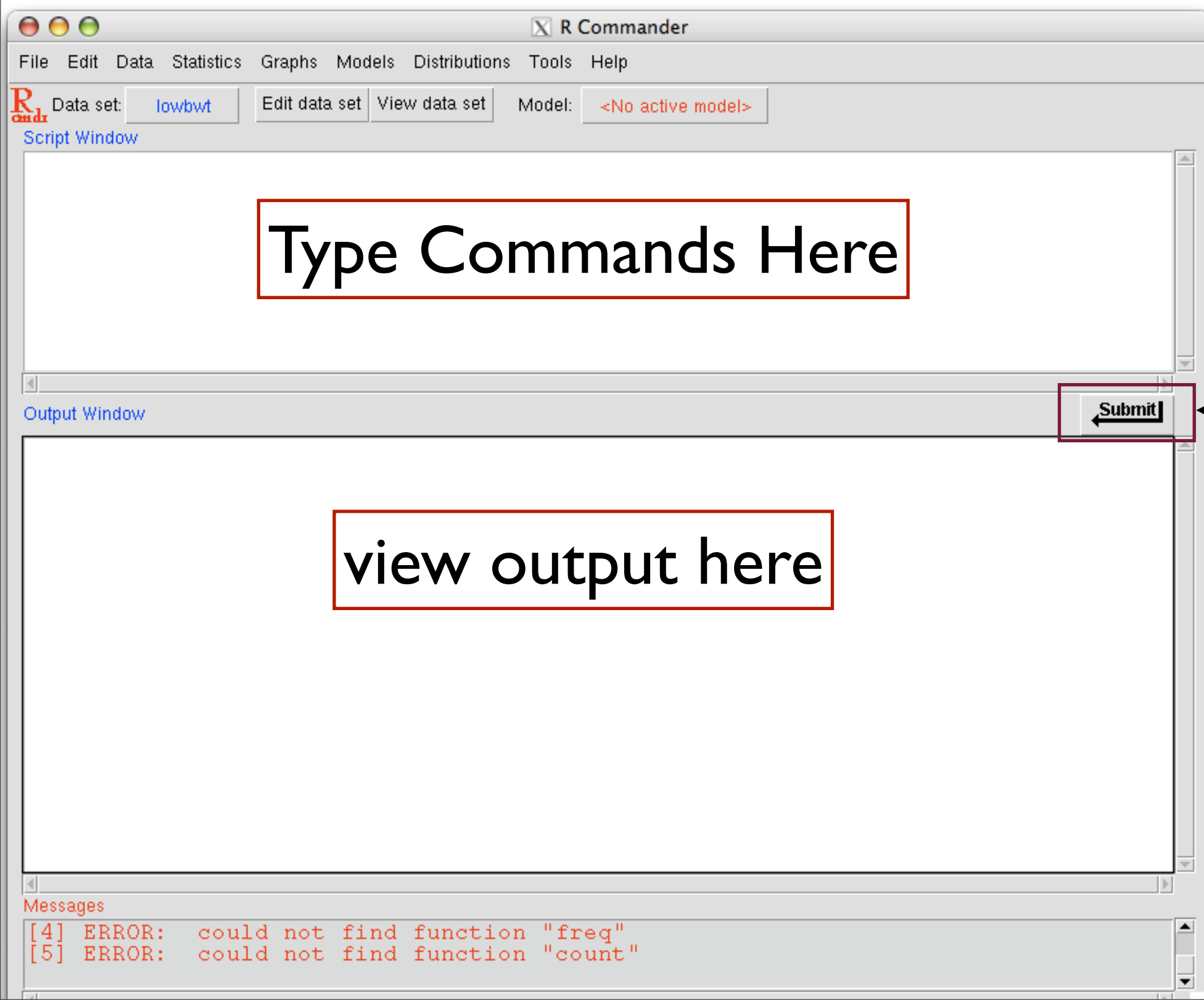


Basic R Commands and editing Rcmdr commands



The screenshot shows the R Commander application window. The menu bar includes File, Edit, Data, Statistics, Graphs, Models, Distributions, Tools, and Help. Below the menu bar, the 'Data set:' dropdown is set to 'lowbwt', and the 'Model:' dropdown is set to '<No active model>'. The 'Script Window' is the main area for entering commands, with a red box and the text 'Type Commands Here' overlaid. The 'Output Window' is below the script window, with a red box and the text 'view output here' overlaid. A 'Submit' button is located to the right of the output window, with a red box and an arrow pointing to it from a text box on the right. The 'Messages' window at the bottom shows two error messages: '[4] ERROR: could not find function "freq"' and '[5] ERROR: could not find function "count"'. A text box on the right side of the image contains the instruction: 'highlight commands with the cursor, then click submit'.

