lecture10

author: date:

Today

- Random numbers (and functions for distributions)
- Plotting (part 1)

r, d, p, q

-**r**andom

-**d**ensity

-**p**robability

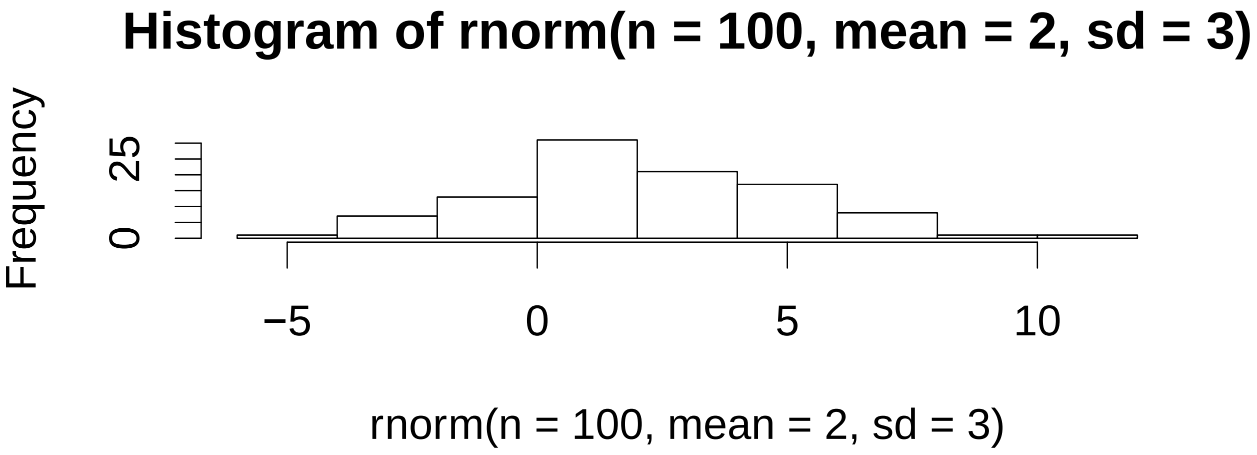
-**q**uantile

-letter followed by distrubtion abbreviation is/does each one

-example abbreviations: norm, t, f, exp, binom, pois, unif, etc.

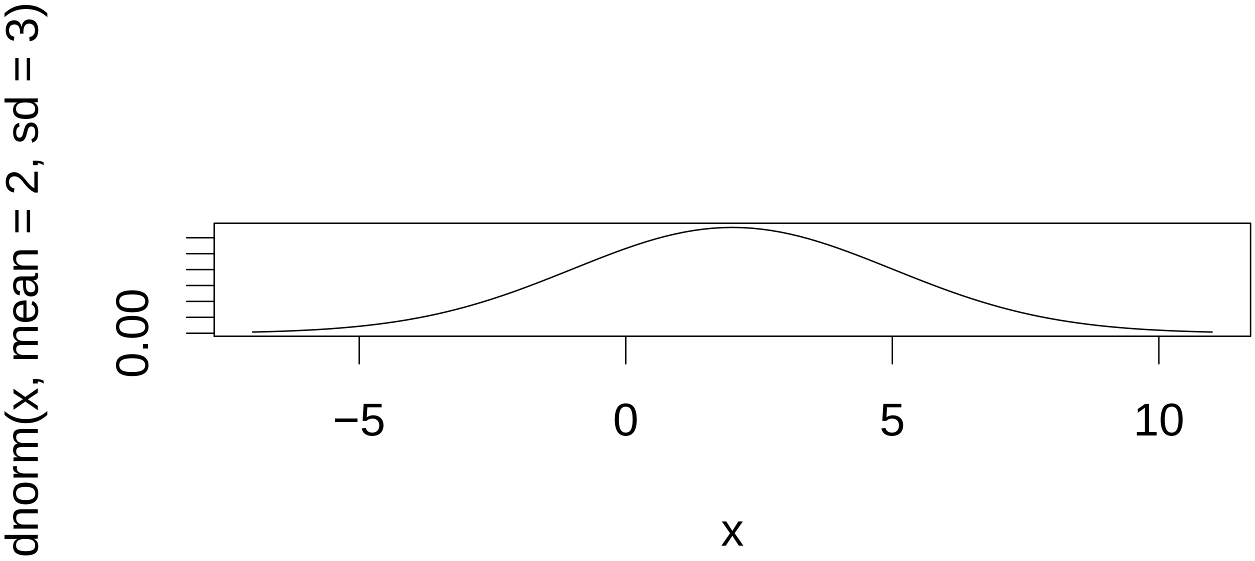
rnorm(n=100,mean=2,sd=3) makes n random #s

```r
# cex makes all everything in the plot 2 time larger
par(cex = 2)
# every distrubtion has ther respective parameters (eg. unif has upper/lower
# boud or gamma has shape/scale) random is great way to make data
hist(rnorm(n = 100, mean = 2, sd = 3))
```



