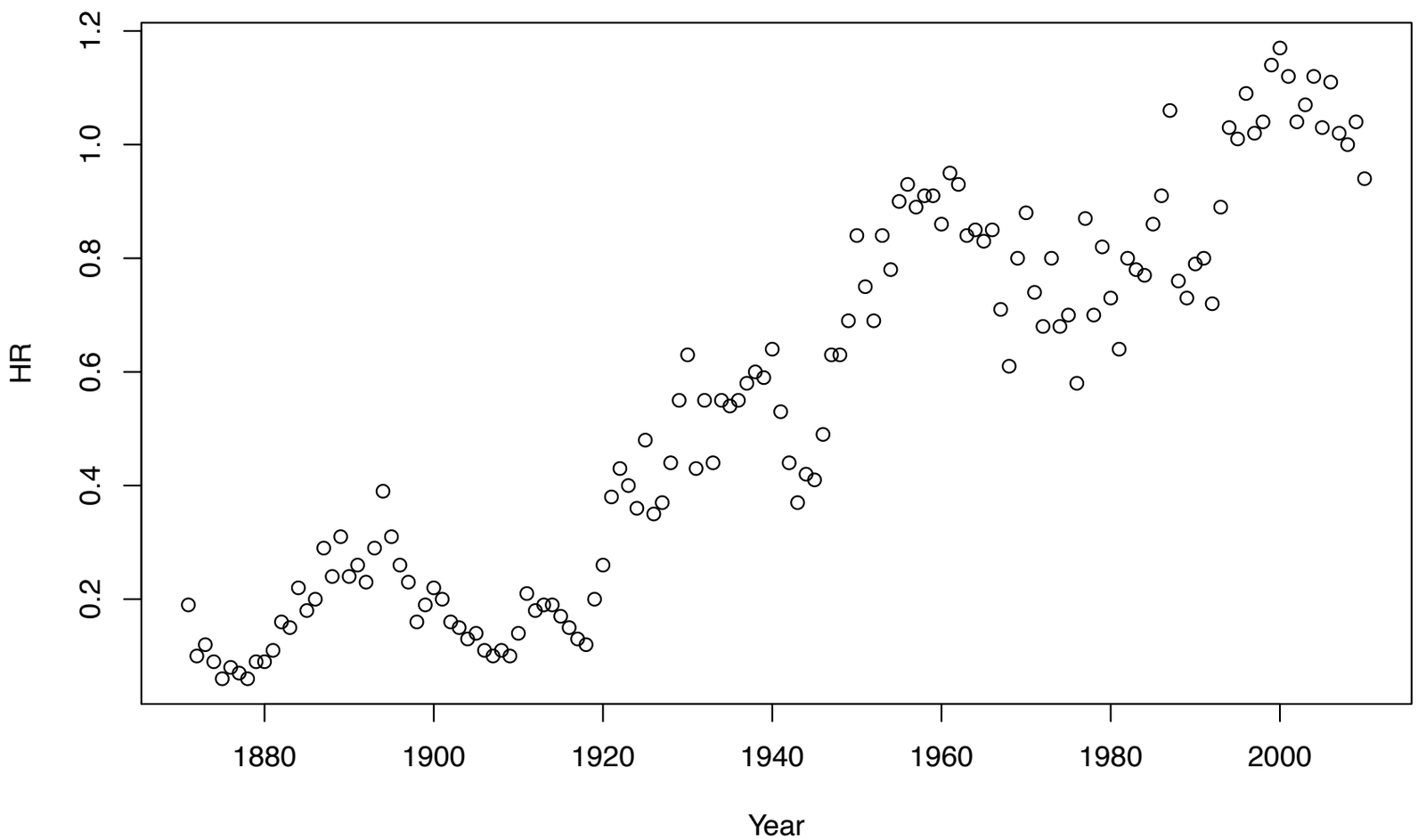
Adjusting Attributes of a Graph

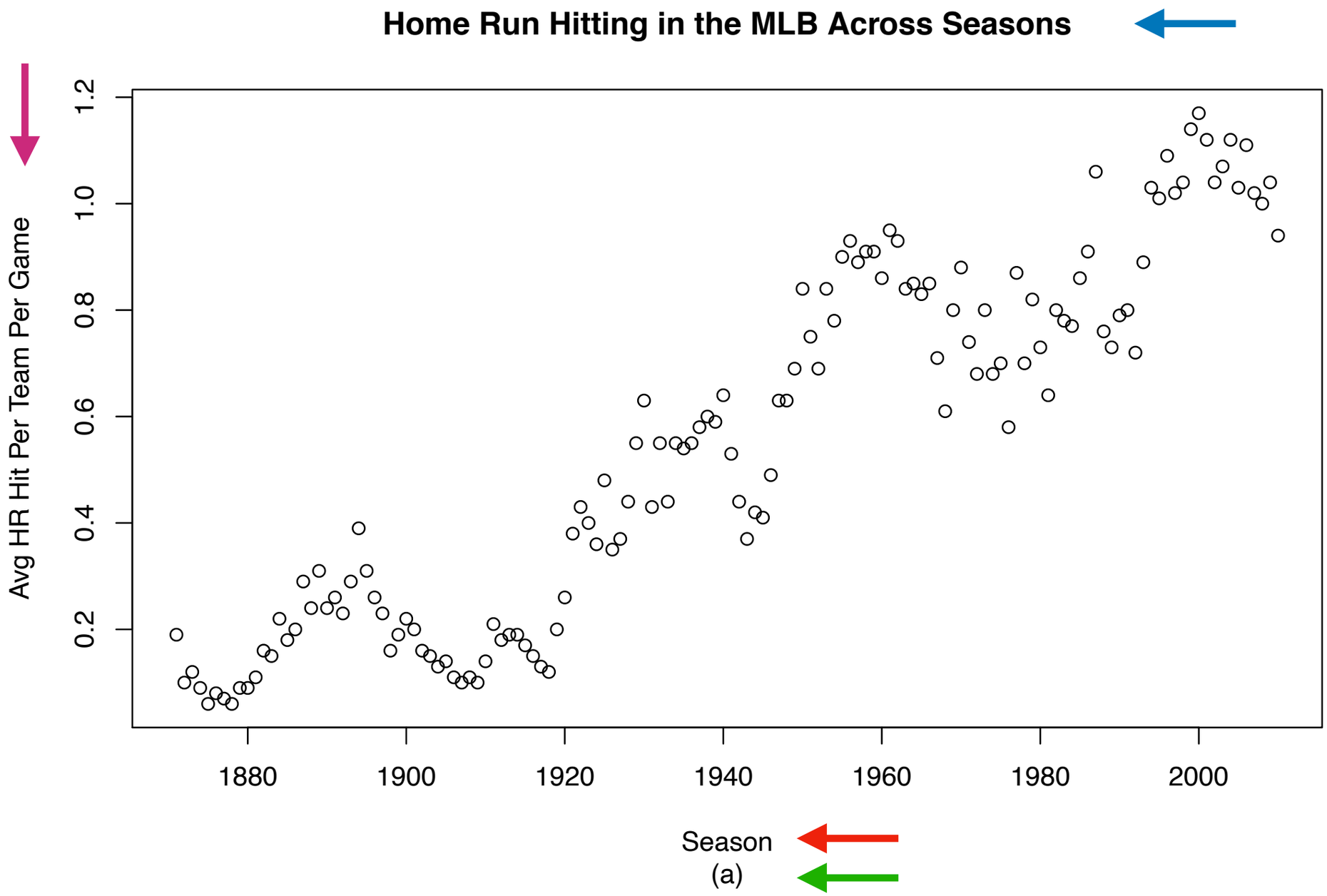
- Parameters in the function `plot`
  - Labeling axes and adding titles/subtitles:

```
> hitting.data = read.table(“batting.history.txt”,
header=TRUE,sep="\t")
> attach(hitting.data)
> plot(Year, HR)
```



