Value at risk for discrete random variables

Dana Uhlig

Tue Jun 26 13:27:23 2018

```
#discrete random variable with 5 states

#Example 6.3 (lecture notes)
y = c(3,7,-3,8,-5)
#P(Y=y)=p, zable 6.1 (a)
p = c(1/5,1/5,1/5,1/5,1/5)
names(p) = sort(y)
p
```

```
##  -5  -3   3   7   8
## 0.2 0.2 0.2 0.2 0.2
```

```
sum(p)
```

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

```
#quantiles
#alpha = c(0.05, 0.25, 0.75, 0.95)
(alpha = seq(from=1/5,to = 1, by = 1/5))
```

```
## [1] 0.2 0.4 0.6 0.8 1.0
```

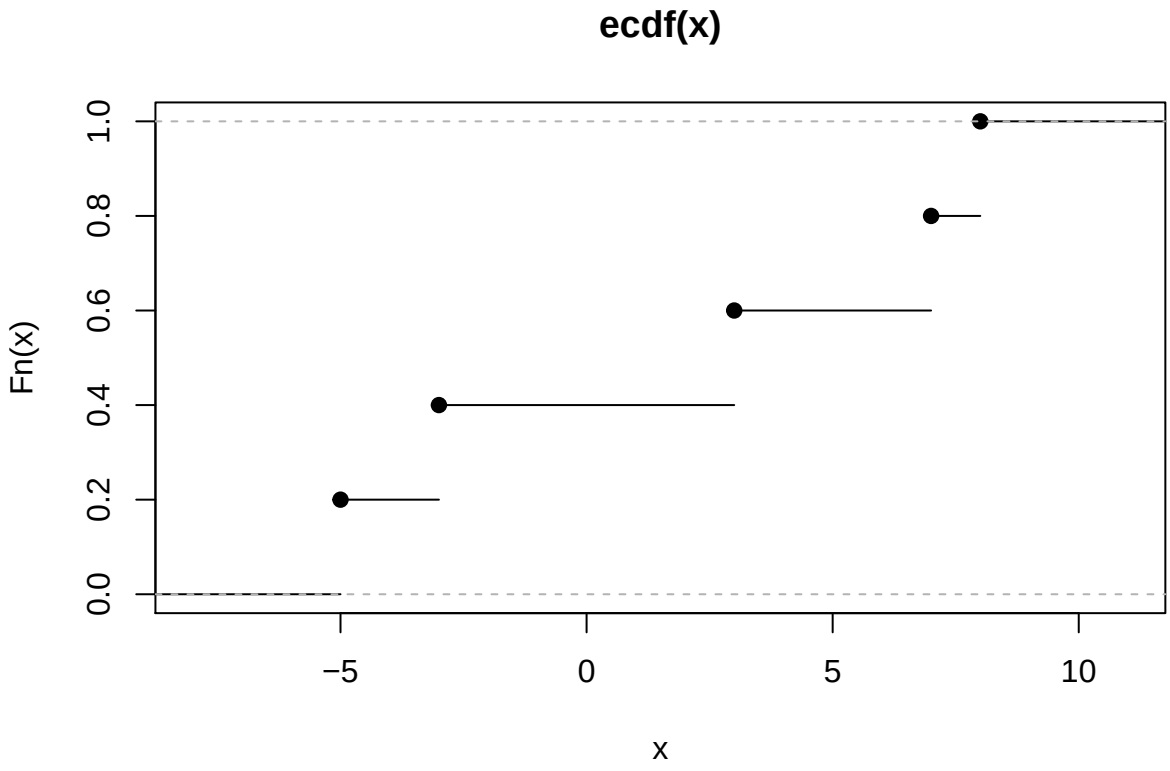
```
'q.alpha = inverse(F)(alpha)'
```

```
## [1] "q.alpha = inverse(F)(alpha)"
```

```
#table 6.1 (c)
(q.alpha = quantile(y, probs = alpha, type = 1))
```

```
##  20%  40%  60%  80% 100%
##   -5   -3    7    7    8
```

```
#distribution function
plot.ecdf(y)
```




```
p
```

```
##  -5  -3   3   7   8
## 0.2 0.2 0.2 0.2 0.2
```

```
cumsum(p)
```

```
##  -5  -3   3   7   8
## 0.2 0.4 0.6 0.8 1.0
```

```
y.dist =ecdf(y)
#curve(y.dist(x),from = min(y)-0.5, to=max(y)+0.5)
'F(q.alpha) = F(inverse(F)(alpha))=alpha'
```

```
## [1] "F(q.alpha) = F(inverse(F)(alpha))=alpha"
```

```
y.dist(q.alpha)
```

```
## [1] 0.2 0.4 0.8 0.8 1.0
```

```
alpha
```

```
## [1] 0.2 0.4 0.6 0.8 1.0
```

