Cwiczenie 3 - Rozkłady empiryczne i teoretyczne

Michał Marosz

35 listopada 2015

Spis treści

Rozkład empiryczny i dystrybuanta empiryczna **6**

Estymacja parametrów rozkładów teoretycznych **8**

Zmienne dyskretne - rozkłady teoretyczne **15**

Rozkład Bernouli'ego 15

Rozkład Poisson'a 16

Zmienne ciągłe - rozkłady teoretyczne **19**

Rozkład Gauss'a 19

Rozkład Gamma 19

Rozkład Weibull'a 19

Rozkład GEV (Generalized Extreme Value) 19

Rozklady wykorzystywane często we wnioskowaniu statystycznym **20**

Rozkład t-Studenta 20

Rozklad χ^2 20

Rozkład Fishera-Snedecora 20

Przykład analizy z wykorzytaniem rozkładu Weibull'a **21**

Analiza właściwości zmiennych jest jednym z podstawowych zadan z jakimi przyjdzie sie Wam zmierzyć, w trakcie analizy danych. Dlatego rozpoczniemy od analizy rozkładów empirycznych a nastepnie wrowadzimy pojęcia i praktyczne zastosowanie dostepnych w R rozkladów teoretycznych, które najczęściej znajduja zastosowanie w analizach z zakresu Klimatologii. W R dostępnych jest cala gama rozkladów teoretycznych. Wystarczy w *pomocy* Rstudio wpisac *distributions* i otrzymamy dostęp do do informacji z tego zakresu.

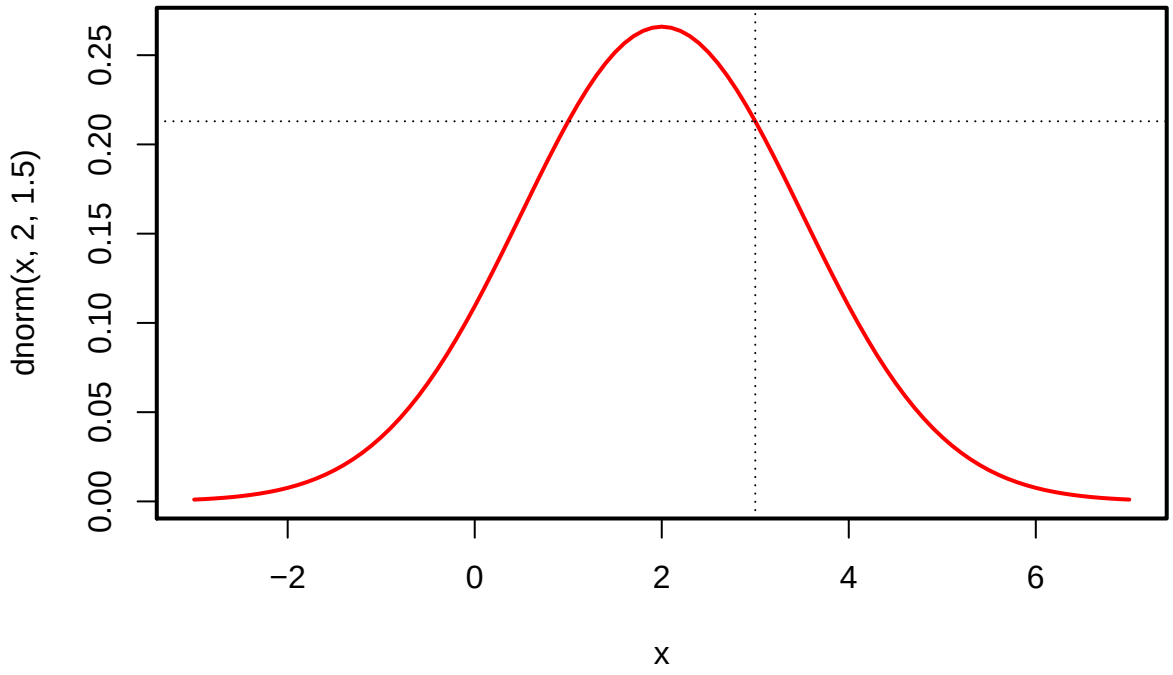
Istotnym jest aby nauczyć sie "zadawać" R odpowiednie *pytania* w analizach rozkladów. I tak, jeżeli chcemy uzyskać informację o wartości gęstości prawdopodobieństwa w określonym rozkładzie i dla konkretnej wartości skrót nazwy rozkładu poprzedzimy literą *d* (od Density) np. wpisując:

```
dnorm(3, 2, 1.5)
```

```
## [1] 0.2129653
```

jako rezultat otrzymamy prawdopodobieństwo wystapienia wartości zmiennej 3, jeżeli ma ona rozklad normalny (Gauss'a) o $\mu = 2$ i $\sigma = 1.5$. Można to graficznie przedstawic w sposob nastepujący:

```
curve(dnorm(x, 2, 1.5), xlim=c(-3, 7), col=2, lwd=2, add=F)
abline(v=3, h=dnorm(3, 2, 1.5), lty=3)
box(lwd=2)
```



Kolejną informację której możemy pożądać stanowi wartość prawdopodobieństwa, że przy założeniu okreslonego rozkładu przekroczona zostanie (lub też nie), określona wartość. Wówczas posłuzymy się przedrostkiem p np.:

```
pnorm(3, 2, 1.5)
```

```
## [1] 0.7475075
```

udzieli nam to odpowiedzi na pytanie jakie jest prawdopodobieństwo że w rozkładzie normalnym o $\mu = 2$ i $\sigma = 1.5$ wartość będzie niższa od 3. Jeżeli natomiast interesuje nas to, czy będzie ona wieksza uzyjemy dodatkowego argumentu *lower.tail=FALSE* np.:

```
pnorm(3, 2, 1.5, lower.tail = FALSE)
```

```
## [1] 0.2524925
```

można oczywiście wykorzystac poprzedni kod ale wynik trzeba odjąc od 1.

```r
1-pnorm(3, 2, 1.5)
```

```
## [1] 0.2524925
```

Np. analizzujemy rozkład wartości średniej dobowej temperatury powietrza dla jednego z miesięcy letnich. Załóżmy, ze jest to rozkład normalny o $\mu = 15$ i $\sigma = 3.2$. jakie jest prawdopodobieństwo, że średnia dobowa temperatura powietrza spadnie poniżej 10°C

```r
pnorm(10, 15, 3.2)
```

```
## [1] 0.05908512
```

albo ze przekroczy 22°C

```r
pnorm(22, 15, 3.2, lower.tail = FALSE)
```

```
## [1] 0.01435302
```

Można oczywiście przemnożyc wynik przez 100 i ladnie zaokrąglić, aby otrzymac wartość w %

```r
round(100*pnorm(22, 15, 3.2, lower.tail = FALSE), digits=1)
```

```
## [1] 1.4
```

Kolejnym z pytań które można zadawać R odnosi się do wartości o konkretnym prawdopodobieństwie przekroczenia, czyli innymi słowy poszukujemy wartości kwantyla. W tym wyapdku nasz predrostek to q a dla rozkładu normalnego o $\mu = 15$ i $\sigma = 3.2$ o wartość której prawdopodobieństwo przekroczenia wynosi 1% można "zapytać się" w sposób nastepujący:

```r
qnorm(0.99, 15, 3.2)
```

```
## [1] 22.44431
```

lub z uwgzlędnieniem argumentu *lower.tail*

```r
qnorm(0.01, 15, 3.2, lower.tail = FALSE)
```

