Practice Lecture 2: Descriptive statistics and models

Erlis Ruli (ruli@stat.unipd.it)

20 October 2021

1 Univariate samples

To illustrate univariate summaries let us use a real dataset. We consider the dataset `cars`, one of the many datasets inside `R`' memory. It reports speed and stopping time for 50 car models.

```
> head(cars)
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

```
> summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

We see that the average speed for this observed sample is equal to 15.4, which is very close the the median, whereas the average distance is 42, slightly different from the median, which in this case is 36. The command `summary` gives also the observed ordered statistics: average, lower quartile, median, upper quartile and the maximum.

The sample variance is computed by

```
> var(cars$speed)
```

```
## [1] 27.95918
```

```
> var(cars$dist)
```

```
## [1] 664.0608
```

We can also compute the median of a sample by

```
> median(cars$speed)
```

```
## [1] 15
```

```
> median(cars$dist)
```

```
## [1] 36
```

The observed quartiles can be obtained by

```
> quantile(cars$speed, probs = 1/4, type=6)# lower quarile
```

```
## 25%
##  12
```

```
> quantile(cars$speed, probs = 3/4, type=6) # upper quartile
```

```
##   75%
## 19.25
```

```
> quantile(cars$speed, probs = 2/4, type=6) # the median
```

```
## 50%
##  15
```

The `quantile` function gives *observed sample quantiles* of any user-supplied level $p \in (0, 1)$. In L2 we defined the observed sample quantile as $x_{([p(n+1)])}$ but there are other ways to define it. These ways can be selected by specifying the `type` argument. Varying the `type` argument could lead to different results, especially if the observed sample has ties. However, in practice such differences for large n are negligible. This issue does not arise when we compute the quantile of a random variable since its definition (based on the infimum) guarantees us a unique solution. In the above example we use `type`=6, which corresponds to our definition. See the help page `?quantile` for more details.

1.1 QQ-plots

Let's first try to build a QQ-plot by hand. Suppose our observed sample is

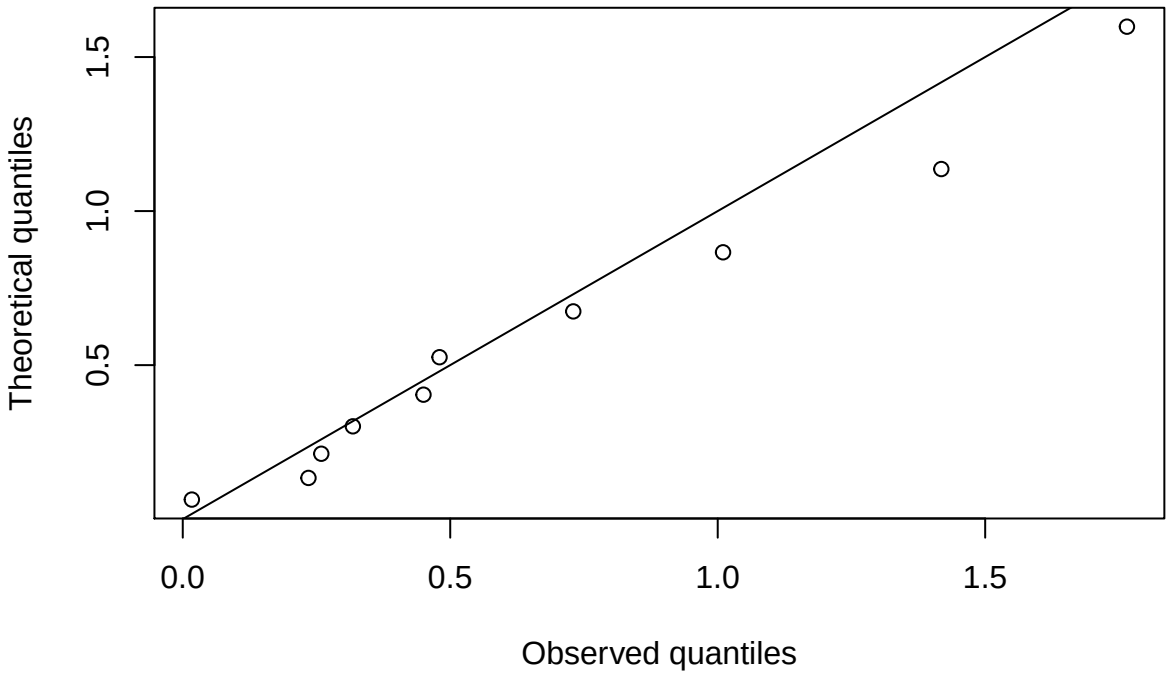
$$0.318, 1.765, 0.259, 0.450, 0.730, 0.235, 0.017, 1.010, 1.418, 0.480$$