R Commander

File Edit Data Statistics Graphs Models Distributions Tools Help

Data set: lowbwt Edit data set View data set Model: <No active model>

Script Window

Type Commands Here

Output Window

Submit

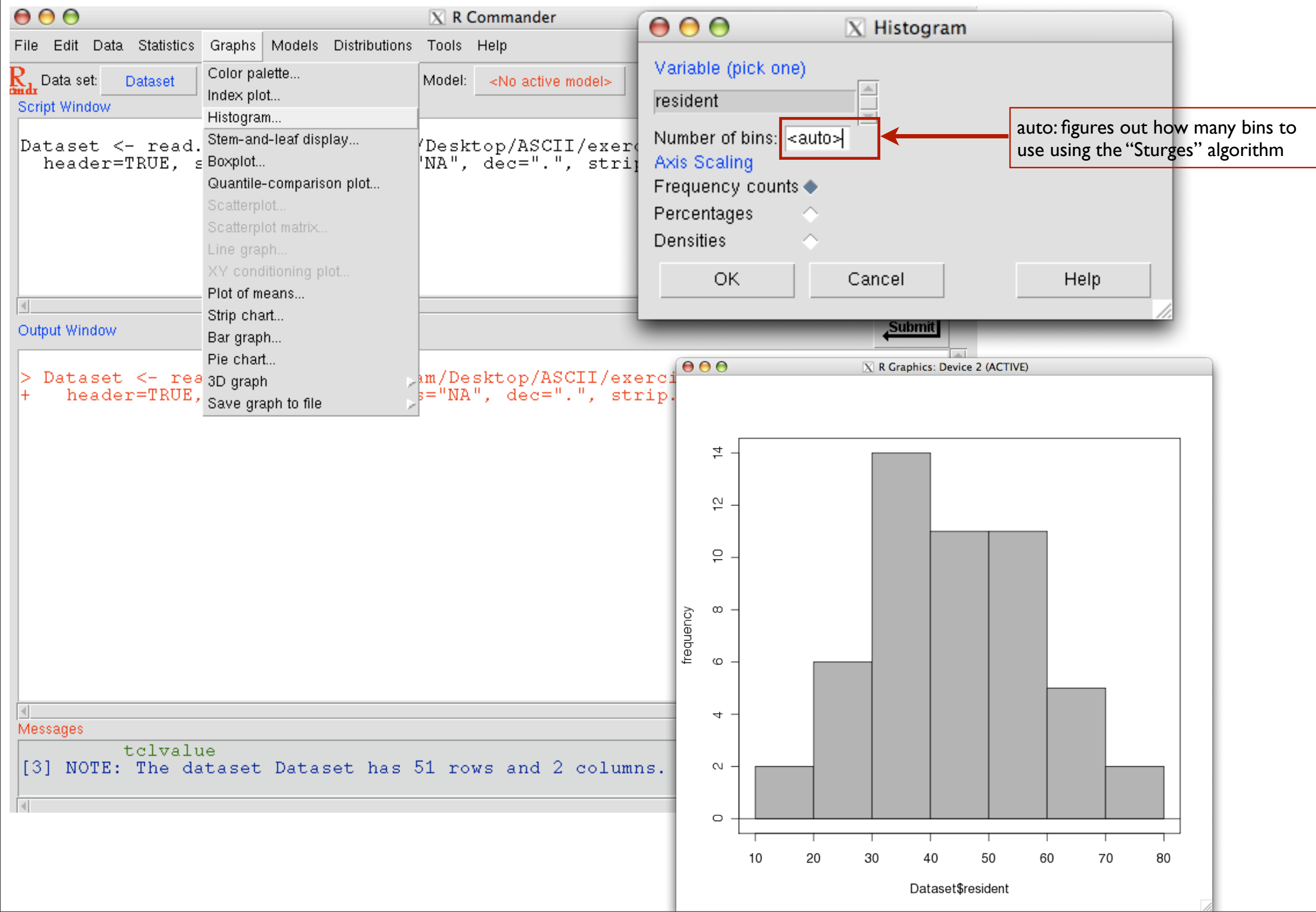
view output here

Messages

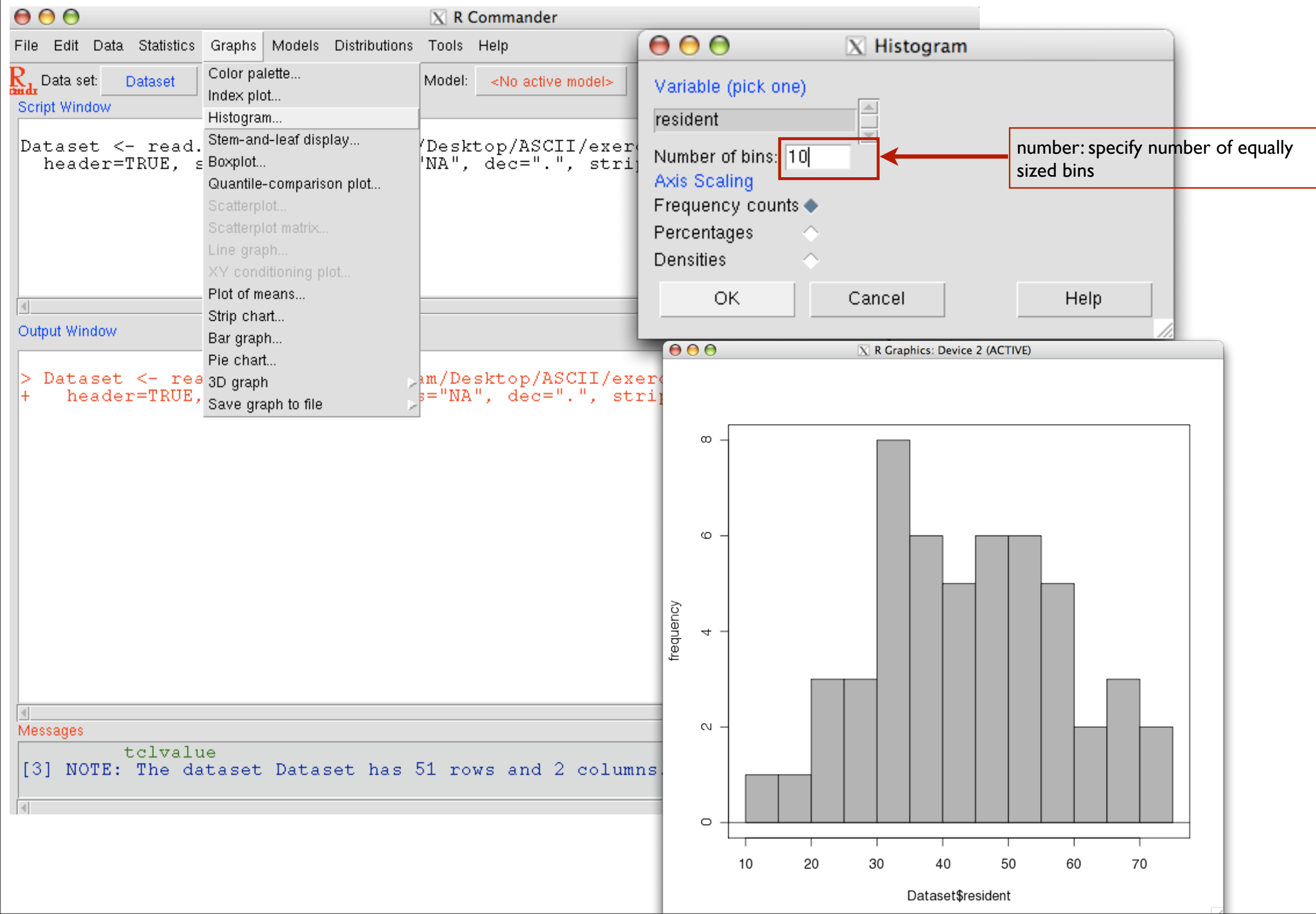
[4] ERROR: could not find function "freq"
[5] ERROR: could not find function "count"

highlight commands with the cursor, then click submit

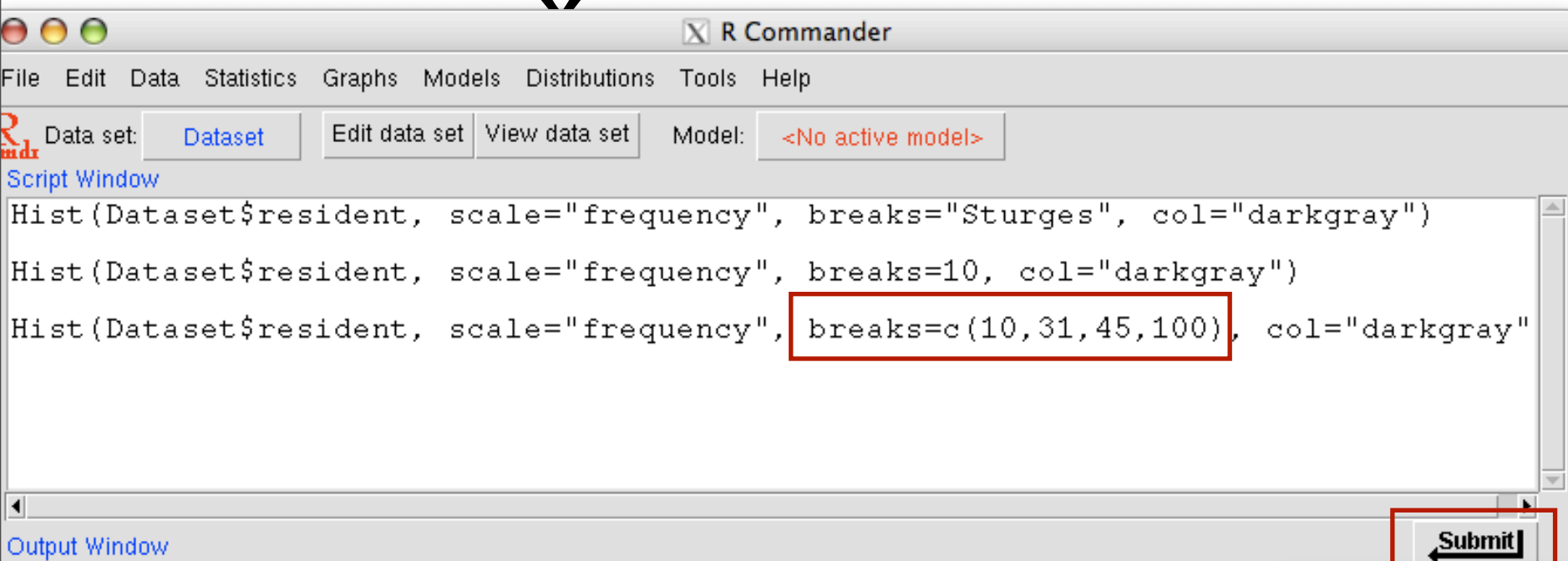
hist()



hist()



hist()



R Commander

File Edit Data Statistics Graphs Models Distributions Tools Help

Data set: Dataset Edit data set View data set Model: <No active model>

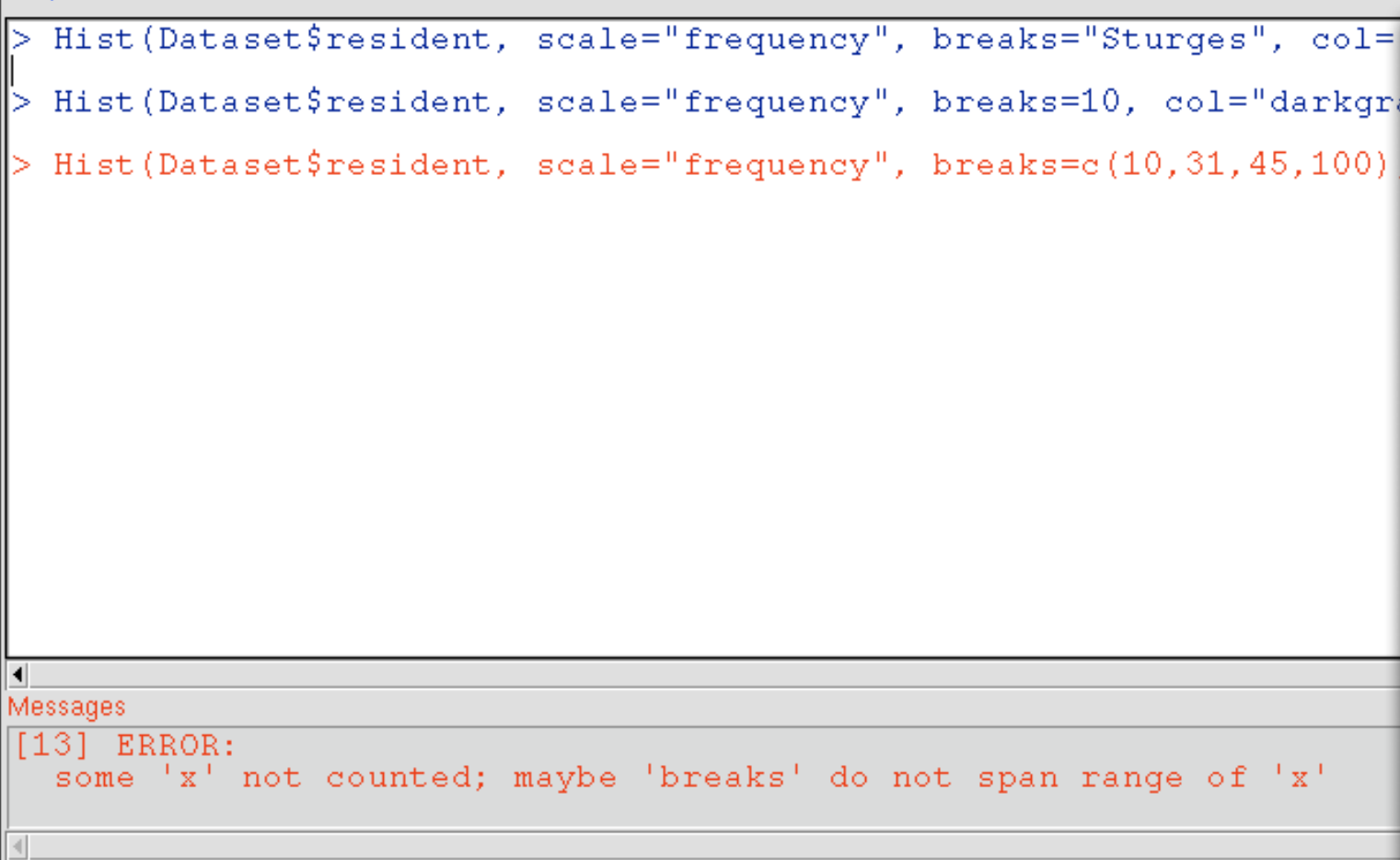
Script Window

```
Hist(Dataset$resident, scale="frequency", breaks="Sturges", col="darkgray")
Hist(Dataset$resident, scale="frequency", breaks=10, col="darkgray")
Hist(Dataset$resident, scale="frequency", breaks=c(10,31,45,100), col="darkgray")
```