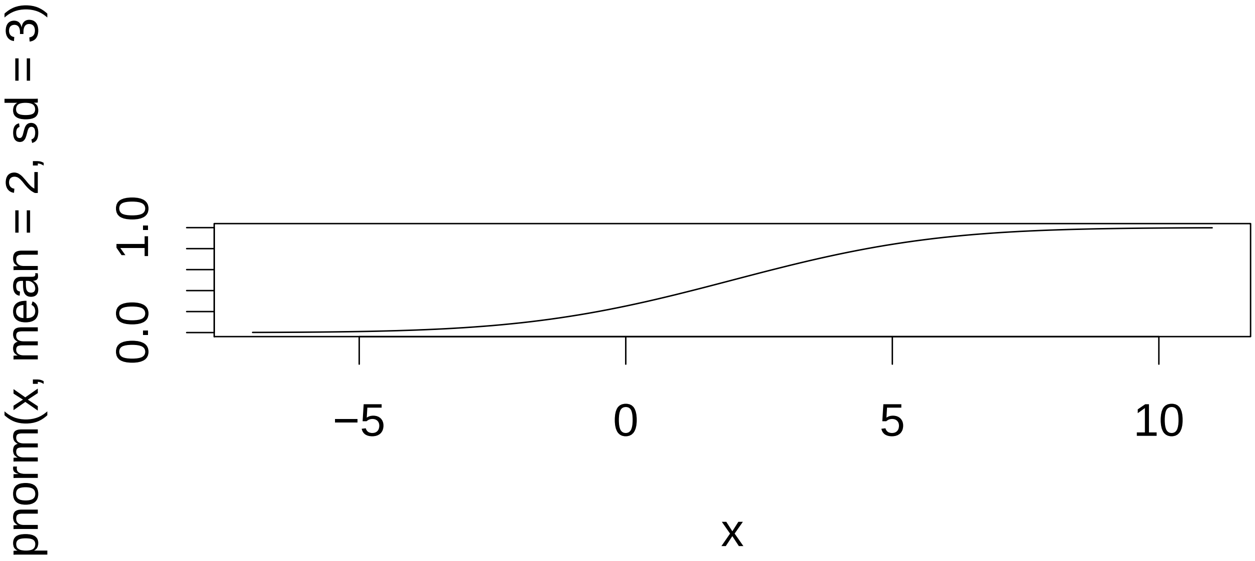
dnorm(x,mean=2,sd=3) evaluates pdf @ x

```
par(cex = 2)
# curve function takes a dummy variable x and take a lower and upper bound
# and plots the function we want in this case we have from = lower bound, to
# = upper bound
curve(dnorm(x, mean = 2, sd = 3), from = 2 - 3 * 3, to = 2 + 3 * 3)
```