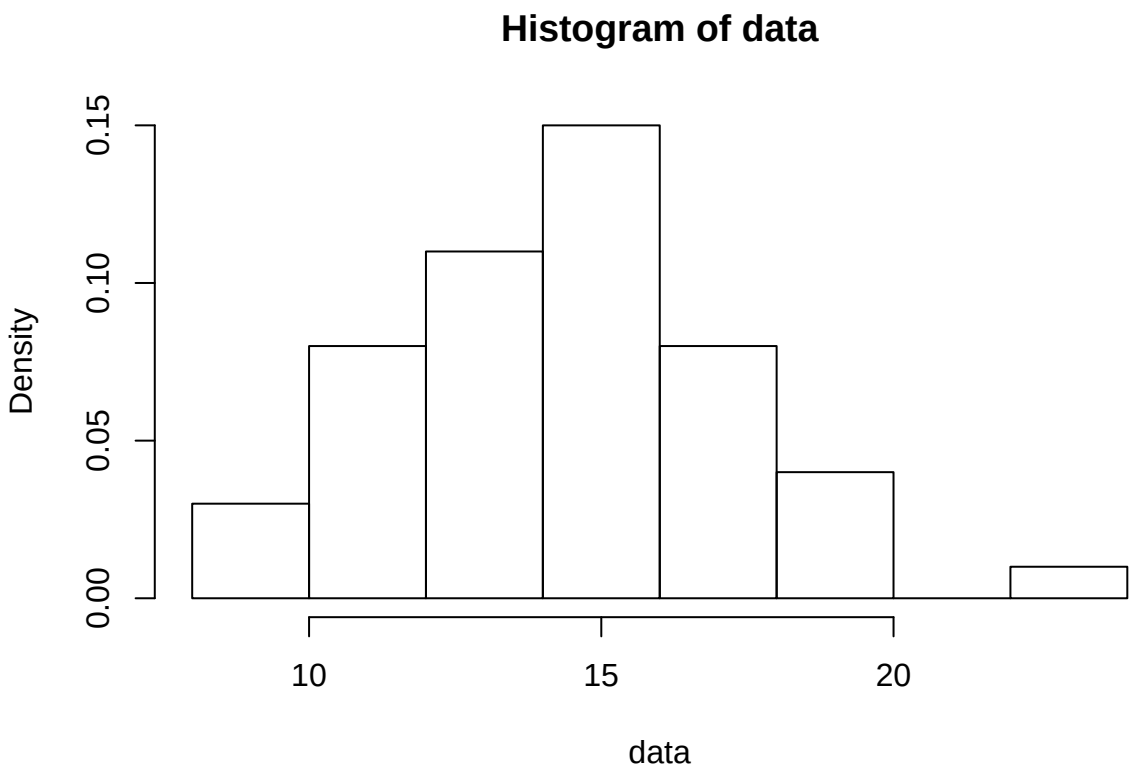
```
## [1] 22.44431
```

Rozkład empiryczny i dystrybuanta empiryczna

Rozkład empiryczny zazwyczaj przedstawia sie z wykorzystaniem histogramu natomiast dystrybuante empiryczną z wykorzystaniem funkcji *ecdf*

Utwórznmy wektor 100 losowych wartości o rozkładzie normalnym o $\mu = 15$ i $\sigma = 3.2$

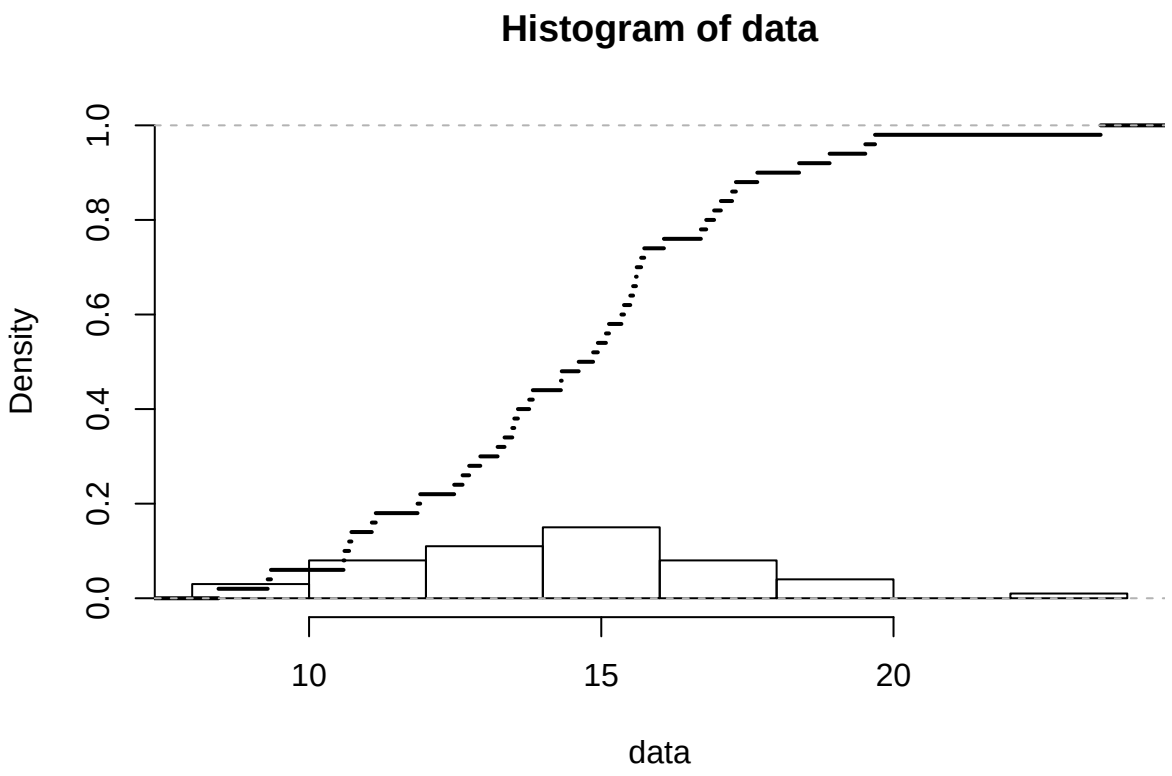
```r
set.seed(1000)
data=rnorm(50, 15, 3.2)
hist(data, prob=T)
```



Aby dodać do powyższego histogramu dystrybuantę empiryczną posłużymy sie funkcją *ecdf*

```
hist(data, prob=T, ylim=c(0,1))
plot(ecdf(data), vertical=FALSE, pch="", add=T, lwd=2)
```

```
## Warning in segments(ti.l, y, ti.r, y, col = col.hor, lty = lty, lwd =
## lwd, : 'vertical' nie jest parametrem graficznym
```



Estymacja parametrów rozkładów teoretycznych

W przypadku rozkładu normalnego (Gauss'a) estymacja parametrów nie nastręcza większych problemów. Średnia μ oraz odchylenie standardowe σ są wysttarczająco dorymi estymatorami parametrów rozkladu normalnego. Jednak dla pozostałych rozkladów niezbędne jest posłużenie sie dodatkowymi funkcjami pozwalającymi na dopasowanie parametrów rozkładów w tym celu niezbędne jest zainstalowanie *paczki fitdistplus*

Dopasujmy parametry rozkładu Gaussa za pomocą funkcji *fitdist* do temperatur powietrza w kwietniu.

```
dane=read.table("air.txt", header=T)
attach(dane)
temp04=TEMP[which(MC==4)]
hist(temp04)
```

policzmy

wartość średnią μ oraz odchylenie standardowe σ

```
mean(temp04)
```

```
## [1] 7.184375
```

```
sd(temp04)
```

```
## [1] 1.564408
```

Teraz spawwdzmy jakie wartości parametrów dopasuje funkcja *fitdis* z wykorzystaniem metody najwiekszej wiarygodności (*mle*), będącej standardem we współczesnych analizach.