Submit

Output Window

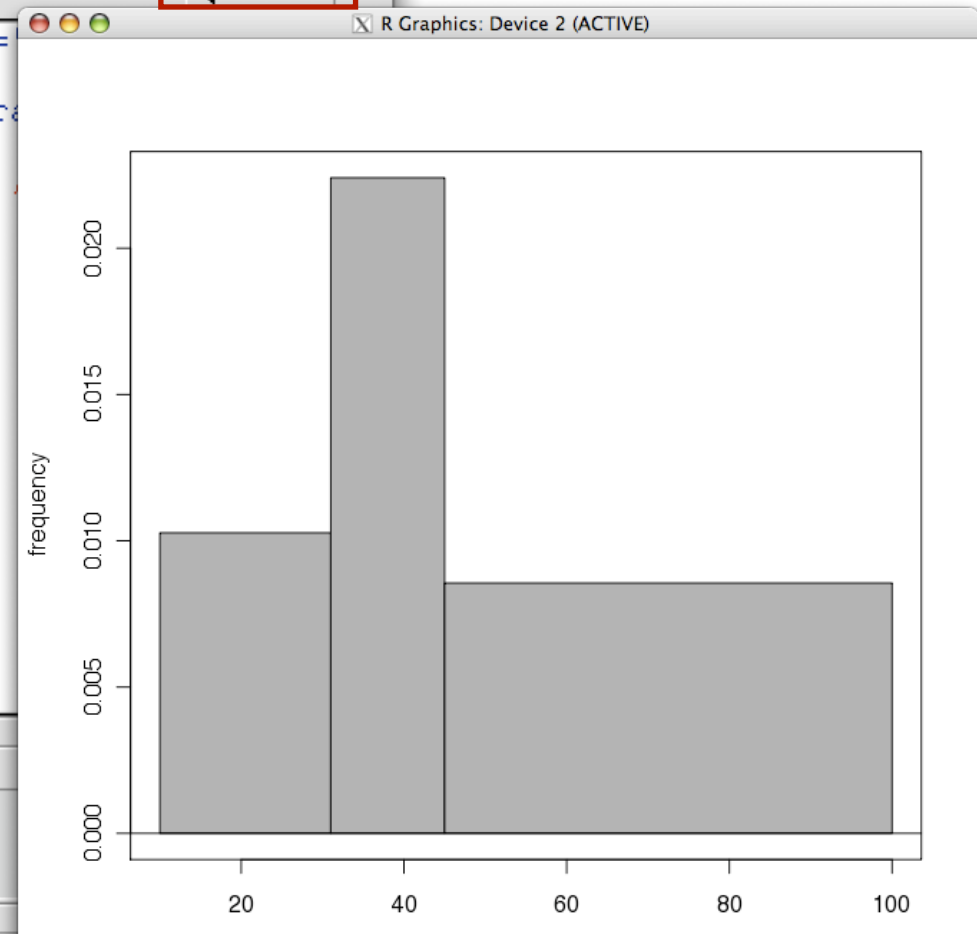
You can edit commands in the Script Window directly. For example, you can set ``breaks'' to arbitrary values to make funny shaped histograms. Though this is not usually recommended



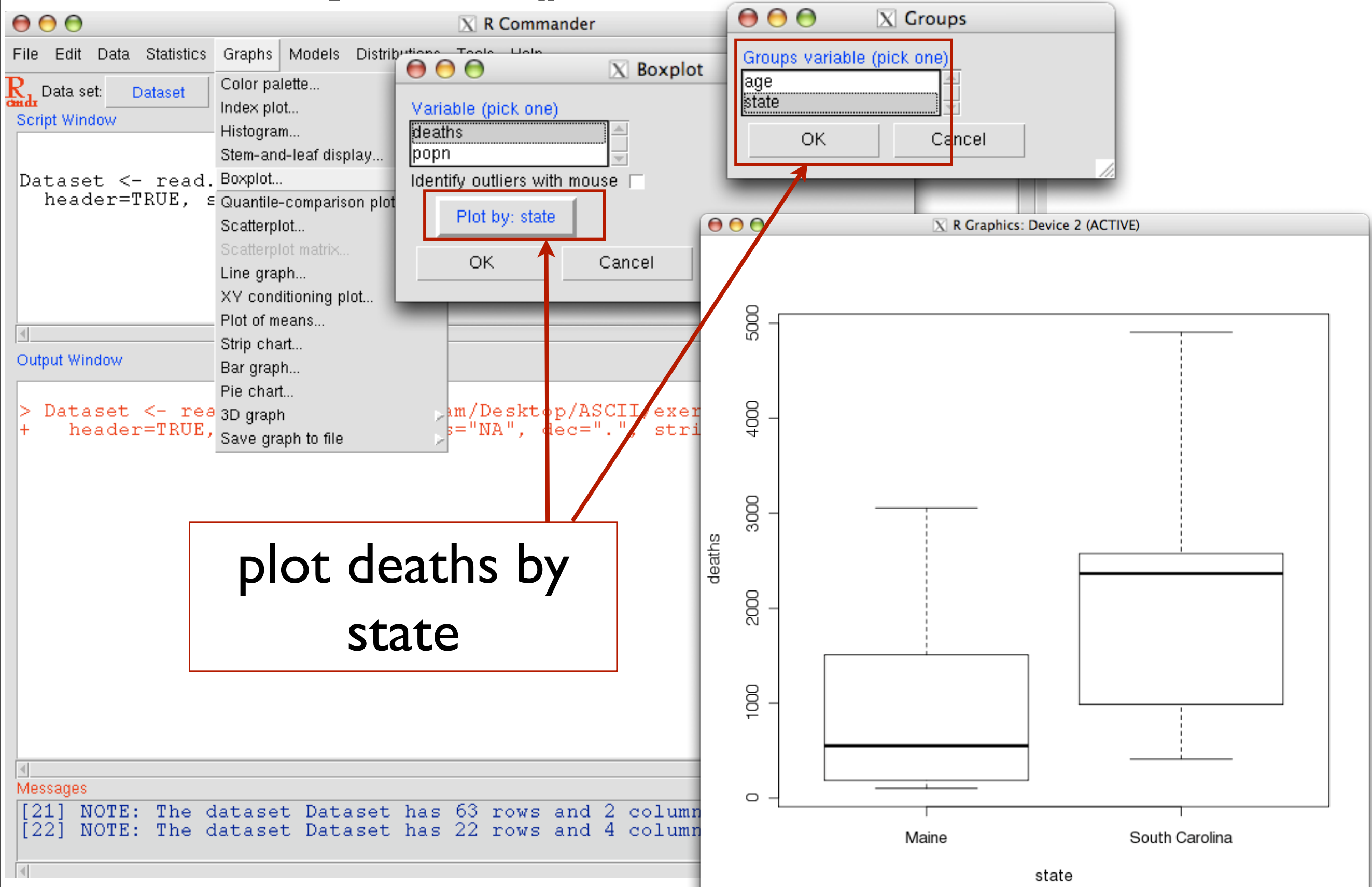
```
> Hist(Dataset$resident, scale="frequency", breaks="Sturges", col="darkgray")
> Hist(Dataset$resident, scale="frequency", breaks=10, col="darkgray")
> Hist(Dataset$resident, scale="frequency", breaks=c(10,31,45,100), col="darkgray")
```

Messages

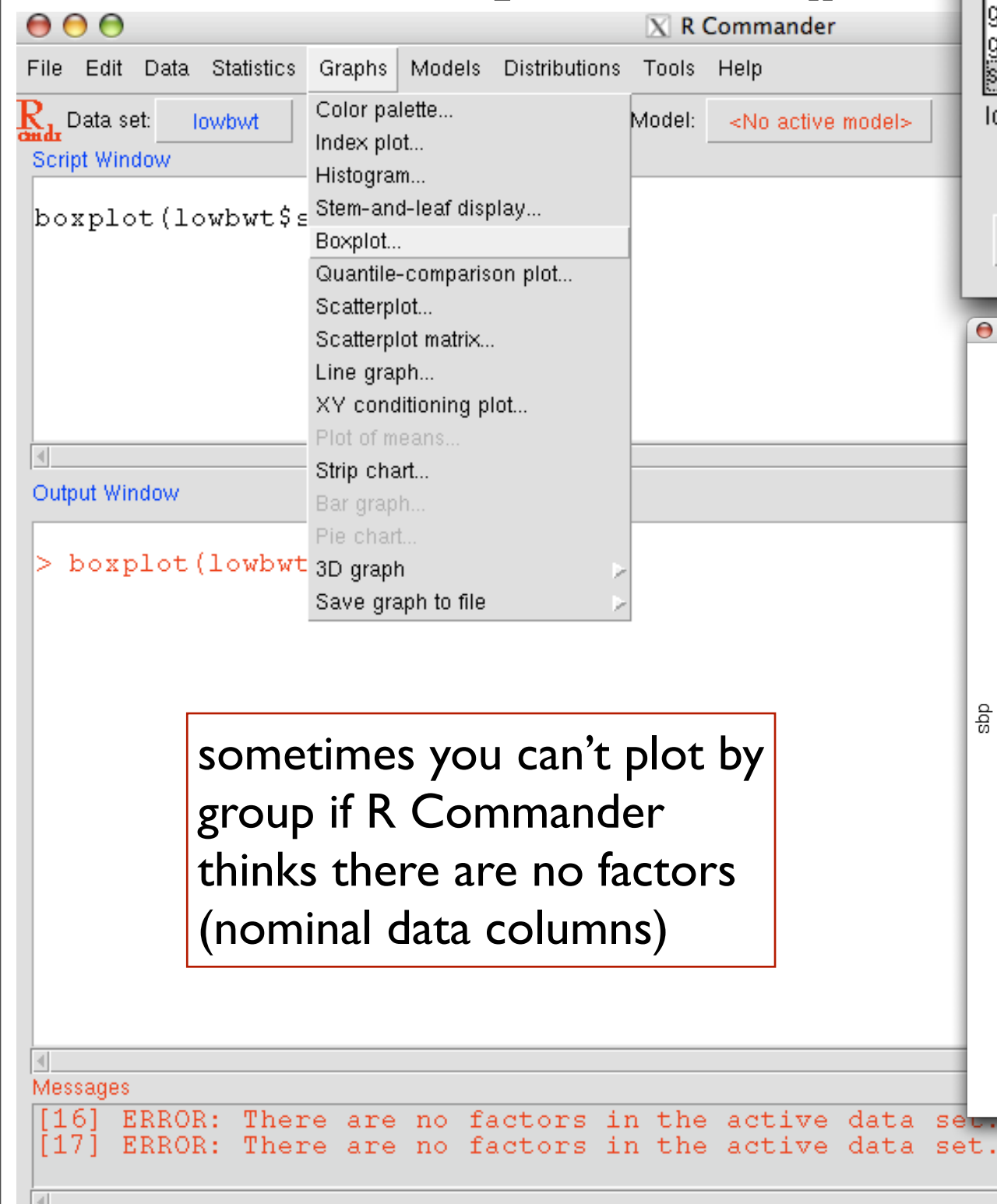
```
[13] ERROR:
      some 'x' not counted; maybe 'breaks' do not span range of 'x'
```



boxplot()