Graphs

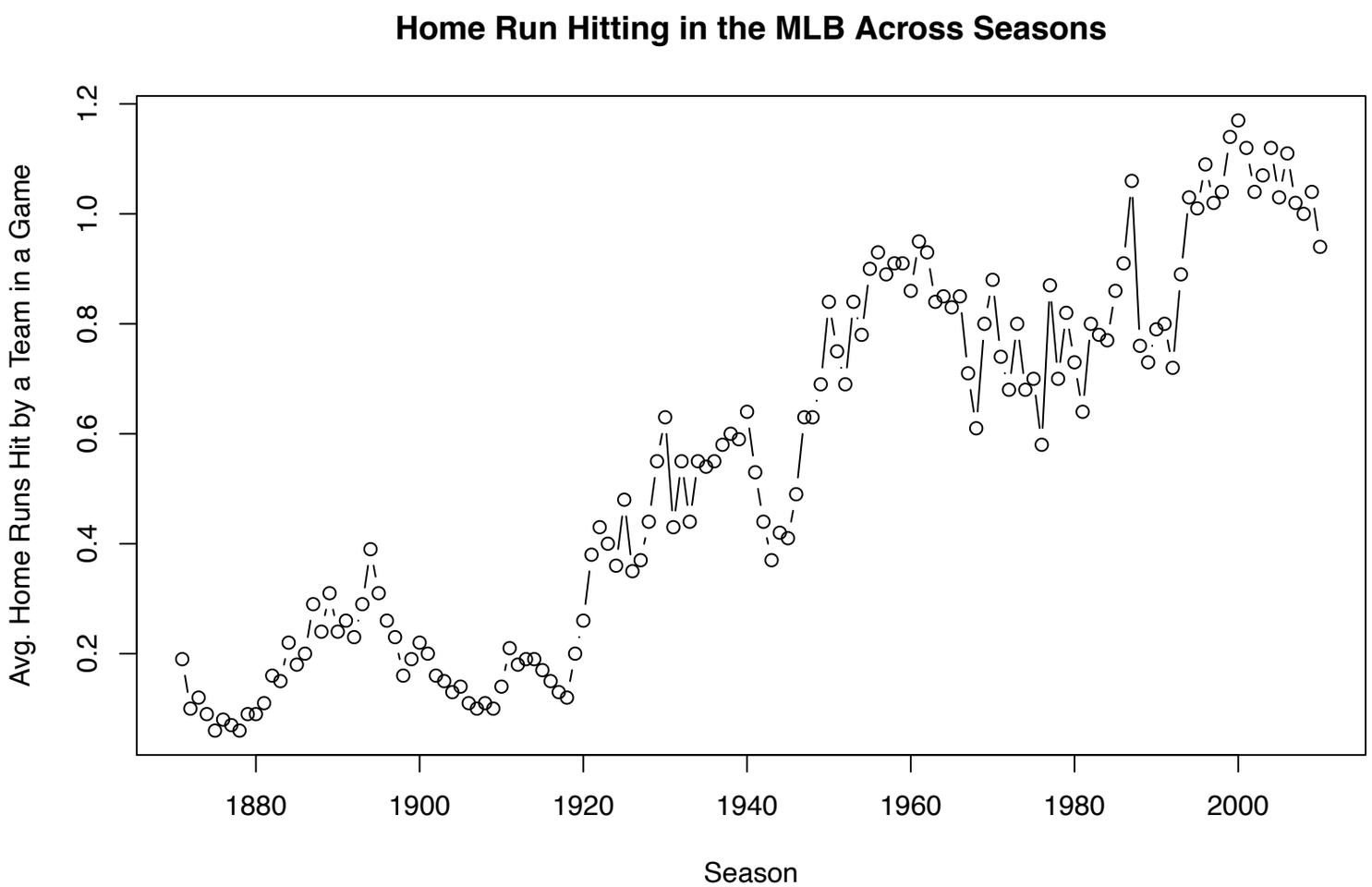
```
> plot(Year, HR, xlab="Season", ylab="Avg HR Hit Per Team Per Game",
+      main="Home Run Hitting in the MLB Across Seasons", sub="(a)")
```



Graphs

- Changing plotting type and plotting symbols:

```
> plot(Year, HR, xlab="Season", type="b",
+      ylab="Avg. Home Runs Hit by a Team in a Game",
+      main="Home Run Hitting in the MLB Across Seasons")
```



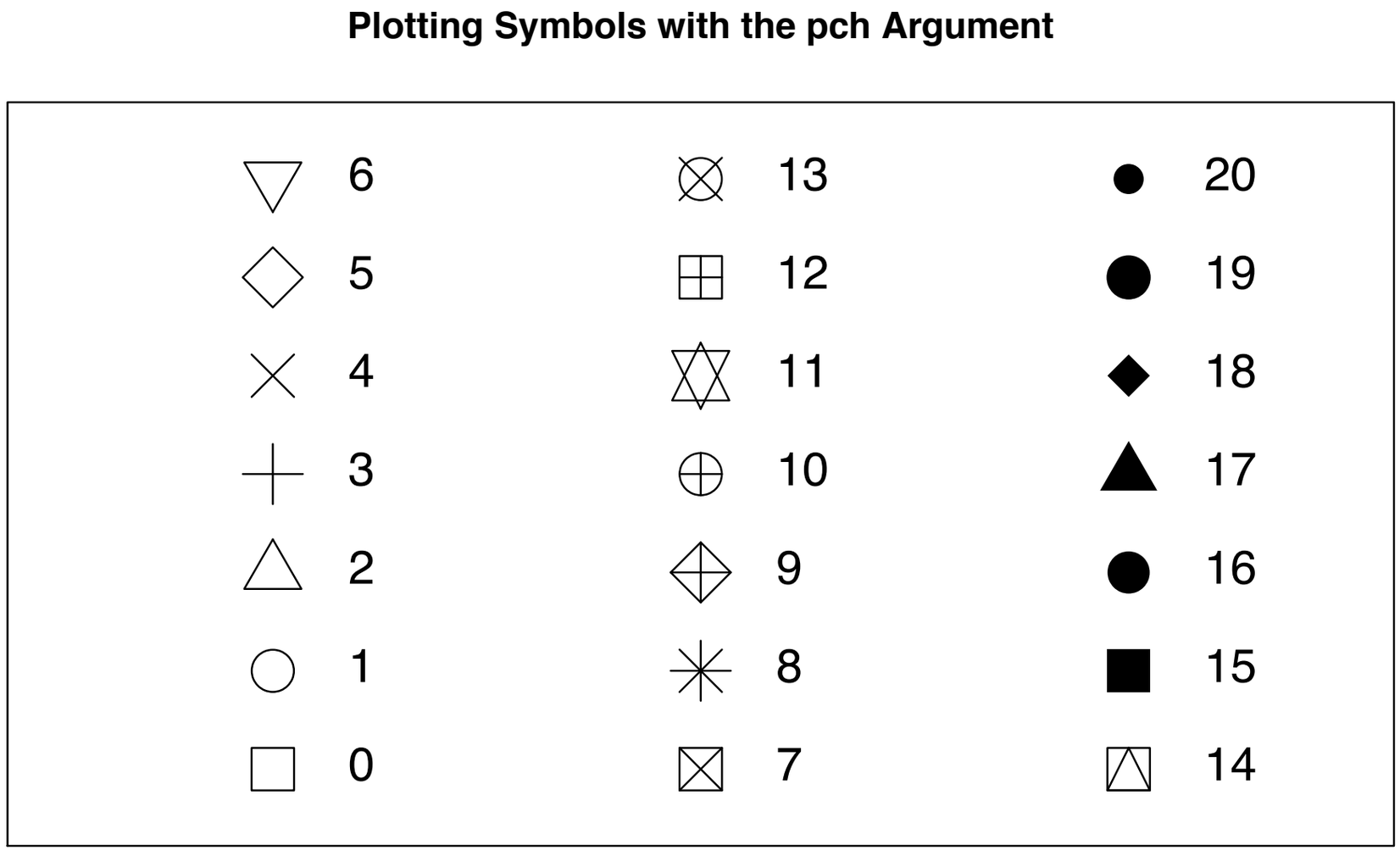
- `type="l"`: lines
- `type="b"`: points and lines
- `type="n"`: no points at all

Graphs

- `xlim` and `ylim` control the horizontal and vertical domains
- `xaxt="n"` and `yaxt="n"` or `axes=F` indicate that no axes will be drawn
- the `points` function is used to add symbols etc to current plot
- the `pch` argument indicates the plotting symbols
- `cex` magnifies/decreases size of the symbols
- `text` adds text to current graph
- `pos` controls the position of the symbols
- `offset` controls the number of spaces in the text
- `title` adds a title to the current plot
- `lines` adds lines to the current plot
- `lwd` controls the width of the line