Wpiszmy nastepujący kod:

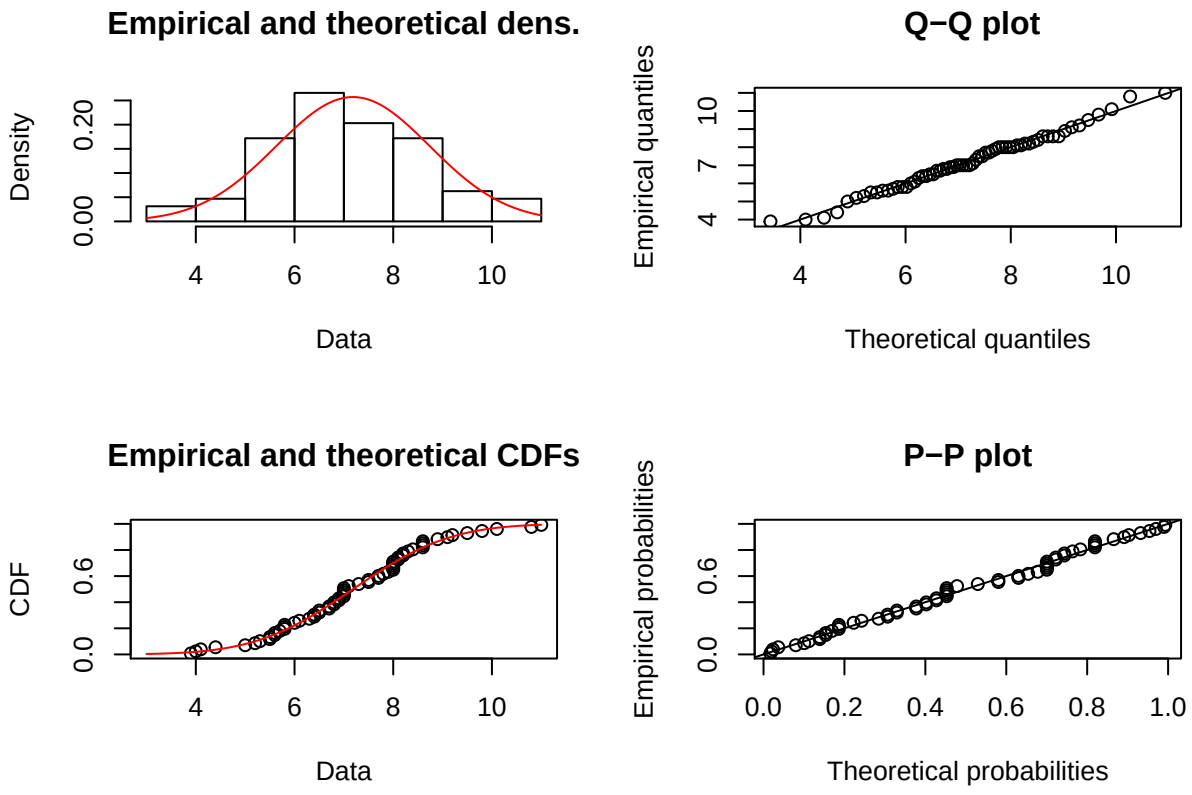
```
library("fitdistrplus", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.2")
```

```
## Loading required package: MASS
```

```
normal_fit = fitdist(temp04, dnorm, method = "mle")
```

Zwróćcie uwagę wynik analiz jest obiektem, ktory można poddać dalszej analizie w celu oceny dopadowania np.:

```
plot(normal_fit)
```



Można również "wyciągnąc" wartości parametrów ze zmiennej *estimate* będącej jedna ze składowych obiektu.

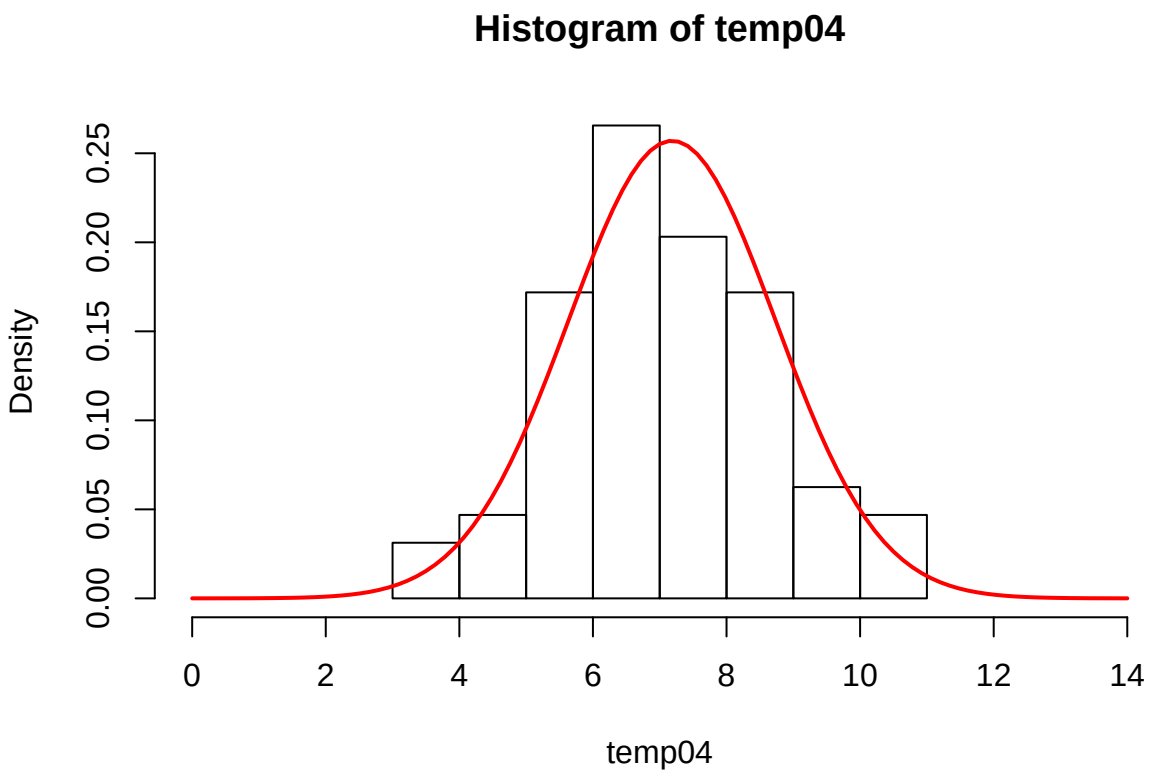
```
normal_fit$estimate
```

```
##     mean       sd
## 7.184375 1.552138
```

Teraz możemy wykreślic ponownie histogram i dodac do niego krzywą rozkładu normalnego wykreslona na podstawie dopasowanych parametrów. “Wypreparujmy” je najpierw z obiektu

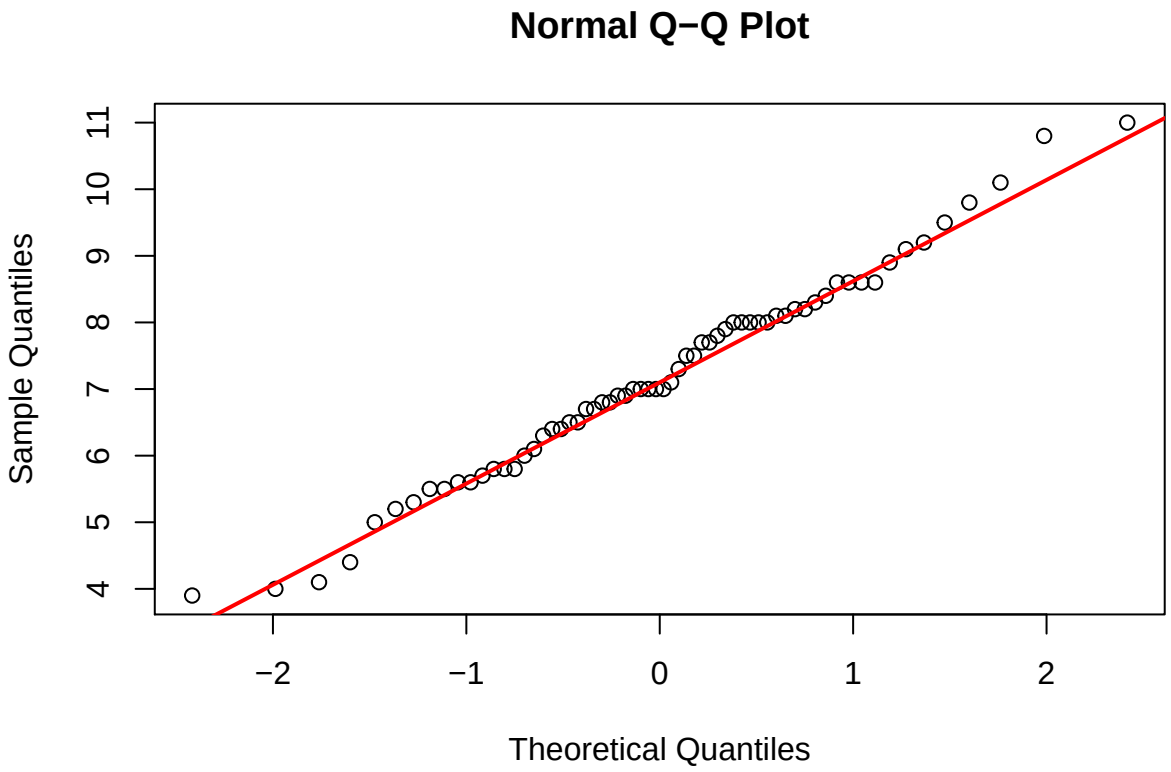
```
mean04=as.numeric(normal_fit$estimate[1])
sd04=as.numeric(normal_fit$estimate[2])
```

```
hist(temp04, prob=T, xlim=c(0,14))
curve(dnorm(x, mean04, sd04), add=T, col=2, lwd=2)
```



