Introduction: Statistical Computing with R

Part3: Simple graphics

Allan Clark (adapted from B. Erni slides)

Department of Statistical Sciences, University of Cape Town

28 February 2021

Content

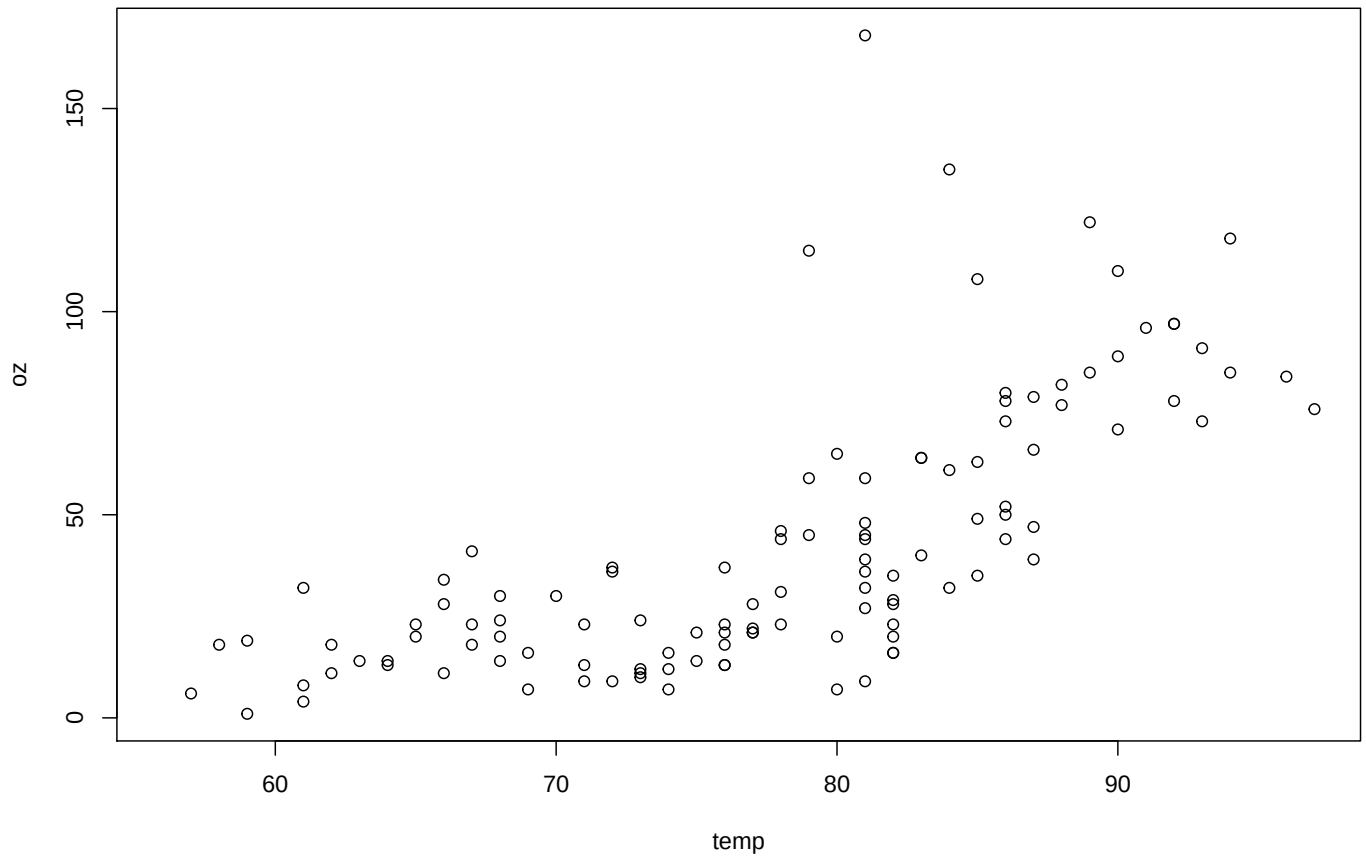
- simple graphics in R

Hans Rosling and Gapminder

- **must see TED talk:**
  https://www.gapminder.org/videos/hans-rosling-ted-2006-debunking-myths-about-the-third-world/
- **Data:** https://www.gapminder.org/data/
- **online visualization tools:**
  https://www.gapminder.org/tools/

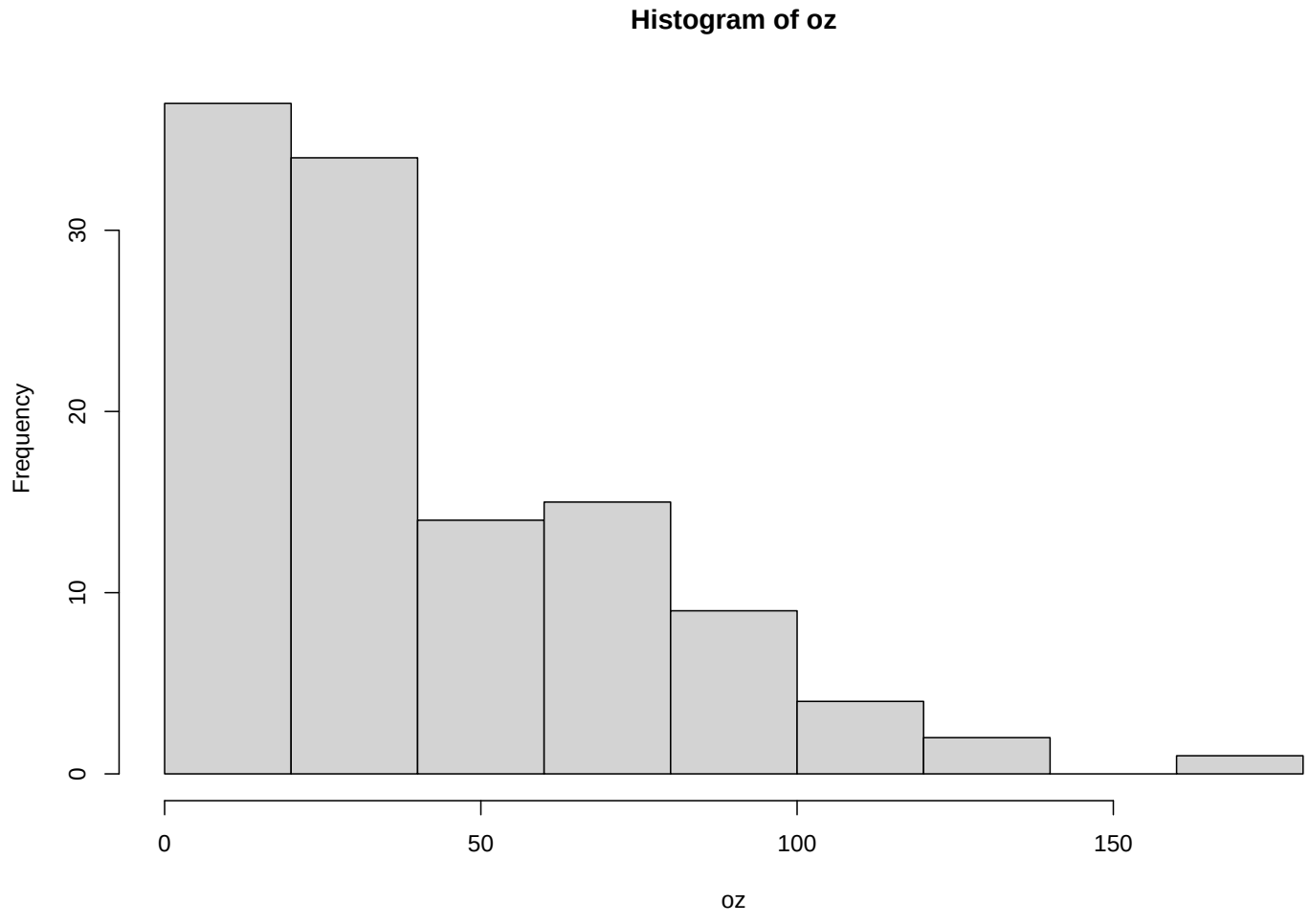
Simple graphics in R: scatter plot

```
oz <- airquality$Ozone
temp <- airquality$Temp

plot(temp, oz)
```



