

Instalacja programu R wraz z konsolą.

Aby program R poprawnie działał i można w nim było przeprowadzać obliczenia, potrzebujemy zainstalować dwa programy. Jeden z nich odpowiada za środowisko, w którym będziemy tworzyć programy, z kolei drugi z nich jest narzędziem, w skład którego wchodzi zbiór zintegrowanych narzędzi do pomocy w trakcie pracy z językiem R. Pomaga on nam w formie graficznej, udostępniając konsolę do wprowadzania linii kodu programów, sprawdzanie ich poprawności oraz rysowanie wykresów i histogramów.

I program: Microsoft R Application Network

Najbardziej optymalnym programem (jest ich więcej niż jeden) jest właśnie program firmy Microsoft, jest on całkowicie darmowy i w pełni wspierany dla każdego systemu Windows. Wykorzystuje on wszystkie rdzenie do obliczeń (w naszych programach jego lepsza prędkość obliczeń nie będzie aż tak zauważalna, ale przy programach działających kilka dni mogą potwierdzić, że jest o wiele bardziej wydajny). Poniżej znajdziecie Państwo link do strony, z której możemy pobrać program:

<https://mran.microsoft.com/open>

Dla osób, które nie mają systemu Windows zainstalowanego na swoim komputerze, zamiast powyższego programu, możecie Państwo ściągnąć następujący:

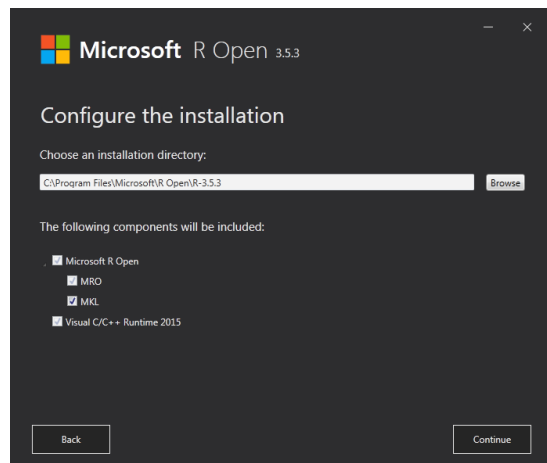
<https://www.r-project.org/>

II program: R-Studio

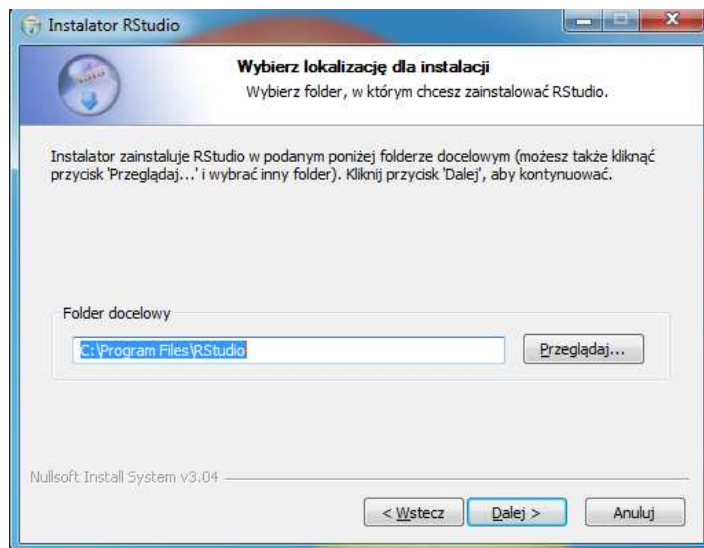
Drugim programem, który jest potrzebny w celu poprawnym posługiwaniem się językiem R jest właśnie R Studio. To w nim będą powstawać nasze programy i w nim będziemy tworzyć wykresy, czy histogramy. Ten pomocniczy program będzie korzystał z mocy obliczeniowej pierwszego programu. Poniżej podaję link do strony, z której możecie Państwo pobrać również darmowy program R-Studio:

<https://rstudio.com/products/rstudio/download/#download>

Na początku instalujemy pierwszy z programów: Microsoft R Application Network. Wybieramy ścieżkę, gdzie instalator zainstaluje program:



Dalej oczywiście akceptujemy warunki licencji użytkowania i instalujemy. W kolejnym kroku należy zainstalować R Studio. Ważne, aby z linku, który Państwu podałem wybrać taką wersję programu jaki mamy system (Windows 32-bitowy, 64-bitowy i jaka wersja systemu; gdyby był problem służę pomocą). Znowu wybieramy ścieżkę dostępu:



Również wybieramy folder, w jakim zostanie zainstalowany. Domyślnie mamy folder R Studio i taki też może pozostać. Instalujemy. Jeśli pojawi się problem podczas instalowania tych dwóch programów, od razu proszę o kontakt.

