Exam exercise for Module 1: Wind speed distributions

In this workshop we consider a continuous probability distribution called the Weibull distribution. Among other things, it is used to model wind speed distributions.

We recommend that you answer the exercises using Rmarkdown (you can simply use the exam Rmarkdown file as a starting point).

Part I: The Weibull distribution

The Weibull distribution depends on two parameters $k > 0$ and $\lambda > 0$. If X follows a Weibull distribution with parameters k and λ, we write $X \sim \texttt{weibull}(k, \lambda)$. In this case, X has the probability density function

$$f(x) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}, & x \geq 0, \\ 0, & x < 0, \end{cases}$$

and the distribution function

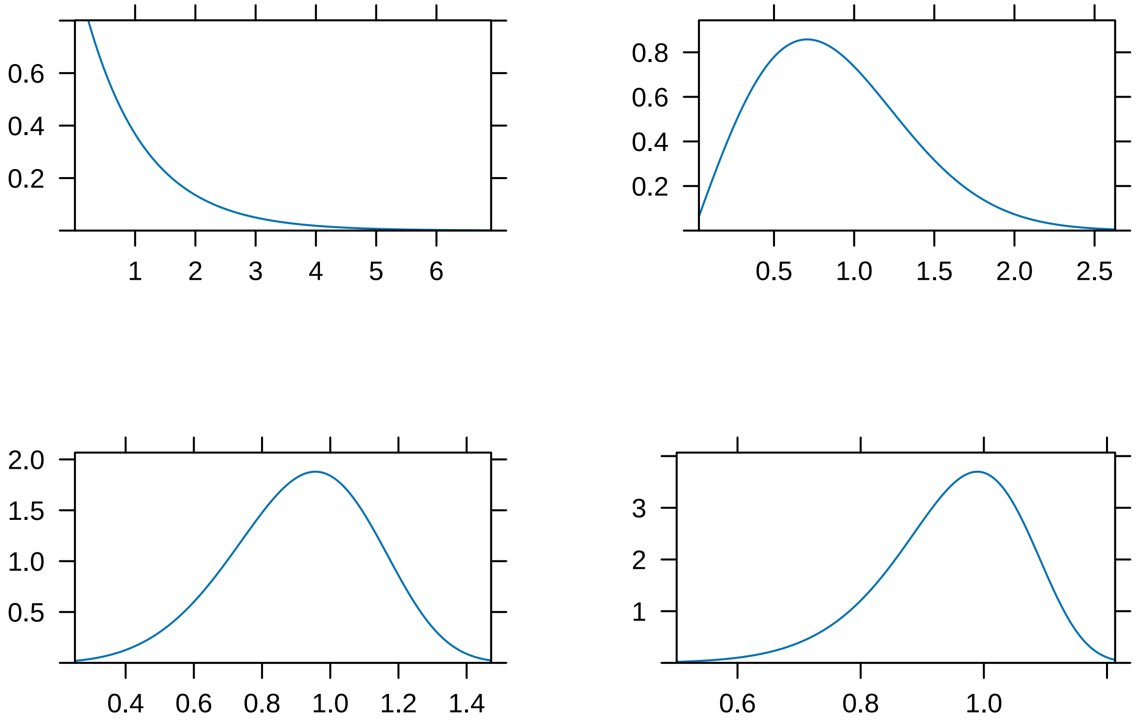
$$F(x) = \begin{cases} 1 - e^{-(x/\lambda)^k}, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

The parameter k is called the shape parameter, since it determines the shape of the distribution, while λ is called the scale parameter, because it works by scaling the x-axis.

1. Use the `mosaic` package with the `plotDist` function to make plots of different parameter combinations to demonstrate that λ is a scale parameter and k is a shape parameter. (Hint: `plotDist("weibull", params = list(shape = ., scale = .))`) First we run the variable k or shape through 1 to 10 and see how the shape changes, it changes from skewed to the left to be skewed to the right.

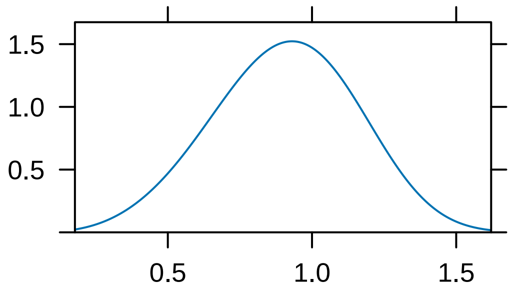
```
lambda <- 1
k <- 1
w1 <- plotDist("weibull", params = list(shape = k, scale = lambda))
k <- 2
w2 <- plotDist("weibull", params = list(shape = k, scale = lambda))
k <-5
w3 <- plotDist("weibull", params = list(shape = k, scale = lambda))
```

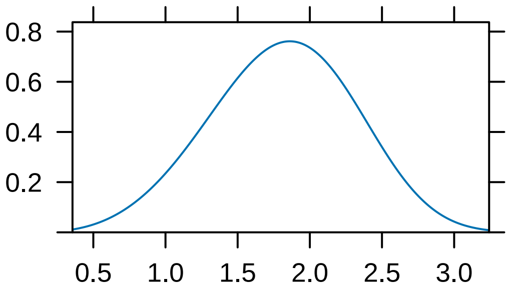
```
k <- 10
w4 <- plotDist("weibull", params = list(shape = k, scale = lambda))
grid.arrange(w1, w2, w3, w4, ncol = 2)
```