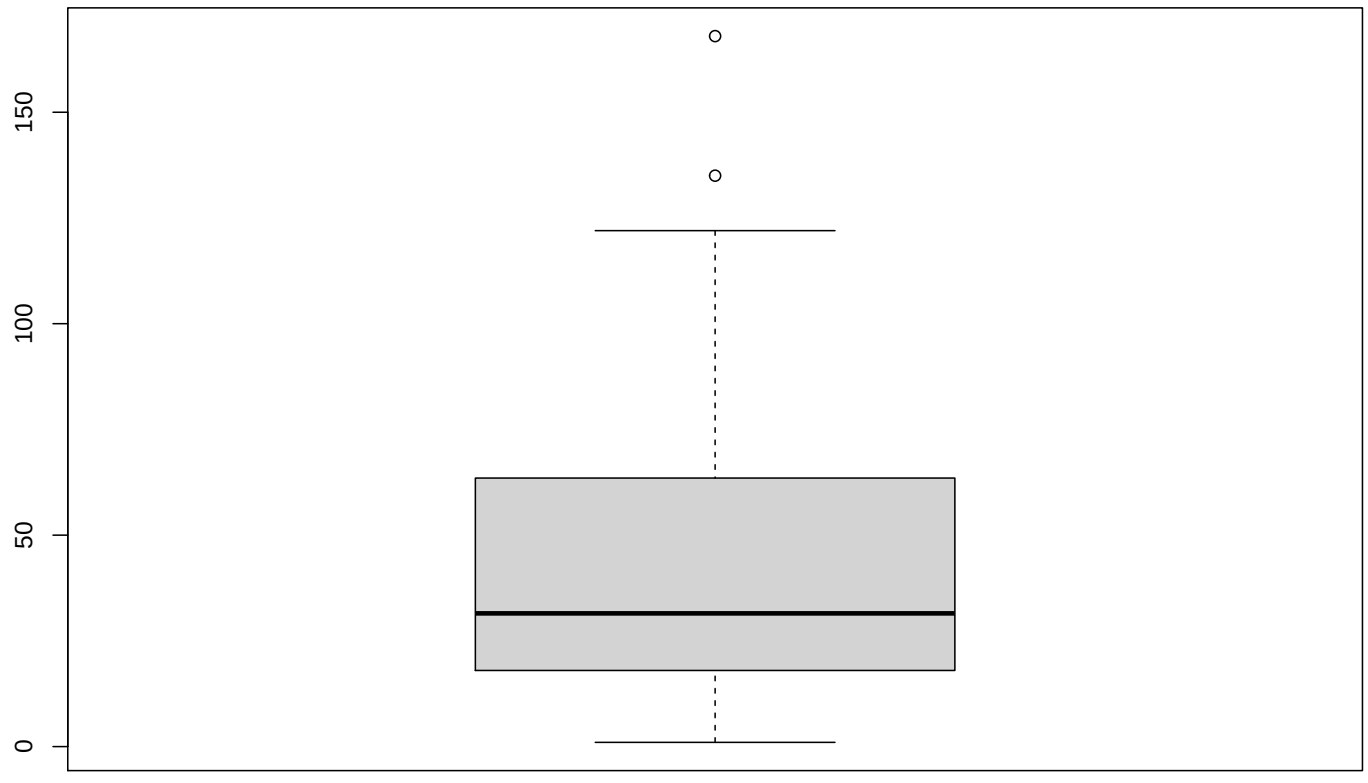
Simple graphics in R: histogram

```
hist(oz)
```



