Financial Risk Forecasting
Seminar 3

Jon Danielsson
London School of Economics

Version 4.0 August 2024

3 Distributions and analysis

3.1 The plan for this week

1. Work with distributions
2. Visualise, analyse, and comment on the prices of a stock
3. Perform graphical analyses and statistical tests

3.2 Loading data and libraries

```
library(tseries)
library(car)
library(lubridate)
library(zoo)
library(moments)
```

```
load('Returns.RData')
load('Prices.RData')
```

3.3 Links from the R notebook

Descriptive statistics

3.4 Work with distributions

We do a lot of work with statistical distributions, such as the normal, log-normal, Student-t, binomial, Bernoulli, and Chi-square. R can, of course, handle a lot more distributions, but those three are the ones we mostly use here. Each comes in 4 versions:

1. Density (pdf) — `d`
2. Distribution or cumulative density (cdf) — `p`

3. Quantiles — `q`
4. Random numbers — `r`

The function name is created by pre-fixing one of these four letters to the distribution name. For the normal, `dnorm, pnorm, qnorm, rnorm` and the t `dt, pt, qt, rt`.

We deal with random numbers in a later Seminar.

Here is an example showing how to plot the distributions over their domain.

```
x=seq(-3,3,length=1000)
z=seq(0,1,length=1000)
par(mfrow=c(2,2))
plot(x,dnorm(x), main="Normal Density")
plot(x,pnorm(x), main="Cumulative Density")
plot(z,qnorm(z), main="Normal Quantile")
```

3.5 Comparing the normal distribution with the Student-t

The Student-t distribution has fatter tails than the normal.

```
x <- seq(-3, 3, length=1000)
normal <- dnorm(x)
st2 <- dt(x, df = 2)
st3 <- dt(x, df = 3)
st10 <- dt(x, df = 10)
plot(x, normal, type = "l", main = "Comparing distributions", col = 1, xlab = "x", ylab =
lines(x, st2, col = 2)
lines(x, st3, col = 3)
lines(x, st10, col = 4)
legend("topright",
    legend = c("Normal", "T - 2 df", "T - 3 df", "T - 10 df"),
    col = c(1:4),
    lty=1,
    bty='n'
)
```

3.6 Visualising and commenting on the price of a stock

```
head(Prices)
```

Now, let's extract the information for General Electrics.

```
plot(Prices$GE, type = "l", main = "Price of GE")
```

This plot is useless for analysis without including dates:

```
date.ts=ymd(Prices$date)
plot(date.ts,Prices$GE, type = "l", main = "Price of GE")
```

Find what the maximum price of GE was and when it was reached:

```
max(Prices$GE)
date.ts[Prices$GE == max(Prices$GE)]
GE[which.max(Prices$GE),]
```

Jack Welch, who was the CEO of GE for 20 years, retired on September 7, 2001. He was considered to be one of the most valuable CEOs of all time. Let's add a vertical line on our plot to reflect this:

```
plot(date.ts,Prices$GE, type = "l", main = "Price of GE")
abline(v = ymd(20010907), lwd = 2, col = "blue")
```

3.7 Using zoo

Converting a vector into a time series vector, i.e. one where R knows the date, can be useful.

```
Prices.ts=zoo(Prices[,2:dim(Prices)[2]],order.by=date.ts)
head(Prices.ts)
```

```
plot(Prices.ts)
plot(Prices.ts,screen=FALSE)
```

Now, let's zoom into the crisis in 2008.

```
Prices.crisis=window(Prices.ts,
      start= ymd(20080101),
      end=ymd(20100101)
)
plot(Prices.crisis,screen=FALSE,col=1:8)
legend("topright",
      legend=names(Prices.crisis),
      lty=1,
      col=1:8,
      bty='n'
)
```

3.8 Normalising prices

```
first=unclass(Prices.crisis[1,])
for(i in 1:dim(Prices.crisis)[2]) Prices.crisis[,i]=Prices.crisis[,i]/first[i]
plot(Prices.crisis,screen=FALSE,col=1:8)
legend("bottomleft",
      legend=names(Prices.crisis),
      lty=1,
      col=1:8,
      bty='n'
)
```

Try to repeat the same analysis for returns.

3.9 Graphical analyses and statistical tests

We can do some of the basic statistical and graphical analysis shown at the start of the course. See the statistics chapter in the risk forecasting notebook.

Pick JP Morgan

```
y=Returns$JPM
```

First, print some summary statistics and run the Jarque Bera and Box tests.

```
mean(y)
sd(y)
skewness(y)
kurtosis(y)
jarque.bera.test(y)
Box.test(y, type = "Ljung-Box")
Box.test(y^2, type = "Ljung-Box")
```

Then, plot the autocorrelation function of returns and returns squared. What information does the latter plot provide?

```
acf(y, main = "Autocorrelation of returns")
acf(y^2, main = "Autocorrelation of returns squared")
```

These ACF plots look like they could be nicer. The notebook shows some better alternatives.

Finally, the QQ plot can be informative about the distribution of the returns, especially in the tails. Note the difference between the lower and upper tails.

```
x=qqPlot(y, distribution = "norm", envelope = FALSE,xlab="normal")
x=qqPlot(y, distribution = "t", df = 4, envelope = FALSE,xlab="t(4)")
x=qqPlot(y, distribution = "t", df = 3.5, envelope = FALSE,xlab="t(3.5)")
x=qqPlot(y, distribution = "t", df = 3, envelope = FALSE,xlab="t(3)")
```

3.10 Recap

In this seminar, we have covered the following:

- Distributions
- Testing
- Time series plots

Some new functions used:

- `zoo()`
- `abline()`
- `skewness()`
- `zoo()`
- `jarque.bera.test()`
- `Box.test()`
- `acf()`
- `qqPlot()`

3.11 Optional exercises

1. Make a table with the key summary statistics for all stocks in our sample
2. Make a plot for each stock and put the key sample statistics on each figure