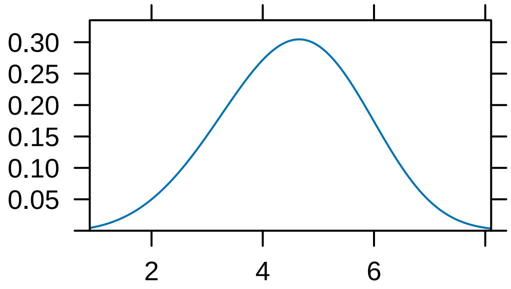


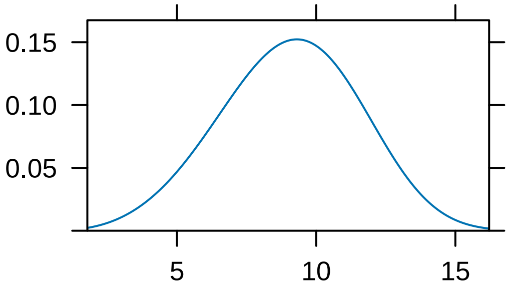
We then the run the same changes but for lambda and see how the scale changes:

```
lambda <- 1
k <- 4
w1 <- plotDist("weibull", params = list(shape = k, scale = lambda))
lambda <- 2
w2 <- plotDist("weibull", params = list(shape = k, scale = lambda))
lambda <-5
w3 <- plotDist("weibull", params = list(shape = k, scale = lambda))
lambda <- 10
w4 <- plotDist("weibull", params = list(shape = k, scale = lambda))
grid.arrange(w1, w2, w3, w4, ncol = 2)
```







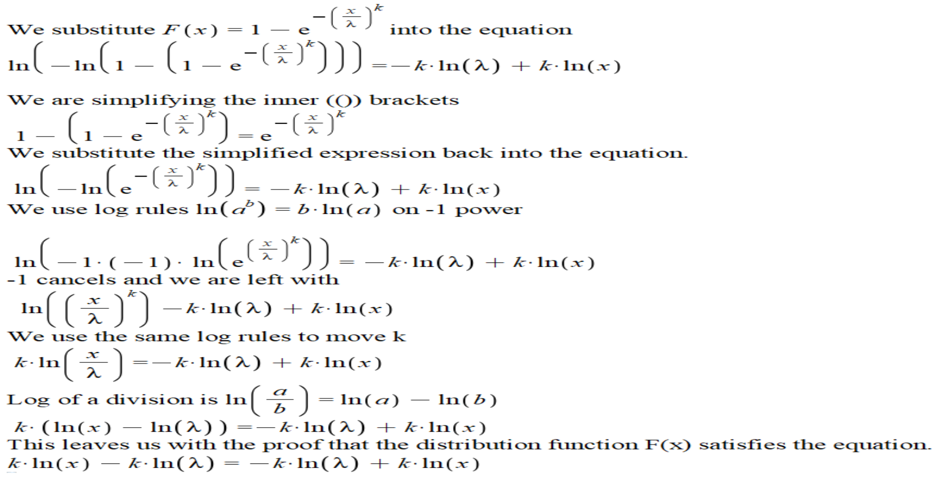


Here we see that lambda defines how “sharp” or “pointy” the function is, larger lambda means longer tails or larger variance. It also seems that the center is around lambdas value.

2. Assume that $x \geq 0$. Show that the distribution function $F(x)$ satisfies

$$\ln(-\ln(1 - F(x))) = -k\ln(\lambda) + k\ln(x).$$

```r
library("png")
pp <- readPNG("1.2.png")
plot.new()
rasterImage(pp,0,0,1,1)
```



We substitute $F(x) = 1 - e^{-\left(\frac{x}{\lambda}\right)^k}$ into the equation

$\ln\left(-\ln\left(1 - \left(1 - e^{-\left(\frac{x}{\lambda}\right)^k}\right)\right)\right) = -k \cdot \ln(\lambda) + k \cdot \ln(x)$

We are simplifying the inner (()) brackets

$1 - \left(1 - e^{-\left(\frac{x}{\lambda}\right)^k}\right) = e^{-\left(\frac{x}{\lambda}\right)^k}$

We substitute the simplified expression back into the equation.

$\ln\left(-\ln\left(e^{-\left(\frac{x}{\lambda}\right)^k}\right)\right) = -k \cdot \ln(\lambda) + k \cdot \ln(x)$

We use log rules $\ln(a^b) = b \cdot \ln(a)$ on -1 power

$\ln\left(-1 \cdot (-1) \cdot \ln\left(e^{\left(\frac{x}{\lambda}\right)^k}\right)\right) = -k \cdot \ln(\lambda) + k \cdot \ln(x)$

-1 cancels and we are left with

$\ln\left(\left(\frac{x}{\lambda}\right)^k\right) - k \cdot \ln(\lambda) + k \cdot \ln(x)$

We use the same log rules to move k

$k \cdot \ln\left(\frac{x}{\lambda}\right) = -k \cdot \ln(\lambda) + k \cdot \ln(x)$

Log of a division is $\ln\left(\frac{a}{b}\right) = \ln(a) - \ln(b)$

$k \cdot (\ln(x) - \ln(\lambda)) = -k \cdot \ln(\lambda) + k \cdot \ln(x)$

This leaves us with the proof that the distribution function F(x) satisfies the equation.