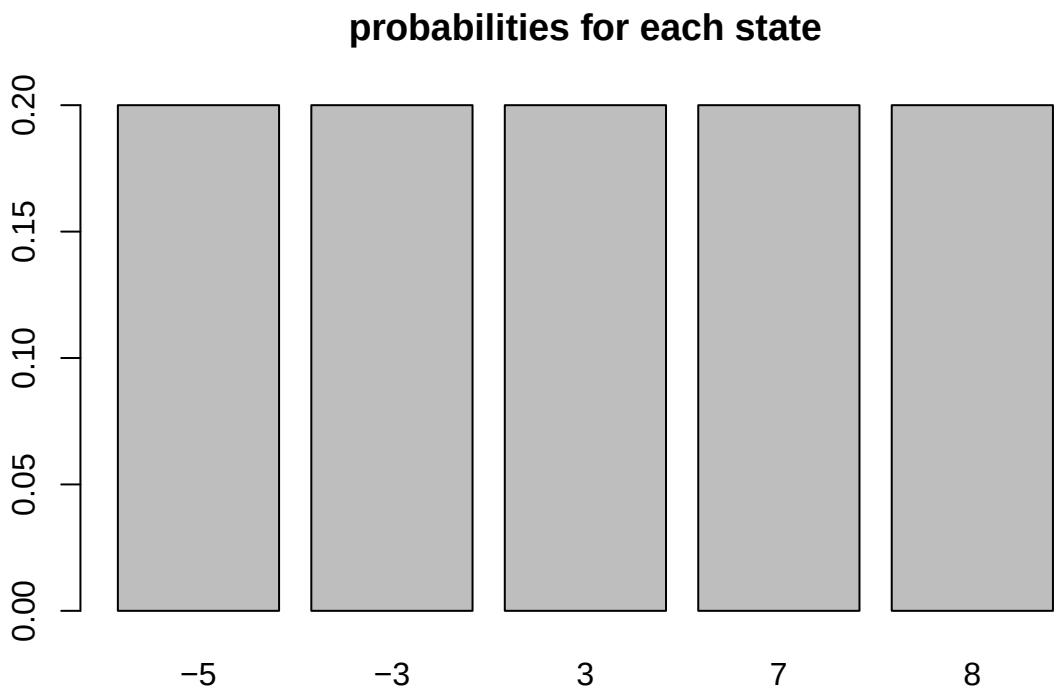
```
'table 6.1 (b)'
```

```
## [1] "table 6.1 (b)"
```

```
y.dist(sort(y))
```

```
## [1] 0.2 0.4 0.6 0.8 1.0
```

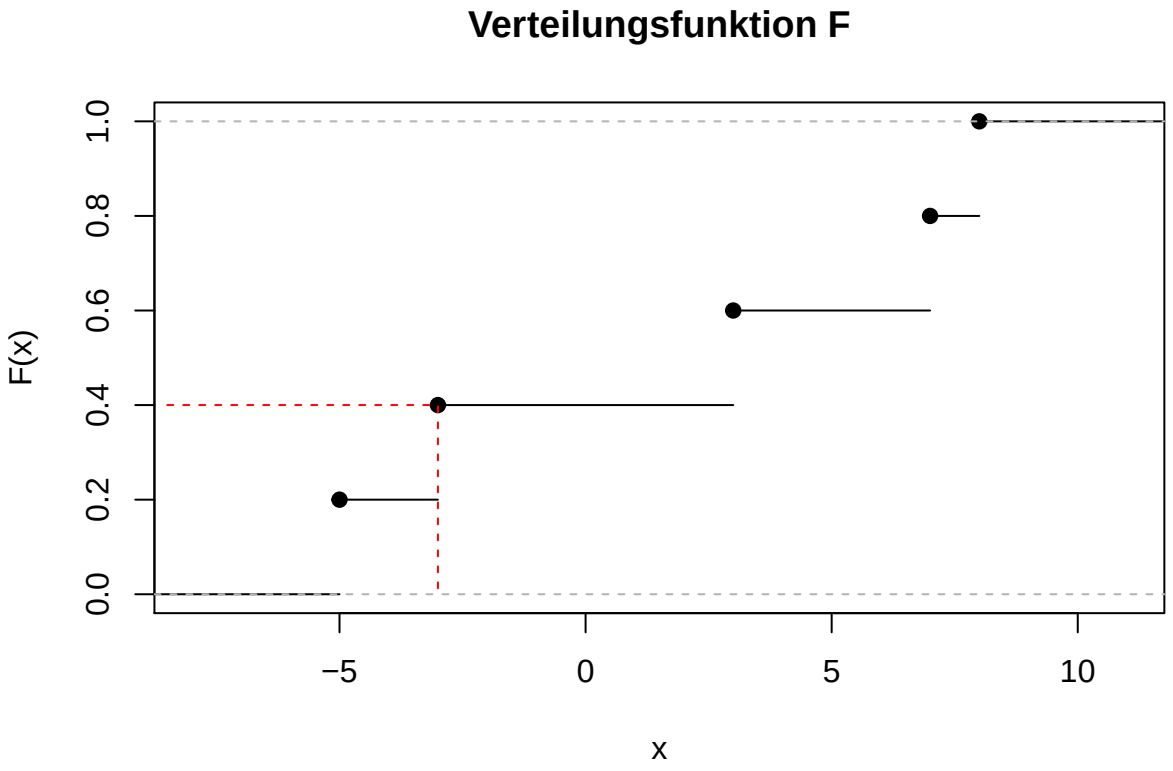
```r
#graphical visualization
#density / probabilities
barplot(table(y), height = p, names.arg = sort(y), main = "probabilities for each state")
```



