INTRODUCTION TO R PART 2

Distribution-related functions

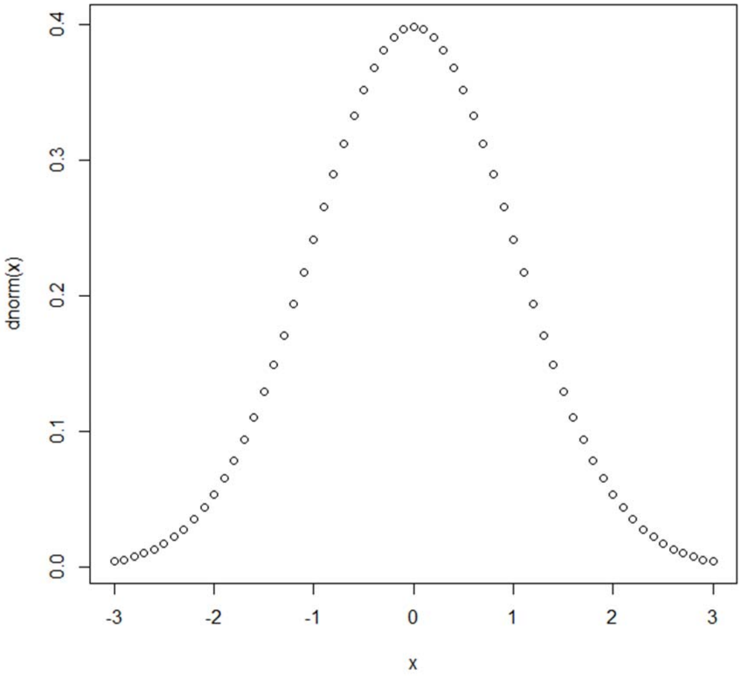
For normal distributions

- **`dnorm(x, mean = 0, sd = 1)`**
  - Returns density function (“d” -> density)
- **`pnorm(q, mean = 0, sd = 1)`**
  - Returns distribution function (“p” -> probability)
- **`qnorm(p, mean = 0, sd = 1)`**
  - Returns quantile function (“q” -> quantile)
- **`rnorm(n, mean = 0, sd = 1)`**
  - Returns random observations (“r”-> random)

Similar: **`dbinom, pbinom, qbinom, rbinom`**

Examples

- `x<-seq(-3,3,0.1)`
- `plot(x,dnorm(x))`



Figures!!

- R’s best advantage

Linear Regression Example

- Create a dataset by simulating 200 observations from the following linear model:
  - Y = beta0 + beta1 * X1 + noise
- where beta0=1, beta1=2
- X1 ~ N(1, 4), noise ~ N(0,1)

