



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Tento projekt je spolufinancován Evropským sociálním fondem a Státním rozpočtem ČR
InoBio – CZ.1.07/2.2.00/28.0018

Statistical Analysis in Ecology using R

R graphics

Ing. Daniel Volařík, Ph.D.

12. 11. 2013



INVESTMENTS IN EDUCATION DEVELOPMENT

Recap

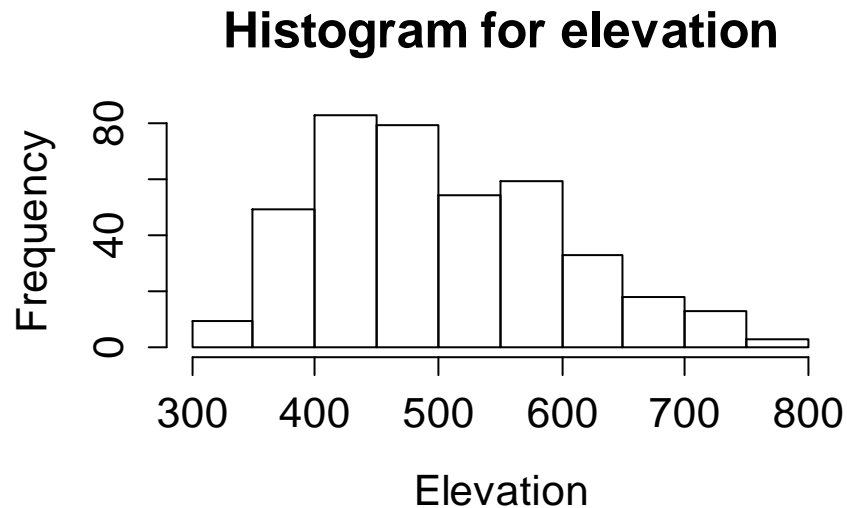
- R object, how to import data into R
- Subsetting data (function select or [])
- Models
 - Central tendency – mean, median
 - Dispersion – standard deviation
- Normal distribution
- Correlations

Why graphs?

- Plotting data is a very important part of data exploration – it should be done easily
- Graphs are a perfect way how to present your data in publications – they should look perfect
- Both you can do in R

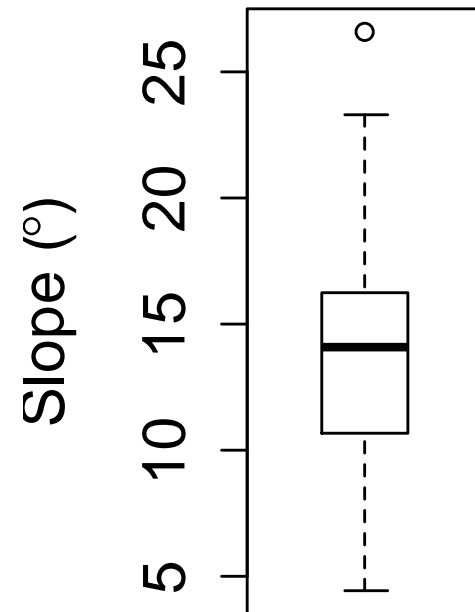
Histograms

- How are data distributed
- Data are divided into cells and count of samples (or their density or relative frequency) in each cell is plotted using bars.
- Function hist()



Boxplots

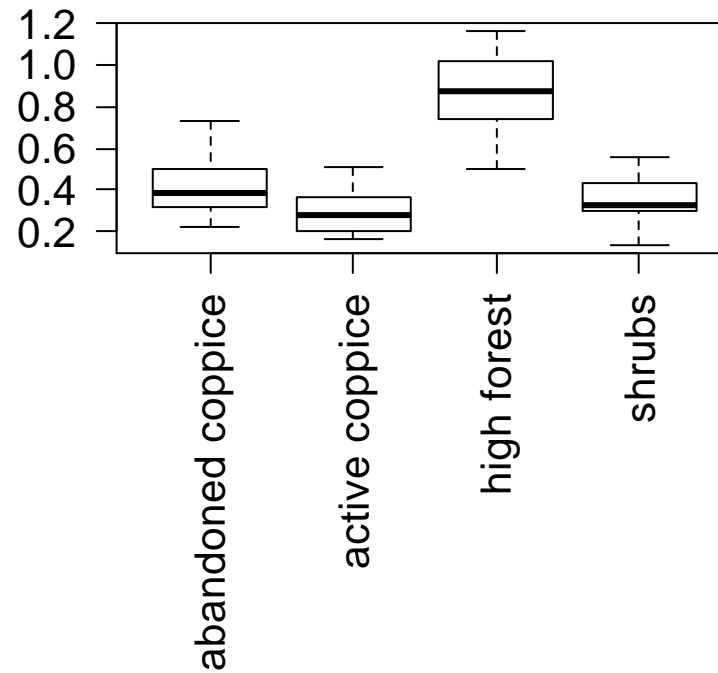
- show data distribution (mean, spread, shape, outliers) in condensed way.
- Constructed from:
 - Midpoint – usually median
 - Hinges (ends of the box) – 25% and 75% quartiles (50 % of values are inside the box) – interquartile range
 - Lines (whiskers) are drawn to the last value which falls within 1.5 times the interquartile range
 - Outliers – any values out of this range
- Function `boxplot()`



