Making maps

Making maps in R

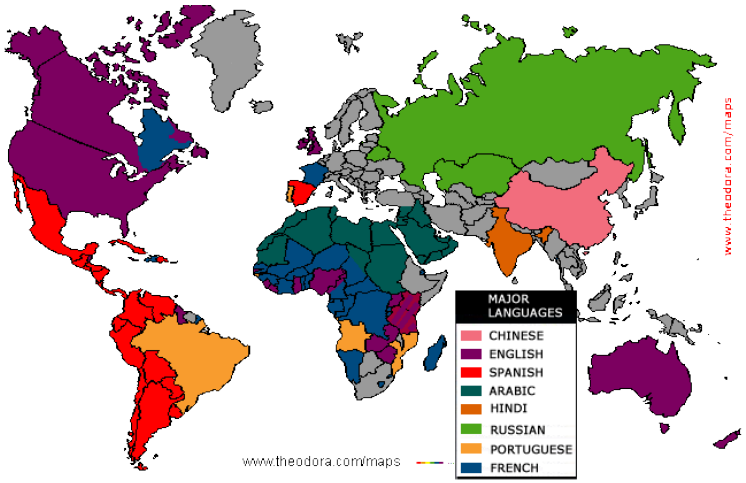
One way of visually representing data is to plot them on a geographical map.

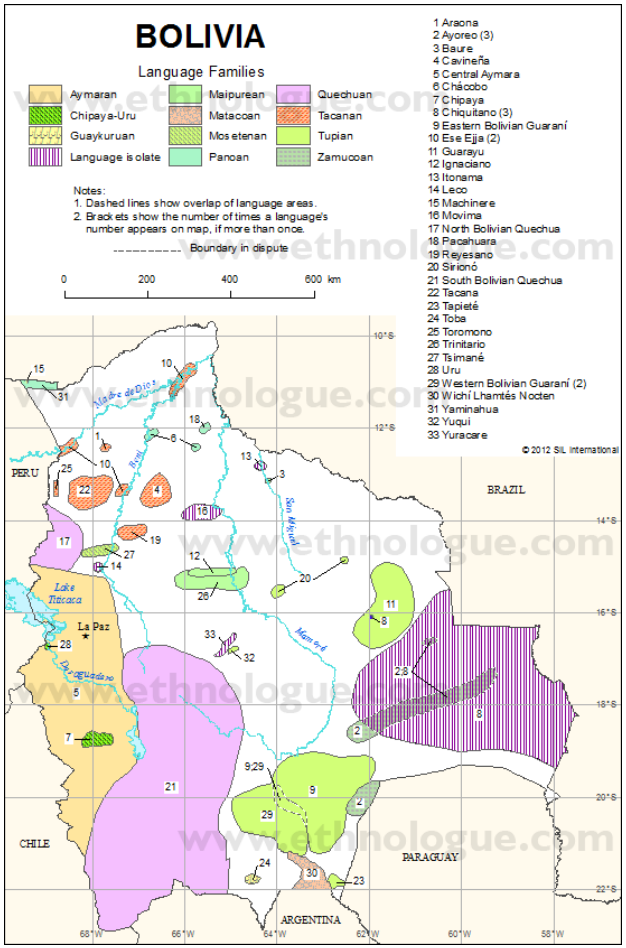
This gives you a first impression of skewed distributions.

The following tutorial is based on the one created by Michael Dunn, available at

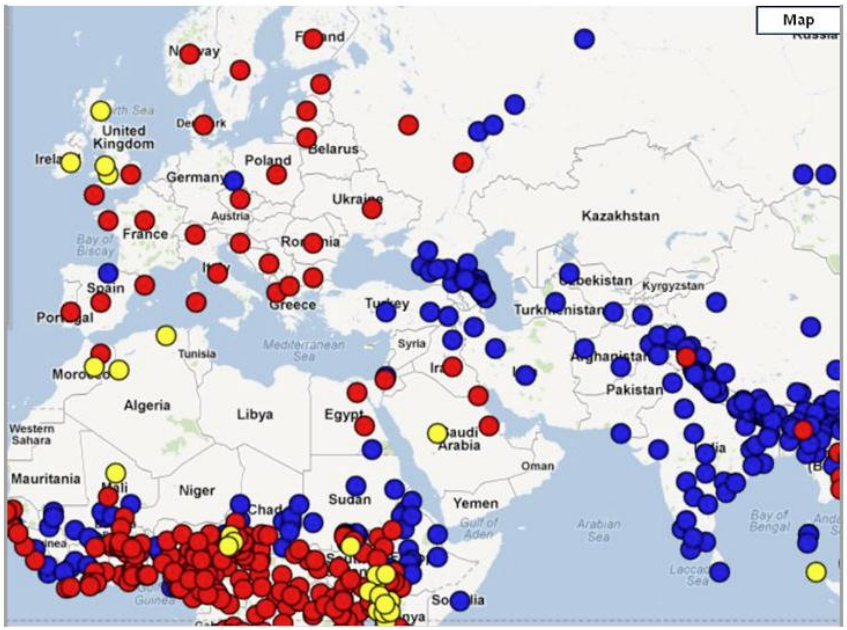
http://www.mpi.nl/departments/independent-research-groups/evolutionary-processes/tools/mapping-with-r