the sample we saw in L2. Suppose also that we suspect that Nature generated this data in an iid fashion from $Exp(3/2)$. Then $F(x) = 1 - e^{-3x/2}$ and $F^{-1}(p) = -2\log(1-p)/3$.

```
> x <- c(0.318, 1.765, 0.259, 0.450, 0.730, 0.235, 0.017, 1.010, 1.418, 0.480)
> (x.order <- sort(x))
```

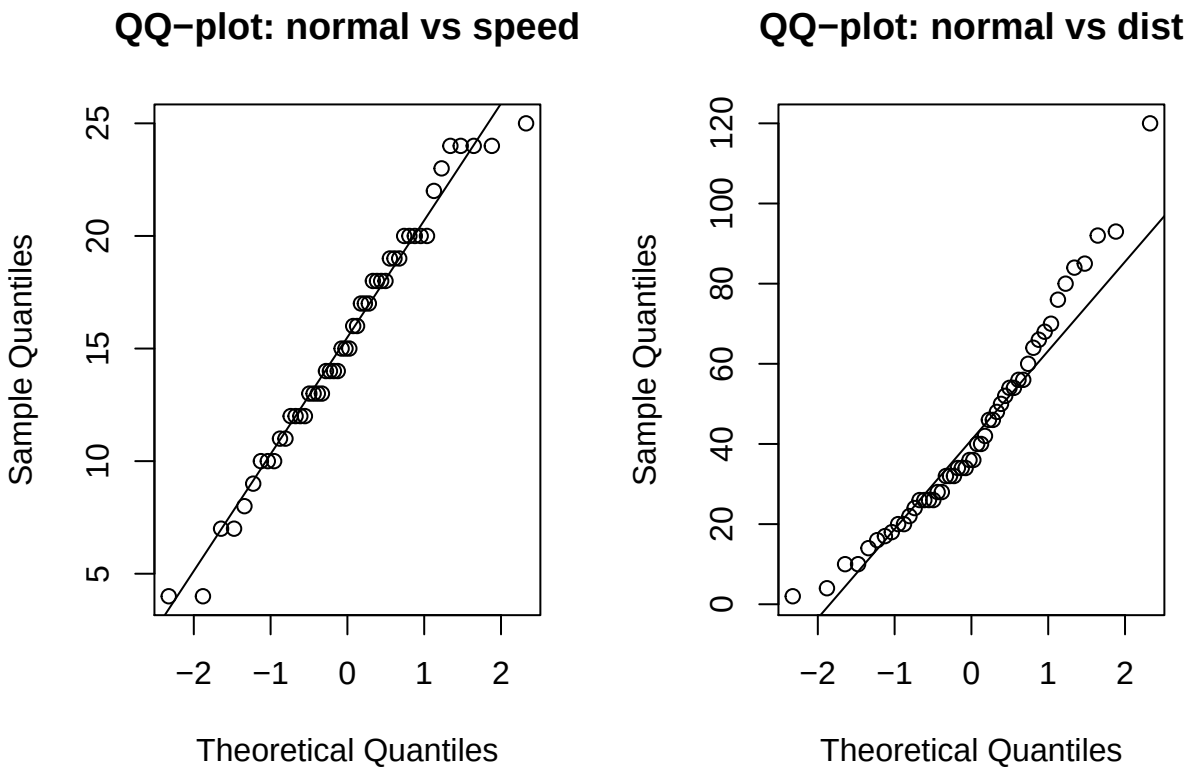
```
##  [1] 0.017 0.235 0.259 0.318 0.450 0.480 0.730 1.010 1.418 1.765
```

```
> n <- length(x)
> k <- (1:n)/(n+1)
> qq <- -2*log(1-k)/3
> plot(x=x.order, y=qq, xlab="Observed quantiles", ylab="Theoretical quantiles")
> abline(a=0, b=1)
```



To build QQ-plots against the normal quantiles for the variables of the speed dataset we can use the following commands

```
> par(mfrow=c(1,2))
> qqnorm(cars$speed,main="QQ-plot: normal vs speed")
> qqline(cars$speed)
>
> qqnorm(cars$dist,main="QQ-plot: normal vs dist")
> qqline(cars$dist)
```



