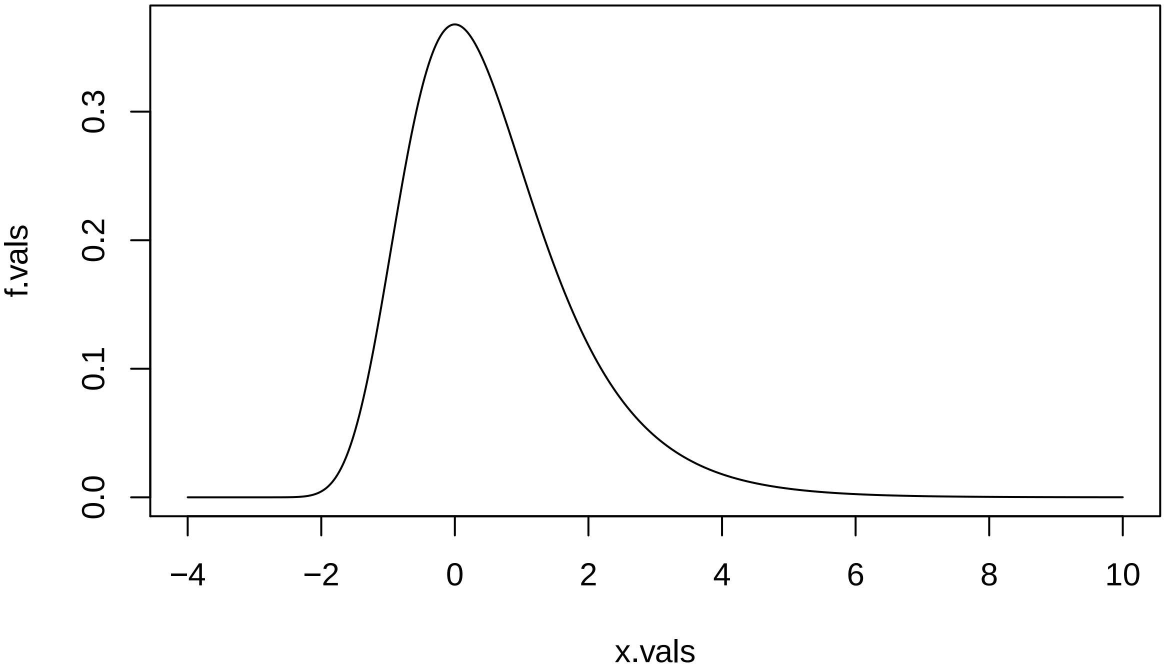
hw2

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

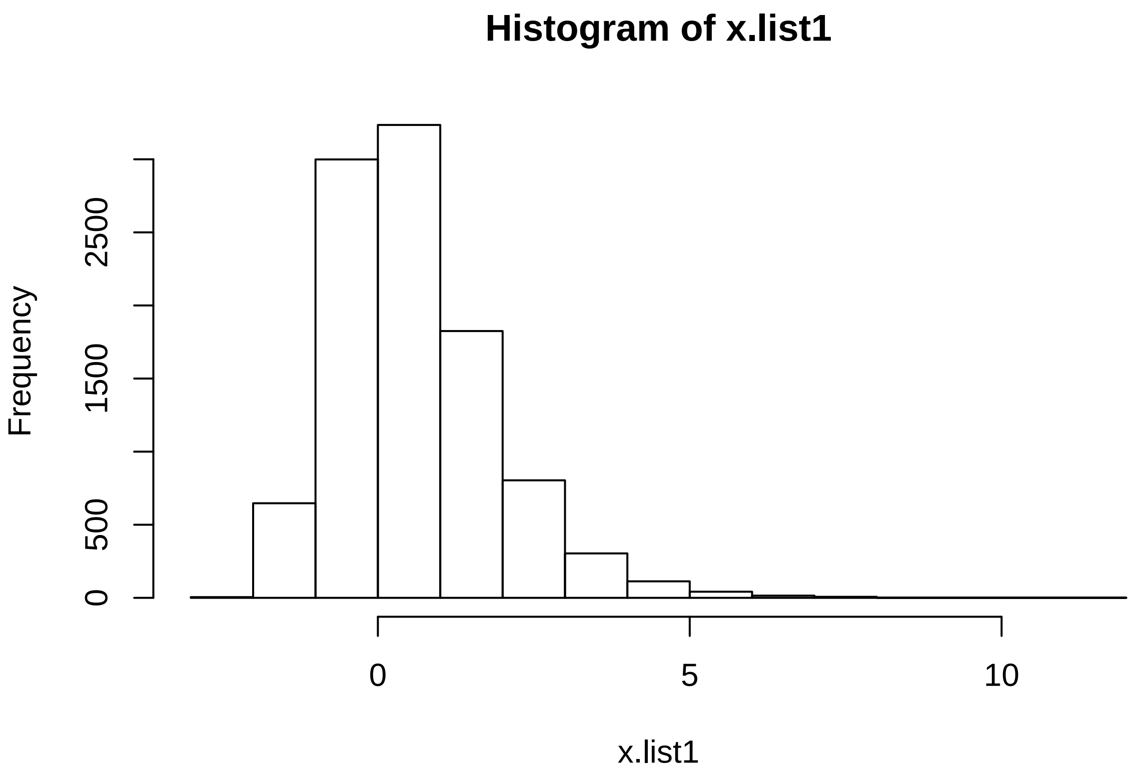
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```r