Code

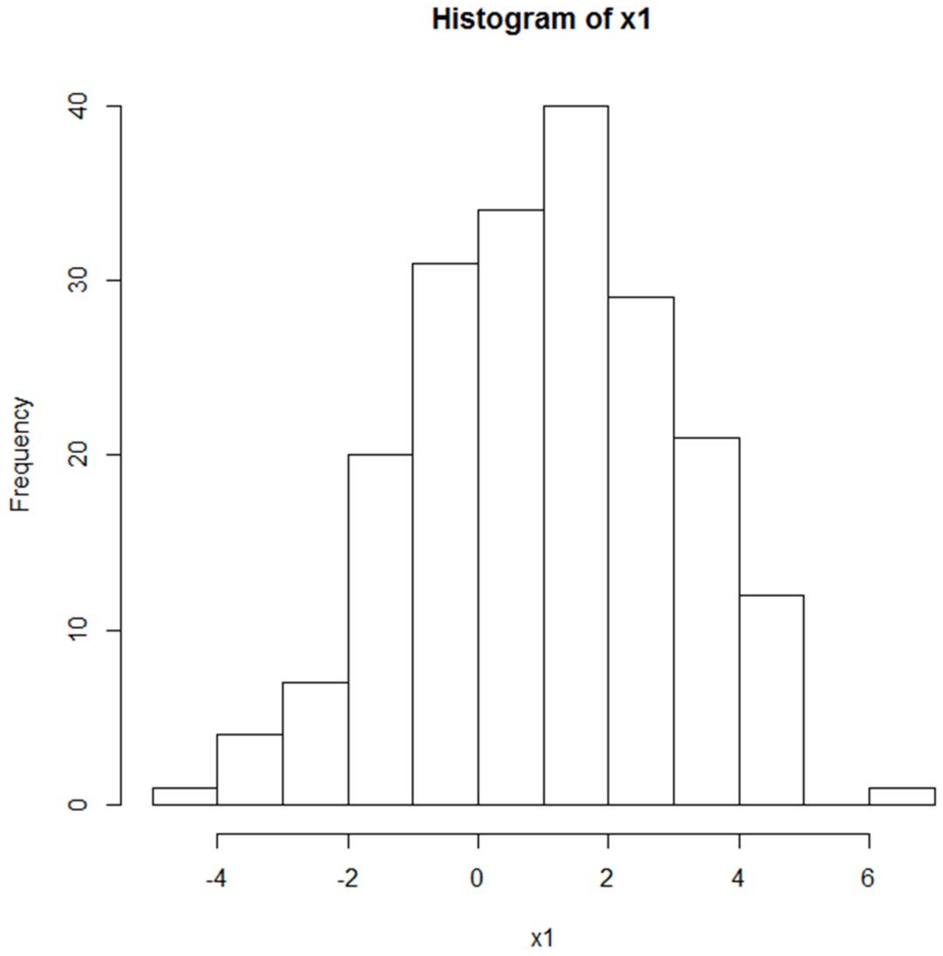
```
x1<-rnorm(200,mean=1,sd=2)
noise<-rnorm(200)
y<1+2*x1+noise
simulation<-data.frame(x1,y)
```

Plots that show the distribution of data values

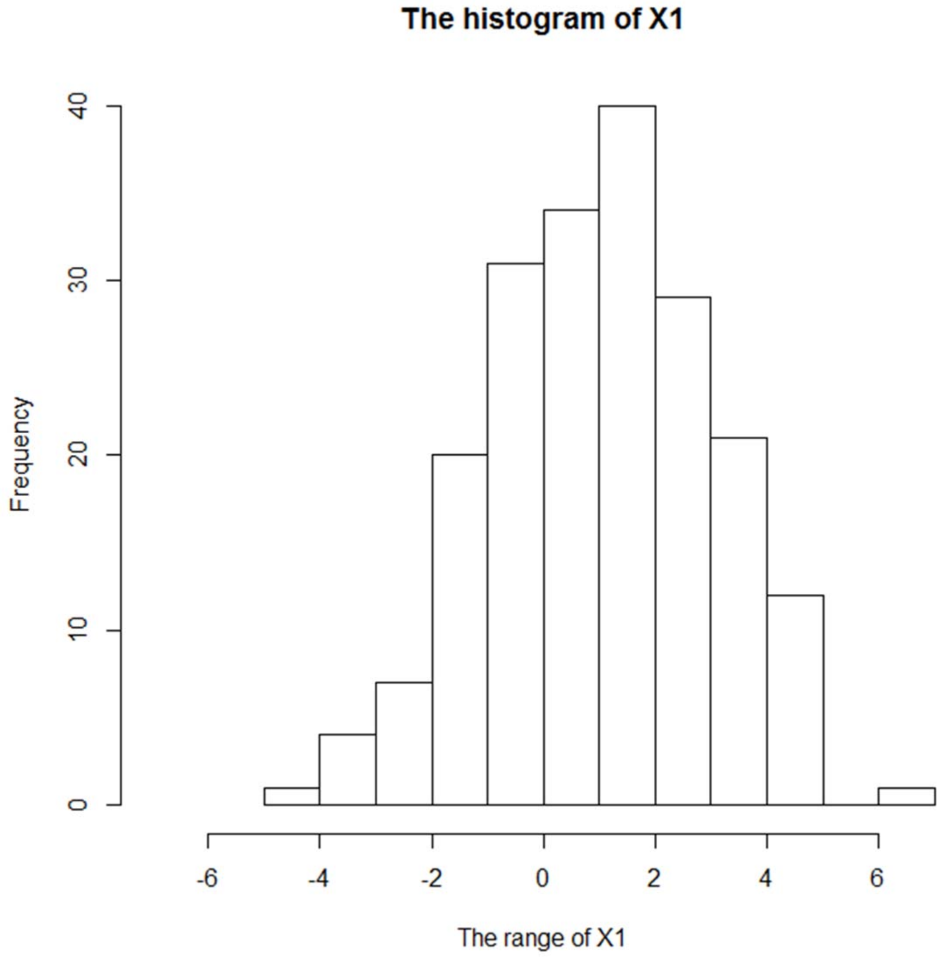
- Histograms -- hist(…)