$k \cdot \ln(x) - k \cdot \ln(\lambda) = -k \cdot \ln(\lambda) + k \cdot \ln(x)$

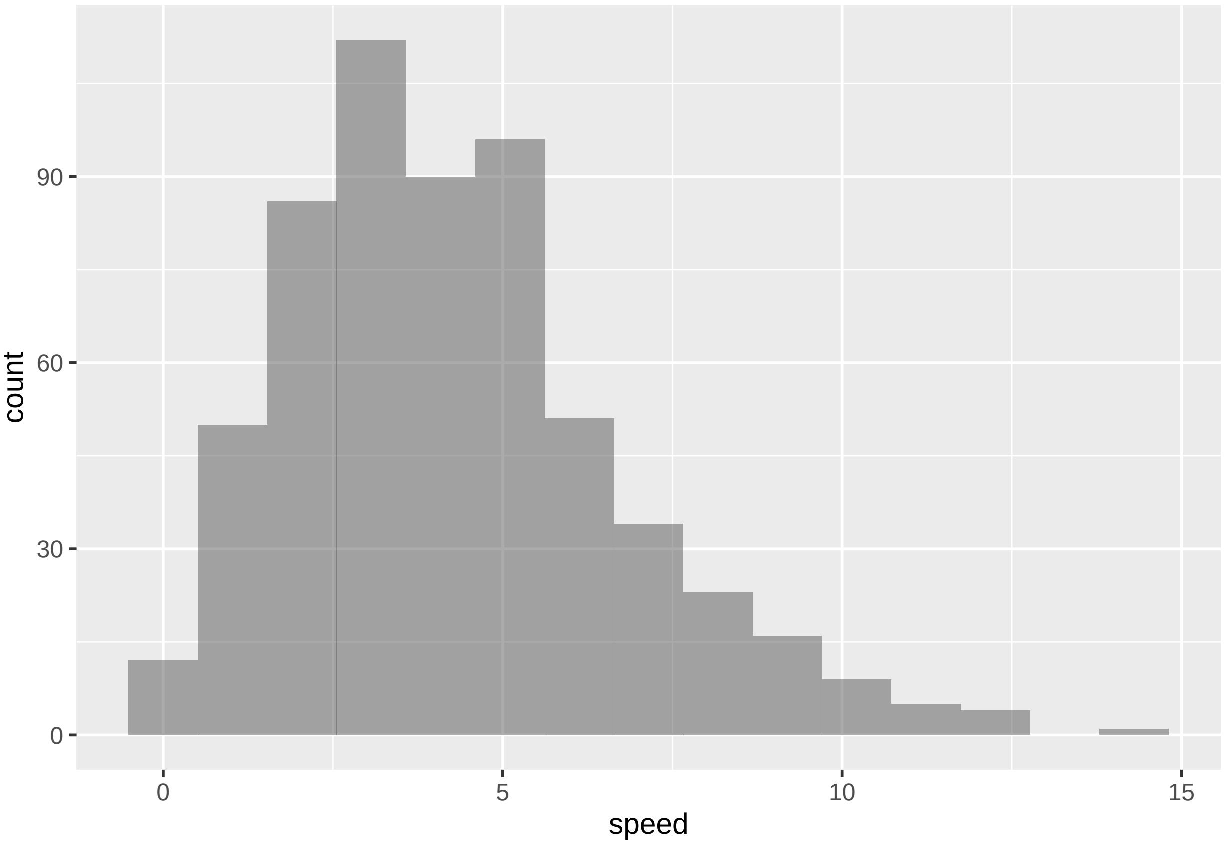
Part II: Wind speed measurements

In this part we consider a data set containing wind speed measurements from a Danish weather station located at Sjælsmark. The data set contains the wind speed measured at 12 noon every day of January in the years 2001-2019. We first load the data set:

```r
speed<-read.delim("https://asta.math.aau.dk/datasets?file=windSpeed.txt",header=FALSE)[,1]
```

1. Draw a histogram of the wind speed observations by editing the R chunk below. Explain how a histogram is constructed. Do you think the observations come from a normal distribution?

```r
#hist(speed)
gf_histogram(~speed,bins=15)
```



A histogram shows how many times something ocoured in a given interval. in this instance, we want 15 bins, such that the largest bin describe the maximum outcome. In this instance the max is 15, and thus we get that the bin size should be 15/15 = 1. Thus the first bin describes all counts in the interval 0 - 0.999 m/s. bin 2 describes from 1.. to 1.999 m/s etc. The y axis is the count, i.e. it is the total number of ocourances that fall in a given interval. A histrogram can be used to show the distribution of a measured system.

This is clearly bot gaussian (normal), it is skewed, and looks a lot like Weibull dist. with k approx. around 2, see figures from task 1.

In the following we will convince ourselves that the data actually comes from a Weibull distribution. We order the $n = 589$ observations from smallest to largest

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}.$$

2. Argue that $F(x_{(i)}) \approx \frac{i}{n}$ for $i = 1, \dots, n$. (Hint: How many observations are less than or equal to $x_{(i)}$?)

We know from 2.1) that the values Xi aproximately follow a Weibull distribution. We also know that F(xi) is the weibull distribution function. Because the sample values ordered from low to high follow a Weibull distribution as shown in 2.1) it is reasonable to assume that they follow F(x.) Also We are asked if F(x) ~= i/n for i = 1..n we know that lim(i(n)) = 1 when i -> n We also know that F(x) = 1-exp(-(x/lambda)^l) for any values of lambda and k this will also converge at 1 when x -> infinity. Therefore i/n and F(x) are approximately equal.

3. Using Exercise 2 in Part I, argue that if the observations come from a `weibull`(k, λ) distribution, then

$$\ln(-\ln(1-\tfrac{i}{n})) \approx -k\ln(\lambda) + k\ln(x_{(i)}).$$

Since we have concluded in 2.2) that F(x) = i/n when Xi is a ranked list of observations from low to high, it is fair to say that the approximation i/n will also fulfill the empirical distribution function ln(-ln(1-F(x))) = -k*ln(lambda)* + *k*ln(xi) this means that the empirical distribution function equation ln(-ln(1-i/n)) ~= -k*ln(lambda)* + *k*ln(xi) is also valid.