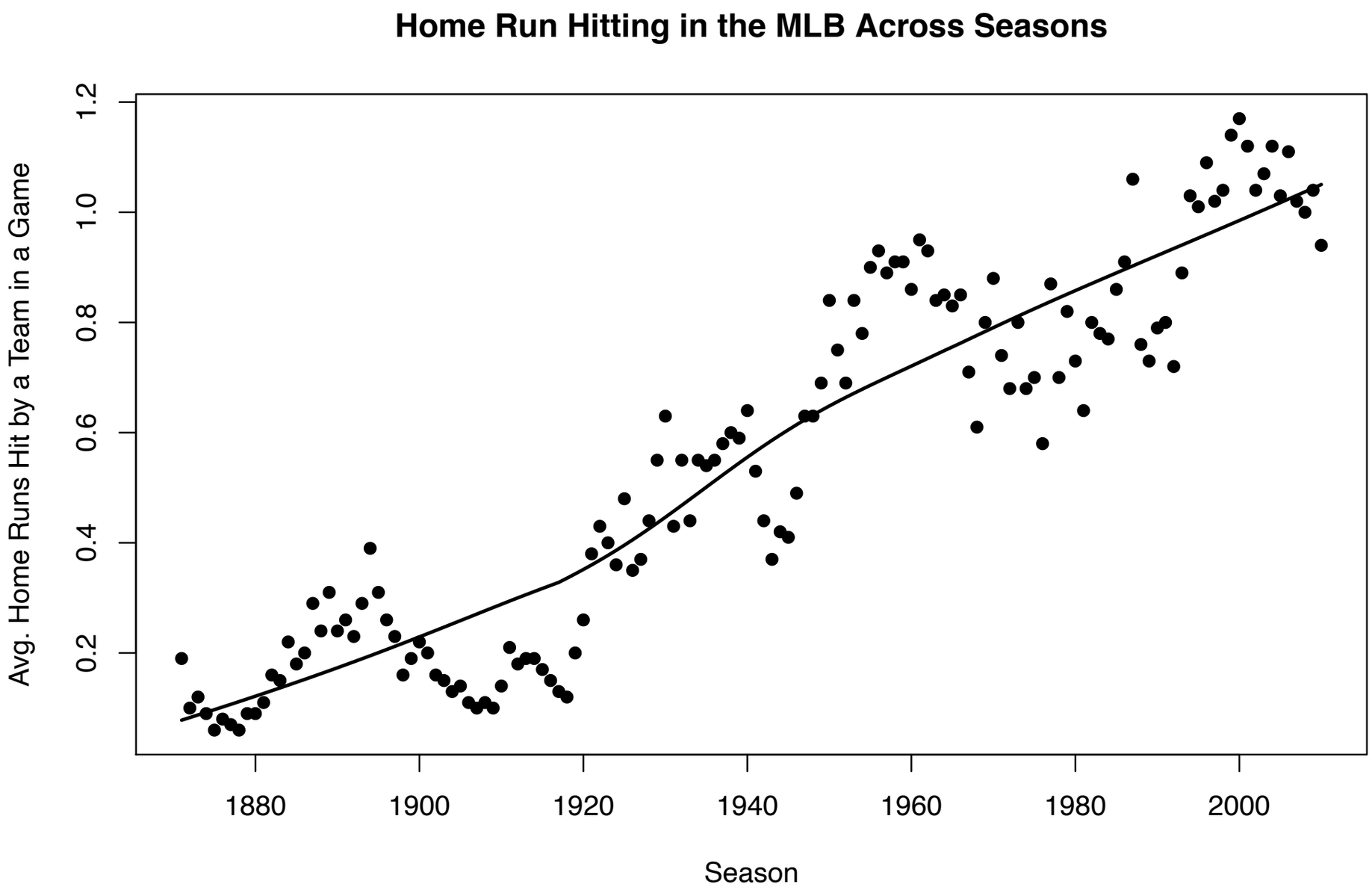
Graphs

```
> row = rep(1:3, each=7)
> col = rep(1:7, times=3)
> plot(2, 3, xlim=c(.5,3.5), ylim=c(.5,7.5),
+      type="n", xaxt = "n", yaxt = "n", xlab="", ylab="")
> points(row, col, pch=0:20, cex=3)
> text(row, col, 0:20, pos=4, offset=2, cex=1.5)
> title("Plotting Symbols with the pch Argument")
```



Graphs

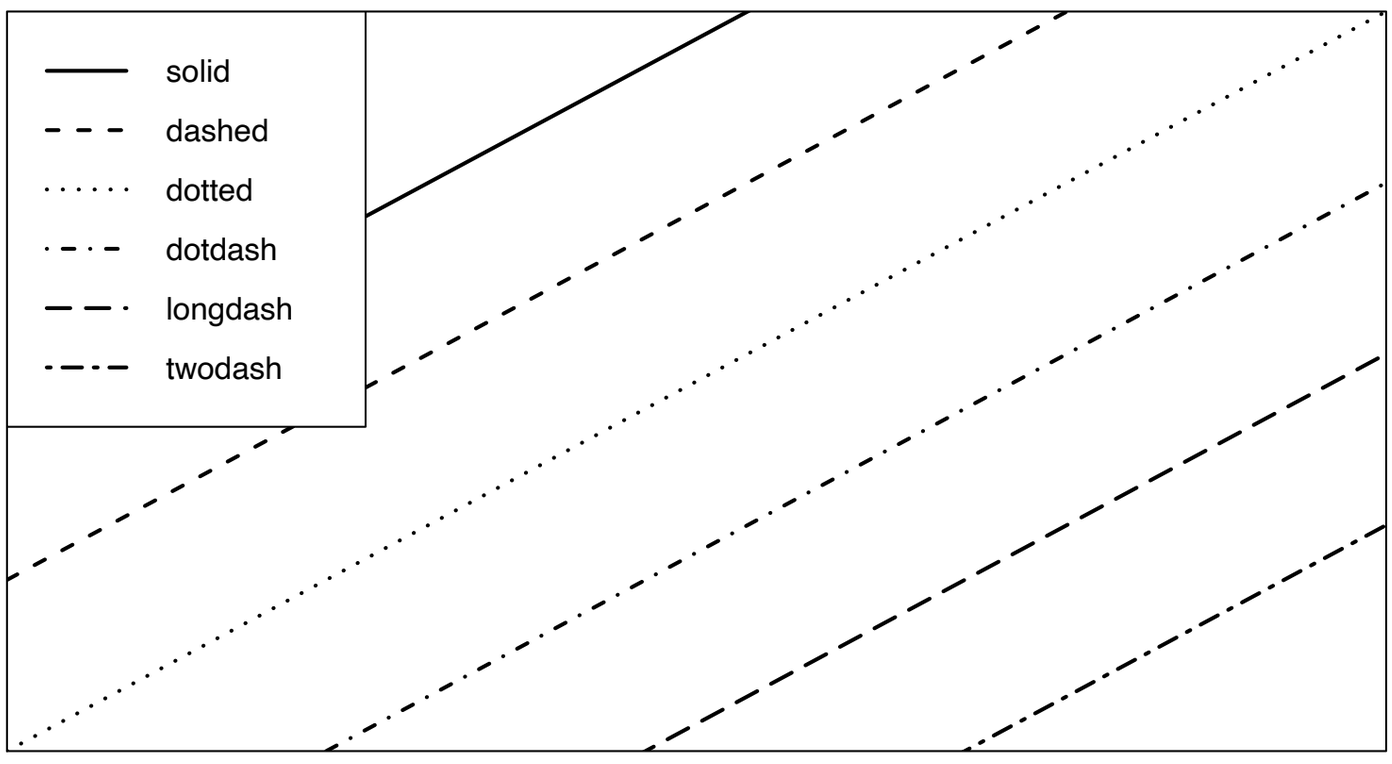
```
> plot(Year, HR, xlab="Season",pch=19,cex=0.9,
+      ylab="Avg. Home Runs Hit by a Team in a Game",
+      main="Home Run Hitting in the MLB Across Seasons")
> lines(lowess(Year, HR),lwd=2)
```