Simple graphics in R: boxplot

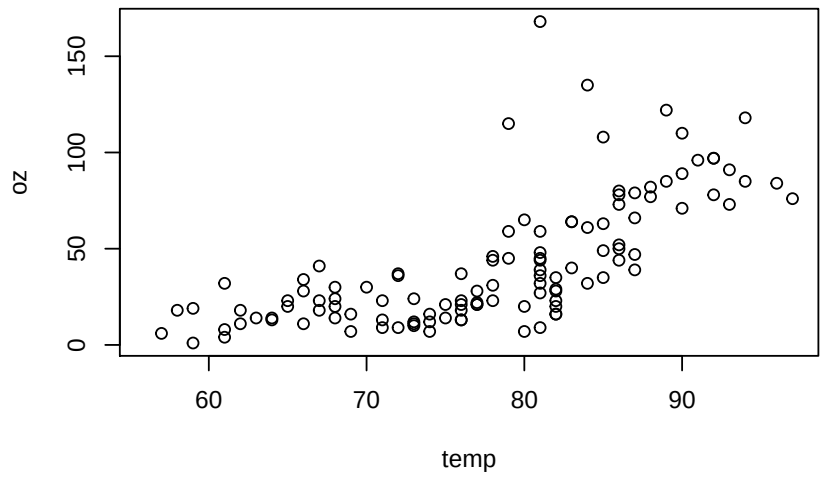
```
boxplot(oz)
```

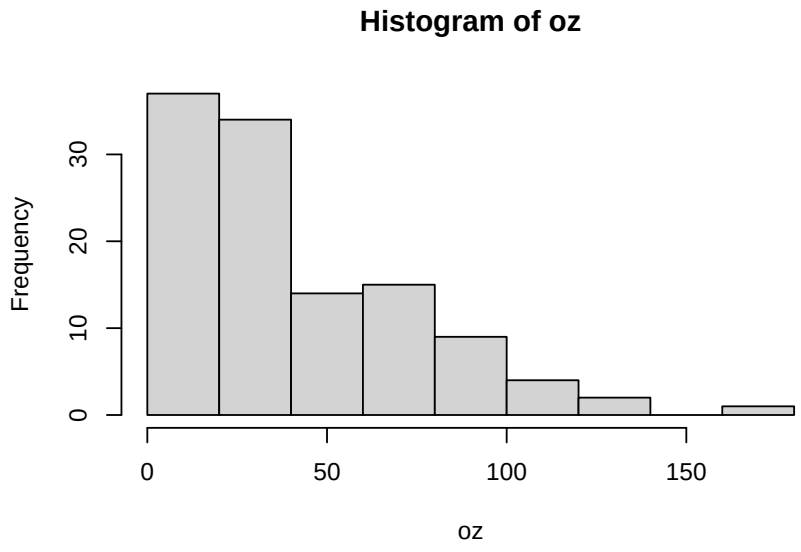


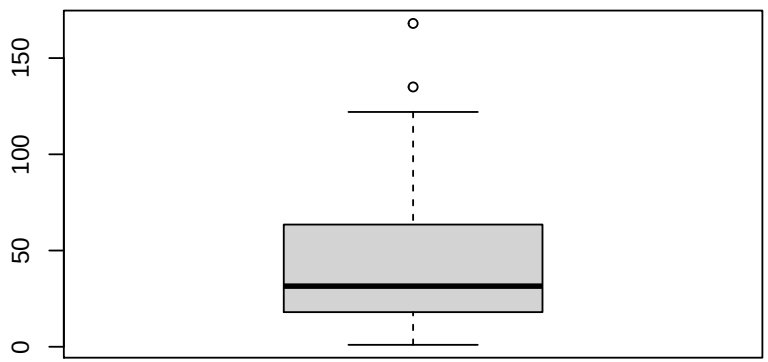
Graphics: layout

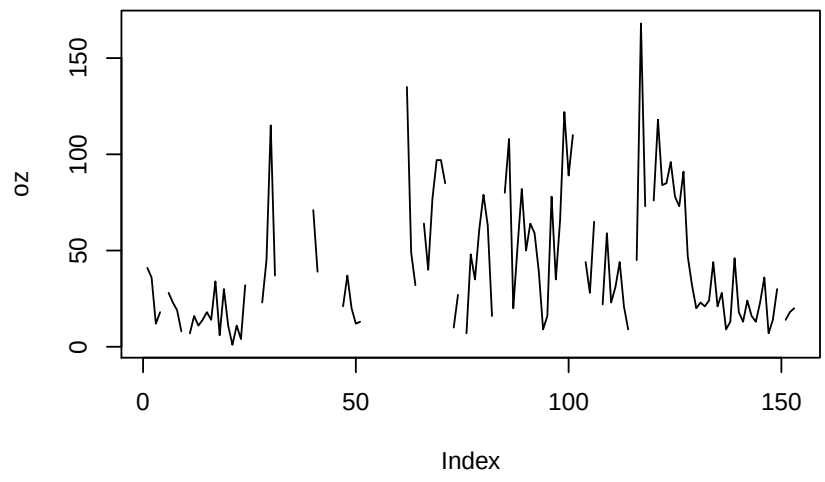
```
## split window into 2x2 matrix
par(mfrow = c(2, 2))

plot(temp, oz)
hist(oz)
boxplot(oz)
plot(oz, type = "l")    # type = line
```

Graphics: layout







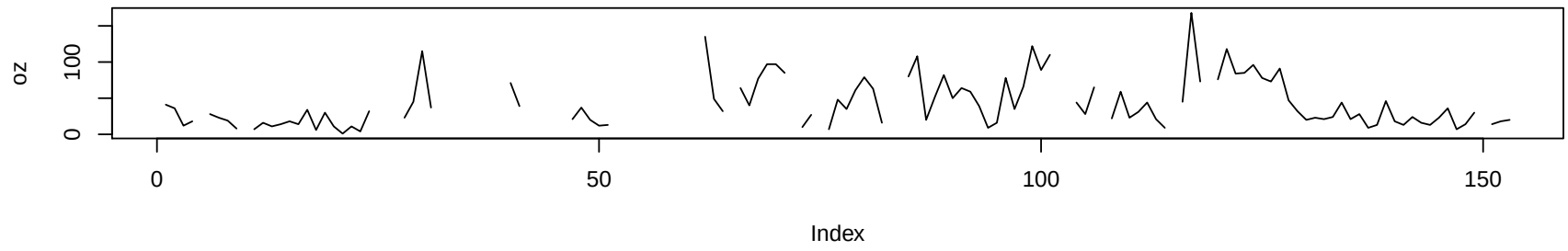


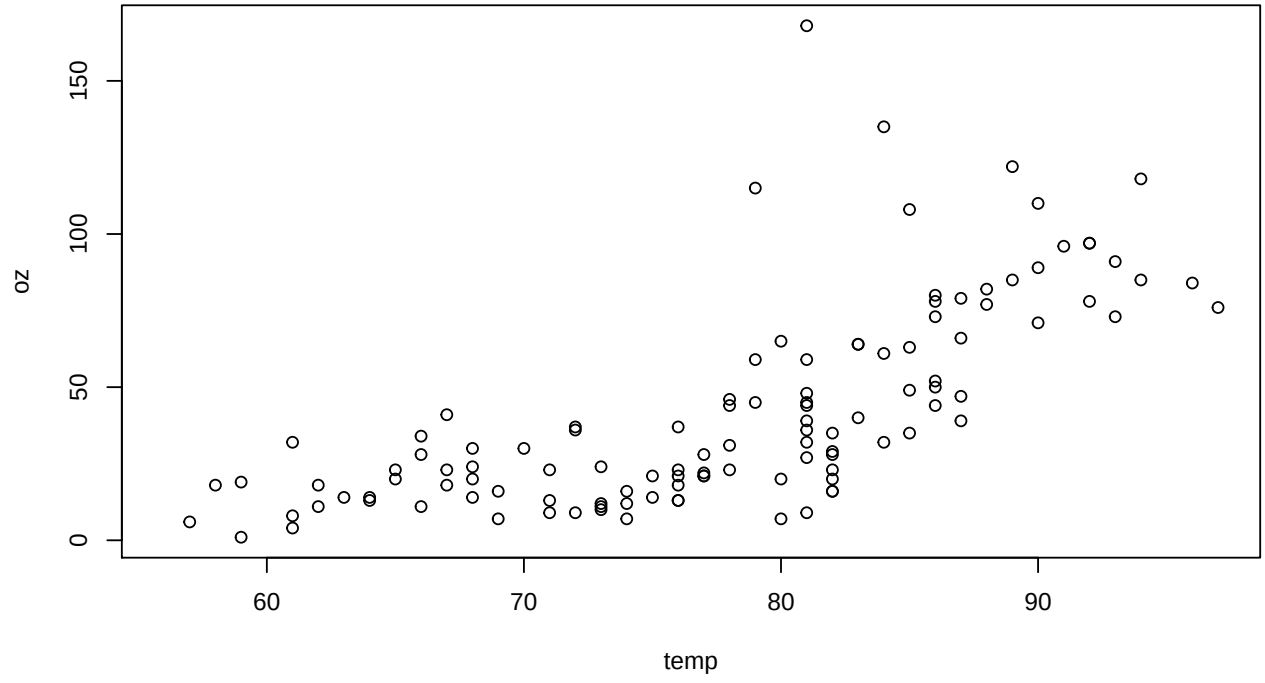
Graphics layout 2

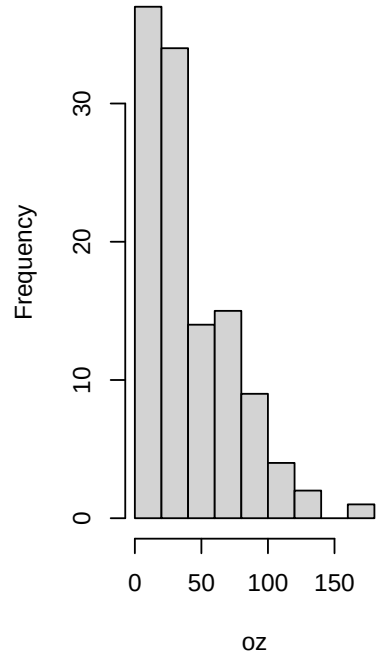
Here one can use the 'layout' function. Take note of how the width and the heights are specified. Play around with different values to observe the difference in the plot window.

```
# One figure in row 1 and two figures in row 2
# row 1 is 1/2 the height of row 2
# column 2 is 1/3 the width of the column 1
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE),
   widths = c(3, 1), heights = c(1,2))
plot(oz, type = "l")    # type = line
plot(temp, oz)
hist(oz, main = "")
```

