Solutions: Practical 2

1. (a) We find

$$F_X(x) = \int_2^x \left(\frac{2}{t}\right) dt$$
$$= 2^3 \int_2^x t^{-3} dt$$
$$= 8\left[-\frac{1}{2t^2}\right]_2^x$$
$$= 1 - \frac{4}{x^2},$$

giving

$$F_X(x) = \begin{cases} 0 & x \leq 2 \\ 1 - \frac{4}{x^2} & x > 2 \end{cases}$$

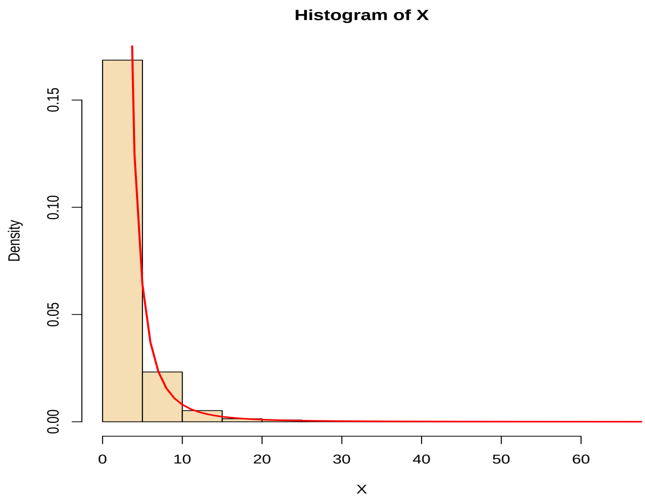
(b) Finding the inverse - rearranging F_X to make x the subject gives

$$F_X^{-1}(x) = \sqrt{\frac{4}{1-x}}, \quad x > 2.$$

(c) R function:

```
inv.cdf = function(N){
  X = vector("numeric", length = N)
  for(i in 1:N){
    U = runif(1)
    X[i] = sqrt(4/(1 - U))}
  return(X)}
```

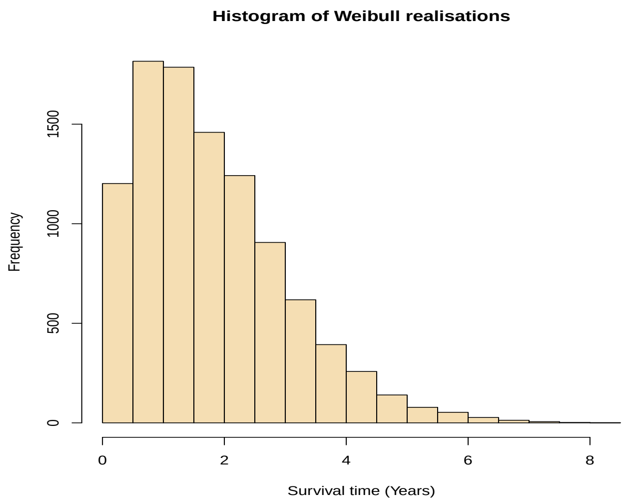
(d)



2. (a) Following the same steps as in question 1 (without having to integrate, as we are given the CDF here!), gives:

```
inv.cdf.weibull = function(N, lambda, kappa){
  X = vector("numeric", length = N)
  for(i in 1:N){
    U = runif(1)
    X[i] = lambda*((-log(1-U))^(1/kappa))}
  return(X)}
```

(b) Implementing the function in (a), with `N=10000`, `lambda=2` and `kapp=1.5` and storing the values in `X`, gives:



(c)

```
mean(X)
[1] 1.800228
length(X[X > 5])/length(X}
[1] 0.018
```

(d) There is! Students should explore the intrinsic function `rweibull`, noting that the first parameter is the shape and the second is the scale, so:

```
X2 = rweibull(10000, 1.5, 2)
```

3. (a) Let $y = 1 - e^{-x/\sigma}$. Then

$$\begin{aligned} -x/\sigma &= \log(1-y) \\ x &= -\sigma \log(1-y), \end{aligned}$$

and so

$$F_X^{-1} = -\sigma \log(1-x).$$

Code:

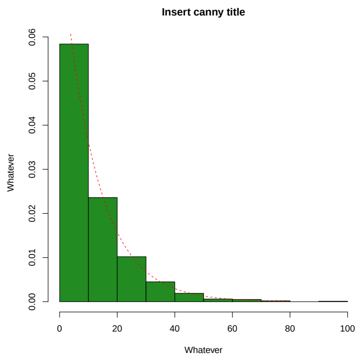
```
inv.cdf.gpd = function(N, sigma){
  X = vector("numeric", length = N)
  for(i in 1:N){
    U = runif(1)
    X[i] = -sigma*log(1-U)}
  return(X)}
```

(b) Code:

```
X = inv.cdf.gpd(1000, 12)
```

(c) Code:

```
hist(X, probability=TRUE, col="ForestGreen", xlab="Whatever", ylab="
   Whatever", main="Insert canny title")
curve((1/12)*exp(-x/12), from=0, to=80, col="red", lty=2, add=TRUE)
```



(d) Code:

```
summary(X)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
 0.00201   3.29576   7.96130  11.63675  16.07250  98.41827
```

The mean and quartiles match up very closely to those obtained from the wind speed data.

(e) Code:

```
length(X[X>50])/length(X)
[1]  0.01
```

So we might expect this wind speed to be exceeded once every 100 observations – or once every 4 days.

(f) Code:

```
X1 = rexp(1000, 1/12)
summary(X1)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
  0.00696   3.20706   7.66012  11.89842  16.94143  76.07372
```

Summaries for `X` and `X1` very similar - this form of the Generalised Pareto Distribution is equivalent to an exponential distribution with rate $1/\sigma$.