Making maps in R





Word Order



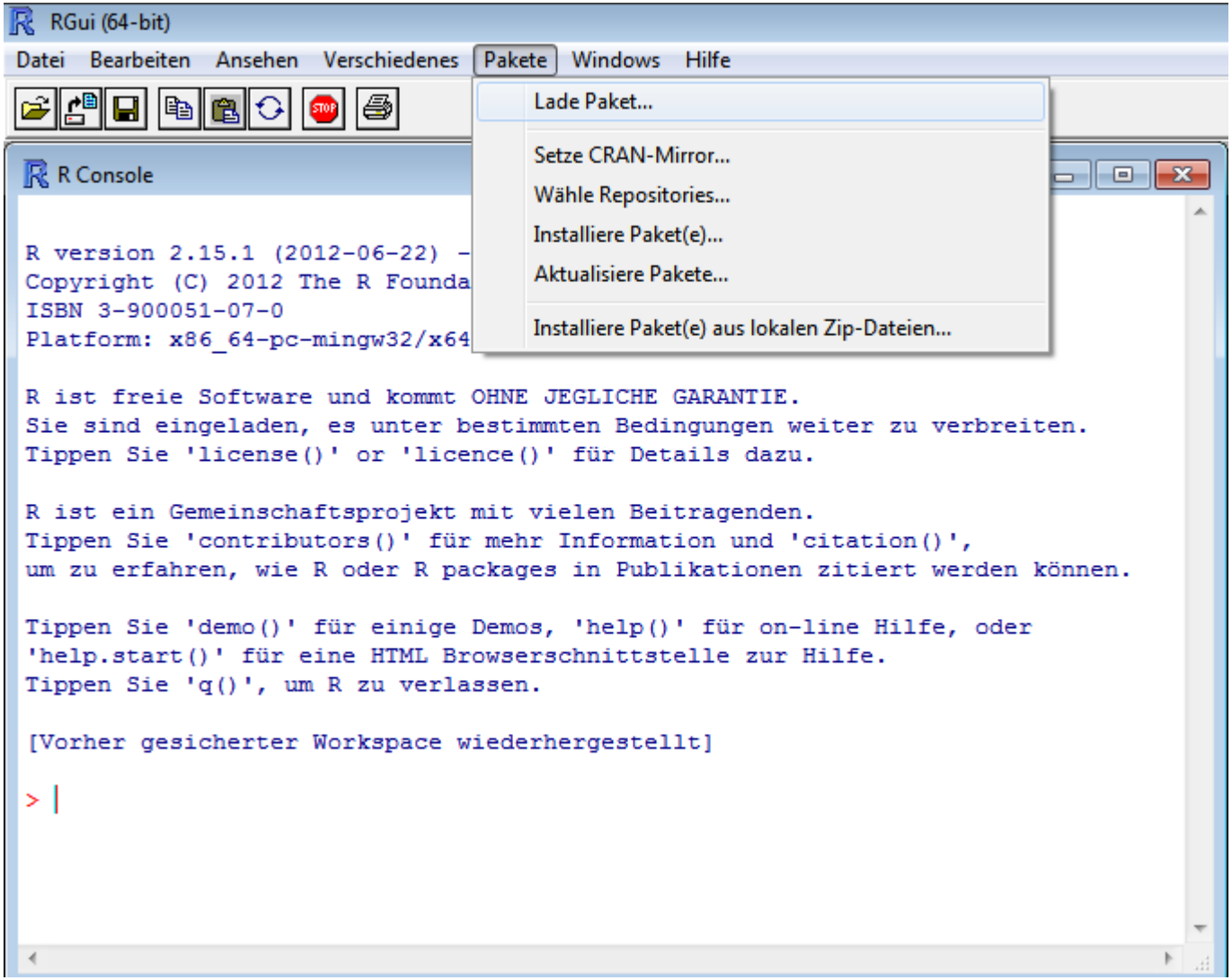
Making maps in R

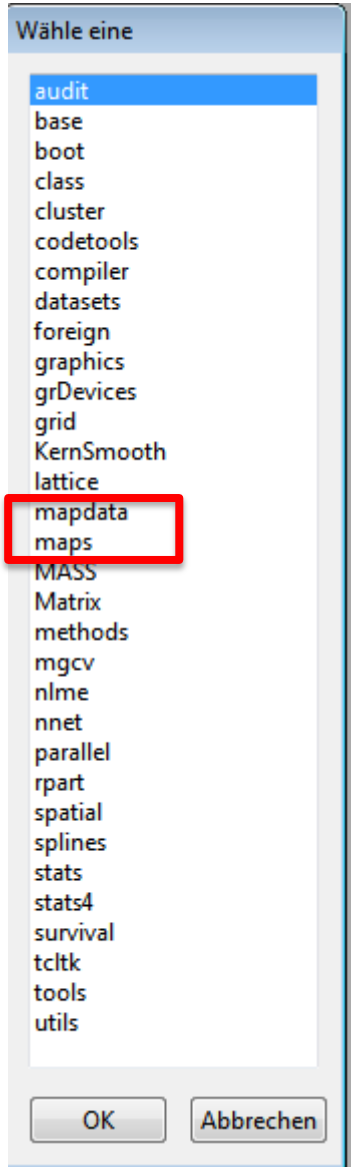
Step 1: Install the non-standard libraries

On the MacOSX version you open the R application, select **Package Installer** from the **Packages & Data** menu and choose **Get List**. When the list appears, scroll down to and select **maps** and **mapdata**, tick the **Install Dependencies** checkbox (to be on the safe side), and click **Install Selected**.

Making maps in R

In the Windows version...





Making maps in R

Test the installation

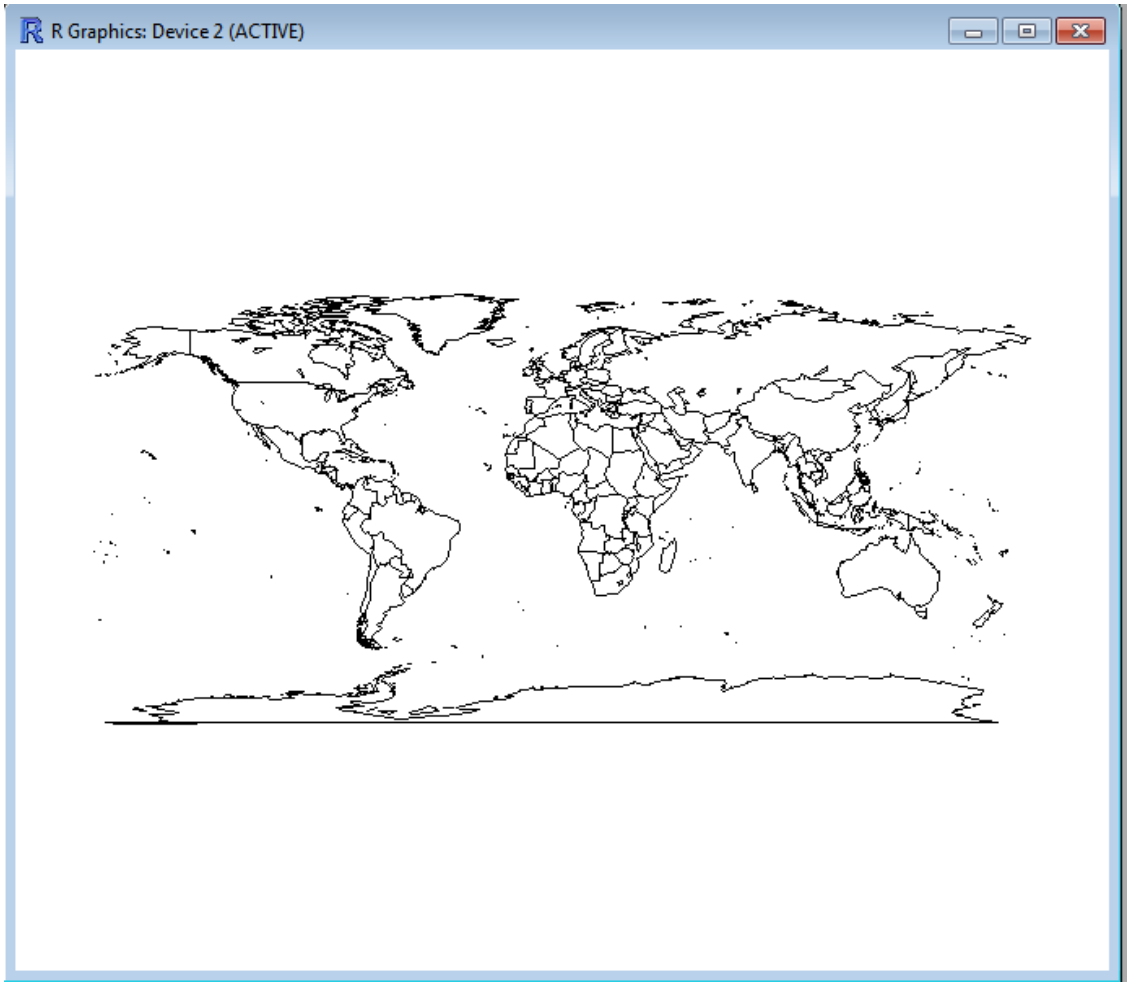
If R does not complain (seriously), you have installed the packages

```
> library(maps)
Warnmeldung:
Paket 'maps' wurde unter R Version 2.15.3 erstellt
> library(mapdata)
Warnmeldung:
Paket 'mapdata' wurde unter R Version 2.15.3 erstellt
>
```