Graphics layout 2







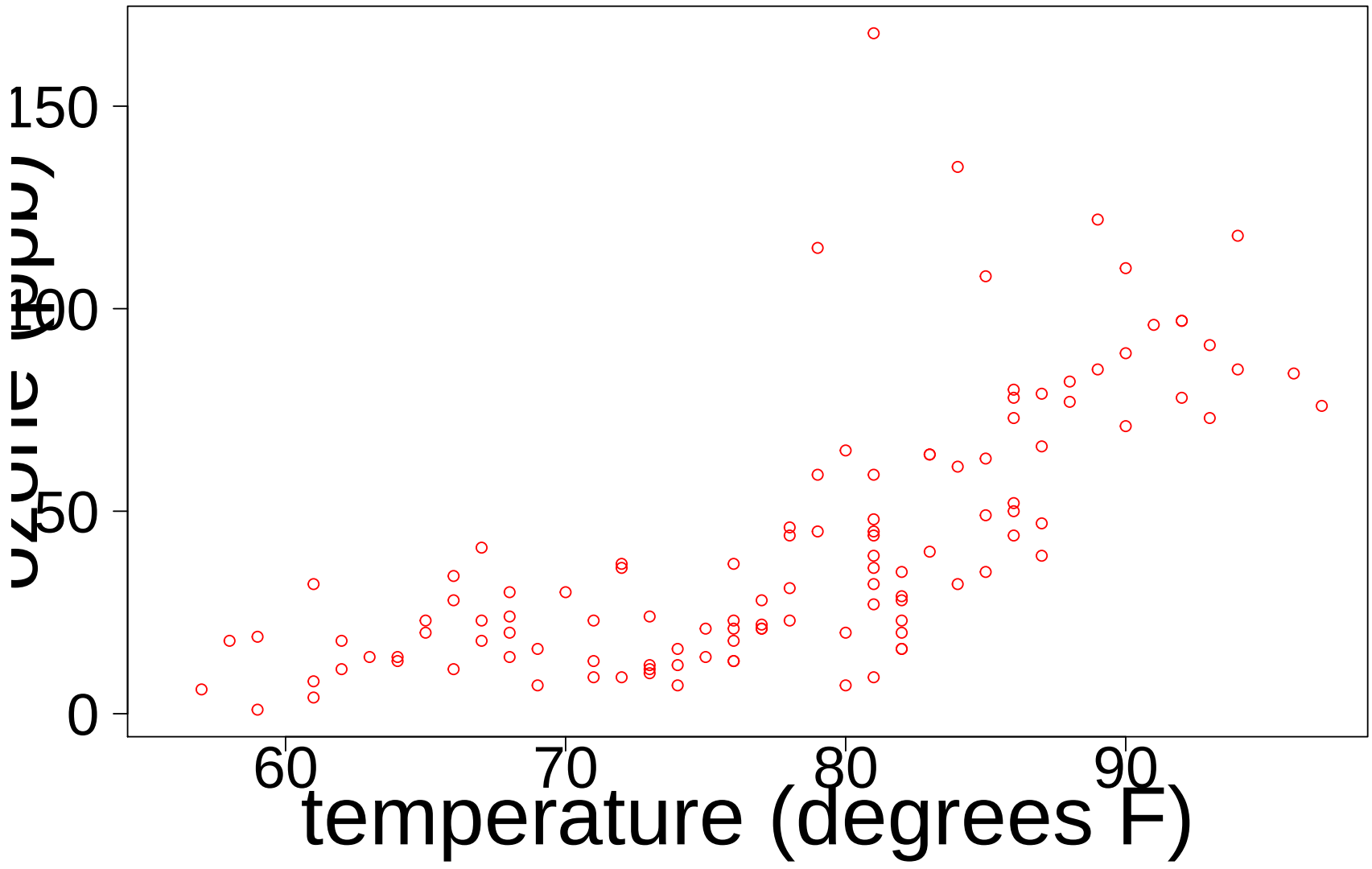
Graphics: fine-tuning

Play around to correct the plot.

```
?par

plot(temp, oz,
     pch = 21,             # plotting character
     las = 1,              # horizontal axis labels
     xlab = "temperature (degrees F)",   # x-axis label
     ylab = "ozone (ppb)",
     cex.axis = 2.5,    # scaling factor for axis values
     cex.lab = 3.5,     # scaling factor for axis labels)
     col = "red")  # colour
```

http://www.statmethods.net/advgraphs/parameters.html
ftp://cran.r-project.org/pub/R/doc/contrib/Short-refcard.pdf

Graphics: fine-tuning



More on R Markdown

Chunk options

- ```` ```{r} ```` is an unnamed chunk
- ```` ```{r chunk1} ```` is a named chunk
- ```` ```{r chunk2, echo = FALSE} ````
- `echo = TRUE`: show code in document (mostly FALSE)
- `eval = TRUE`: run code
- `fig.show = "hide"`: don't print figure
- `fig.cap = "Histogram of ozone values."`
- `fig.width = 4`: in inches

Figures with LaTeX (pdf), html

- **Option 1:** fig.show = "hide", then in text:

LaTeX:
`\includegraphics[width=\textwidth]{Figs/plots3-1.pdf}`

html: `![](Figs/plots3-1.png)`

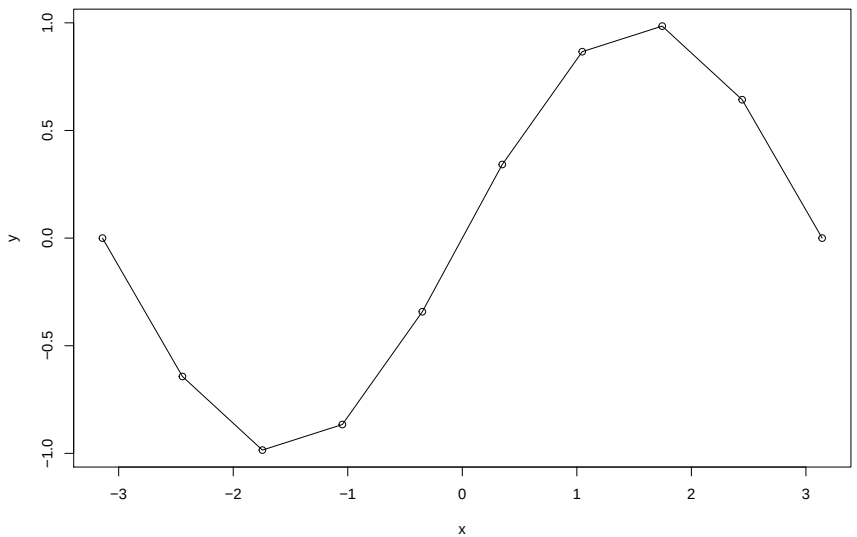
- **Option 2:**

In code chunk: `include_graphics(Figs/plots3-1.png)` (needs library knitr, but works with any type of file)

Plot a function

$$f(x) = sin(x) \qquad -\pi \leq x \leq \pi$$