- Scatter plot -- plot(…)

hist(x1)



```
hist(x1,xlim=c(-7,7),xlab="The range of
X1",main="The histogram of X1")
```



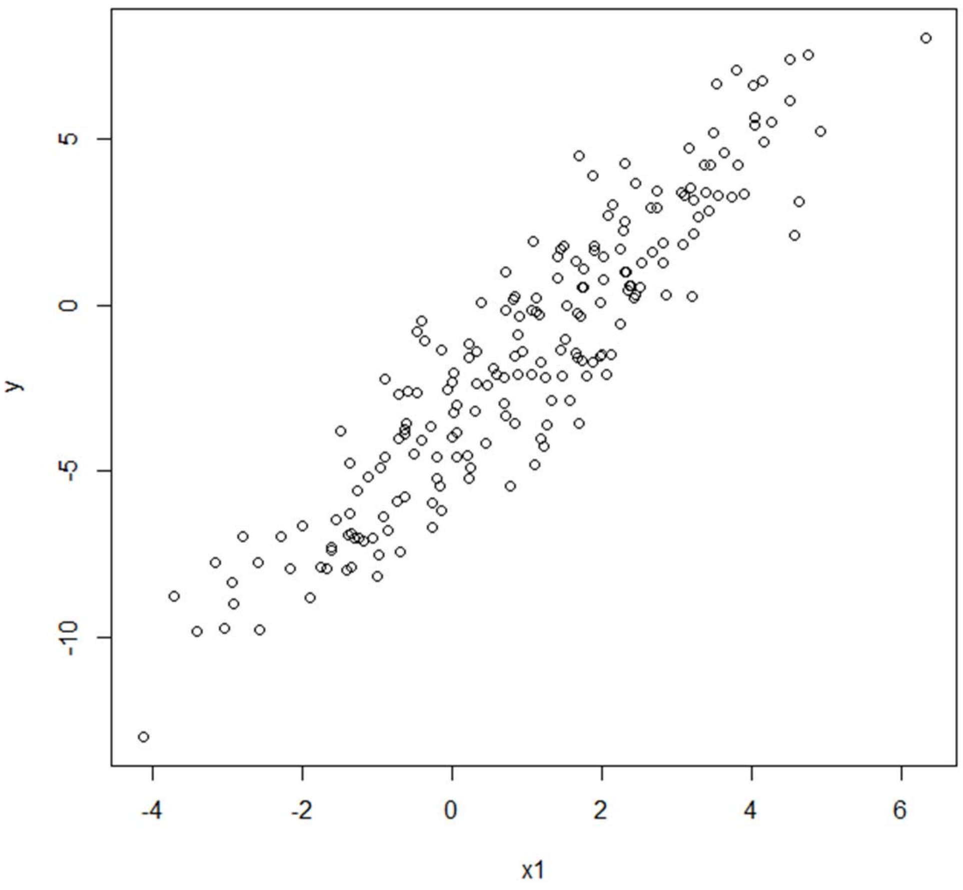
Common options

- **`xlim=c(a,b)`** or **`ylim=c(a,b`**)
  - Specify the range of x (y) axis [a,b]

- **`xlab=“ZZZ”`** or **`ylab=“ZZZ”`**
  - Specify the label attached to the x (y) axis as ZZZ

- **`main=“ZZZ”`**
  - Specify the title attached to the plot as ZZZ

plot(x1,y)



Common options

```
plot(x1,y,pch=2)
```

Specify the symbol for plotting points.