Conditioned boxplots

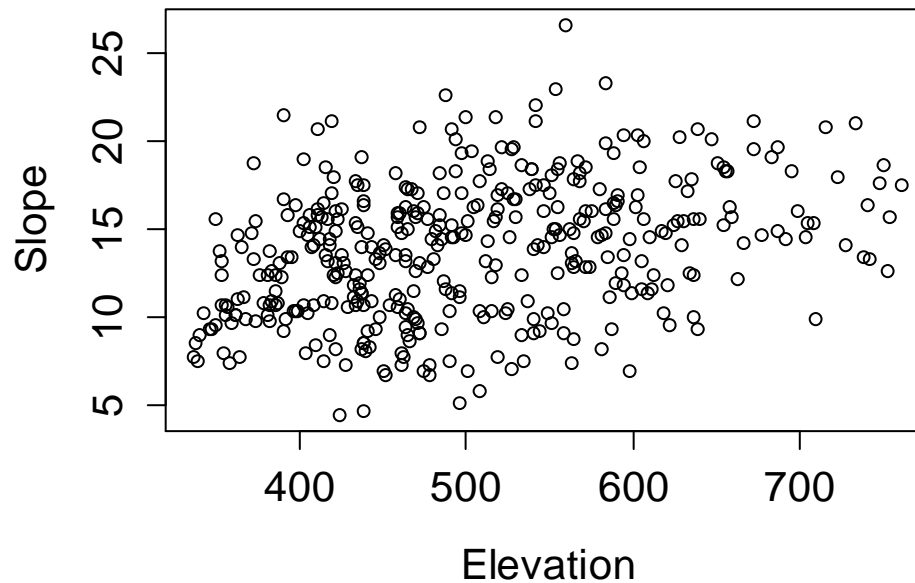
- we can put several boxplots into one graph – e.g. subsets for some factor

> boxplot(variable ~ factor, data = data)



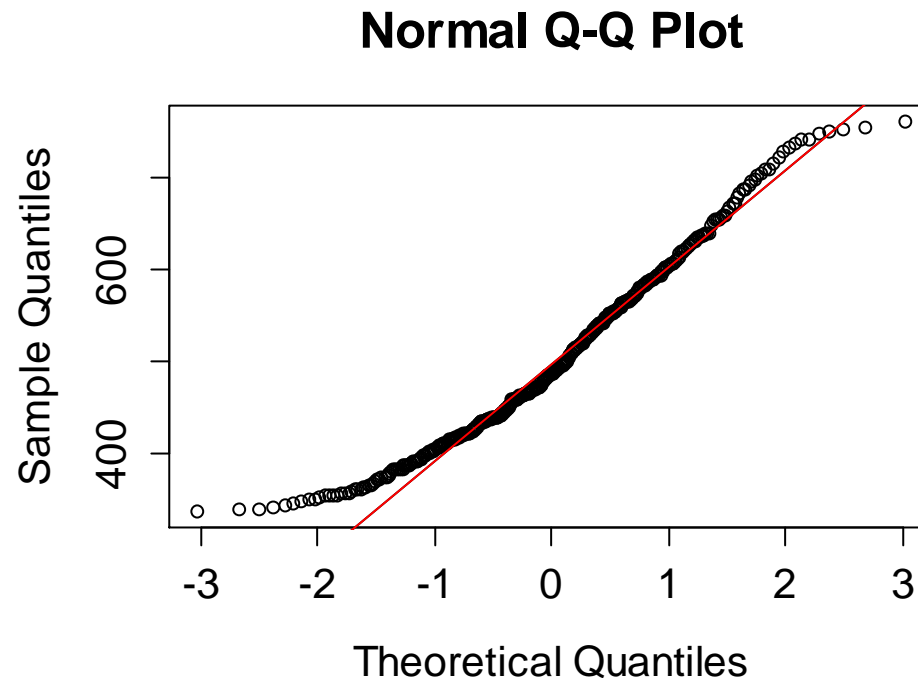
Scatterplot

- 2 variables plotted against each other
 - Tool to find relationship between them
 - Function plot()
- > plot(variable _1 ~ variable _2)