Making maps in R

Step 2: Drawing maps: basics

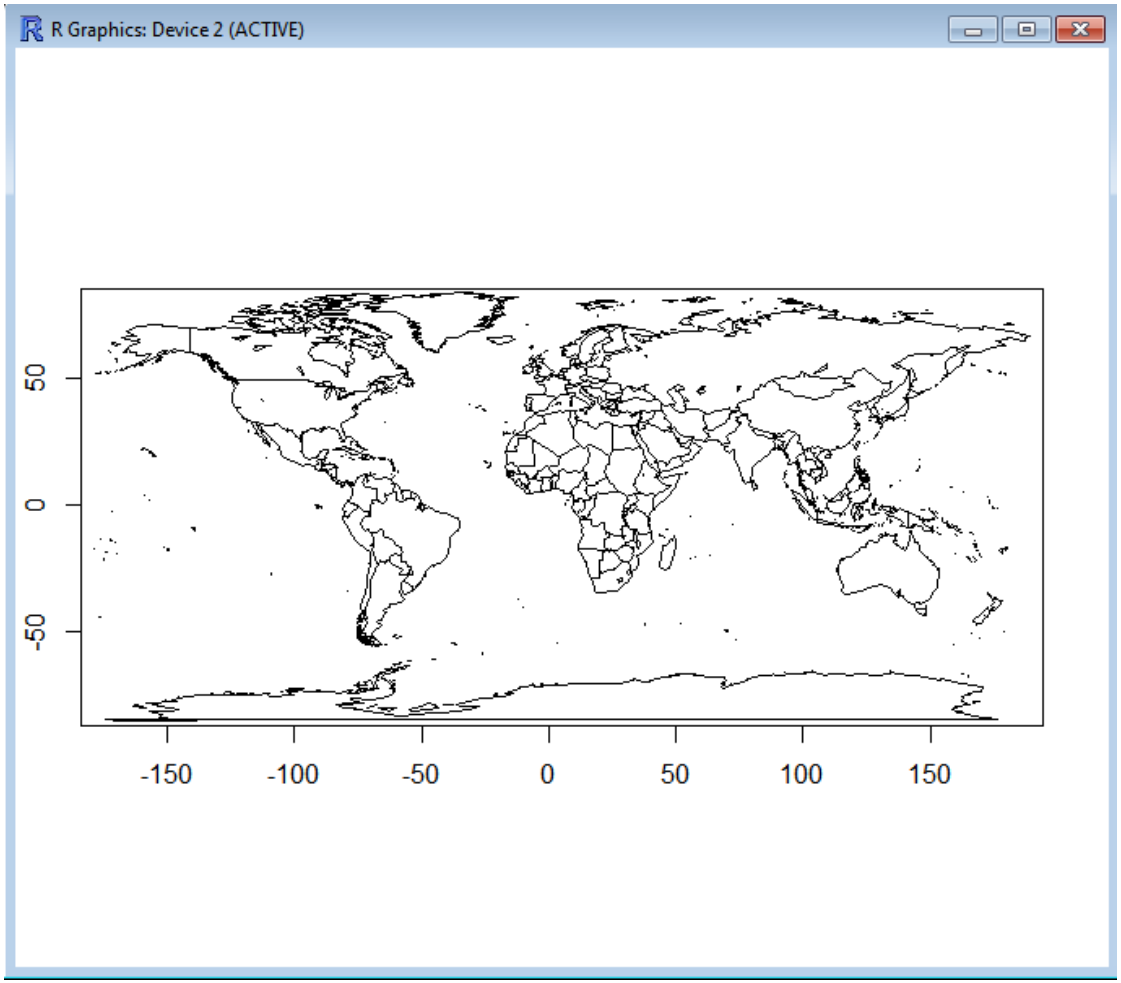
map()



Making maps in R

Step 2: Drawing maps: basics

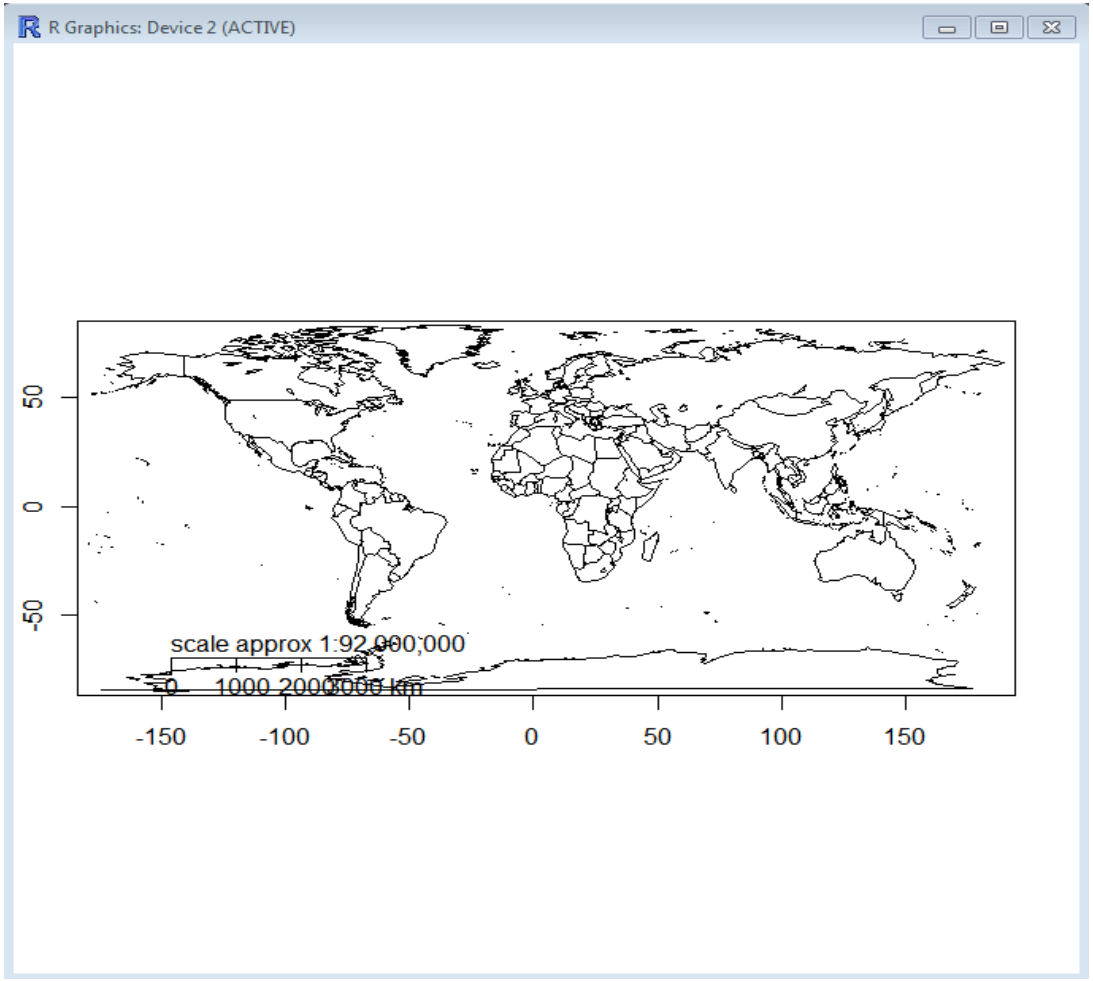
map()
map.axes()



