Introduction to Graphics with ggplot2

Reaction 2017

Flavio Santi

Sept. 6, 2017

Graphics with ggplot2

> *ggplot2 [...] allows you to produce graphics using the same structured thinking that you use to design an analysis, reducing the distance between a plot in your head and one on the page. It is especially helpful for students who have not yet developed the structured approach to analysis used by experts.*
>
> Hadley Wickham

Wickham, H. (2009), *ggplot2*, Springer.

How ggplot2 works (1)

> *Wilkinson (2005) created the grammar of graphics to describe the deep features that underlie all statistical graphics. The grammar of graphics is an answer to a question: what is a statistical graphic? The layered grammar of graphics (Wickham, 2009) builds on Wilkinson's grammar, focussing on the primacy of layers and adapting it for embedding within R.*

Hence, a graphic (may) consists of (see Wickham, 2009, p. 3):

- The **data** that you want to visualise and a set of aesthetic mappings describing how variables in the data are mapped to aesthetic attributes that you can perceive.
- Geometric objects, **geoms** for short, represent what you actually see on the plot: points, lines, polygons, etc.

How ggplot2 works (2)

- Statistical transformations, **stats** for short, summarise data in many useful ways. For example, binning and counting observations to create a histogram, or summarising a 2d relationship with a linear model. Stats are optional, but very useful.
- The **scales** map values in the data space to values in an aesthetic space, whether it be colour, or size, or shape. Scales draw a legend or axes, which provide an inverse mapping to make it possible to read the original data values from the graph.
- A coordinate system, **coord** for short, describes how data coordinates are mapped to the plane of the graphic. It also provides axes and gridlines to make it possible to read the graph. We normally use a Cartesian coordinate system, but a number of others are available, including polar coordinates and map projections.

How ggplot2 works (3)

- A **facet**ing specification describes how to break up the data into subsets and how to display those subsets as small multiples. This is also known as conditioning or latticing/trellising.

Remember:

ggplot always requires data to be stored in data.frame objects

Some examples (1)

Consider the following data frame:

```
> str(borsadf)
'data.frame':   2610 obs. of  5 variables:
 $ date     : Date, format: "2002-05-07" "2002-05-08" ...
 $ SP500    : num  1049 1089 1073 1055 1075 ...
 $ rendSP500: num  NA 0.0368 -0.0147 -0.0169 0.0184 ...
 $ ENI      : num  16.5 16.4 16.1 15.9 16 ...
 $ rendENI  : num  NA -0.00357 -0.01747 -0.0153 0.00738 ...
> borsadf[1:4,]
        date   SP500   rendSP500     ENI      rendENI
1 2002-05-07 1049.49          NA 16.4604           NA
2 2002-05-08 1088.85  0.03681776 16.4017 -0.003572508
3 2002-05-09 1073.01 -0.01465431 16.1177 -0.017466941
4 2002-05-10 1054.99 -0.01693650 15.8729 -0.015304794
```

Some examples (2)

Load the ggplot2 package:

```
> library(ggplot2)
```

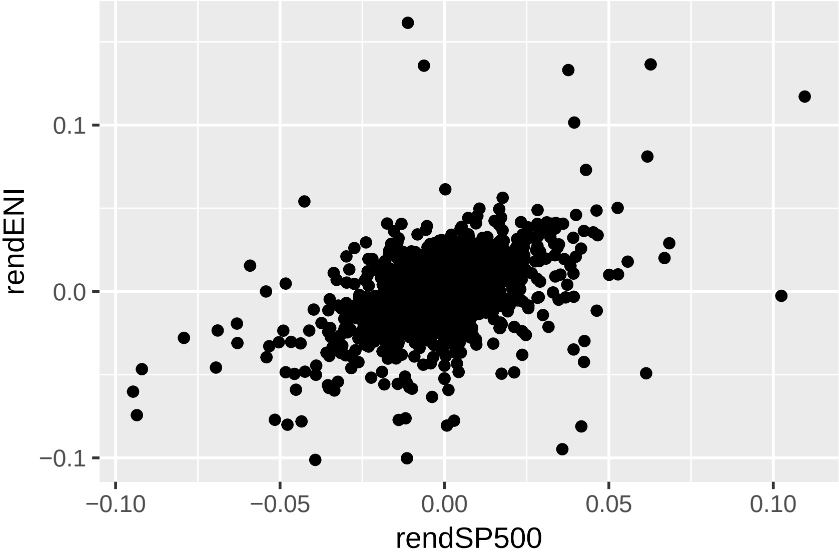
The following command generate a scatterplot of daily returns of S&P500 and ENI:

```
> ggplot(borsadf,aes(x=rendSP500,y=rendENI))+geom_point()
```