Graphs

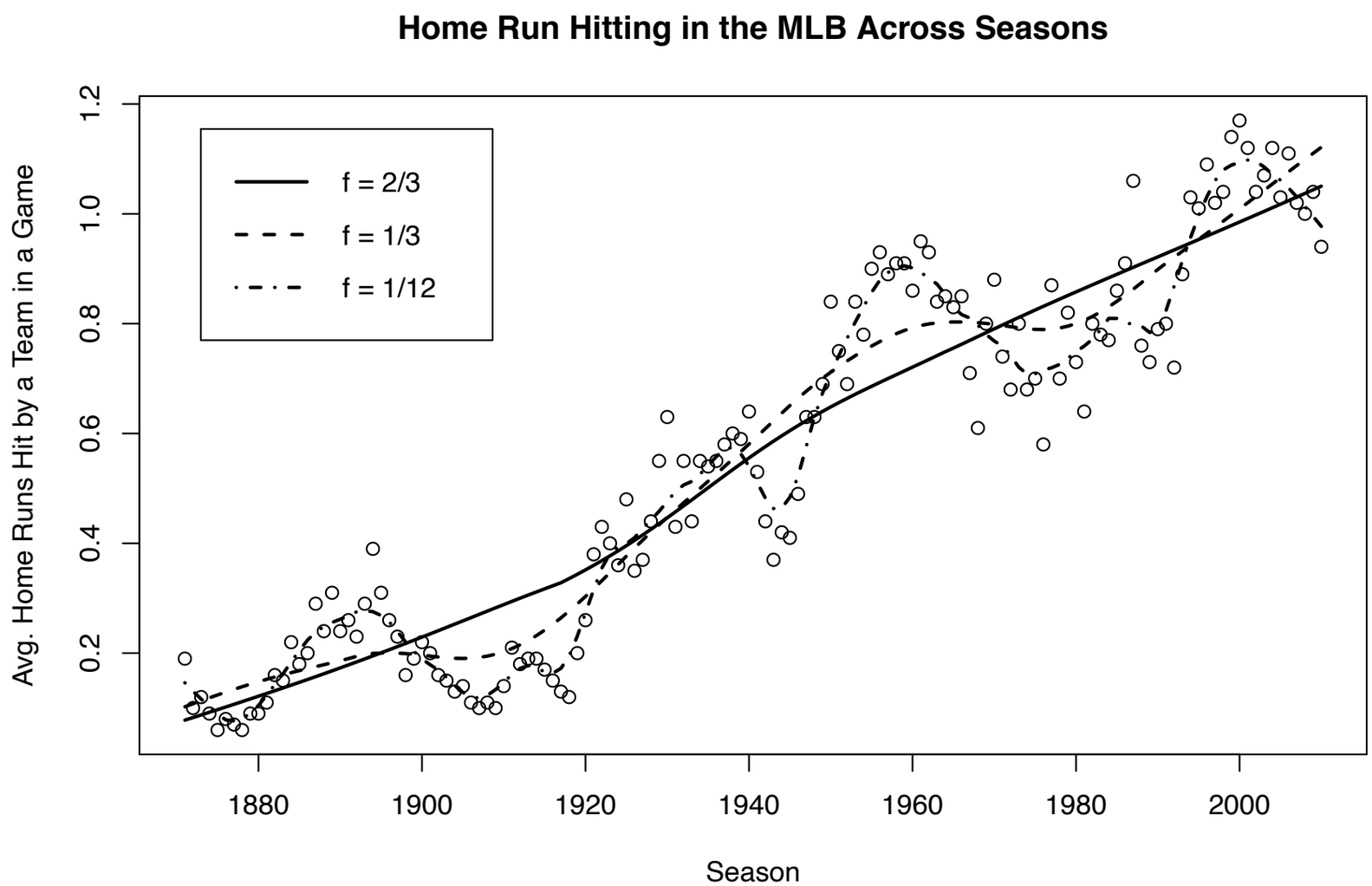
- Types of lines:

```
> plot(0, 0, type="n", xlim=c(-2, 2), ylim=c(-2, 2),
+      xaxt="n", yaxt="n", xlab="", ylab="")
> y = seq(2, -3, -1)
> for(j in 1:6)
+ abline(a=y[j], b=1, lty=j, lwd=2)
> legend("topleft", legend=c("solid", "dashed", "dotted",
+ "dotdash", "longdash", "twodash"), lty=1:6, lwd=2)
> title("Line Styles with the lty Argument")
```



Graphs

```
> plot(Year, HR, xlab="Season",
+      ylab="Avg. Home Runs Hit by a Team in a Game",
+      main="Home Run Hitting in the MLB Across Seasons")
> lines(lowess(Year, HR), lwd=2)
> lines(lowess(Year, HR, f=1 / 3), lty="dashed", lwd=2)
> lines(lowess(Year, HR, f=1 / 12), lty="dotdash", lwd=2)
> legend("topleft", legend=c("f = 2/3", "f = 1/3",
+ "f = 1/12"), lty=c(1, 2, 4), lwd=2, inset=0.05)
```



Colors

- Be aware of color blindness

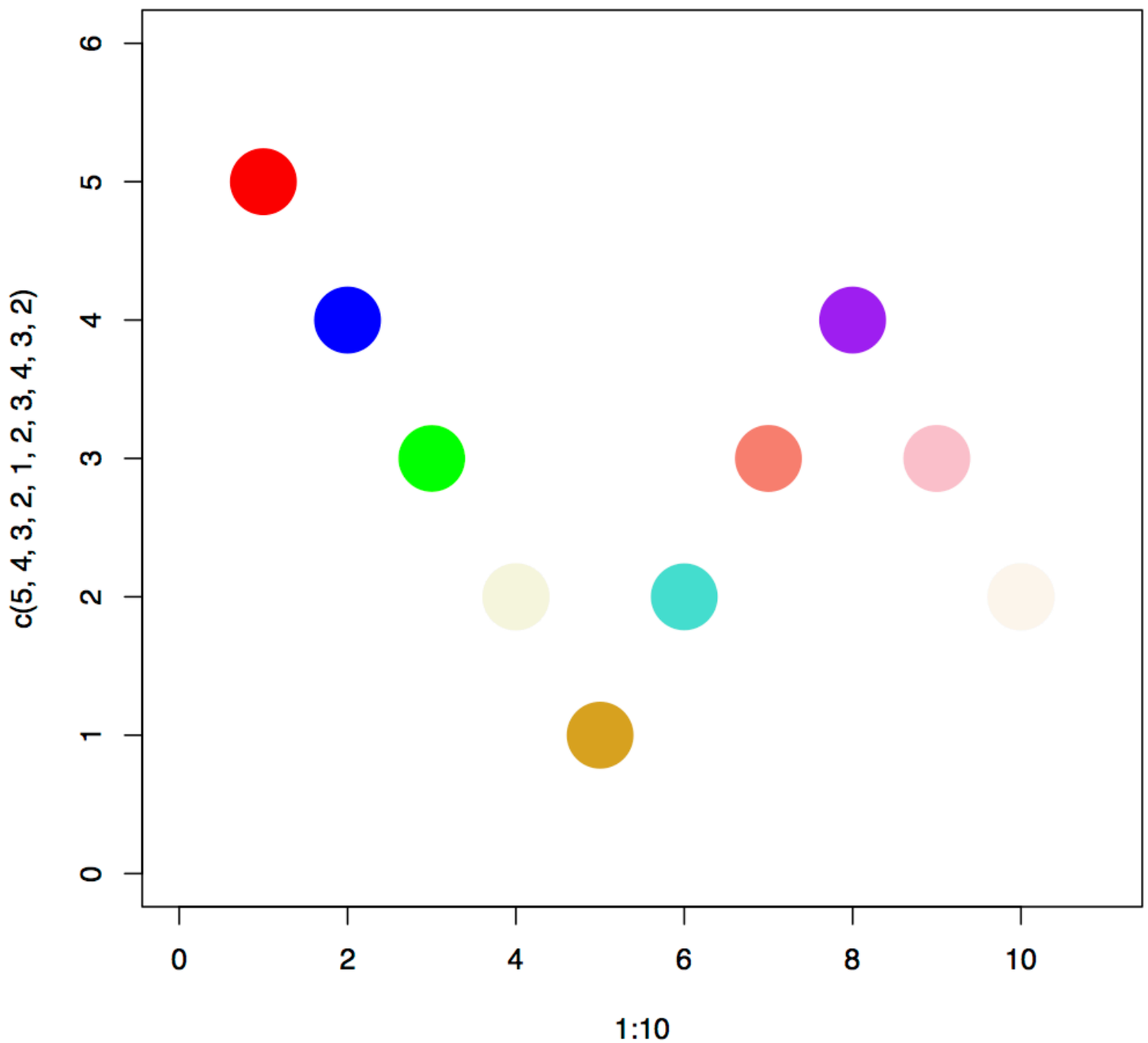
- Colorblind-safe palettes have been created by Cynthia Brewer creator of the ColorBrewer website and corresponding R package. You need to install the package RColorBrewer. The function brewer.pal allows you to choose colors from Brewer's color palettes.

Colors

```
> colors()
  [1] "white"                "aliceblue"            "antiquewhite"
  [4] "antiquewhite1"        "antiquewhite2"        "antiquewhite3"
  [7] "antiquewhite4"        "aquamarine"           "aquamarine1"
 [10] "aquamarine2"          "aquamarine3"          "aquamarine4"
 [13] "azure"                "azure1"               "azure2"
 [16] "azure3"               "azure4"               "beige"
 [19] "bisque"               "bisque1"              "bisque2"
 [22] "bisque3"              "bisque4"              "black"
 [25] "blanchedalmond"       "blue"                 "blue1"
 [28] "blue2"                "blue3"                "blue4"
 [31] "blueviolet"           "brown"                "brown1"
 [34] "brown2"               "brown3"               "brown4"
 [37] "burlywood"            "burlywood1"           "burlywood2"
 [40] "burlywood3"           "burlywood4"           "cadetblue"
 [43] "cadetblue1"           "cadetblue2"           "cadetblue3"
 [46] "cadetblue4"           "chartreuse"           "chartreuse1"
.
.
.
```

Graphs

```
> plot(1:10, c(5, 4, 3, 2, 1, 2, 3, 4, 3, 2), pch=19, cex=5,
ylim=c(0,6),xlim=c(0,11),col=c("red", "blue", "green", "beige",
“goldenrod","turquoise", "salmon", "purple", "pink", "seashell"))
```