QQ plots (Quantile-Quantile Plots)

- Used to compare 2 distributions (e.g. sampled data against normal distribution)
- More powerful than histograms, but needs more skills to interpret
- Quantiles of compared samples are plotted against each other.
- In R functions `qqnorm()` and `qqline()`



Base graphic in R

- Main function for graphs is `plot()`
- Huge number of parameter to customise graphs
 - `xlab=""` – label o x axis
 - `ylab=""`
 - `type=""` – the type of plot produced (“p” for individual points, “l” for lines, “b” points connected by lines, “h” vertical lines from point to zero, ...)
 - `main=""` – the main title of graph
 - `las = 1` – to change orientation of axis labels (0 – paralel with axis, 1 – horizontal, 2 – perpendicular to axis, 3 – vertical)

How to add additional elements to plot

- `points()` – adds points to existing plot, parameters are the same as for `plot()`
- `abline()` – add lines to the plot
 - could be specified by intercept (a) and slope (b)
 - horizontal line – specifying its y value $h =$
 - vertical line – specifying its x value $v =$
- `legend()` – to add a legend

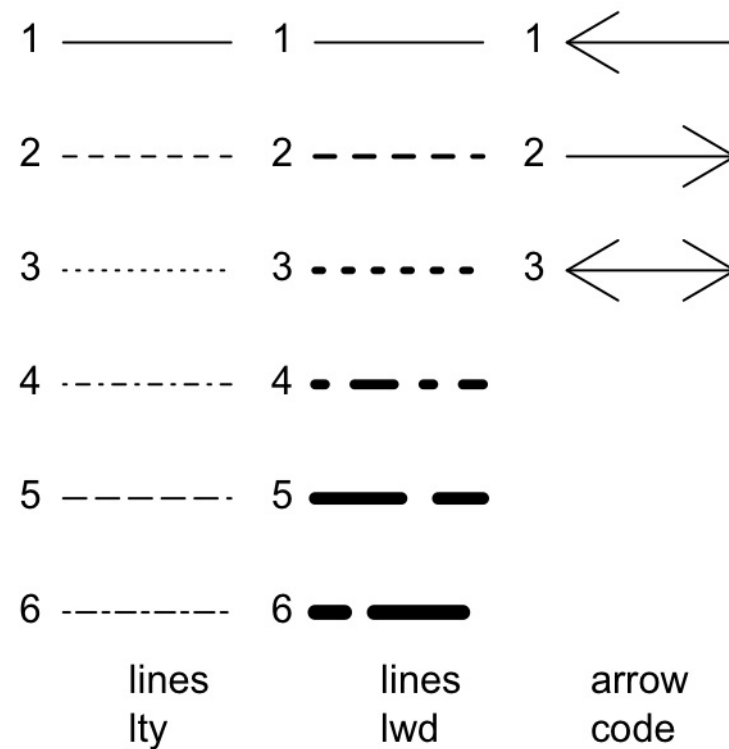
How to customise symbol for points a character of lines?

- Changing color – parameter col (col = “green“)
- Changing symbol for points – parameter pch (pch = 1)
- Changing line style – parameter lty
- Changing line width – parameter lwd

Point symbols



Line character



ggplot graphic

- "Elegant graphics for data analysis"
- it has its own grammar (based on the Grammar of Graphics – Wilkinson 2005); this includes new terminology
- easy to use and very flexible – LM, GLM fits, smoothed curves, facets can be easily added.
- started in 2005
- Package ggplot2 by Wickham
- <http://ggplot2.org/>

ggplot graphic

- components of grammar
 - data and a set of aesthetic mapping – how the variables are mapped to aesthetic attributes
 - geometric object – geoms – what will be on the plot – points, lines, polygons, ...
 - stats – statistical transformations – e.g. binning and counting objects to create histograms, linear model, ...
 - scales – map values in the data space to values in aesthetic space – could be colour, size, shape.
 - coord – how data coordinates are mapped to the plain of graphic
 - facets – how to divide data into subsets and how to draw them.