#1a
set.seed(3301)
x.vals = seq(from=-4, to = 10, length.out= 1e4)
f.vals = exp(-x.vals - exp(-x.vals))
plot(x.vals, f.vals, type="l")
```

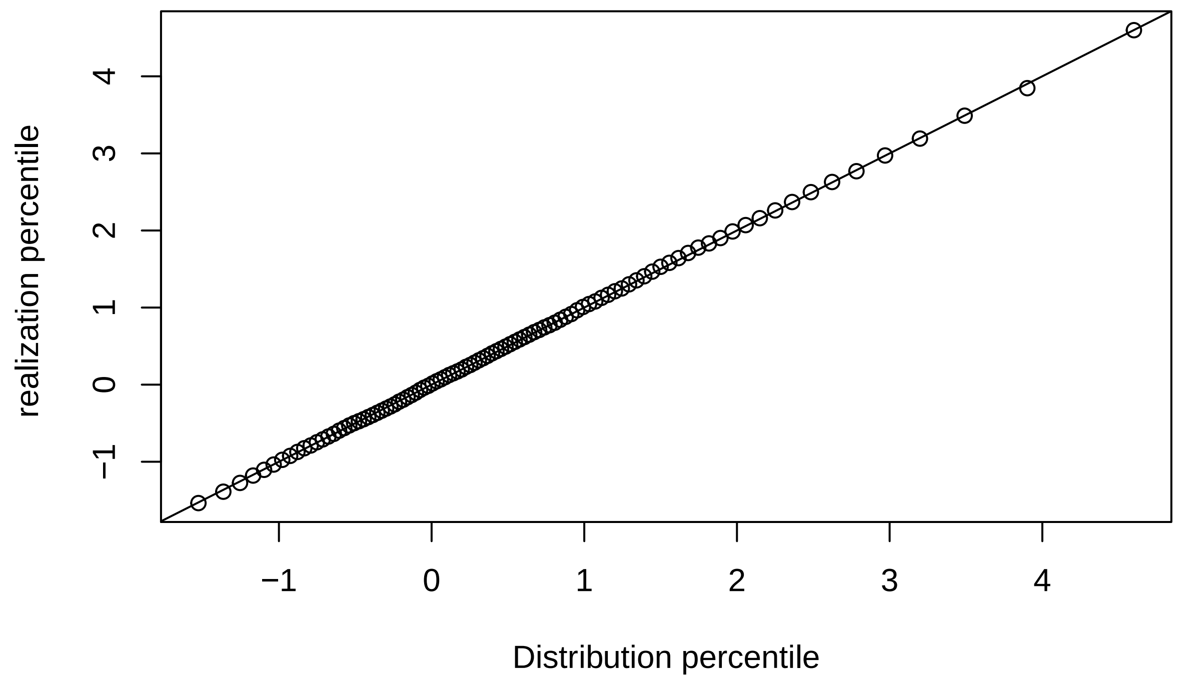


