Ejercicios de repaso

Ejercicio 1

```
set.seed(1)
n <- 50
y <- rchisq(n, 30)
x <- rnorm(n, 30, 5)
```

1. media, desviación típica y 3 cuartiles de y:

```
mean(y)
```

```
## [1] 29.39873
```

```
sd(y)
```

```
## [1] 5.872935
```

```
quantile(y)
```

```
##       0%      25%      50%      75%     100%
## 15.82831 25.51818 28.78737 33.84948 40.01163
```

histograma

```
hist(y, freq=FALSE, ylim=c(0, 0.1))
curve(dnorm(x,mean=mean(y), sd=sd(y)), col="red", add=TRUE)
```

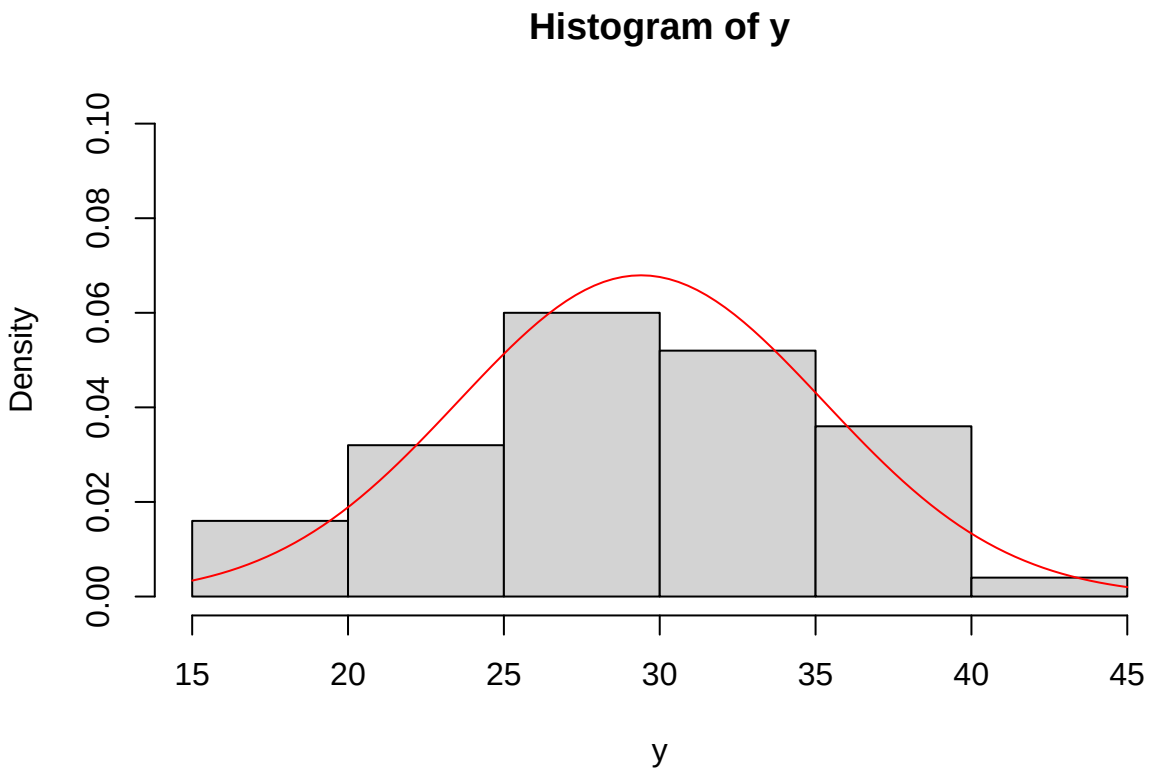


gráfico de probabilidad de los valores de y

```
qqnorm(y)
```

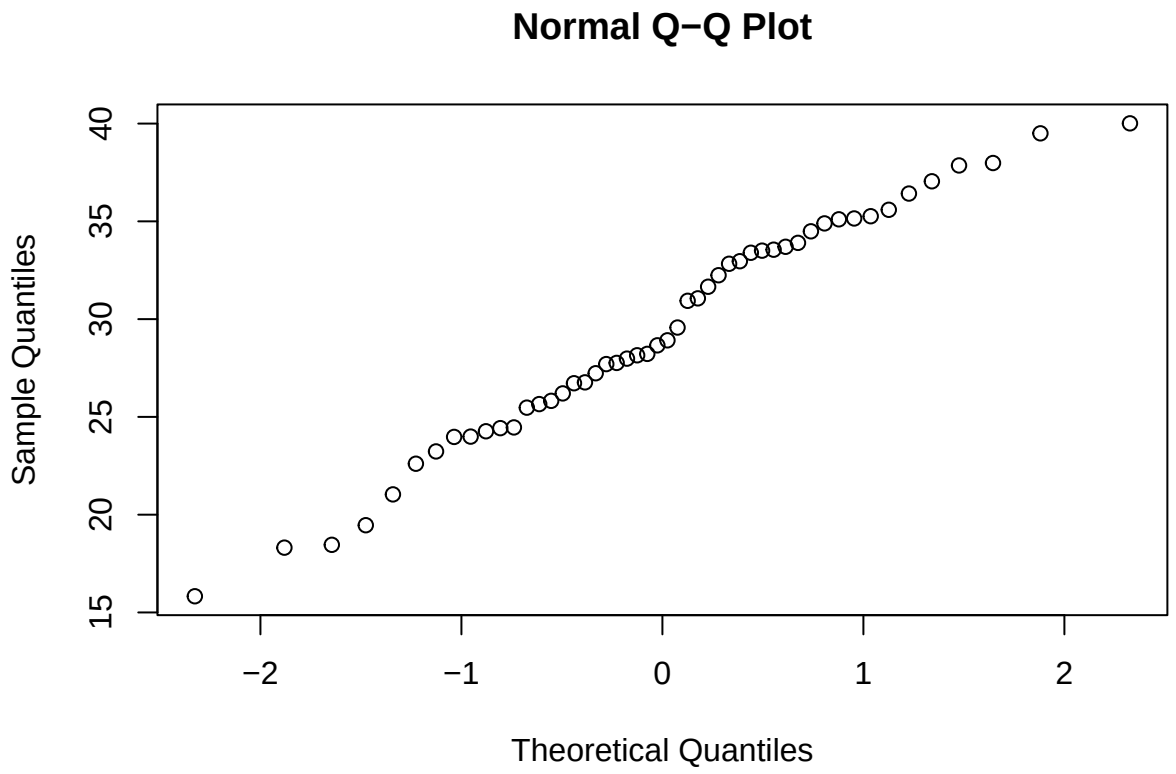


Gráfico de cajas de los valores de y

```
boxplot(y)
```

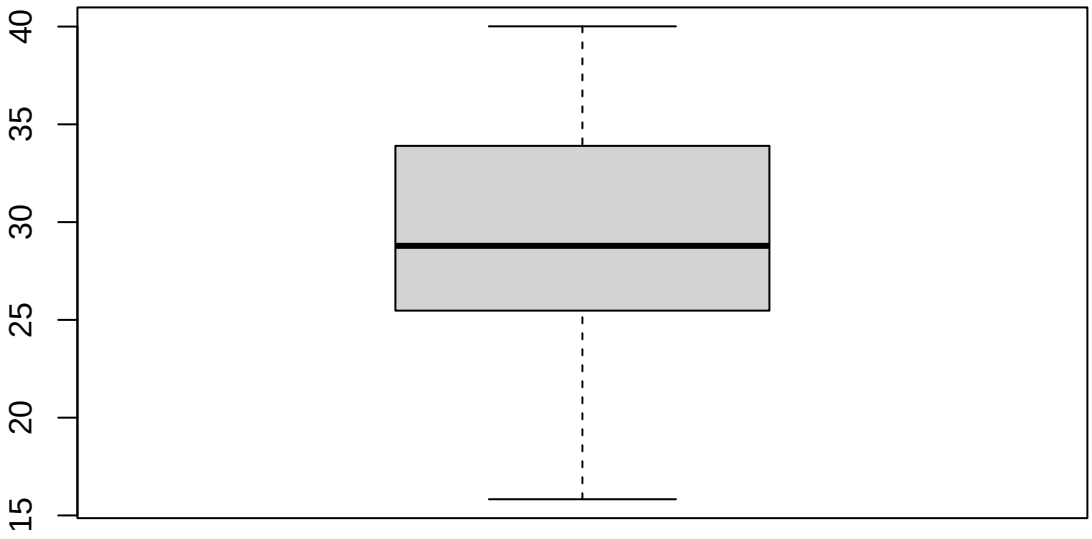


Gráfico de cajas múltiples para comparar distribución de x e y

```
boxplot(x,y, names=c("x", "y"))
```