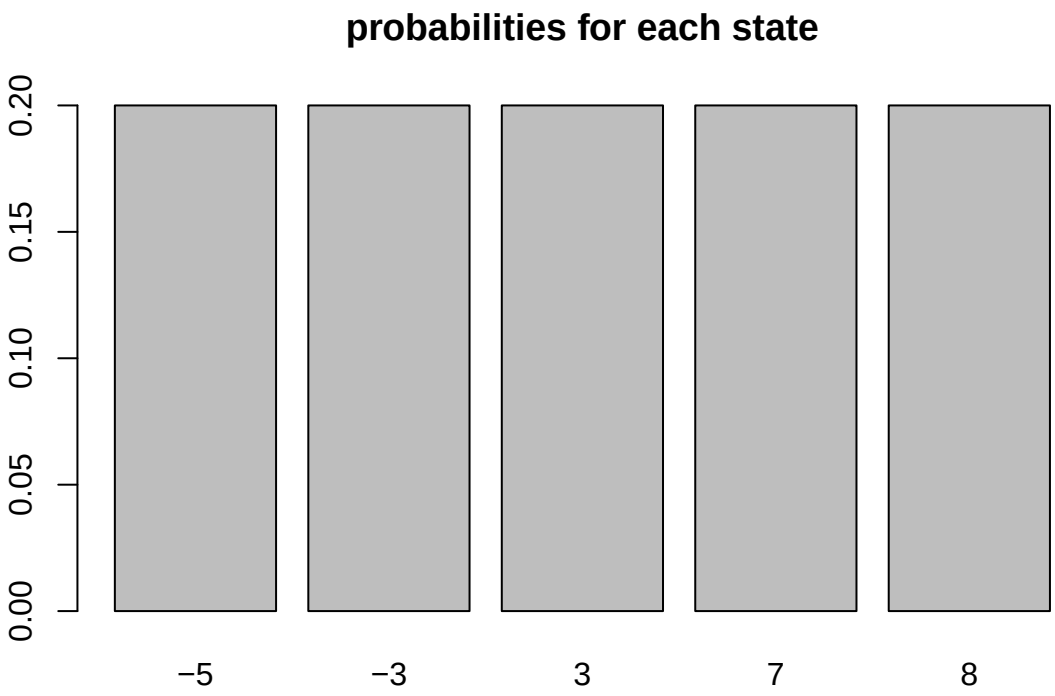

```r
'we want plot the k-th (1,..,5) alpha and the corresponding quantile'
```

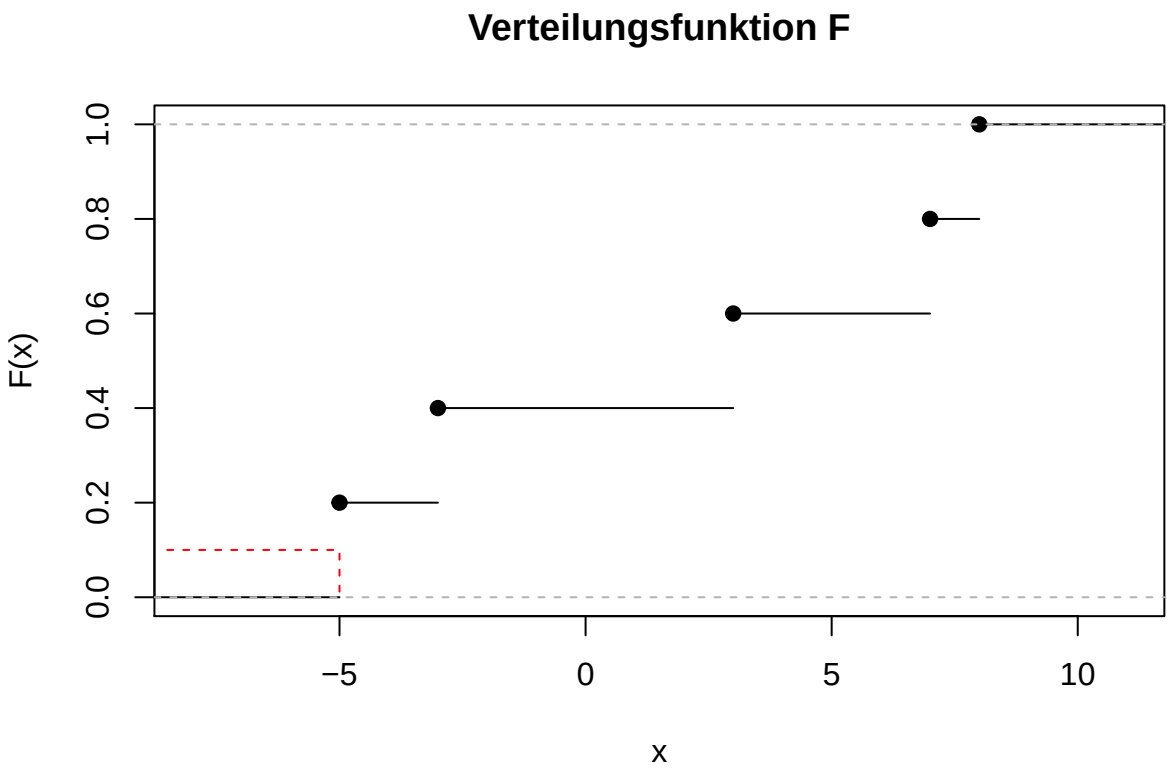
```
## [1] "we want plot the k-th (1,..,5) alpha and the corresponding quantile"
```

```r
k = 2
#distribution
plot.ecdf(y, main="Verteilungsfunktion F", ylab = "F(x)")
#lines(c(0,q.alpha[2],q.alpha[2]),c(alpha[2],alpha[2],0),lty=2,col="red")
lines(c(min(y)-3.5,q.alpha[k],q.alpha[k]),c(alpha[k],alpha[k],0),lty=2,col="red")
```