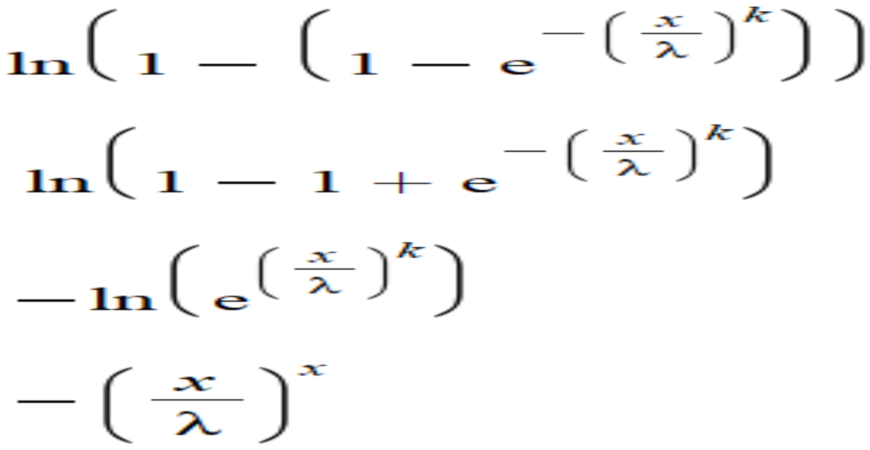
Another way to look at it is that we have concluded in 2.2 that (1-i/n) ~1-exp(-(x/lambda)^k) so we can approximate the two sides of the equation as equal.

4. Argue that the points (u_i, v_i) should lie approximately on a straight line if the observations come from a `weibull`(k, λ) distribution. Edit the code above to check that this is the case.

If the observations come from a Weibull distribution then they will work for the discrete empirical distribution function ln(-ln(1-i/n)) ~= -k*ln(lambda)* + *k*ln(xi). This means the equation is true. If we examine both sides of the equation then the equation is of the form y = ax -b and this is the function for a straight line. They will only follow this straight line if the observations follow a weibull distrubtion because the empirical distribution function is only valid in this case.

We can examine the EMF further to see it is a straight line. We just need to analyze the ln(1-F(x)) expression in the CDF

```r
library("png")
pp <- readPNG("2.4.png")
plot.new()
rasterImage(pp,0,0,1,1)
```



$$\ln\left(1 - \left(1 - e^{-\left(\frac{x}{\lambda}\right)^{k}}\right)\right)$$

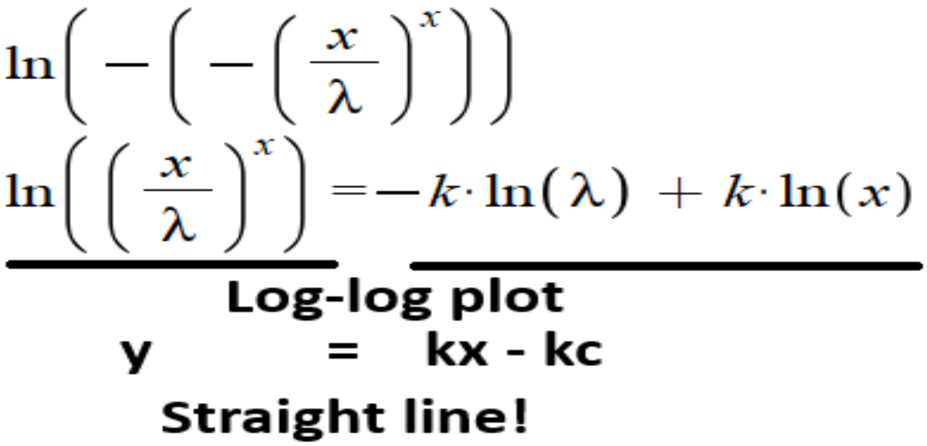
$$\ln\left(1 - 1 + e^{-\left(\frac{x}{\lambda}\right)^{k}}\right)$$

$$-\ln\left(e^{\left(\frac{x}{\lambda}\right)^{k}}\right)$$

$$-\left(\frac{x}{\lambda}\right)^{x}$$

If we substitute this expression back into the CDF we get the following

```r
library("png")
pp <- readPNG("2.4.2.png")
plot.new()
rasterImage(pp,0,0,1,1)
```



$$\ln\left(-\left(-\left(\frac{x}{\lambda}\right)^{x}\right)\right)$$

$$\underline{\ln\left(\left(\frac{x}{\lambda}\right)^{x}\right)} = \underline{-k\cdot\ln(\lambda) + k\cdot\ln(x)}$$

Log-log plot

y = kx - kc

Straight line!