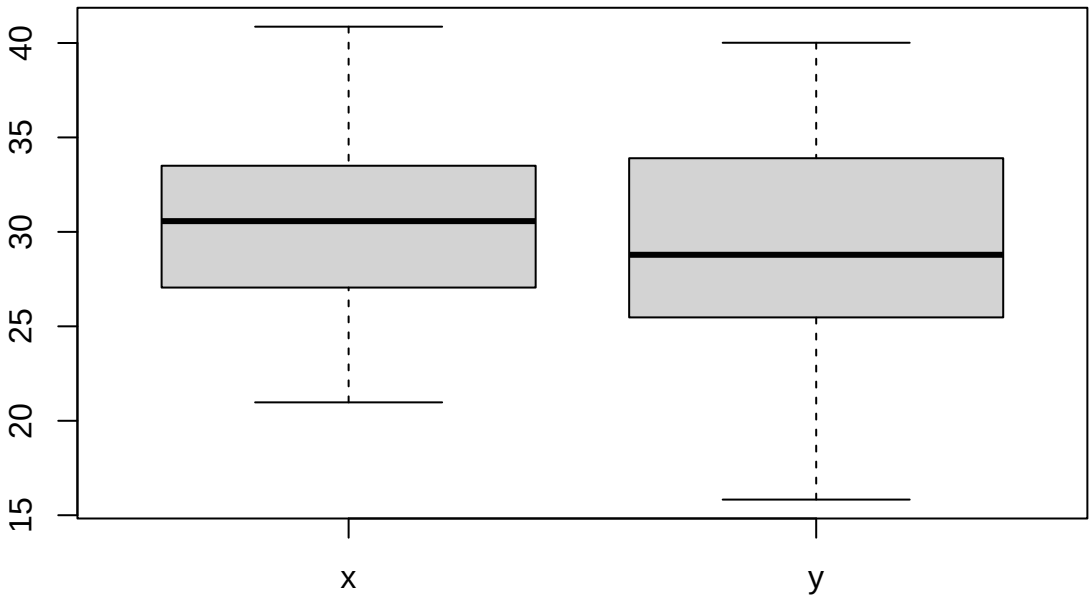
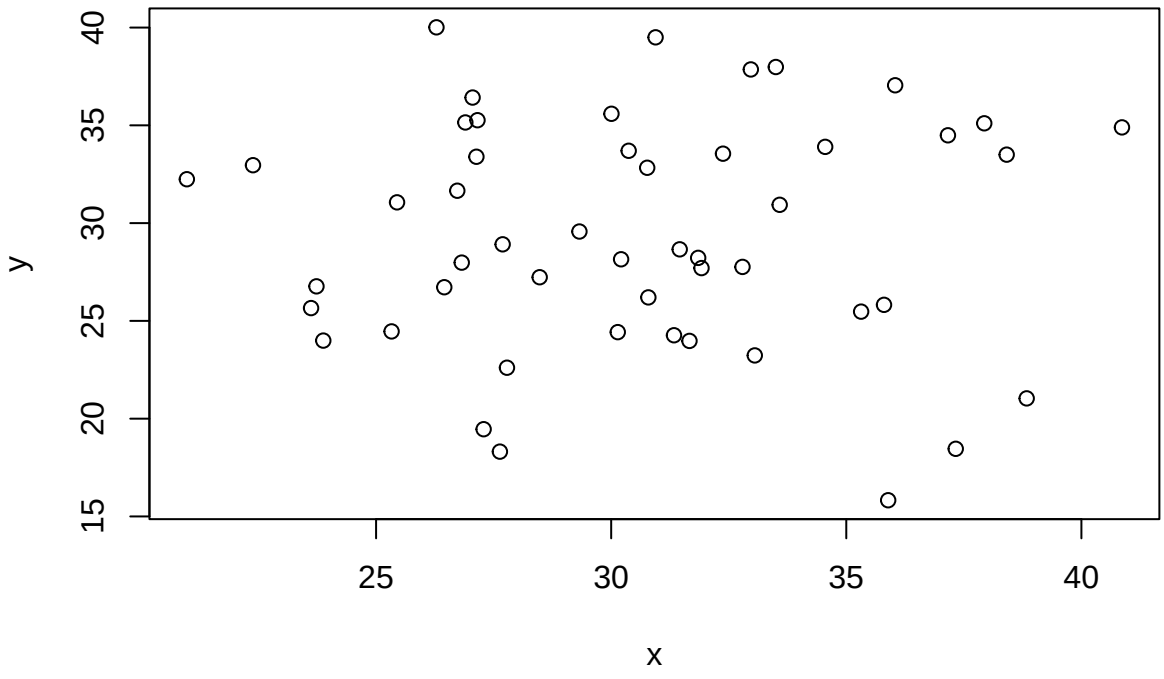