Istnieje również klasa wykresów przeznaczona specjalnie do wizualnego porównywania rozkadów empirycznych z teoretycznym normalnym: *qqnorm*

```r
qqnorm(temp04)
qqline(temp04, col=2, lwd=2)
```



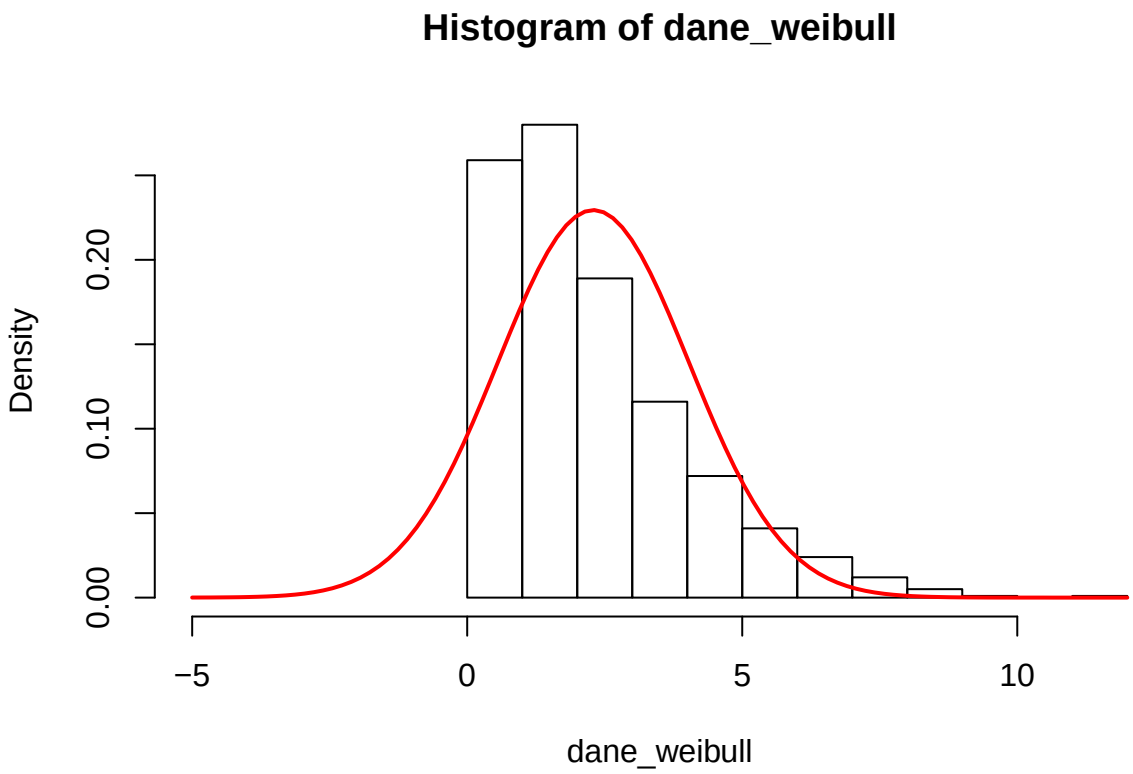
Weryfikację jakości dopasowania można przeprowadzić wizualnie za pomocą uprzednio wywołanej funkcji *plot/qqnorm* lub zaprząc do tego nieco bardziej sformalizowane testy np.: Shapiro-Wilk'a *shapiro.test*

```r
shapiro.test(temp04)
```

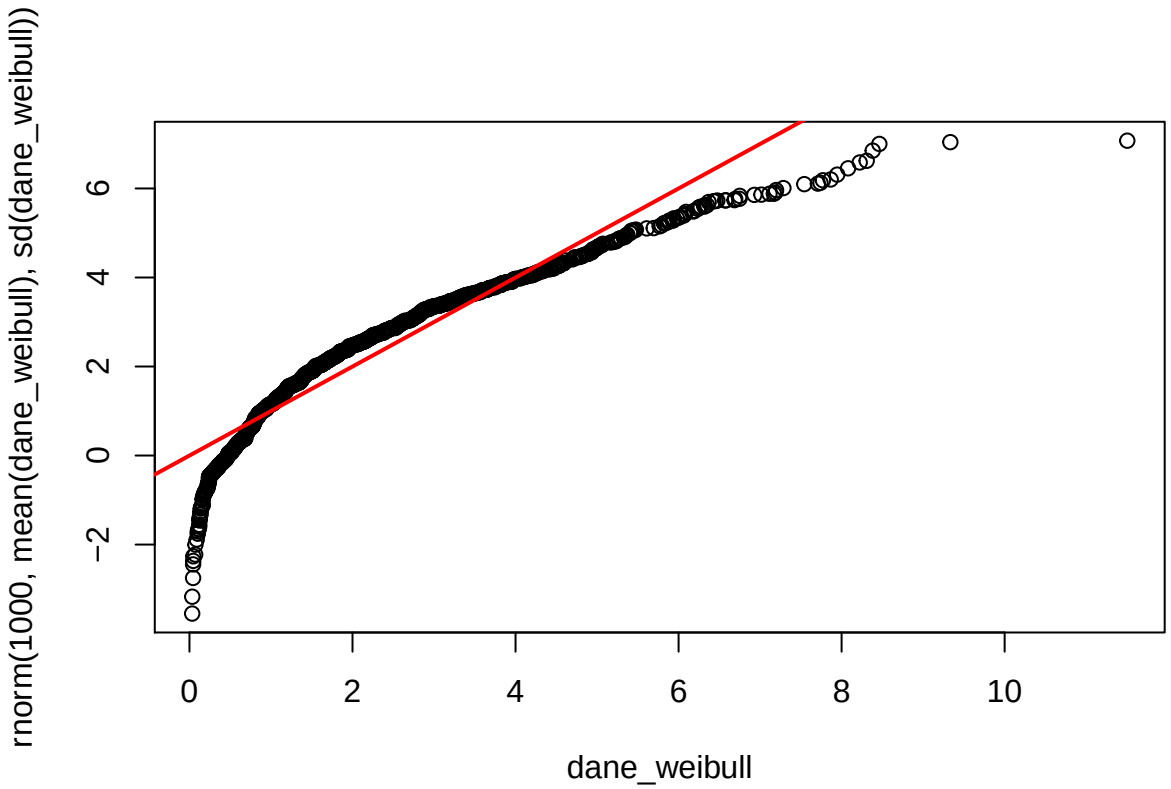
```
##
##  Shapiro-Wilk normality test
##
## data:  temp04
## W = 0.98799, p-value = 0.7913
```

Dla jasności wygenerujmy dane o rozkładzie innym niż normalny i sprawdzmy wykresy oraz wartości z powyższego testu.

```r
set.seed(1000)

dane_weibull=rweibull(1000, 1.3, 2.5)


hist(dane_weibull, prob=T, xlim=c(-5, 12))

curve(dnorm(x, mean(dane_weibull), sd(dane_weibull)), col=2, lwd=2, add=T)
```




```r
qqplot(dane_weibull, rnorm(1000, mean(dane_weibull), sd(dane_weibull)))

abline(0,1, col=2, lwd=2)
```




```
shapiro.test(dane_weibull)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  dane_weibull
## W = 0.90707, p-value < 2.2e-16
```