or, in a script:

```
> gr1<- ggplot(borsadf,aes(x=rendSP500,y=rendENI))+
+   geom_point()
> print(gr1)
```

Some examples (3)



Some examples (4)

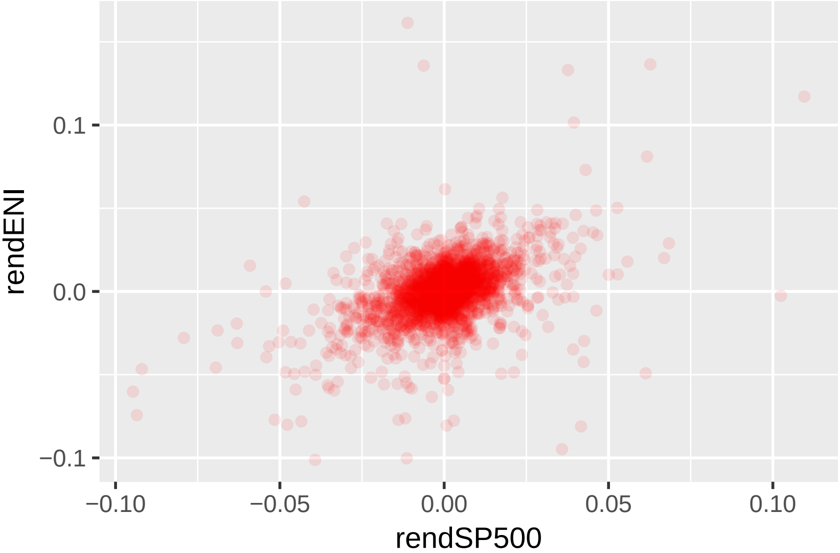
It is also possible to set the colour, and the transparency of **all** points:

```
> gr1<- ggplot(borsadf,aes(x=rendSP500,y=rendENI))+
+   geom_point(colour="red",alpha=0.1)
> print(gr1)
```

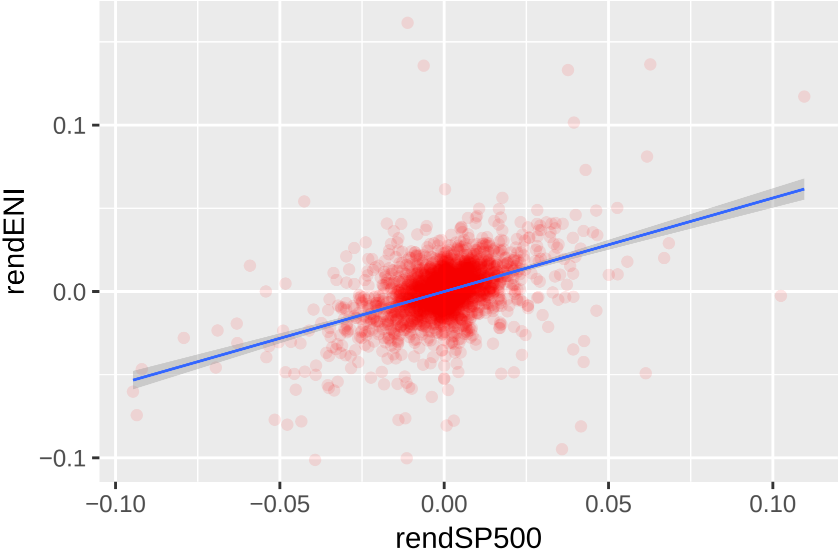
And add a linear regression line with 99% confidence inteval:

```
> gr1<- ggplot(borsadf,aes(x=rendSP500,y=rendENI))+
+   geom_point(colour="red",alpha=0.1)+
+   stat_smooth(method="lm",formula=y~x,level=0.99,
+     size=0.5)
> print(gr1)
```