```
## vector of 10 values between -pi and pi
x <- seq(-pi, pi, length = 10)
y <- sin(x)      # sin(x) for each x value
plot(y ~ x)
lines(x, y)    # adds a line connecting the points
```



Note: x's are sorted!

Prac 4

Create a smooth version of the above sin function: increase the number of x-values at which f(x) is evaluated, e.g to 1000. Your final plot should:

- Plot the line directly, without points first. Increase line width, and change colour to red.
- Add the $cos(x)$ line to the same plot, in blue.
- Improve the general look of the figure.
- What would happen if the x's were a random uniform sample between $-\pi$ and π? Try.

Saving graphics

You can save directly to pdf, svg, . . .

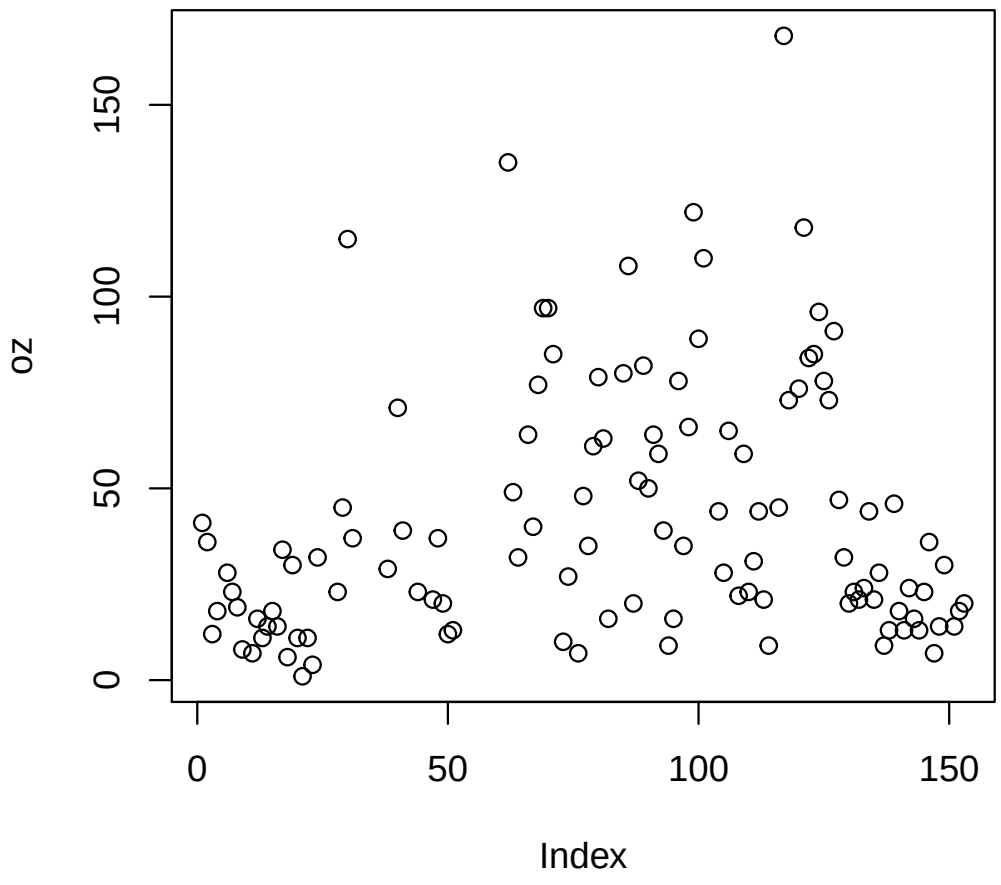
```
#this creates a pdf file and saves it to your working direc
pdf("ozone.pdf", height = 5, width = 5)
  plot(oz)
dev.off()
```

```
## pdf
##   2
```

- Microsoft Word: . . . save as .svg
- Powerpoint: Copy-paste
- LaTeX: pdf works well.

Saving graphics

What's wrong with this figure?



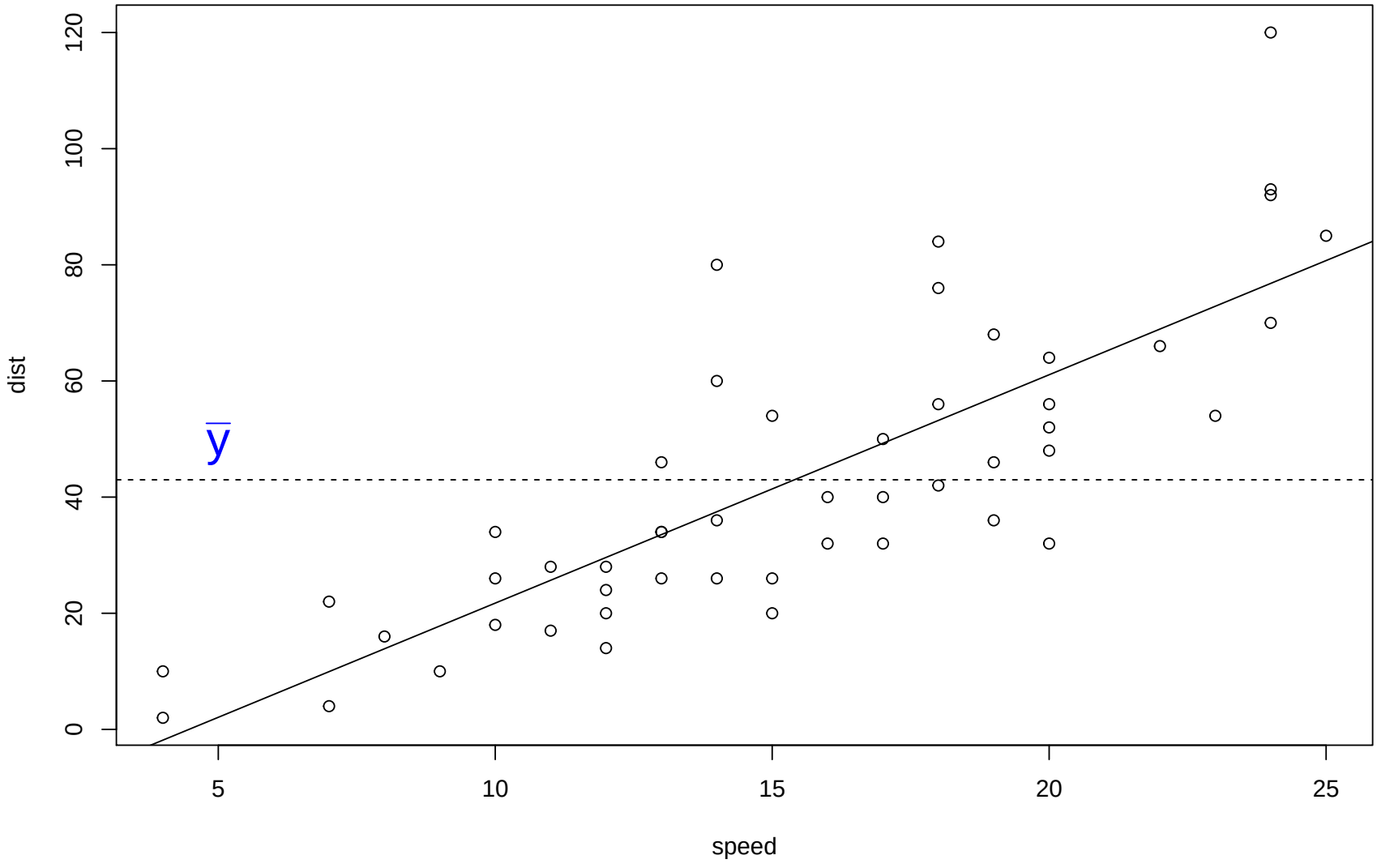
Adding to plots

lines(), abline(), points()

```
m1 <- lm(dist ~ speed, data = cars)
plot(dist ~ speed, data = cars)

abline(m1)          # regression line

ybar <- mean(cars$dist)
abline(h = ybar,    # horizontal line
       lty = 2)     # line type (dashed)