Aby zacząć pracować w programie należy otworzyć właśnie program R Studio. Programu Microsoft R Open nie otwieramy wcale. Gdy otwieramy program R Studio na konsoli od razu widzimy, że program będzie korzystał z mocy obliczeniowej języka R przez Microsoft R Open:

```
Console Terminal x Jobs x
~/

R version 3.5.3 (2019-03-11) -- "Great Truth"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R jest oprogramowaniem darmowym i dostarczany jest BEZ JAKIEJKOLWIEK GWARANCJI.
Możesz go rozpowszechniać pod pewnymi warunkami.
wpisz 'license()' lub 'licence()' aby uzyskać szczegółowy dystrybucji.

R jest projektem kolaboracyjnym z wieloma uczestnikami.
wpisz 'contributors()' aby uzyskać więcej informacji oraz
'citation()' aby dowiedzieć się jak cytować R lub pakiety R w publikacjach.

wpisz 'demo()' aby zobaczyć demo, 'help()' aby uzyskać pomoc on-line, lub
'help.start()' aby uzyskać pomoc w przeglądarce HTML.
wpisz 'q()' aby wyjść z R.

Microsoft R Open 3.5.3
The enhanced R distribution from Microsoft
Microsoft packages Copyright (C) 2019 Microsoft Corporation

Using the Intel MKL for parallel mathematical computing (using 2 cores).

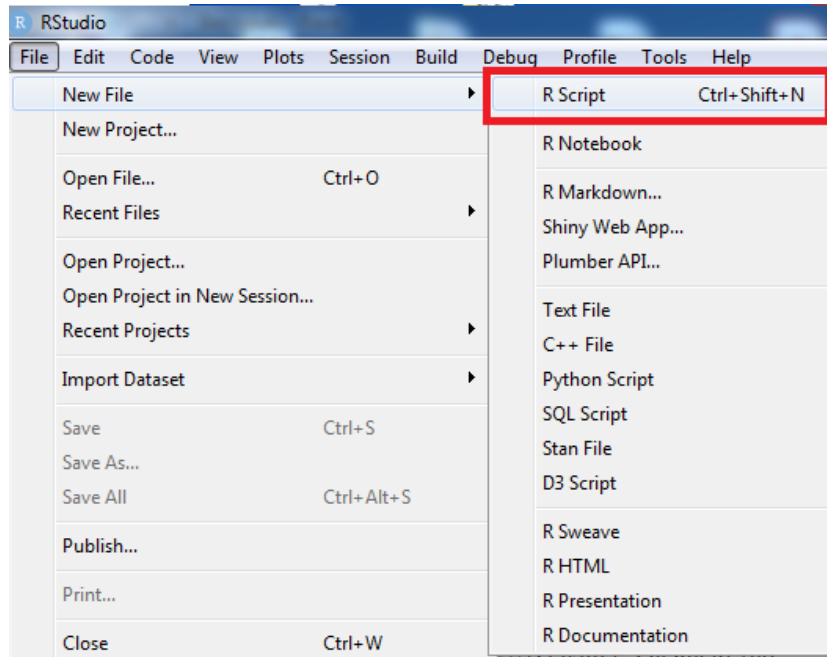
Default CRAN mirror snapshot taken on 2019-04-15.
see: https://mran.microsoft.com/.

[workspace loaded from ~/.RData]
```