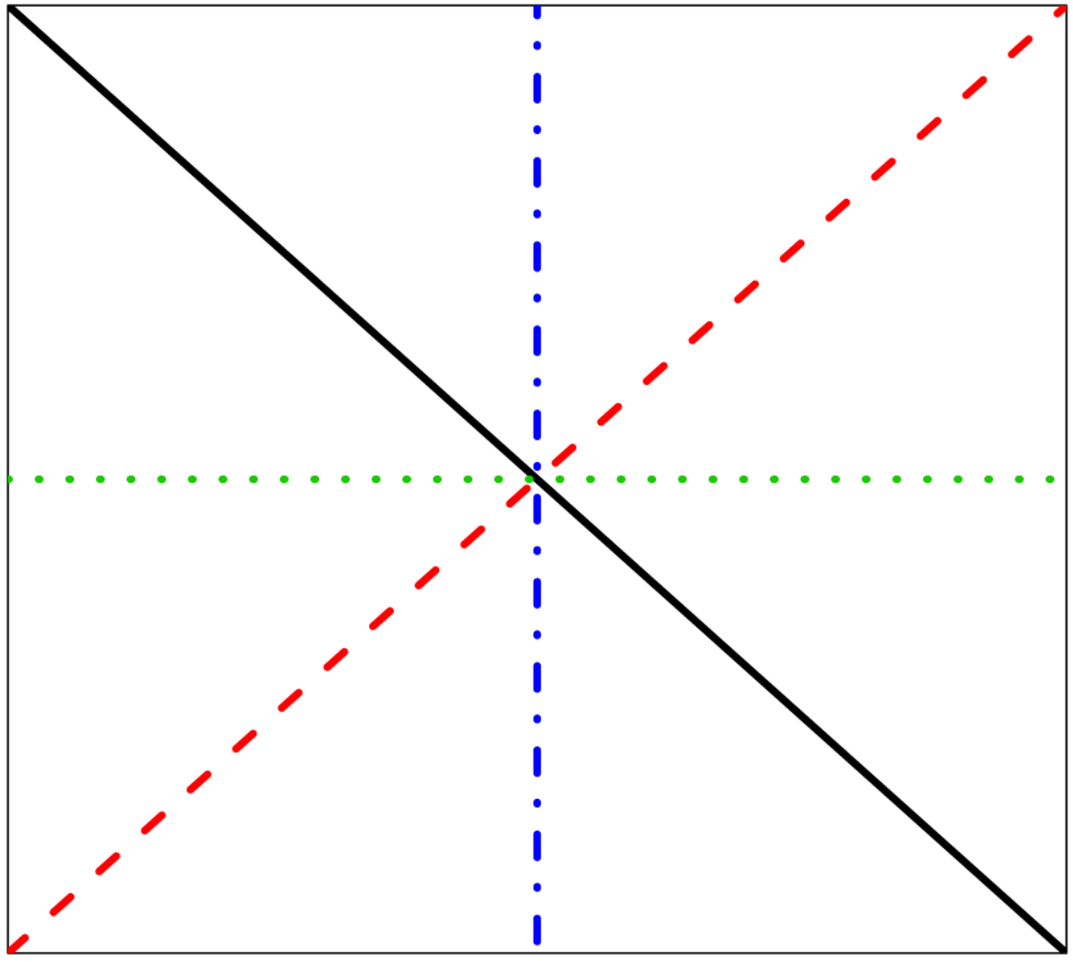


Graphs

- Colors can also be specified using numbers and `col`. Numbers 1-8 correspond to:

```
> palette()
[1] "black"   "red"     "green3"  "blue"    "cyan"    "magenta" "yellow"
[8] "gray"
```

```
> plot(0, 0, type="n", xlim=c(-2, 2),
+ylim=c(-2, 2), xaxt="n", yaxt="n",
xlab="", ylab="")
> y = c(-1, 1, 0, 50000)
> for (j in 1:4)
+   abline(a=0, b=y[j], lty=j,
lwd=4,col=j)
```



Text Format

```
> plot(0, 0, type="n", xlim=c(-1, 6), ylim=c(-0.5, 4),
+      xaxt="n", yaxt ="n", xlab="", ylab="",
+      main="Font Choices Using, font, family and srt Arguments")
> text(2.5, 4, "font = 1 (Default)")
> text(1, 3, "font = 2 (Bold)", font=2, cex=1.0)
> text(1, 2, "font = 3 (Italic)", font=3, cex=1.0)
> text(1, 1, "font = 4 (Bold Italic), srt = 20", font=4,
+      cex=1.0, srt=20)
> text(4, 3, 'family="serif"', cex=1.0, family="serif")
> text(4, 2, 'family="sans"', cex=1.0, family="sans")
> text(4, 1, 'family="mono"', cex=1.0, family="mono")
> text(2.5, 0, 'family = "HersheyScript"', cex=2.5,
+      family="HersheyScript", col="red")
```

font style

rotation angle

size

font family

Font Choices Using, font, family and srt Arguments

font = 1 (Default)

font = 2 (Bold)

family="serif"

font = 3 (Italic)

family="sans"

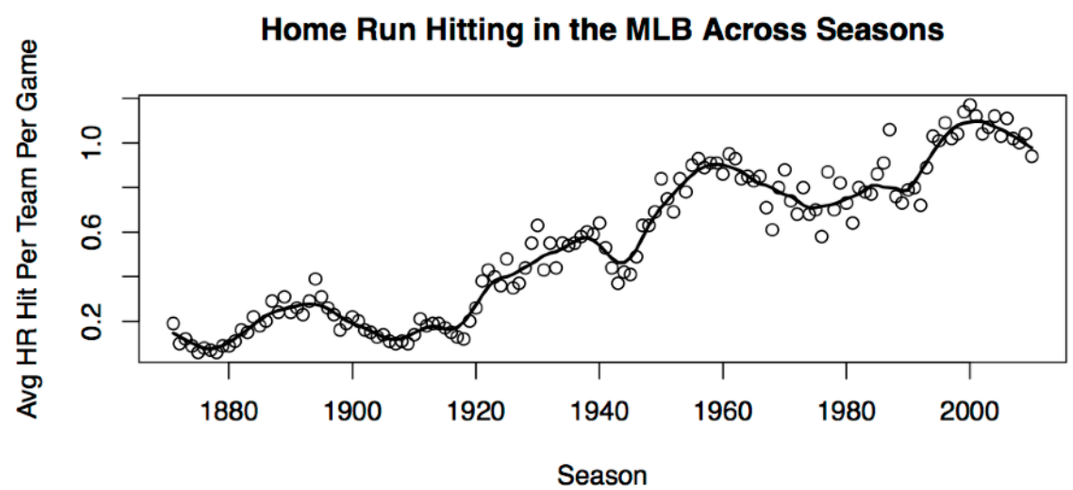
font = 4 (Bold Italic), srt = 20

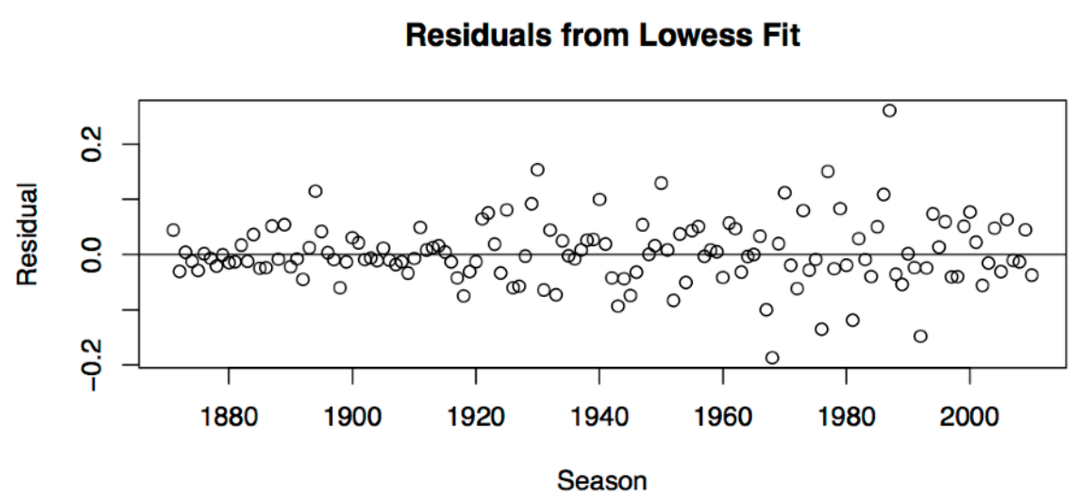
family="mono"

family = "HerseyScript"

Multiple figures in a window

```
> par(mfrow=c(2,1))
> plot(Year, HR, xlab="Season",
+      ylab="Avg HR Hit Per Team Per Game",
+      main="Home Run Hitting in the MLB Across Seasons")
> lines(fit, lwd=2)
> plot(Year, Residual, xlab="Season",
+      main="Residuals from Lowess Fit")
> abline(h=0)
```