```r
#1c
set.seed(3301)
rgumbel = function(n){
  x.list = numeric(n)
  for(i in 1:n){
    x.vals1 = runif(1)
    x.list[i] = -log(-log(x.vals1)) # ith realization
  }
  return(x.list)
}
```

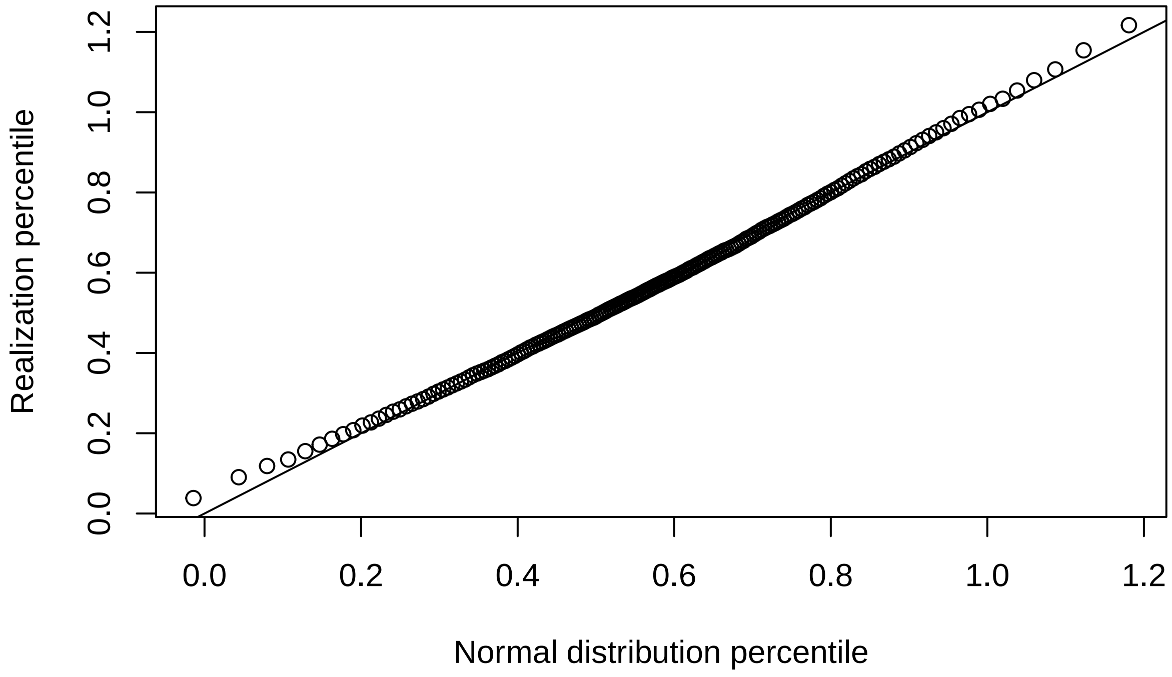
```
#1d
set.seed(3301)
n= 1e4
x.list1 = rgumbel(n)
hist(x.list1)
```



```
probs = seq(from =.01, to=.99, by=.01)
plot(-log(-log(probs)), quantile(x.list1, probs), xlab="Distribution percentile", ylab="realization perc
abline(0,1)
```



