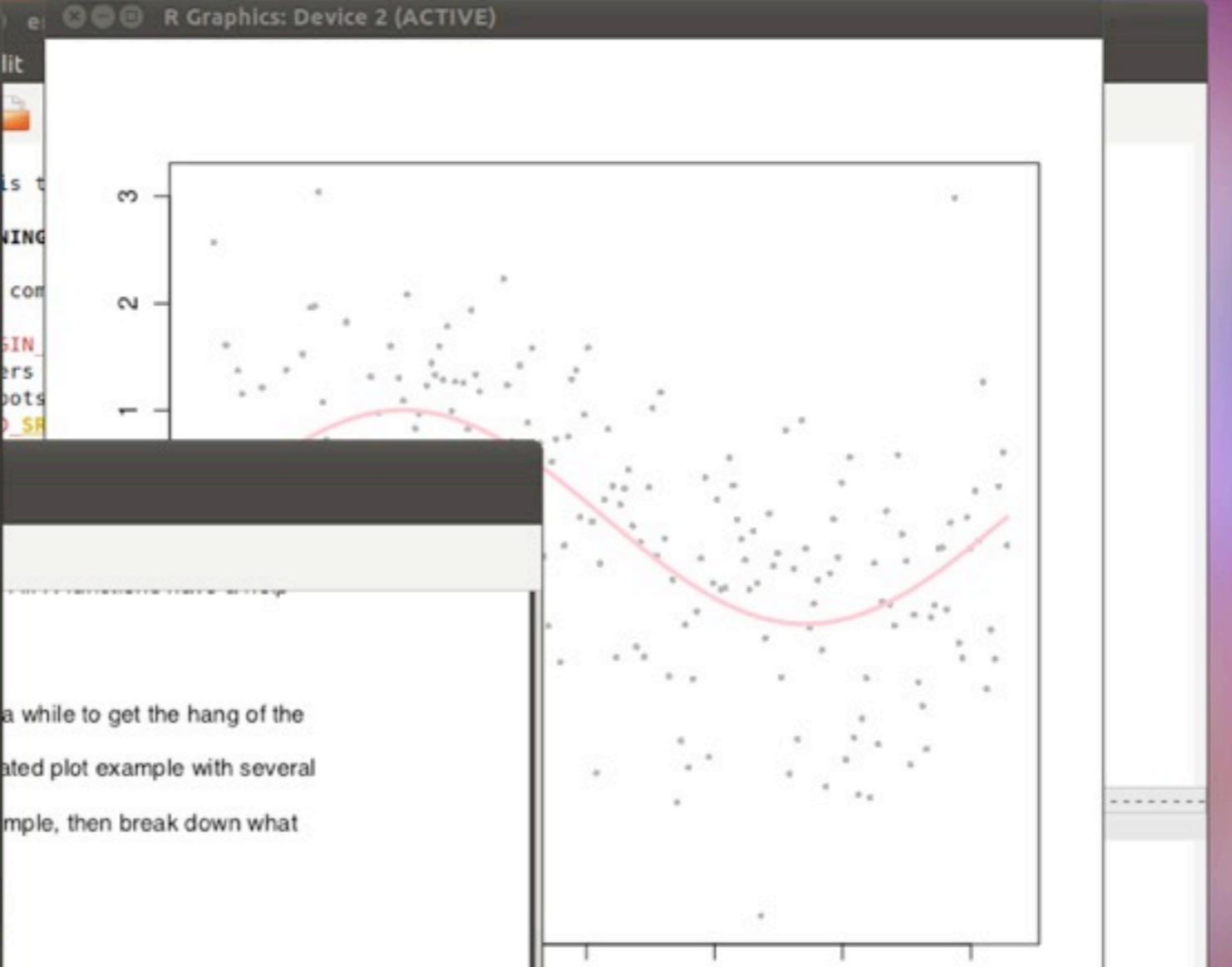
4

5

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
Error in rnorm(lenght(y.true)) : could not find function "lenght"
> y.obs = y.true + rnorm(length(y.true))
> y
 [1] 0.38070405 0.70888962 0.26294848 0.93555879 0.23653209 0.98141344
 [7] 0.22387453 0.85294810 0.92143502 0.60954259 0.16528519 0.40504211
[13] 0.30759486 0.95356759 0.77035950 0.76423660 0.47279298 -0.03375378
[19] 0.53919154 0.28237485 0.31779874 0.44245689 0.11865432 0.67254863
[25] -0.08576078 0.47330848 0.02666198 0.32440023 0.91024358 0.27136509
[31] 0.42767611 0.29522879 0.37724780 0.72155036 0.05652227 1.04286504
[37] 1.19401811 0.72647059 0.85447750 0.86676835 0.03062600 0.99766974
[43] 0.33133839 1.32746554 0.57030640 0.38954883 0.15711842 0.39196118
[49] 1.19686059 0.93438992 0.14405303 0.84631178 0.65502037 0.21170073
[55] 0.79344456 0.57308377 -0.03905062 0.47399737 0.33357678 0.44934856
[61] -0.09815956 0.43713004 0.31309569 0.97812095 0.67841770 0.75187395
[67] 0.67360936 0.77008910 0.50299565 0.38971149 0.88743474 0.07199642
[73] -0.09656263 0.20341761 0.11653821 1.18388344 0.44747044 0.73582272
[79] 0.02241823 0.38938948 0.87819312 1.12761336 0.57130928 0.27501661
[85] 0.34760132 0.46575411 -0.19376734 0.93431750 0.37287705 1.09851433
[91] 0.39029975 0.58280824 0.57384136 -0.06849502 0.78833032 0.70441729
[97] 0.93772624 0.30803622 0.69246134 -0.18310708
> y.smoo = smooth.split(x,y.obs)
Error: could not find function "smooth.split"
> y.smoo = smooth.spline(x,y.obs)
> y.low = lowesss(x, y.obs)
Error: could not find function "lowesss"
> y.low = lowess(x, y.obs)
> help(lowess)
> plot(x,y.obs, pch=16, cex=0.6, col='darkgray', ylab='y')
> lines(x, y.true, col='pink', lwd=3)
Error in plot.xy(xy.coords(x, y), type = type, ...) :
  invalid color name 'pink'
> lines(x, y.true, col='pink', lwd=3)
>

```



```

> y.true = sin(x)
# 2
> y.obs = y.true + rnorm(length(y.true))
# 3
> y.smoo = smooth.spline(x, y.obs)
# 4
> y.low = lowess(x, y.obs)
# 5
> plot(x, y.obs, pch=16, cex=0.6, col='darkgray', ylab='y')
# 6
> lines(x, y.true, col='pink', lwd=3)
# 7
> lines(y.smoo$x, y.smoo$y, lwd=2, col='purple')
# 8
> lines(x, y.low, col='green', lwd=2)

```

researchtools@ubuntu: ~/class/25

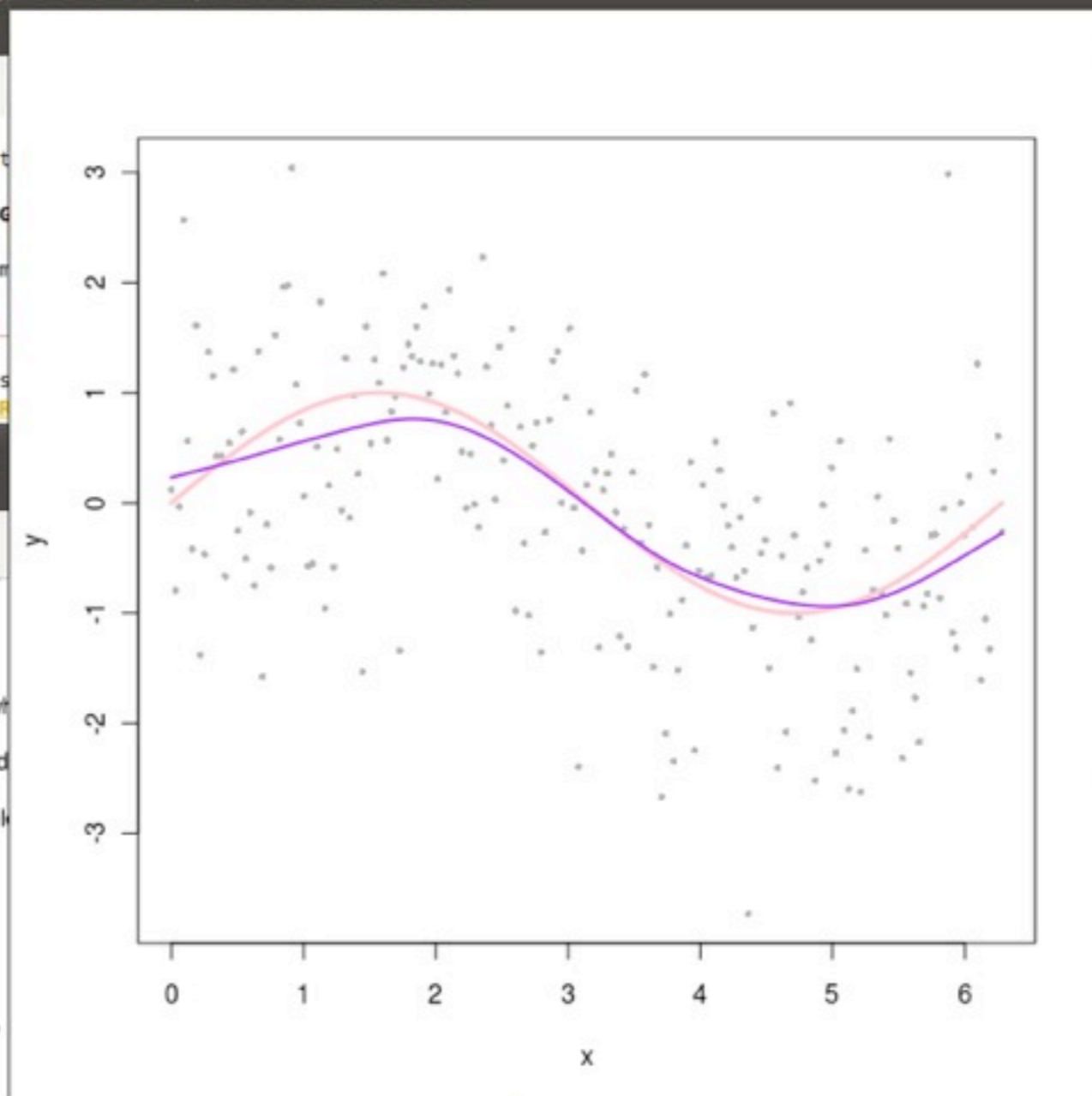
File Edit View Search Terminal Help

```

> y.obs = y.true + rnorm(length(y.true))
> y
 [1]  0.38070405  0.70888962  0.26294848  0.93555879  0.23653209  0.98141344
 [7]  0.22387453  0.85294810  0.92143502  0.60954259  0.16528519  0.40504211
[13]  0.30759486  0.95356759  0.77035950  0.76423660  0.47279298 -0.03375378
[19]  0.53919154  0.28237485  0.31779874  0.44245689  0.11865432  0.67254863
[25] -0.08576078  0.47330848  0.02666198  0.32440023  0.91024358  0.27136509
[31]  0.42767611  0.29522879  0.37724780  0.72155036  0.05652227  1.04286504
[37]  1.19401811  0.72647059  0.85447750  0.86676835  0.03062600  0.99766974
[43]  0.33133839  1.32746554  0.57030640  0.38954883  0.15711842  0.39196118
[49]  1.19686059  0.93438992  0.14405303  0.84631178  0.65502037  0.21170073
[55]  0.79344456  0.57308377 -0.03905062  0.47399737  0.33357678  0.44934856
[61] -0.09815956  0.43713004  0.31309569  0.97812095  0.67841770  0.75187395
[67]  0.67360936  0.77008910  0.50299565  0.38971149  0.88743474  0.07199642
[73] -0.09656263  0.20341761  0.11653821  1.18388344  0.44747044  0.73582272
[79]  0.02241823  0.38938948  0.87819312  1.12761336  0.57130928  0.27501661
[85]  0.34760132  0.46575411 -0.19376734  0.93431750  0.37287705  1.09851433
[91]  0.39029975  0.58280824  0.57384136 -0.06849502  0.78833032  0.70441729
[97]  0.93772624  0.30803622  0.69246134 -0.18310708
> y.smoo = smooth.split(x,y.obs)
Error: could not find function "smooth.split"
> y.smoo = smooth.spline(x,y.obs)
> y.low = lowesss(x, y.obs)
Error: could not find function "lowesss"
> y.low = lowess(x, y.obs)
> help(lowess)
> plot(x,y.obs, pch=16, cex=0.6, col='darkgray', ylab='y')
> lines(x, y.true, col='pink', lwd=3)
Error in plot.xy(xy.coords(x, y), type = type, ...) :
  invalid color name 'pink'
> lines(x, y.true, col='pink', lwd=3)
> lines(y.smoo$x, y.smoo$y, lwd=2, col='purple')
>

```

R Graphics: Device 2 (ACTIVE)



```

> y.true = sin(x)
# 2
> y.obs = y.true + rnorm(length(y.true))
# 3
> y.smoo = smooth.spline(x, y.obs)
# 4
> y.low = lowess(x, y.obs)
# 5
> plot(x, y.obs, pch=16, cex=0.6, col='darkgray', ylab='y')
# 6
> lines(x, y.true, col='pink', lwd=3)
# 7
> lines(y.smoo$x, y.smoo$y, lwd=2, col='purple')
# 8
> lines(x, y.low, col='green', lwd=2)

```

researchtools@ubunt...

emacs23@ubuntu

[researchtools@ubun...

R-lab 1-Intro

R Graphics: Device 2 (...)

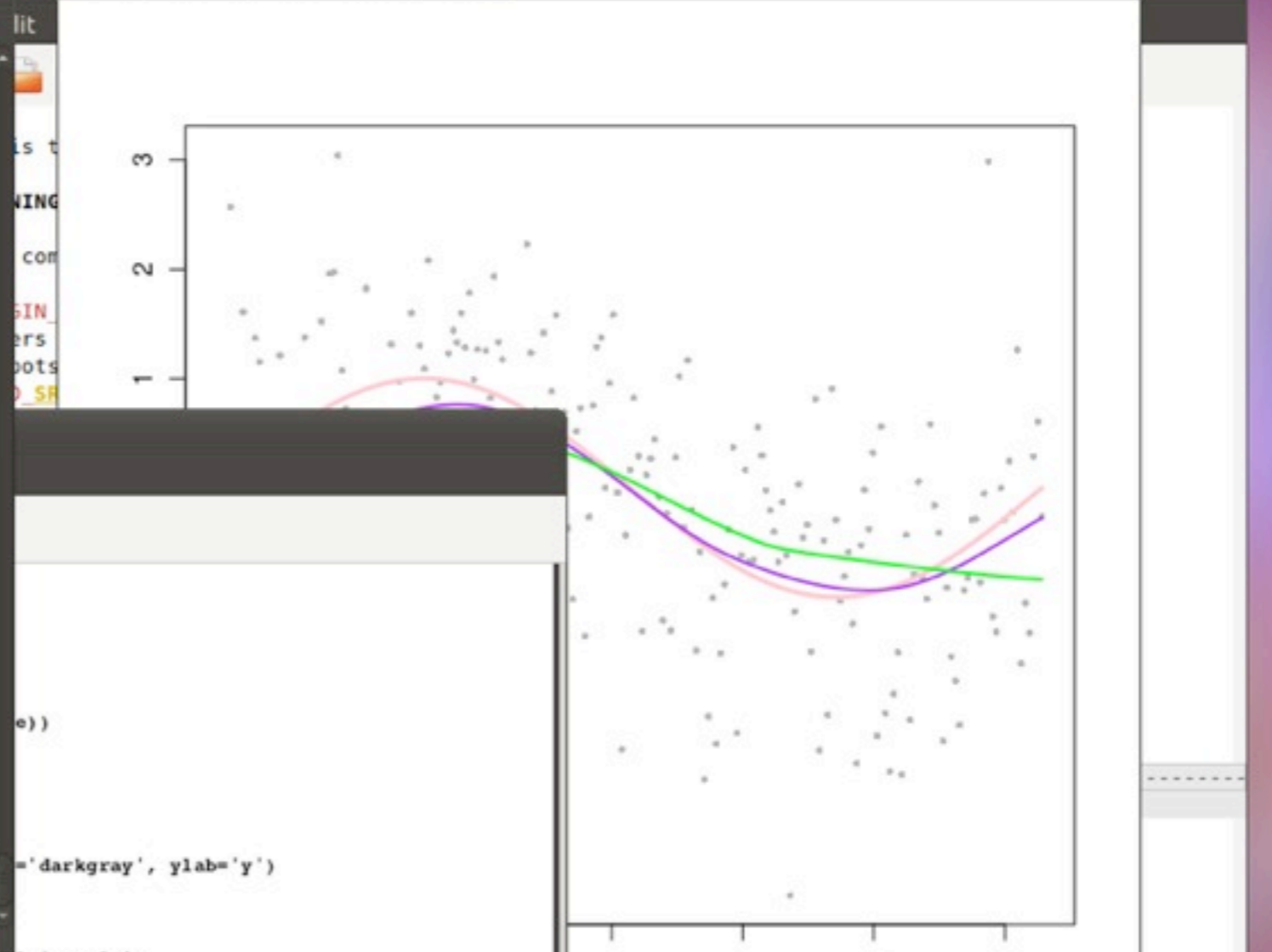
RC)

```

> y
[1] 0.38070405 0.70888962 0.26294848 0.93555879 0.23653209 0.98141344
[7] 0.22387453 0.85294810 0.92143502 0.60954259 0.16528519 0.40504211
[13] 0.30759486 0.95356759 0.77035950 0.76423660 0.47279298 -0.03375378
[19] 0.53919154 0.28237485 0.31779874 0.44245689 0.11865432 0.67254863
[25] -0.08576078 0.47330848 0.02666198 0.32440023 0.91024358 0.27136509
[31] 0.42767611 0.29522879 0.37724780 0.72155036 0.05652227 1.04286504
[37] 1.19401811 0.72647059 0.85447750 0.86676835 0.03062600 0.99766974
[43] 0.33133839 1.32746554 0.57030640 0.38954883 0.15711842 0.39196118
[49] 1.19686059 0.93438992 0.14405303 0.84631178 0.65502037 0.21170073
[55] 0.79344456 0.57308377 -0.03905062 0.47399737 0.33357678 0.44934856
[61] -0.09815956 0.43713004 0.31309569 0.97812095 0.67841770 0.75187395
[67] 0.67360936 0.77008910 0.50299565 0.38971149 0.88743474 0.07199642
[73] -0.09656263 0.20341761 0.11653821 1.18388344 0.44747044 0.73582272
[79] 0.02241823 0.38938948 0.87819312 1.12761336 0.57130928 0.27501661
[85] 0.34760132 0.46575411 -0.19376734 0.93431750 0.37287705 1.09851433
[91] 0.39029975 0.58280824 0.57384136 -0.06849502 0.78833032 0.70441729
[97] 0.93772624 0.30803622 0.69246134 -0.18310708

> y.smoo = smooth.split(x,y.obs)
Error: could not find function "smooth.split"
> y.smoo = smooth.spline(x,y.obs)
> y.low = lowesss(x, y.obs)
Error: could not find function "lowesss"
> y.low = lowess(x, y.obs)
> help(lowess)
> plot(x,y.obs, pch=16, cex=0.6, col='darkgray', ylab='y')
> lines(x, y.true, col='pink', lwd=3)
Error in plot.xy(xy.coords(x, y), type = type, ...) :
invalid color name 'ping'
> lines(x, y.true, col='pink', lwd=3)
> lines(y.smoo$x, y.smoo$y, lwd=2, col='purple')
> lines(y.low$x, y.low$y, lwd=2, col='green')

```