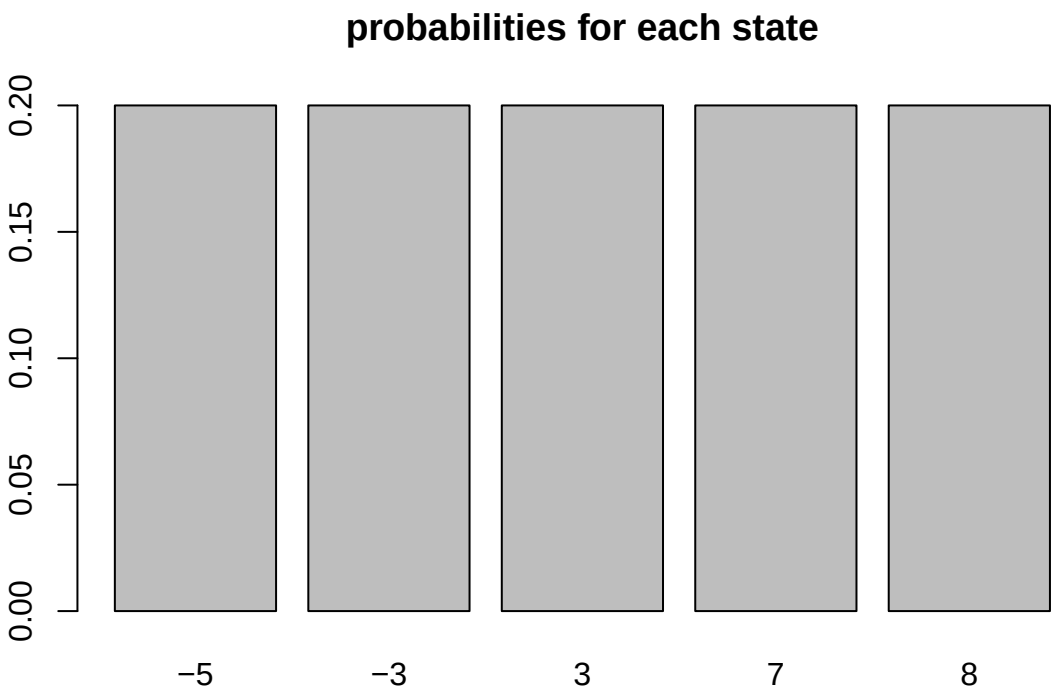

```
################################################################################
#writing a own function doing this job

value.at.risk = function(alpha = 0.05, discrete.values, prob){
  y = discrete.values
  q.alpha = quantile(y, probs = alpha, type = 1)
  y.dist =ecdf(y)

  g1 = barplot(table(y), height = p, names.arg = sort(y), main = "probabilities for each state")

  g2 = plot.ecdf(y, main="Verteilungsfunktion F", ylab = "F(x)")
  lines(c(min(y)-3.5,q.alpha,q.alpha),c(alpha,alpha,0),lty=2,col="red")
  return(q.alpha)
}


value.at.risk(alpha = 0.1, y, p)
```

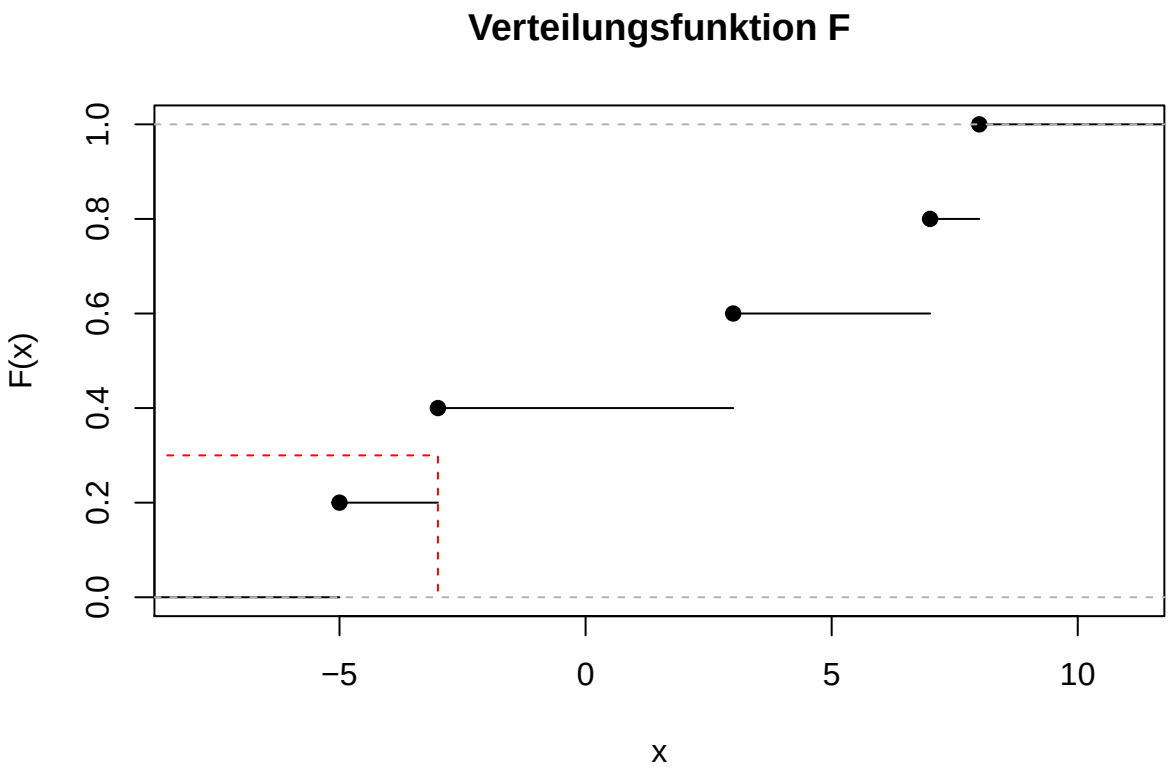

probabilities for each state
0.00
0.05
0.10
0.15
0.20
−5
−3
3
7
8




```
## 10%
##  -5
```