```
#1e
set.seed(3301)
reps =1e4
n =30
x.list2 = numeric(reps)
for(i in 1:reps){
  n.list = rgumbel(n)
  x.list2[i] = sum(n.list)/n # equivalent to Xbar
}
probs = seq(from=.005, to = .995, by =.005 )
plot(qnorm(probs, mean = mean(x.list2), sd= sd(x.list2)), quantile(x.list2, probs), xlab="Normal distri
abline(0,1)
```



```
#2a
set.seed(3301)
x.list3 = numeric(10)
theta =4
n =10
reps = 1e4
sf.pt = theta * .75^(1/n)
sf.pt
```

```
## [1] 3.886567
```

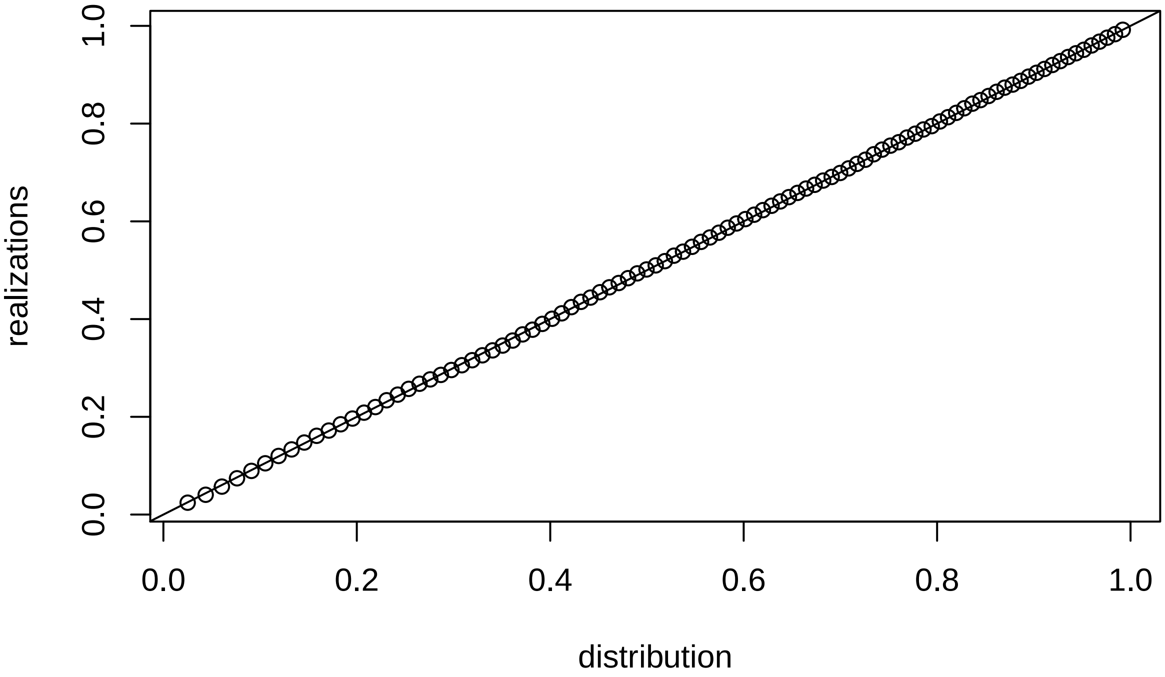
```
#2b
set.seed(3301)
theta = 4; a = 0.05; n =10; reps1=1e4
captured.list = numeric(reps)

for(i in 1:reps1){
  un.list= runif(n=n, min=0, max = theta)
  m1.list = max(un.list)
  left.pt = m1.list  #lower bound
  right.pt = (m1.list)/a^(1/n)  # upperbound
  captured.list[i] = 1*(left.pt <= theta)*(theta <= right.pt )
}
mean(captured.list)
```