Zmienne dyskretne - rozkłady teoretyczne

Rozkład Bernouli'ego

Prawdopodobieństo osiągnięcia dokładnie k sukcesów w n próbach jest okreslone jako

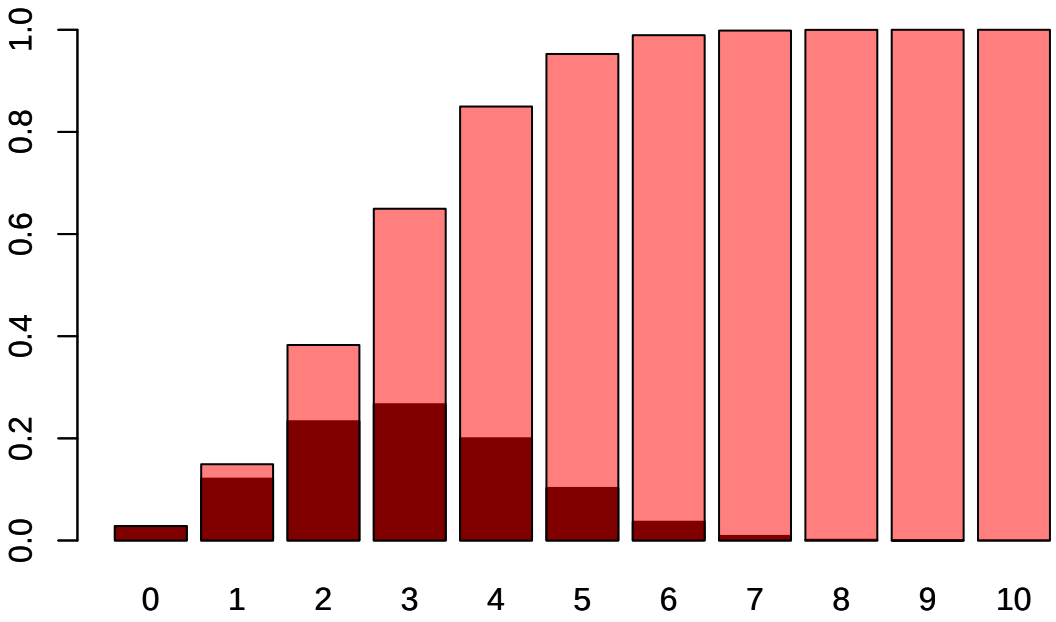
$$f(k; n, p) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

dla k = 0, 1, 2, ..., n, gdzie:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Na przykład jeżeli prawdopodobieństwo jednostkowego sukcesu wynosi 0.3 to wówczas PDF oraz CDF dla 10 prób wyglądają następująco:

```
barplot(dbinom(0:10, 10, 0.3), names=0:10, col="Black", ylim=c(0, 1))
barplot(pbinom(0:10, 10, 0.3), names=0:10,add=T, col=rgb(1,0,0,0.5))
```



Zadanie: Załóżmy, że posiadamy 150 letnią serię informacji o zamarzaniu akwenu X. W tym czasie wystąpiło 17 lat w których zamarzł on całkowicie uniemożliwiając żegluge. Wykorzystując rozklad dwumianowy oblicz

a) że zamarznie on w ciągu dekady 3 razy lub więcej

b) że zamarznie on conajwyżej 2 razy

Rozkład Poisson'a

$$f(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

gdzie:

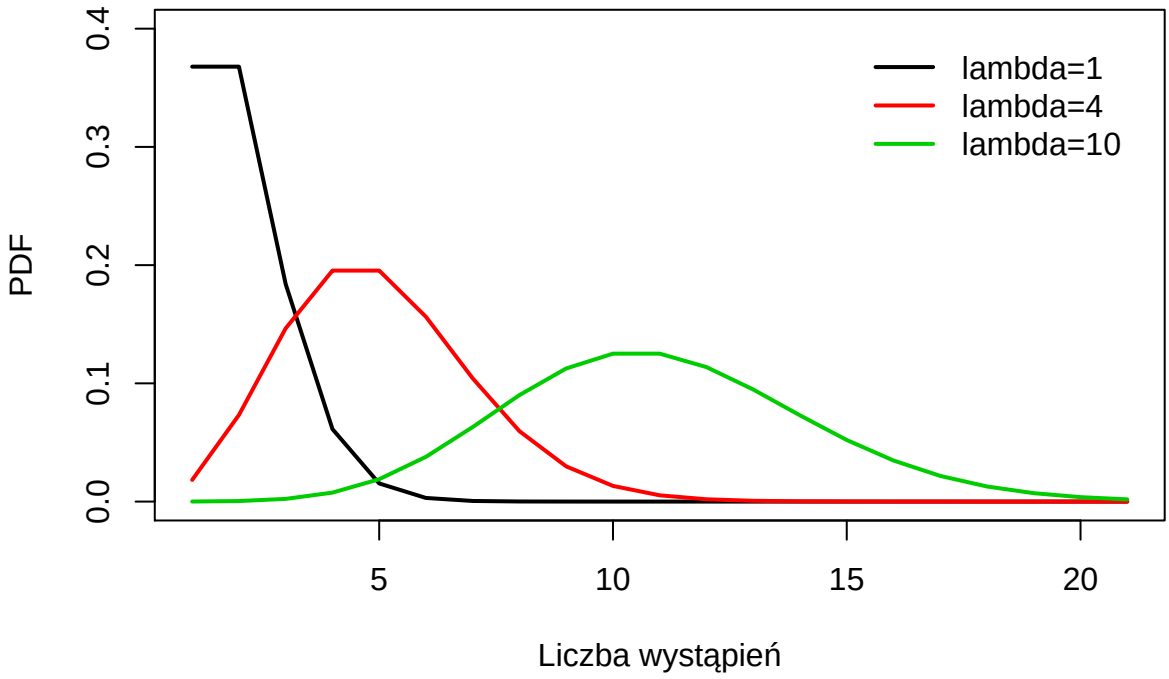
e jest podstawą logarytmu naturalnego (e = 2,71828 ...)