Making maps in R

Step 2: Drawing maps: basics

map()
map.axes()
map.scale()



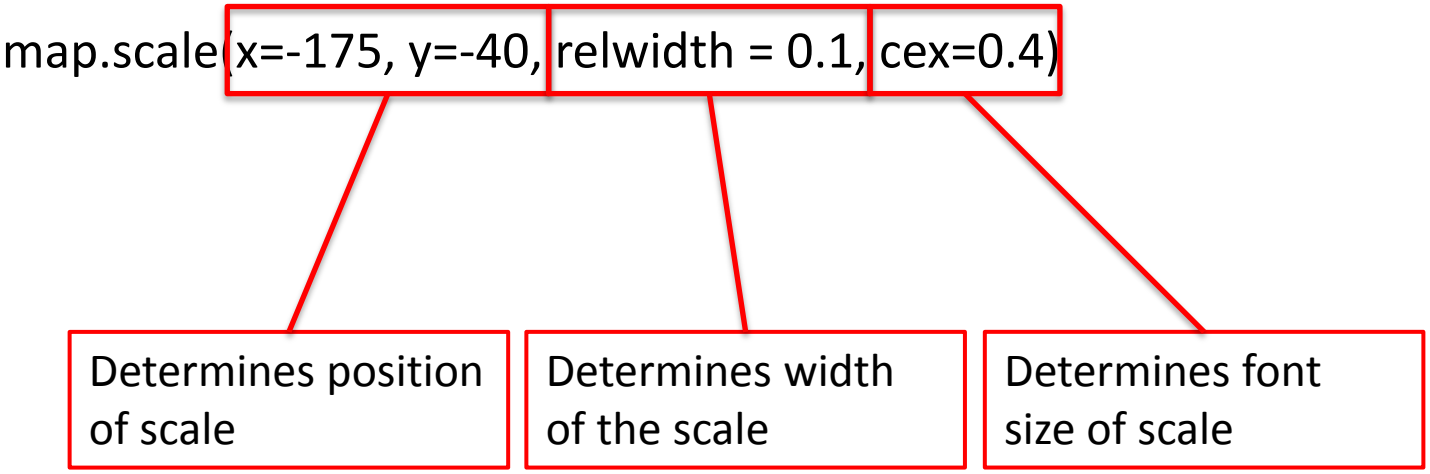
Making maps in R

Step 2: Drawing maps: basics

```
map.scale(x=-175, y=-40, relwidth = 0.1, cex=0.4)
```