```
## [1] 0.9534
```

```
set.seed(3301)
#3b
rfincrease = function(n, theta){
  x.list5 = numeric(n)
  for( i in 1:n ){
    u=runif(1) # u interval [0,1]
    x.list5[i] = (u)^(1/(theta+1)) #realization
  }
  return(x.list5)
}
```

```
#3c
set.seed(3301)
thet= 1/4
xreal.list= rfincrease(1e4, thet)
probs = seq(from=.01, to = .99, by =.01)
plot(probs^(1/(thet+1)), quantile(xreal.list, probs),xlab="distribution", ylab="realizations")
abline(0,1)
```

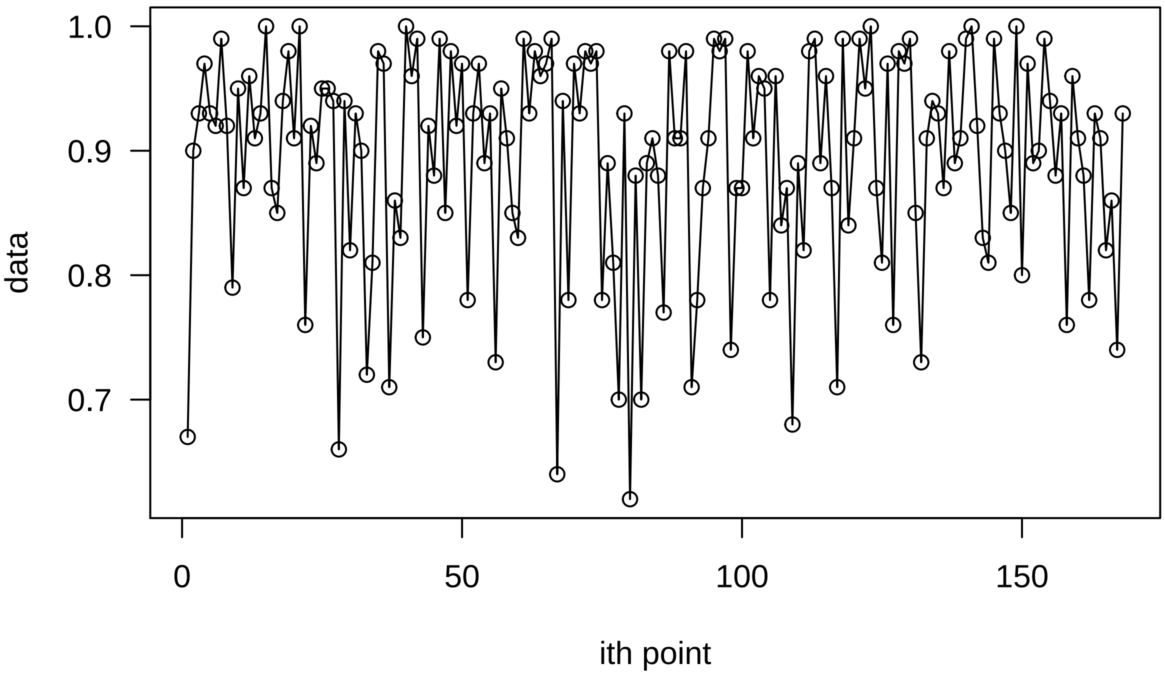


```
 #3d
set.seed(3301)
theta = 1/4
a=.05
n=100
reps=1e4
captured.list1 = numeric(reps)
for(i in 1:reps){
  n.list1= rfincrease(n, theta) #realizations
  lx.list= numeric(n) # list for the sum of log realizations
  for(j in 1:n){
    lx.list[j] = log(n.list1[j])
  }
  est1= -1-(n/sum(lx.list)) #estimator
  left.pt1= est1-qnorm(1-a/2)*(est1+1)/sqrt(n) #lowerbound
  right.pt1= est1+qnorm(1-a/2)*(est1+1)/sqrt(n) # upperbound
  captured.list1[i]= 1*(left.pt1<=theta)*(theta<=right.pt1) #check theta within the lower and upper bou
}