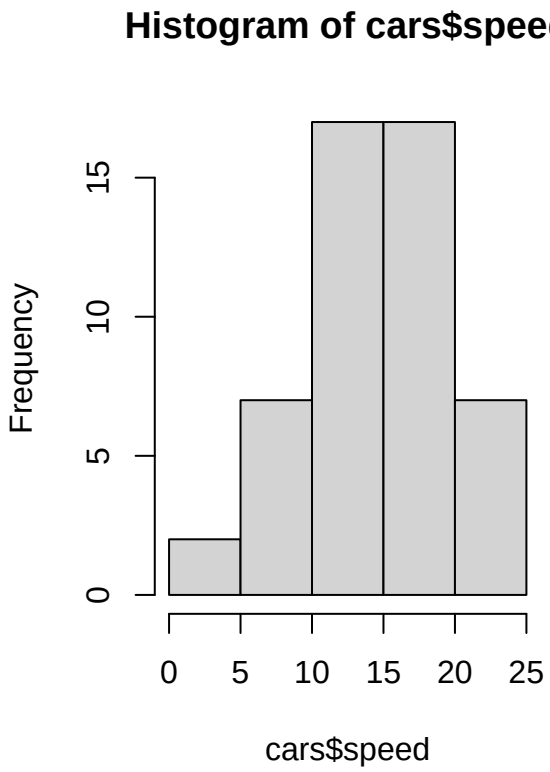
We see that the quantiles of `speed` are quite close to those of the normal distribution whereas those of `dist` are quite different. This suggests that the normal model could be a good model for `speed` but perhaps not for `dist`. In particular, we notice that `dist` has a skewed distribution. For the half-plane line we used the command `qqline` which is a robust version of the $y = x$ line.

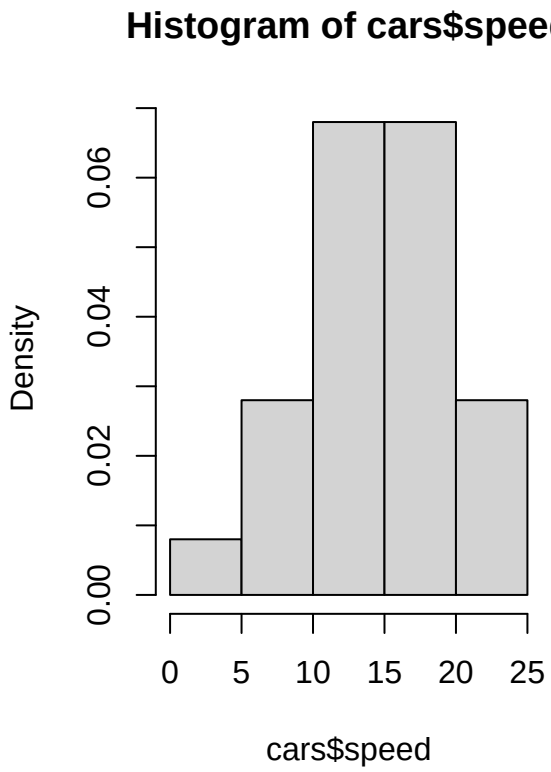
Exercise For the variable speed only, try to build the QQ-plot "by hand". For this, suppose that F is N(15.4, 28). The quantile function of the normal distribution is given by the command `qnorm`.

1.2 Histograms

The command to be used is `hist`, its option `breaks` controls the number of points in the partition and the option `freq = FALSE` gives the histogram as defined in L2.

```
> par(mfrow=c(1,2))
> # a non scaled histogram
> hist(cars$speed)
> # a scaled histogram
> hist(cars$speed, freq = FALSE)
```





1.3 Empirical distribution function

Let's try first to compute it by hand for variable `speed`. Let's first compute its frequency distribution

```
> table(cars$speed)
```

```
##
##  4  7  8  9 10 11 12 13 14 15 16 17 18 19 20 22 23 24 25
##  2  2  1  1  3  2  4  4  4  3  2  3  4  3  5  1  1  4  1
```