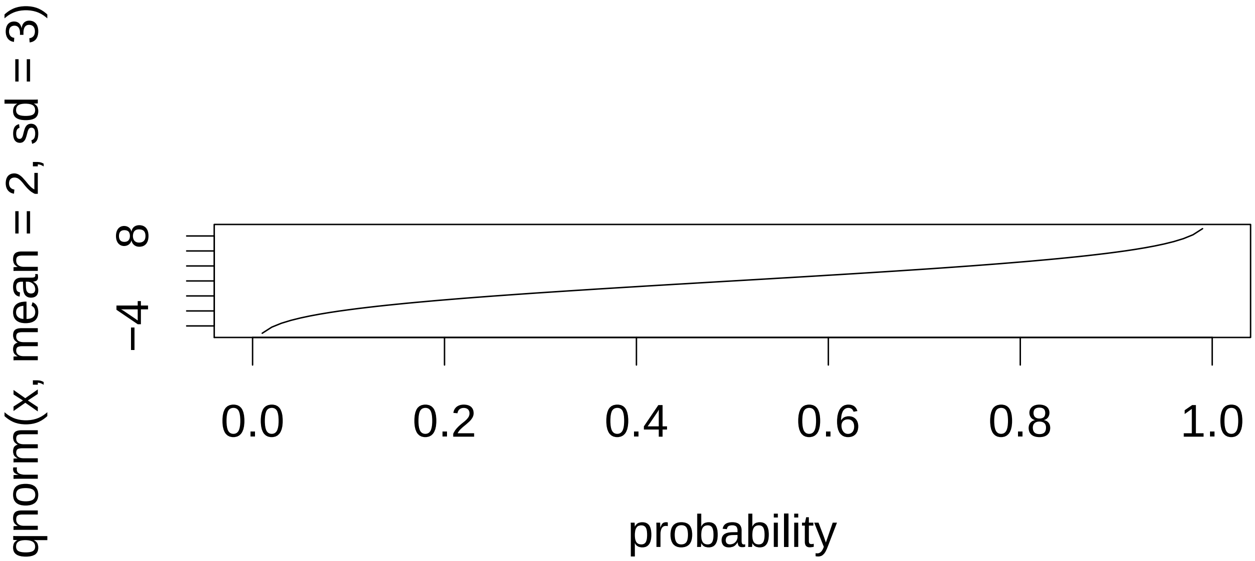
pnorm(x,mean=2,sd=3) evaluates cdf @ x

```
par(cex = 2)
# cummulative density function
curve(pnorm(x, mean = 2, sd = 3), from = 2 - 3 * 3, to = 2 + 3 * 3)
```



qnorm(p,mean=2,sd=3) inverse cdf @ p

```r
par(cex = 2)
# inverse probability (note in book) used in confidence intervals when
# computing th z,t,f.etc values
curve(qnorm(x, mean = 2, sd = 3), from = 0, to = 1, xlab = "probability")
```



Plotting - If you can think it, R can draw it

```r
example(plot) # try these on your own
example(barplot)
example(boxplot)
example(dotchart)
example(coplot)
example(hist)
example(fourfoldplot)
example(stars)
example(image)
example(contour)
example(filled.contour)
example(persp)
```

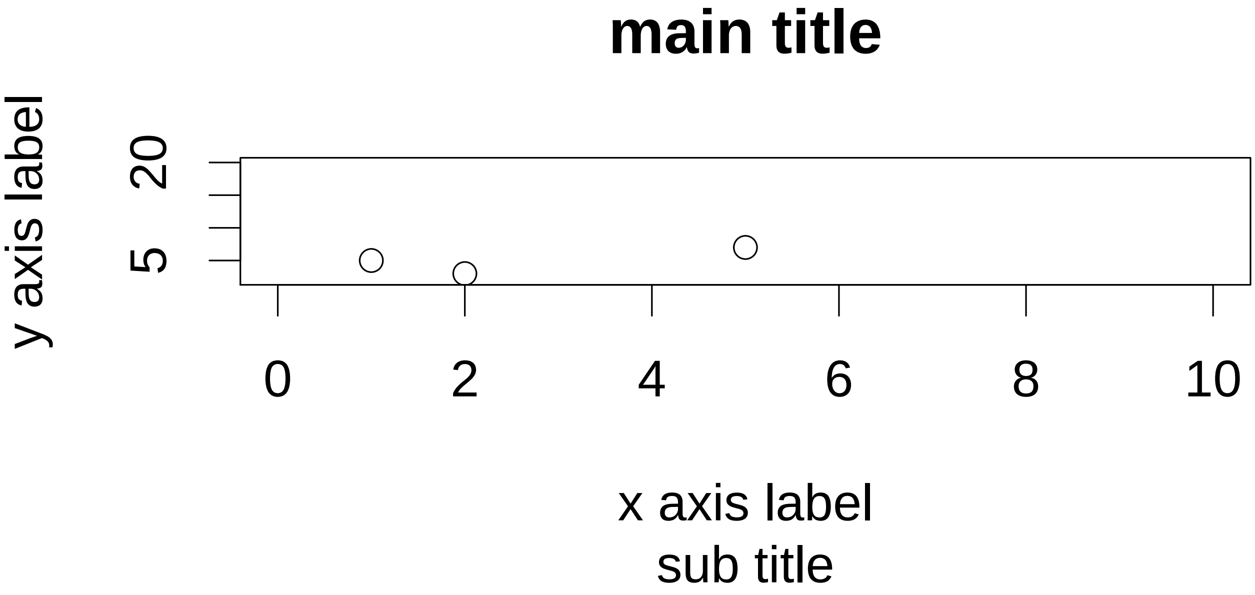
Five topics (1 and 2 today)