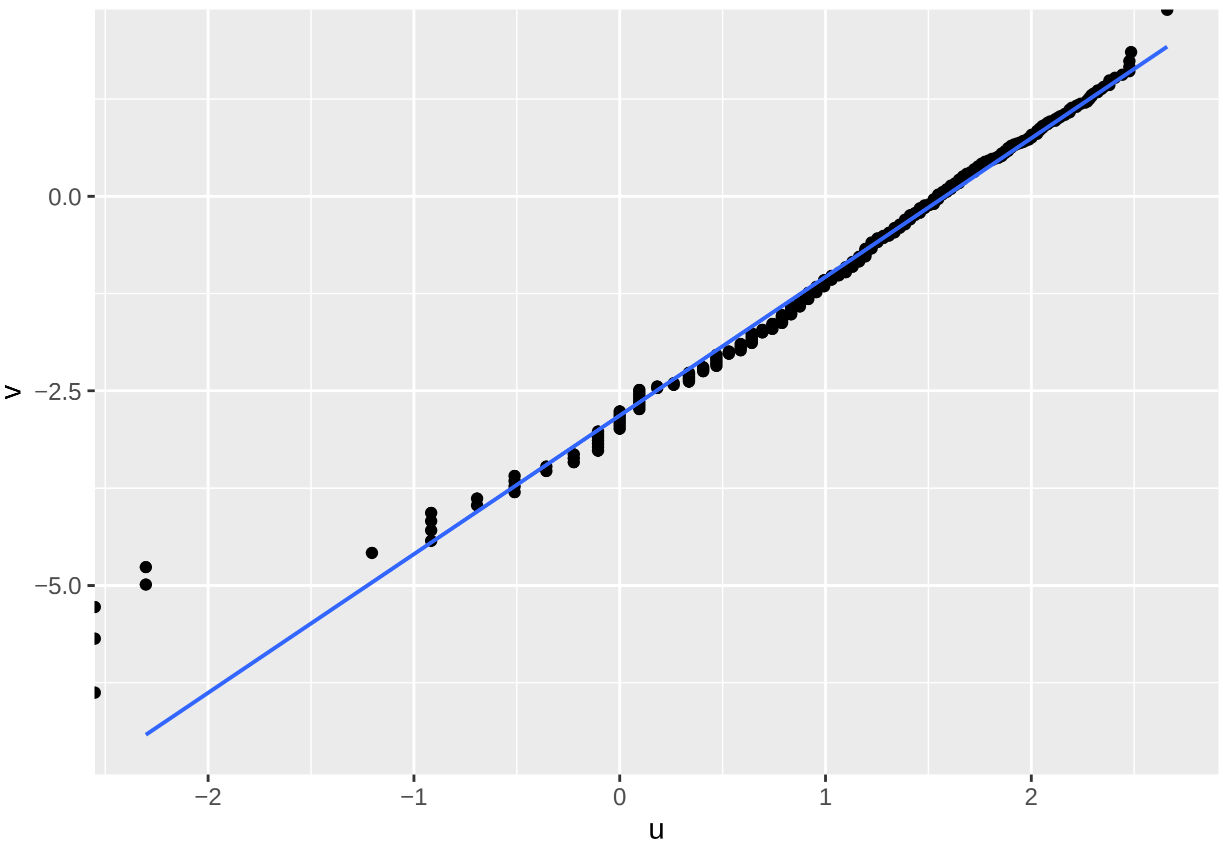
The picture shows that the log-log plot should produce a straight line if the points follow a weibull distribution. We can plot the points and see if true.

The code below computes a vector containing the values $v_i = \ln(-\ln(1 - \frac{i}{n}))$ and a vector containing the values $u_i = \ln(x_{(i)})$.

```r
n<-length(speed)
sortedSpeed<-sort(speed)
u<-log(sortedSpeed)
CDF<-(1:n)/n
v<-log(-log(1-CDF))
gf_point(v~u) %>%gf_lm()
```

```
## Warning: Removed 4 rows containing non-finite outside the scale range
## (`stat_lm()`).
```

```
## Warning: Using the `size` aesthetic with geom_line was deprecated in ggplot2 3.4.0.
## i Please use the `linewidth` aesthetic instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

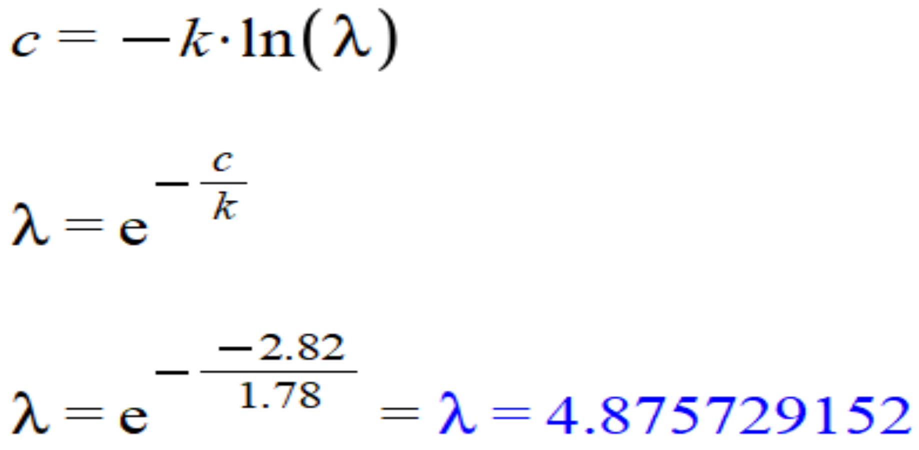


5. The intercept and slope of the line can be found to be -2.82 and 1.78, respectively. Use this to give estimates of the parameters k and λ of the model. Insert these values in the code below to plot the histogram together with the approximate density (`shape` is k and `scale` is λ).

We can find values of lambda and k from the straight line equation. We know from the picture on 2.4 that “k” is the slope of the line as this is the value that is multiplied by x. Our slope is then given as k = 1.78

The lambda value can be found from the intercept with the y-axis on the straight line as shown on the picture