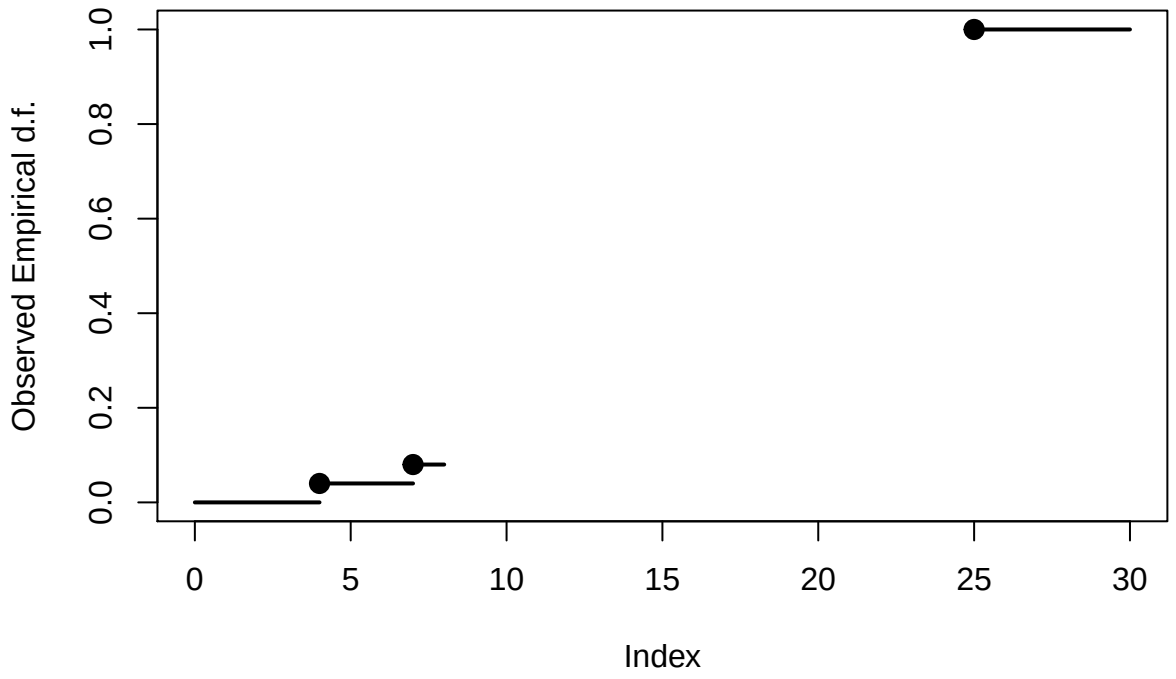
We see that there are two car which have speed equal to 1, other two cars have speed equal to 7 and so on. Denoting by $\hat{F}_n(x)$ the empirical d.f., we have that

$$\hat{F}_n(x) = \begin{cases} 0 & \text{if } x < 4 \\ 2/50 & \text{if } 4 \leq x < 7 \\ 4/50 & \text{if } 7 \leq x < 8 \\ \dots & \dots \\ 49/50 & \text{if } 24 \leq x < 25 \\ 50/50 & \text{if } 25 \leq x. \end{cases}$$