# x, y, text, position, colour, zoom factor
text(5, ybar, expression(bar(y)), pos = 3, col = "blue",
     cex = 2)
```

Adding to plots

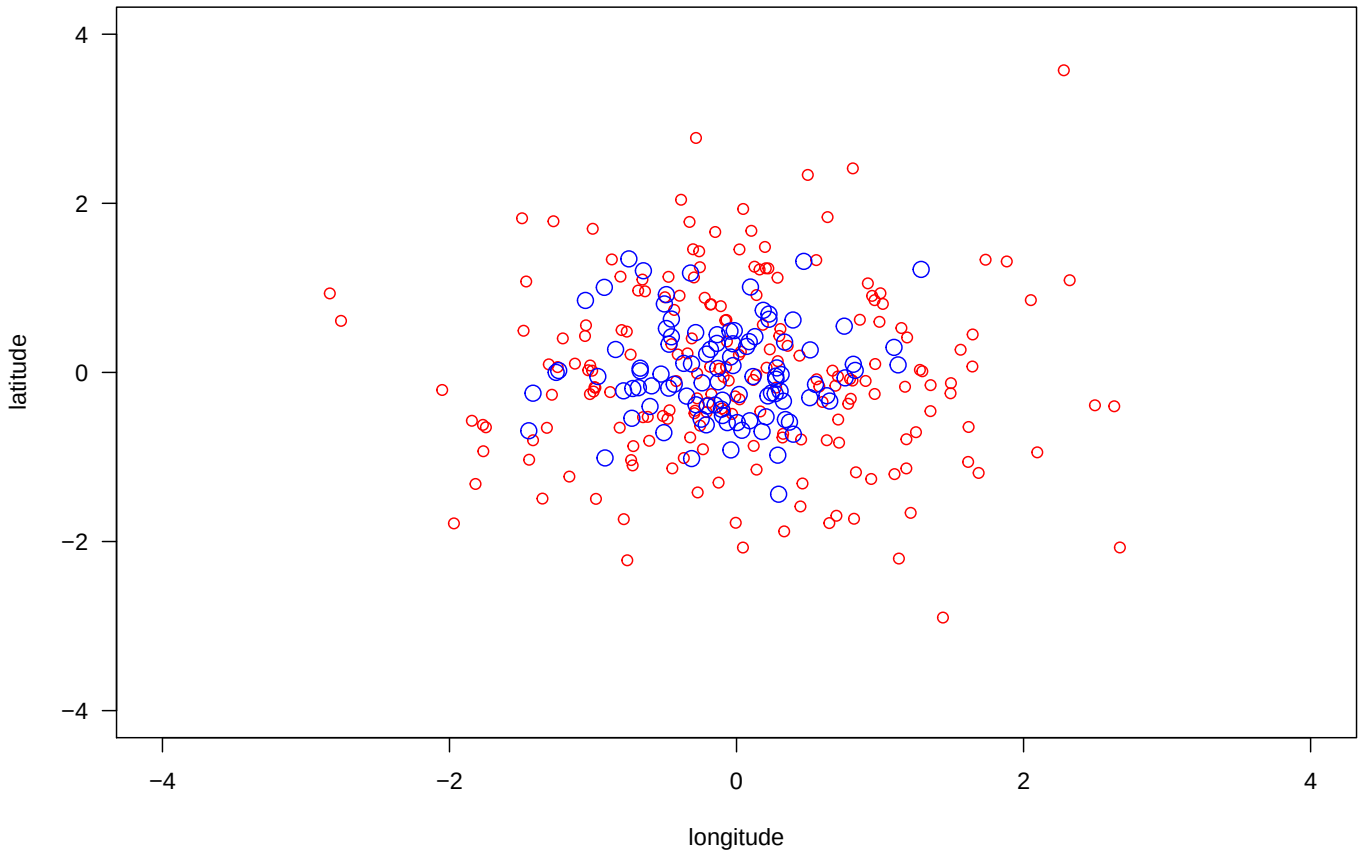
lines(), abline(), points()



Add Points

```
# setting up coord. system
plot(-4:4, -4:4, type = "n",
     xlab = "longitude", ylab = "latitude", las = 1)

points(rnorm(200), rnorm(200), col = "red")     # x, y

points(rnorm(100)/2, rnorm(100)/2,
       col = "blue", cex = 1.5)
```

Add Points



Prac 5

1. Generate 1000 values from a $N(1, 2)$ distribution and summarise these in a histogram. Change the histogram so that density and not frequency is shown on the y-axis. Calculate the true $N(1, 2)$ density over the domain of $N(1, 2)$ and plot this on top of the histogram using a thick red line.
2. Save different formats of this (see help for png `?png`). Copy all of these into a word document and comment on the differences. Which clearly works best, even when you really zoom in?

More about Factors

```
library(gapminder)

conti <- gapminder$continent

str(conti)
```

```
##  Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3
```

```
typeof(conti)    # factors are weird!
```

```
## [1] "integer"
```

```
class(conti)
```

```
## [1] "factor"
```

More about Factors

```
levels(conti)[1:4]
```

```
## [1] "Africa"   "Americas" "Asia"     "Europe"
```

```
nlevels(conti)  # number of levels
```

```
## [1] 5
```

```
head(conti)
```

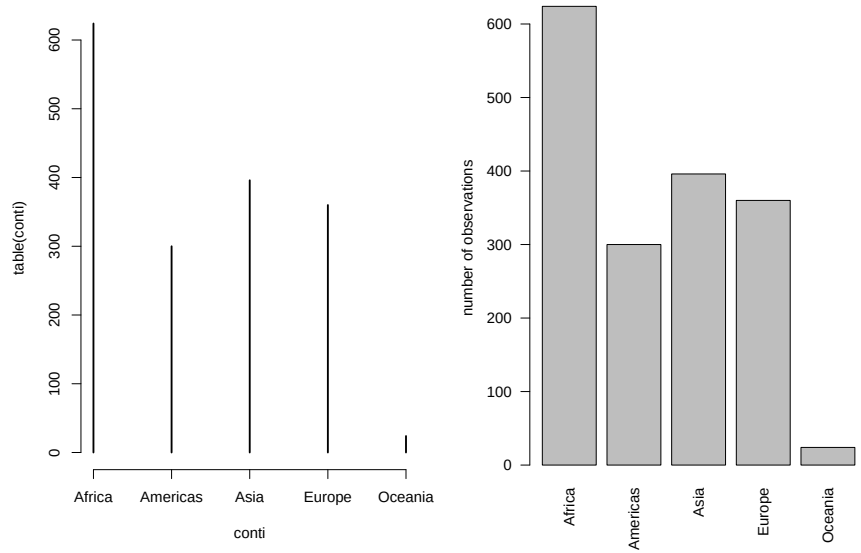
```
## [1] Asia Asia Asia Asia Asia Asia
## Levels: Africa Americas Asia Europe Oceania
```