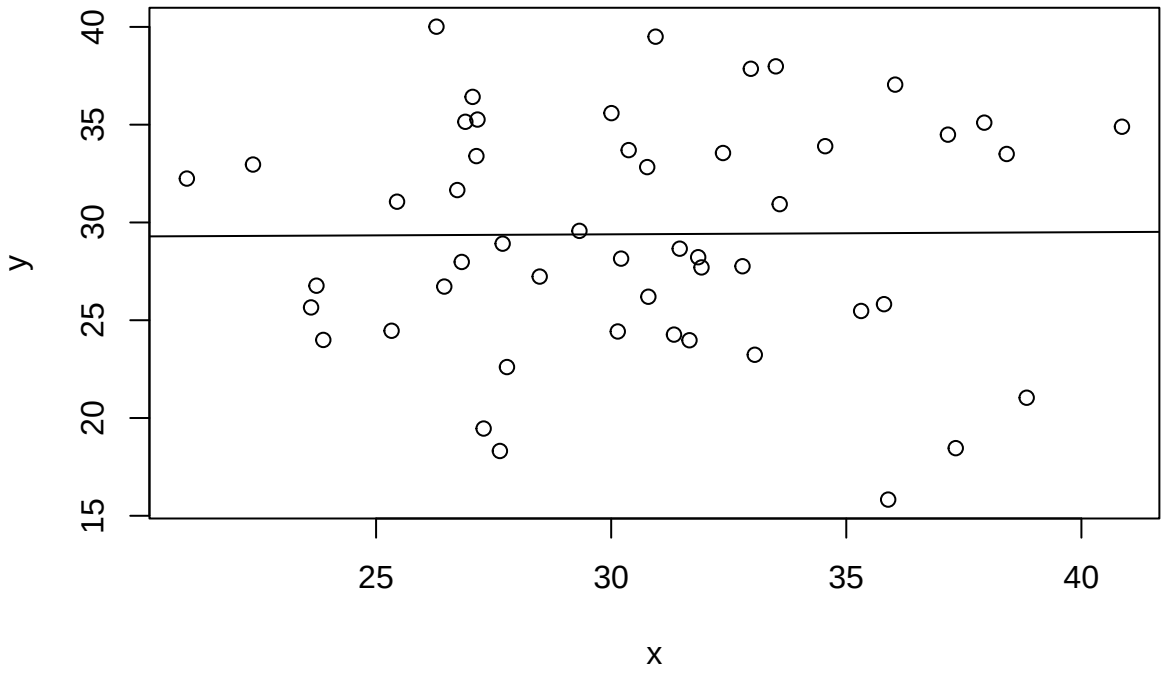
Diagrama de dispersión de valores de x frente a los de y

```
plot(x,y)
```



Regresion lineal

```
fit <- lm(y~x)
plot(x,y)
abline(fit)
```



Ejercicio 2

```
tirardados <- function(n=4){
  tiradas <- sample(1:6, n, replace=TRUE)
  res <- FALSE

  if (sum(tiradas==6) > 0)
    res <- TRUE
  return(res)

}

tirardados(10)
```

```
## [1] TRUE
```

```
nsim <- 10000

lanzamientos <- numeric(length = nsim)
for (i in 1:nsim){
  lanzamientos[i] <- tirardados()
}

print(mean(lanzamientos))
```

```
## [1] 0.5159
```

```
print(1-(5/6)^4)
```

```
## [1] 0.5177469
```