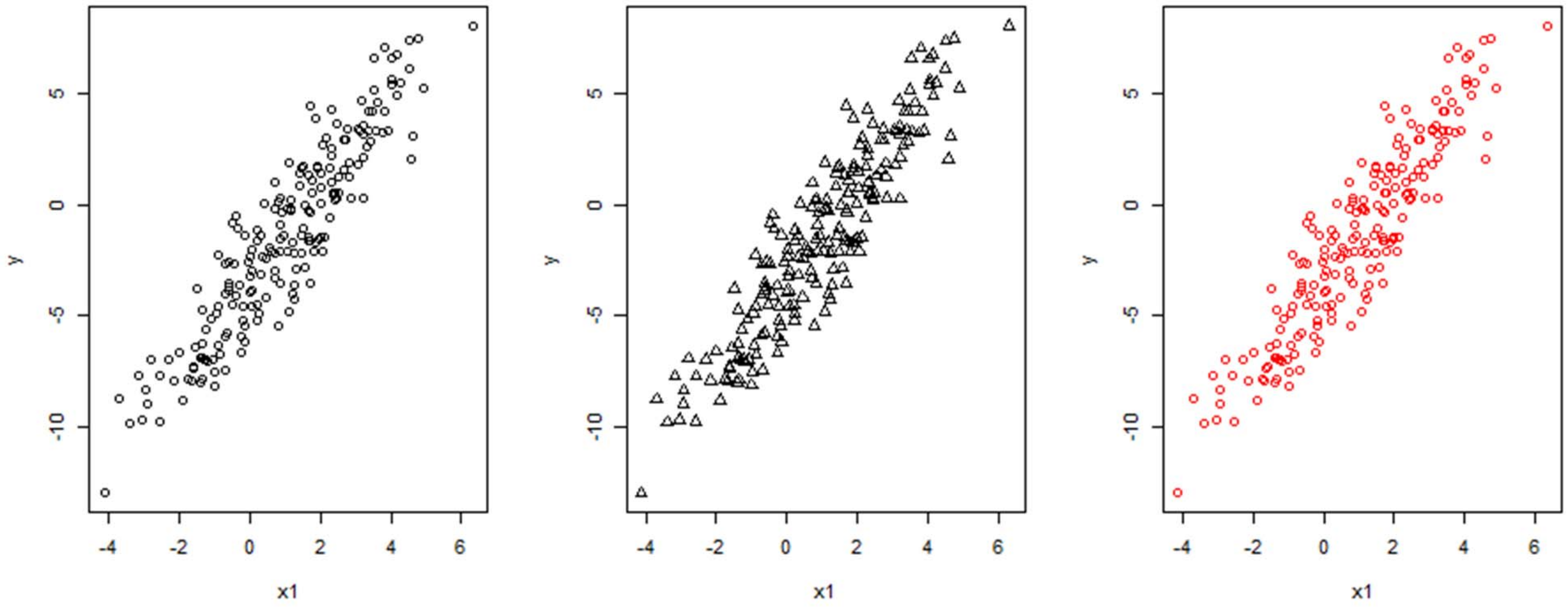
```
plot(x1,y,col="red")
```

Specify the color for plotting points.

Try the follow commands:

```
> par(mfrow=c(1,3))
> plot(y~x1)
> plot(y~x1,pch=2)
> plot(y~x1,col="red")
```