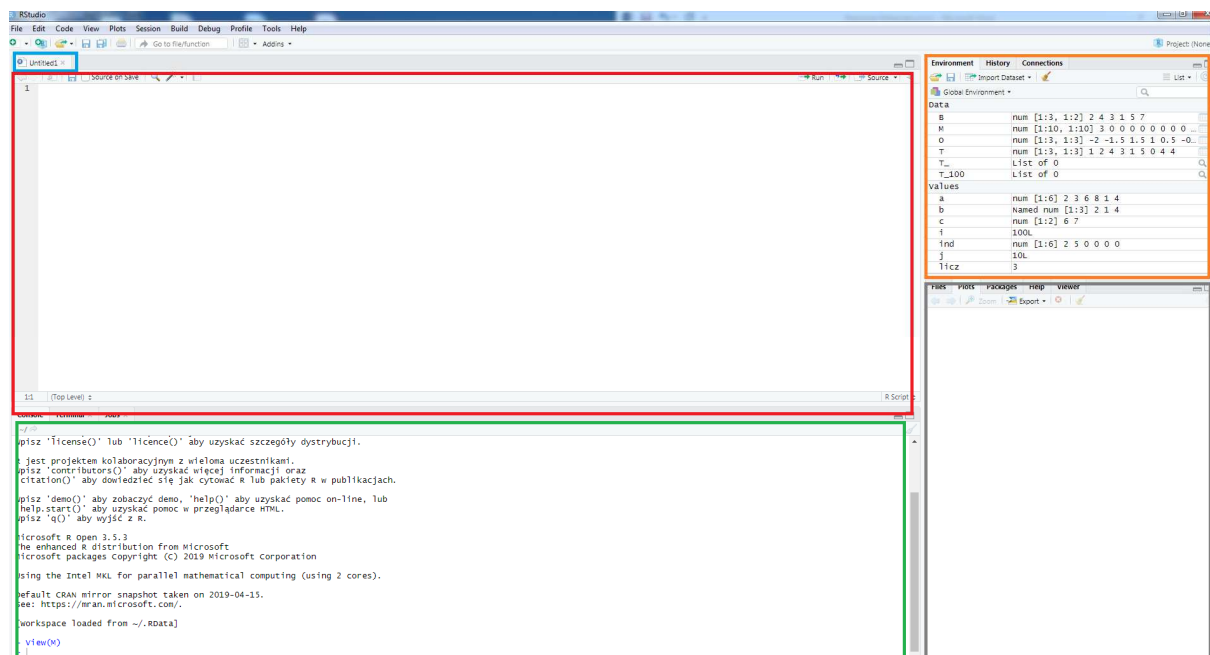
Zapoznanie się z programem

Na samym początku chciałbym Państwu przedstawić podstawy programu jak i języka R. Wszystko będzie wyjaśnione krok po kroku, więc myślę, że nie będzie problemów dla Państwa również z tym przedmiotem.

Aby zacząć nowy plik z programem należy utworzyć nowy skrypt. Wykonujemy to poprzez opcję na pasku menu na samej górze programu. Wybieramy opcję File, dalej New File i na końcu wybieramy pierwszą od góry opcję R Script:



Po takiej aktywności powinniśmy otrzymać nasz program w postaci jaką pokazano poniżej:

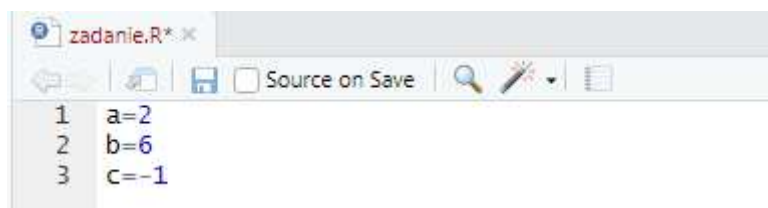


Ramkami różnych kolorów zazaczyłem najbardziej istotne elementy dla umiejętności posługiwania się programem R Studio. Wyjaśnijmy je po kolei. W ramce niebieskiej (górny lewy narożnik) mamy nazwę naszego pliku z programem, który powstał poprzez utworzenie nowego R skryptu. Jeśli jeszcze go nie zapisaliśmy, oczywiście posiada nazwę Untitled z pewnym numerem (w zależności ile takich plików posiadamy). Aby zapisać nasz program, możemy to zrobić poprzez kombinację klawiszy Ctrl+S lub z menu File wybrać opcję Save As. Pliki zapisują się z rozszerzeniem R. W czerwonym obszarze znajduje się miejsce do wpisania linijek kodu naszego programu. To tutaj mamy przestrzeń roboczą, gdzie powinien znaleźć się nasz program (analogicznie do programu MatLab). Wszystkie nasze programy będą tworzone właśnie w tym miejscu.

Zielony prostokąt to konsola, gdzie będą wyświetlane wyniki każdej procedury naszego programu. Także tutaj możemy wpisać jedną komendę bądź wykonać pewne działanie. Ramka pomarańczowa to przede wszystkim spis zmiennych jakie są wykorzystywane i zadeklarowane w naszym programie. Będziemy w stanie dostrzec ich nazwę, typ oraz rozmiar. Ostatnia część, szary obszar to również kilka istotnych aspektów. Tutaj będą nam się pojawiać wykresy bądź histogramy. Tutaj też będziemy mogli dodatkowo dołączyć odpowiedni pakiet do konkretnych obliczeń (w trakcie dowiemy się, że jest również inna możliwość) oraz także w tym okienku będziemy w stanie zajrzeć do pomocy w celu sprawdzenia nazw funkcji bądź ich argumentów.