To plot this function we could run

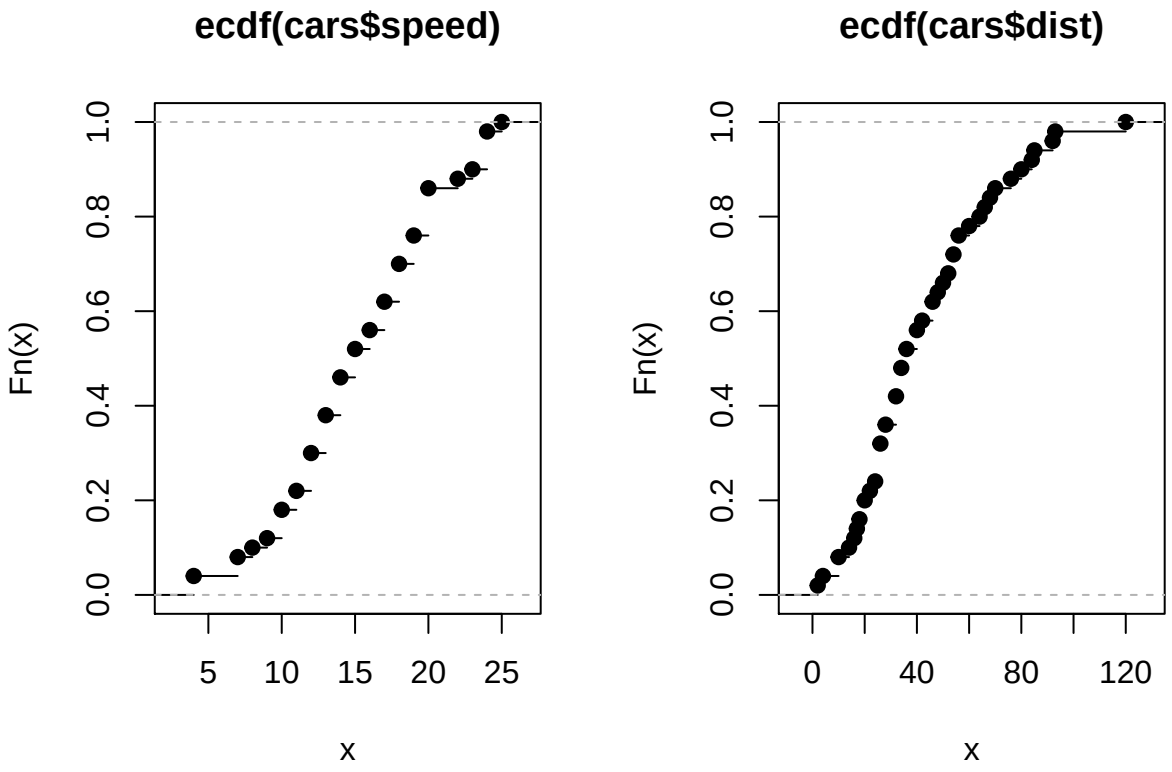
```
> plot(0,xlim=c(0,30), ylim=c(0,1), cex=0, ylab="Observed Empirical d.f.")
> segments(x0=0,x1=4,y0=0,y1=0, lwd=2)
>
> segments(x0=4,x1=7,y0=2/50,y1=2/50, lwd=2)
> points(x=4, y=2/50, pch=20, cex=2)
>
> segments(x0=7,x1=8,y0=4/50,y1=4/50, lwd=2)
```

```
> points(x=7, y=4/50, pch=20, cex=2)
>
> # etc
> # etc
>
> segments(x0=25,x1=30,y0=50/50,y1=50/50, lwd=2)
> points(x=25, y=50/50, pch=20, cex=2)
```



The R command for the empirical distribution function is

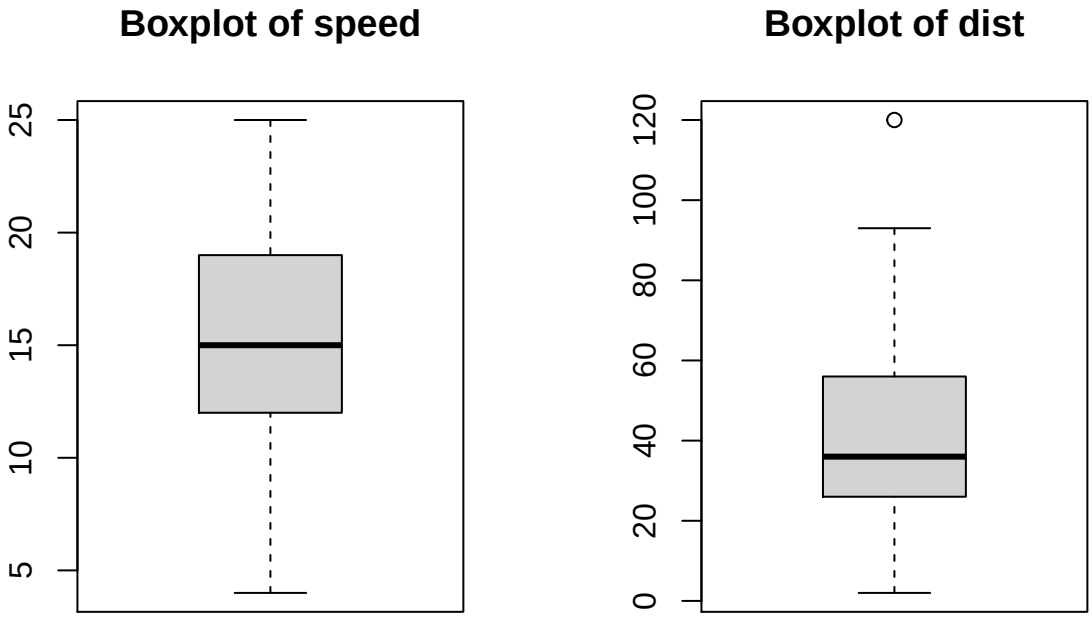
```
> par(mfrow=c(1,2))
> plot(ecdf(cars$speed))
> plot(ecdf(cars$dist))
```



1.4 Boxplots

Boxplots are computed with

```
> par(mfrow=c(1,2))
> boxplot(cars$speed, main="Boxplot of speed")
> boxplot(cars$dist, main="Boxplot of dist")
```



For the variable `dist` we notice an unusual observation, i.e. an outlier.