ggplot graphic

- grammar doesn't specify how data should look like
- this is done by theming system

ggplot graphic

- function `qplot()` (abbreviation for quick plot)

```
> qplot(x, y, data = data)
```

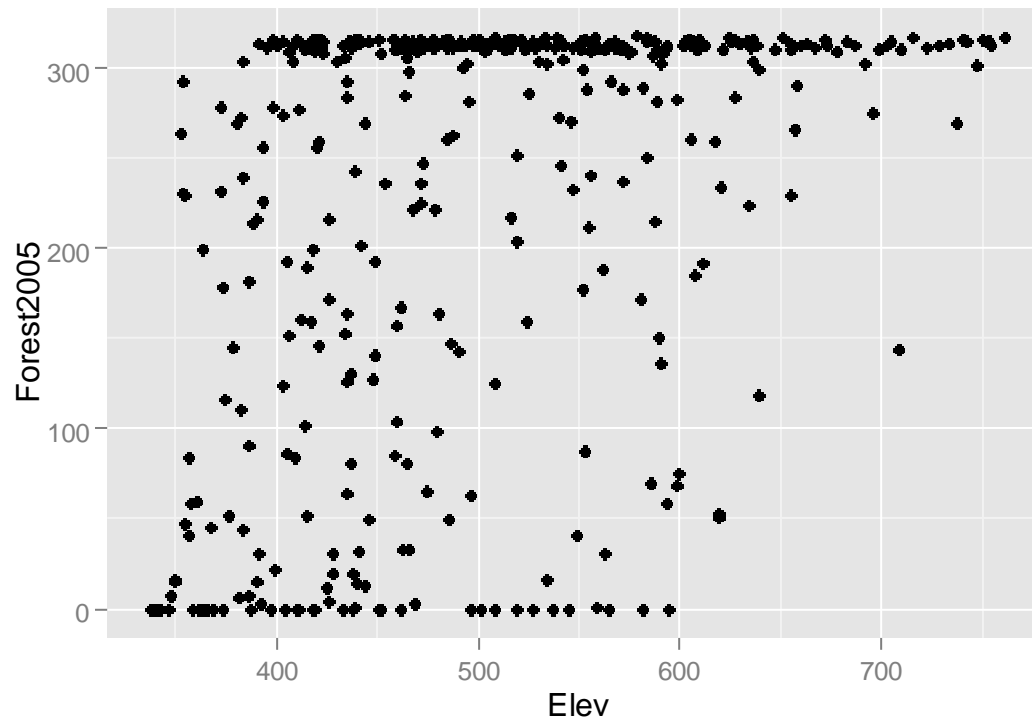
colours or shapes according to some factor are easily added by:

```
> qplot(x, y, data = data, colour = factor)
```

```
> qplot(x, y, data = data, shape = factor)
```

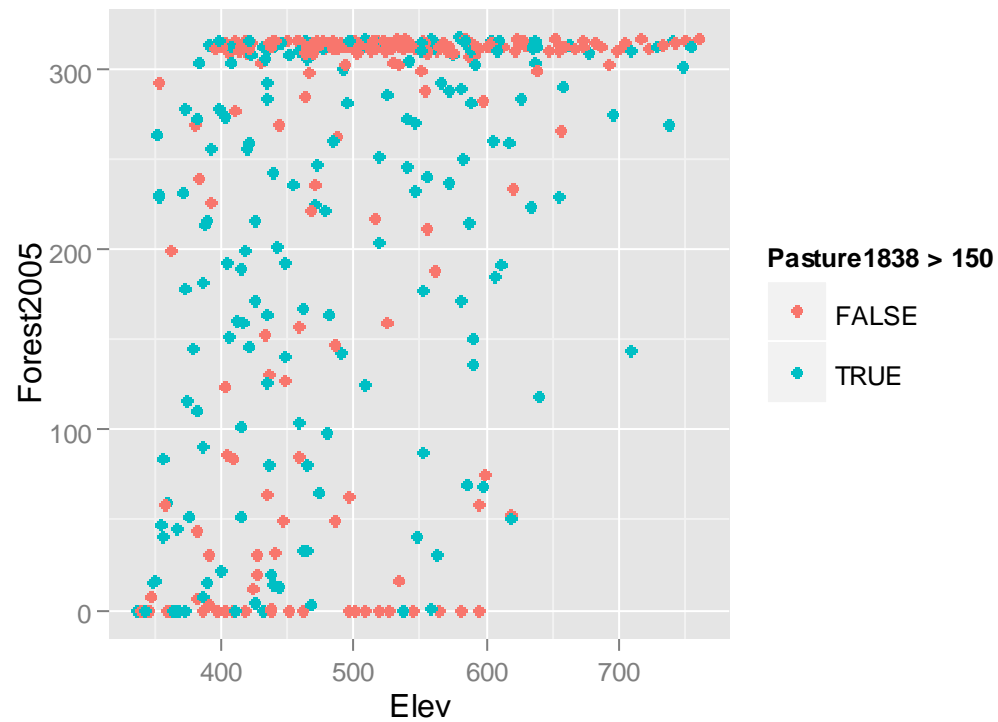
ggplot graphic

```
> qplot(x, y, data = data)
```