```

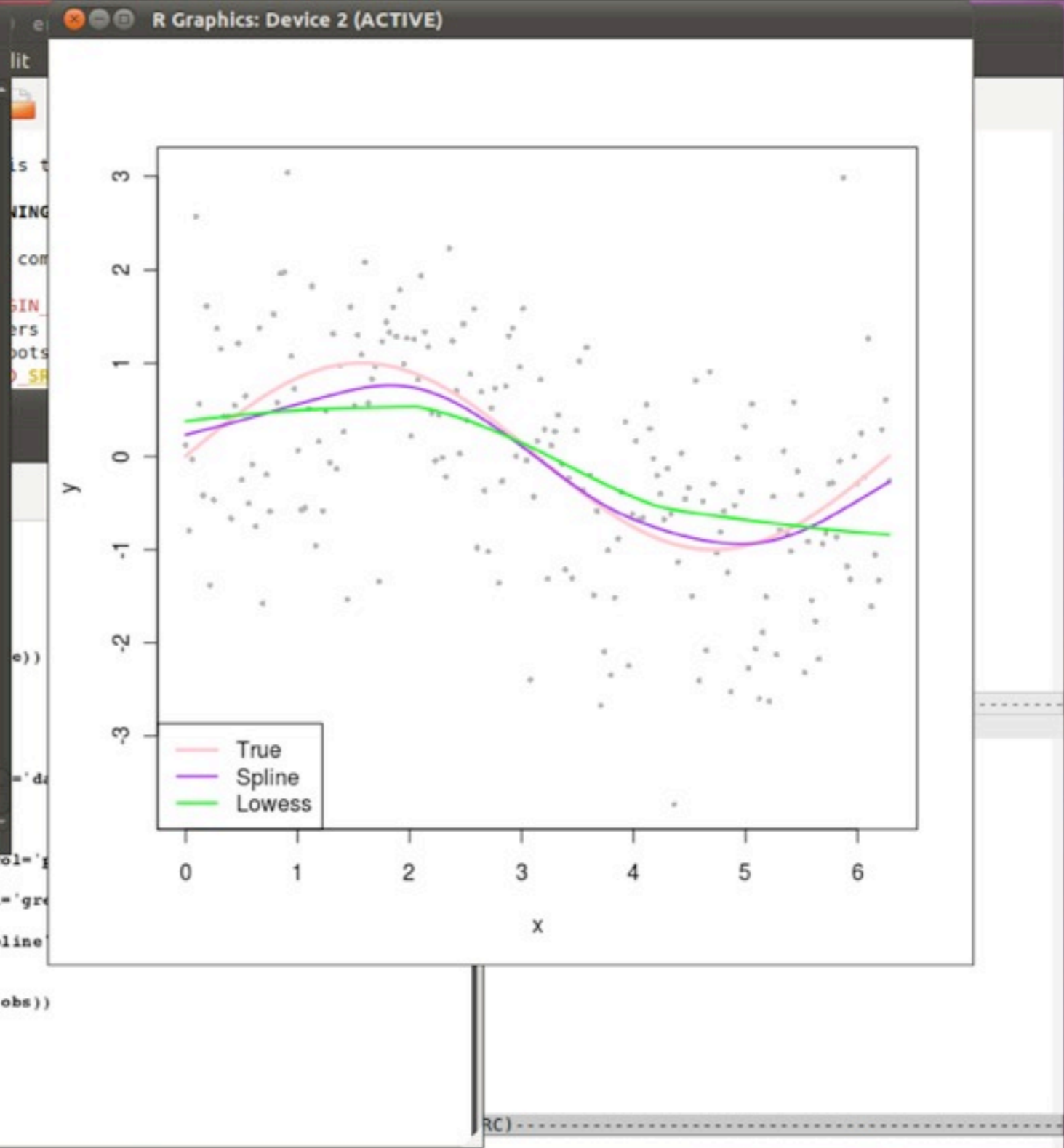
> lines(y.smoo$x, y.smoo$y, lwd=2, col='purple')
# 8
> lines(y.low$x, y.low$y, lwd=2, col='green')
# 9
> legend('bottomleft', c('True', 'Spline', 'Lowess'),
lwd=c(3,2,2), # 10
+ col=c('pink','purple','green'))
> ind.max = which(y.obs == median(y.obs))
# 11

```

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
[7] 0.22387453 0.85294810 0.92143502 0.60954259 0.16528519 0.40504211
[13] 0.30759486 0.95356759 0.77035950 0.76423660 0.47279298 -0.03375378
[19] 0.53919154 0.28237485 0.31779874 0.44245689 0.11865432 0.67254863
[25] -0.08576078 0.47330848 0.02666198 0.32440023 0.91024358 0.27136509
[31] 0.42767611 0.29522879 0.37724780 0.72155036 0.05652227 1.04286504
[37] 1.19401811 0.72647059 0.85447750 0.86676835 0.03062600 0.99766974
[43] 0.33133839 1.32746554 0.57030640 0.38954883 0.15711842 0.39196118
[49] 1.19686059 0.93438992 0.14405303 0.84631178 0.65502037 0.21170073
[55] 0.79344456 0.57308377 -0.03905062 0.47399737 0.33357678 0.44934856
[61] -0.09815956 0.43713004 0.31309569 0.97812095 0.67841770 0.75187395
[67] 0.67360936 0.77008910 0.50299565 0.38971149 0.88743474 0.07199642
[73] -0.09656263 0.20341761 0.11653821 1.18388344 0.44747044 0.73582272
[79] 0.02241823 0.38938948 0.87819312 1.12761336 0.57130928 0.27501661
[85] 0.34760132 0.46575411 -0.19376734 0.93431750 0.37287705 1.09851433
[91] 0.39029975 0.58280824 0.57384136 -0.06849502 0.78833032 0.70441729
[97] 0.93772624 0.30803622 0.69246134 -0.18310708
> y.smoo = smooth.split(x,y.obs)
Error: could not find function "smooth.split"
> y.smoo = smooth.spline(x,y.obs)
> y.low = lowesss(x, y.obs)
Error: could not find function "lowesss"
> y.low = lowess(x, y.obs)
> help(lowess)
> plot(x,y.obs, pch=16, cex=0.6, col='darkgray', ylab='y')
> lines(x, y.true, col='pink', lwd=3)
Error in plot.xy(xy.coords(x, y), type = type, ...) :
  invalid color name 'ping'
> lines(x, y.true, col='pink', lwd=3)
> lines(y.smoo$x, y.smoo$y, lwd=2, col='purple')
> lines(y.low$x, y.low$y, lwd=2, col='green')
> legend('bottomleft', c('True', 'Spline', 'Lowess'), lwd=c(3,2,2),
+ col=c('pink', 'purple', 'green'))
>

```



```

> lines(y.smoo$x, y.smoo$y, lwd=2, col='purple') # 8
> lines(y.low$x, y.low$y, lwd=2, col='green') # 9
> legend('bottomleft', c('True', 'Spline', 'Lowess'), lwd=c(3,2,2), # 10
+ col=c('pink', 'purple', 'green'))
> ind.max = which(y.obs == median(y.obs)) # 11

```

researchtools@ubuntu: ~/class/25

File Edit View Search Terminal Help

which package:base R Documentation

Which indices are TRUE?

## Description:

Give the 'TRUE' indices of a logical object, allowing for array indices.

## Usage:

```
which(x, arr.ind = FALSE, useNames = TRUE)
arrayInd(ind, .dim, .dimnames = NULL, useNames = FALSE)
```

## Arguments:

x: a 'logical' vector or array. 'NA's are allowed and omitted (treated as if 'FALSE').

arr.ind: logical; should \*arr\*ay \*ind\*ices be returned when 'x' is an array?

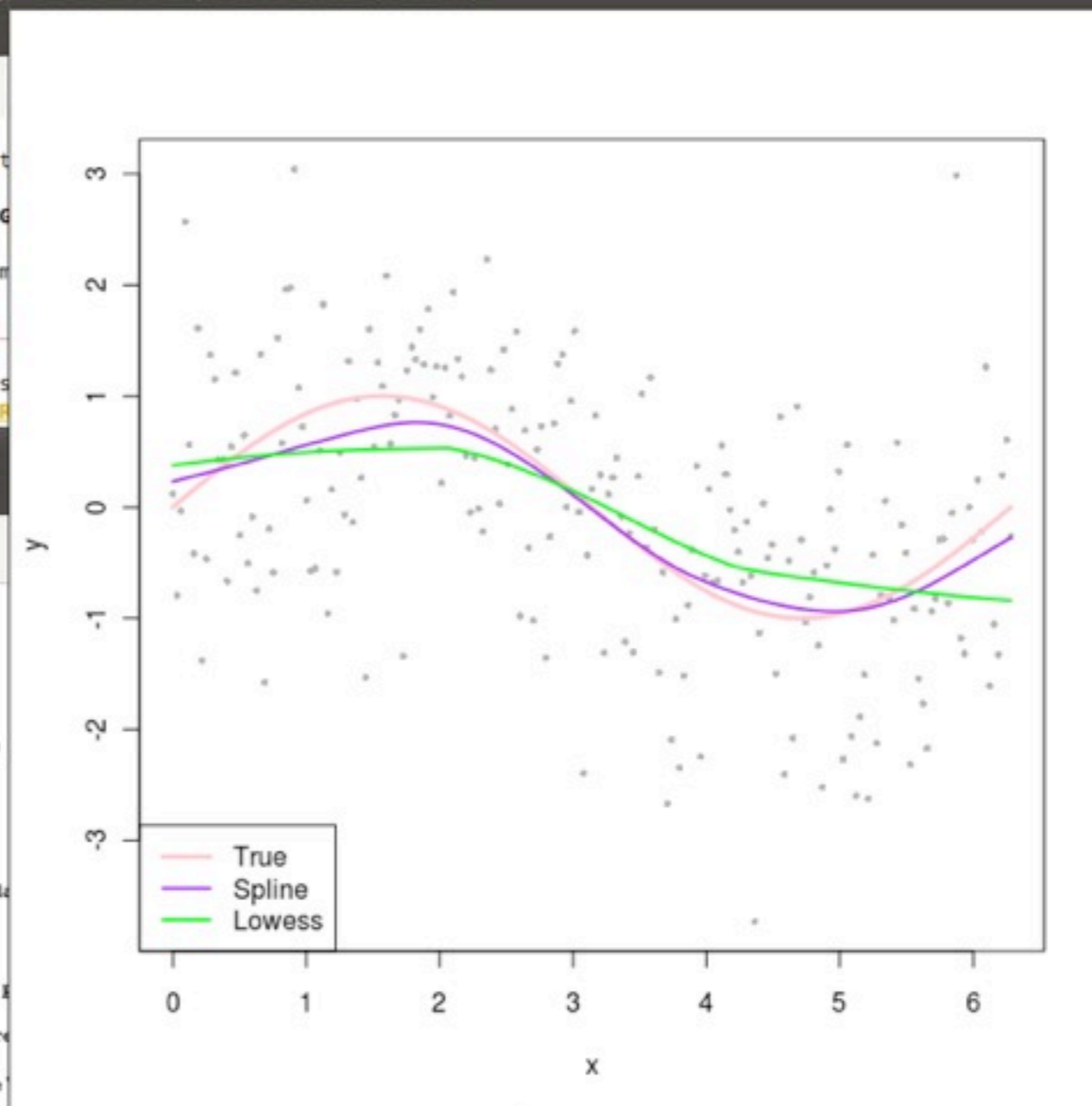
ind: integer-valued index vector, as resulting from 'which(x)'.  
 .dim: 'dim(.)' integer vector

.dimnames: optional list of character 'dimnames(.)', of which only '.dimnames[[1]]' is used.

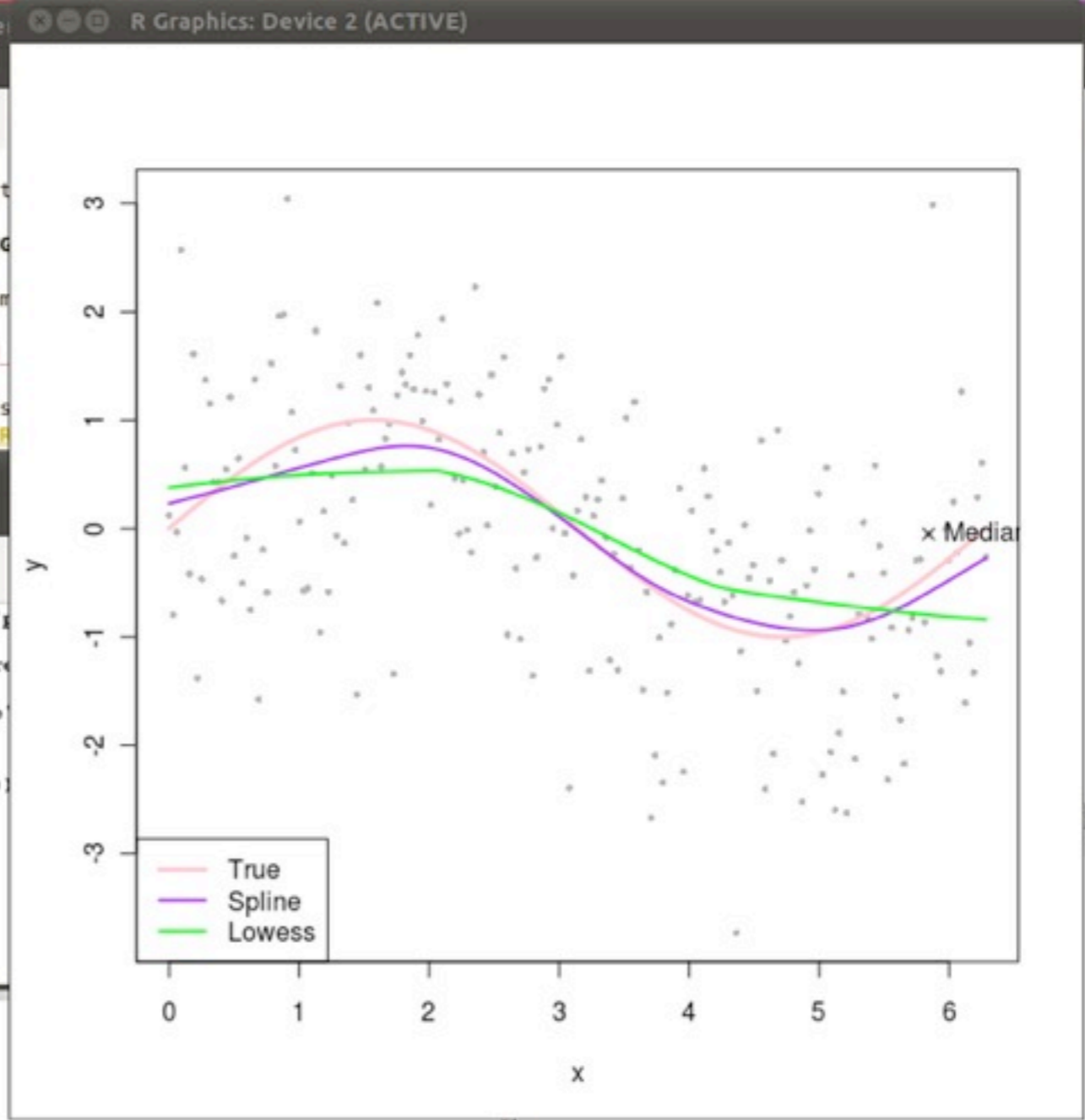
useNames: logical indicating if the value of 'arrayInd()' should have (non-null) dimnames at all.

```
> lines(y.smoo$x, y.smoo$y, lwd=2, col='pink')
# 8
> lines(y.low$x, y.low$y, lwd=2, col='green')
# 9
> legend('bottomleft', c('True', 'Spline', 'Lowess'),
lwd=c(3,2,2), # 10
+ col=c('pink', 'purple', 'green'))
> ind.max = which(y.obs == median(y.obs))
# 11
```

R Graphics: Device 2 (ACTIVE)



```
researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
[91] 0.39029975 0.58280824 0.57384136 -0.06849502 0.78833032 0.70441729
[97] 0.93772624 0.30803622 0.69246134 -0.18310708
> y.smoo = smooth.split(x,y.obs)
Error: could not find function "smooth.split"
> y.smoo = smooth.spline(x,y.obs)
> y.low = lowesss(x, y.obs)
Error: could not find function "lowesss"
> y.low = lowess(x, y.obs)
> help(lowess)
> plot(x,y.obs, pch=16, cex=0.6, col='darkgray', ylab='y')
> lines(x, y.true, col='pink', lwd=3)
Error in plot.xy(xy.coords(x, y), type = type, ...) :
  invalid color name 'pink'
> lines(x, y.true, col='purple', lwd=3)
> lines(y.smoo$x, y.smoo$y, lwd=2, col='purple')
> lines(y.low$x, y.low$y, lwd=2, col='green')
> legend('bottomleft', c('True', 'Spline', 'Lowess'), lwd=c(3,2,2),
+ col=c('pink', 'purple', 'green'))
> ind.max = which(y.obs == median(y.obs))
> ind.max
[1] 187
> median(y.obs)
[1] -0.05056401
> help(which)
> x.max = x[ind.max]
> x.max
[1] 5.843362
> y.max = y.obs[ind.max]
> y.max
[1] -0.05056401
> points(x.max, y.max, pch=4)
> text(x.max,y.max, paste("Median:",format(y.max,digits=4)), pos=4)
>
```



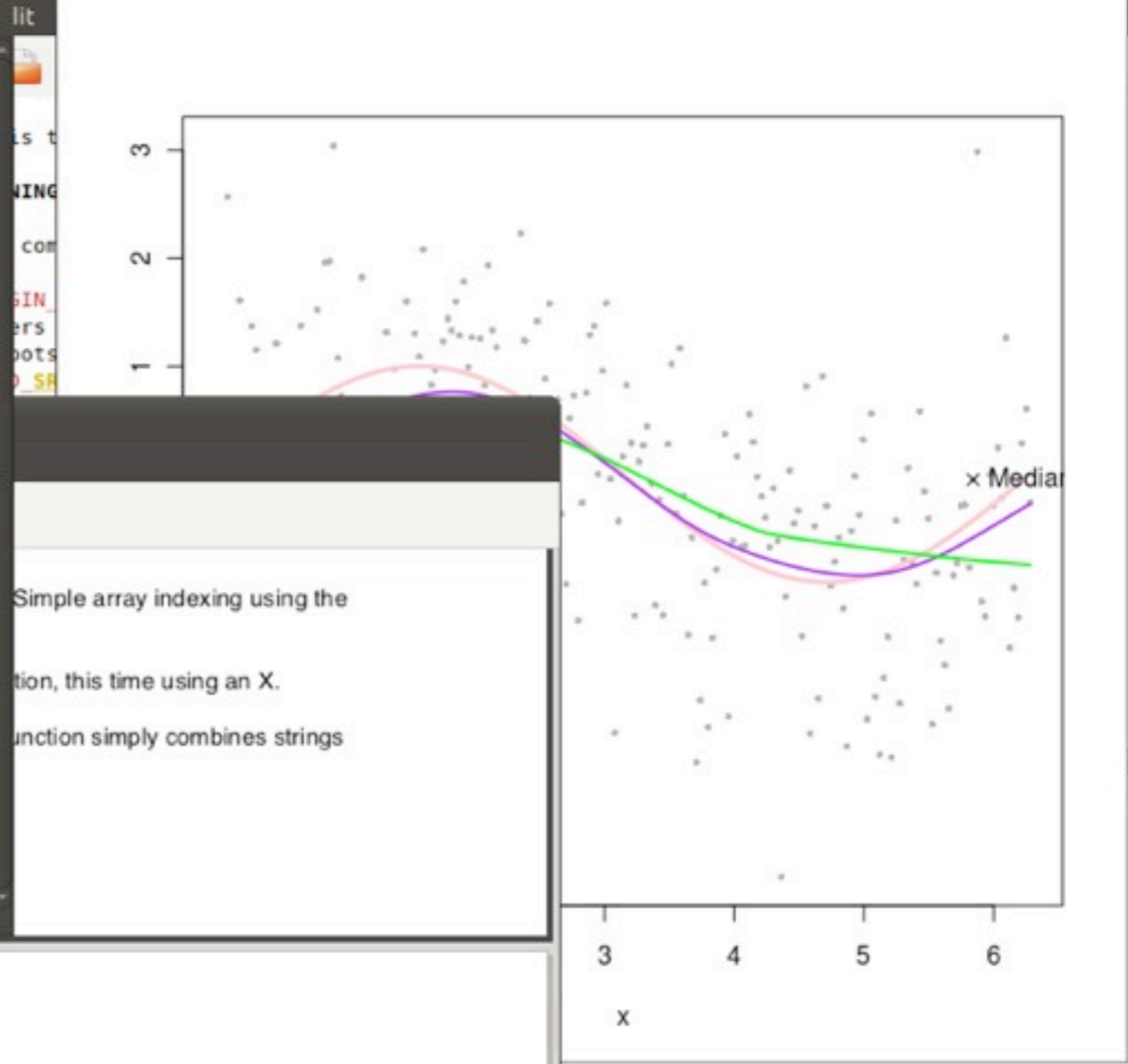
```
> x.max=x[ind.max]
# 12
> y.max=y.obs[ind.max]
# 13
> points(x.max, y.max, pch=4)
# 14
> text(x.max, y.max, paste("Median:",format(y.max, digits=4)),
pos=4) # 15
```