1. Setting up the plotting frame: `plot`()
2. Adding to an existing ploting frame: `lines`(), `points`(), `text`(), `polygon`(), `arrows`(), . . .
3. More control over the plotting frame: `par`()
4. Multiple plots in a frame: `par`(`mfrow` = `c`(`rows`, `columns`)), `layout`()

5. Making pdfs: `pdf()`, `dev.off()`

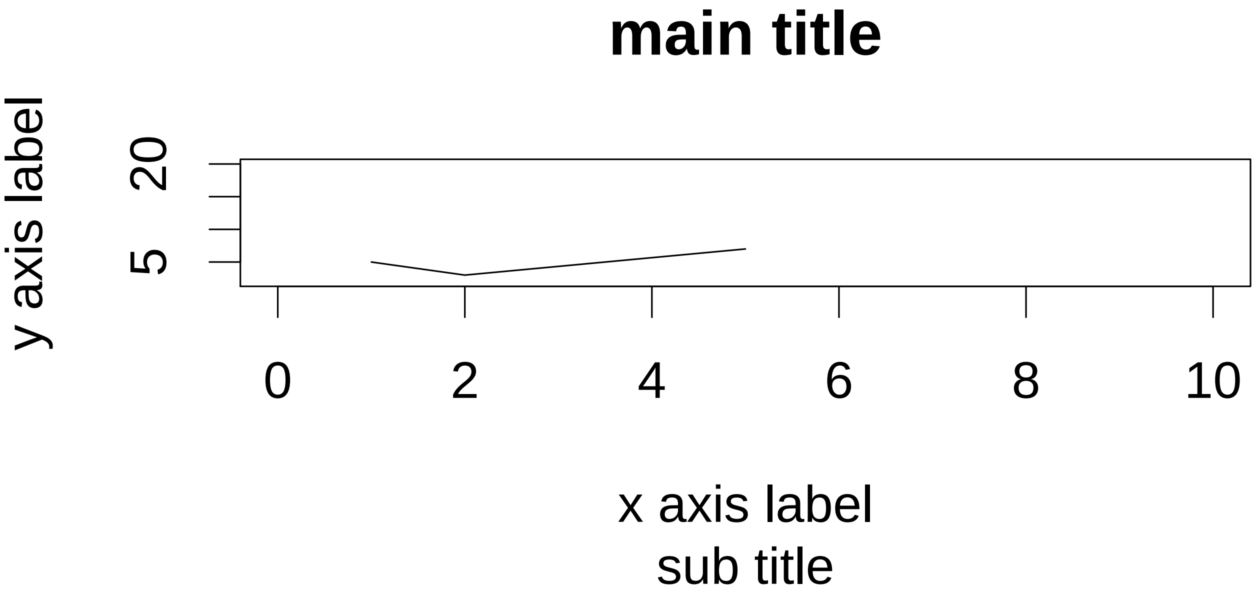
Establish a plotting frame

```r
x <- c(1, 2, 5)
y <- c(5, 3, 7)
par(cex = 2)
# required arguments in plot i the x and the y vector which HAVE to be the
# same length you can also add the labels and title/subtitles for labels you
# can use paste statements because they are text x and y axis limits using
# xlim,ylim you can use range of the the x vector to get xlim
plot(x, y, xlab = "x axis label", ylab = "y axis label", main = "main title",
    sub = "sub title", xlim = c(0, 10), ylim = c(2, 20), type = "p")
```