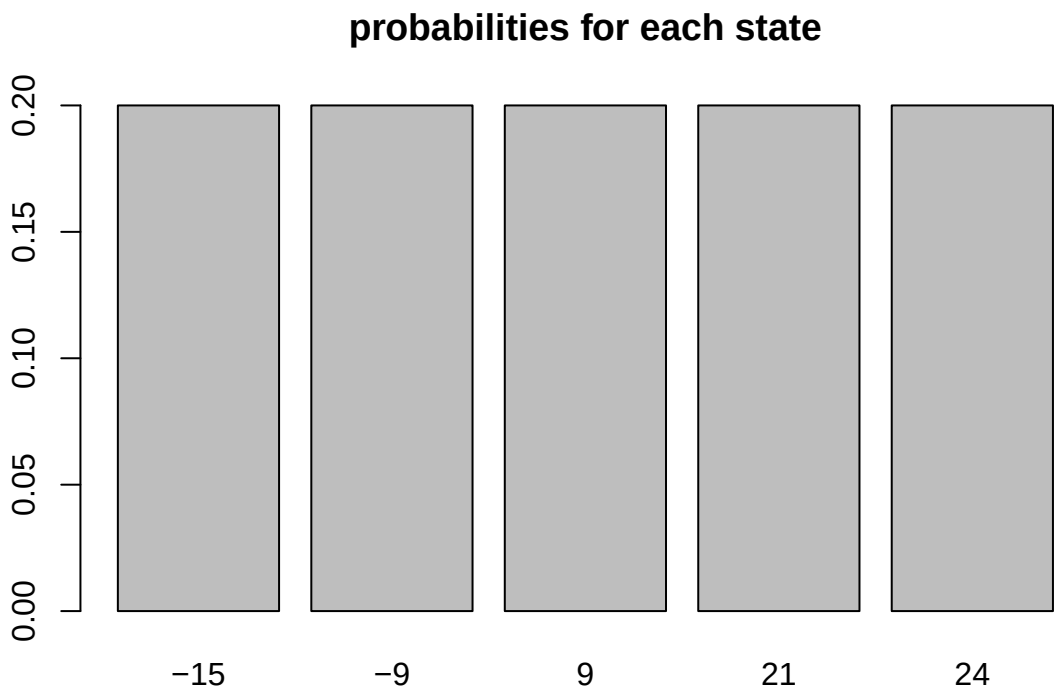
```
(v1 = value.at.risk(alpha = 0.3, y, p))
```

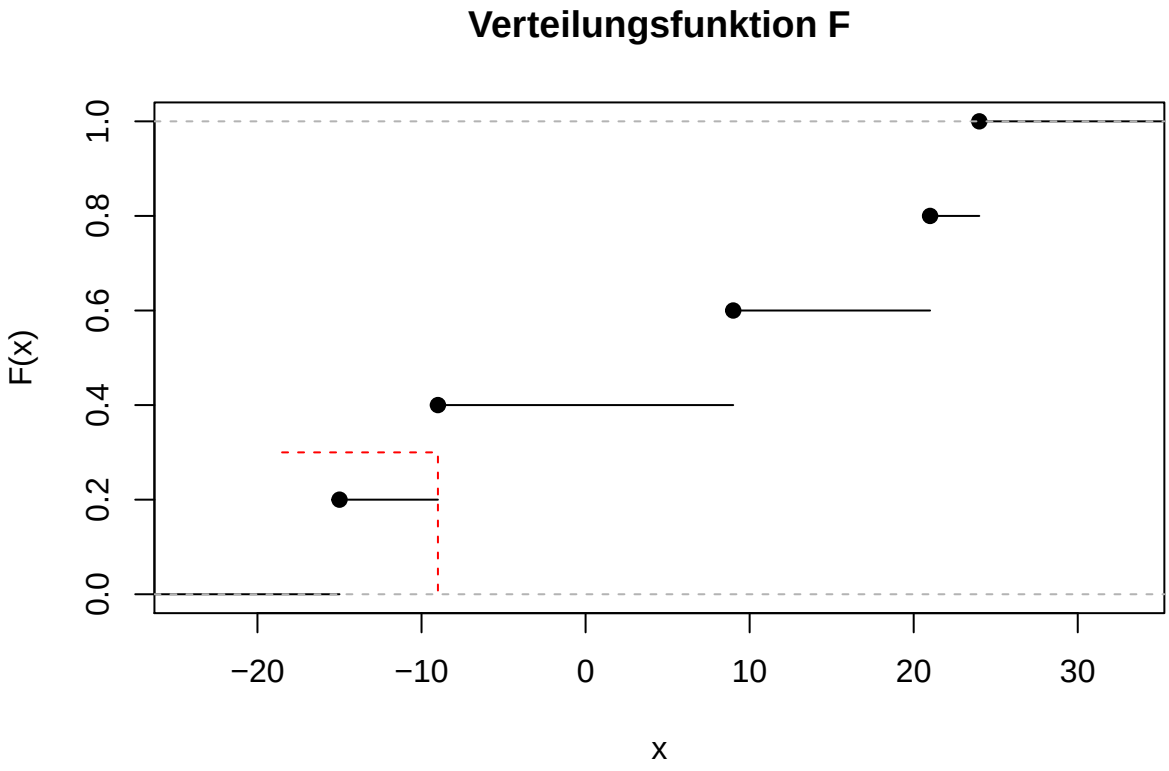

probabilities for each state
0.00
0.05
0.10
0.15
0.20
−5
−3
3
7
8




```
## 30%
##  -3
```

```
#homogeneity
value.at.risk(alpha = 0.3, 3*y, p)
```