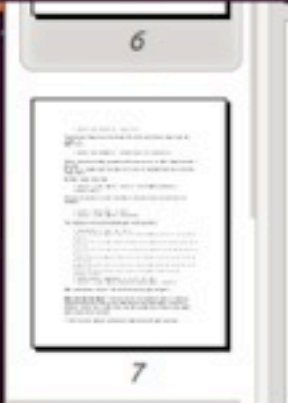
```

File Edit View Search Terminal Help
paste package:base R Documentation
Concatenate Strings
Description:
Concatenate vectors after converting to character.
Usage:
paste(..., sep = " ", collapse = NULL)
Arguments:
...: one or more R objects, to be converted to character vectors.
sep: a character string to separate the terms. Not 'NA_character_'.
collapse: an optional character string to separate the results. Not 'NA_character_'.
Details:
'paste' converts its arguments (via 'as.character') to character strings, and concatenates them (separating them by the string given by 'sep'). If the arguments are vectors, they are concatenated term-by-term to give a character vector result. Vector arguments are recycled as needed, with zero-length arguments being recycled to ''.
Note that 'paste()' coerces 'NA_character_', the character missing

```



Simple array indexing using the  
 tion, this time using an X.  
 nction simply combines strings



```

> paste('the answer is', log(1.31))
The other new thing here is the format() function, which I am using to trim the
number of
digits. Try it:

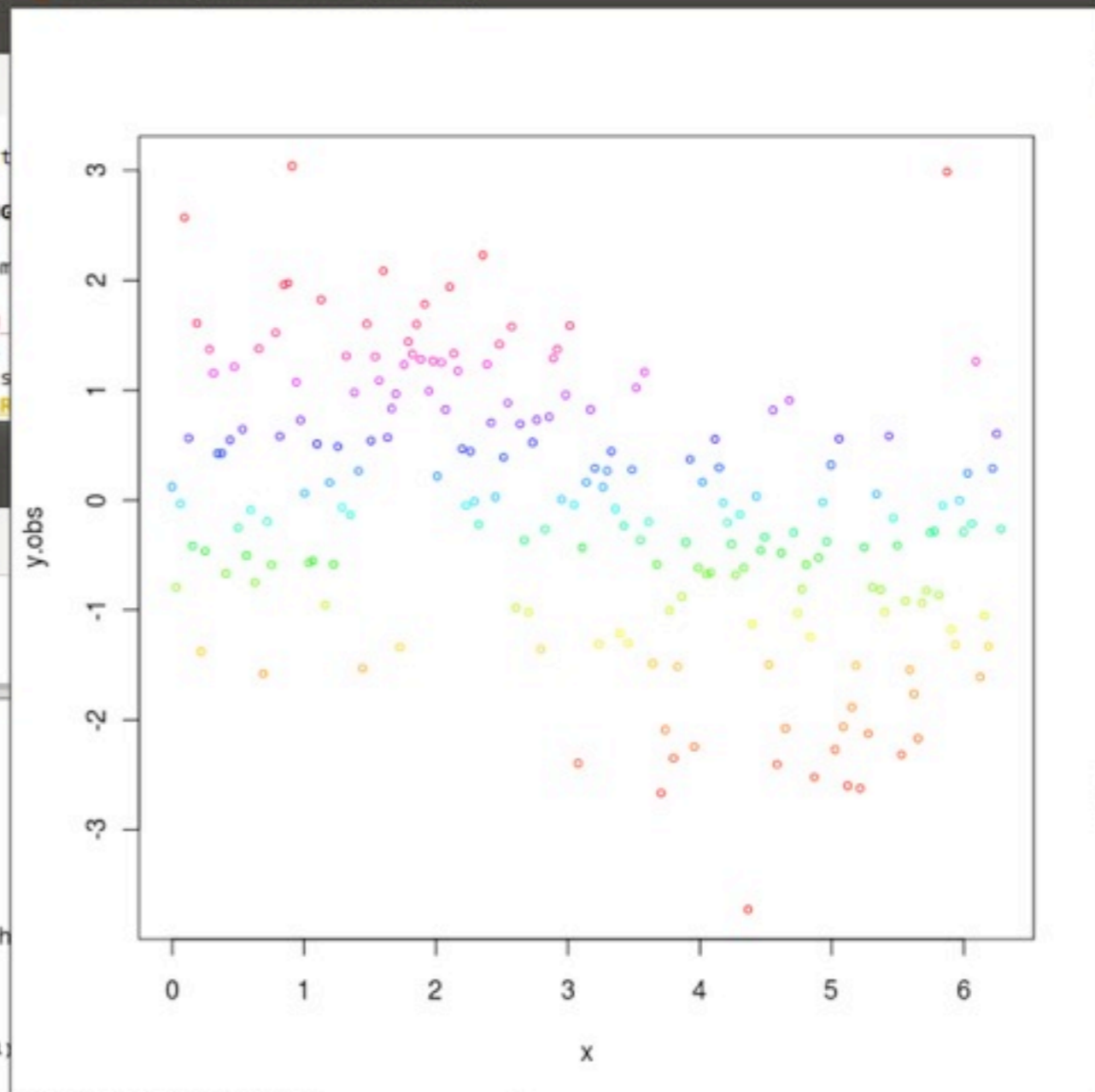
```

researchtools@ubuntu: ~/class/25

File Edit View Search Terminal Help

```
> legend('bottomleft', c('True', 'Spline', 'Lowess'), lwd=c(3,2,2),
+ col=c('pink', 'purple', 'green'))
> ind.max = which(y.obs == median(y.obs))
> ind.max
[1] 187
> median(y.obs)
[1] -0.05056401
> help(which)
> x.max = x[ind.max]
> x.max
[1] 5.843362
> y.max = y.obs[ind.max]
> y.max
[1] -0.05056401
> points(x.max, y.max, pch=4)
> text(x.max, y.max, paste("Median:", format(y.max, digits=4)), pos=4)
> help(paste)
> paste('the answer is', log(1.31))
[1] "the answer is 0.27002713721306"
> paste('the answer is', format(log(1.31), digits=3))
[1] "the answer is 0.27"
> plot(x, y.obs, pch=16, cex=0.6, col=rainbow(length(x))
+ [rank(y.obs)])
Warning messages:
1: In plot.window(...) : "pch" is not a graphical parameter
2: In plot.xy(xy, type, ...) : "pch" is not a graphical parameter
3: In axis(side = side, at = at, labels = labels, ...) :
"pch" is not a graphical parameter
4: In axis(side = side, at = at, labels = labels, ...) :
"pch" is not a graphical parameter
5: In box(...) : "pch" is not a graphical parameter
6: In title(...) : "pch" is not a graphical parameter
>
```

R Graphics: Device 2 (ACTIVE)



digits. Try it:

```
> paste('the answer is', format(log(1.31), digits=3))
```

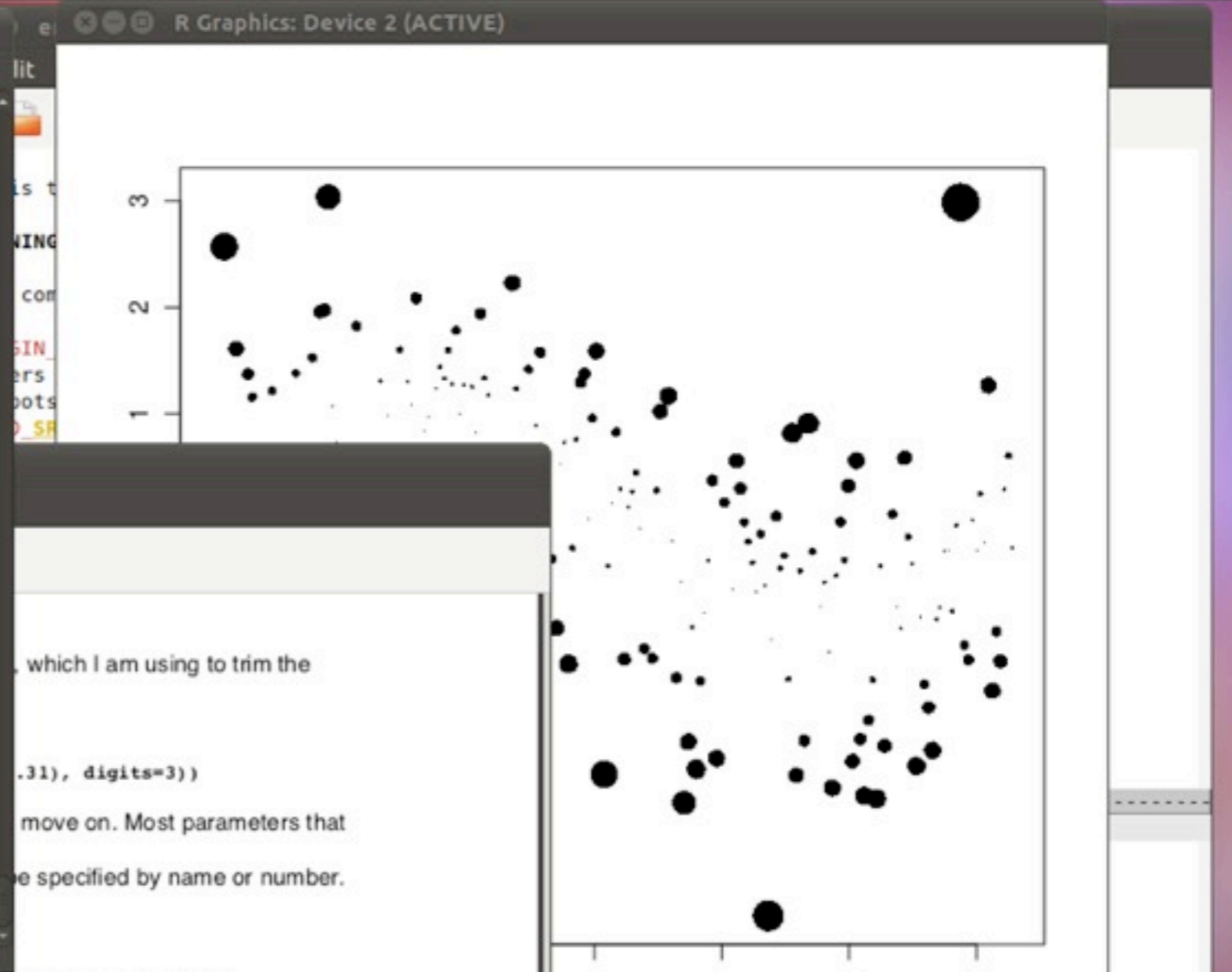
A final note about plotting parameters before we move on: most parameters that involve a choice (e.g., symbol type, line type, color), can be specified by name or number. The power of the latter can be seen here:

```
> plot(x, y.obs, pch=16, cex=0.6, col=rainbow(length(x))
+ [rank(y.obs)])
```

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
> median(y.obs)
[1] -0.05056401
> help(which)
> x.max = x[ind.max]
> x.max
[1] 5.843362
> y.max = y.obs[ind.max]
> y.max
[1] -0.05056401
> points(x.max, y.max, pch=4)
> text(x.max,y.max, paste("Median:",format(y.max,digits=4)), pos=4)
> help(paste)
> paste('the answer is', log(1.31))
[1] "the answer is 0.27002713721306"
> paste('the answer is', format(log(1.31), digits=3))
[1] "the answer is 0.27"
> plot(x,y.obs, pcg=16, cex=0.6, col=rainbow(length(x))
+ [rank(y.obs)])
Warning messages:
1: In plot.window(...) : "pcg" is not a graphical parameter
2: In plot.xy(xy, type, ...) : "pcg" is not a graphical parameter
3: In axis(side = side, at = at, labels = labels, ...) :
"pcg" is not a graphical parameter
4: In axis(side = side, at = at, labels = labels, ...) :
"pcg" is not a graphical parameter
5: In box(...) : "pcg" is not a graphical parameter
6: In title(...) : "pcg" is not a graphical parameter
> noise = abs(y.true - y.obs)
> plot(x,y, pch=16, cex=noise)
Error in xy.coords(x, y, xlabel, ylabel, log) :
'x' and 'y' lengths differ
> plot(x,y.obs, pch=16, cex=noise)
>

```



which I am using to trim the

.31), digits=3))

move on. Most parameters that

be specified by name or number.

```

> plot(x, y.obs, pch=16, cex=0.6, col=rainbow(length(x))
+ [rank(y.obs)])

```

Similarly, the options that are real-valued, can take a vector of numbers, for example:

```

> noise = abs(y.true - y.obs)
> plot(x, y.obs, pch=16, cex=noise)

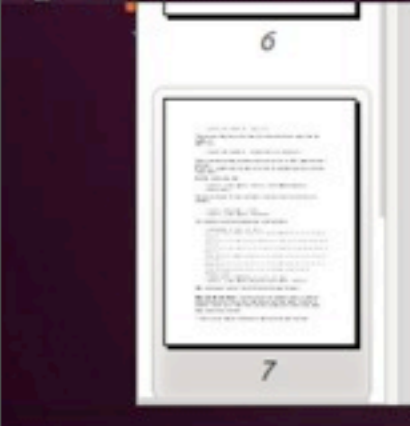
```

Or a slightly more complicated example involving factors:

```

> cut(noise, 3, c('L', 'M', 'H'))
[1] L L L M L L H M L L M L L L L M M L H M M L L L L L L M M L

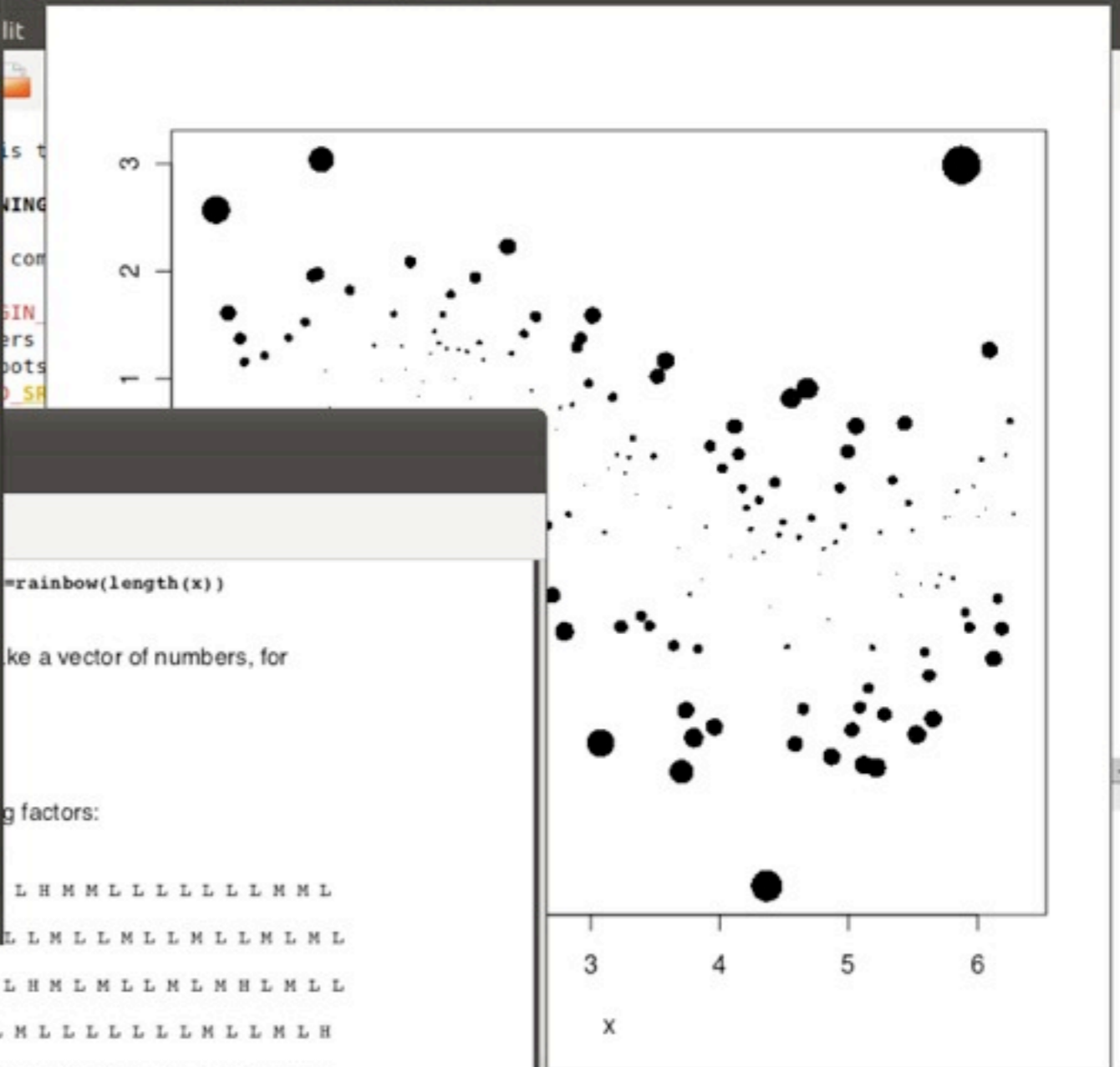
```



```

> paste('the answer is', log(1.31))
[1] "the answer is 0.27002713721306"
> paste('the answer is', format(log(1.31), digits=3))
[1] "the answer is 0.27"
> plot(x,y.obs, pch=16, cex=0.6, col=rainbow(length(x))
+ [rank(y.obs)])
Warning messages:
1: In plot.window(...) : "pch" is not a graphical parameter
2: In plot.xy(xy, type, ...) : "pch" is not a graphical parameter
3: In axis(side = side, at = at, labels = labels, ...) :
"pch" is not a graphical parameter
4: In axis(side = side, at = at, labels = labels, ...) :
"pch" is not a graphical parameter
5: In box(...) : "pch" is not a graphical parameter
6: In title(...) : "pch" is not a graphical parameter
> noise = abs(y.true - y.obs)
> plot(x,y, pch=16, cex=noise)
Error in xy.coords(x, y, xlabel, ylabel, log) :
'x' and 'y' lengths differ
> plot(x,y.obs, pch=16, cex=noise)
> cut(noise, 3, c('L', 'M', 'H'))
[1] L L L H L L M M L L L L L L L L L L L L L M L M L M L L M M M L L L M M L L
[38] M L M L L L L L L L H L L L L L L L L L L L L L L L L L L L L L L L L L L
[75] L M L L L L L L L M L L M L L M L L L M L L M L H L L L L M L L L L L L L
[112] L M L M L L L M M L M L L L L M L L L L M M L L L L L H L L L L L M M L
[149] L M L L L L L M L L L M M M M L L M L M L L L L M L L M L L M M L L L L
[186] L L H L L L L L L M M L M L L L
Levels: L M H
> help(math)
No documentation for 'math' in specified packages and libraries:
you could try '??math'
> ??math
> ??math

```