1.5 Distribution of an ordered statistic

In L2 we saw that if $X_1, \ldots, X_n$, with $X_i \underset{iid}{\sim} F$, with F being continuous, then the largest $X_{(n)}$ has p.d.f. $f_{X_{(n)}}(x) = n[F(x)]^{n-1} f(x)$.

Let's check this by simulation in a concrete example. In particular, let F be the d.f. of an exponential r.v. with $\lambda = 1$. In this case

$$f_{X_{(n)}}(x) = n(1 - e^{-x})^{n-1} e^{-x}.$$

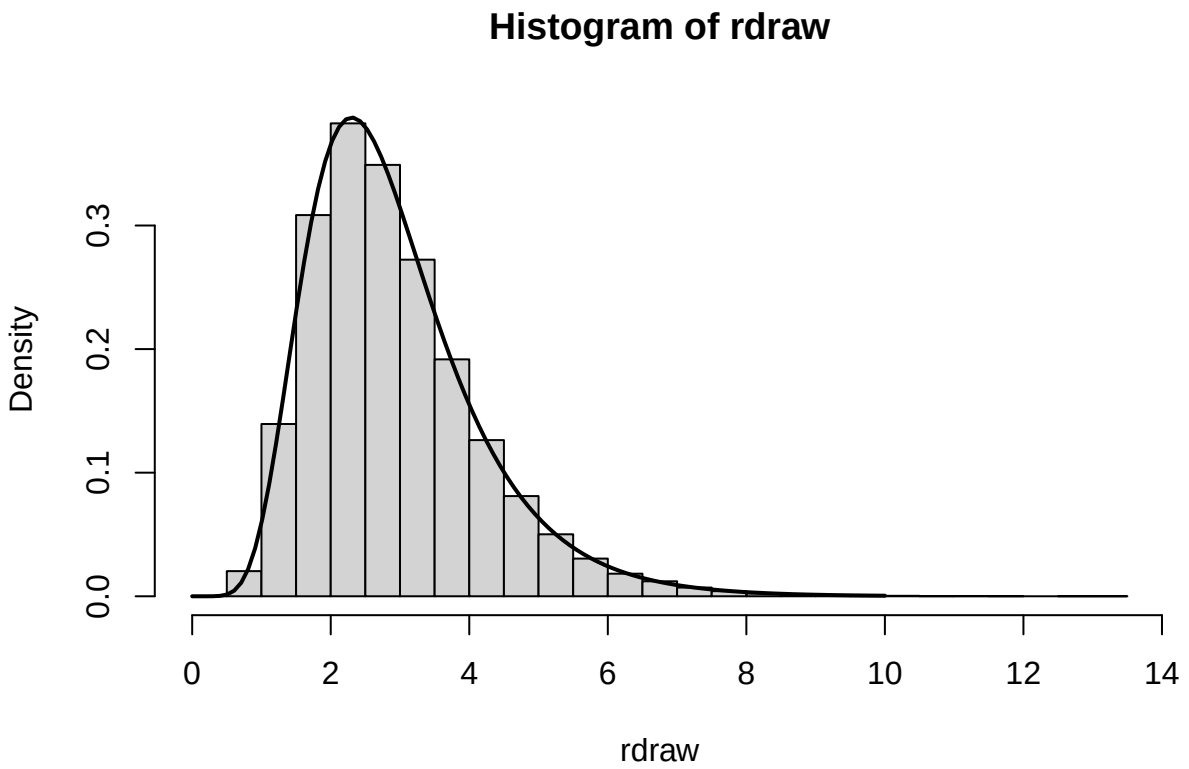
Note that is not immediate how to draw random variates from this distribution since it does not exists in R. But, taking advantage of its relation with Exp(1), we can draw a single random value from this distribution by:

- drawing n random variates $x_1, \ldots, x_n$ from Exp(1)
- and setting the desired draw equal to $\max(x_1, \ldots, x_n)$.