probabilities for each state
0.00
0.05
0.10
0.15
0.20
−15
−9
9
21
24

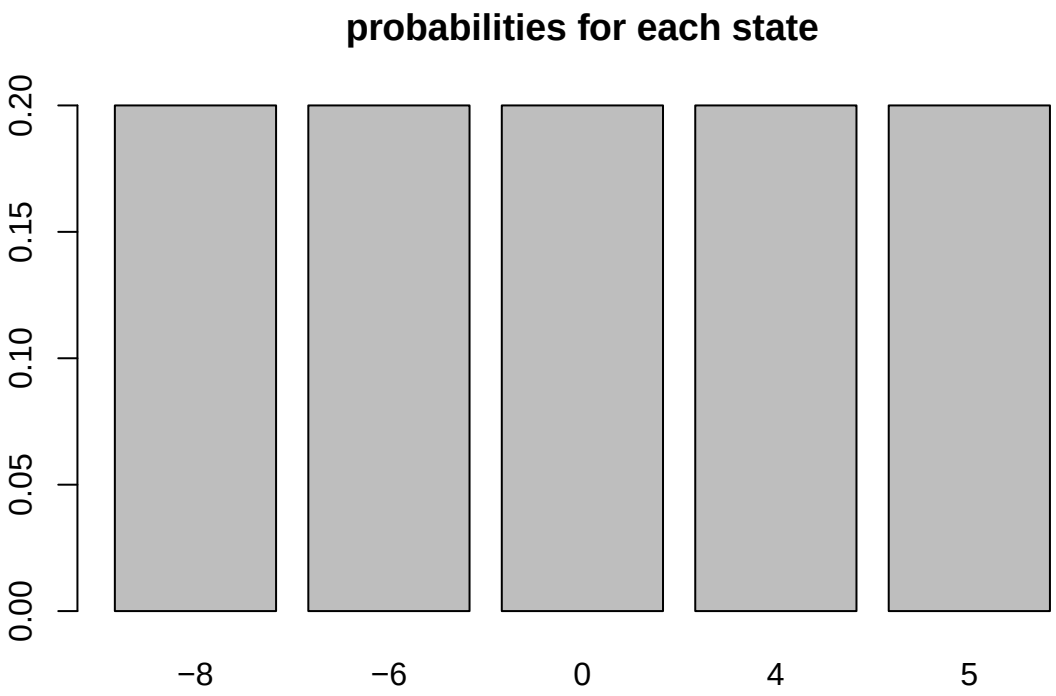


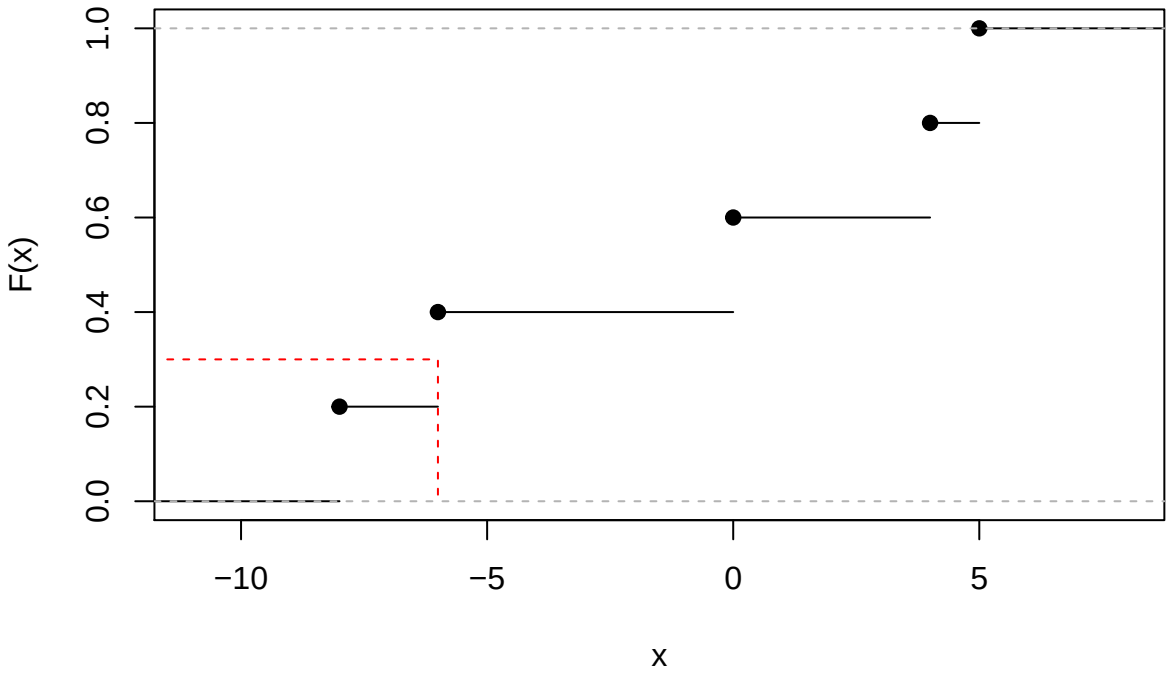

```
## 30%
##  -9
```

```
3*v1
```

```
## 30%
##  -9
```

```
#adding constant risk: cash invariance
value.at.risk(alpha = 0.3, y-3, p)
```


probabilities for each state
0.00
0.05
0.10
0.15
0.20
−8
−6
0
4
5




```
## 30%
##  -6
```

```
v1 - 3
```

```
## 30%
##  -6
```

```
#############################################################################

# Exercise 6.3

xi1 = c(5,-4,-2)
xi2 = c(8,2,0)
xi3 = c(4,1,0)
xi4 = c(-9,0,-10)

(Xi = rbind(xi1,xi2,xi3,xi4))
```