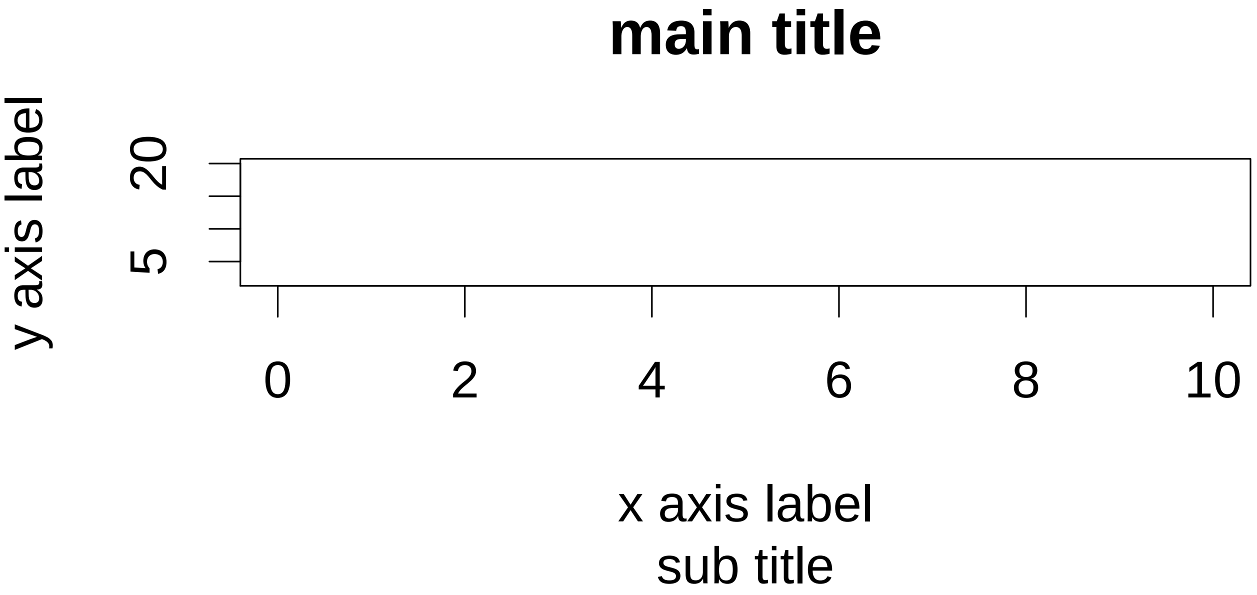
Establish a plotting frame =====

```r
par(cex = 2)
# type='l' connects the dots and makes a line
plot(x, y, xlab = "x axis label", ylab = "y axis label", main = "main title",
    sub = "sub title", xlim = c(0, 10), ylim = c(2, 20), type = "l")
```



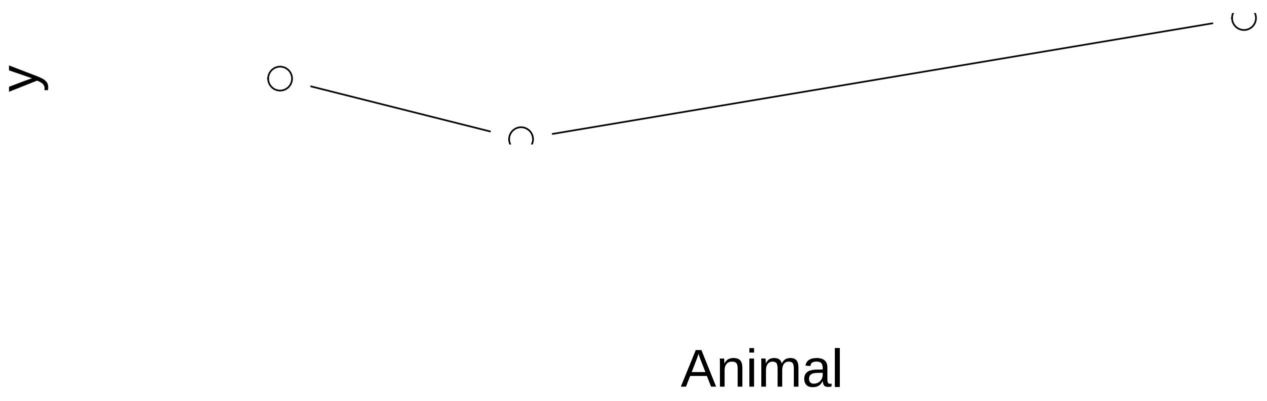
Establish a plotting frame =====

```
par(cex = 2)
# type='n' is nothing (empty plot)
plot(x, y, xlab = "x axis label", ylab = "y axis label", main = "main title",
    sub = "sub title", xlim = c(0, 10), ylim = c(2, 20), type = "n")
```