Looping over this for say m times we have m random draws from the distribution of $X_{(n)}$.

```
> set.seed(2020)
> m = 1e+5
> n <- 10
> # to be filled by exponential r.variates
> x.exp <- vector(mode = "numeric", n)
>
> # to be filled by draws from X_{(n)}
> rdraw <- vector(mode = "numeric", m)
>
> for(i in 1:m) {
+   x.exp <- rexp(n,rate=1)
+
+   # sort exponential r.variates
+   x.exp.order <- sort(x.exp)
+
+   # save the largest
+   rdraw[i] <- x.exp.order[n]
+ }
```

```
> ff <- function(x,n) n*(1-exp(-x))^(n-1)*exp(-x)
>
> hist(rdraw, freq = F, breaks = 40)
> plot(function(x) ff(x,n=n), xlim=c(0,10), n=100, add=T, lwd=2)
```

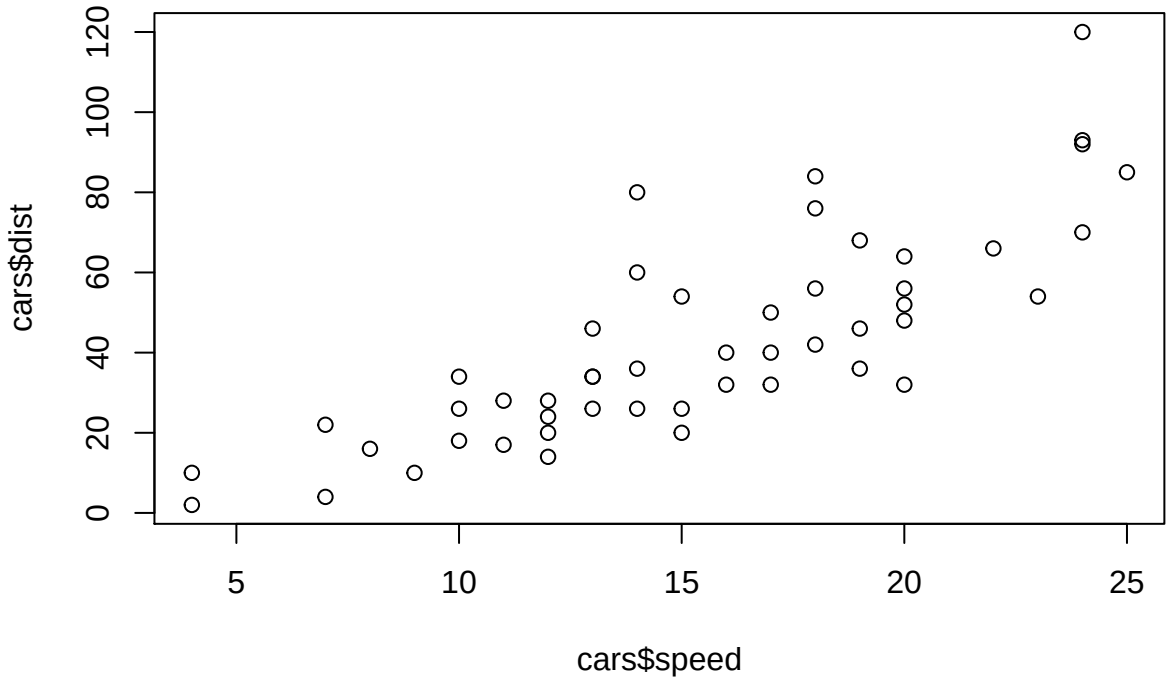


EXERCISE 1 Can you think about a different algorithm for generating random values from the distribution of $X_{(n)}$? If so, compare this algorithm with that illustrated above. (Hint: see Lecture 0.)

2 Bivariate samples

From the scatter plot of the two variables shown below we notice that there is a positive linear association between the two variables.

```
> plot(cars$speed, cars$dist)
```



Indeed, the sample covariance and sample correlation indices which are

```
> # covariance
> cov(cars$speed, cars$dist)
```

```
## [1] 109.9469
```

```
> # correlation: option 1
> cor(cars$speed, cars$dist)
```

```
## [1] 0.8068949
```

```
> # correlation: option 2
> cov(cars$speed, cars$dist)/
+   sqrt(var(cars$speed)*var(cars$dist))
```

```
## [1] 0.8068949
```