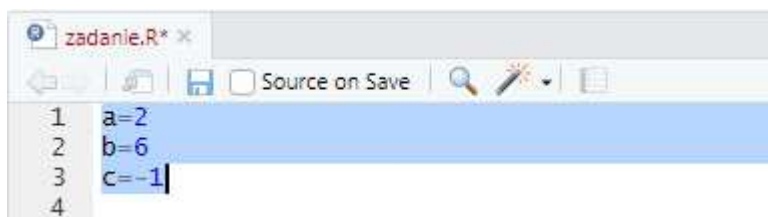
Z pewnością najlepiej przećwiczyć sam program na przykładach. Przepraszam, że są one tak łatwe, ale mają pełnić tylko i wyłącznie funkcję poznawczą z programem 😊. Zapiszmy nasz program pod nazwą zadanie.

W pierwszych trzech linijkach kodu, przypiszmy do zmiennych *a, b, c* wartości 2, 6 oraz -1. Przypisanie wartości odbywa się za pomocą pojedynczego znaku równości. Wówczas mamy:



```
1 a=2
2 b=6
3 c=-1
```

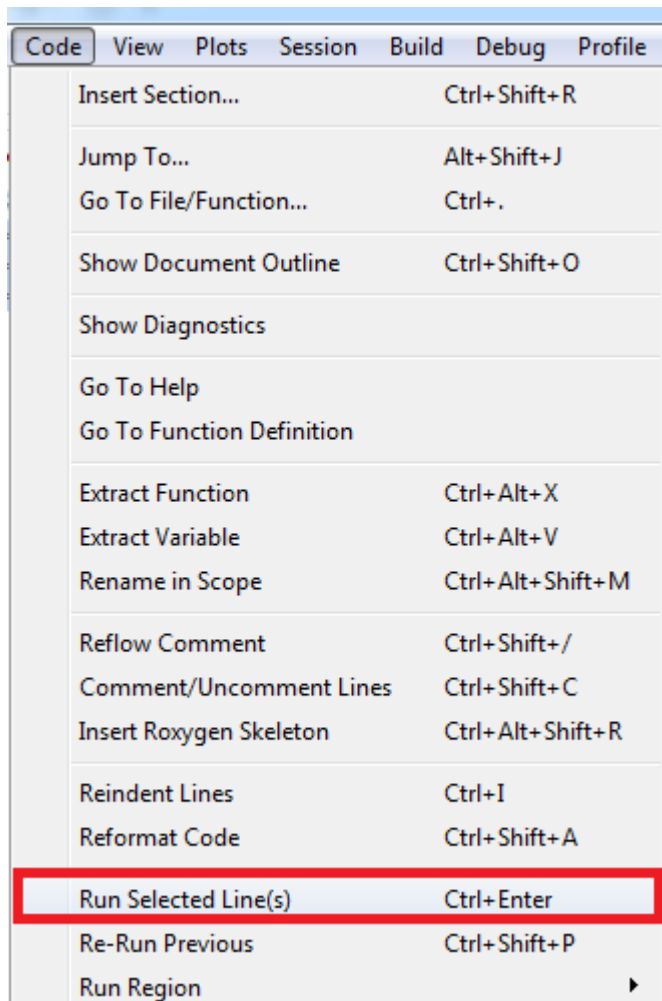
Jak uruchomić program, tzn. w jaki sposób uruchomić pojedynczą linijkę kodu lub większą część programu? Aby uruchomić linijkę kodu, w której znajduje się aktualnie kursor należy przytrzymać klawisz Ctrl oraz wcisnąć klawisz Enter. Wówczas dana linijka programu zostanie wykonana, kursor sam przejdzie do kolejnej linii. Możemy również zaznaczyć wszystkie trzy linie, które chcemy aby były wykonane za jednym razem i również zastosować kombinację klawiszy Ctrl+Enter.



```
1 a=2
2 b=6
3 c=-1
4
```

To samo możemy również wykonać przy pomocy opcji z paska menu znajdującego się na samej górze programu. Gdy wejdziemy do opcji Code, tak wybrać możemy Run Selected Line(s), efektem czego będzie uruchomienie aktywnej linii programu bądź większej ich liczby, jeśli są zaznaczone.