```r
library("png")
pp <- readPNG("2.4.3.png")
plot.new()
rasterImage(pp,0,0,1,1)
```



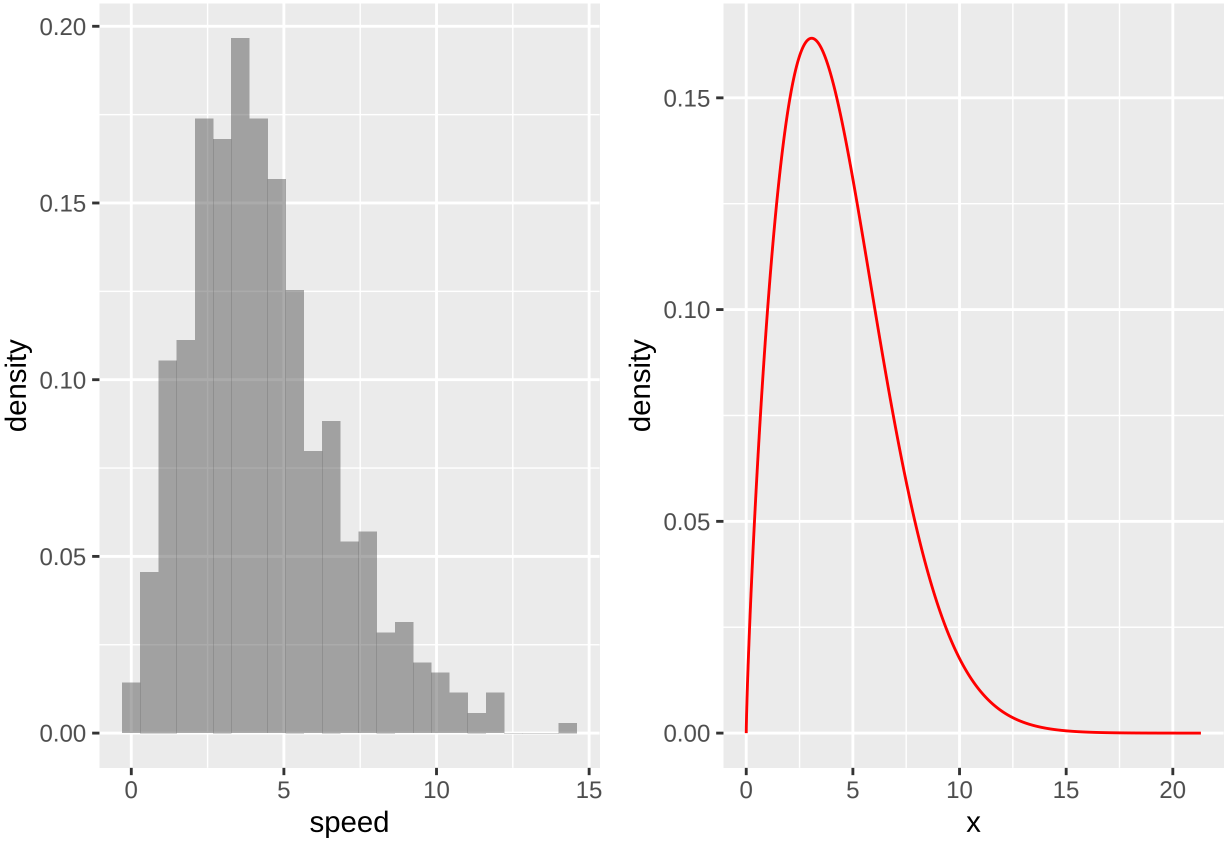
$$c = -k \cdot \ln(\lambda)$$

$$\lambda = e^{-\frac{c}{k}}$$

$$\lambda = e^{-\frac{-2.82}{1.78}} = \lambda = 4.875729152$$

Using k = 1.78 and lambda = 4.876 we can plot the approximate distribution along with the histogram

```
k1 <- 1.78
lambda1  <- 4.876
w5 <- gf_dhistogram( ~ speed, bins = 25) #%>%
w6 <- gf_dist("weibull", shape = k1, scale = lambda1, col = "red")
grid.arrange(w5,w6, ncol = 2)
```



We can see that the approximated distribution function looks very similar to the histogram.

Part III: Sample mean and the central limit theorem

In this last exercise, we investigate the distribution of the sample mean when a random sample is taken from a population having a `weibull`(k, λ) distribution. We will use the values of k and λ that you found in Part II, Exercise 5 to mimic a sample of wind speed measurements.

Denote by μ the mean of the population distribution, `weibull`(k, λ), and by σ^2 the variance of the population distribution.

The numeric values of μ and σ^2 for choices of λ and k can be calculated $\mu = \lambda\Gamma(1 + 1/k)$ and $\sigma^2 = \lambda^2 \left[\Gamma\left(1 + 2/k\right) - \left\{\Gamma\left(1 + 1/k\right)\right\}^2\right]$, where $\Gamma(x)$ denotes the gamma function.

1. Using the values of k and λ from Part II, Exercise 5, what is the mean and standard deviation? (Hint: You can use the function `gamma()` in R to compute the gamma function.)

We found that k = 1.78 and $\lambda = 4.876$ in part 2. And so, we simply plug in the numbers to find the population mean μ and standard distribution σ

```
#The values we found are
k <- 1.78
```

```
lambda <- 4.876

#For mean we get
u = lambda * gamma(1+(1/k))
#mu = 4.339
print(u)
```

```
## [1] 4.338598
```

```
#For variance we get
o_squared =  lambda^2 *(gamma(1+(2/k)) -(gamma(1+(1/k)))^2)
#Variance is 6.348

#Standard deviation will be
o = sqrt(o_squared)
#Standard deviation is 2.520
```

2. Suppose that a sample consists of 30 observations from this distribution. We denote the sample mean by `x_bar`. Using the central limit theorem, answer the following questions:

- What is the expected value of `x_bar`?
- What is the standard deviation of `x_bar` (also called the standard error)?
- What is the approximate distribution of `x_bar`?

The code below generates 30 independent realizations of a Weibull distribution with parameters k and λ. One may think of this of as simulated random sample of 30 independent wind speed observations.