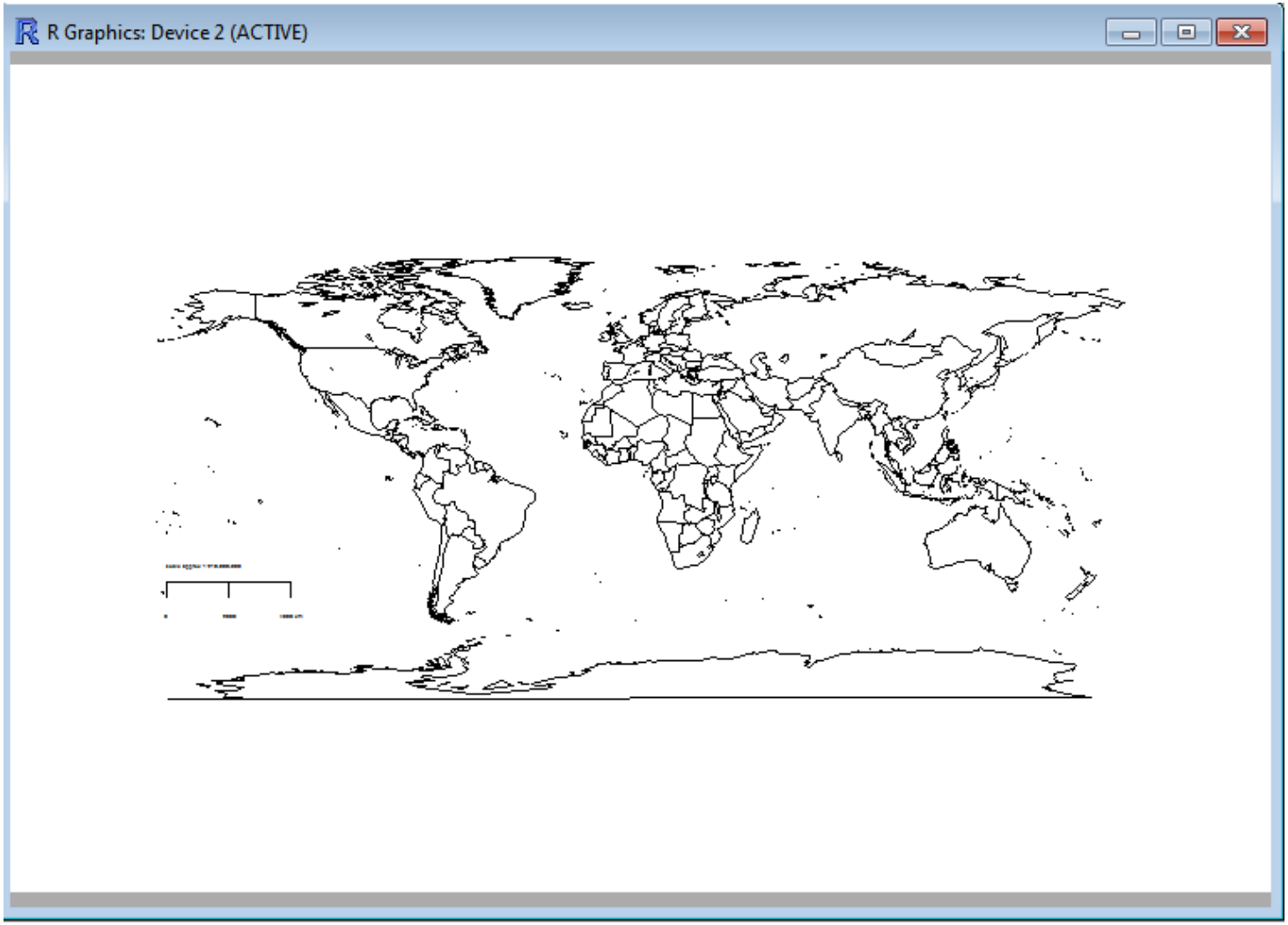
Making maps in R

Step 2: Drawing maps: basics



Making maps in R

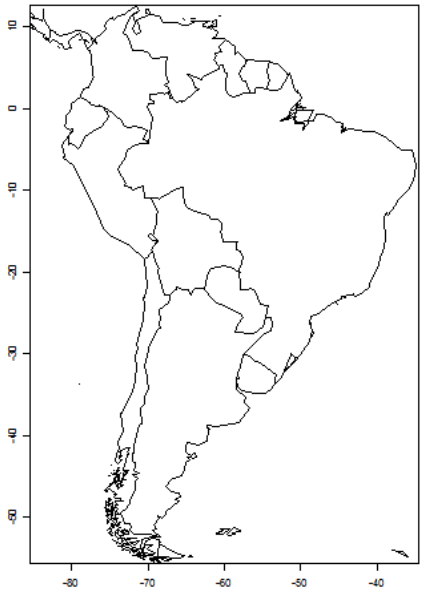
Step 2: Drawing maps: basics

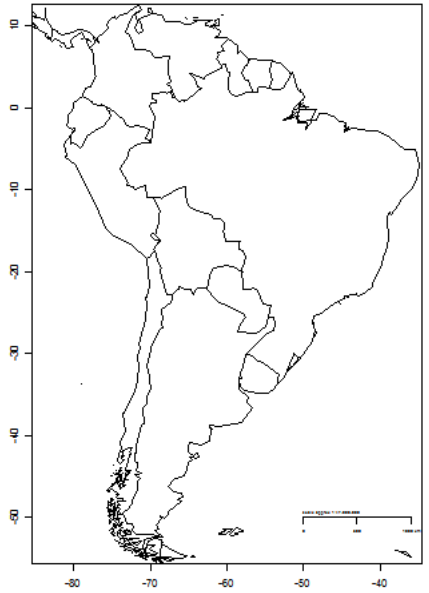


Making maps in R

Step 3: Zooming in

```
map(xlim=c(-85, -35), ylim=c(-55,12))
```





Making maps in R

Step 4: Colors, captions

Colors:

```
map(xlim=c(-85, -35), ylim=c(-55,12), fill=TRUE, col="white", bg="darkseagreen")
```

Determines background color

Determines the foreground (countries) color

Makes sure that the col argument applies to the countries and not just the borders

Making maps in R

Step 4: Colors, captions

Captions:

```
map(xlim=c(-85, -35), ylim=c(-55,12), fill=TRUE, col="white", bg="darkseagreen")
title(main="South America")
```

Making maps in R

Step 4: Colors, captions

If you want to change the color of the caption, e.g. because you want to have a black background, you can do this by adding another argument to the title() function: col.main=

Try the following:

1. set background color to black
2. set caption color to white

Making maps in R

Step 4: Colors, captions

Caption color: col.main=

```
map(xlim=c(-85, -35), ylim=c(-55,12), fill=TRUE, col="white", bg="black")
title(main="South America by night", col.main="white")
```

Making maps in R

Step 5: Plotting data points onto a map

1. Read the dataset languagesSA into R, assign it to a variable

	B	C	D	E	F	G	H	I	J	K
1	ISOCODE	AFFILIATION	COUNTRY	LATITUDE	LONGITUDE	SPEAKERS	STATUS	WORDORDER	CASE	AGREEMENT
2	agr	Jivaroan	Peru	-5	-78	38300	4	SOV	Y	Y
3	apu	Arawakan	Brazil	-9	-67	2780	6b	SOV	Y	N
4	kwi	Barbacoan	Ecuador	1.5	-78.25	13000	6b	SOV	Y	N
5	ayr	Aymaran	Bolivia	-17	-69	2589000	6b	SOV	Y	N
6	brg	Arawakan	Bolivia	-13.09	-64.17	40	8a	SOV	Y	N
7	cav	Tacanan	Bolivia	-13.33	-66.5	1680	6b	SOV	Y	N
8	quz	Quechuan	Peru	-14.5	-71	1500000	3	SOV	Y	N
9	des	Tucanoan	Colombia	0.83	-69.83	3420	6b	SOV	Y	Y
10	eme	Cariban	French Guyana	3.2	-52.4	400	6b	SOV	Y	Y
11	hix	Cariban	Brazil	-1	-59	600	4	SOV	Y	Y
12	qub	Quechuan	Peru	-6	-76	40000	5	SOV	Y	Y
13	jup	Makuan	Brazil	0.17	-69.25	1240	5	Free	Y	Y
14	arh	Aruakan	Colombia	10.67	-73.75	8000	5	Free	Y	Y
15	ito	Isolate	Bolivia	-12.83	-64.33	5	8b	Free	Y	Y
16	jaa	Arawan	Brazil	-7.5	-65.5	800	4	Free	Y	Y
17	kay	Tupian	Brazil	-12	-52.67	400	6a	Free	Y	Y
18	kxo	Isolate	Brazil	-12.2	-64.6	5	8b	Free	Y	Y
19	ktn	Tupian	Brazil	-9.5	-64	210	5	Free	Y	Y
20	arr	Tupian	Brazil	-10.33	-62	210	7	SVO	Y	Y
21	cod	Tupian	Peru	-5	-74.5	250	8a	SVO	Y	N
22	xwa	Isolate	Brazil	-12.58	-60.67	25	8b	SVO	Y	N
23	lec	Isolate	Bolivia	-15	-67.9	20	8a	SVO	Y	N
24	wmd	Nambikwaran	Brazil	-12.75	-59.17	330	6b	SVO	Y	N
25	arn	Mapudungun	Chile	-38	-72	258620	4	SVO	Y	N

Making maps in R

Step 5: Plotting data points onto a map

1. Read the dataset languagesSA into R, assign it to a variable

```
> mySAdata<-read.delim(file.choose())
> head(mySAdata)
  LANGUAGE ISOCODE AFFILIATION COUNTRY LATITUDE LONGITUDE SPEAKERS STATUS WORDORDER CASE AGREEMENT
1 Aguaruna     agr    Jivaroan    Peru    -5.00    -78.00    38300      4        SOV    Y         Y
2  Apurina     apu    Arawakan  Brazil    -9.00    -67.00     2780     6b        SOV    Y         N
3  Awa Pit     kwi   Barbacoan Ecuador     1.50    -78.25    13000     6b        SOV    Y         N
4   Aymara     ayr     Aymaran Bolivia   -17.00    -69.00  2589000     6b        SOV    Y         N
5    Baure     brg    Arawakan Bolivia   -13.09    -64.17       40     8a        SOV    Y         N
6 Cavinena     cav     Tacanan Bolivia   -13.33    -66.50     1680     6b        SOV    Y         N
>
```

2. Plot the languages onto the map using the function points() with the arguments latitude and longitude of your object (remember: $)

Making maps in R

Step 5: Plotting data points onto a map

```
> map(xlim=c(-85, -35), ylim=c(-55,12), fill=TRUE, col="white", bg="darkseagreen")
> title(main="South America")
> points(mySAdata$LONGITUDE, mySAdata$LATITUDE)
>
```

Making maps in R

Step 5: Plotting data points onto a map

```
> map(xlim=c(-85, -35), ylim=c(-55,12), fill=TRUE, col="white", bg="darkseagreen")
> title(main="South America")
> points(mySAdata$LONGITUDE, mySAdata$LATITUDE)
>
```

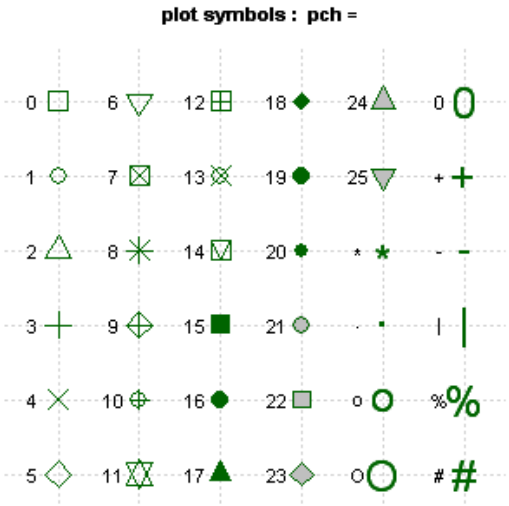
And now of course we can manipulate the points by giving them another color (col=), another form (pch=), and - if possible - another fill.