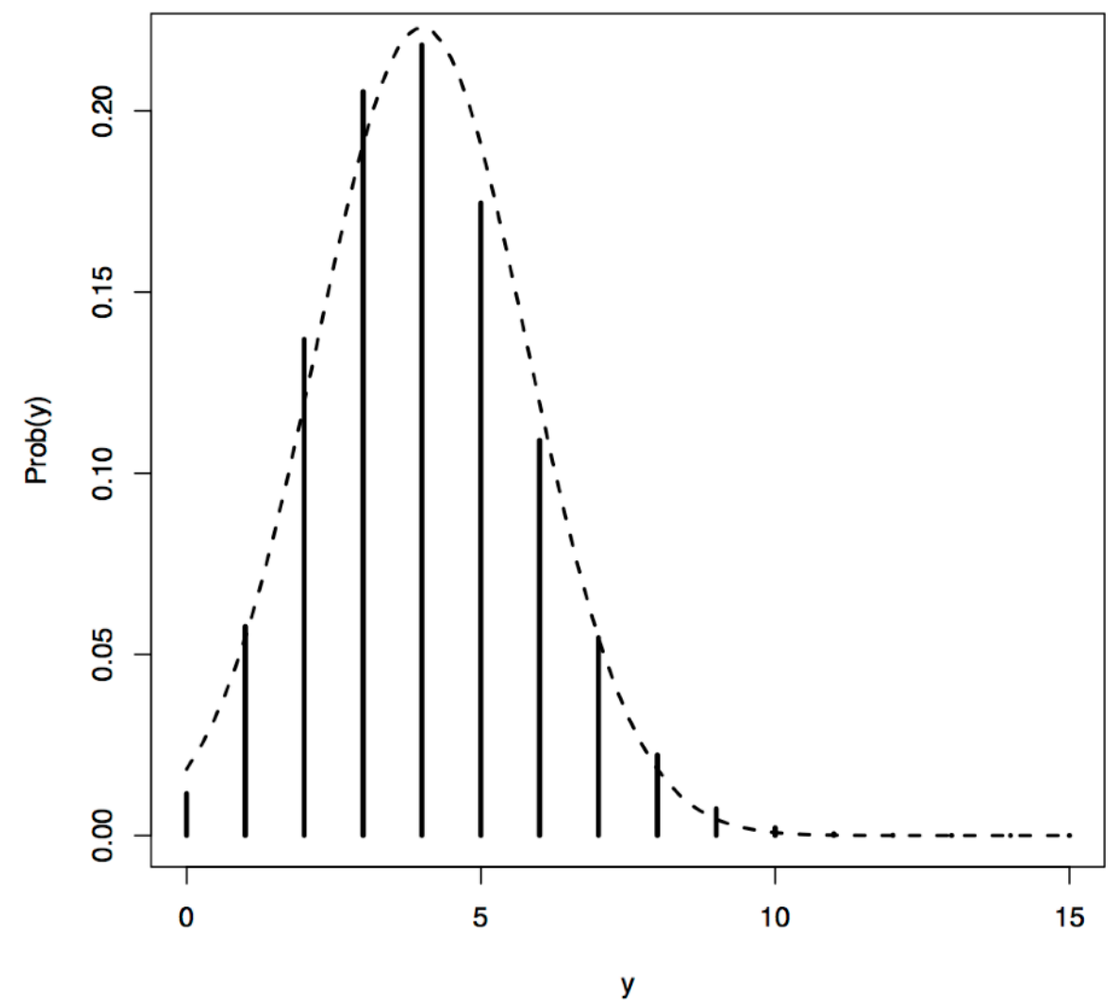




Overlaying a curve and adding mathematical expressions

Example: The binomial can be approximated by the normal distribution. We consider the distribution Binom(n,p) with n=20 and p=0.2 and the normal approximation N(m,v) with m=n*p and v=n*p*(1-p).

normal density is added with `curve`



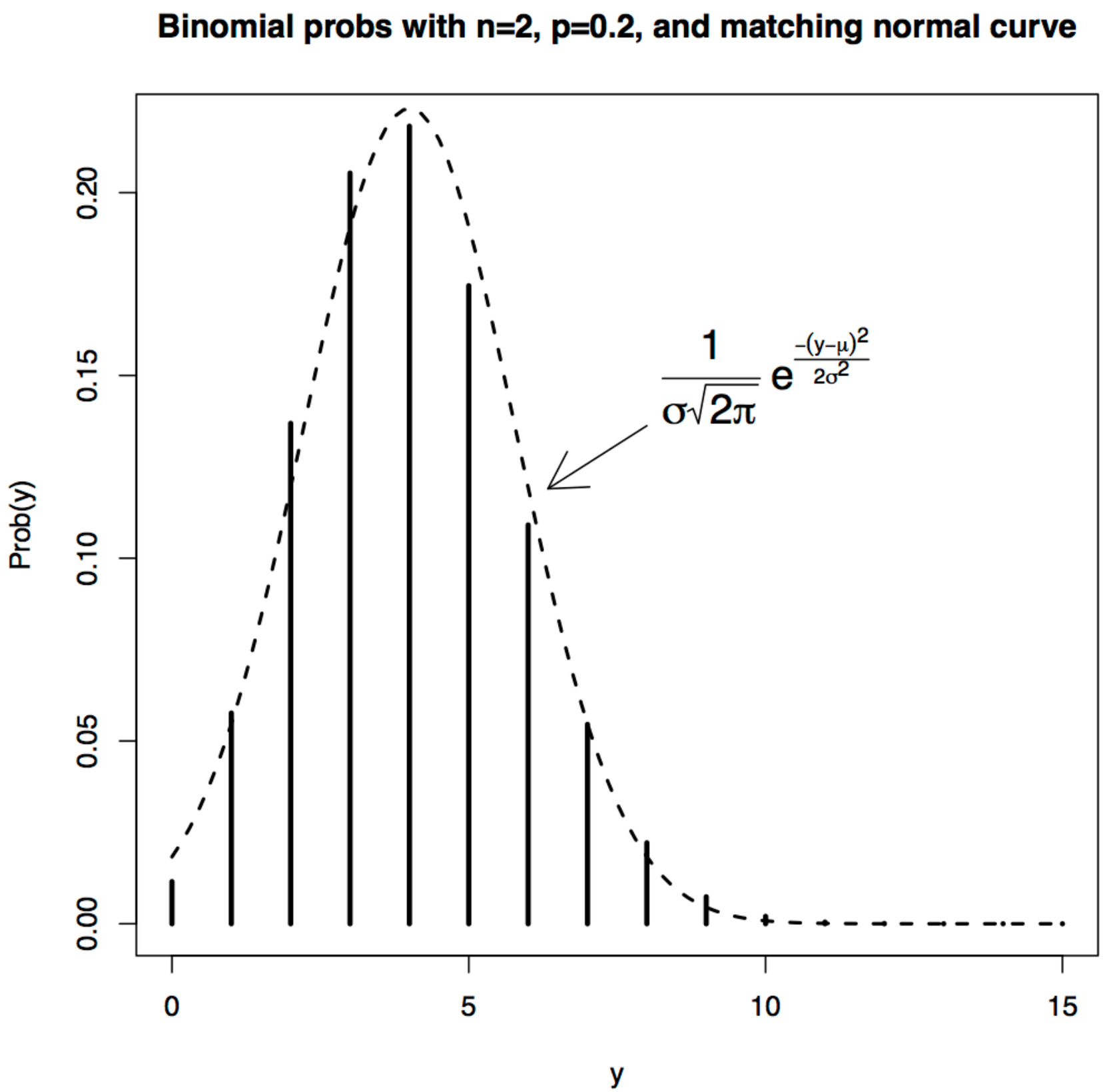

```
> n = 20; p = 0.2
> y = 0:20
> py = dbinom(y, size=n, prob=p)
> plot(y, py, type="h", lwd=3,
+      xlim=c(0, 15), ylab="Prob(y)")
> mu = n * p; sigma = sqrt(n * p * (1 - p))
> curve(dnorm(x, mu, sigma), add=TRUE, lwd=2, lty=2)
```

Graphs

- To add mathematical expressions we use the function `expression`

```
> text(10, 0.15,
+   expression(paste(frac(1, sigma*sqrt(2*pi)), " ",
+   e^{frac(-(y-mu)^2, 2*sigma^2)})), cex = 1.5)
> title("Binomial probs with n=2, p=0.2, and matching
normal curve”)
> locs=locator(2)
> arrows(locs$x[1], locs$y[1], locs$x[2], locs$y[2])
```

Graphs


Binomial probs with n=2, p=0.2, and matching normal curve
Prob(y)
0.00
0.05
0.10
0.15
0.20
0
5
10
15
y
$\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(y-\mu)^2}{2\sigma^2}}$