ggplot graphic

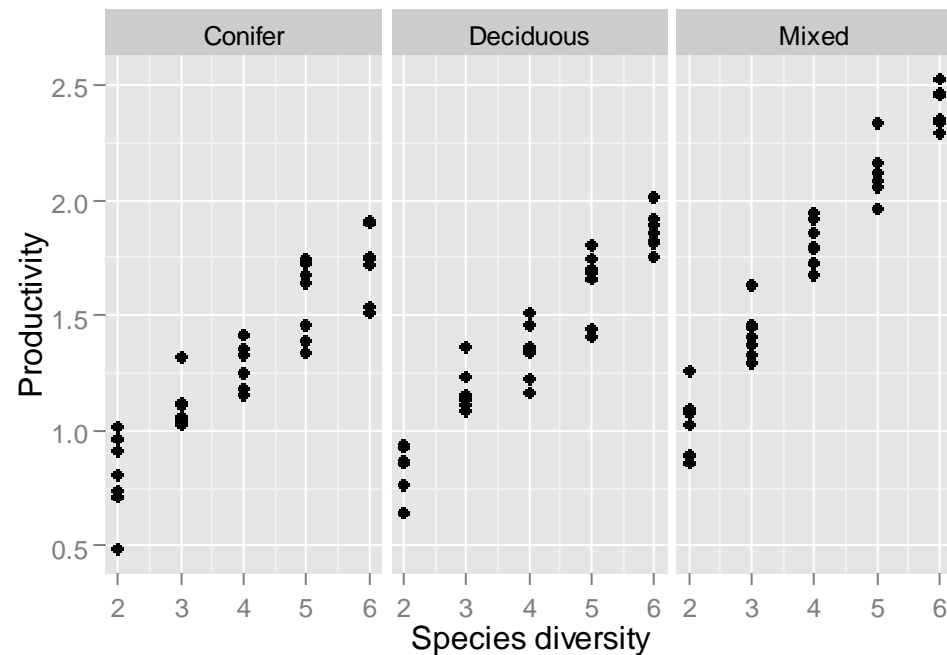
```
> qplot(x, y, data = data, colour = factor)
```



ggplot graphic – faceting

- Alternative to shape and color

```
> qplot(x, y, data = data) + facet_grid(. ~ factor)
```