Wydaję mi się jednak, że pierwszy sposób jest o wiele szybszy i wygodniejszy, ale pozostawiam to dla Państwa do wyboru:



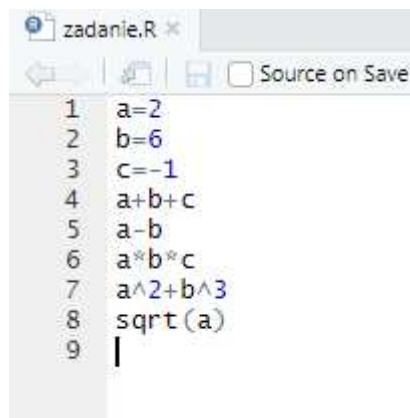
Po każdej tego typu aktywności, widzimy w konsoli znajdującej się poniżej, że dana linijka programu faktycznie została aktywowana. Zauważmy również, że po uruchomieniu programu faktycznie otrzymaliśmy trzy różne zmienne, które już są zdefiniowane w naszym środowisku:

values	
a	2
b	6
c	-1

Dalej, w następnej linii dodajmy do siebie wszystkie trzy wartości. Wówczas w konsoli dostaniemy na początku informację, którą procedurę wywołaliśmy, a dalej otrzymamy wynik tej procedury:

```
> a+b+c  
[1] 7  
> |
```

W kolejnych wierszach programu wprowadźmy jeszcze kilka przykładowych procedur i funkcji. Niech będzie tam uwzględnione odejmowanie zmiennych, ich iloczyn, potęgowanie, a także pierwiastkowanie. Wówczas nasz program prezentuje się następująco:



```
1 a=2
2 b=6
3 c=-1
4 a+b+c
5 a-b
6 a*b*c
7 a^2+b^3
8 sqrt(a)
9 |
```

A wyniki, które otrzymaliśmy z konsoli są następujące:

```
> a=2
> b=6
> c=-1
> a+b+c
[1] 7
> a-b
[1] -4
> a*b*c
[1] -12
> a^2+b^3
[1] 220
> sqrt(a)
[1] 1.414214
> |
```

Są to najbardziej podstawowe funkcje. Mam nadzieję, że dotychczas jest dla Państwo wszystko zrozumiałe. Poniżej (jak i w osobnym pliku, jaki Państwu wysłałem) przedstawione są najczęściej używane funkcje. Oczywiście nie musicie Państwo tego umieć na pamięć, nawet jeśli wrócimy do stacjonarnych zajęć w tym semestrze taki zestaw funkcji możecie mieć Państwo cały czas przy sobie. A w trakcie zajęć omówimy większość z nich:

Opis	Polecenie w R
Instalacja pakietu	Menu Packages/Installpackage
Ładowanie pakietu	library(<i>nazwa pakietu</i>)
Operator przypisania	= lub <-
Komentarz	Linijka zaczynająca się od #
Pierwiastek	sqrt(<i>liczba</i>)
Sinus	sin(argument)
Cosinus	cos(argument)
Liczba e	exp(1)
Logarytm naturalny z x	log(x, base=exp(1))
Wektor x	x=c(a, b, ...)
Wartości: minimalna, 1, 2 i 3 kwartył, maksymalna	quantile(wektor)
i-ty wyraz wektora x	x[i]
Odległość międzykwartylowa	IQR(wektor)
Generowanie liczb z podanego przedziału	:
Generowanie liczb do wektora x z podanego przedziału oraz z pewnym krokiem	x=seq(<i>początek</i> , koniec, by= <i>krok</i>)
Generowanie liczb do wektora x z podanego przedziału oraz z pewną długością	x=seq(<i>początek</i> , koniec, length= <i>długość</i>)
Generowanie wektora x będącego powtórzeniem innego	x=rep(wektor, ilość powtórzeń)
Generowanie wektora x powtarzając każdy element innego wektora	x=rep(wektor, each=ilość powtórzeń)
Wszystkie elementy wektora x oprócz wskazanego	x[-numer elementu]
Wszystkie elementy wektora x spełniające warunek	x[x>2]
Zgrupowanie danych	data.frame(wektor1, wektor2, wektor3)
Suma elementów wektora	sum(wektor)
Macierz składająca się z elementów wektora o n wierszach i m kolumnach	matrix(wektor, ncol=m, nrow=n)
Transpozycja macierzy	t(macierz)
Wyznacznik macierzy	det(macierz)
Iloczyn macierzy	macierz1%*%macierz2
Macierz diagonalna/ Wektor elementów na przekątnej	diag(macierz)
Macierz odwrotna	solve(macierz współczynników)
Wartości i wektory własne macierzy	eigen(macierz)
Tworzenie tablicy z wektorów (kolumnowo)	cbind(wektor1, wektor2,...)
Tworzenie tablicy z wektorów (wierszowo)	rbind(wektor1, wektor2,...)
Usunięcie i-tego wiersza z macierzy M	M[-i,]
Usunięcie j-tej kolumny z macierzy M	M[, -j]
Instrukcja warunkowa	if (warunek) {działanie jeżeli prawda} else {działanie jeżeli fałsz}