```

> rainbow(length(x))
# Make a vector of numbers, for
# ...
> plot(x, y, pch=rainbow(length(x)))
# ...
> plot(x, y, pch=as.character(noise.level), cex=0.6)

```



```

[1] L L L H L L M M L L L L L L L L L L L L L M L M L M L L M M M L L L M M L L
[38] M L M L L L L L L L H L L L L L L L L L L L L L L L L L L L L L L L L L L
[75] L M L L L L L L L M L L M L L M L L L M L L M L H L L L L M L L L L L L L
[112] L M L M L L L M M L M L L L L M L L L L M M L L L L L H L L L L L M M L
[149] L M L L L L L M L L L M M M M L L M L M L L L L M L L M L L M M L L L L
[186] L L H L L L L L L M M L M L L L
Levels: L M H
> noise.level = cut(noise, 3, c('L', 'M', 'H'))
> plot(x, y.obs, pch=as.character(noise.level), cex=0.6)

```

What exactly does cut() do? Use the R help utility; type help(cut).

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
Help files with alias or concept or title matching 'math' using regular
expression matching:

base::DateTimesClasses Date-Time Classes
base::Date Date Class
base::abs Miscellaneous Mathematical Functions
base::Special Special Functions of Mathematics
base::difftime Time Intervals
base::factor Factors
base::S3groupGeneric S3 Group Generic Functions
graphics::strwidth Plotting Dimensions of Character Strings and
Math Expressions
grDevices::plotmath Mathematical Annotation in R
lattice::panel.identify Functions to Interact with Lattice Plots
lattice::panel.functions Useful Panel Functions
lattice::panel.qqmath Default Panel Function for qqmath
lattice::panel.qqmathline Useful panel function with qqmath
lattice::prepanel.default.bwplot Default Prepanel Functions
lattice::prepanel.lmline Useful Prepanel Function for Lattice
lattice::qqmath Q-Q Plot with Theoretical Distribution
lattice::tmd Tukey Mean-Difference Plot
MASS::fractions Rational Approximation
Matrix::CsparseMatrix-class Class "CsparseMatrix" of Sparse Matrices in
Column-compressed Form
Matrix::dMatrix-class (Virtual) Class "dMatrix" of "double" Matrices
:

```



```

> rainbow(length(x))
> # make a vector of numbers, for
> # coloring factors:
> L H M M L L L L L L M M L
> L L M L L M L L M L L M L M L
> M H M M
[71] L L L L H L L H L L L L L L M L L H M L M L L M H L M L L
M L L L
[106] M M L L L M M L L M H L L L L L M L L L L L L M L L M L H
L L H M L
[141] L L L M L L L L M L M L L L H M L H M H M L N L M H M L L L
L L H L L
[176] M M M M H L L M L M L L M L L M M M L L M M L M M M
Levels: L M H
> noise.level = cut(noise, 3, c('L', 'M', 'H'))
> plot(x, y.obs, pch=as.character(noise.level), cex=0.6)

```

```

> plot(x, y.obs, pch=as.character(noise.level), cex=0.6)

```

What exactly does cut() do? Use the R help utility; type help(cut).



Google Custom Search - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Google Custom Search

www.google.com/cse?cx=partner-pub-9300639326172081%3A11243405138

Google custom search  Search

About 48 results (0.05 seconds)

Ads by Google

**Accountable Care (ACOs)**  
[www.med3000.com](http://www.med3000.com) Learn How to Navigate Regulations & Rapidly Align Physicians w/ Webinar  
 Video Testimonials About Us  
 Products & Services Request Information

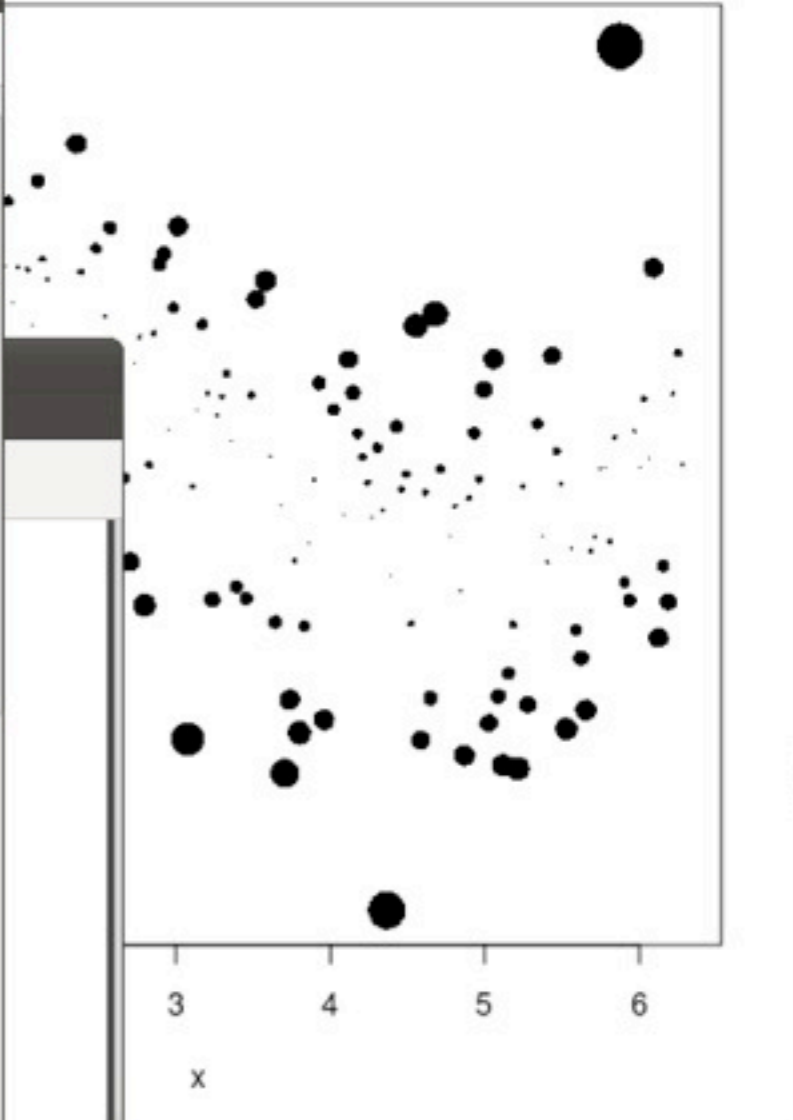
**ACOs**  
[www.healthcarestrategygroup.com](http://www.healthcarestrategygroup.com) A Partner in ACO Development Integration and Implementation

**Trig.Rd**  
 \usage[ cos(x) sin(x) tan(x) acos(x) asin(x) atan(x) atan2(y, x) ] \arguments{ \item{ x, ...  
 \details{ The arc-tangent of two arguments \code{atan2(y, x)} returns the ...  
[svn.r-project.org/R/trunk/src/library/base/man/Trig.Rd](http://svn.r-project.org/R/trunk/src/library/base/man/Trig.Rd)

**Package 'Rmpfr'**  
 File Format: PDF/Adobe Acrobat  
 Sep 10, 2011 ... **atan2** signature(x = "ANY", y = "mpfr"): compute the arc-tangent of two ...  
**atan2**(y, x) returns the angle between the x-axis and the vector from the ...  
[cran.r-project.org/web/packages/Rmpfr/Rmpfr.pdf](http://cran.r-project.org/web/packages/Rmpfr/Rmpfr.pdf)

**Package 'RSQLite.extfuns'**  
 File Format: PDF/Adobe Acrobat  
 May 30, 2010 ... Math: acos, asin, atan, atan2, acosh, asinh, atanh, differ- ence,  
 degrees, radi - ans, cos, sin, tan, cot, cosh, sinh, tanh, coth, exp, log, log10, ...  
[cran.r-project.org/web/packages/RSQLite.extfuns/RSQLite.extfuns.pdf](http://cran.r-project.org/web/packages/RSQLite.extfuns/RSQLite.extfuns.pdf)

**File src/library/base/R/New-Internal.R # Part of the R package, http ...**  
 Internal("comment<-"(x, value)) logb <- function(x, base=exp(1)) if(missing(base)) log(x) else  
 log(x, base) atan2 <- function(y, x) Internal(atan2(y, x)) beta



7

What exactly does cut() do? Use the R help utility; type help(cut).

researchtools@ubuntu: ~/class/25  
researchtools@ubuntu: ~  
emacs23@ubuntu  
lit Options Buffers Tools ERC Help

Google Custom Search - Mozilla Firefox  
File Edit View History Bookmarks Tools Help

Google Custom Search  
www.google.com/cse?cx=partner-pub-9300639326172081%3A11243405138 Google

Google custom search site:r-project.org hyperbolic arc tangent Search

About 11 results (0.07 seconds)

Ads by Google

**Tangent Save 15%**  
www.royaldesign.com/Tangent Buy **Tangent** online. Prices from \$187. Fast delivery - Buy now!

**WildTangent Games**  
www.wildtangent.com Play 1,000's of Games on our New Games App. Tell us What you Think!

**Hyperbolic Tangent**  
www.izito.com/Hyperbolic+Tangent Info on **Hyperbolic Tangent** Search **Hyperbolic Tangent**

Did you mean: [site:r-project.org hyperbolic arctangent](#)

**Trig.Rd**  
\details{ The **arc-tangent** of two arguments `atan2(y, x)` returns the angle between the ... Exponential, Circular and **Hyperbolic** Functions } \keyword{math}  
svn.r-project.org/R/trunk/src/library/base/man/Trig.Rd

**proto: An R Package for Prototype Programming**  
File Format: PDF/Adobe Acrobat  
transforming the correlation coefficient using the **hyperbolic arc tangent** function yields a random variable approximately distributed with an.  $N(p,1)$ .  $f. (n-3)$  ...  
cran.r-project.org/web/packages/proto/vignettes/proto.pdf

**Package 'multinomRob'**  
File Format: PDF/Adobe Acrobat  
May 9, 2010 ... The **inverse** of the hessian matrix (observed formation). covmat. Sandwich ...  
Multinomial Regression **Hyperbolic Tangent** (Tanh) Estimator ...  
cran.r-project.org/web/packages/multinomRob/multinomRob.pdf

ng you for more stuff  
t\* zero.

---

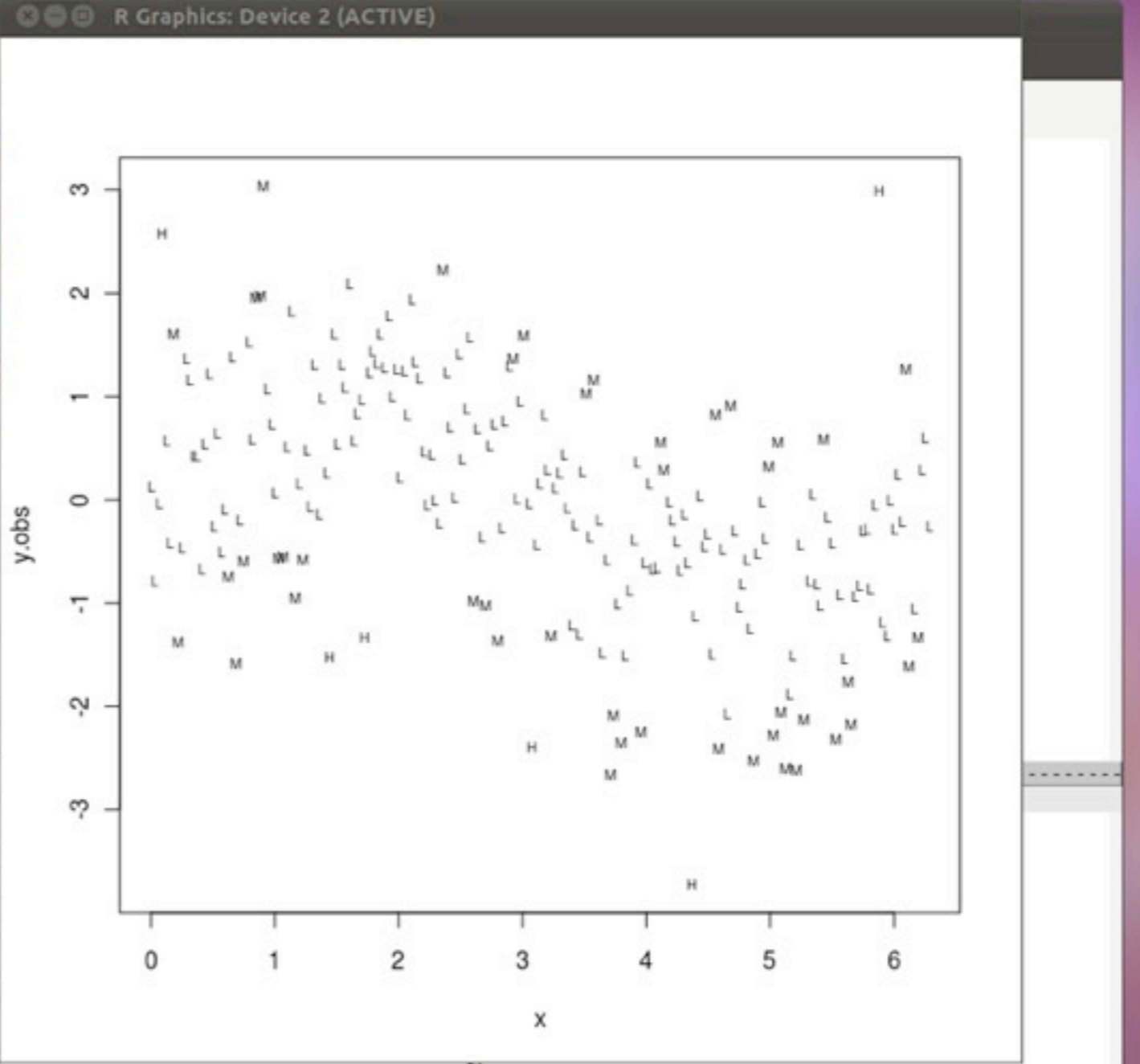
54/  
ough the examples? [11:38]  
[11:39]  
on things! [12:02]  
come upon [12:03]

---

RC)

```
researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help

[1] "the answer is 0.27"
> plot(x,y.obs, pcg=16, cex=0.6, col=rainbow(length(x))
+ [rank(y.obs)])
Warning messages:
1: In plot.window(...): "pcg" is not a graphical parameter
2: In plot.xy(xy, type, ...): "pcg" is not a graphical parameter
3: In axis(side = side, at = at, labels = labels, ...):
  "pcg" is not a graphical parameter
4: In axis(side = side, at = at, labels = labels, ...):
  "pcg" is not a graphical parameter
5: In box(...): "pcg" is not a graphical parameter
6: In title(...): "pcg" is not a graphical parameter
> noise = abs(y.true - y.obs)
> plot(x,y, pch=16, cex=noise)
Error in xy.coords(x, y, xlabel, ylabel, log) :
  'x' and 'y' lengths differ
> plot(x,y.obs, pch=16, cex=noise)
> cut(noise, 3, c('L', 'M', 'H'))
 [1] L L L H L L M M L L L L L L L L L L L L L L L L L L L L L L L L L L M L M L M L L M M M L L L M M L L
[38] M L M L L L L L L L H L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L
[75] L M L L L L L L L L M L L M L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L
[112] L M L M L L L M M L M L L L L M L L L L L L L L L L H L L L L L L M M L
[149] L M L L L L M L L L M M M M L L M L M L L L L M L L M L L M M L L L L
[186] L L H L L L L L L M M L M L L L
Levels: L M H
> help(math)
No documentation for 'math' in specified packages and libraries:
you could try '?math'
> ??math
> ??math
> noise.level = cut(noise, 3, c('L', 'M', 'H'))
> plot(x, y.obs, pch=as.character(noise.level), cex=0.6)
> help
```



Levels: L M H  
 > noise.level = cut(noise, 3, c('L', 'M', 'H'))  
 > plot(x, y.obs, pch=as.character(noise.level), cex=0.6)

What exactly does cut() do? Use the R help utility; type `?cut`

**More on data structures.** The other commonly used data types in R that you might encounter are: lists, arrays, data frames, and time series. I will briefly introduce these here, but their uses and idiosyncrasies will be more apparent when you actually use them.

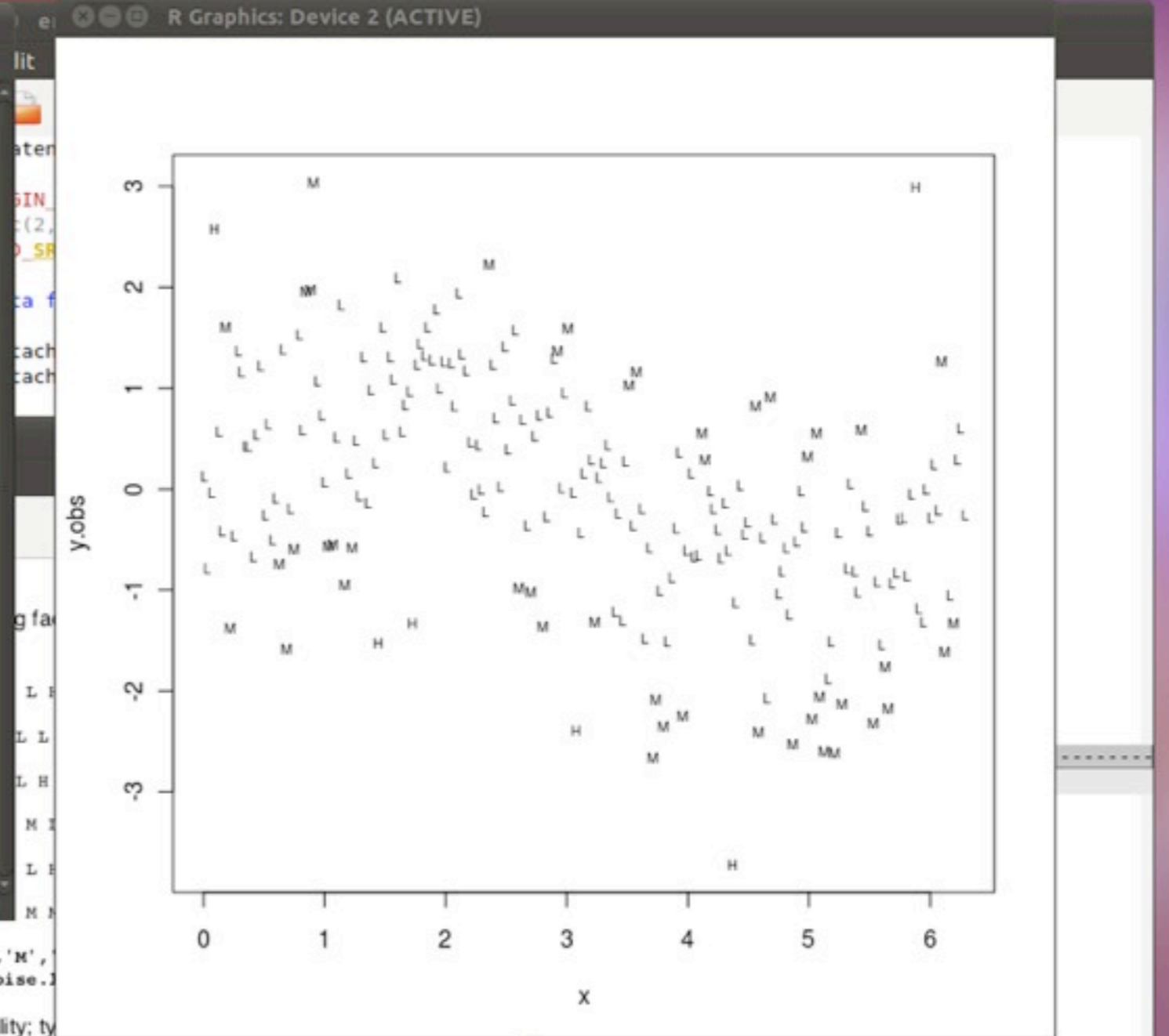
1. A list is just an ordered collection of objects of pretty much any type.



```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
cut package:base R Documentation
Convert Numeric to Factor
Description:
'cut' divides the range of 'x' into intervals and codes the values
in 'x' according to which interval they fall. The leftmost
interval corresponds to level one, the next leftmost to level two
and so on.
Usage:
cut(x, ...)
## Default S3 method:
cut(x, breaks, labels = NULL,
    include.lowest = FALSE, right = TRUE, dig.lab = 3,
    ordered_result = FALSE, ...)
Arguments:
x: a numeric vector which is to be converted to a factor by
cutting.
breaks: either a numeric vector of two or more cut points or a single
number (greater than or equal to 2) giving the number of
intervals into which 'x' is to be cut.
labels: labels for the levels of the resulting category. By default,
labels are constructed using "(a,b)" interval notation. If
'labels = FALSE', simple integer codes are returned instead

```



```

Levels: L M H
> noise.level = cut(noise, 3, c('L','M','H'))
> plot(x, y.obs, pch=as.character(noise.level))

```

What exactly does cut() do? Use the R help utility; type ?cut

**More on data structures.** The other commonly used data types in R that you might encounter are: lists, arrays, data frames, and time series. I will briefly introduce these here, but their uses and idiosyncrasies will be more apparent when you actually use them.