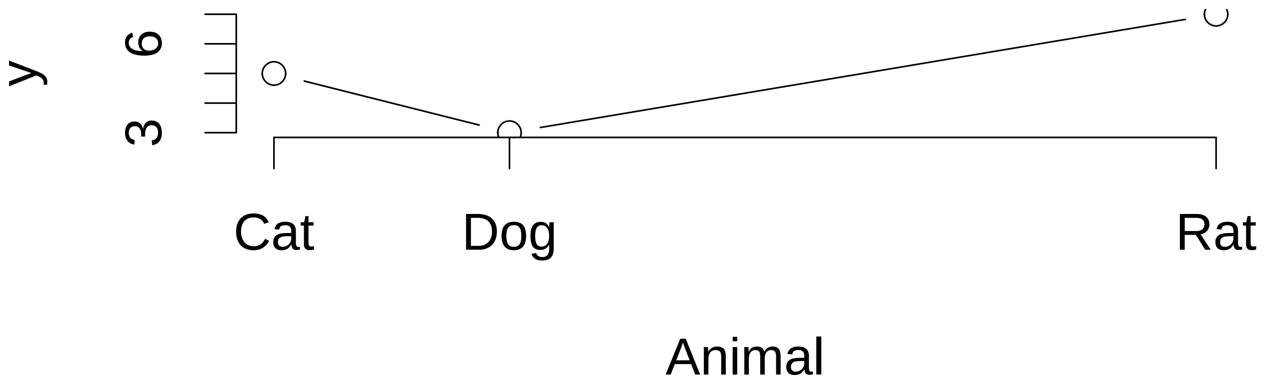
Establish a plotting frame

```
par(cex = 2)
# axes=F turns the axes off
plot(x, y, type = "b", axes = F, xlab = "Animal")
```



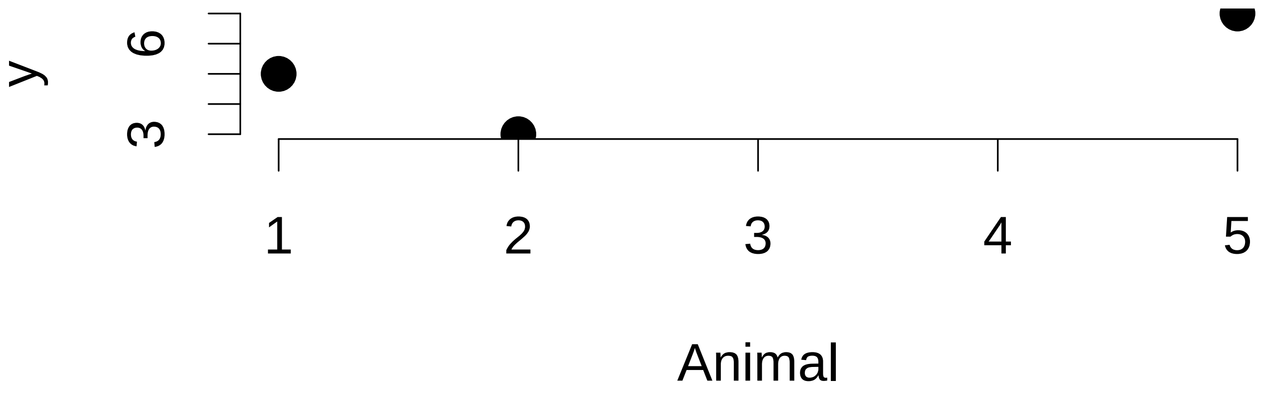
Establish a plotting frame

```r
par(cex = 2)
# type='b' adds both line and points
plot(x, y, type = "b", axes = F, xlab = "Animal")
# axis (1) adds the x axis at shows where to make the ticks
axis(1, at = c(1, 2, 5), labels = c("Cat", "Dog", "Rat"))
# axis (2) adds y axis
axis(2)
```



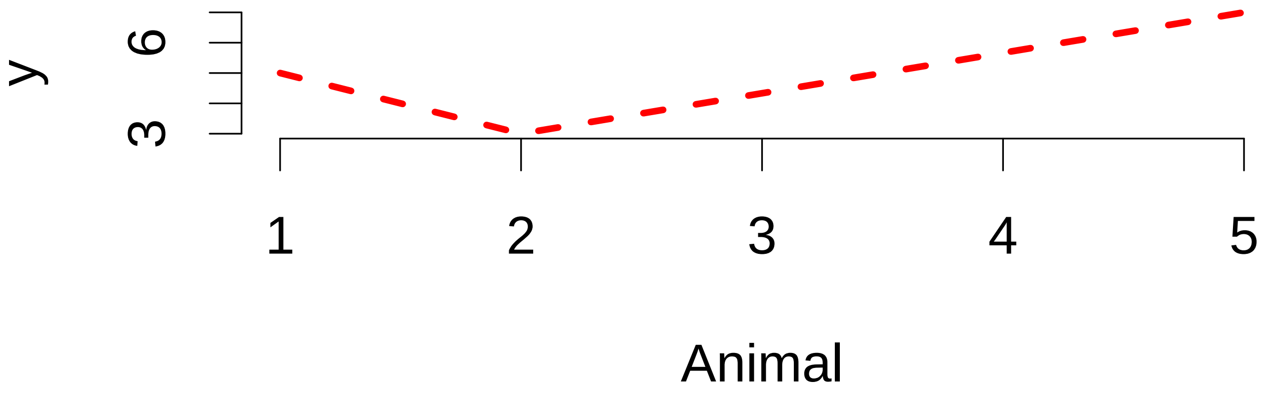
Adding to an existing ploting frame

```r
par(cex = 2)
plot(x, y, type = "n", axes = F, xlab = "Animal")
# adds points into the frams pch is the shape of the point cex makes them
# bigger/smaller
points(x, y, pch = 16, cex = 1.5)
axis(1)
axis(2)
```