k jest liczbą wystąpień zdarzenia - prawdopodobieństwo, dane funkcją

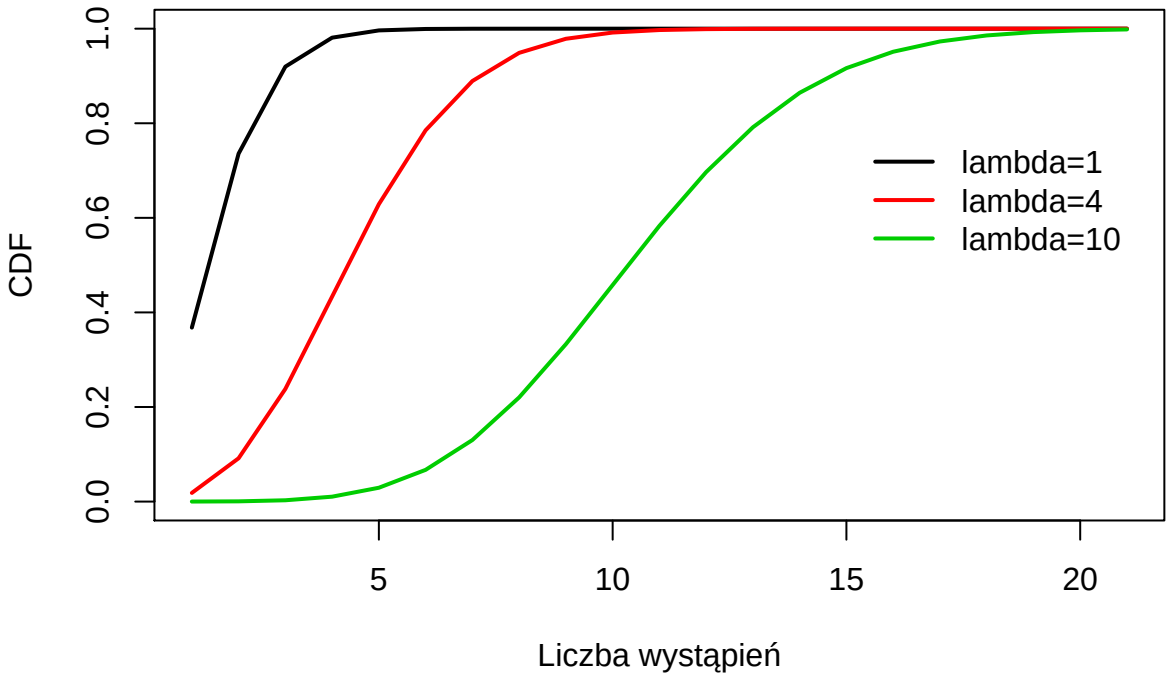
k! jest silnią k

λ jest dodatnią liczbą rzeczywistą, równą oczekiwanej liczbie zdarzeń w danym przedziale czasu

```
plot(dpois(0:20, 1), ylim=c(0,0.40), type="l", lwd=2, ylab="PDF",
     xlab="Liczba wystąpień")
lines(dpois(0:20, 4), col=2, lwd=2)
lines(dpois(0:20, 10), col=3, lwd=2)
legend(15, 0.4, legend=c("lambda=1", "lambda=4", "lambda=10"),
       lwd=c(2,2,2), col=c(1,2,3), box.lty = 0)
```




```
plot(ppois(0:20, 1), ylim=c(0,1), type="l", lwd=2, ylab="CDF",
     xlab="Liczba wystąpień")
lines(ppois(0:20, 4), col=2, lwd=2)
lines(ppois(0:20, 10), col=3, lwd=2)
legend(15, 0.8, legend=c("lambda=1", "lambda=4", "lambda=10"),
       lwd=c(2,2,2), col=c(1,2,3), box.lty = 0)
```



Zadanie:

Załóżmy że obserwacje wykazały, iż przeciętnie w sezonie letnim (VI-VIII) wystepuje 5,4 burze. Wykorzystując funkcje dostepne w R oblicz jakie jest prawdopodobieństwo, że w sezonie letnim wystapi ponad 10 burz.

Zmienne ciągłe - rozkłady teoretyczne

Rozkład Gauss'a

$$f(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Rozkład Gamma

$$f(x; k, \theta) = \frac{x^{k-1} e^{-\frac{x}{\theta}}}{\theta^k \Gamma(k)} \quad \text{for } x > 0 \text{ and } k, \theta > 0.$$

Rozkład Weibull'a

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

Rozkład GEV (Generalized Extreme Value)

Rozklady wykorzystywane często we wnioskowaniu statystycznym

Rozkład t-Studenta

Rozklad χ^2

Rozkład Fishera-Snedecora

Przykład analizy z wykorzytaniem rozkładu Weibull'a

Załóżmy, że naszym zadaniem jest dopasowanie parametrów rozkładu Wiebull'a do danych odnoszących się do prędkości wiatru w określonym punkcie.

Wczytajmy dane

```
dane = read.table("data.txt", header=T)
```

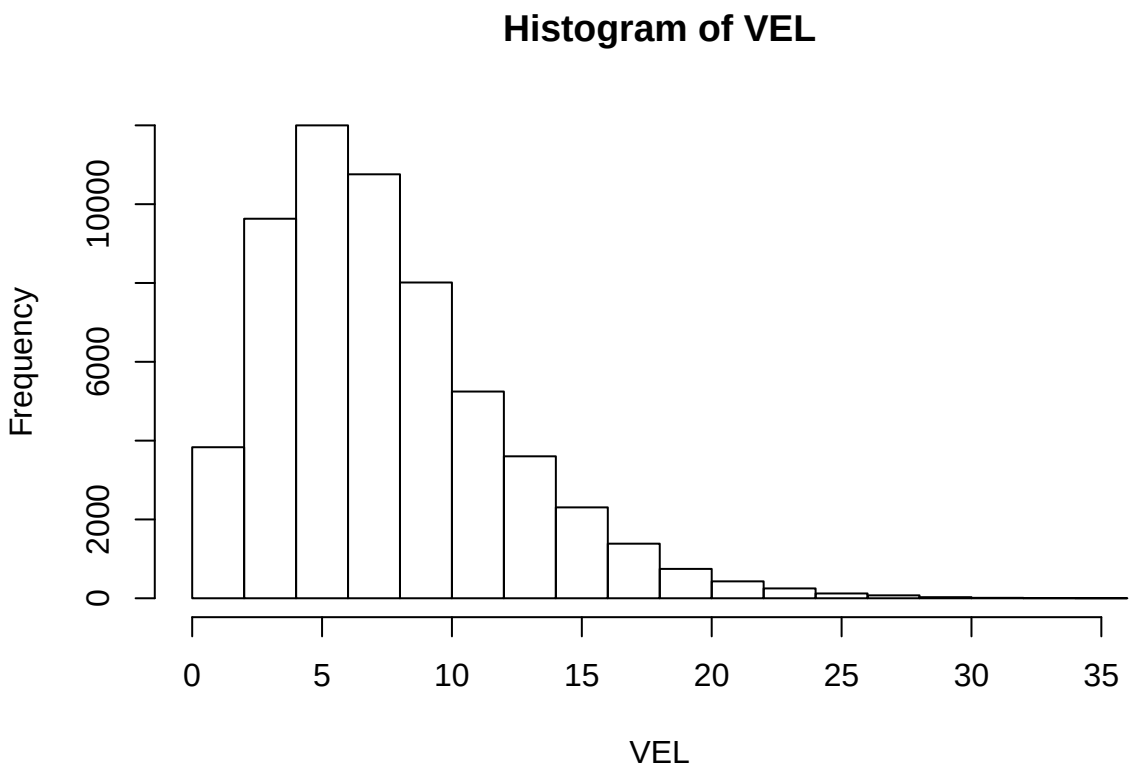
Dołączmy obiekt *dane* w celu łatwiejszego korzystania z zawartych w nim zmiennych

```
attach(dane)
```

```
## Następujący obiekt został zakryty z dane (pos = 5):
##
##     MC
```

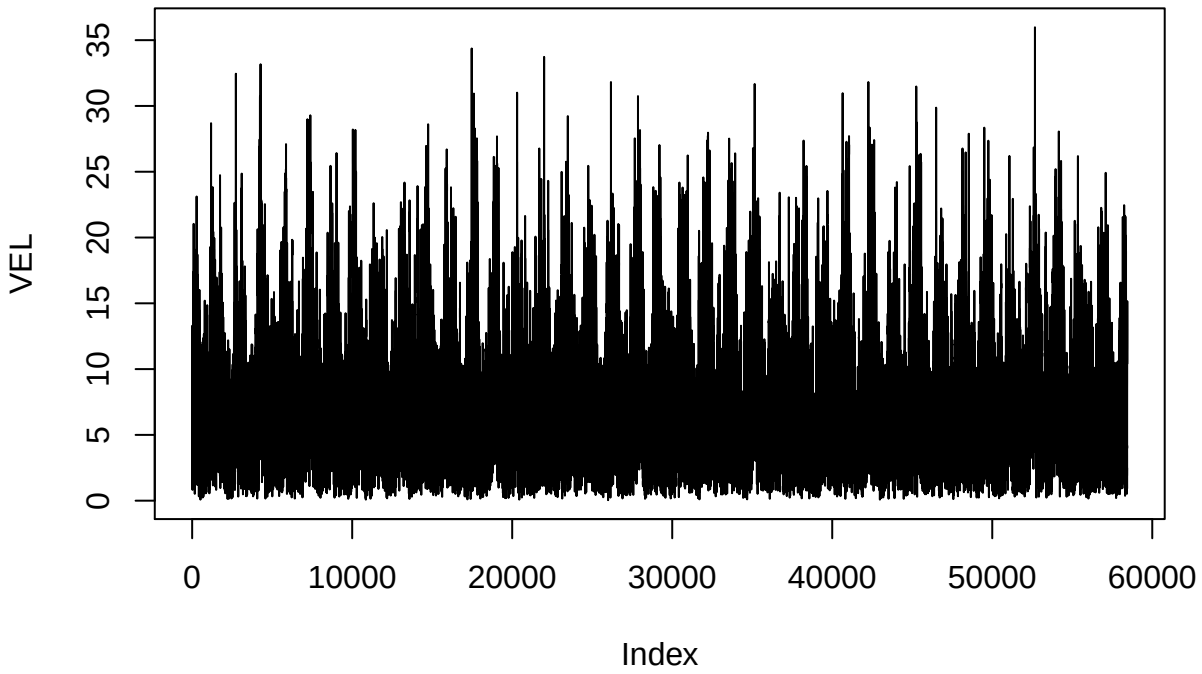
Zobaczmy jak wygląda rozklad empiryczny danych

```
hist(VEL)
```