```
table(conti)          # frequency table
```

```
## conti
##   Africa Americas     Asia   Europe  Oceania
##      624      300      396      360       24
```

More about Factors

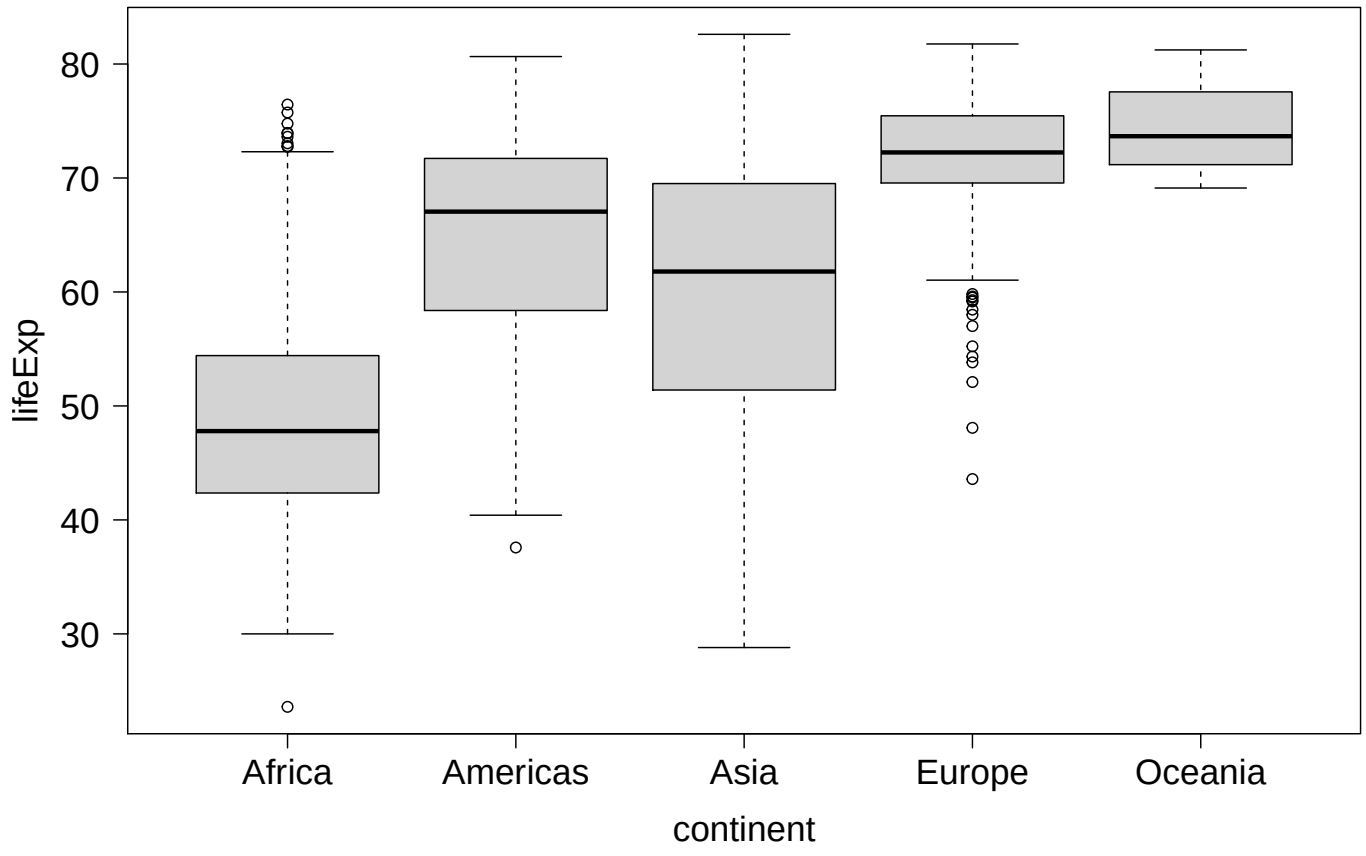
```
par(mfrow = c(1, 2))
plot(table(conti))
barplot(table(conti), las = 2,
        ylab = "number of observations")
```



Barplots often have a very low information/ink ratio! Nothing that one can't learn from the table.

Factors when visualizing data

```
plot(lifeExp ~ continent, data = gapminder,
     cex.axis = 1.5, cex.lab = 1.5, las = 1)
```



Prac 6: Produce a beautiful plot in a document

- Gapminder: Plot life expectancy against GDP, don't attach, colour by continent.
- Does a log-transformation help to bring out the information more clearly?
- R Markdown document (html, word, pdf)
- Figure caption
- Axis labels and sizing. Improve visually. Legend.
- Improve size and placement.
- How many different countries occur in this data set?
- How many African countries?
- Which countries have the lowest and highest life expectancy, respectively? In which years? (There are several observations/years per country).

Advanced: ggplot2

Not for this course, but if you have extra time.

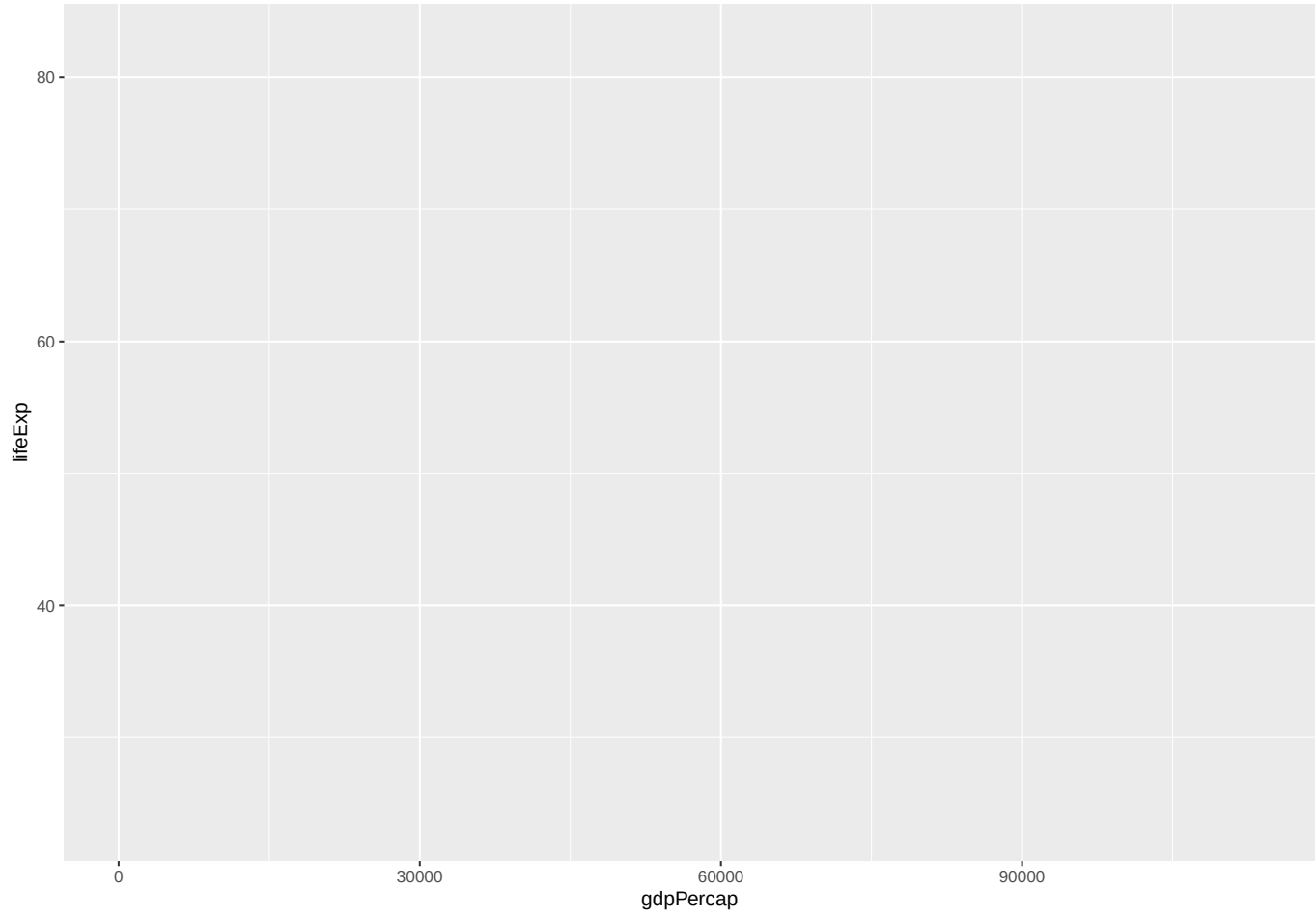
- http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html
- http://stats.idre.ucla.edu/r/seminars/ggplot2_intro/
- *Data Visualization with ggplot2* cheatsheet under RStudio Help

ggplot2

```
library(ggplot2)
library(gapminder)
```

```
#declare data and x and y aesthetics, but no shapes yet
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp))