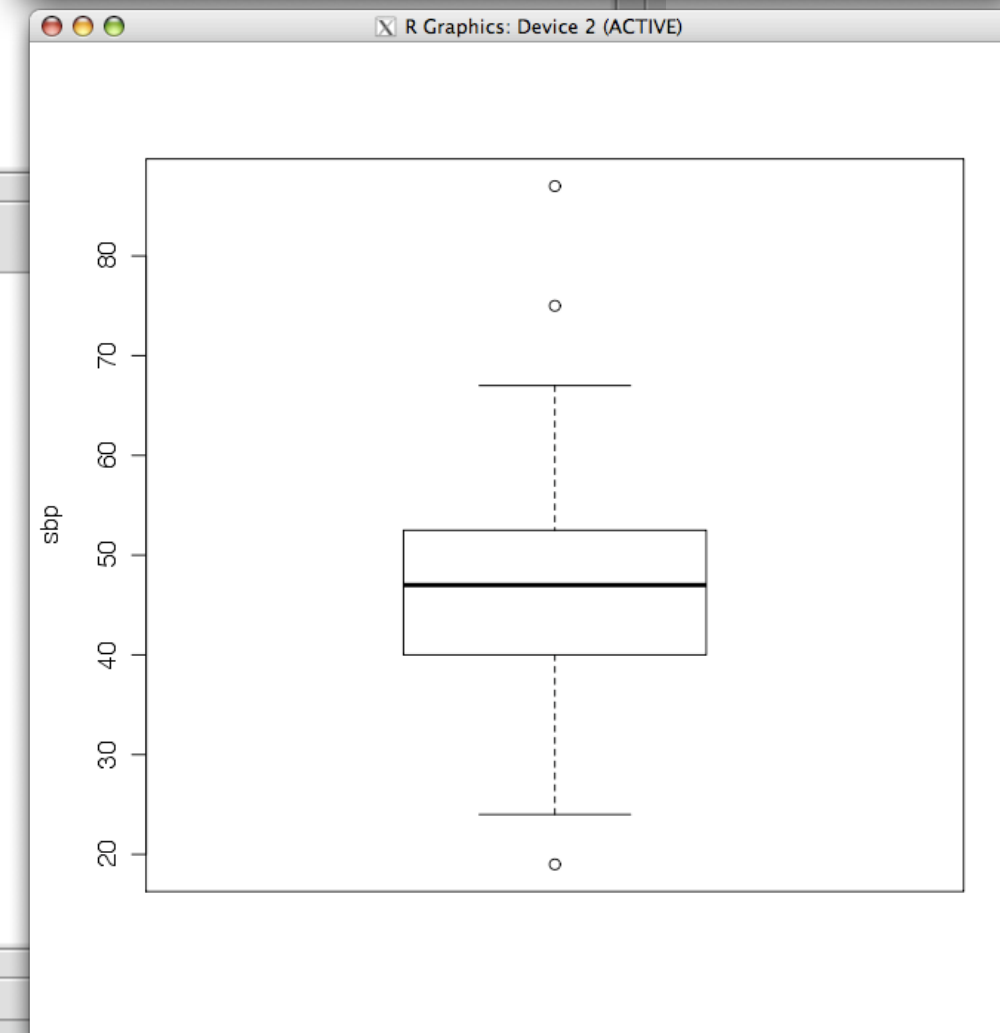
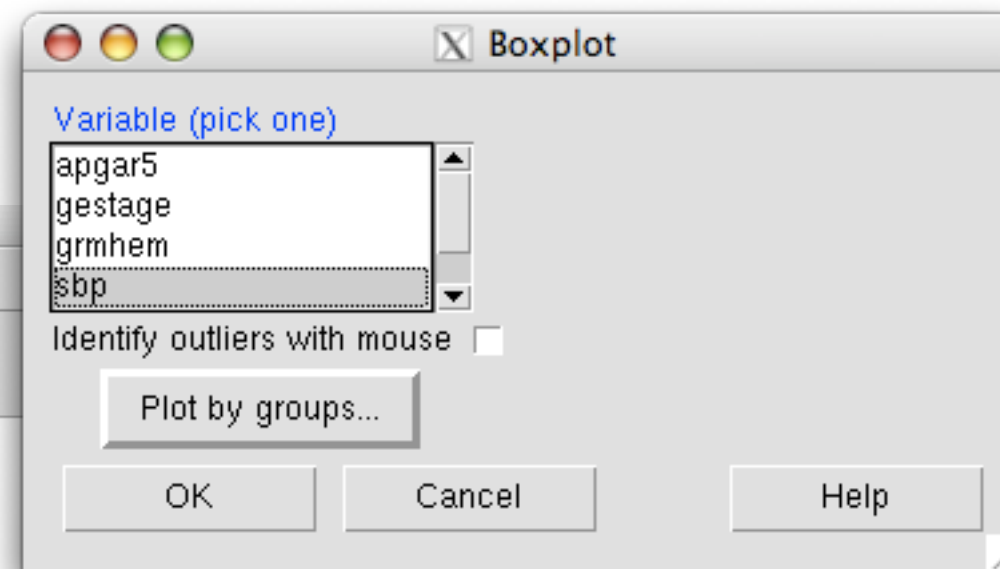
boxplot()



The screenshot shows the R Commander application window. The 'Data set' is 'lowbwt'. The 'Model' is '<No active model>'. The 'Graphs' menu is open, and 'Boxplot...' is selected. The 'Script Window' contains the command `boxplot(lowbwt$`. The 'Output Window' contains the command `> boxplot(lowbwt`. The 'Messages' window at the bottom shows two error messages:

```
[16] ERROR: There are no factors in the active data set.  
[17] ERROR: There are no factors in the active data set.
```

sometimes you can't plot by group if R Commander thinks there are no factors (nominal data columns)



boxplot()

R Commander

File Edit Data Statistics Graphs Models Distributions Tools Help

Data set: **lowbwt** Edit data set View data set Model: **<No active model>**

Script Window

```
boxplot(sbp ~ sex, data=lowbwt, ylab="sbp")
```

plot this

by this group

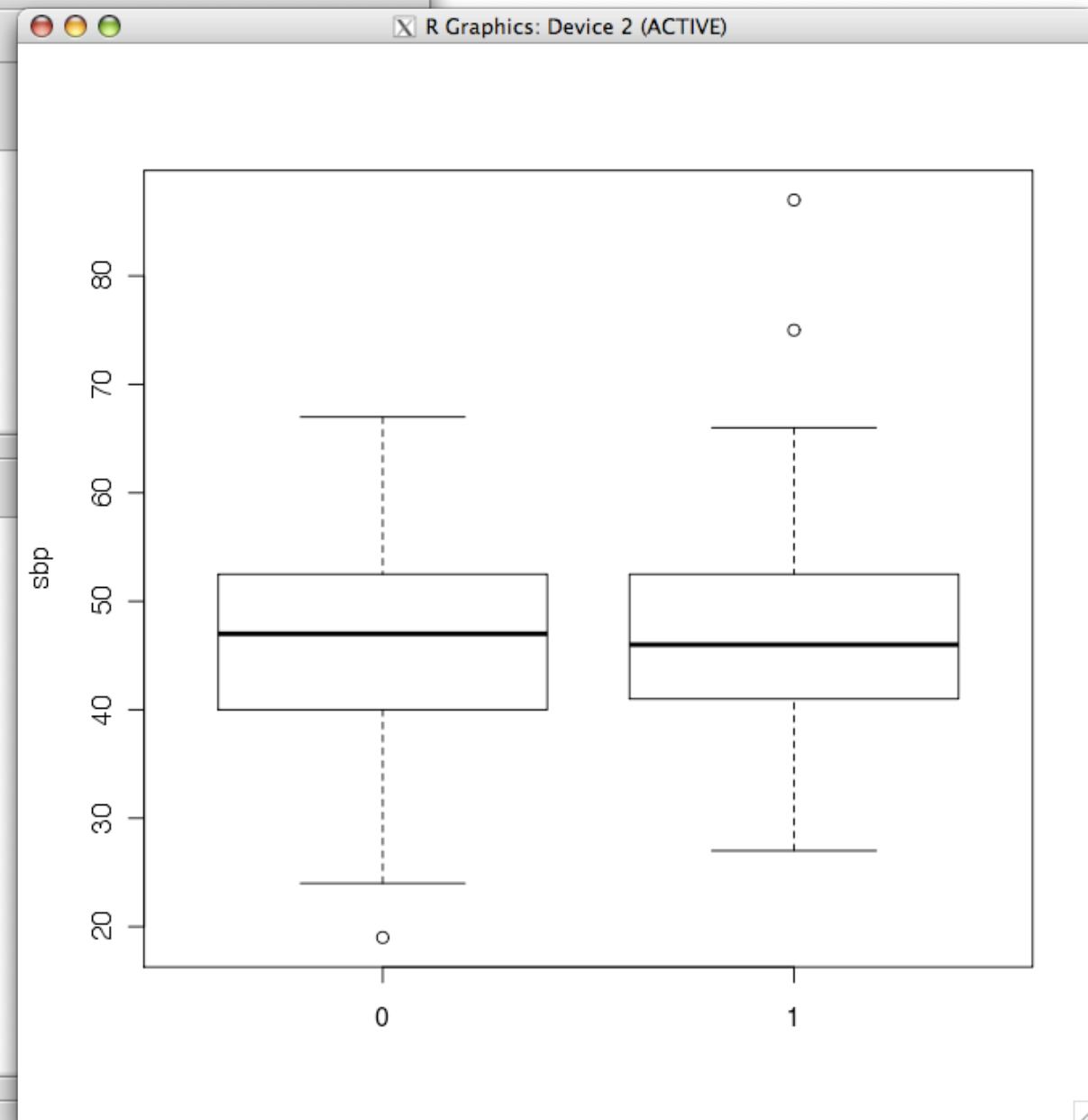
Output Window

```
> boxplot(lowbwt$sbp, ylab="sbp")  
> boxplot(sbp ~ sex, data=lowbwt, ylab="sbp")
```

