

PSP – Raport 2

Nela Tomaszewicz

Kwiecień 2020

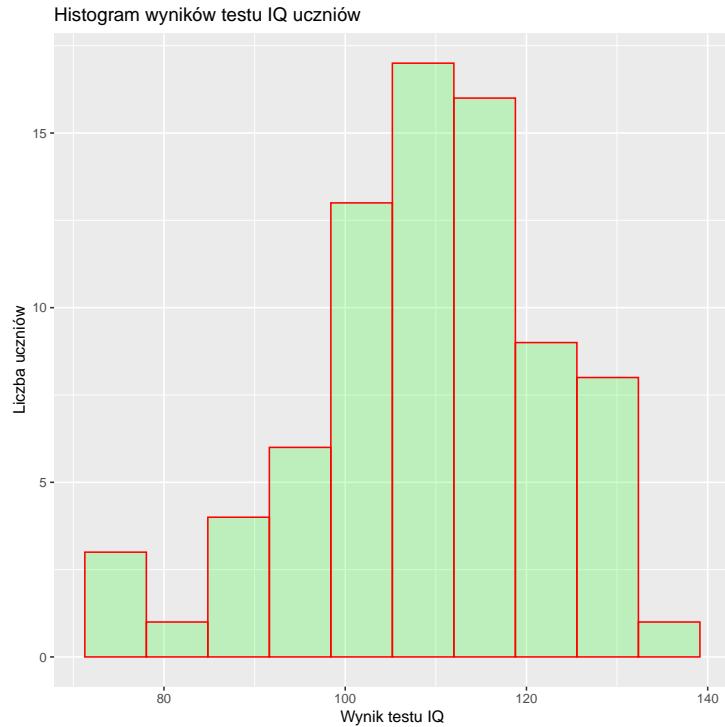
1 Zadanie 1

W zadaniu 1 sprawdzamy, czy rozkład wyników testu IQ z omawianego w poprzednim raporcie zbioru danych dotyczących 78 uczniów jest rozkładem normalnym. Dla przypomnienia przeprowadzony test IQ to najprawdopodobniej *Wechsler Intelligence Scale for Children* i interpretacja jego wyników jest następująca.

Przedział IQ	Interpretacja
≥ 130	Inteligencja bardzo wysoka
120-129	Inteligencja wysoka
110-119	Inteligencja powyżej przeciętnej
90-109	Inteligencja przeciętna
70-89	Inteligencja niższa niż przeciętna
≤ 69	Upośledzenie umysłowe

W zadaniu będziemy sprawdzać, czy rozkład wyników testu IQ jest normalny na dwa sposoby. Pierwszym jest **reguła 68% - 95% - 99.7%** polegająca na sprawdzeniu, czy w przybliżeniu 68% obserwacji zawiera się w przedziale $[\bar{x} - s, \bar{x} + s]$, 95% obserwacji zawiera się w przedziale $[\bar{x} - 2s, \bar{x} + 2s]$, a 99.7% obserwacji zawiera się w $[\bar{x} - 3s, \bar{x} + 3s]$, gdzie \bar{x} jest średnią obserwacji, a s odchyleniem standardowym. Drugim omawianym sposobem jest narysowanie **wykresu kwantylowego**, czyli **QQ-plot**. Jeśli dane mają rozkład zbliżony do normalnego, punkty na wykresie powinny leżeć blisko prostej wyznaczonej przez pakiet.

Wstępnią informację dotyczącą rozkładu zmiennej można również zauważać analizując histogram.



Histogram nie do końca przypomina symetryczną krzywą dzwonową charakterystyczną dla rozkładu normalnego z powodu niepokojącej asymetrii pomiędzy prawą stroną a lewą, patrząc od najwyższego binu. Dodatkowo pierwszy bin jest wyższy niż drugi, a gdyby dane miały rozkład normalny, to byłby on niższy. Sprawdźmy jednak normalność wspomnianymi wcześniej sposobami.