Multiple plots in one figure



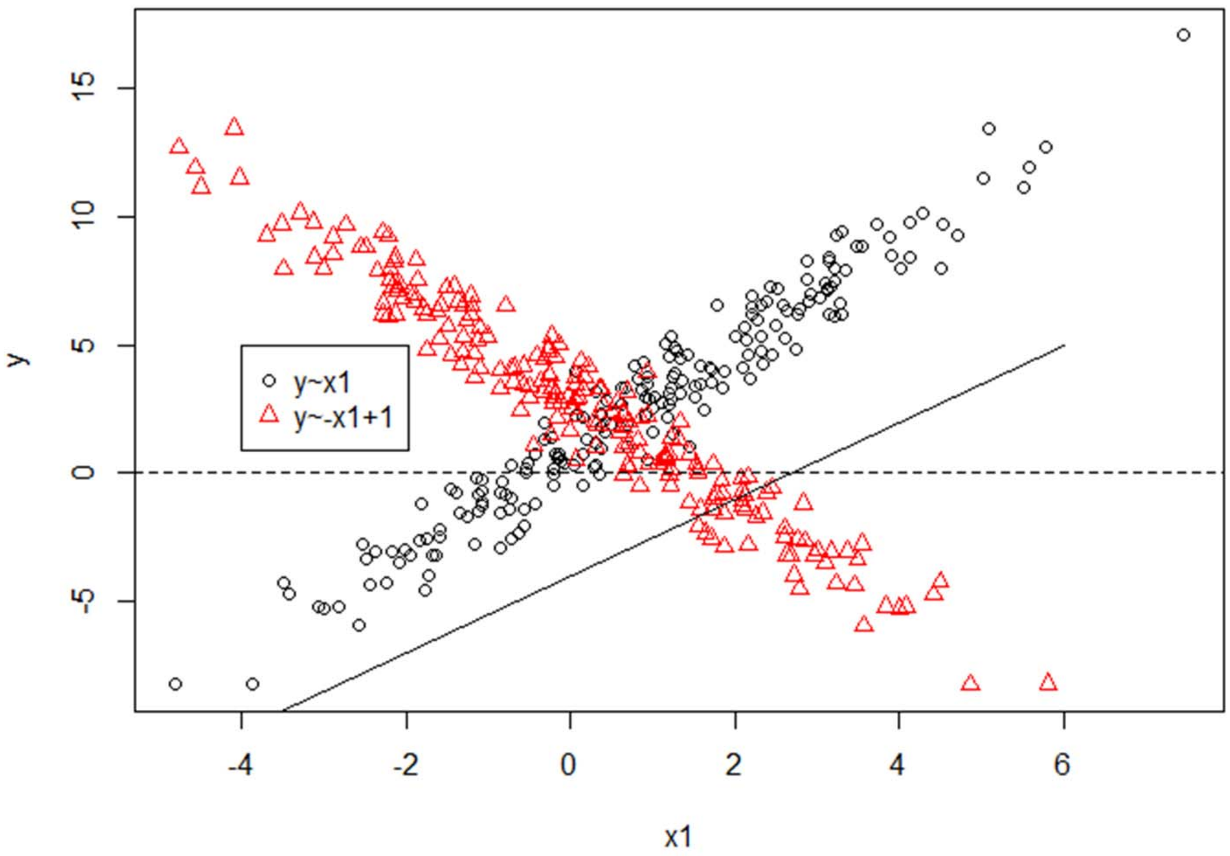
Add points or lines to an existing plot

```
plot(x1,y)
points(-x1+1, y, pch=2,col="red")
abline(h=0,lty=2)
lines(x=c(-4,6),y=c(-10,5))
legend(-4,5,legend=c("y~x1","y~x1+1"),pch=c(1,2),col=c("black","red"))
```

Tips and tricks:

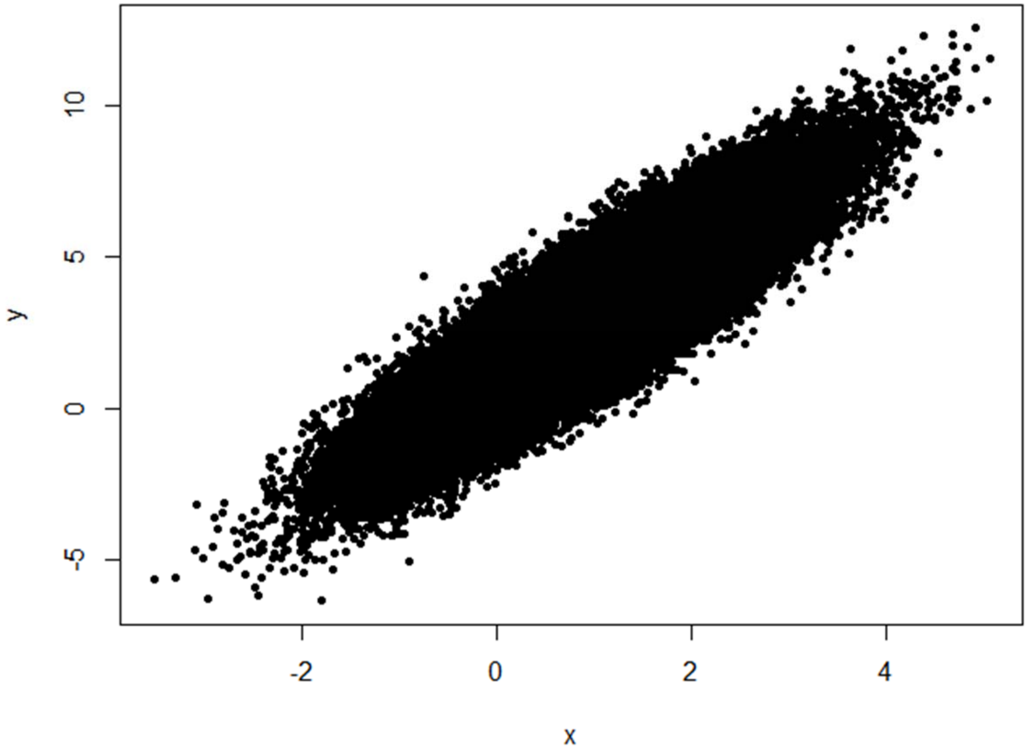
- The function points(…) adds points to an existing plot
- The functions abline(…) and lines(…) adds lines to an existing plot
- The function legend(…) attach a legend to an existing plot.