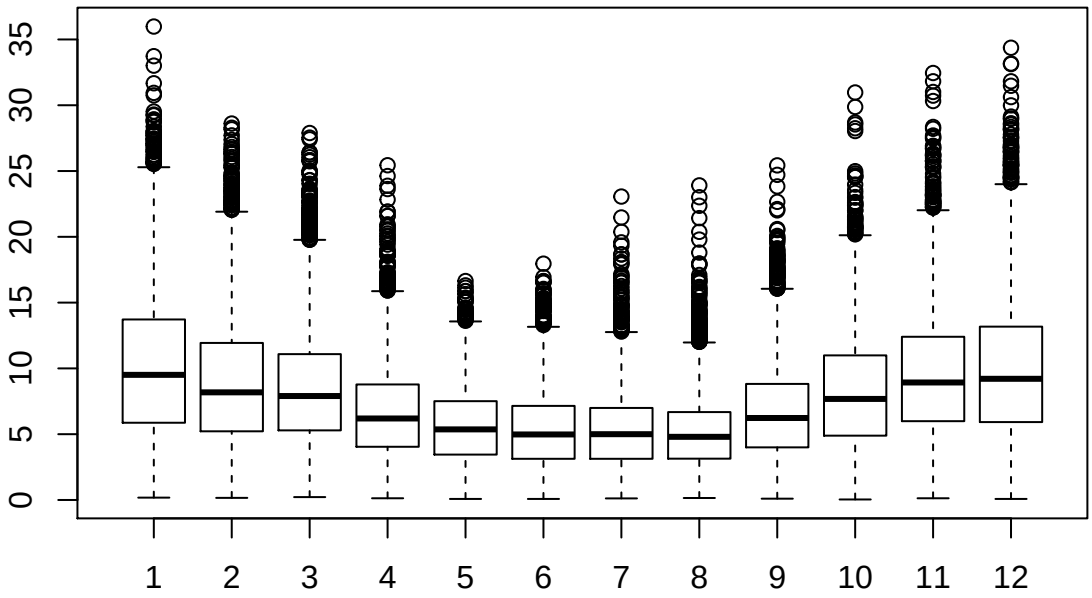
oraz wykres liniowy

```
plot(VEL, type="l")
```



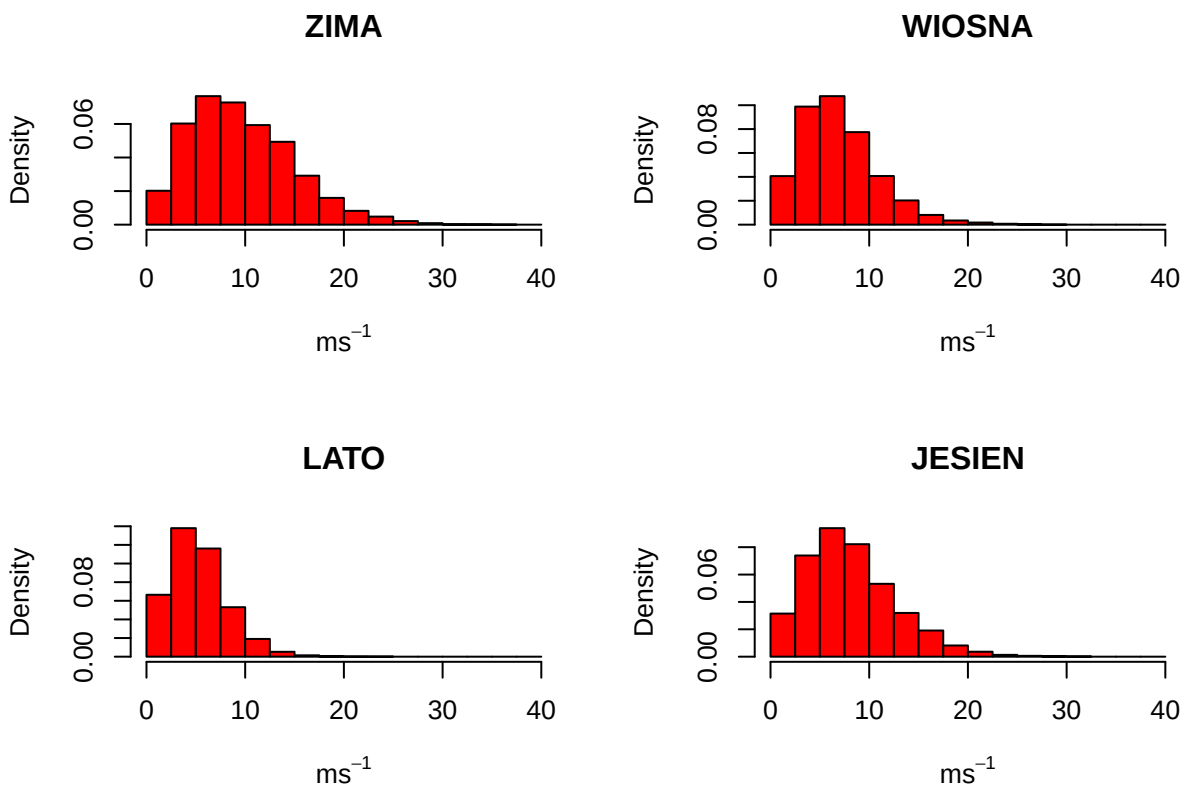
Cykl roczny danych

```
boxplot(VEL~MC)
```



Nauczmy się rysowac osobne histogramy dla pór roku z wykorzsytaniem pętli *for*

```
par(mfrow=c(2,2))
for(i in c("ZIMA", "WIOSNA", "LATO", "JESIEN"))
{
  hist(VEL[which(SEASON == i)], prob=T, main= i, xlab=expression(ms^-1),
       breaks=seq(0, 40, 2.5), col=2
       )
}
```