You can get around this if you directly tell R which groups to plot by

Messages

```
[16] ERROR: There are no factors in the active data set.  
[17] ERROR: There are no factors in the active data set.
```



boxplot()

R Commander

File Edit Data Statistics Graphs Models Distributions Tools Help

Data set: **lowbwt** Edit data set View data set Model: **<No active model>**

Script Window

```
boxplot(sbp ~ apgar5, data=lowbwt, ylab="sbp")
```

plot this

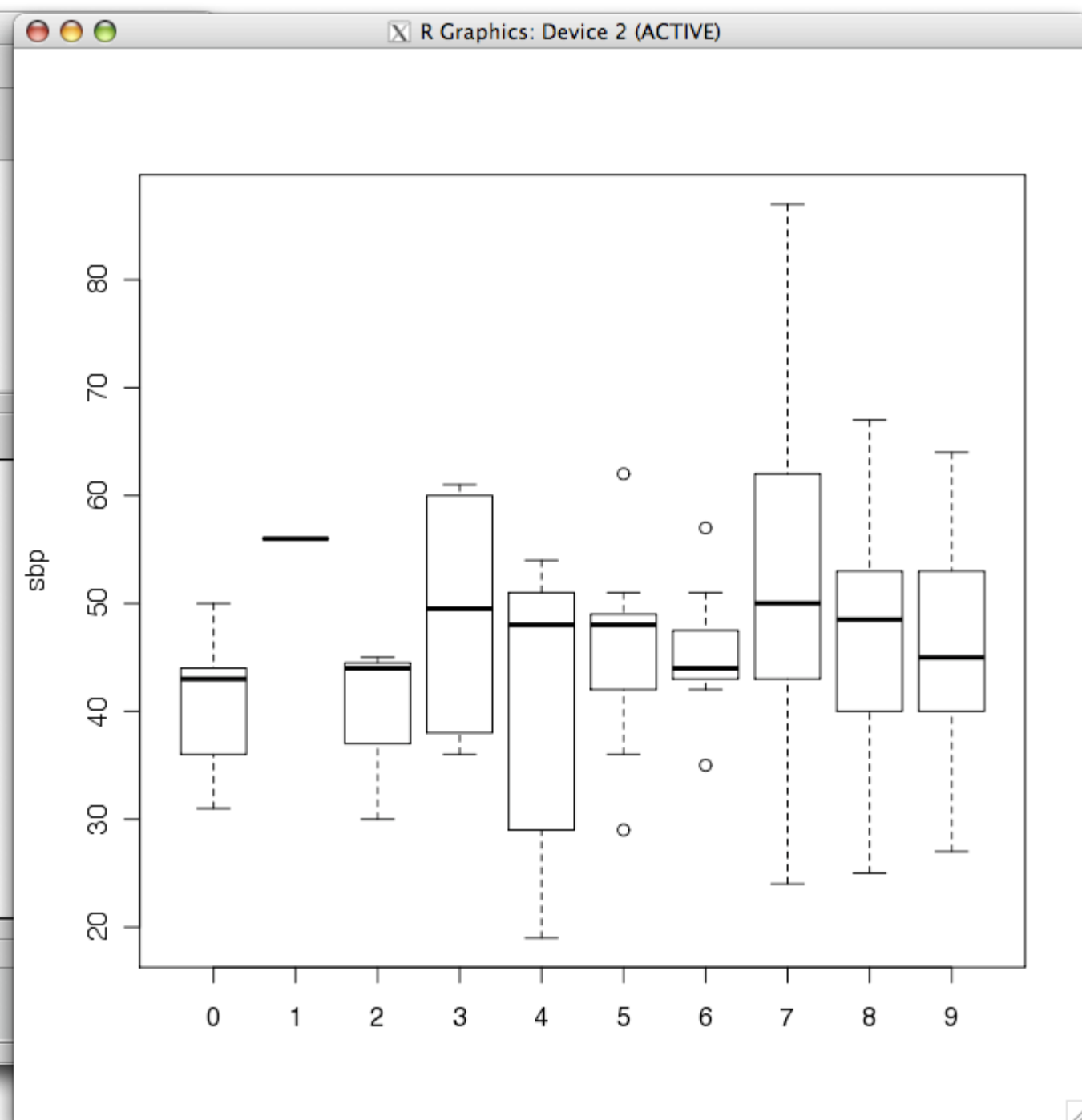
by this group

Output Window

```
> boxplot(lowbwt$sbp, ylab="sbp")  
> boxplot(sbp ~ sex, data=lowbwt, ylab="sbp")  
> boxplot(sbp ~ apgar5, data=lowbwt, ylab="sbp")
```

Messages

```
[16] ERROR: There are no factors in the active data set.  
[17] ERROR: There are no factors in the active data set.
```

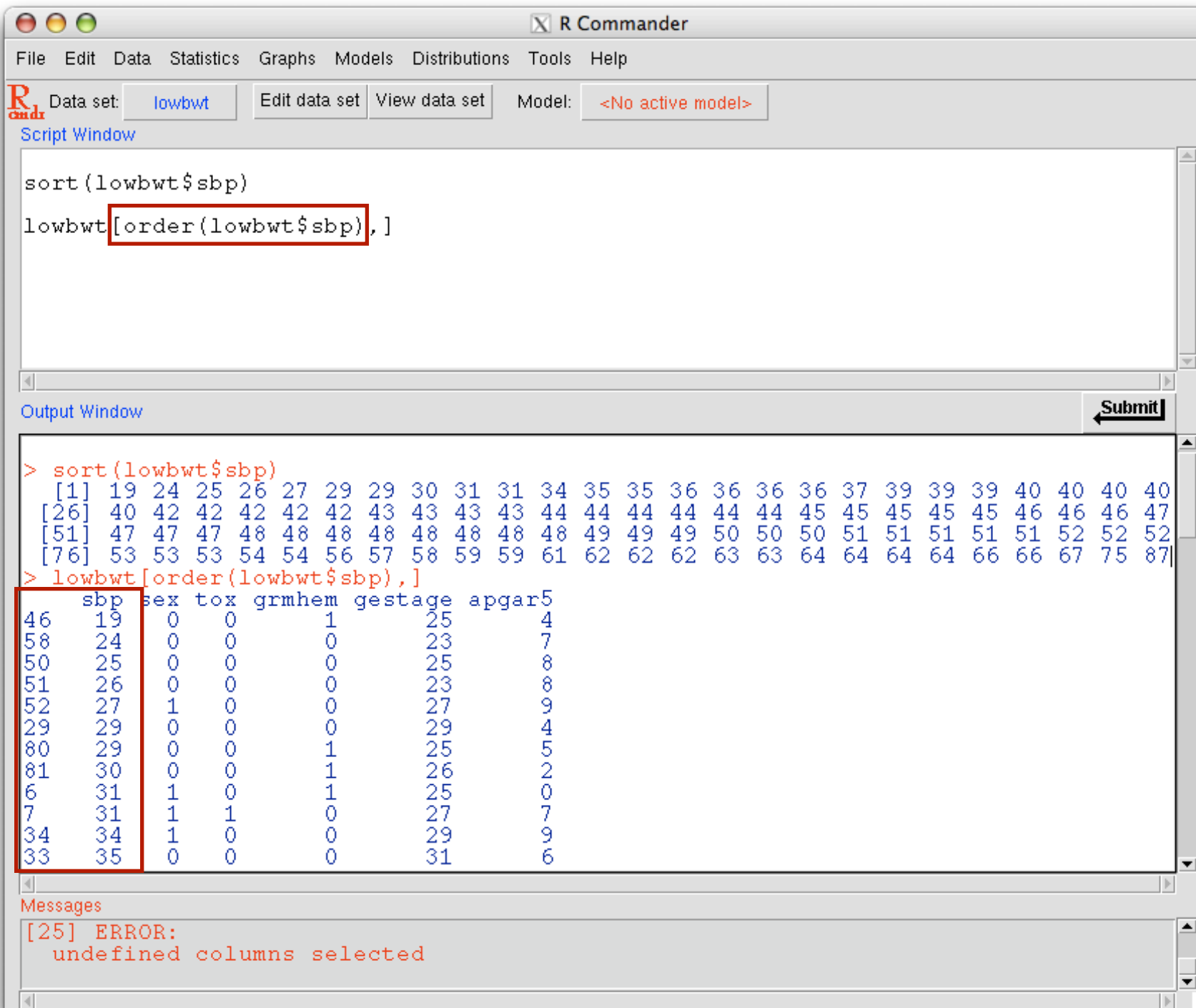


Useful commands

- sorting
 - `sort(X)`
 - `sort(X, decreasing=TRUE)`
 - `order(X)`
 - `order(X, decreasing=TRUE)`
- mean, median, standard deviation, mode
 - `mean(X)`
 - `median(X)`
 - `sd(X)`
 - `head(sort(table(X), decreasing=TRUE))`