Ejercicio 3

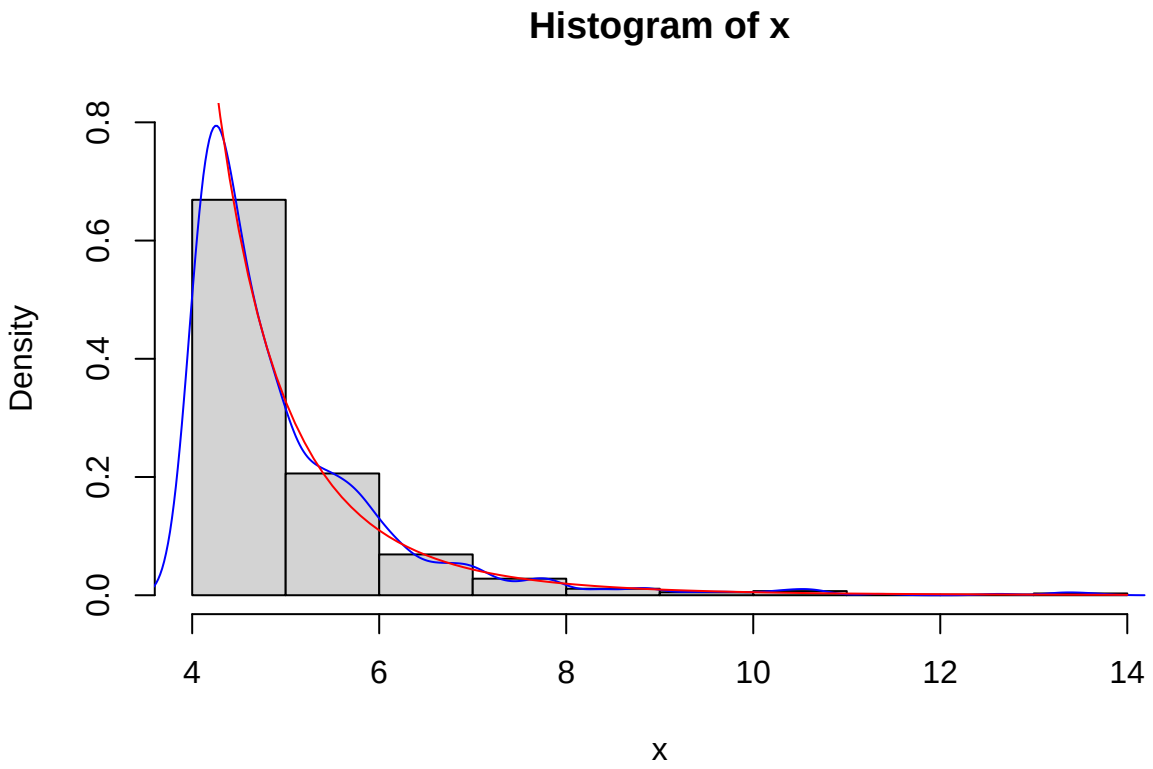
```
nsim <- 1000
a <- 5
b <- 4

u <- runif(nsim)
x <- b/( (1-u)^(1/a) )

pareto <- function(x,a,b){
  return((a*b^a)/(x^(a+1)))
}

paretodist <- function(x,a,b){
    return((1-(b/x)^a)*(x >= b))
}

hist(x, freq=FALSE, ylim=c(0,0.8))
lines(density(x), col="blue")
curve(pareto(x,a,b), col="red", add=TRUE)
```




```
ks.test(x,paretodist, a=a,b=b)
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  x
## D = 0.017746, p-value = 0.9112
## alternative hypothesis: two-sided
```

Ejercicio 4

```
nsim <- 1000

f <- function(x){
  res <- (1/(1+x^2)) *(x >= 0 & x <= 1)
  return(res)
}

x <- runif(nsim)

fx <- sapply(x, f)

mean(fx)
```

```
## [1] 0.7786068
```

```
pi/4
```

```
## [1] 0.7853982
```

```
estim <- cumsum(fx)/1:nsim

estim.err <- sqrt(cumsum((fx-estim)^2))/1:nsim

plot(1:nsim,estim,type='l',ylab='Aproximación y límites de error',
xlab='Número de simulaciones')
z<-qnorm(0.025,lower.tail = FALSE)
lines(estim - z*estim.err,col='blue',lwd=2,lty=3)
lines(estim + z*estim.err,col='blue',lwd=2,lty=3)
abline(h=pi/4,col=2)
```