Multiple plots and varying parameters

- the function `layout` can be used to divide a window into several regions
- the default location of the plot region can be displayed using the `plt` argument of `par`

Example: Golden Snowball Award given to the city in NY state with the most snowfall during the winter (2 winters available: 09-10 and 10-11)

```
> snow.yr1 = c(85.9, 71.4, 68.8, 58.8, 34.4)
> snow.yr2 = c(150.9, 102.0, 86.2, 80.1, 63.8)
> layout(matrix(c(1, 2), ncol=1), heights=c(6, 4))
> par("plt")
[1] 0.1171429 0.9400000 0.3642857 0.7071429
```

Graphing region for first plot:

- x direction: (x1,x2)= (0.117, 0.94)
- y direction: (y1,y2)= (0.36, 0.707)

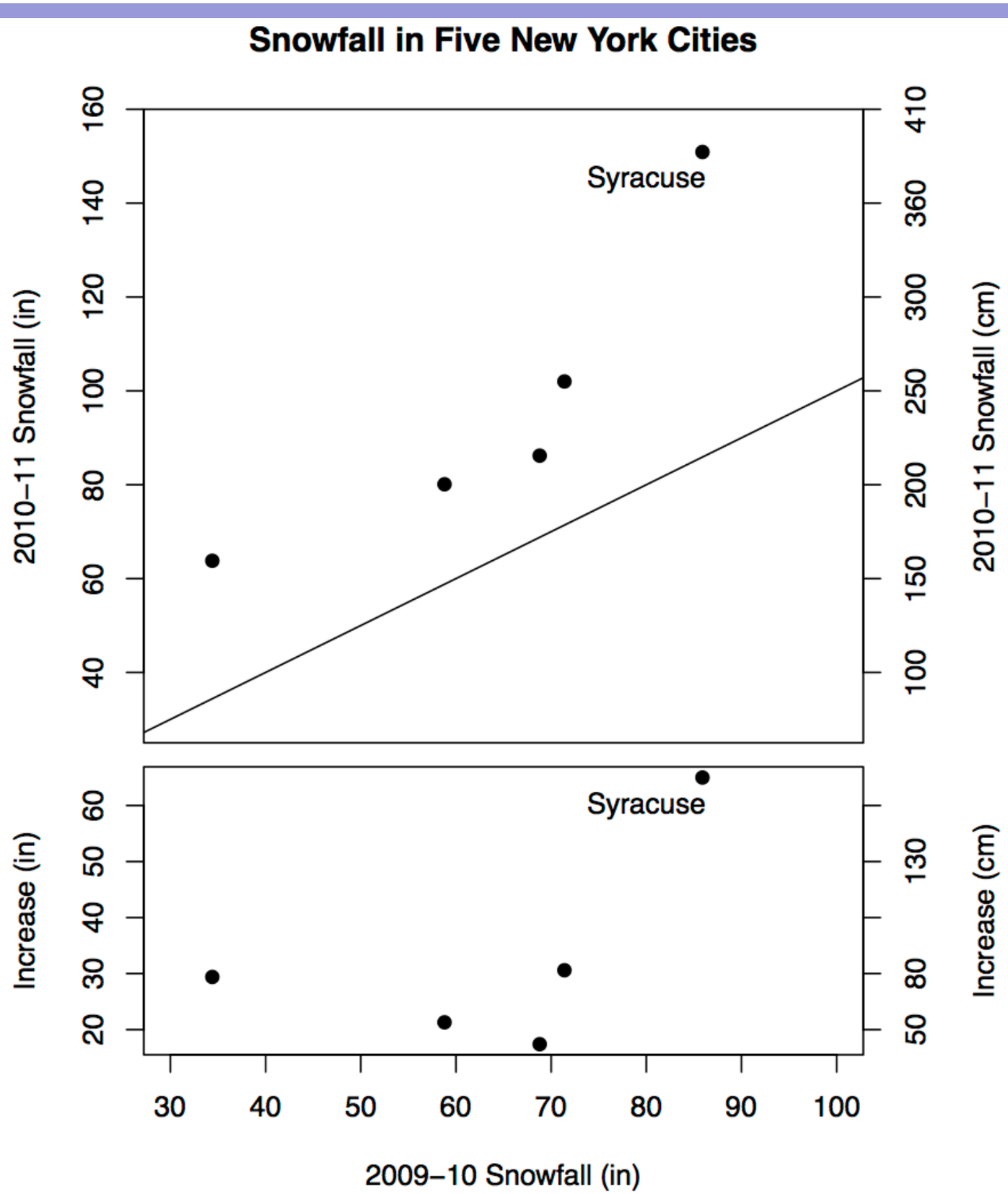
Graphs

```
> snow.yr1 = c(85.9, 71.4, 68.8, 58.8, 34.4)
> snow.yr2 = c(150.9, 102.0, 86.2, 80.1, 63.8)
> layout(matrix(c(1, 2), ncol=1), heights=c(6, 4))
> par("plt")
[1] 0.1171429 0.9400000 0.3642857 0.7071429
> par(plt=c(0.20, 0.80, 0, 0.88), xaxt="n")
> plot(snow.yr1, snow.yr2, xlim=c(30, 100), ylim=c(30, 155),
+      ylab="2010-11 Snowfall (in)", pch=19,
+      main="Snowfall in Five New York Cities")
> abline(a=0, b=1)
> text(80, 145, "Syracuse")
> tm = par("yaxp")
> tm
[1]  40 160   6
> ticmarks = seq(tm[1], tm[2], length=tm[3]+1)
> ticmarks
[1]  40  60  80 100 120 140 160
> axis(4, at=ticmarks,
+      labels=as.character(round(2.54 * ticmarks, -1)))
> mtext("2010-11 Snowfall (cm)", side=4, line=3)
```

next plot:

```
> par(plt=c(0.20, 0.80, 0.35, 0.95), xaxt="s")
> plot(snow.yr1, snow.yr2 - snow.yr1, xlim=c(30, 100),
+      xlab="2009-10 Snowfall (in)", pch=19,
+      ylab="Increase (in)")
> text(80, 60, "Syracuse")
> tm=par("yaxp")
> ticmarks=seq(tm[1], tm[2], length=tm[3] + 1)
> axis(4, at=ticmarks,
+      labels=as.character(round(2.54 * ticmarks, -1)))
> mtext("Increase (cm)", side=4, line=3)
```

Graphs



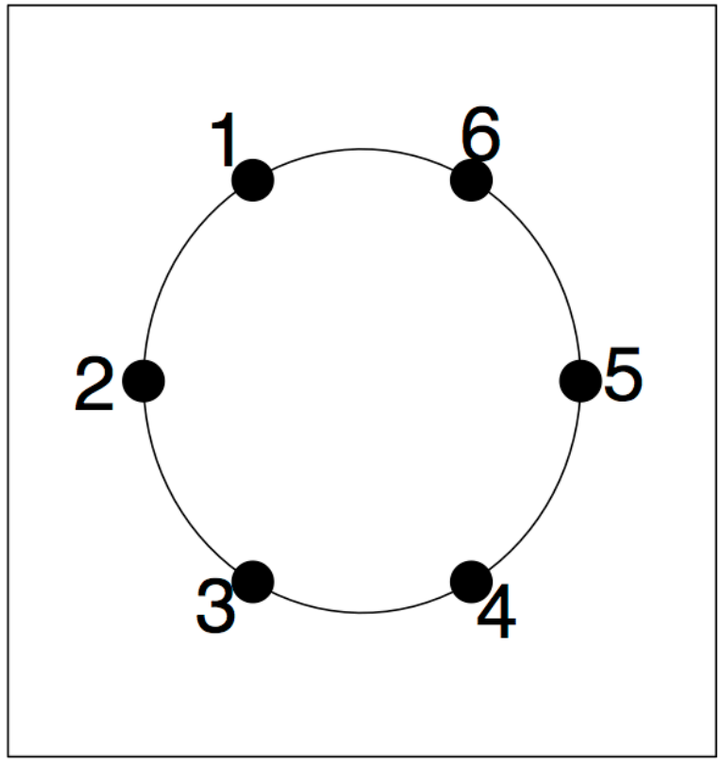
What is the graph telling us?