1. A list is just an ordered collection of objects of pretty much any type.

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
> lest(23, 'resp', c(1,4,2), 33.3, list('day', 5), 1024)
Error: could not find function "lest"
> list(23, 'resp', c(1,4,2), 33.3, list('day', 5), 1024)
[[1]]
[1] 23

[[2]]
[1] "resp"

[[3]]
[1] 1 4 2

[[4]]
[1] 33.3

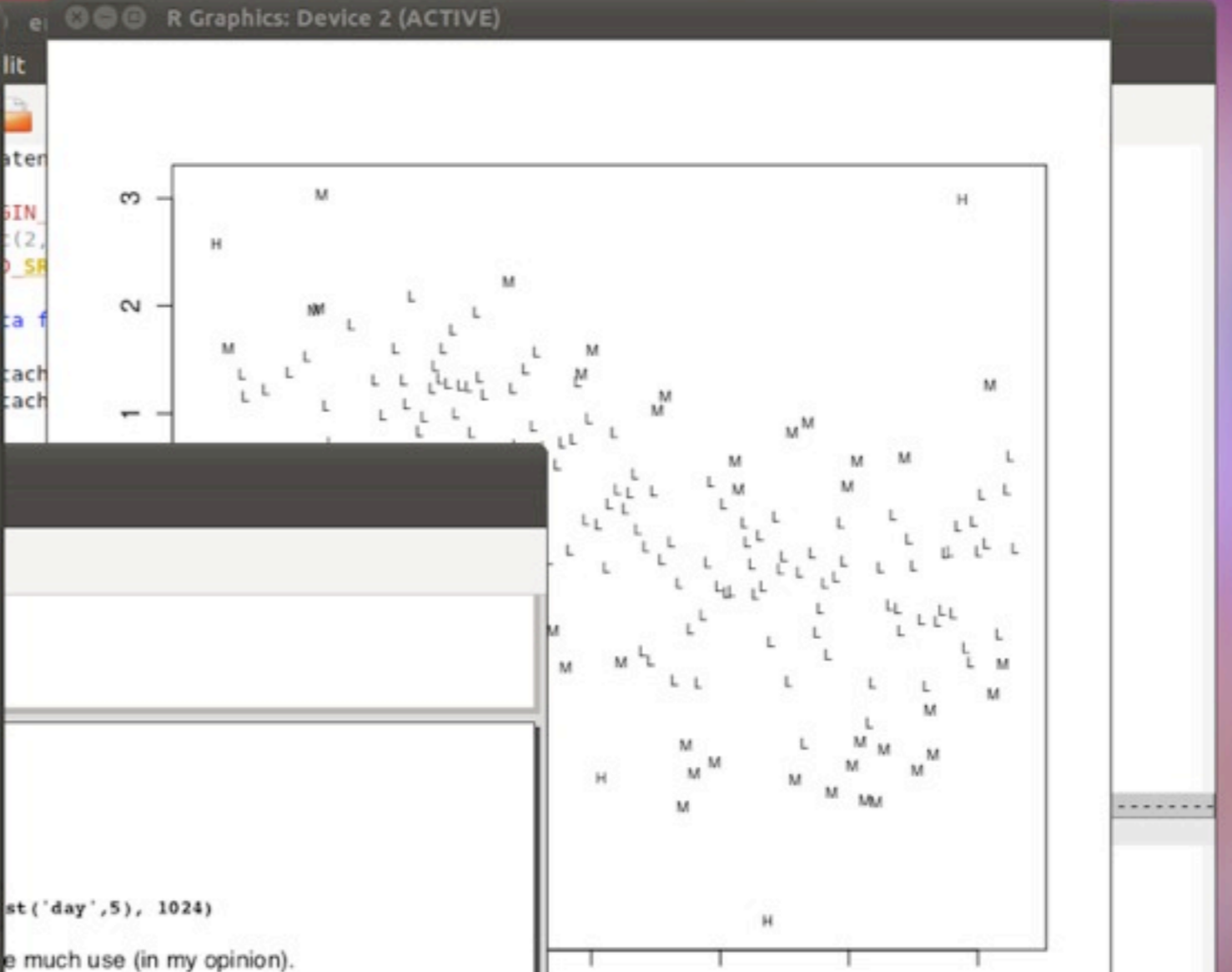
[[5]]
[[5]][[1]]
[1] "day"

[[5]][[2]]
[1] 5

[[6]]
[1] 1024

> dat = c(1,3,7,9,11,35)
> dat
[1] 1 3 7 9 11 35
> dim(dat)
NULL
> help(dim)
> dim(dat)

```



```

...st('day',5), 1024)
...e much use (in my opinion).

```



2. An array is a vector (ordered sequence of numbers) with a dimension attribute:

```

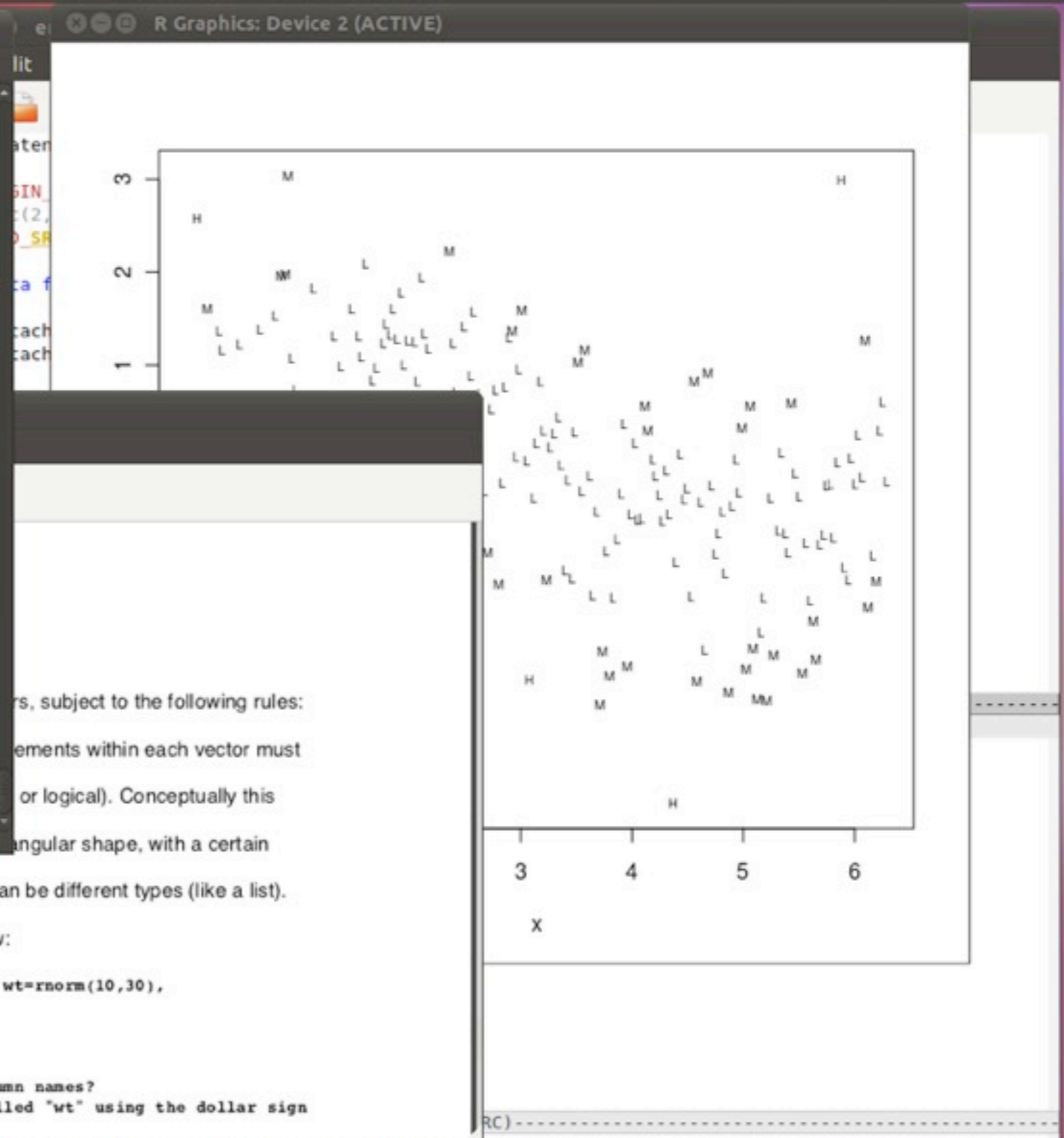
> dat = c(1,3,7,9,11,35)
> dat
[1] 1 3 7 9 11 35
> dim(dat)
NULL
> dim(dat) = c(2,3)
> dat
      [,1] [,2] [,3]
[1,]  1   7  11
[2,]  3   9  35

```

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
NULL
> help(dim)
> dim(dat) = c(2,3)
> dat
  [,1] [,2] [,3]
[1,]  1   7  11
[2,]  3   9  35
>
> my.data = data.frame(age=seq(10), wt=rnorm(10,30), grp=sample(c("A","B"),10,repl=T))
> mydata
Error: object 'mydata' not found
> my.data
  age      wt grp
1  1 29.29847  A
2  2 28.15915  A
3  3 30.20436  B
4  4 29.52910  A
5  5 29.18417  B
6  6 31.55178  B
7  7 28.22086  A
8  8 30.45687  B
9  9 30.64960  B
10 10 29.97496  B
> my.data[4,]
  age      wt grp
4  4 29.52910  A
> my.data[,2]
[1] 29.29847 28.15915 30.20436 29.52910 29.18417 31.55178 28.22086 30.45687
[9] 30.64960 29.97496
> names(my.data)
[1] "age" "wt" "grp"
>

```



ers, subject to the following rules:  
 elements within each vector must  
 or logical). Conceptually this  
 angular shape, with a certain

number of rows  
 and columns (like an array), but the columns can be different types (like a list).  
 You will use  
 data frames a lot. Let's make a simple one now:

```

> my.data = data.frame(age=seq(10), wt=rnorm(10,30),
+ grp=sample(c("A","B"),10,repl=T))
> my.data
> my.data[4,] # fourth row
> my.data[,2] # second column
> names(my.data) # what are the column names?
> my.data$wt # access the column called "wt" using the dollar sign
> plot(my.data$age, my.data$wt)

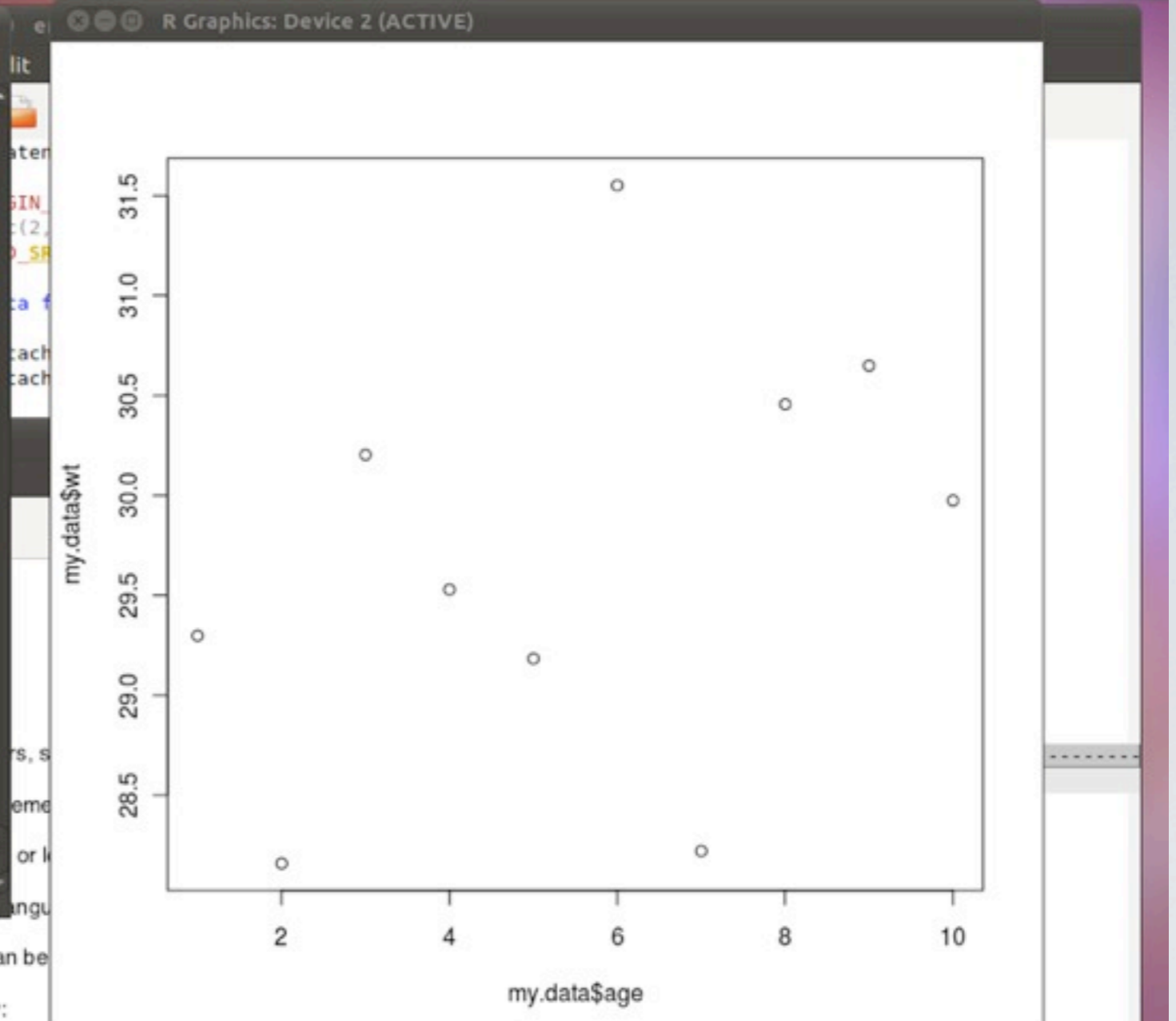
```



```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
[ ,1] [ ,2] [ ,3]
[1,] 1 7 11
[2,] 3 9 35
>
> my.data = data.frame(age=seq(10), wt=rnorm(10,30), grp=sample(c("A","B"),10,rep
pl=T))
> mydata
Error: object 'mydata' not found
> my.data
  age      wt grp
1  1 29.29847  A
2  2 28.15915  A
3  3 30.20436  B
4  4 29.52910  A
5  5 29.18417  B
6  6 31.55178  B
7  7 28.22086  A
8  8 30.45687  B
9  9 30.64960  B
10 10 29.97496  B
> my.data[4,]
  age      wt grp
4  4 29.52910  A
> my.data[,2]
[1] 29.29847 28.15915 30.20436 29.52910 29.18417 31.55178 28.22086 30.45687
[9] 30.64960 29.97496
> names(my.data)
[1] "age" "wt" "grp"
> my.data$wt
[1] 29.29847 28.15915 30.20436 29.52910 29.18417 31.55178 28.22086 30.45687
[9] 30.64960 29.97496
> plot(my.data$age, my.data$wt)
>

```



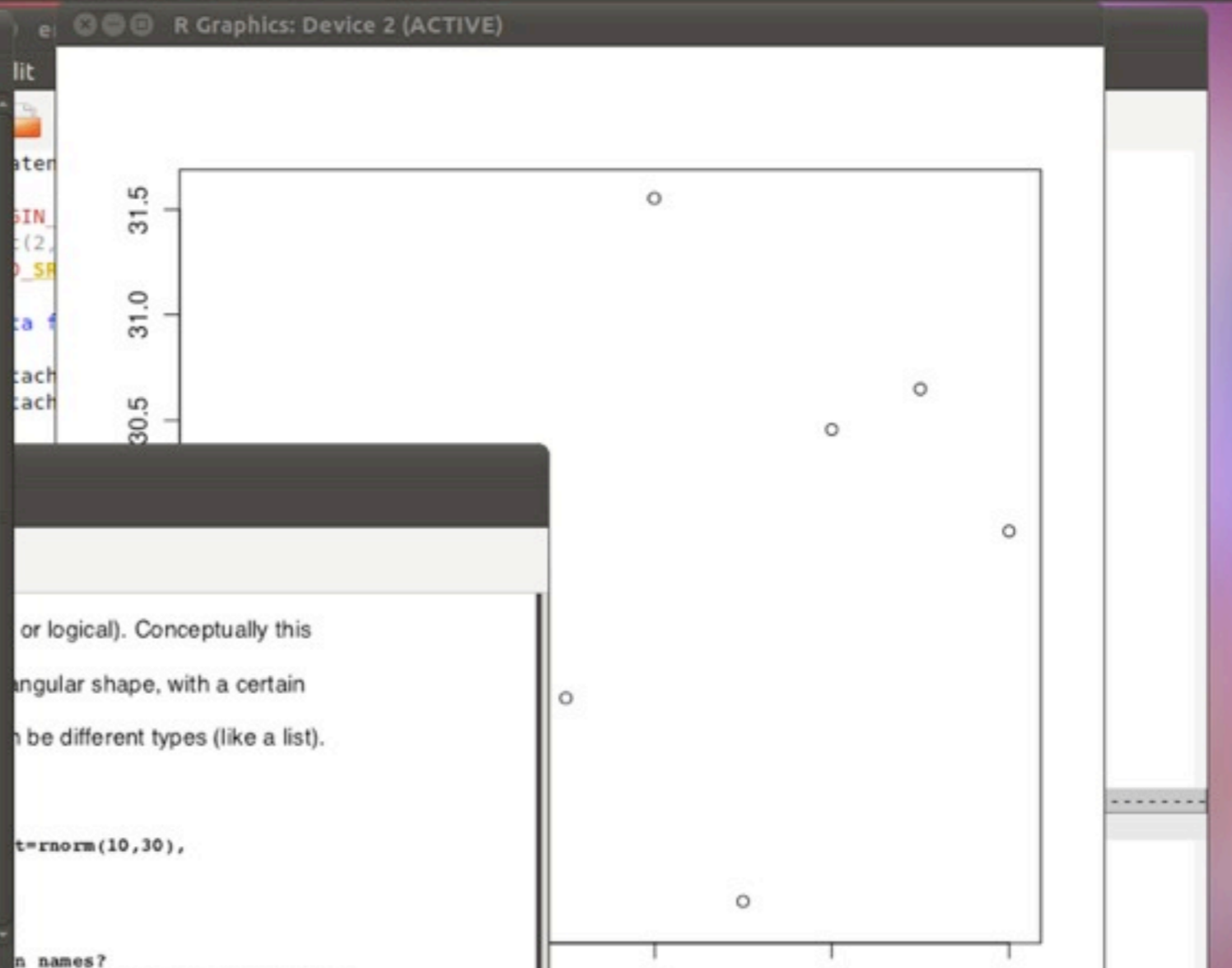
```

number of rows
and columns (like an array), but the columns can be
You will use
data frames a lot. Let's make a simple one now:

> my.data = data.frame(age=seq(10), wt=rnorm(10,30),
+ grp=sample(c("A", "B"), 10, repl=T))
> my.data
> my.data[4,] # fourth row
> my.data[,2] # second column
> names(my.data) # what are the column names?
> my.data$wt # access the column called "wt" using the dollar sign
> plot(my.data$age, my.data$wt)