```
#We generate the 30 samples with the found k and lambda values
k <- 1.78
lambda <- 4.876
#Generate random samples with rweibull()
x<-rweibull(30, shape=k, scale =lambda )
#We calculate the mean/expected value of the random variable. It is also xbar = (1/n)*sum(xi) for i = 1
x_bar <- mean(x)
print(x_bar)
```

```
## [1] 4.292447
```

The expected value/mean of x_bar is found above. We must find the sample variance S^2 in order to find the standard deviation of the sample. Sample variance is found with $S^2 = (1/(n-1))\sum_{i=1}^{n}(X_i - \bar{X})^2$

```
S_squared <- var(x)
print(S_squared)
```

```
## [1] 3.07216
```

To find the standard error of the mean of out sample relative to the population mean we must find the sample standard deviation. This can be done with $S = \sqrt{S^2}$. WE use the sd() command to do this.

```
S <- sd(x)
print(S)
```

```
## [1] 1.752758
```

We can now find the standard error of the mean $SEM = S/\sqrt{n}$. We have n = 30 samples

```
n <- 30
SEM = S/sqrt(n)
print(SEM)
```

```
## [1] 0.3200083
```

SEM is our standard error. This means that the true mean has a 68% probability of being inside this estimated mean +/- the standard error.

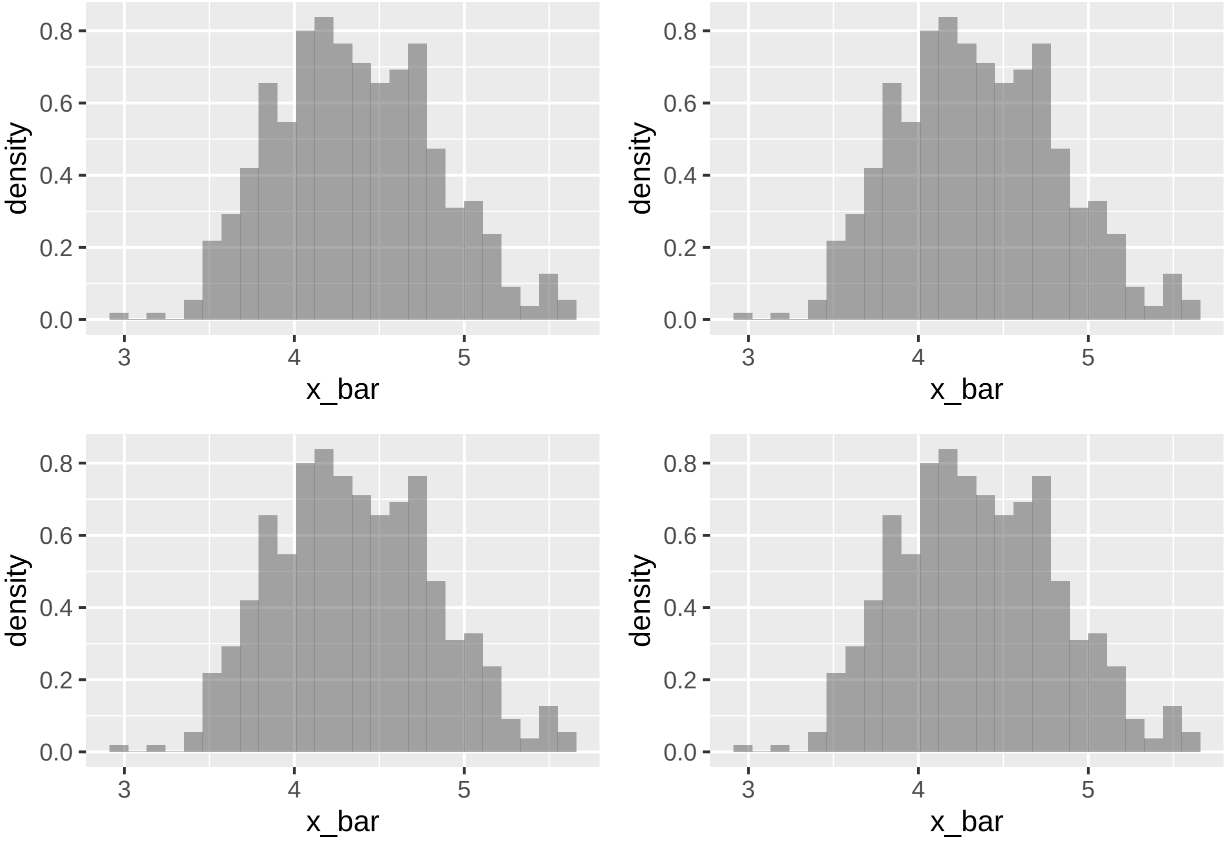
3. Insert the values of k and λ from Part II, Exercise 5 in the code. Run the command a few times. Is each sample mean close to what you expected?

Use `replicate` to repeat the sampling 500 times and save each mean value in the vector `x_bar`:

```
#We had the follow k and lambda
k=1.78
lambda = 4.876
#We generate 500 samples, calculate the mean and save the value in x_bar
x_bar <- replicate(500, mean(rweibull(30, shape=k, scale = lambda) ))
```

We expect it to be normally distributed. We can plot all the sample means in a histogram and see if this looks like a standard distribution.