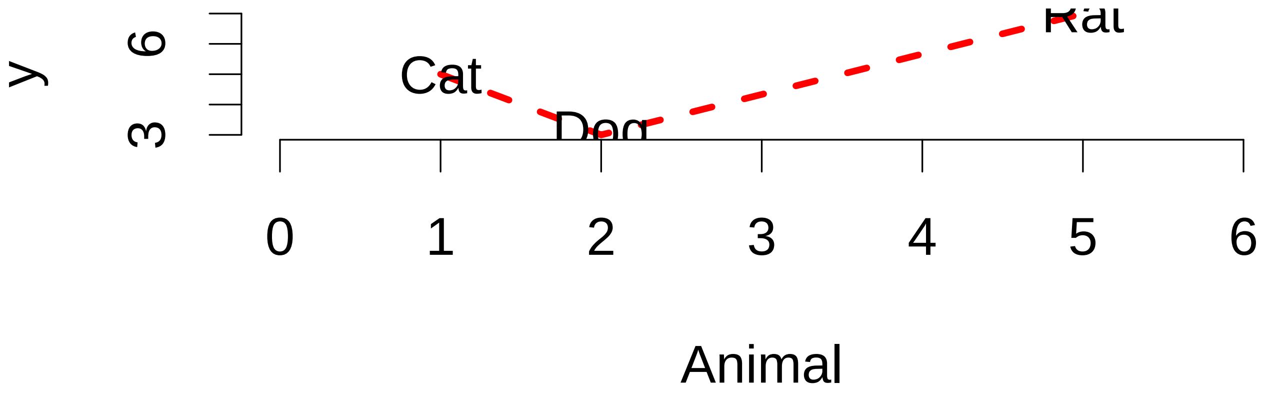
Adding to an existing ploting frame

```r
par(cex = 2)
plot(x, y, type = "n", axes = F, xlab = "Animal")
# plots lines lty is for line type lwd is the thickness of the line color of
# the lines
lines(x, y, lty = 2, lwd = 4, col = "red")
axis(1)
axis(2)
```



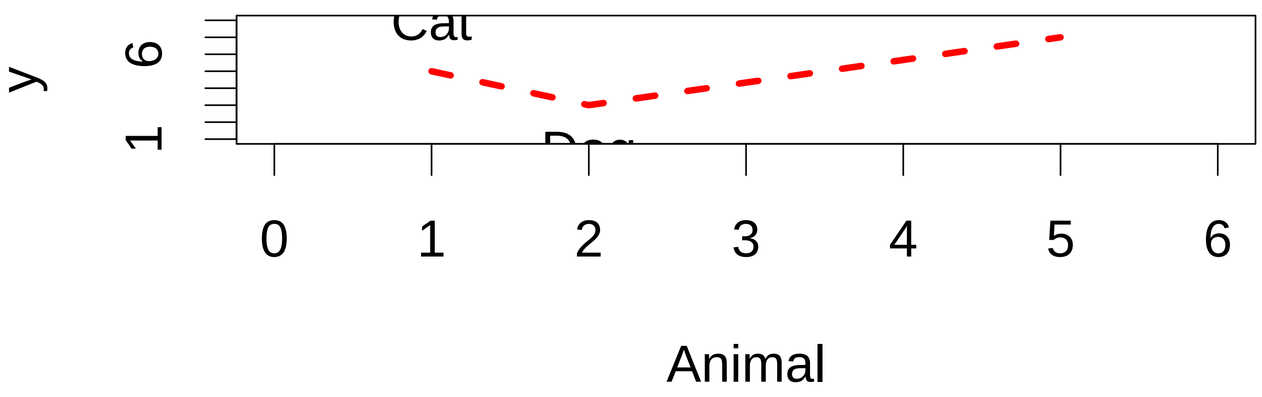
Adding to an existing ploting frame

```r
par(cex = 2)
plot(x, y, type = "n", axes = F, xlab = "Animal", xlim = c(0, 6))
lines(x, y, lty = 2, lwd = 4, col = "red")
# adds text giving it x,y coordinates tells it that we want the text placed
# at those coordinates
text(x, y, c("Cat", "Dog", "Rat"))
axis(1)
axis(2)
```