Sorting

`sort()` sorts a variable's data points
`order()` returns the positions of the sorted data points



The screenshot shows the R Commander window with the following components:

- Menu Bar:** File, Edit, Data, Statistics, Graphs, Models, Distributions, Tools, Help.
- Data set:** lowbwt (selected). Buttons: Edit data set, View data set.
- Model:** <No active model>
- Script Window:**

```
sort(lowbwt$sbp)
lowbwt[order(lowbwt$sbp),]
```
- Output Window:**

```
> sort(lowbwt$sbp)
 [1] 19 24 25 26 27 29 29 30 31 31 34 35 35 36 36 36 36 37 39 39 39 40 40 40 40
[26] 40 42 42 42 42 42 43 43 43 43 44 44 44 44 44 45 45 45 45 45 46 46 46 47
[51] 47 47 47 48 48 48 48 48 48 48 49 49 49 50 50 50 51 51 51 51 51 52 52 52
[76] 53 53 53 54 54 56 57 58 59 59 61 62 62 62 63 63 64 64 64 64 66 66 67 75 87

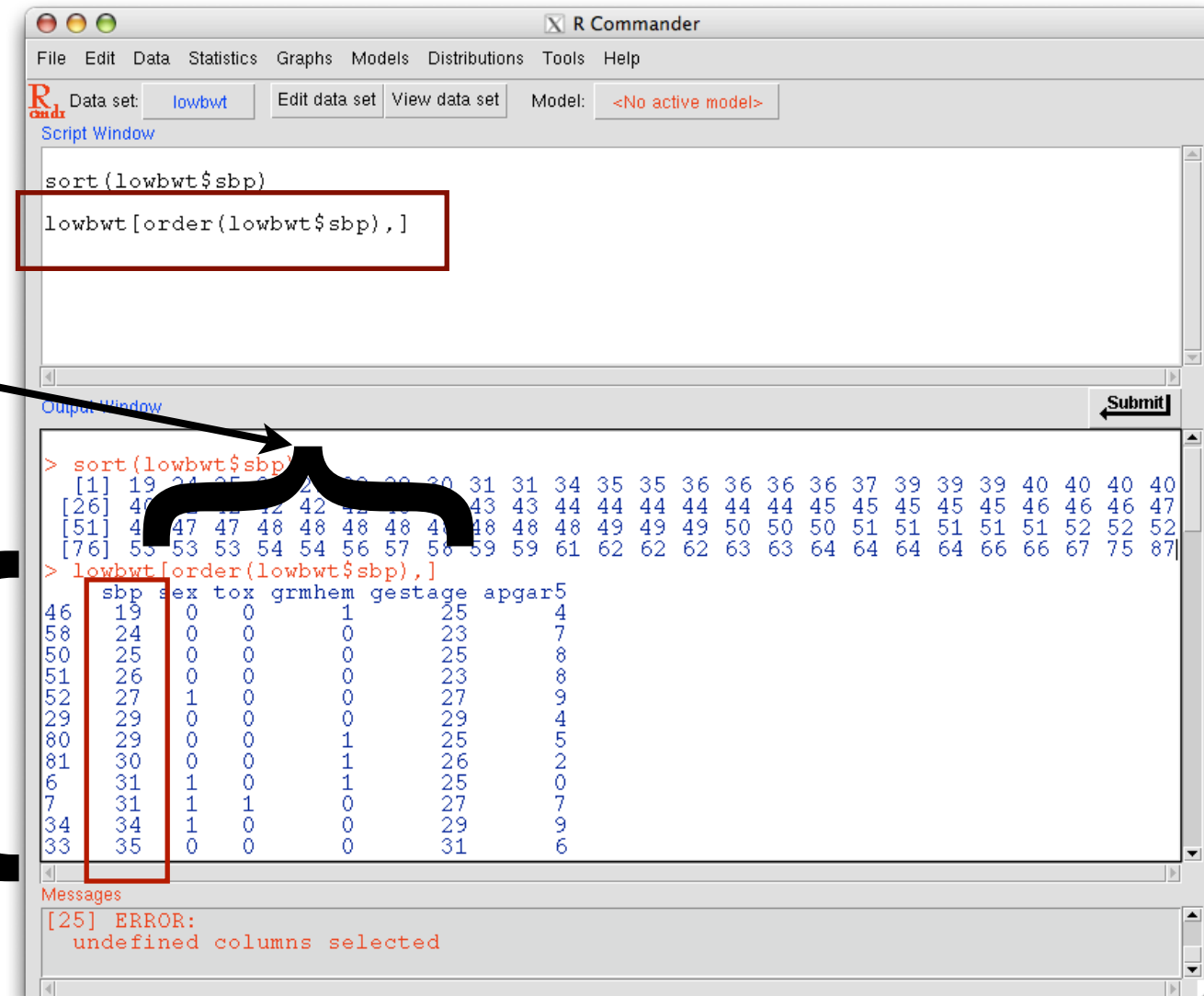
> lowbwt[order(lowbwt$sbp),]
  sbp sex tox grmhem gestage apgar5
46  19  0  0      1      25      4
58  24  0  0      0      23      7
50  25  0  0      0      25      8
51  26  0  0      0      23      8
52  27  1  0      0      27      9
29  29  0  0      0      29      4
80  29  0  0      1      25      5
81  30  0  0      1      26      2
6   31  1  0      1      25      0
7   31  1  1      0      27      7
34  34  1  0      0      29      9
33  35  0  0      0      31      6
```
- Messages:**

```
[25] ERROR:
      undefined columns selected
```

Sorting

dataset[ROWS, COLUMNS]

Now all the columns in lowbwt are displayed sorted by sbp. Notice that when no columns are specified, they are all selected.



The screenshot shows the R Commander interface. The Script Window contains the following code:

```
sort(lowbwt$sbp)
lowbwt[order(lowbwt$sbp),]
```

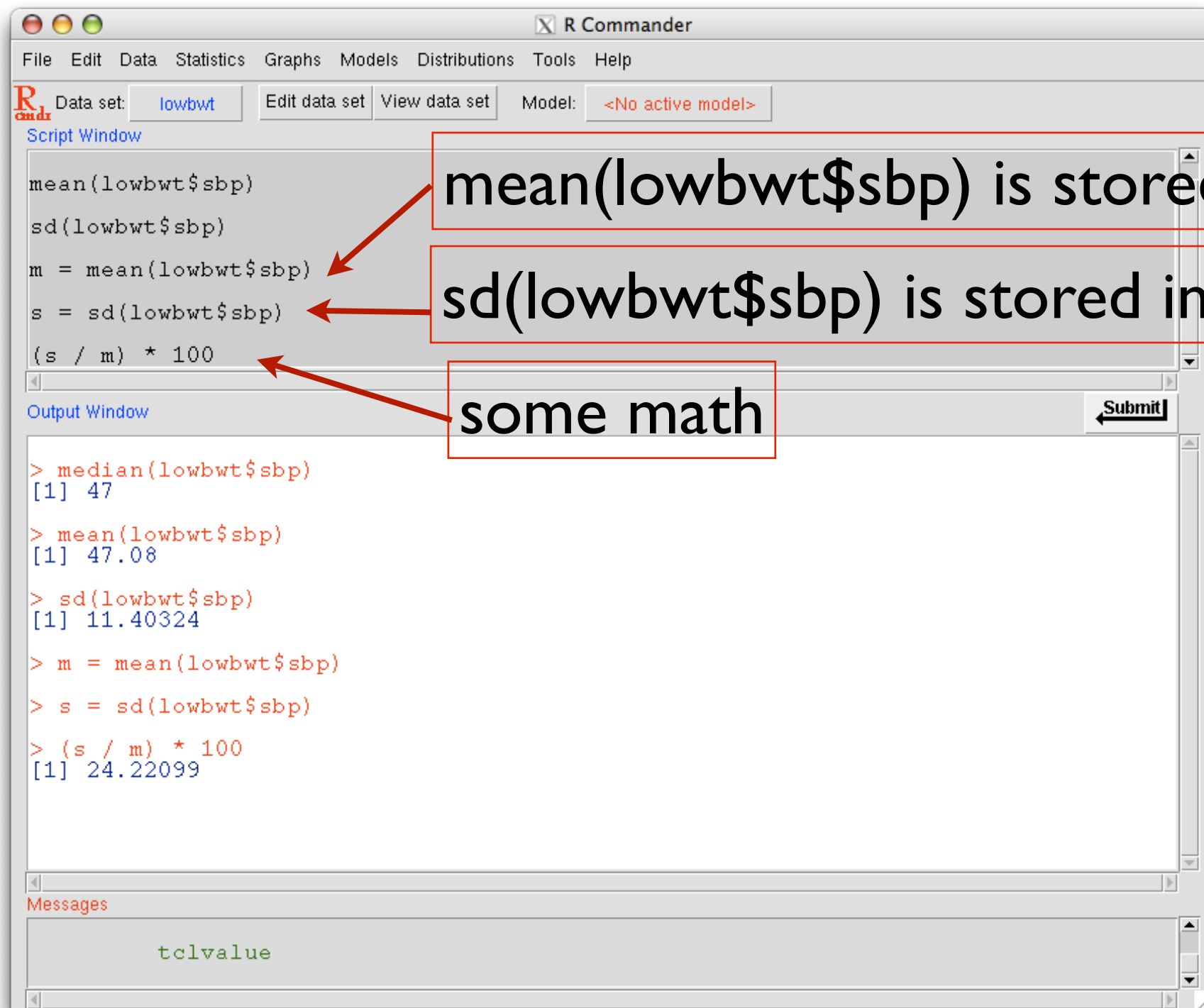
The Output Window shows the result of the second command, displaying a data frame with columns: sbp, sex, tox, grmhem, gestage, and apgar5. The rows are sorted by the sbp column. A red box highlights the first column (sbp) in the output, and a bracket indicates that all columns are selected when no column index is specified in the R command.