```

```
researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
tilde package:base R Documentation
Tilde Operator
Description:
Tilde is used to separate the left- and right-hand sides in model formula.
Usage:
y ~ model
Arguments:
y, model: symbolic expressions.
Details:
The left-hand side is optional, and one-sided formulae are used in some contexts.
References:
Chambers, J. M. and Hastie, T. J. (1992) Statistical models. Chapter 2 of Statistical Models in S, eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole.
See Also:
'formula'
```



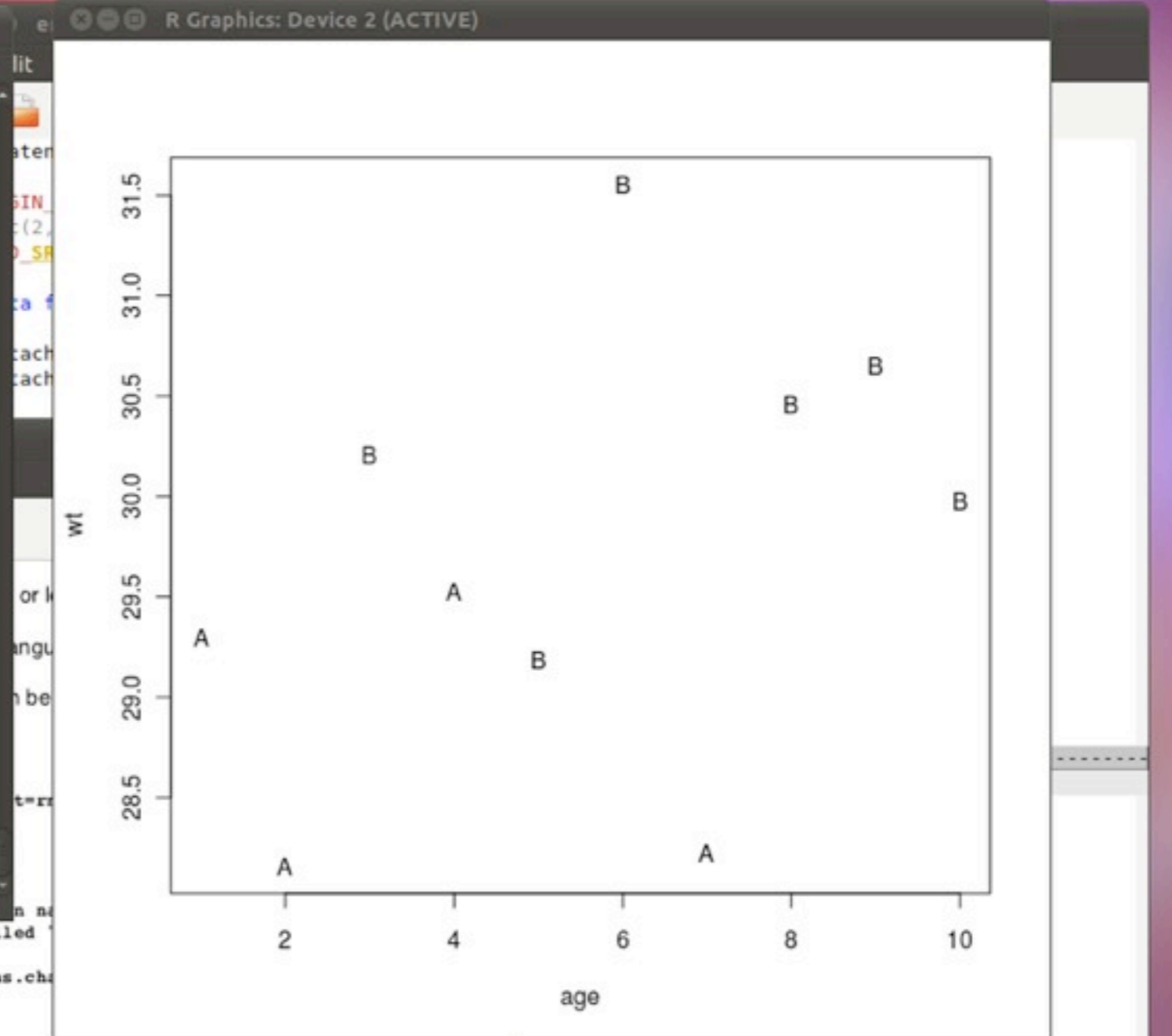
```
> my.data$wt # access the column called "wt" using the dollar sign
> plot(my.data$age, my.data$wt)
> plot(wt ~ age, data=my.data, pch=as.character(grp)) # what just happened?
> age
> attach(my.data)
> age # what just happened? Type "help('attach')".
> plot(grp, wt) # this is how factors affect plotting
> t.test(wt[grp=="A"], wt[grp=="B"]) # but we're getting ahead of ourselves
```



```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
> mydata
Error: object 'mydata' not found
> my.data
  age      wt grp
1  1 29.29847  A
2  2 28.15915  A
3  3 30.20436  B
4  4 29.52910  A
5  5 29.18417  B
6  6 31.55178  B
7  7 28.22086  A
8  8 30.45687  B
9  9 30.64960  B
10 10 29.97496  B
> my.data[4,]
  age      wt grp
4  4 29.52910  A
> my.data[,2]
[1] 29.29847 28.15915 30.20436 29.52910 29.18417 31.55178 28.22086 30.45687
[9] 30.64960 29.97496
> names(my.data)
[1] "age" "wt" "grp"
> my.data$wt
[1] 29.29847 28.15915 30.20436 29.52910 29.18417 31.55178 28.22086 30.45687
[9] 30.64960 29.97496
> plot(my.data$age, my.data$wt)
> plot(wt ~
+
> help (~)
Error: unexpected ')' in "help (~)"
> help ("~")
> plot(wt ~ age, data=my.data, pch=as.character(grp))

```



```

> my.data$wt # access the column called
> plot(my.data$age, my.data$wt)
> plot(wt ~ age, data=my.data, pch=as.ch
happened?
> age
> attach(my.data)
> age # what just happened? Type "help('attach')".
> plot(grp, wt) # this is how factors affect plotting
> t.test(wt[grp=='A'], wt[grp=='B']) # but we're getting ahead of
ourselves

```

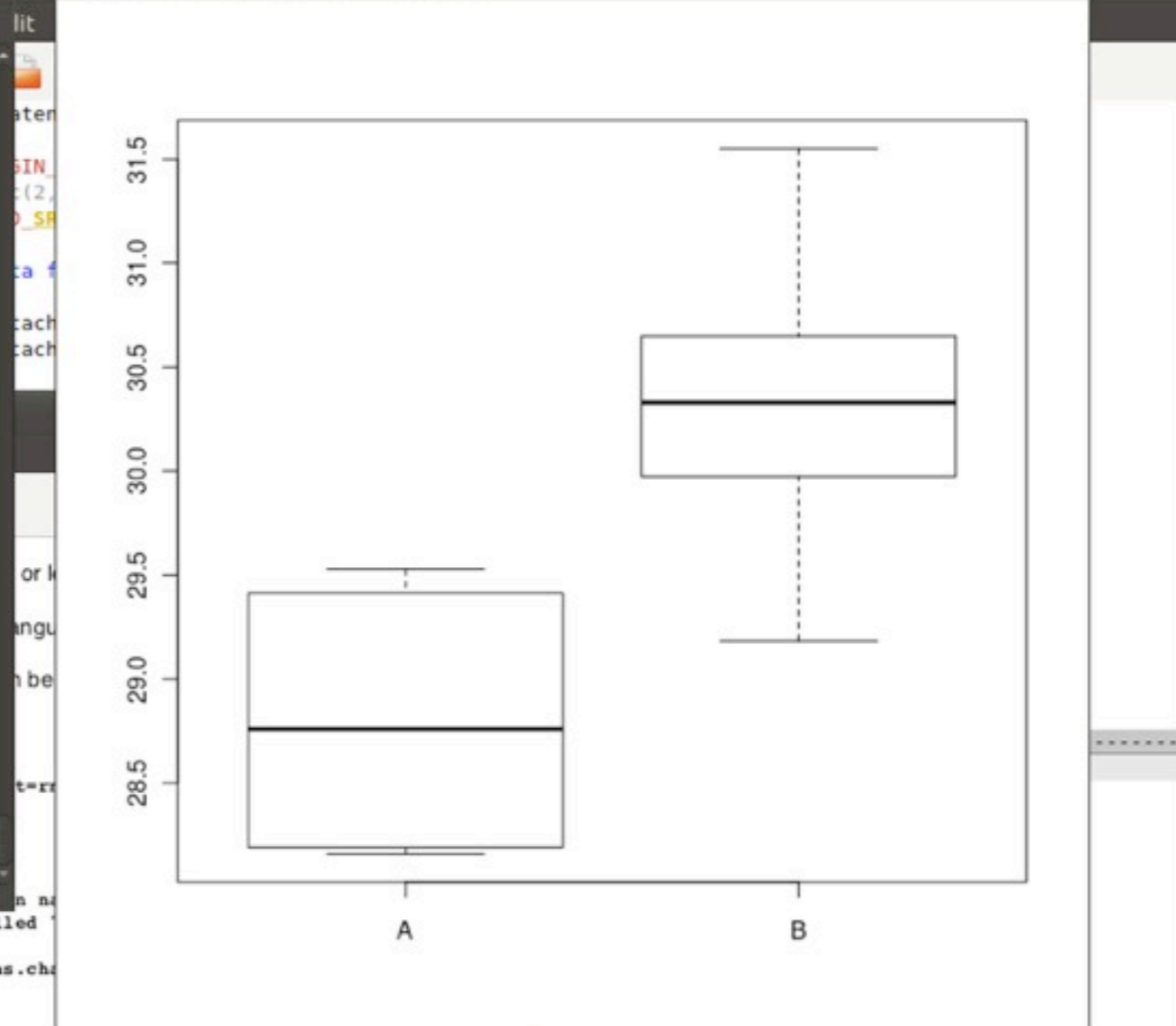
researchtools@ubuntu: ~/class/25

```

File Edit View Search Terminal Help
3 3 30.20436 B
4 4 29.52910 A
5 5 29.18417 B
6 6 31.55178 B
7 7 28.22086 A
8 8 30.45687 B
9 9 30.64960 B
10 10 29.97496 B
> my.data[4,]
age wt grp
4 4 29.52910 A
> my.data[,2]
[1] 29.29847 28.15915 30.20436 29.52910 29.18417 31.55178 28.22086 30.45687
[9] 30.64960 29.97496
> names(my.data)
[1] "age" "wt" "grp"
> my.data$wt
[1] 29.29847 28.15915 30.20436 29.52910 29.18417 31.55178 28.22086 30.45687
[9] 30.64960 29.97496
> plot(my.data$age, my.data$wt)
> plot(wt ~
+
> help (~)
Error: unexpected ')' in "help (~)"
> help ("~")
> plot(wt ~ age, data=my.data, pch=as.character(grp))
> age
Error: object 'age' not found
> attach(my.data)
> age
[1] 1 2 3 4 5 6 7 8 9 10
> plot(grp,wt)

```

R Graphics: Device 2 (ACTIVE)



```

> my.data$wt # access the column called
> plot(my.data$age, my.data$wt)
> plot(wt ~ age, data=my.data, pch=as.ch
happened?
> age
> attach(my.data)
> age # what just happened? Type "help('attach')".
> plot(grp, wt) # this is how factors affect plotting
> t.test(wt[grp=='A'], wt[grp=='B']) # but we're getting ahead of
ourselves

```

```

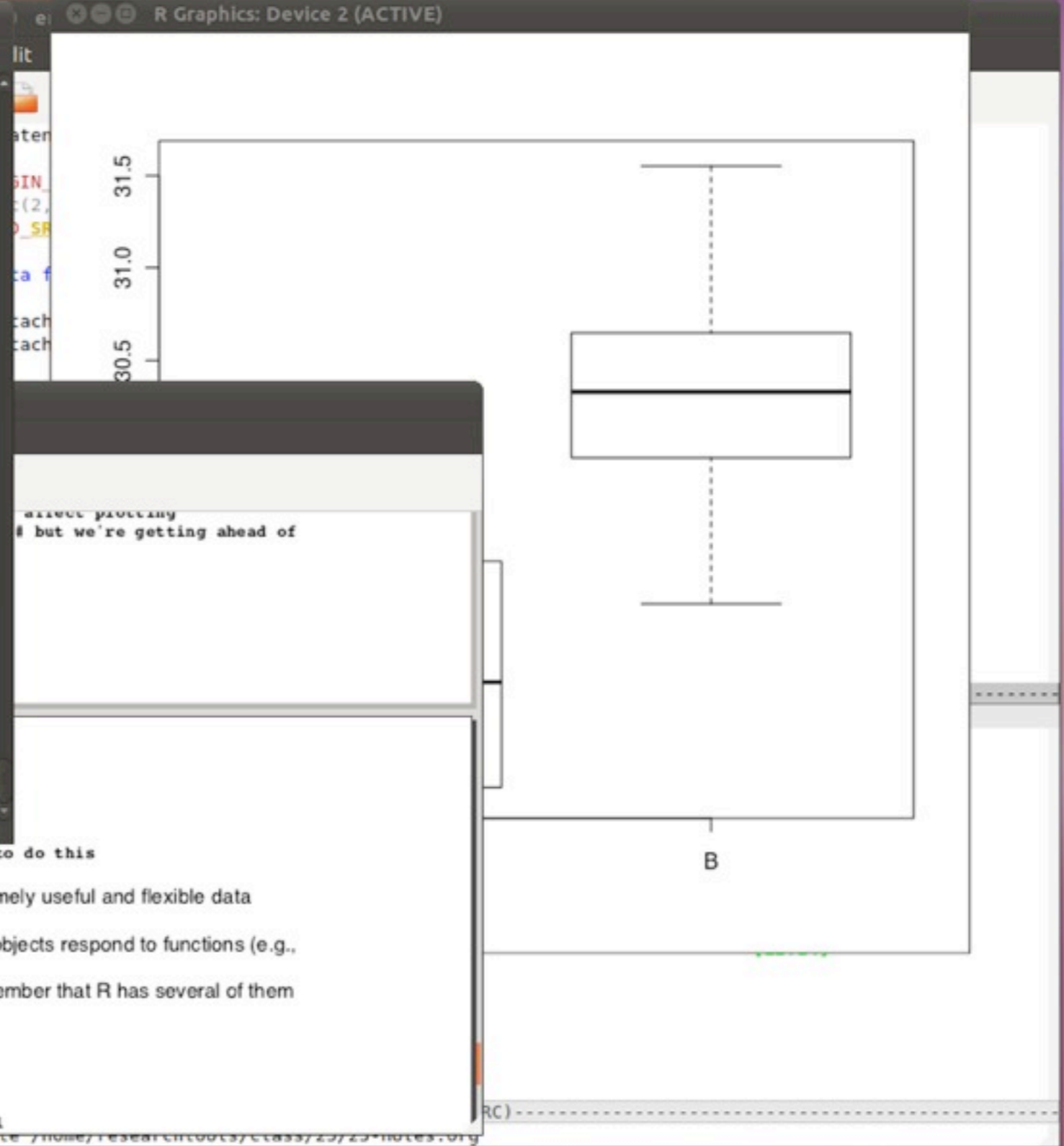
researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
> names(my.data)
[1] "age" "wt" "grp"
> my.data$wt
[1] 29.29847 28.15915 30.20436 29.52910 29.18417 31.55178 28.22086 30.45687
[9] 30.64960 29.97496
> plot(my.data$age, my.data$wt)
> plot(wt ~
+
> help (~)
Error: unexpected ')' in "help (~)"
> help ("~")
> plot(wt ~ age, data=my.data, pch=as.character(grp))
> age
Error: object 'age' not found
> attach(my.data)
> age
[1] 1 2 3 4 5 6 7 8 9 10
> plot(grp,wt)
> t.test(wt[grp=='A'], wt[grp=='B'])

Welch Two Sample t-test

data: wt[grp == "A"] and wt[grp == "B"]
t = -3.2053, df = 7.033, p-value = 0.01486
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-2.6664347 -0.4036889
sample estimates:
mean of x mean of y
28.80190 30.33696

> detach(my.data)
>