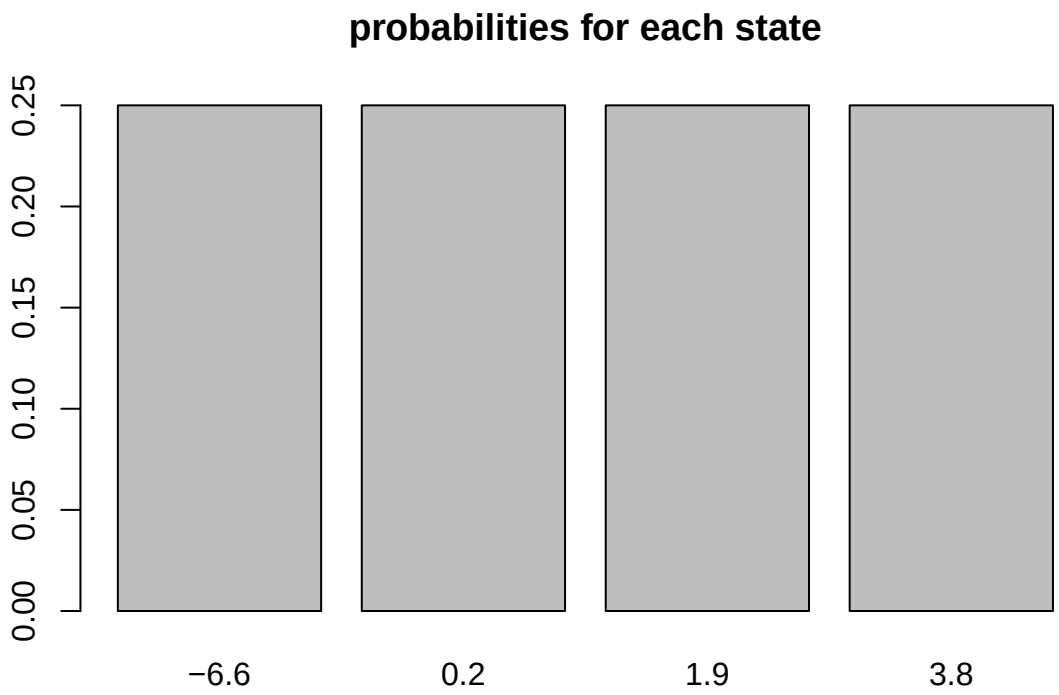
```
##      [,1] [,2] [,3]
## xi1    5   -4   -2
## xi2    8    2    0
## xi3    4    1    0
## xi4   -9    0  -10
```

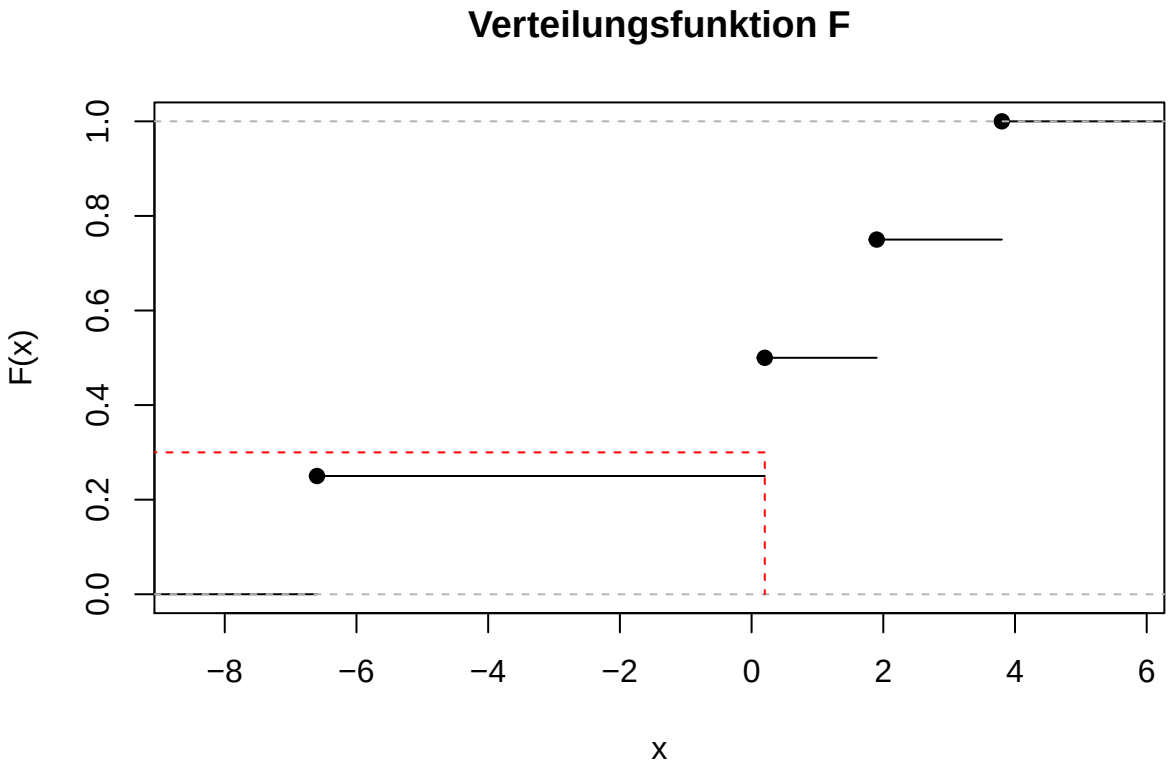
```
n = dim(Xi)[1]
J = dim(Xi)[2]
#portfolio investment
```

```r
x = c(0.4,0.3,0.3)

(r = Xi %*% x)
```

```
##      [,1]
## xi1  0.2
## xi2  3.8
## xi3  1.9
## xi4 -6.6
```