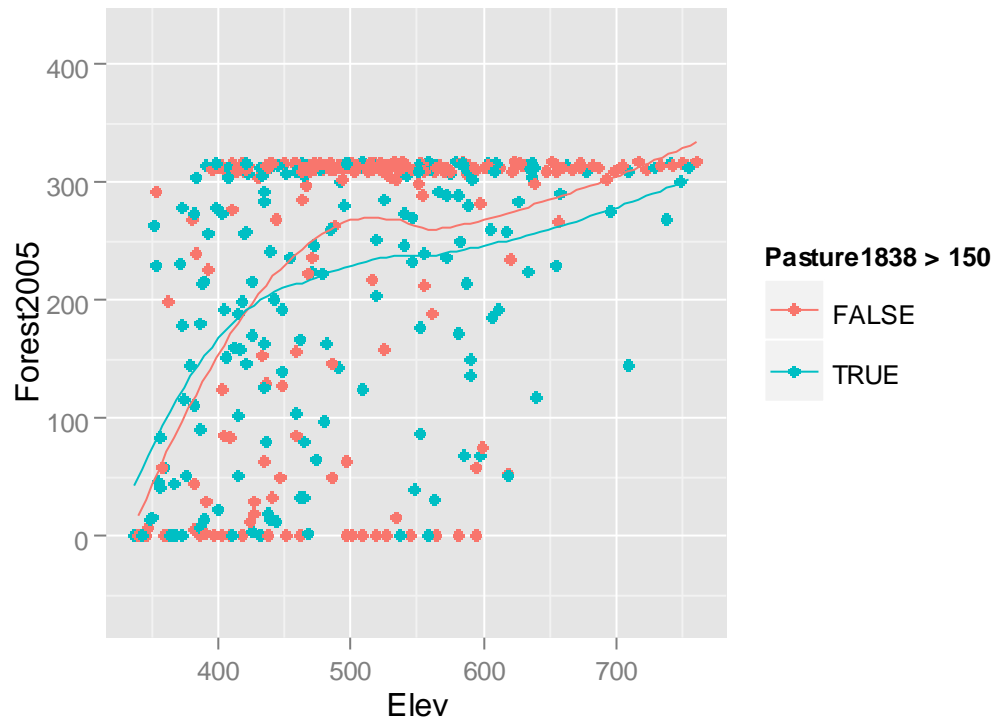


ggplot graphic

- smoothed line or fit of some model can be easily added including confidence intervals

```
> qplot(x, y, data = data, colour = factor, geom = c("point", "smooth"))
```

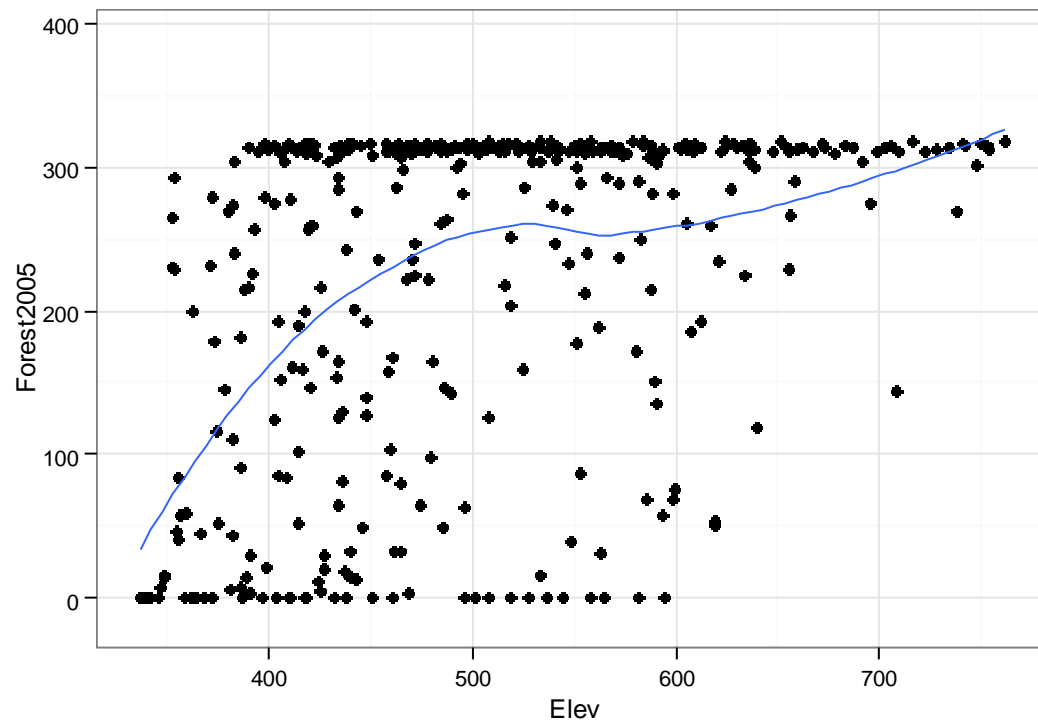
```
> qplot(x, y, data = data, colour = factor) + geom_smooth()
```



Customising ggplot graphic

- using system of themes
- special theme_bw(base_size = 12) – change theme to black and white, moreover you can specify base text size here)

```
> qplot(Elev, Forest2005, data = silver_fir_data, geom = c("point", "smooth")) +  
  theme_bw(10)
```



How to export graphs from R

- in RStudio, the easy way is to use Export option in the menu above graph – but you can't adjust parameters efficiently like size and resolution of the graph
- function `tiff()` – it changes graphical output to tif file
> `tiff("filename.tif", width = 10, height = 10, units = "cm",
pointsize = 12, compression = "lzw", res = 1000)`
... commands for graph creation
> `dev.off()`
- `pointsize` setting doesn't work with `ggplot2` graphic (or `lattice` graphic)