Overlay points

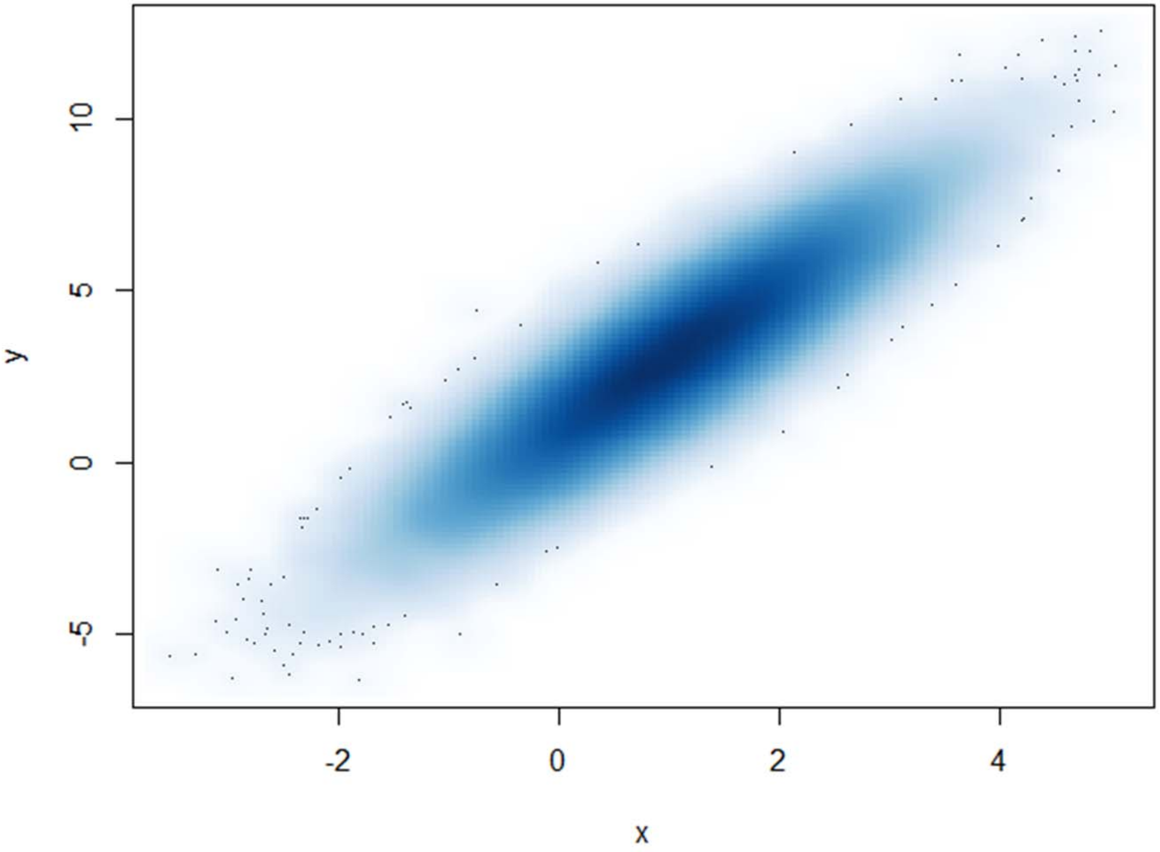


Plot many points?

```
x=rnorm(100000,1,1)
y=1+2*x+rnorm(100000)
plot(x,y,pch=20)
```



```
smoothScatter(x1,y)
# note the captial “S”
```



Linear regression

- General form
  `lm(response ~ var1+var2+…+vark, data=…)`

Note: “lm” stands for linear model

Example:

```
model<-lm(y~x1)
class(model)
```

Check the results of…

```
model
summary(model)
names(model)
model$residuals
model$fitted
coef(model)
vcov(model)
```

summary(model)

```
> summary(model)

Call:
lm(formula = y ~ x1)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.80526 -0.67816 -0.01294  0.72986  2.87535 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.92370    0.07852   11.76   <2e-16 ***
x1           2.02910    0.03495   58.06   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.019 on 198 degrees of freedom
Multiple R-squared:  0.9445,     Adjusted R-squared:  0.9442 
F-statistic:  3371 on 1 and 198 DF,  p-value: < 2.2e-16
```

names(model)

```
> names(model)
 [1] "coefficients"  "residuals"     "effects"
 [4] "rank"          "fitted.values" "assign"
 [7] "qr"            "df.residual"   "xlevels"
[10] "call"          "terms"         "model“
```

Note:

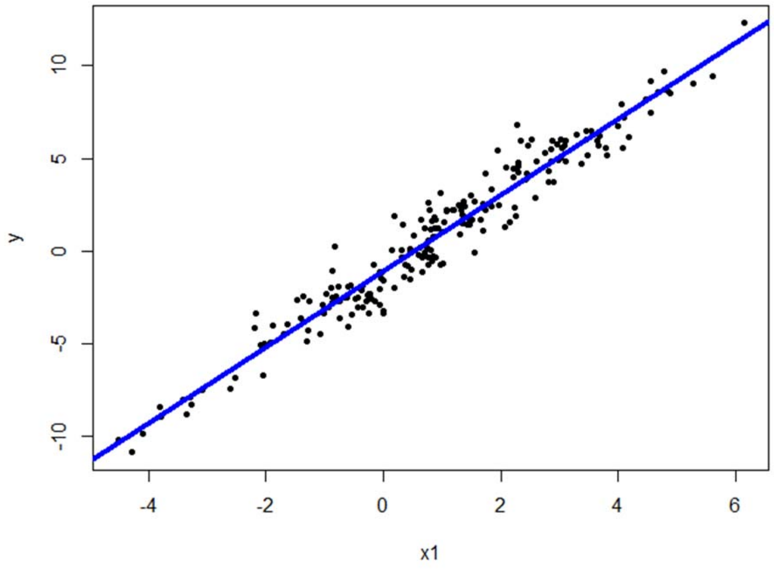
- “model” is an “lm” object.
- The command “names(model)” gives out the names of all elements in the “lm” object “model”.
- These elements can be called using their corresponding names. For example, model$residual, model$fitted

```
> coef(model)  # Extracts the coefficients
(Intercept)          x1          x2
  0.8236048   2.0009571  -1.4344263


> vcov(model) # Extracts the variance-covariance matrix
             (Intercept)            x1            x2
(Intercept)  0.052704697 -0.0012452737 -0.0151757639
x1          -0.001245274  0.0016738863 -0.0001222336
x2          -0.015175764 -0.0001222336  0.0050466167
```

Plot linear regression

```
x1=rnorm(200,1,2)
y=-1+2*x1+rnorm(200)
plot(x1,y,pch=20)
model=lm(y~x1)
abline(a=model$coef[1], b=model$coef[2], lwd=4, col="blue")
```


10
5
0
-5
-10
y
-4
-2
0
2
4
6
x1