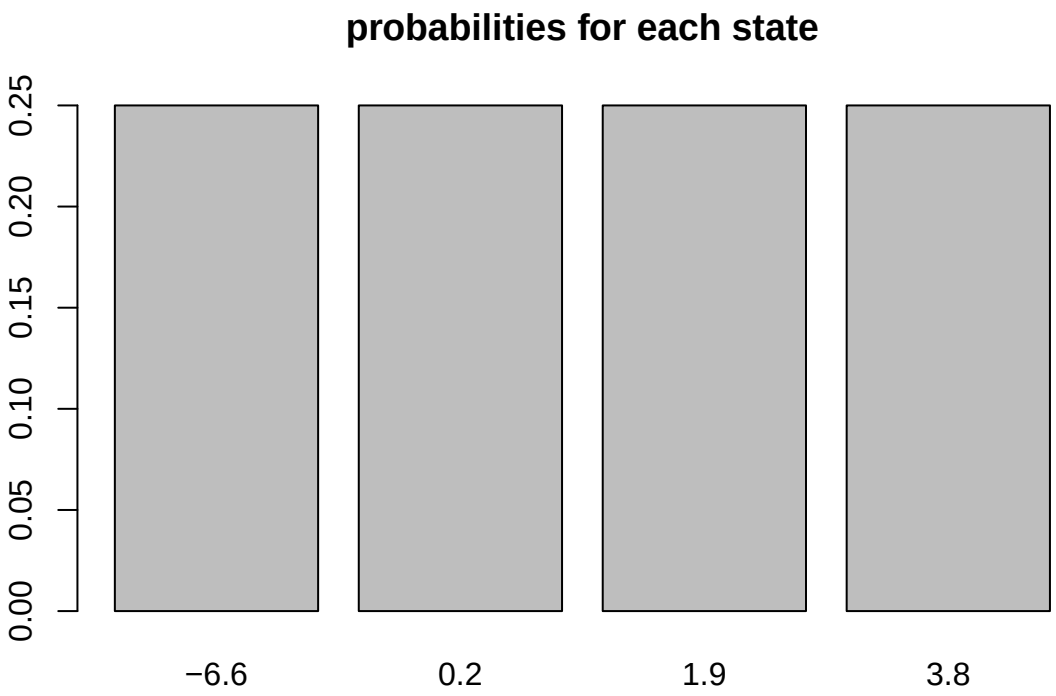
```r
p = rep(1, times=n)/n

value.at.risk(alpha=0.3, discrete.values = r, prob = p)
```

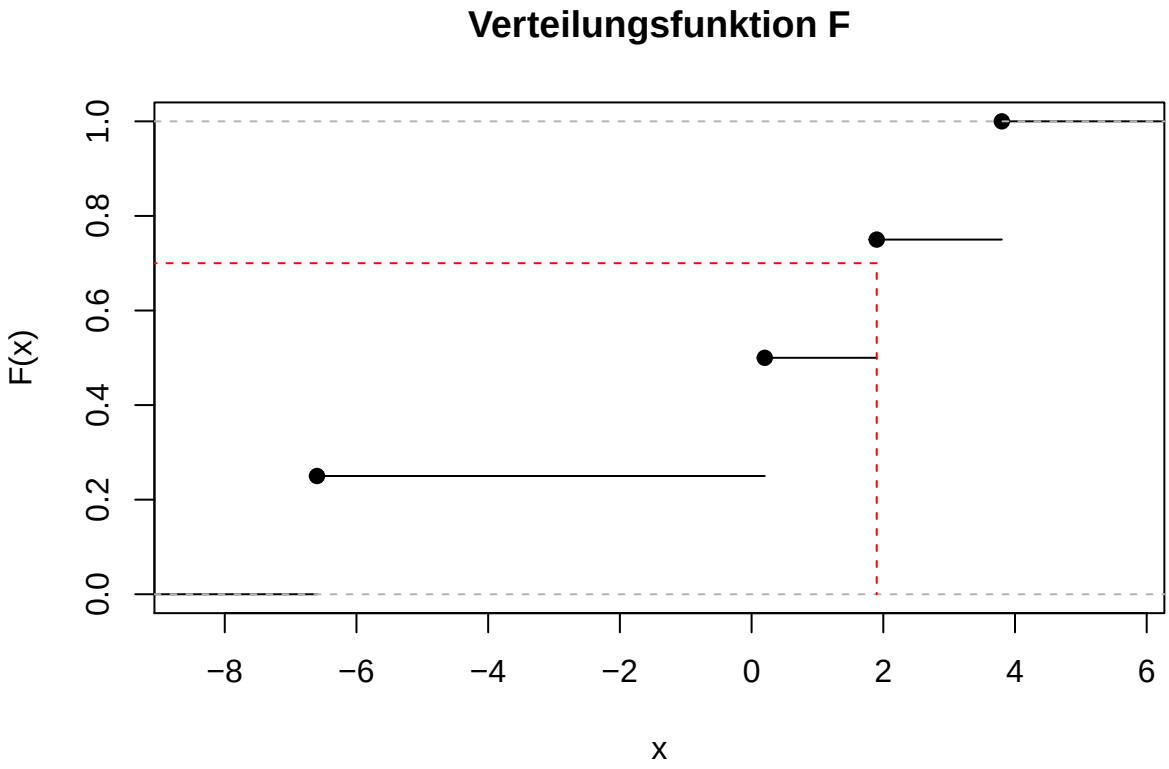





```
## 30%
## 0.2
```

```
value.at.risk(alpha=0.7, discrete.values = r, prob = p)
```


probabilities for each state
0.00
0.05
0.10
0.15
0.20
0.25
−6.6
0.2
1.9
3.8




```
## 70%
## 1.9
```