Some examples (5)



Some examples (6)



Some examples (7)

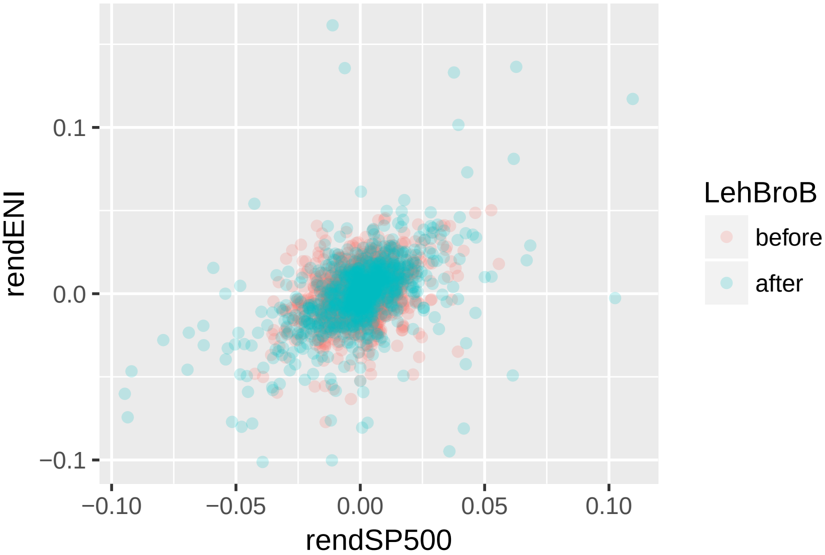
Assume that we want to distinguish between returns before and after the Lehman Brothers bankruptcy (15 September 2008). We define a new variable (as a factor) called “LehBroB”:

```
> borsadf$LehBroB<- factor(borsadf$date<"2008-09-15",
+   c(TRUE,FALSE),c("before","after"))
```

Now we plot the points as before, but we specify (in “aes”) that points should be coloured according to the value of variable “LehBroB”:

```
> gr1<- ggplot(borsadf,aes(x=rendSP500,y=rendENI))+
+   geom_point(aes(colour=LehBroB),alpha=0.1)
> print(gr1)
```

Some examples (8)

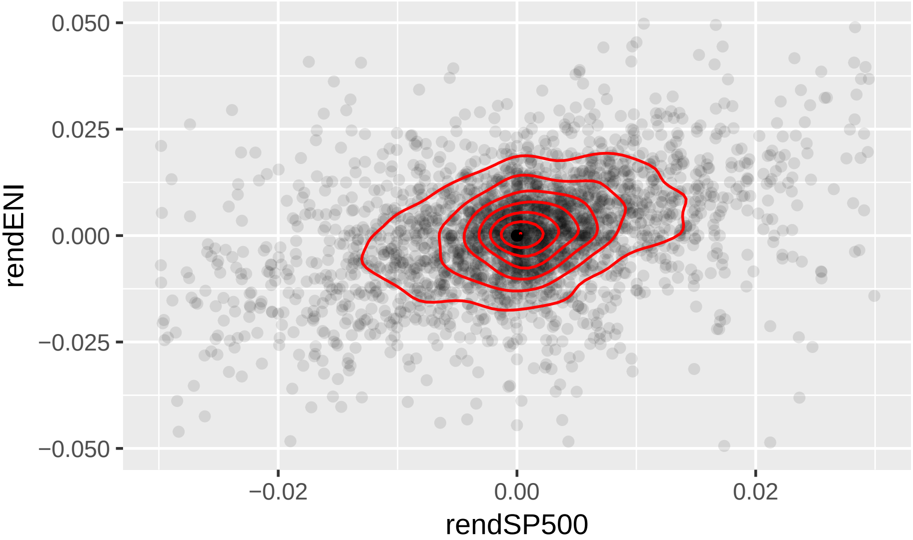


Some examples (9)