mean(captured.list1)
```

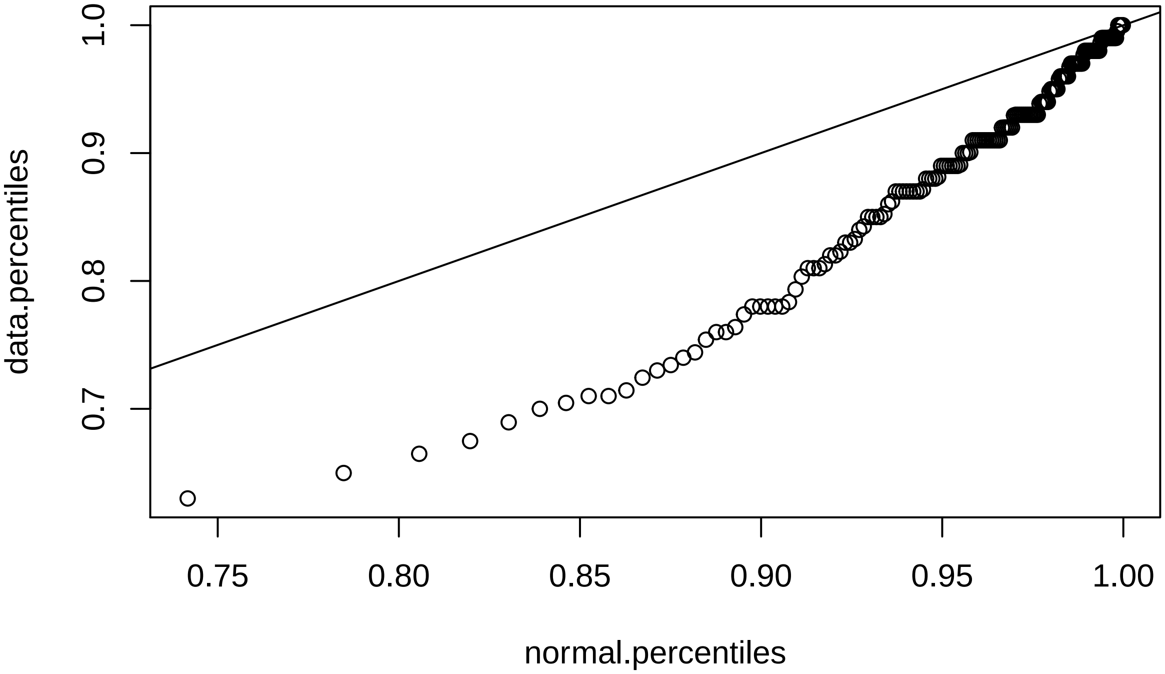
```
## [1] 0.9523
```

```
#3e
dat = read.table("hw2-data.txt")
x.list8 = dat$x  # data from the datatable
num <- 1:168
names(x.list8) = dat$num
plot(dat$x, dat$num, type ="o", las=1, xlab="ith point", ylab="data") # plotting time
```



```
m =mean(x.list8) #x.list8 is the data points
s= sd(x.list8)
set.seed(3301)
n = 168
lx1.list = numeric(168)

for(j in 1:168){
  lx1.list[j] = log10(x.list8[j])
}
est2 = -1-(n/sum(lx1.list)) # estimator of theta
real.list = rfincrease(n, est2) #realization with the estimator from the data
probs = ppoints(n)
data.percentiles =  quantile(x.list8, probs) #data points quantile
normal.percentiles = probs^(1/(est2+1))  #used the estimator derived from the dataset as theta and plug
plot(normal.percentiles, data.percentiles)  # based on the plot the data doesn't fit the given distribu
abline(0,1)
```


data.percentiles
0.7
0.8
0.9
1.0
0.75
0.80
0.85
0.90
0.95
1.00
normal.percentiles