- all cities had more snow during 10-11
- Syracuse had > 60in additional inches and the rest had > 25in

Creating graphs using low-level functions

Example: Graphing a circle with labels

```
> plot.new()
> plot.window(xlim=c(-1.5, 1.5), ylim=c(-1.5, 1.5), pty="s")
> theta = seq(0, 2*pi, length=100)
> lines(cos(theta), sin(theta))
> theta=seq(0, 2*pi, length=7)[-7]
> points(cos(theta), sin(theta), cex=3, pch=19)
> pos = locator(6)
> text(pos, labels=1:6, cex=2.5)
> box()
```



Exporting Graphs

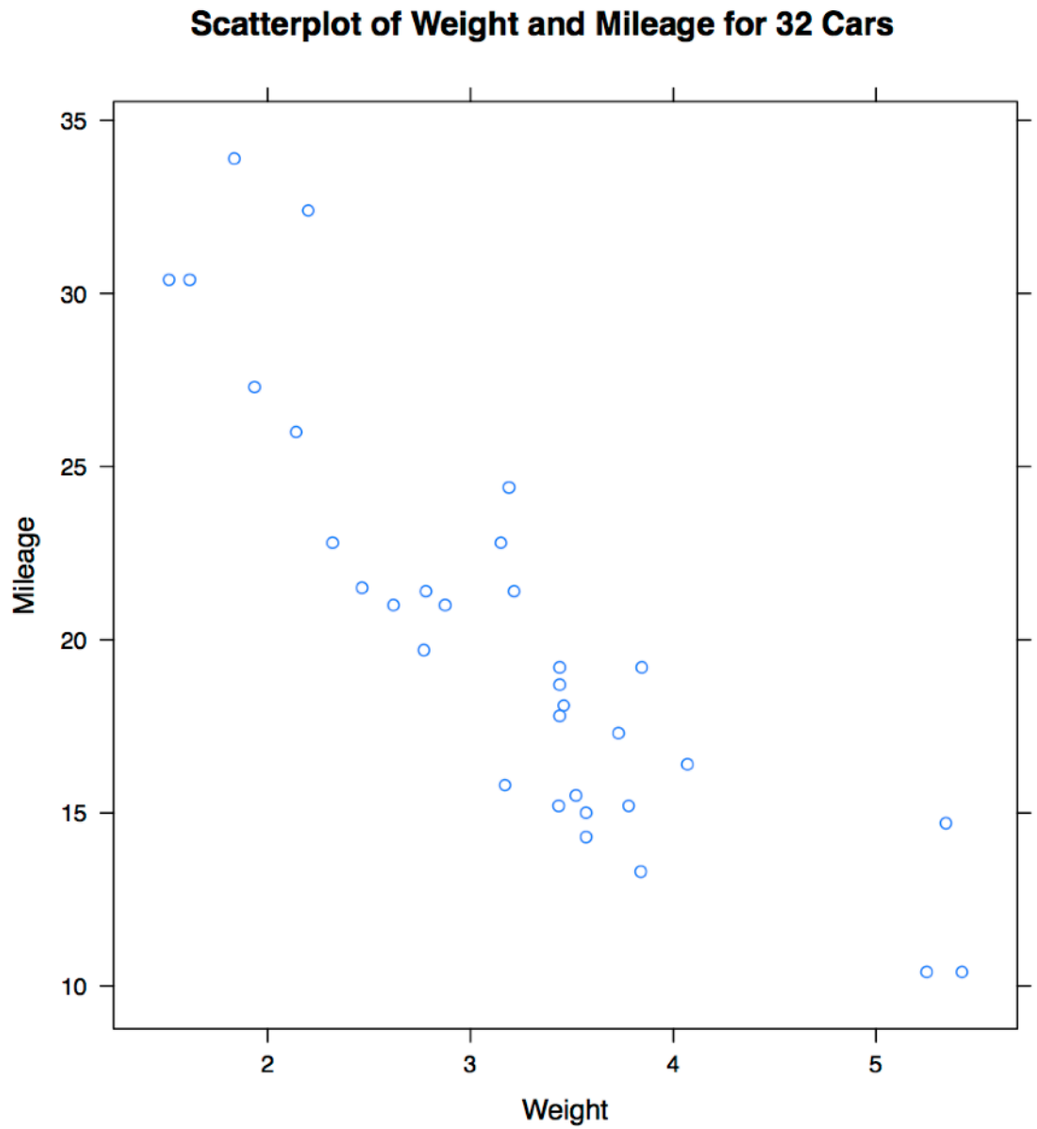
- Several options are available, including pdf, gif, png, jpeg, ps. See `?Devices` for more details.

```
> pdf(file="plot.pdf")
> plot(x,y)
> dev.off()

> plot(x,t)
> dev.copy2pdf(file="plot.pdf",width=4,height=4)

> png(file="plot.png")
> plot(x,y)
> dev.off()
```

Graphs

The `lattice` package is an implementation of the Trellis graphics for **R** developed by D. Sarkar**.** Trellis was originally developed for S and S-PLUS at Bell Labs.

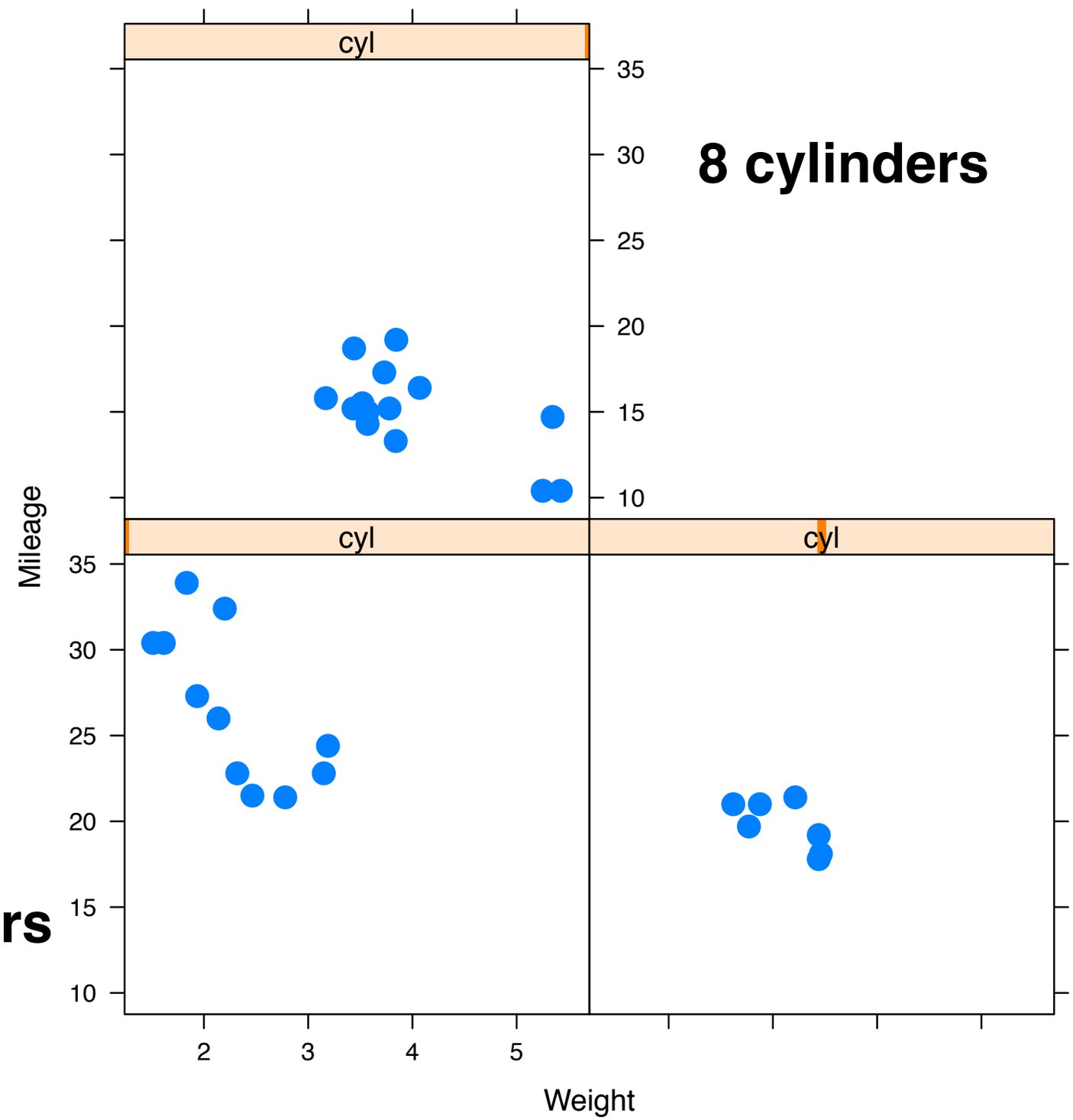
```
> library(lattice)
> xyplot(mpg ~ wt, data=mtcars, xlab="Weight",ylab="Mileage",
+        main="Scatterplot of Weight and Mileage for 32 Cars")
```



- Milage depends on the number of cylinders
- If we control for number of cylinders is there still an association between milage and weight?

Graphs

```
> xyplot(mpg ~ wt | cyl, data=mtcars, pch=19, cex=1.5,
+        xlab="Weight", ylab="Mileage")
```



8 cylinders

4 cylinders

6 cylinders

Graphs

```
> densityplot(~ wt, groups=cyl, data=mtcars,
+             auto.key=list(space="top"))
```