Adding to an existing ploting frame

```r
par(cex = 2)
plot(x, y, type = "n", axes = T, xlab = "Animal", xlim = c(0, 6), ylim = c(1,
    8))
lines(x, y, lty = 2, lwd = 4, col = "red")
# text has a position argument 3 is above/ 1 is below, etc...  point of this
# is that it labels without overwriting the point
text(x, y, c("Cat", "Dog", "Rat"), pos = c(3, 1, 3))
```



Example:

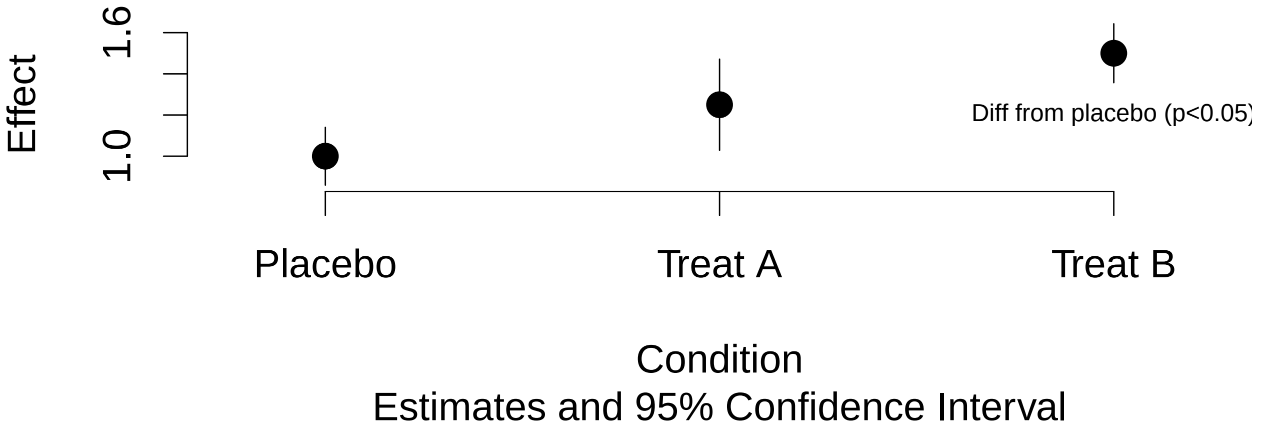
```r
# this is made up data
# in real get estimates
    # variances
    # sample size
ests <- c(1,1.25,1.5)
ns <- c(20,15,10)
sds <- c(.3,.4,.2)
est.table <- data.frame(condition=c("Placebo","Treatment A","Treatment B"),
                        est.effect=ests,
              # calculating confidence intervals for each of the estimates
                            lowers=ests-qt(.975,df=(ns-1))*sds/sqrt(ns),
                            uppers=ests+qt(.975,df=(ns-1))*sds/sqrt(ns))
est.table
```

```
##     condition est.effect    lowers   uppers
## 1     Placebo       1.00 0.8595957 1.140404
## 2 Treatment A       1.25 1.0284874 1.471513
## 3 Treatment B       1.50 1.3569286 1.643071
```

Example:

```
# setting up an empty plot ylim is the range of y
plot(1:3, 1:3, xlim = c(0.75, 3.25), ylim = range(est.table[, -1]), type = "n",
    axes = F, xlab = "Condition", ylab = "Effect", sub = "Estimates and 95% Confidence Interval")
# adding the axis and putting ticks and labels by the treatment type
axis(2)
axis(1, at = 1:3, labels = est.table$condition)
# loop through the table of estimates and plot
for (i in 1:dim(est.table)[1]) {
    # plot the points
    points(i, est.table$est.effect[i], pch = 16, cex = 1.5)
    # add the lines for upper and lower confidence
    lines(c(i, i), c(est.table$lowers[i], est.table$uppers[i]))
}
# adding text to the third point
text(3, est.table$lowers[3], pos = 1, "Diff from placebo (p<0.05)", cex = 0.6)
```

Example:



To be continued

(Next 3 topics on another day…)