Pętla FOR	for (i in zmienna, dla której wykonywana jest pętla) {działania}
Pętla WHILE	while {warunek} {polecenia}
Wykres	curve
Wykres	plot (x,y)
- symbol znacznika	pch=symbol znacznika (od 0 do 25)
- zakres osi X (osi Y)	xlim= (ylim=)
- tytuł osi X (osi Y)	Xlab= (ylab=)
- tytuł wykresu	main=
- grubość linii	lwd=
- kolor wykresu	col="color"
Wczytywanie plików csv/txt	dane=read.csv("Nazwa pliku.csv", dec=".") dane=read.csv2("Nazwa pliku.csv", dec=".") dane=read.table("Nazwa pliku.txt", dec=".")
Zapisywanie wyników do plików	write.table(dane, file="Nazwa pliku.txt") write.csv(dane, file="Nazwa pliku.csv") write.csv2(dane, file="Nazwa pliku.csv")
Histogram	hist()
- suma wysokości kolumn wynosi 1	prob="true"
- brak przerwy do osi i	yaxs="i"
Wykres pudełkowy	boxplot()
Wariancja	var(wektor)
Odchylenie standardowe	sd(wektor)
Korelacja	cor(wektor1, wektor2), cor(macierz)
Kowariancja	Cov(wektor1, wektor2)
Przedział ufności dla średniej z nieznaną sigma	t.test(wektor, conf.level=współ. ufności)
Przedział ufności dla średniej ze znaną sigma	z.test(wektor, sigma.x=sigma, conf.level=współ. ufności)
Przedział ufności dla średniej (duża próba)	zsum.test(średnia z próby, odchylenie standardowe, liczebność, conf.level=współ. ufności)
Test hipotezy o średniej mu (sigma nieznanie)	t.test(dane, mu=mu, alternative='greater')
Test hipotezy o średniej mu (sigma znane)	library(PASWR) z.test(dane,mu=mu,sigma.x=sigma)
Test hipotezy o średniej (duża próba)	library(PASWR) zsum.test(średnia, odchylenie standardowe, liczebność, mu=mu, conf.level= 1- poziom istotności, alternative='less')
Test hipotezy o wariancji	library(TeachingDemos) sigma.test(dane, sigma=sigma, alternative="two.sided")
Test dla dwóch wariancji	var.test(dane1, dane2)

Test dla dwóch średnich (jednorodność wariancji)	t.test(dane1, dane2)
--	----------------------

A także podaję tabelkę związaną z rozkładami prawdopodobieństwa, które przydadzą się nam w trakcie kolejnych zajęć:

Rozkład	Dystrybuanta	Gęstość	Kwantyl	Generator	Uwagi
Dwumianowy	pbinom	dbinom	qbinom	rbinom	size, prob
Poissona	ppois	dpois	qpois	rpois	lambda
Ujemny dwumianowy	pnbinom	dnbinom	qnbinom	rnbinom	
Geometryczny	pgeom	dgeom	qgeom	rgeom	prob
Hipergeometryczny	phyper	dhyper	qhyper	rhyper	
Wielomianowy		dmultinom		rmultinom	
Jednostajny	punif	dunif	qunif	runif	min, max
Beta	pbeta	dbeta	qbeta	rbeta	
Wykładniczy	pexp	dexp	qexp	rexp	rate
Gamma	pgamma	dgamma	qgamma	rgamma	
Normalny	pnorm	dnorm	qnorm	rnorm	mean,sd
Weibulla	pweibull	dweibull	qweibull	rweibull	
Chi-kwadrat	pchisq	dchisq	qchisq	rchisq	
T	pt	dt	qt	rt	df
F	pf	df	qf	rf	