Making maps in R

Step 5: Plotting data points onto a map

```
> map(xlim=c(-85, -35), ylim=c(-55,12), fill=TRUE, col="white", bg="darkseagreen")
> title(main="South America")
> points(mySAdata$LONGITUDE, mySAdata$LATITUDE)
>
```

And now of course we can manipulate the points by giving them another color (col=), another form (pch=), and - if possible - another fill.



Making maps in R

Step 5: Plotting data points onto a map


```
> points(mySAdata$LONGITUDE, mySAdata$LATITUDE, pch=20, col="red")
> 
```

Making maps in R

Step 5: Plotting data points onto a map

That's all very fine, but we would like the colors to be meaningful.

Display the content sof your object with str()

```
> points(mySAdata$LONGITUDE, mySAdata$LATITUDE, pch=20, col="red")
>
```

Making maps in R

Step 5: Plotting data points onto a map

```
> str(mySAdata)
'data.frame':   40 obs. of  11 variables:
 $ LANGUAGE   : Factor w/ 40 levels "Aguaruna","Apurina",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ ISOCODE    : Factor w/ 40 levels "agr","ame","apu",..: 1 3 22 7 9 11 30 14 15 16 ...
 $ AFFILIATION: Factor w/ 21 levels "Arawakan","Arawan",..: 11 1 5 4 1 19 18 20 7 7 ...
 $ COUNTRY    : Factor w/ 9 levels "Argentina","Bolivia",..: 9 3 6 2 2 2 9 5 7 3 ...
 $ LATITUDE   : num  -5 -9 1.5 -17 -13.1 ...
 $ LONGITUDE  : num  -78 -67 -78.2 -69 -64.2 ...
 $ SPEAKERS   : int  38300 2780 13000 2589000 40 1680 1500000 3420 400 600 ...
 $ STATUS     : Factor w/ 8 levels "3","4","5","6a",..: 2 5 5 5 7 5 1 5 5 2 ...
 $ WORDORDER  : Factor w/ 7 levels "Free","OSV","OVS",..: 4 4 4 4 4 4 4 4 4 4 ...
 $ CASE       : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
 $ AGREEMENT  : Factor w/ 2 levels "N","Y": 2 1 1 1 1 1 1 2 2 2 ...
> 
```

Now suppose we wanted to display which languages had case marking and which of the languages don't.

Making maps in R

Step 5: Plotting data points onto a map

```
map(xlim=c(-85, -35), ylim=c(-55,12), fill=TRUE, col="white", bg="darkseagreen")
title(main="South America")
```

1. Add a new column to the data frame filled with the value "gray"

```
mySAdata$my.col = "gray"
```

2. change the values of the $my.col column to "red" where the value for the $CASE column is "Y"

```
mySAdata$my.col[mySAdata$CASE == "Y"] = "red"
```

3. change the values of the $my.col column to "blue" where the value for the $CASE column is "N"

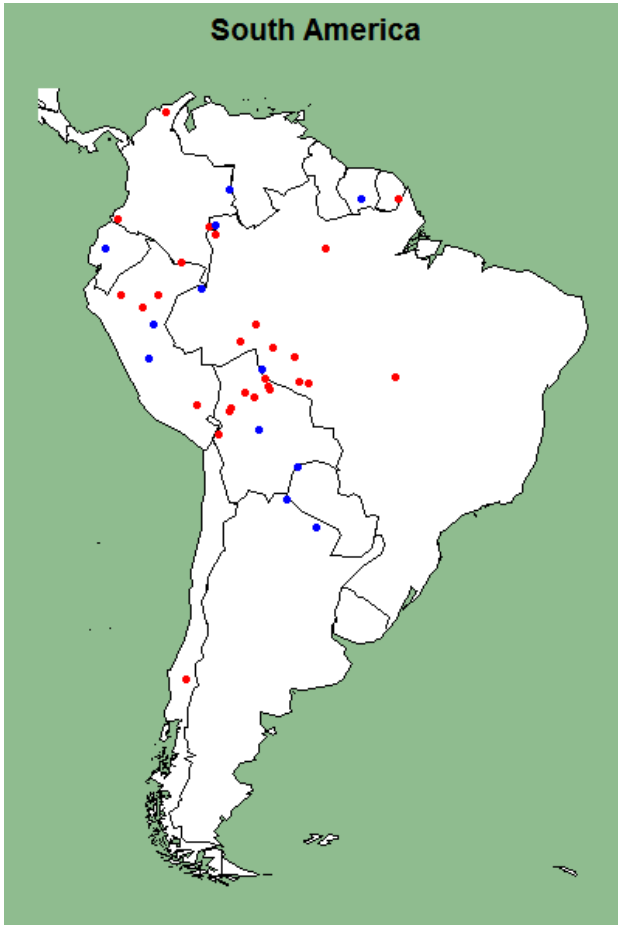
```
mySAdata$my.col[mySAdata$CASE == "N"] = "blue"
```

4. use the $my.col column of data as the values of the col attribute

```
points(mySAdata$LONGITUDE, mySAdata$LATITUDE, pch=20, col=mySAdata$my.col)
```

Making maps in R

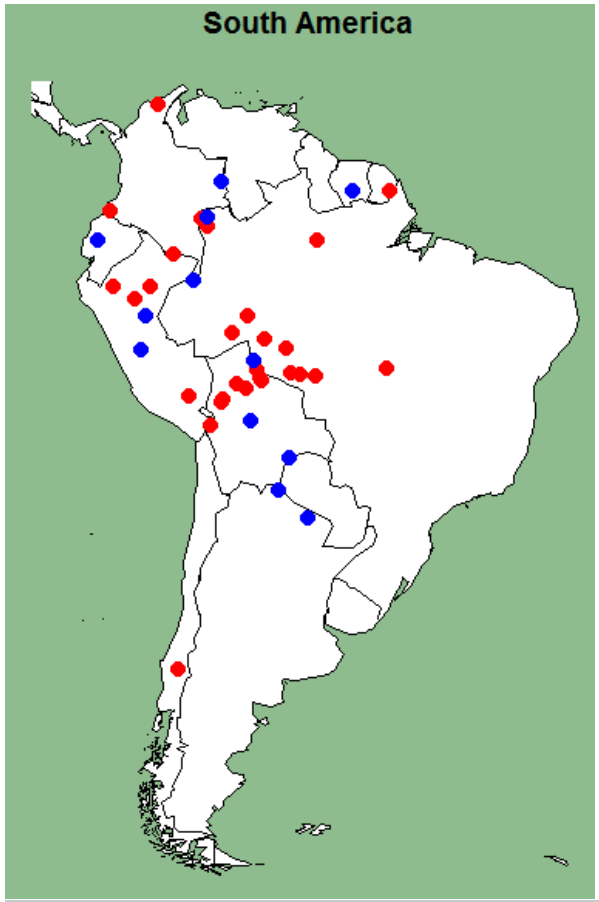
Step 5: Plotting data points onto a map



Now maybe we want to make the dots a little bigger

Making maps in R

Step 5: Plotting data points onto a map



What about language names?
You can use the function text()
Type help(text)

```
> points(mySAdata$LONGITUDE, mySAdata$LATITUDE, pch=20, col=mySAdata$my.col, cex=2)
>
```

Making maps in R

Description

`text` draws the strings given in the vector `labels` at the coordinates given by `x` and `y`. `y` may be missing since `xy.coords(x,y)` is used for construction of the coordinates.