## add layers
## it already knows what the x and y variables are from
## the ggplot part

ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) +
  scale_x_log10() +   # log x axis
  geom_point(aes(color = continent)) + # colour by continent
  geom_smooth()
```

ggplot2



ggplot2

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s
```

ggplot2

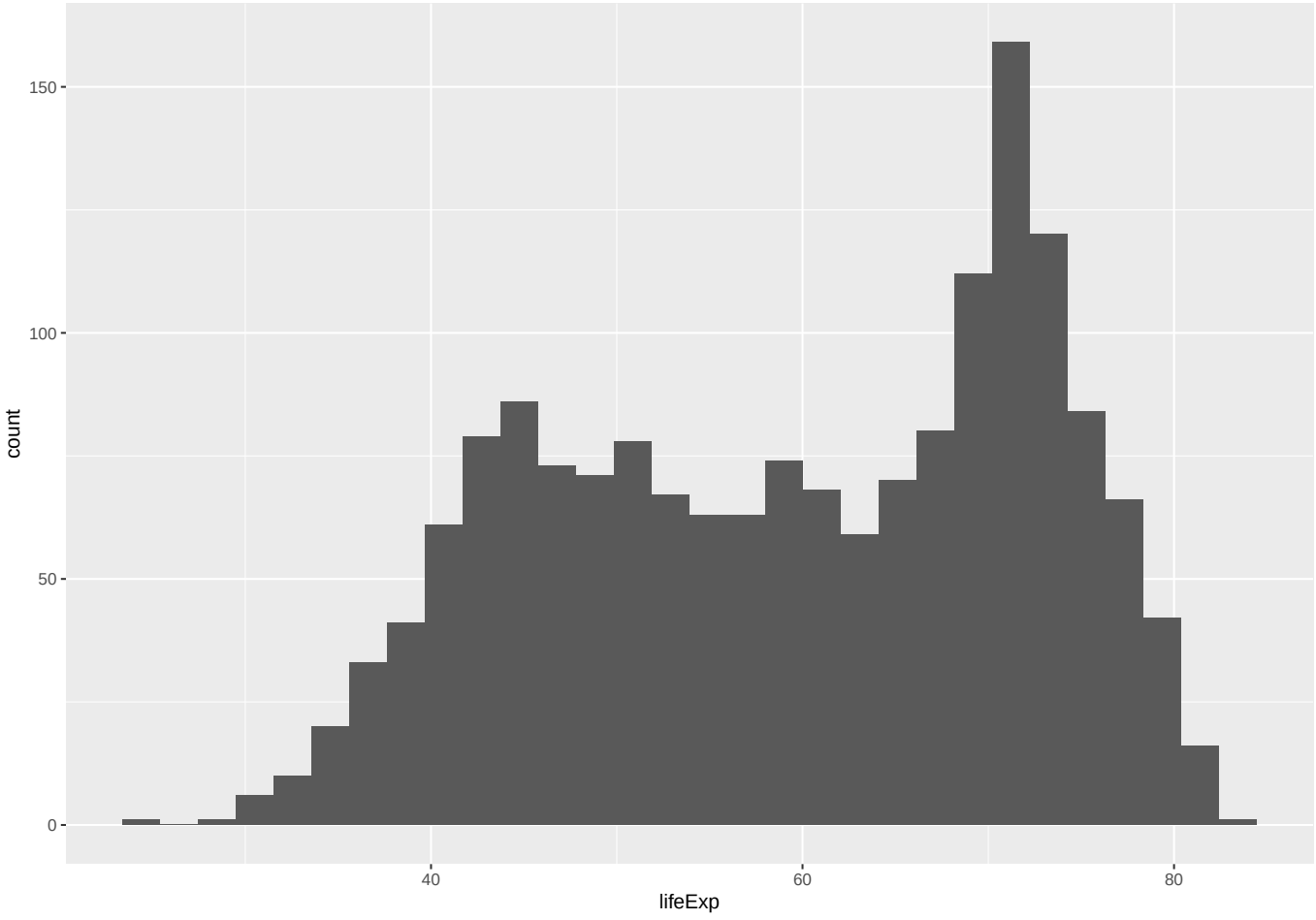
```
## define data and x for univariate plots
f1 <- ggplot(gapminder, aes(x = lifeExp))
```

ggplot output is an object. So one can add to it.

```
## histogram
f1 + geom_histogram()
```

ggplot2

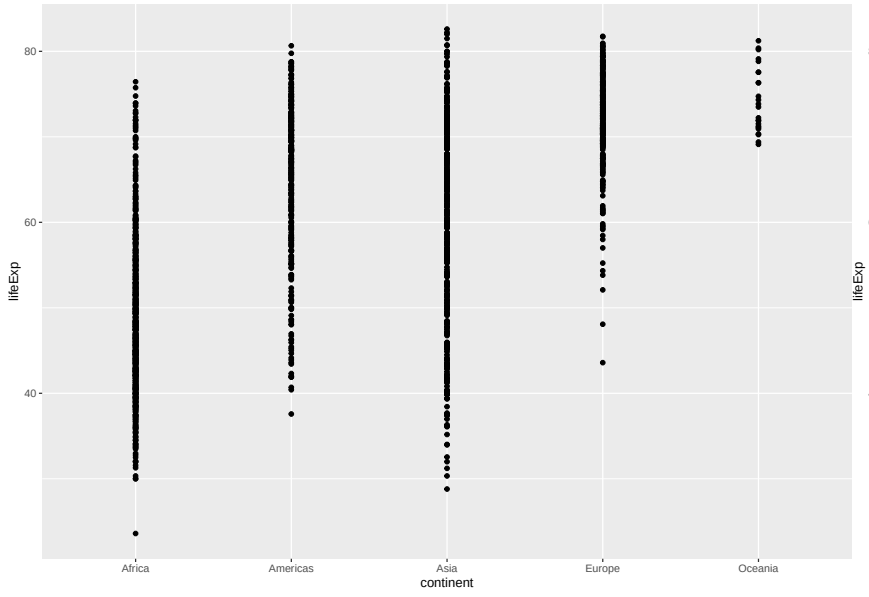
```
## `stat_bin()` using `bins = 30`. Pick better value with `
```

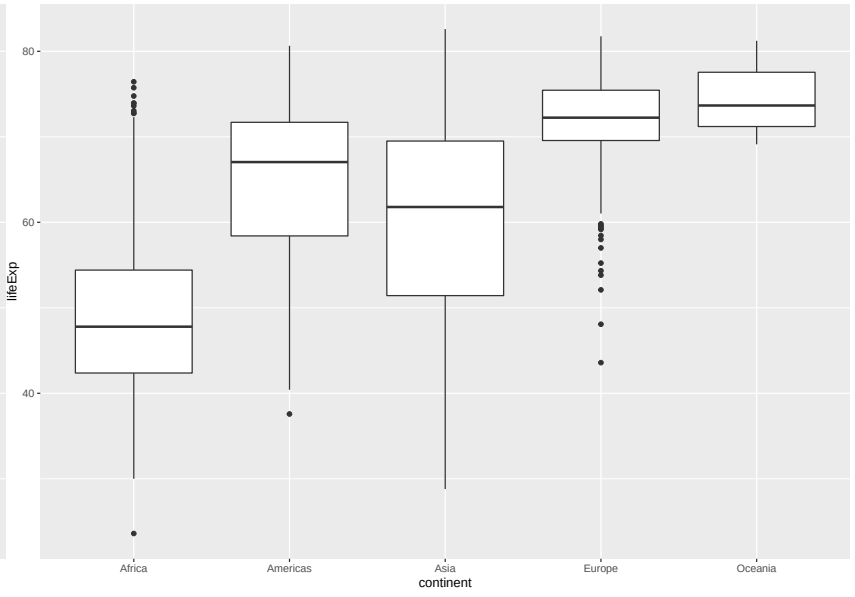


ggplot2

```
## define data, x and y, x is a factor
tp <- ggplot(gapminder, aes(x = continent, y = lifeExp))

## scatter plot and boxplot by continent
tp + geom_point()
tp + geom_boxplot(aes(group = continent))
```

ggplot2


80
60
40
lifeExp
Africa
Americas
Asia
Europe
Oceania
continent


80
60
40
lifeExp
Africa
Americas
Asia
Europe
Oceania
continent