

PRAWO WIELKICH LICZB**Zad. 1** (Ilustracja Prawa Wielkich Liczb)

Wygeneruj zmienną `coins500_` przechowującą wynik 500 rzutów monetą, gdzie przyjmijmy, że 1 oznacza orła, natomiast 0 – reszkę.

Zbuduj ramkę `coinsFrame_` z kolumnami 'Lp' i 'Wynik'. Poszerz ramkę o kolejną kolumnę 'Suma częściowa' z narastającymi sumami częściowymi uzyskiwanych wyników.

Do powstałej ramki dodaj również kolumnę 'Częstość', tj. iloraz sumy częściowej i liczby dotychczasowo oddanych rzutów.

W końcu, wygeneruj wykres ilustrujący częstość zmieniającą się wraz z narastającą liczbą oddanych rzutów monetą. Użyj w tym celu polecenia `plot(...)` z argumentami `Lp`, `Częstość` oraz z zakresem danych na osi poziomej od 1 do 500.

Ustaw kolor wykresu na ciemno czerwony ('darkred') i sprawdź jak zmienia się typ wykresu gdy atrybutowi `type` nadasz wartości 'p', 'l', 'b', 'c', 'o', 'h', 's'.

Uwaga: co robi wbudowana w R funkcja `cumsum(...vector...)`?

WYKRESY – CIĄG DALSZY, DYSTRYBUANTA EMPIRYCZNA**Zad. 2**

Wygenerujemy wykres dystrybucyjny zmiennej losowej o rozkładzie normalnym ze średnią 0 i odchyleniem standardowym 4, zadając przedział argumentów `[-20, 20]` oraz przedział wartości od 0 do 1.

Użyjemy w tym celu funkcji `curve(...)` przyjmującej jako I argument funkcję `pnorm(x, 0, 4)`.

Co zwraca polecenie `pnorm(x, mean=..., sd=...)`? Sprawdź jak zmienia się typ i grubość linii gdy atrybutom `lty` oraz `lwd` (line type i line width, odpowiednio) nadawać będziesz kolejne wartości naturalne.

Innym kolorem, w tym samym układzie współrzędnych, narysuj dystrybucję empiryczną (empirical cumulative distribution function => `ecdf(...wektor danych...)`) dla 25 – elementowej próby losowej wygenerowanej z w/w rozkładu => `sample = rnorm(liczebność próby, parametr I, parametr II)`.

Co to jest dystrybucja empiryczna?

Uwaga: aby nanieść na istniejący wykres inny wykres należy użyć polecenia `lines()` z argumentem będącym zmienną przechowującą wykres (który chcemy nałożyć na istniejący już wykres) oraz pozostałymi pożądanymi atrybutami – np. `col=2...`

KORELACJA**Zad. 3**

Ze strony internetowej Bayern Monachium odczytujemy wzrost i wagę zawodników z I składu drużyny męskiej i żeńskiej.

Mężczyźni:

wzrost:

1.93, 1.86, 1.92, 1.82, 1.80, 1.76, 1.89, 1.74, 1.92, 1.72, 1.81, 1.89, 1.81, 1.81, 1.84, 1.86, 1.84, 1.75, 1.86, 1.79;

waga: 92, 76, 90, 76, 78, 73, 84, 70, 86, 68, 74, 76, 72, 81, 80, 80, 78, 75, 75, 75

Kobiety:

wzrost:

1.75, 1.79, 1.75, 1.78, 1.75, 1.69, 1.64, 1.78, 1.73, 1.79, 1.74, 1.71, 1.75, 1.61, 1.72, 1.71, 1.73, 1.64, 1.63, 1.58, 1.83, 1.70, 1.59, 1.59, 1.64  
waga: 73, 76, 70, 70, 63, 63, 59, 67, 60, 66, 60, 60, 64, 62, 62, 61, 63, 58, 60, 52, 68, 64, 55, 58, 59.

Utwórz wektory: `menHeight`, `menWeight`, `womenHeight`, `womenWeight`.

Narysuj wykresy (punktowe) zestawiające wysokość i wagę:

```
> plot(menHeight, menWeight, type='p')
```

```
> plot(womenHeight, womenWeight, type='o')
```

aby zauważyć korelację między wzrostem i wagą tak w przypadku drużyny męskiej jak i drużyny żeńskiej. Czym jest współczynnik korelacji i co możemy nim zmierzyć?

Policz – używając funkcji `cor`(..I zestaw danych.., ..II zestaw danych..) korelację między wzrostem i wagą dla każdej płci z osobna.

Połącz wektory wysokości dla obu drużyn i zapisz pod zmienną *height*, podobnie utwórz wektor łączny *weight*.

Narysuj znów wykres punktowy ilustrujący zależność między wzrostem i wagą. Zobacz co się zmieni jeśli parametrowi *type* nadasz wartość 'h', 'o', 's', ...

PACZKA GGPlot2 - wstęp

#### **Zad. 4**

Zainstaluj paczkę `ggplot2` oraz paczkę `SmarterPoland`.

Przyjrzyj się ramce danych przechowywanej pod zmienną *countries*.