	sbp	sex	tox	grmhem	gestage	apgar5
[1]	19	0	0	1	25	4
[26]	46	0	0	0	23	7
[51]	47	0	0	0	25	8
[76]	53	0	0	0	23	8
	27	1	0	0	27	9
	29	0	0	0	29	4
	29	0	0	1	25	5
	30	0	0	1	26	2
	31	1	0	1	25	0
	31	1	1	0	27	7
	34	1	0	0	29	9
	35	0	0	0	31	6

Messages

```
[25] ERROR:
      undefined columns selected
```

mean, median, sd, mode



The screenshot shows the R Commander window with the following components:

- Menu Bar:** File, Edit, Data, Statistics, Graphs, Models, Distributions, Tools, Help.
- Data set:** lowbwt (selected). Buttons: Edit data set, View data set.
- Model:** <No active model>
- Script Window:**

```
mean(lowbwt$sbp)
sd(lowbwt$sbp)
m = mean(lowbwt$sbp)
s = sd(lowbwt$sbp)
(s / m) * 100
```
- Output Window:**

```
> median(lowbwt$sbp)
[1] 47

> mean(lowbwt$sbp)
[1] 47.08

> sd(lowbwt$sbp)
[1] 11.40324

> m = mean(lowbwt$sbp)

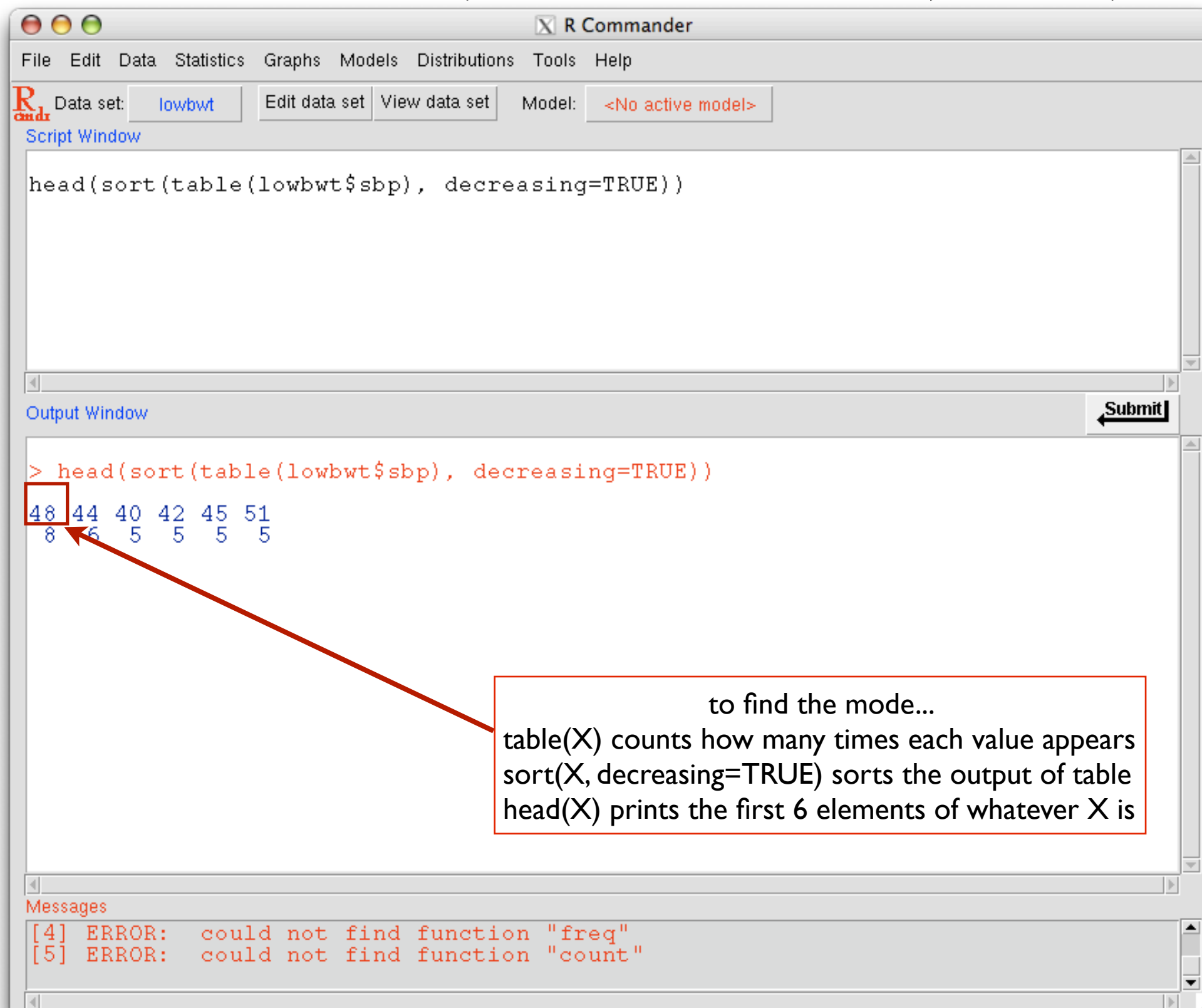
> s = sd(lowbwt$sbp)

> (s / m) * 100
[1] 24.22099
```
- Messages:** tclvalue

Annotations with red arrows point from text boxes to specific lines in the script:

- mean(lowbwt\$sbp) is stored in m** points to the line `m = mean(lowbwt$sbp)`.
- sd(lowbwt\$sbp) is stored in s** points to the line `s = sd(lowbwt$sbp)`.
- some math** points to the line `(s / m) * 100`.

mean, median, sd, mode



The screenshot shows the R Commander interface. The 'Data set' is 'lowbwt'. The 'Script Window' contains the command: `head(sort(table(lowbwt$sbp), decreasing=TRUE))`. The 'Output Window' shows the result of this command: a table with two rows. The first row contains the values 48, 44, 40, 42, 45, 51. The second row contains the counts 8, 6, 5, 5, 5, 5. A red box highlights the value 48 in the first row, and a red arrow points from a text box to this value. The text box contains the following text: 'to find the mode... table(X) counts how many times each value appears sort(X, decreasing=TRUE) sorts the output of table head(X) prints the first 6 elements of whatever X is'. The 'Messages' window at the bottom shows two error messages: '[4] ERROR: could not find function "freq"' and '[5] ERROR: could not find function "count"'.

R Commander

File Edit Data Statistics Graphs Models Distributions Tools Help

Data set: **lowbwt** Edit data set View data set Model: **<No active model>**

Script Window

```
head(sort(table(lowbwt$sbp), decreasing=TRUE))
```

Output Window

```
> head(sort(table(lowbwt$sbp), decreasing=TRUE))
```

48	44	40	42	45	51
8	6	5	5	5	5

to find the mode...
table(X) counts how many times each value appears
sort(X, decreasing=TRUE) sorts the output of table
head(X) prints the first 6 elements of whatever X is

Messages

```
[4] ERROR: could not find function "freq"  
[5] ERROR: could not find function "count"
```

Subsetting data

- `dataset[Rows]`
- `which(TEST)`
- `length(X)`

dataset[Rows]

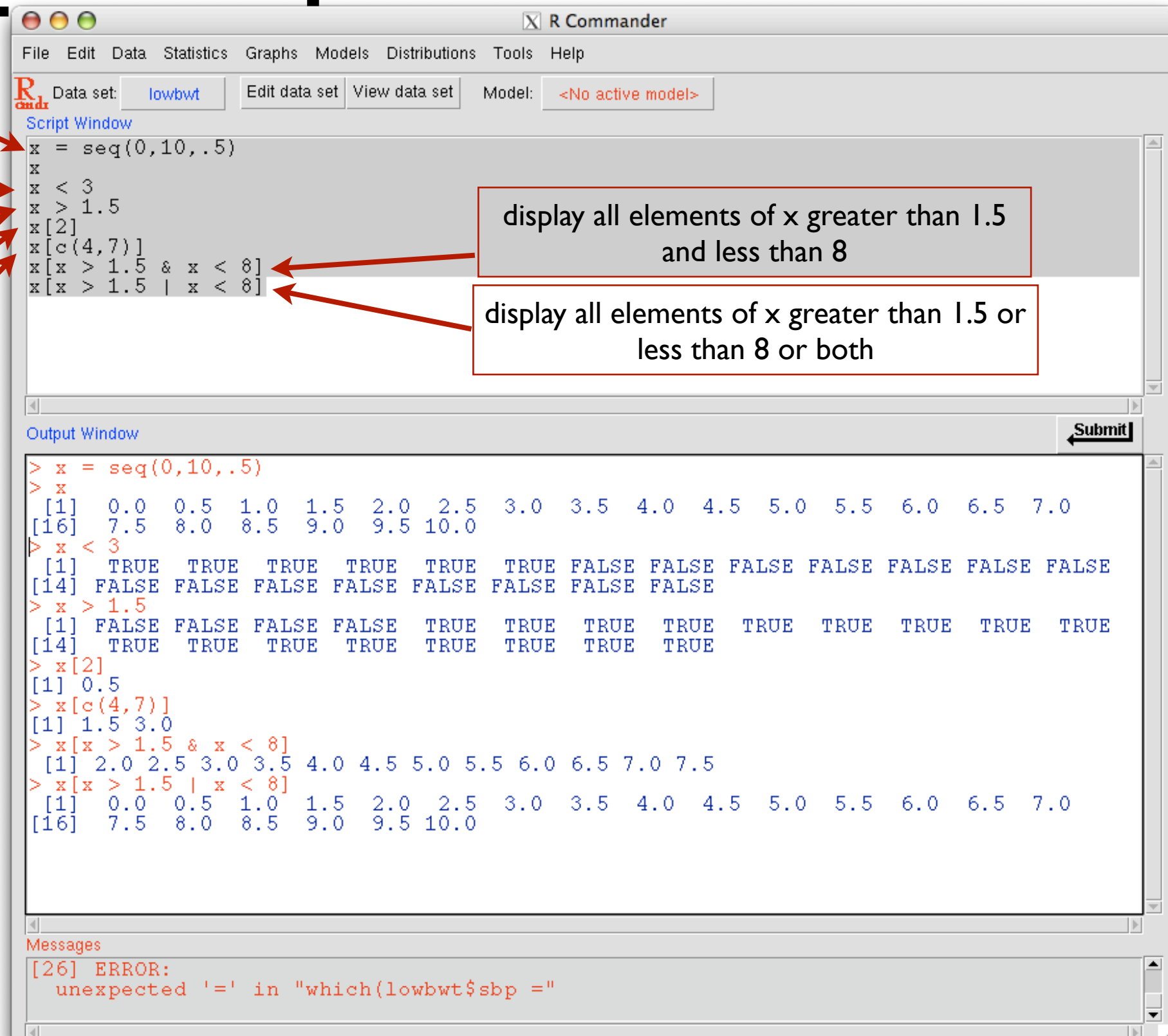
make a sequence of numbers from 0 to 10 in steps of 0.5