It is also possible to derive a 2-dimensional density function:

```
> gr1<- ggplot(borsadf,aes(x=rendSP500,y=rendENI))+
+   geom_point(alpha=0.1)+
+   geom_density2d(colour="red",binwidth=0.001,bins=7)+
+   xlim(c(-0.03,0.03))+
+   ylim(c(-0.05,0.05))
> print(gr1)
```

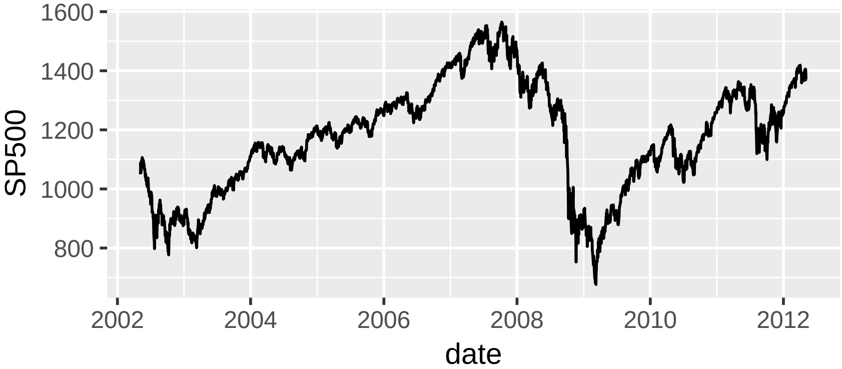
Some examples (10)



Some examples (11)

Now we are going to draw the time series of the price of the S&P500:

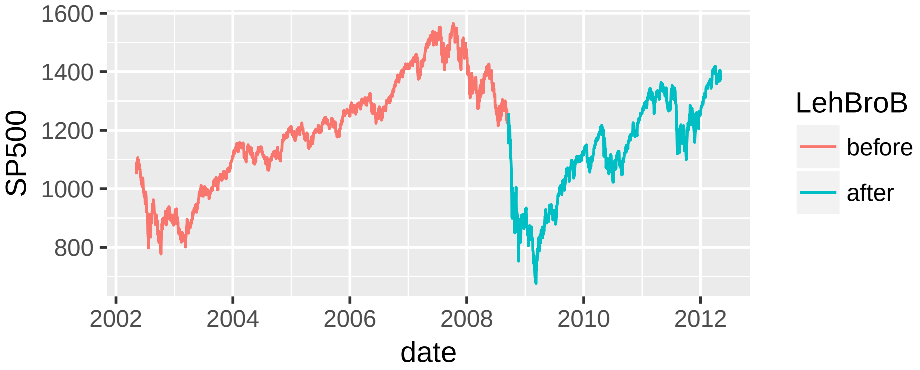
```
> gr1<- ggplot(borsadf,aes(x=date,y=SP500))+geom_line()
> print(gr1)
```



Some examples (12)

You can colour the line according to the value of another variable:

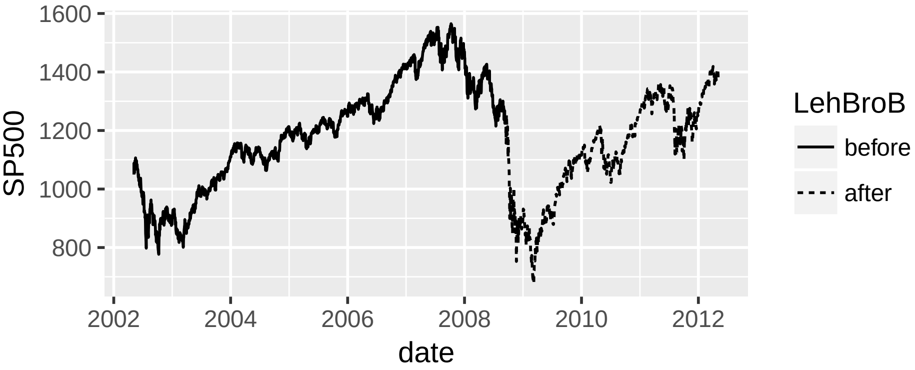
```
> gr1<- ggplot(borsadf,aes(x=date,y=SP500))+
+   geom_line(aes(colour=LehBroB))
> print(gr1)
```



Some examples (13)

Otherwise, you can change the type of the line:

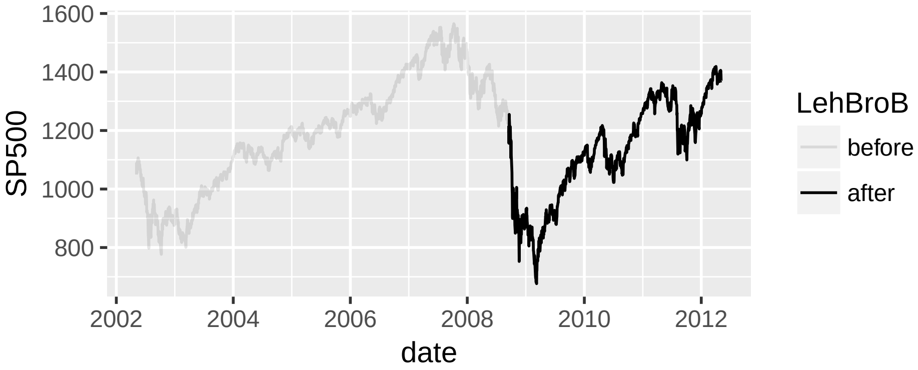
```
> gr1<- ggplot(borsadf,aes(x=date,y=SP500))+
+   geom_line(aes(linetype=LehBroB))
> print(gr1)
```



Some examples (14)

Or any other aestetics:

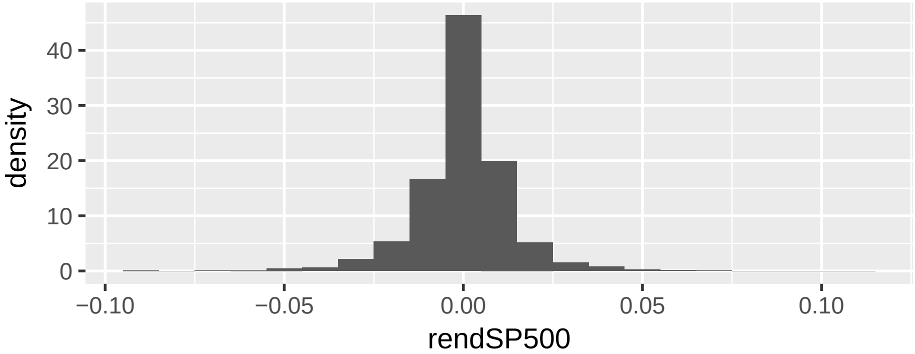
```
> gr1<- ggplot(borsadf,aes(x=date,y=SP500))+
+   geom_line(aes(alpha=LehBroB))
> print(gr1)
```



Some examples (15)

The histogram of S&P500 returns can be obtained with:

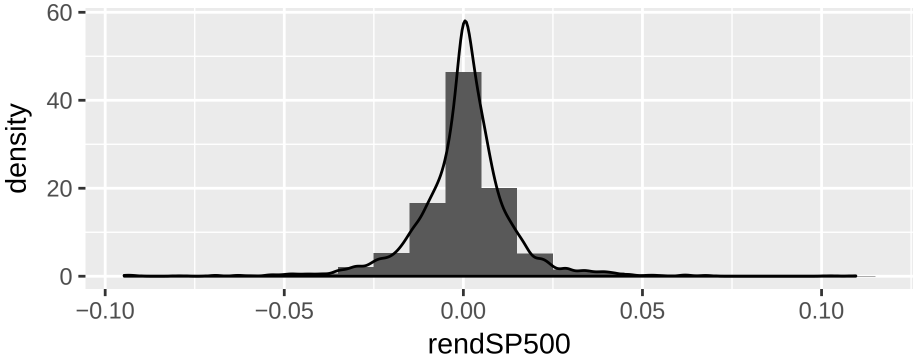
```
> gr1<- ggplot(borsadf,aes(x=rendSP500,y=..density..))+
+   geom_histogram(binwidth=0.01)
> print(gr1)
```



Some examples (16)

And it is possible to add an estimated density function:

```
> gr1<- ggplot(borsadf,aes(x=rendSP500,y=..density..))+
+   geom_histogram(binwidth=0.01)+geom_density()
> print(gr1)
```



Some examples (17)

If you need to plot more than one variable on the same plot, the `data.frame` object should have a long-table format.

Assume that you want to compare the distributions of returns of ENI and S&P500 by means of a boxplot.

Extract the variables you are interested in:

```
> tempBorsa<- borsadf[,c("date","rendENI","rendSP500")]
> head(tempBorsa)
        date      rendENI   rendSP500
1 2002-05-07           NA          NA
2 2002-05-08 -0.003572508  0.03681776
3 2002-05-09 -0.017466941 -0.01465431
4 2002-05-10 -0.015304794 -0.01693650
5 2002-05-13  0.007375290  0.01837999
6 2002-05-14  0.012176663  0.02092311
```

Some examples (18)

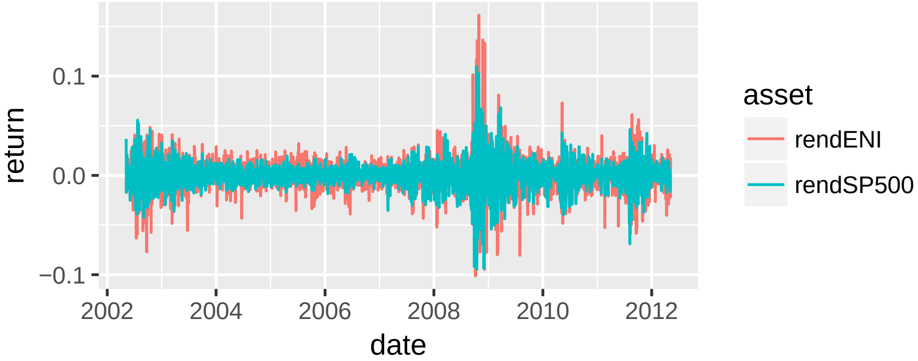
Convert tempBorsa into long-table format:

```
> library(reshape2)
> tempBorsa<- melt(tempBorsa,id.vars="date",
+   variable.name="asset",value.name="return")
> str(tempBorsa)
'data.frame':   5220 obs. of  3 variables:
 $ date  : Date, format: "2002-05-07" "2002-05-08" ...
 $ asset : Factor w/ 2 levels "rendENI","rendSP500": 1 1 1 1 1
 $ return: num  NA -0.00357 -0.01747 -0.0153 0.00738 ...
> head(tempBorsa,3)
        date   asset       return
1 2002-05-07 rendENI           NA
2 2002-05-08 rendENI -0.003572508
3 2002-05-09 rendENI -0.017466941
```

Some examples (19)

Plot the time series:

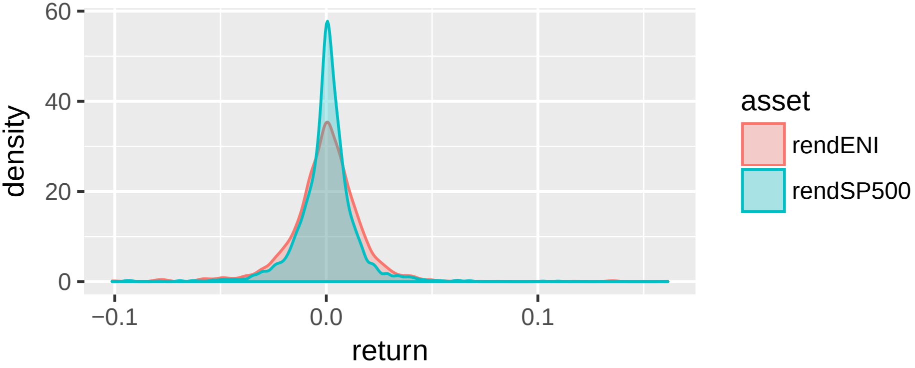
```
> gr1<- ggplot(tempBorsa,aes(x=date,y=return))+
+   geom_line(aes(colour=asset))
> print(gr1)
```



Some examples (20)

Plot the densities:

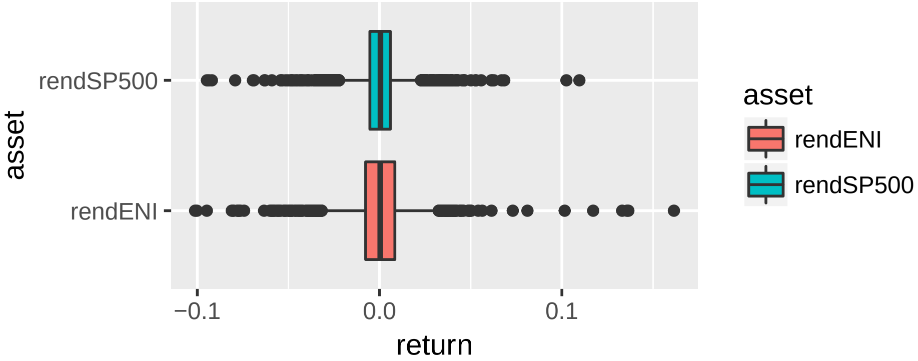
```
> gr1<- ggplot(tempBorsa,aes(x=return,y=..density..,
+   fill=asset,colour=asset))+geom_density(alpha=0.3)
> print(gr1)
```



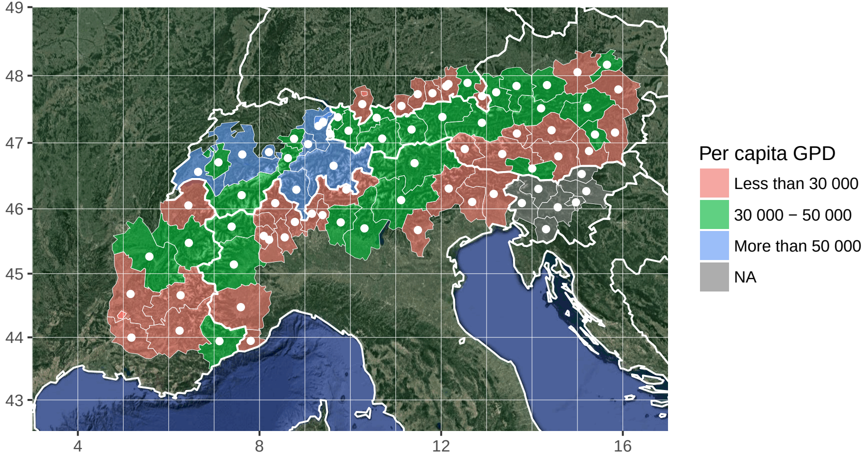
Some examples (21)

Plot the densities:

```
> gr1<- ggplot(tempBorsa,aes(x=asset,y=return,fill=asset))+
+   geom_boxplot()+coord_flip()
> print(gr1)
```



Some examples (22)



thank you!