```
x_bar1 <- replicate(500, mean(rweibull(30, shape=k, scale = lambda) ))
x_bar2 <- replicate(500, mean(rweibull(30, shape=k, scale = lambda) ))
x_bar3 <- replicate(500, mean(rweibull(30, shape=k, scale = lambda) ))
x_bar4 <- replicate(500, mean(rweibull(30, shape=k, scale = lambda) ))
hist1 <- gf_dhistogram( ~ x_bar, bins = 25)
hist2 <- gf_dhistogram( ~ x_bar, bins = 25)
hist3 <- gf_dhistogram( ~ x_bar, bins = 25)
hist4 <- gf_dhistogram( ~ x_bar, bins = 25)
grid.arrange(hist1, hist2, hist3, hist4, ncol = 2)
```



The means look like they are normally distributed around a mean of about 4.5. We know that population mean is $\mu = 4.339$ and we have a standard error of the mean of $SEM \approx 0.4127$. This means the true mean from the samples have 68% probability of being the interval $4.1 < x_bar < 4.9$. On the histogram we see a mean of about 4.5 so this is within the standard error and is what we would expect from a large sample. The mean from the samples is also very close to the population mean.

4. Calculate the mean and standard deviation of the values in `x_bar`. How do they match with what you expected?

We do this like in 2.2) with mean() and sd()

```
#We find the mean of 500 samples
mu_xbar1 <- mean(x_bar1)
print(mu_xbar1)
```

```
## [1] 4.36899
```

We find the standard deviation of xbar1

```
sd_xbar1 <- sd(x_bar1)
print(sd_xbar1)
```

```
## [1] 0.4626565
```

We can also find the standard error of the mean of 500 samples

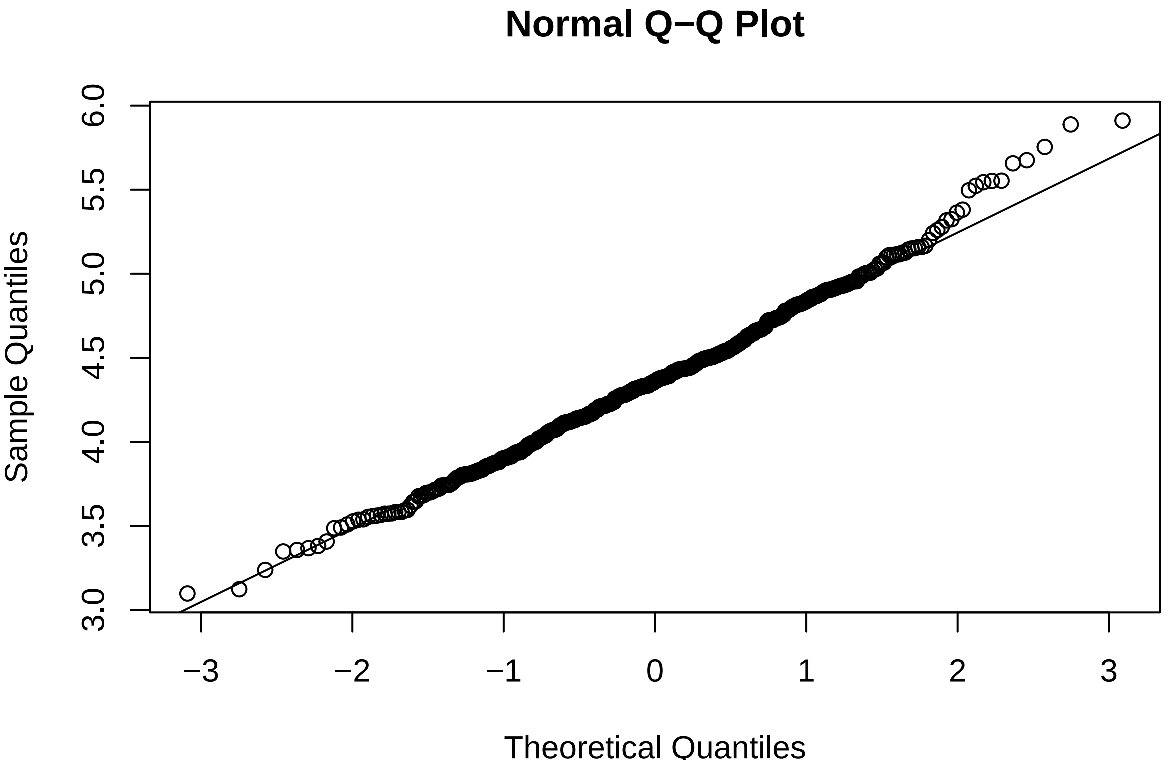
```
#We find the standard error of the mean of the 500 samples
SEM_xbar1 <- sd(x_bar1)/sqrt(500)
print(SEM_xbar1)
```

```
## [1] 0.02069063
```

The standard deviation and standard error of the mean are drastically reduced by increasing the sample size from 30 to 500 samples. The values at about $SEM \approx 0.0198$ is drastically reduced by increasing the samples size from 30 to 500 samples. This is expected as we are basing the SEM on a mean of 500 means.

5. Make a QQ-plot to assess the distribution of `x_bar`. Does this look like what you would expect?

```
#We plot the values of x_Bar1
qqnorm(x_bar1)
qqline(x_bar1)
```



The QQ plot shows a straight line as expected, because the random variable x_bar is normally distributed. The slope of the QQ plot corresponds to an approximation of the standard deviation of x_bar while the intersection of the y-axis corresponds to an approximation of the sample mean.

We find an estimate of the standard deviation from the slope of the line. We pick the x values 0 and 1 along with the y-values 4.4 and 4.75.

```
a <- (4.75-4.4)/(1-0)
print(a)
```

```
## [1] 0.35
```

We calculated the standard deviation of the 500 samples to be $sd \approx 0.44$ so an approximation of 0.35 is ok. The intercept at y = 2.8 is also an estimation of the mean, but this one is deviating from the expected 4.33 mu.