Usage

```
text(x, ...)

## Default S3 method:
 text(x, y = NULL, labels = seq_along(x), adj = NULL,
     pos = NULL, offset = 0.5, vfont = NULL,
     cex = 1, col = NULL, font = NULL, ...)
```

Arguments

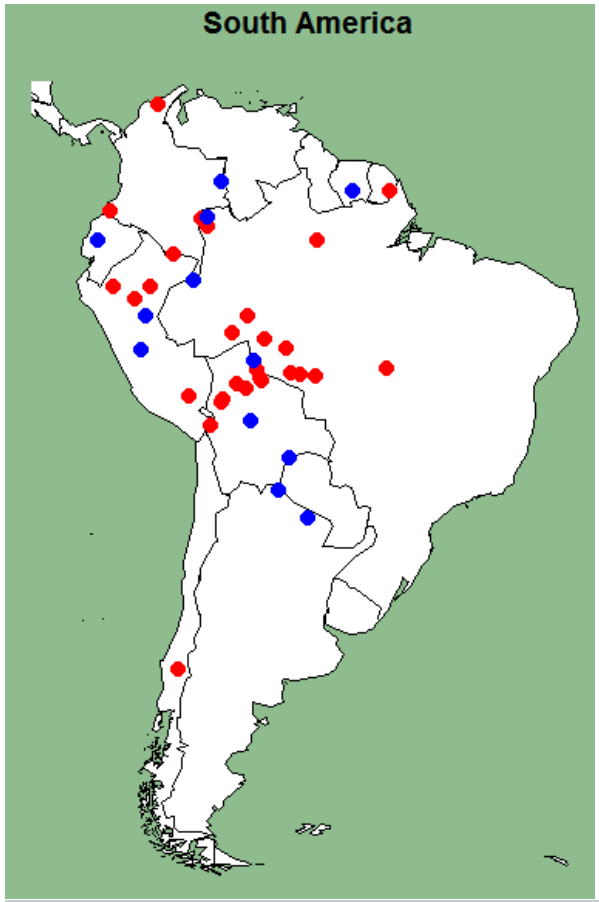
`x, y`	numeric vectors of coordinates where the text `labels` should be written. If the length of `x` and `y` differs, the shorter one is recycled.
`labels`	a character vector or expression specifying the *text* to be written. An attempt is made to coerce other language objects (names and calls) to expressions, and vectors and other classed objects to character vectors by `as.character`. If `labels` is longer than `x` and `y`, the coordinates are recycled to the length of `labels`.
`adj`	one or two values in *[0, 1]* which specify the x (and optionally y) adjustment of the labels. On most devices values outside that interval will also work.
`pos`	a position specifier for the text. If specified this overrides any `adj` value given. Values of `1`, `2`, `3` and `4`, respectively indicate positions below, to the left of, above and to the right of the specified coordinates.
`offset`	when `pos` is specified, this value gives the offset of the label from the specified coordinate in fractions of a character width.
`vfont`	`NULL` for the current font family, or a character vector of length 2 for Hershey vector fonts. The first element of the vector selects a typeface and the second element selects a style. Ignored if `labels` is an expression.
`cex`	numeric character expansion factor; multiplied by `par("cex")` yields the final character size. `NULL` and `NA` are equivalent to `1.0`.
`col, font`	the color and (if `vfont = NULL`) font to be used, possibly vectors. These default to the values of the global graphical parameters in `par()`.
`...`	further graphical parameters (from `par`), such as `srt`, `family` and `xpd`.

For x and y, we can use ....

For the labels we can use ...

Making maps in R

Step 5: Plotting data points onto a map



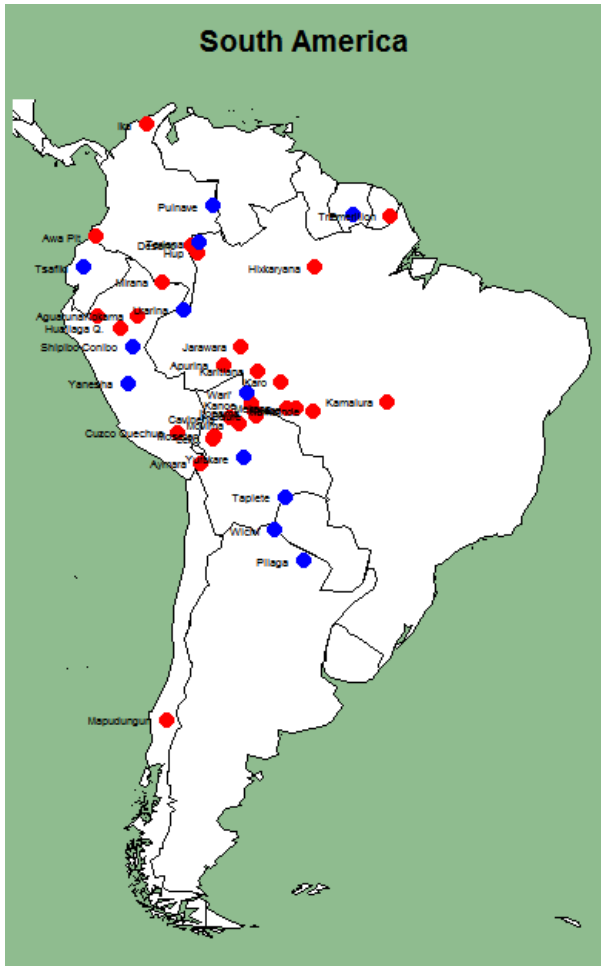
What about language names?
You can use the function text()
Type help(text)

```
> points(mySAdata$LONGITUDE, mySAdata$LATITUDE, pch=20, col=mySAdata$my.col, cex=2)
>
```

Making maps in R

Step 5: Plotting data points onto a map

With these larger maps with data points relatively close to each other, the labels do not work very well. We can try to fix it by using the iso codes instead

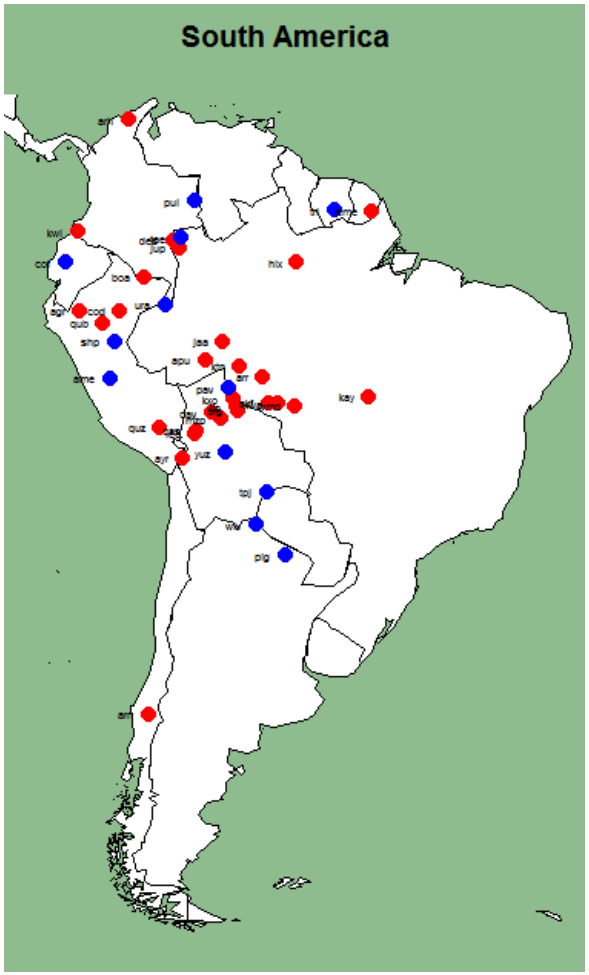



```
> text(mySAdata$LONGITUDE, mySAdata$LATITUDE, labels=mySAdata$LANGUAGE,
+ cex=.4, adj=.2, pos=2)
>
```