```



```

> detach(my.data)# try to remember to do this

As you can see, data frames provide an extremely useful and flexible data
structure. Also,
you might have noticed that different types of objects respond to functions (e.g.,
plot())
differently. Let's try a different data frame, remember that R has several of them
pre-loaded:

> trees
> help(trees)
> names(trees)
> summary(trees) # often very useful

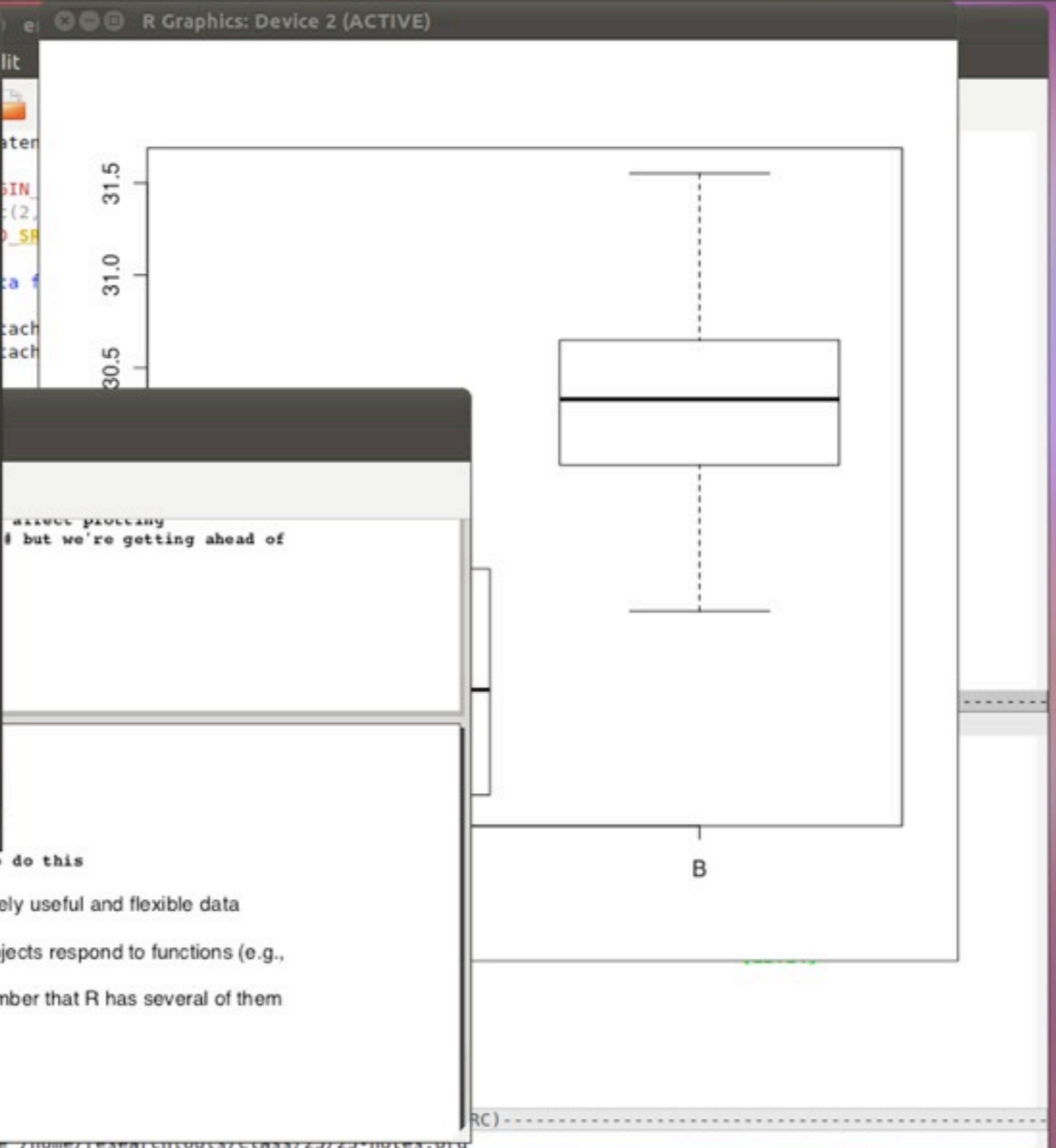
```



```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
2 8.6 65 10.3
3 8.8 63 10.2
4 10.5 72 16.4
5 10.7 81 18.8
6 10.8 83 19.7
7 11.0 66 15.6
8 11.0 75 18.2
9 11.1 80 22.6
10 11.2 75 19.9
11 11.3 79 24.2
12 11.4 76 21.0
13 11.4 76 21.4
14 11.7 69 21.3
15 12.0 75 19.1
16 12.9 74 22.2
17 12.9 85 33.8
18 13.3 86 27.4
19 13.7 71 25.7
20 13.8 64 24.9
21 14.0 78 34.5
22 14.2 80 31.7
23 14.5 74 36.3
24 16.0 72 38.3
25 16.3 77 42.6
26 17.3 81 55.4
27 17.5 82 55.7
28 17.9 80 58.3
29 18.0 80 51.5
30 18.0 80 51.0
31 20.6 87 77.0
> names(trees)
[1] "Girth" "Height" "Volume"
>

```



```

> detach(my.data)# try to remember to do this

As you can see, data frames provide an extremely useful and flexible data
structure. Also,
you might have noticed that different types of objects respond to functions (e.g.,
plot())
differently. Let's try a different data frame, remember that R has several of them
pre-loaded:

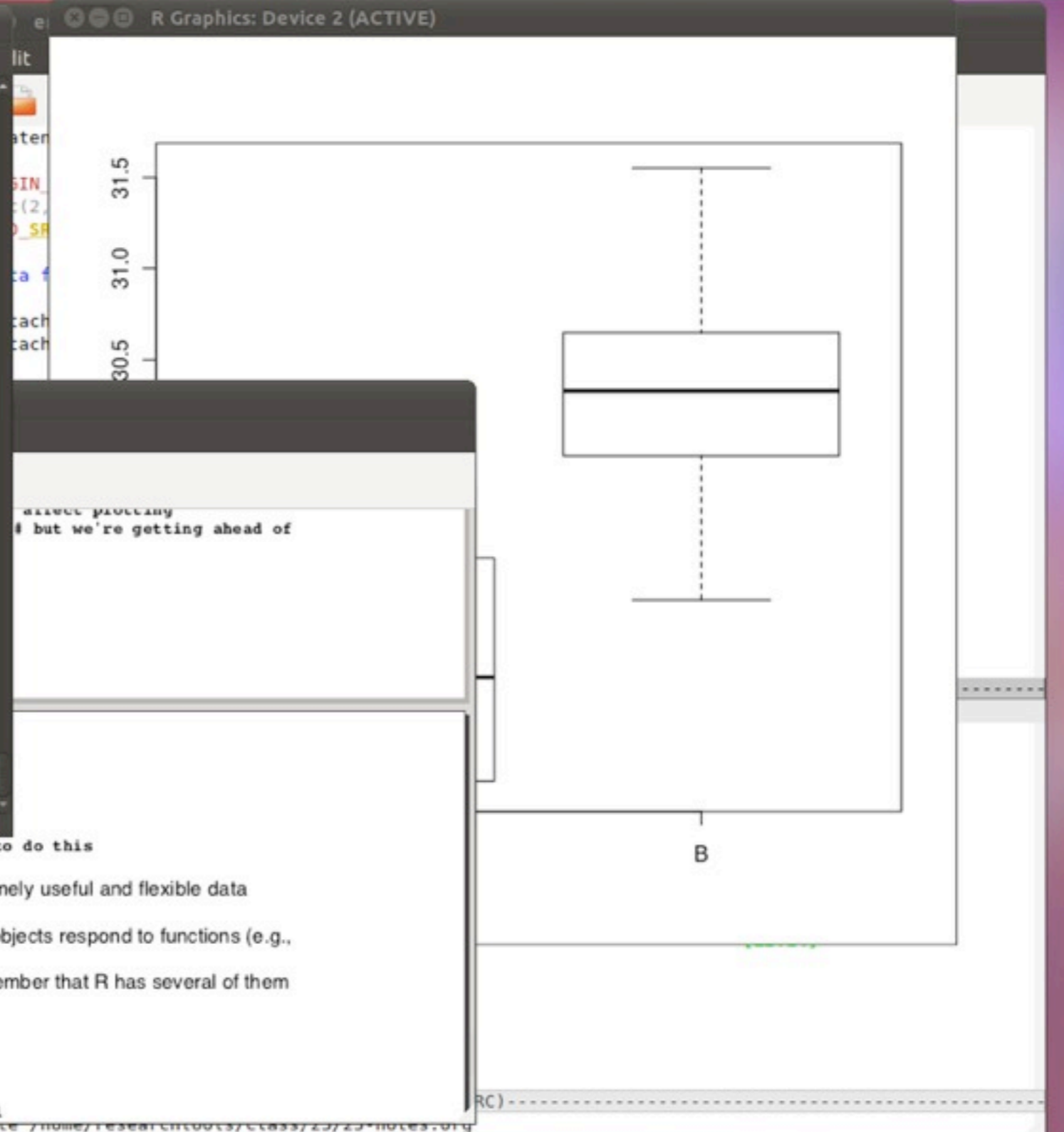
> trees
> help(trees)
> names(trees)
> summary(trees) # often very useful

```

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
20 13.8 64 24.9
21 14.0 78 34.5
22 14.2 80 31.7
23 14.5 74 36.3
24 16.0 72 38.3
25 16.3 77 42.6
26 17.3 81 55.4
27 17.5 82 55.7
28 17.9 80 58.3
29 18.0 80 51.5
30 18.0 80 51.0
31 20.6 87 77.0
> names(trees)
[1] "Girth" "Height" "Volume"
> summary(trees)
  Girth      Height      Volume
Min.   : 8.30   Min.   :63    Min.   :10.20
1st Qu.:11.05  1st Qu.:72    1st Qu.:19.40
Median :12.90  Median :76    Median :24.20
Mean   :13.25  Mean   :76    Mean   :30.17
3rd Qu.:15.25  3rd Qu.:80    3rd Qu.:37.30
Max.   :20.60  Max.   :87    Max.   :77.00
> names(trees)
[1] "Girth" "Height" "Volume"
> summary(trees)
  Girth      Height      Volume
Min.   : 8.30   Min.   :63    Min.   :10.20
1st Qu.:11.05  1st Qu.:72    1st Qu.:19.40
Median :12.90  Median :76    Median :24.20
Mean   :13.25  Mean   :76    Mean   :30.17
3rd Qu.:15.25  3rd Qu.:80    3rd Qu.:37.30
Max.   :20.60  Max.   :87    Max.   :77.00
> help(trees)

```



> detach(my.data)# try to remember to do this

As you can see, data frames provide an extremely useful and flexible data structure. Also, you might have noticed that different types of objects respond to functions (e.g., plot()) differently. Let's try a different data frame, remember that R has several of them pre-loaded:

```

> trees
> help(trees)
> names(trees)
> summary(trees) # often very useful

```

```
22 14.2 80 31.7
23 14.5 74 36.3
24 16.0 72 38.3
25 16.3 77 42.6
26 17.3 81 55.4
27 17.5 82 55.7
28 17.9 80 58.3
29 18.0 80 51.5
30 18.0 80 51.0
31 20.6 87 77.0
```

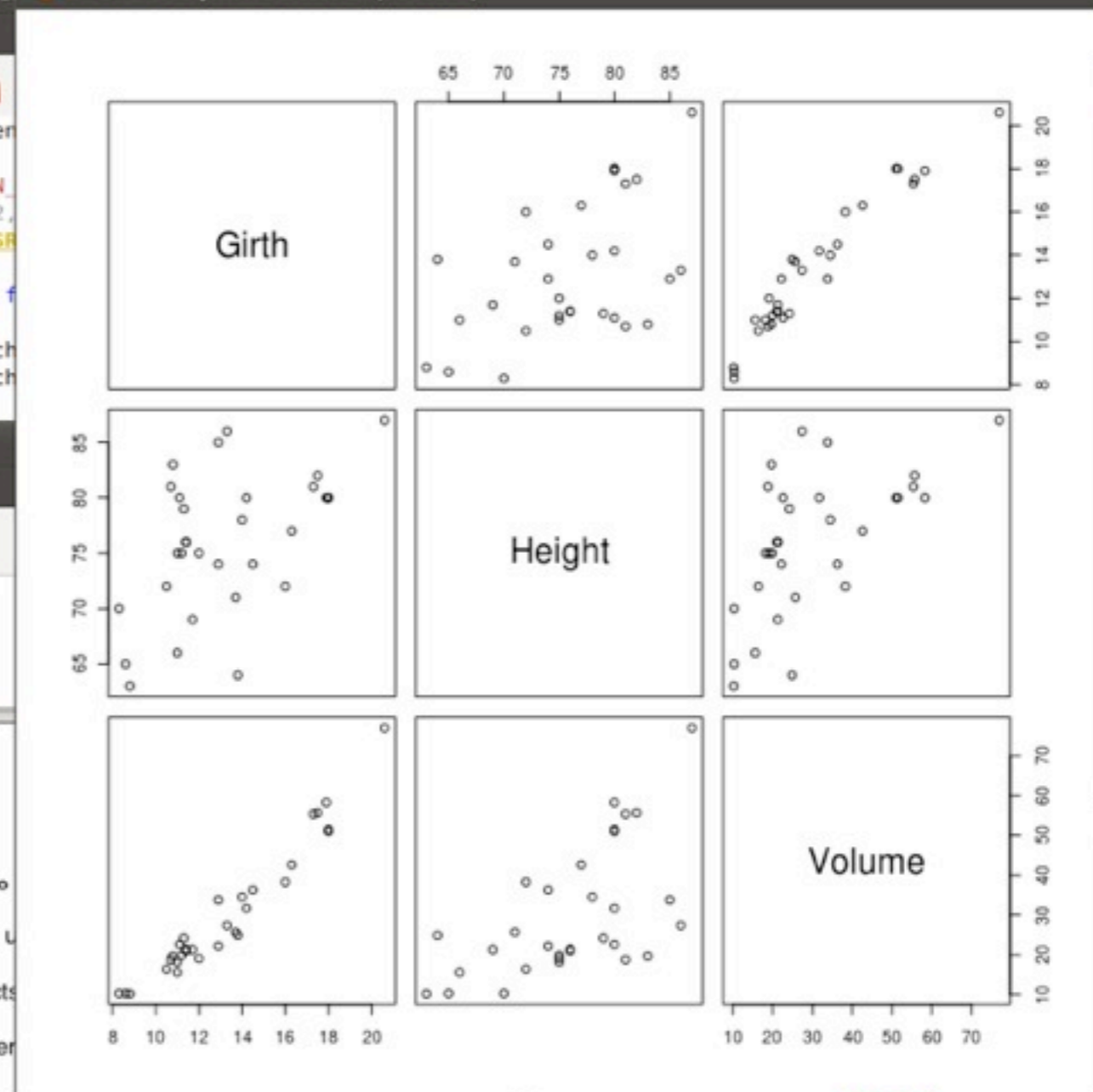
```
> names(trees)
[1] "Girth" "Height" "Volume"
```

```
> summary(trees)
  Girth      Height      Volume
Min.   : 8.30   Min.   :63   Min.   :10.20
1st Qu.:11.05   1st Qu.:72   1st Qu.:19.40
Median :12.90   Median :76   Median :24.20
Mean   :13.25   Mean   :76   Mean   :30.17
3rd Qu.:15.25   3rd Qu.:80   3rd Qu.:37.30
Max.   :20.60   Max.   :87   Max.   :77.00
```

```
> names(trees)
[1] "Girth" "Height" "Volume"
```

```
> summary(trees)
  Girth      Height      Volume
Min.   : 8.30   Min.   :63   Min.   :10.20
1st Qu.:11.05   1st Qu.:72   1st Qu.:19.40
Median :12.90   Median :76   Median :24.20
Mean   :13.25   Mean   :76   Mean   :30.17
3rd Qu.:15.25   3rd Qu.:80   3rd Qu.:37.30
Max.   :20.60   Max.   :87   Max.   :77.00
```

```
> help(trees)
> plot(trees)
```



you might have noticed that different types of objects plot() differently. Let's try a different data frame, remember pre-loaded:

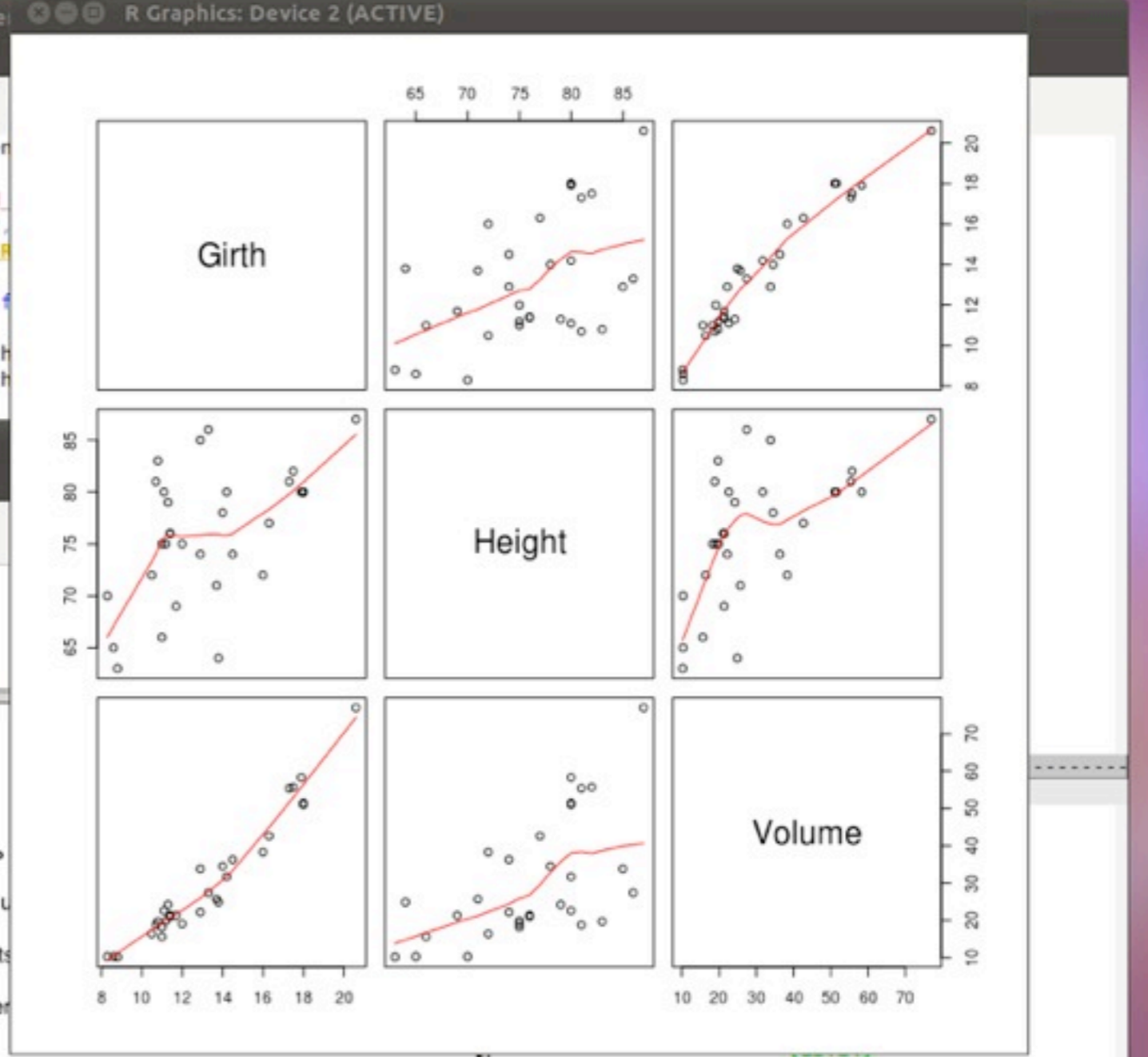
```
> trees
> help(trees)
> names(trees)
> summary(trees) # often very useful
> plot(trees) # actually calls a special plotting method for data frames
> plot(trees, panel=panel.smooth)
> plot(trees$Height, panel=panel.smooth) # see?
```

write ~/home/researchtools/class/25/25-notes.org

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
23 14.5 74 36.3
24 16.0 72 38.3
25 16.3 77 42.6
26 17.3 81 55.4
27 17.5 82 55.7
28 17.9 80 58.3
29 18.0 80 51.5
30 18.0 80 51.0
31 20.6 87 77.0
> names(trees)
[1] "Girth" "Height" "Volume"
> summary(trees)
  Girth      Height      Volume
Min.   : 8.30   Min.   :63   Min.   :10.20
1st Qu.:11.05   1st Qu.:72   1st Qu.:19.40
Median :12.90   Median :76   Median :24.20
Mean   :13.25   Mean   :76   Mean   :30.17
3rd Qu.:15.25   3rd Qu.:80   3rd Qu.:37.30
Max.   :20.60   Max.   :87   Max.   :77.00
> names(trees)
[1] "Girth" "Height" "Volume"
> summary(trees)
  Girth      Height      Volume
Min.   : 8.30   Min.   :63   Min.   :10.20
1st Qu.:11.05   1st Qu.:72   1st Qu.:19.40
Median :12.90   Median :76   Median :24.20
Mean   :13.25   Mean   :76   Mean   :30.17
3rd Qu.:15.25   3rd Qu.:80   3rd Qu.:37.30
Max.   :20.60   Max.   :87   Max.   :77.00
> help(trees)
> plot(trees)
> plot(trees, panel=panel.smooth)

```



```

you might have noticed that different types of objects
plot())
differently. Let's try a different data frame, remember
pre-loaded:

> trees
> help(trees)
> names(trees)
> summary(trees) # often very useful
> plot(trees) # actually calls a special plotting method for data
frames
> plot(trees, panel=panel.smooth)
> plot(trees$Height, panel=panel.smooth) # see?