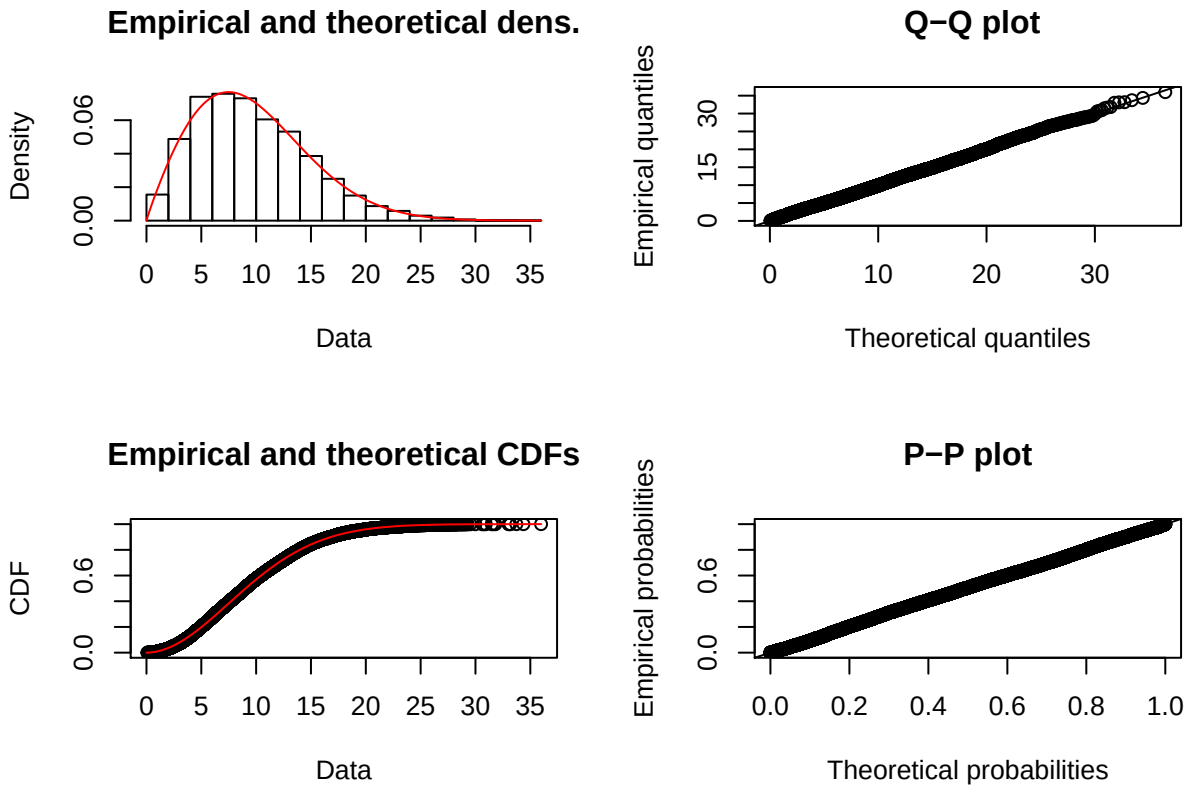
Dokonajmy przykładowego dopasowania i jego analizy dla sezonu zimowego.

Wczytajmy teraz bibliotekę *fitdistplus* w celu estymacji parametrów rozkladu Weibull'a.

Sprawdzmy uprzedimo w *helpie* jakie parametry będziemy estymować

```
fit_wei_DJF = fitdist(VEL[which(SEASON== "ZIMA")], distr = "weibull",
                      method = "mle")
```

```
plot(fit_wei_DJF)
```



Odczytanie parametrów odbywa się poprzez zapytanie odnośnie jednego z obiektów w dopasowanym modelu (obiekcie: fit_wei_DJF). Jeżeli rozwiniecie szczegóły zobaczycie, że jest tam obiekt *estimate* który składa się z dwóch liczb. Liczby te to dopasowane parametry rozkładu Weibull'a

```
fit_wei_DJF$estimate
```

```
##    shape    scale
##  1.93162 10.92883
```

Jak widać mamy dwa parametry: shape - kształtu oraz scale - skali.

aby wykorzystać je jako wartości liczbowe, np. w kresleniu dopasowanego rozkładu na histogramie, lub określaniu wartości kwantyli należy wykorzytac polecenie *as.numeric*

```
parametry_DJF = as.numeric(fit_wei_DJF$estimate)
```

do tego obiektu możemy odwoływac sie juz bezpośrednio

```
parametry_DJF
```

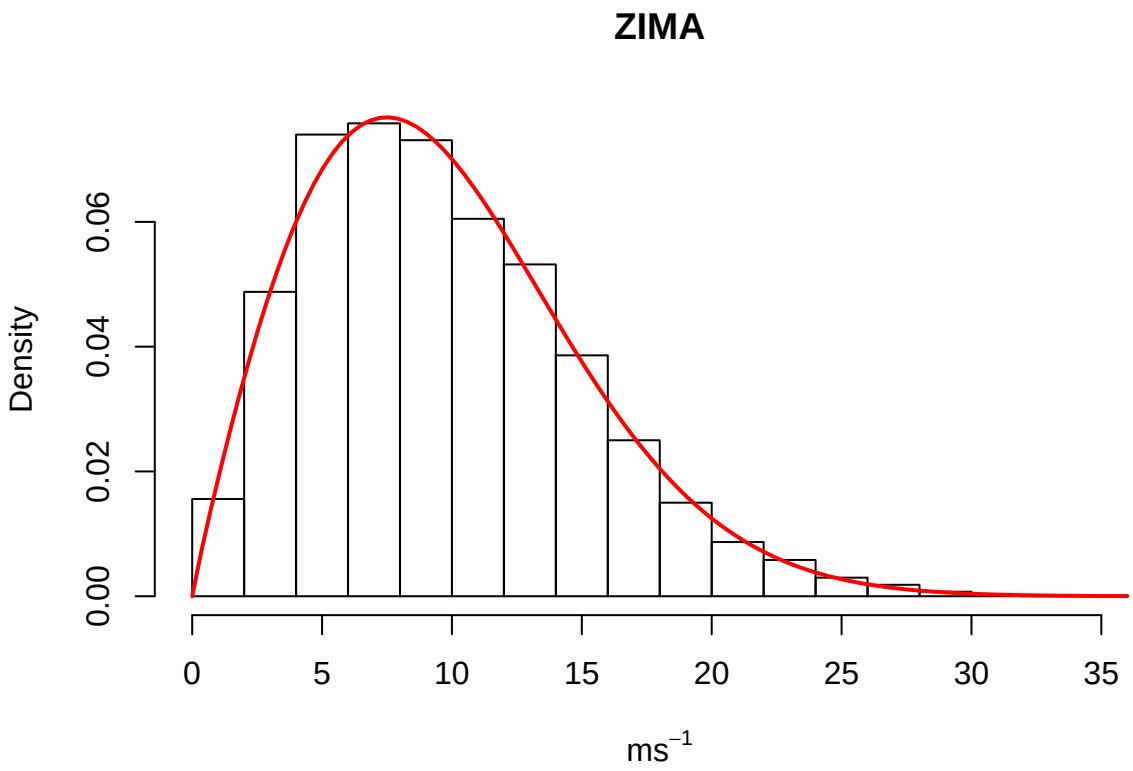
```
## [1]  1.93162 10.92883
```

Wykreślmy histogram z dopadowanym rozkładem Weibulla, resetując uprzednio parametry graficzne R.

```
dev.off()
```

```
## null device
##           1
```

```
hist(VEL[which(SEASON== "ZIMA")], prob=T, xlab=expression(ms^-1), main="ZIMA")
curve(dweibull(x, parametry_DJF[1], parametry_DJF[2]), add=T, col=2, lwd=2)
```



Jak widac, dopasowanie jest bardzo dobre.

Obliczmy w takim razie wartości kwantyli o okreslonym prawdopodobieństwie przekroczenia.

Np. 1%

```
qweibull(0.99, parametry_DJF[1], parametry_DJF[2])
```

```
## [1] 24.0955
```

Interpretacja: raz na sto przypadków prędkość wiatru przekroczy w sezonie zimowym 24ms-1.. Nalezy przy tym uwzględnić rozdzielczość czasową danych (w tym wyapdku 6h). Czyli w miesiącu mamy 124 (31 dniowy) lub 120 (30 dniowy) "pomiarów". Tak więc takie wartości będa przekraczane przynajniej raz w miesiącu.

Można oczywiście obliczyć kwantyle wyższych rzędów np.: 0.999, 0.9999. Mozna to wykonać "za jednym zamachem" wpisując określone wartości bezpośrednio w funkcji

```
qweibull(c(0.99, 0.999, 0.9999), parametry_DJF[1], parametry_DJF[2])
```

```
## [1] 24.09550 29.72340 34.49684
```

Otrzymamy wowczas wartości kwantyli o prawdopodobieństwie przekroczenia: 1%, 0.1% oraz 0.01% czyli takich które wystapią 1/100, 1/1000 oraz 1/10000 pomiarów.

Zadanie: Na podstawie otrzymanych danych. Dopasuj parametry rozkladu Weibull'a dla poszczególnych miesięcy oraz dokonaj analizy wystepowania ekstremalnych wartości prędkości wiatru w cyklu rocznym. Wykorzystaj R, przygotuj wykresy i tabele, a analize graficzną okraś opisem na 2000 znaków.