Making maps in R

Step 5: Plotting data points onto a map

We may also choose to place the labels directly onto the data points by losing the pos= and adj= arguments



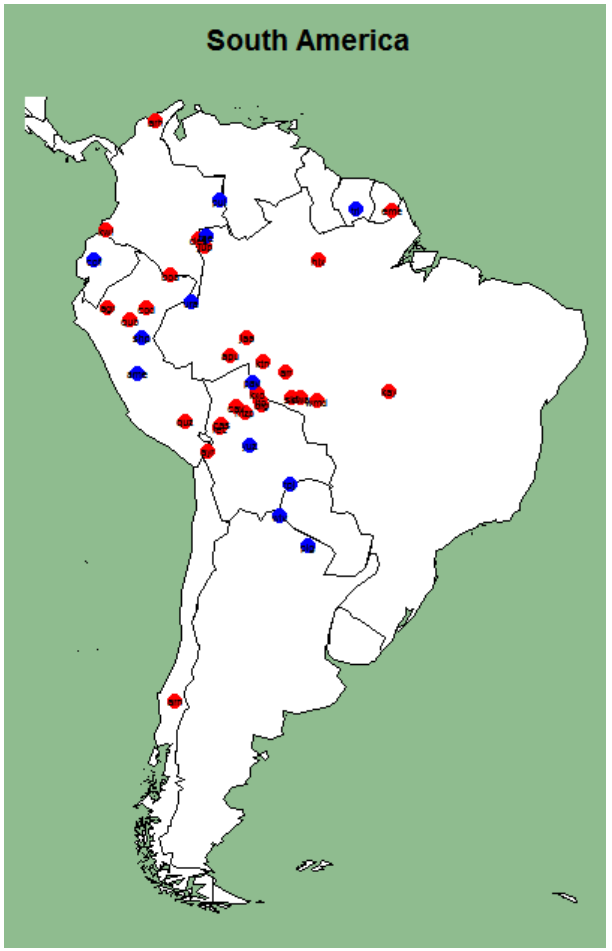

```
> text(mySAdata$LONGITUDE, mySAdata$LATITUDE, labels=mySAdata$LANGUAGE,
+ cex=.4, adj=.2, pos=2)
>
```

Making maps in R

Step 5: Plotting data points onto a map

We may also choose to place the labels directly onto the data points by losing the pos= and adj= arguments

...and change the font color to something else with the argument col= within the text function




```
> map(xlim=c(-85, -35), ylim=c(-55,12), fill=TRUE, col="white", bg="darkseagreen")
> title(main="South America")
> points(mySAdata$LONGITUDE, mySAdata$LATITUDE, pch=20, col=mySAdata$my.col, cex=2)
> text(mySAdata$LONGITUDE, mySAdata$LATITUDE, labels=mySAdata$ISOCODE, cex=.4)
>
```

Making maps in R

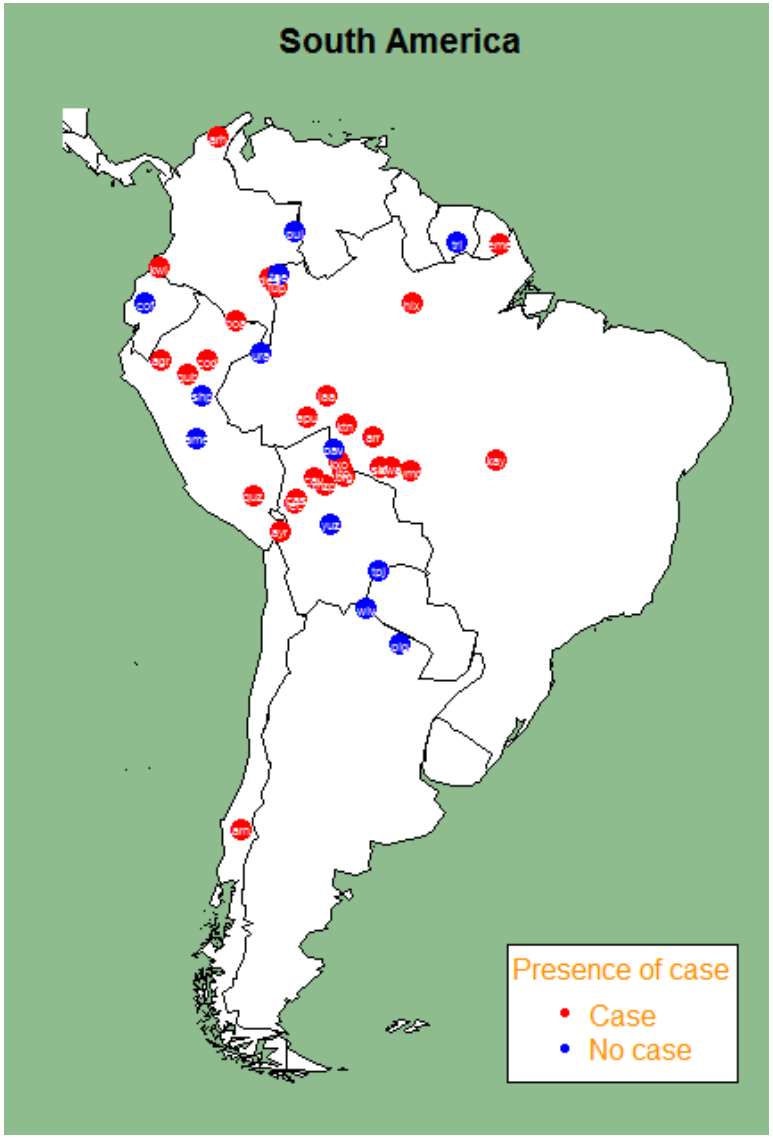
Step 6: Adding a legend

```
legend(
"bottomright",
legend=c("Case", "No case"),
pch=20,
col=c("red", "blue"),
title="Presence of case",
text.col="darkorange", bg="white")
```

- Function legend() creates a legend
- Determine the position of the legend
- Add names to the categories
- Determine form of symbol in legend
- Determine color of legend symbols
- Add title to a legend
- Determine some further graphical parameters such as text color and background color

Making maps in R

Ta-taaaa!



Making maps in R

Some things to play around with

- Try and do the same for word order
- Try and project plots for combinations of features with the logical operator &
- Once you have a dataset with latitude and longitude, you can determine the range of the map you need by using the functions range() or extendrange() which only display the part of the map between the points in the dataset that are furthest apart (range) plus a small margin (extendrange).

See (also for much more on coloring):
http://www.mpi.nl/departments/independent-research-groups/evolutionary-processes/tools/mapping-with-r