```

researchtools@ubuntu: ~/class/25

File Edit View Search Terminal Help

```
[1] "Girth" "Height" "Volume"
> summary(trees)
  Girth      Height      Volume
Min.   : 8.30   Min.   :63   Min.   :10.20
1st Qu.:11.05   1st Qu.:72   1st Qu.:19.40
Median :12.90   Median :76   Median :24.20
Mean   :13.25   Mean   :76   Mean   :30.17
3rd Qu.:15.25   3rd Qu.:80   3rd Qu.:37.30
Max.   :20.60   Max.   :87   Max.   :77.00
> names(trees)
[1] "Girth" "Height" "Volume"
> summary(trees)
  Girth      Height      Volume
Min.   : 8.30   Min.   :63   Min.   :10.20
1st Qu.:11.05   1st Qu.:72   1st Qu.:19.40
Median :12.90   Median :76   Median :24.20
Mean   :13.25   Mean   :76   Mean   :30.17
3rd Qu.:15.25   3rd Qu.:80   3rd Qu.:37.30
Max.   :20.60   Max.   :87   Max.   :77.00
> help(trees)
> plot(trees)
> plot(trees, panel=panel.smooth)
> plot(trees$Height, panel=panel.smooth)
Error in plot.default(trees$Height, panel = panel.smooth) :
  argument 2 matches multiple formal arguments
> plot(trees$Height, panel=panel.smooth)
Error in plot.default(trees$Height, panel = panel.smooth) :
  argument 2 matches multiple formal arguments
> help(savehistory)
Error: unexpected symbol in "help savehistory"
> help(savehistory)
> savehistory('class-25-history.r')
>
```

emacs23@ubuntu

File Edit Options Buffers Tools Operate Mark Regexp Immediate Subdir Help

```
/home/researchtools/class/25:
total used in directory 240 available 9875636
drwxr-xr-x  2 researchtools researchtools 4096 2011-11-29 12:21 .
drwxr-xr-x 20 researchtools researchtools 4096 2011-11-29 10:17 ..
-rw-r--r--  1 researchtools researchtools 1424 2011-11-29 12:04 25-notes.org
-rw-r--r--  1 researchtools researchtools  33 2011-11-29 11:13 25-notes.org~
-rw-r--r--  1 researchtools researchtools 74929 2011-11-29 09:31 25-R-lab1-Intro.pdf
-rw-r--r--  1 researchtools researchtools 97334 2011-11-29 09:31 25-R-lab2-CO2.pdf
-rw-r--r--  1 researchtools researchtools 41263 2011-11-29 09:31 25-R-lab3-ANOVA.pdf
lrwxrwxrwx  1 researchtools researchtools  68 2011-11-29 10:17 25-r-statistics.org -> /home/res
searchtools/projects/researchtools/class/25-r-statistics.org
-rw-----  1 researchtools researchtools 3158 2011-11-29 12:21 class-25-history.r
```

-U:%%- 25 All L9 (Dired by name)

```
kurtvm on #unhresearchtools (+,lag:0)
*** #unhresearchtools modes: +
<kurtvm> http://waage.sr.unh.edu/~braswell/eos864/
<matt_w> hi
<kurtvm> how are you all doing with working through the examples? [11:38]
<matt_w> good so far [11:39]
<kurtvm> Please keep asking when you get stuck on things!
<kurtvm> site:r-project.org atan2 [12:02]
<cenglert> everyone seems to ask the question i come upon [12:03]
<kurtvm> site:r-project.org hyperbolic arc tangent [12:04]
<nhassan> I'm good [12:14]
ERC>
```

-U:\*\*- #unhresearchtools@Unknown Bot L45 (ERC)

8

pre-loaded:

```
> trees
> help(trees)
> names(trees)
> summary(trees) # often very
> plot(trees) # actually call
frames
> plot(trees, panel=panel.smo
> plot(trees$Height, panel=pa
```

9

That's enough for now.

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
> plot(trees, panel=panel.smooth)
> plot(trees$Height, panel=panel.smooth)
Error in plot.default(trees$Height, panel = panel.smooth) :
  argument 2 matches multiple formal arguments
> plot(trees$Height, panel=panel.smooth)
Error in plot.default(trees$Height, panel = panel.smooth) :
  argument 2 matches multiple formal arguments
> help savehistory
Error: unexpected symbol in "help savehistory"
> help (savehistory)
> savehistory('class-25-history.r')
> q()
Save workspace image? [y/n/c]: y
researchtools@ubuntu:~/class/25$ ls -ltra
total 264
-rw-r--r-- 1 researchtools researchtools 41263 2011-11-29 09:31 25-R-lab3-ANOVA.pdf
-rw-r--r-- 1 researchtools researchtools 97334 2011-11-29 09:31 25-R-lab2-CO2.pdf
-rw-r--r-- 1 researchtools researchtools 74929 2011-11-29 09:31 25-R-lab1-Intro.pdf
drwxr-xr-x 20 researchtools researchtools 4096 2011-11-29 10:17 ..
lrwxrwxrwx 1 researchtools researchtools 68 2011-11-29 10:17 25-r-statistics.org -> /home/researchtools/projects/researchtools/class/25-r-statistics.org
-rw-r--r-- 1 researchtools researchtools 33 2011-11-29 11:13 25-notes.org-
-rw-r--r-- 1 researchtools researchtools 1424 2011-11-29 12:04 25-notes.org
-rw----- 1 researchtools researchtools 3158 2011-11-29 12:21 class-25-history.r
-rw----- 1 researchtools researchtools 3162 2011-11-29 12:23 .Rhistory
-rw-r--r-- 1 researchtools researchtools 17188 2011-11-29 12:23 .RData
drwxr-xr-x 2 researchtools researchtools 4096 2011-11-29 12:23 .
researchtools@ubuntu:~/class/25$ less .Rhistory
researchtools@ubuntu:~/class/25$

```

```

emacs23@ubuntu
lit Options Buffers Tools ERC Help
/home/researchtools/class/25:
total used in directory 240 available 9875636
-rwxr-xr-x 2 researchtools researchtools 4096 2011-11-29 12:21 .
-rwxr-xr-x 20 researchtools researchtools 4096 2011-11-29 10:17 ..
-r--r--r-- 1 researchtools researchtools 1424 2011-11-29 12:04 25-notes.org
-r--r--r-- 1 researchtools researchtools 33 2011-11-29 11:13 25-notes.org-
-r--r--r-- 1 researchtools researchtools 74929 2011-11-29 09:31 25-R-lab1-Intro.pdf
-r--r--r-- 1 researchtools researchtools 97334 2011-11-29 09:31 25-R-lab2-CO2.pdf
-r--r--r-- 1 researchtools researchtools 41263 2011-11-29 09:31 25-R-lab3-ANOVA.pdf
-rwxrwxrwx 1 researchtools researchtools 68 2011-11-29 10:17 25-r-statistics.org -> /home/researchtools/projects/researchtools/class/25-r-statistics.org
-rw-r--r-- 1 researchtools researchtools 3158 2011-11-29 12:21 class-25-history.r

```

do this

ely useful and flexible data

jects respond to functions (e.g.,

number that R has several of them

ial plotting method for data

```

name) -----
54/
ough the examples? [11:38]
on things! [11:39]
come upon [12:02]
ent [12:03]
[12:04]
[12:14]
can use this command at the end [12:22]
ERC) -----

```



```

> plot(trees, panel=panel.smooth)
> plot(trees$Height, panel=panel.smooth) # see?

That's enough for now.

> q()

```

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
29 18.0      80  51.5
30 18.0      80  51.0
31 20.6      87  77.0
> plot(trees$Height, panel=panel.smooth)
Error in plot.default(trees$Height, panel = panel.smooth) :
  argument 2 matches multiple formal arguments
> def.off()
Error: could not find function "def.off"
> dev.off()
Error in dev.off() : cannot shut down device 1 (the null device)
> pdf('plot.pdf')
> plot(rnorm(100))
> dev.off()
null device
      1
> ^Z
[1]+  Stopped                  R
researchtools@ubuntu:~/class/25$ ls -l
total 244
-rw-r--r-- 1 researchtools researchtools 1424 2011-11-29 12:04 25-notes.org
-rw-r--r-- 1 researchtools researchtools   33 2011-11-29 11:13 25-notes.org-
-rw-r--r-- 1 researchtools researchtools 74929 2011-11-29 09:31 25-R-lab1-Intro.
pdf
-rw-r--r-- 1 researchtools researchtools 97334 2011-11-29 09:31 25-R-lab2-C02.pd
f
-rw-r--r-- 1 researchtools researchtools 41263 2011-11-29 09:31 25-R-lab3-ANOVA.
pdf
lrwxrwxrwx 1 researchtools researchtools   68 2011-11-29 10:17 25-r-statistics.
org -> /home/researchtools/projects/researchtools/class/25-r-statistics.org
-rw----- 1 researchtools researchtools  3158 2011-11-29 12:21 class-25-history
.r
-rw-r--r-- 1 researchtools researchtools  9097 2011-11-29 12:26 plot.pdf
researchtools@ubuntu:~/class/25$

```


```

emacs23@ubuntu
lit Options Buffers Tools ERC Help
/home/researchtools/class/25:
total used in directory 240 available 9875636
-rwxr-xr-x  2 researchtools researchtools  4096 2011-11-29 12:21 .
-rwxr-xr-x 20 researchtools researchtools  4096 2011-11-29 10:17 ..
-rw-r--r--  1 researchtools researchtools  1424 2011-11-29 12:04 25-notes.org
-rw-r--r--  1 researchtools researchtools    33 2011-11-29 11:13 25-notes.org-
-rw-r--r--  1 researchtools researchtools 74929 2011-11-29 09:31 25-R-lab1-Intro.pdf
-rw-r--r--  1 researchtools researchtools 97334 2011-11-29 09:31 25-R-lab2-C02.pdf
-rw-r--r--  1 researchtools researchtools 41263 2011-11-29 09:31 25-R-lab3-ANOVA.pdf
-rwxrwxrwx  1 researchtools researchtools   68 2011-11-29 10:17 25-r-statistics.org -> /home/res
tools/projects/researchtools/class/25-r-statistics.org
-rw-----  1 researchtools researchtools  3158 2011-11-29 12:21 class-25-history.r

25          All L11      (Dired by name)-----
on #unhresearchtools (+,lag:0)
kurtvm> site:r-project.org hyperbolic arc tangent           [12:04]
kurtvm> I'm good                                           [12:14]
kurtvm> if you want to save your history, you can use this command at the end
of class:                                                  [12:22]
kurtvm> savehistory('class-25-history.log')
<kjerram> Anyone having trouble with the last "plot" command for the "trees"
data frame?                                              [12:23]
<matt_w> yes
<kurtvm> me too                                           [12:24]
*** bwelton (~chatzill@lab6.ccom.nh) has quit: Client closed connection
ERC>
-U:**- #unhresearchtools@Unknown Bot L53 (ERC)-----

```

8



9

```

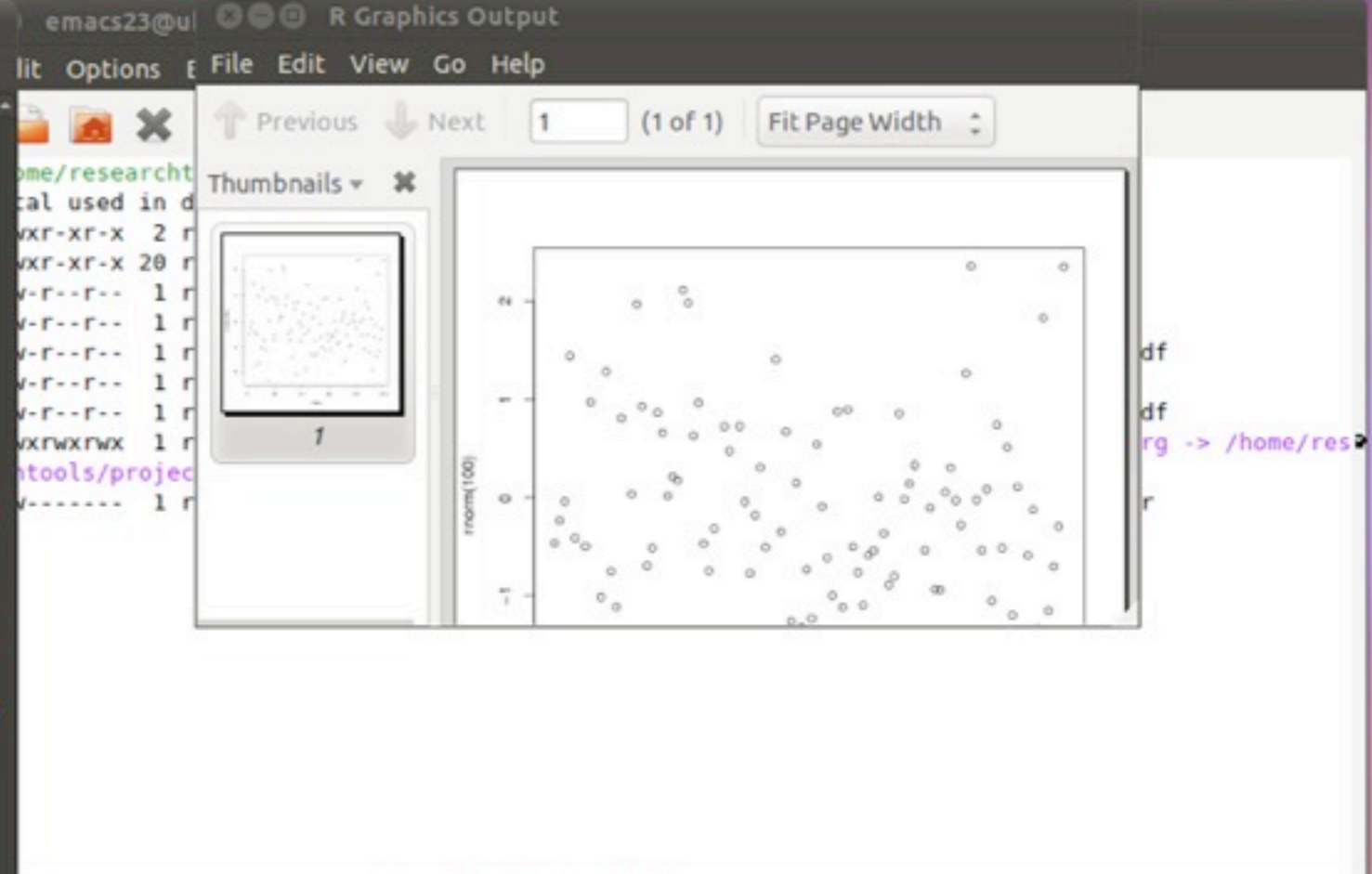
> plot(trees, panel=panel.smooth)
> plot(trees$Height, panel=panel.smooth)
That's enough for now.
> q()

```

```

researchtools@ubuntu: ~/class/25
File Edit View Search Terminal Help
Error in dev.off() : cannot shut down device 1 (the null device)
> pdf('plot.pdf')
> plot(rnorm(100))
> dev.off()
null device
1
> ^Z
[1]+  Stopped                  R
researchtools@ubuntu:~/class/25$ ls -l
total 244
-rw-r--r-- 1 researchtools researchtools 1424 2011-11-29 12:04 25-notes.org
-rw-r--r-- 1 researchtools researchtools 33 2011-11-29 11:13 25-notes.org~
-rw-r--r-- 1 researchtools researchtools 74929 2011-11-29 09:31 25-R-lab1-Intro.
pdf
-rw-r--r-- 1 researchtools researchtools 97334 2011-11-29 09:31 25-R-lab2-C02.pdf
f
-rw-r--r-- 1 researchtools researchtools 41263 2011-11-29 09:31 25-R-lab3-ANOVA.
pdf
lrwxrwxrwx 1 researchtools researchtools 68 2011-11-29 10:17 25-r-statistics.
org -> /home/researchtools/projects/researchtools/class/25-r-statistics.org
-rw----- 1 researchtools researchtools 3158 2011-11-29 12:21 class-25-history
.r
-rw-r--r-- 1 researchtools researchtools 9097 2011-11-29 12:26 plot.pdf
researchtools@ubuntu:~/class/25$ open plot.pdf
Couldn't get a file descriptor referring to the console
researchtools@ubuntu:~/class/25$ xdg-open plot.pdf
researchtools@ubuntu:~/class/25$ fg
R
> plot(trees$Height, panel=panel.smooth)
Error in plot.default(trees$Height, panel = panel.smooth) :
  argument 2 matches multiple formal arguments
>

```



```

25 All L11 (Dired by name)-----
on #unhresearchtools (+,lag:0)
vm> site:r-project.org hyperbolic arc tangent [12:04]
san> I'm good [12:14]
vm> if you want to save your history, you can use this command at the end
of class: [12:22]
vm> savehistory('class-25-history.log')
<kjerram> Anyone having trouble with the last "plot" command for the "trees"
data frame? [12:23]
<matt_w> yes
<kurtvm> me too [12:24]
*** bwelton (~chatzill@lab6.ccom.nh) has quit: Client closed connection
ERC>
-U:**- #unhresearchtools@Unknown Bot L53 (ERC)-----

```

```

> plot(trees, panel=panel.smooth)
> plot(trees$Height, panel=panel.smooth)
That's enough for now.
> q()

```



```
researchtools@ubuntu: ~/class/screencapture/cap-25
File Edit View Search Terminal Help
123141.png
123147.png
123152.png
123158.png
123203.png
123209.png
123214.png
123220.png
123225.png
123231.png
123236.png
123241.png
123247.png
123252.png
123258.png
123303.png
123309.png
123314.png
123320.png
123325.png
123331.png
```

```
> help(plot-data.frame)
Error in help(plot - data.frame) :
'topic' should be a name, length-one character vector or reserved word
> plot(Height, Volume)
> plot(Height)
> savehistory('class-25-history-2.log')
>
```

```
emacs23@ubuntu
lit Options Buffers Tools ERC Help
atenate:
GIN_SRC r
(2,4,5,60)
D_SRC
ca frames
cach
cach
arching for help
chis with Google searches:
GIN_EXAMPLE
r-project.org hyperbolic arc tangent
D_SRC
```

```
-U:--- 25-notes.org Bot L69 (Org Fly)-----
kurtvm on #unhresearchtools (+,lag:0)
<kurtvm> site:r-project.org hyperbolic arc tangent [12:04]
<nhassan> I'm good [12:14]
<kurtvm> if you want to save your history, you can use this command at the end
of class: [12:22]
<kurtvm> savehistory('class-25-history.log')
<kjerram> Anyone having trouble with the last "plot" command for the "trees"
data frame? [12:23]
<matt_w> yes
<kurtvm> me too [12:24]
*** bwelton (-chatzill@lab6.ccom.nh) has quit: Client closed connection
<nhassan> same error [12:28]
*** ygh2 (-chatzill@lab10.ccom.nh) has quit: "ChatZilla 0.9.87 [Firefox
6.0.1/20110830202826]" [12:32]
*** hminami (-chatzill@192.168.8.245) has quit: "ChatZilla 0.9.87 [Firefox
6.0.2/20110905174115]" [12:33]
<kjerram> thanks!
-U:***- #unhresearchtools@Unknown 61% L47 (ERC)-----
```