test if each element of x is less than 3

test if each element of x is more than 1.5

display the 2nd element of x

display the 4th and 7th element of x



The screenshot shows the R Commander interface with the following components:

- Script Window:** Contains the following R code:

```
x = seq(0,10,.5)
x
x < 3
x > 1.5
x[2]
x[c(4,7)]
x[x > 1.5 & x < 8]
x[x > 1.5 | x < 8]
```

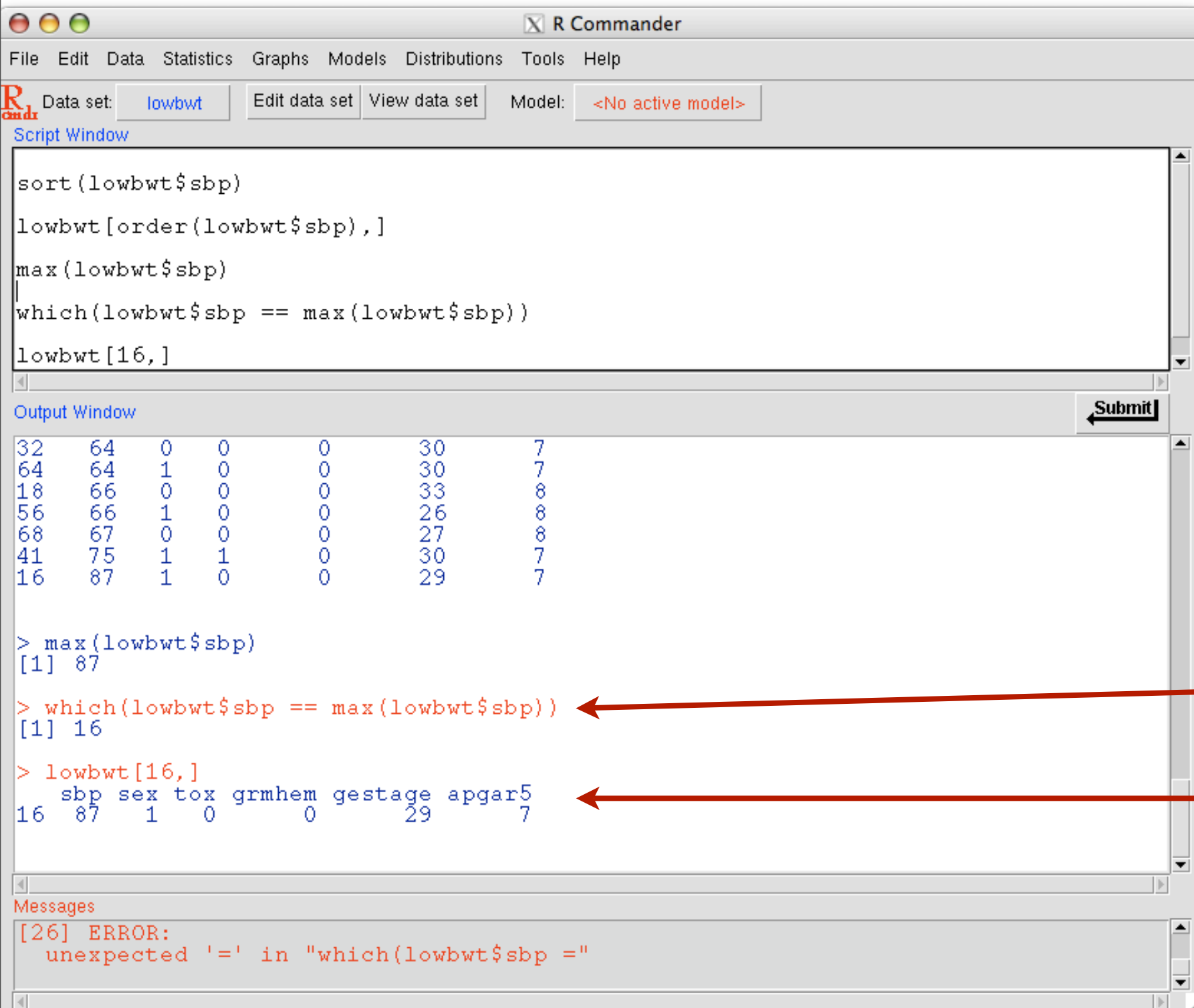
Annotations with red arrows point from text boxes on the left to specific lines of code:

 - From "make a sequence of numbers from 0 to 10 in steps of 0.5" to `x = seq(0,10,.5)`
 - From "test if each element of x is less than 3" to `x < 3`
 - From "test if each element of x is more than 1.5" to `x > 1.5`
 - From "display the 2nd element of x" to `x[2]`
 - From "display the 4th and 7th element of x" to `x[c(4,7)]`
 - From "display all elements of x greater than 1.5 and less than 8" to `x[x > 1.5 & x < 8]`
 - From "display all elements of x greater than 1.5 or less than 8 or both" to `x[x > 1.5 | x < 8]`
- Output Window:** Shows the results of the code execution:

```
> x = seq(0,10,.5)
> x
[1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0
[16] 7.5 8.0 8.5 9.0 9.5 10.0
> x < 3
[1] TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE
[14] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
> x > 1.5
[1] FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[14] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
> x[2]
[1] 0.5
> x[c(4,7)]
[1] 1.5 3.0
> x[x > 1.5 & x < 8]
[1] 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5
> x[x > 1.5 | x < 8]
[1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0
[16] 7.5 8.0 8.5 9.0 9.5 10.0
```
- Messages:** Shows an error message:

```
[26] ERROR:
unexpected '=' in "which(lowbwt$sbp ="
```


which(TEST)



The screenshot shows the R Commander interface. The 'Script Window' contains the following R code:

```
sort(lowbwt$sbp)
lowbwt[order(lowbwt$sbp),]
max(lowbwt$sbp)
which(lowbwt$sbp == max(lowbwt$sbp))
lowbwt[16,]
```

The 'Output Window' displays the results of the code execution:

sbp	sex	tox	grmhem	gestage	apgar5	
32	64	0	0	0	30	7
64	64	1	0	0	30	7
18	66	0	0	0	33	8
56	66	1	0	0	26	8
68	67	0	0	0	27	8
41	75	1	1	0	30	7
16	87	1	0	0	29	7

Below the table, the output of the R commands is shown:

```
> max(lowbwt$sbp)
[1] 87

> which(lowbwt$sbp == max(lowbwt$sbp))
[1] 16

> lowbwt[16,]
      sbp sex tox grmhem gestage apgar5
16  87  1  0    0      29      7
```

The 'Messages' window at the bottom shows an error message:

```
[26] ERROR:
      unexpected '=' in "which(lowbwt$sbp ="
```

`which()`, like `order()` returns the positions of data points that match a test

test: which `lowbwt$sbp` is the max `lowbwt$sbp`?
output: the 16th data point

display the 16th data point
(16th row, all columns)

length()

The screenshot shows the R Commander interface with the following components:

- Script Window:** Contains the following R code:

```
nurse
dim(nurse)
length(nurse$resident)
nurse[nurse$resident < 35,]
nurse$state[nurse$resident < 35]
nurse[nurse$resident < 35,1]
```
- Output Window:** Contains the following R output:

```
> dim(nurse)
[1] 51 2

> length(nurse$resident)
[1] 51

> nurse[nurse$resident < 35,]
  state resident
2  Alaska   22.5
3  Arizona   21.7
5  California 26.7
9 District of Columbia 34.4
10 Florida   23.1
12 Hawaii   13.6
13 Idaho    31.3
29 Nevada   18.6
31 New Jersey 33.4
32 New Mexico 30.2
34 North Carolina 30.4
38 Oregon    30.2
41 South Carolina 28.8
45 Utah      29.1
```
- Messages Window:** Contains an error message:

```
[20] ERROR:
      undefined columns selected
```

Annotations on the screenshot include:

- A red box around the code in the Script Window with the text: "Use length() or dim() to count".
- A red box around the output `[1] 51 2` with the text: "51 rows, 2 columns".
- A red box around the output `[1] 51` with the text: "51 elements".
- A large black box with white text: "Try it yourself! Use length() to find the number of states where resident has a value of less than 35.".
- A red box around the output of the subsetting operation with the text: "rows where resident < 35, all columns".

More help

- To get help with any command...
- `?command`
- `help(command)`