1.1 Reguła 68% - 95% - 99.7 %

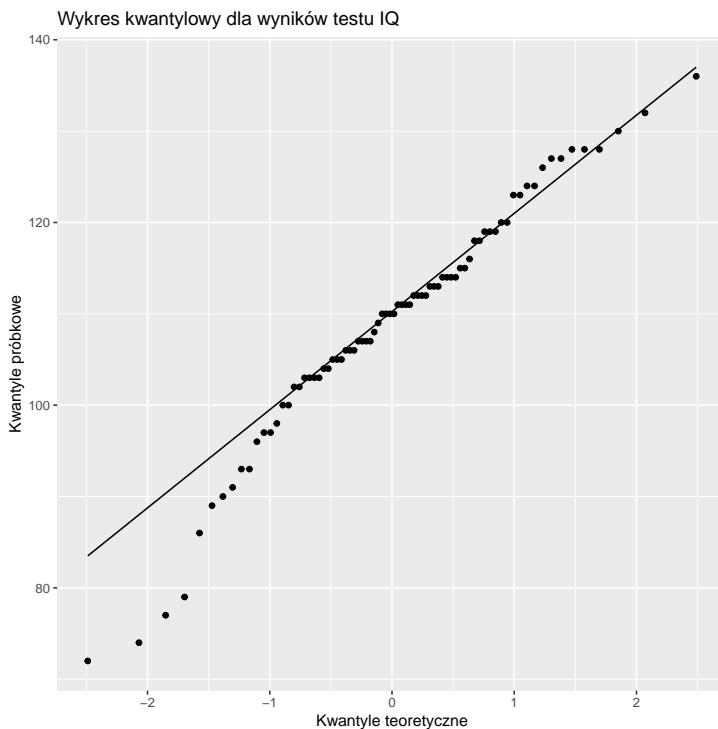
Liczbę i procent obserwacji w poszczególnych przedziałach pokażę w tabeli.

Przedział	Liczba obserwacji	Procent wszystkich obserwacji
$[\bar{x} - s, \bar{x} + s]$	55	$\approx 70.5\%$
$[\bar{x} - 2s, \bar{x} + 2s]$	73	$\approx 93.5\%$
$[\bar{x} - 3s, \bar{x} + 3s]$	78	100%

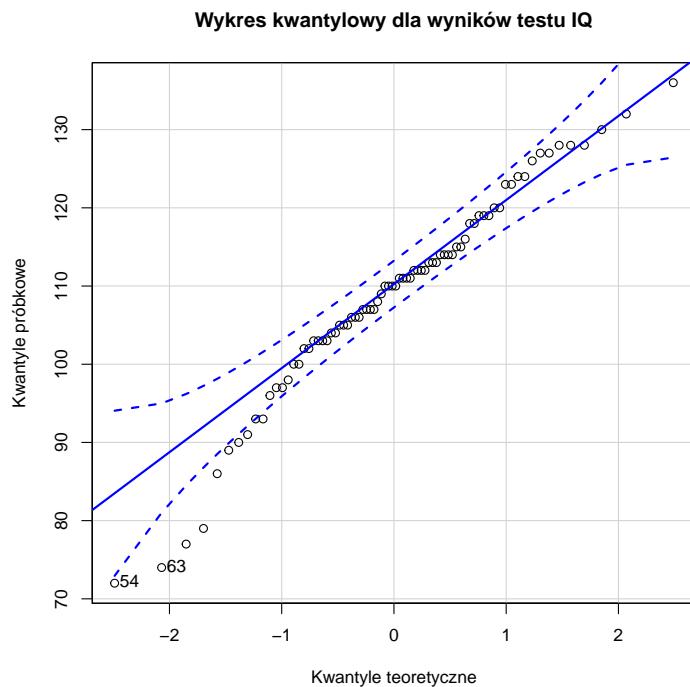
Wyliczone wartości procentowe odpowiadają w przybliżeniu wartościom dla rozkładu normalnego.

1.2 QQ–plot

QQ–plot dla wyników testu IQ został przedstawiony poniżej.



Większość punktów leży na lub w pobliżu prostej. Mediana wyników testu IQ wynosi 110 i możemy zauważać, że na przecięciu 110 i $z = 0$ (kwantyl teoretyczny) leży punkt. Punkty wyznaczające niskie wyniki testu nie leżą na prostej i są od niej oddalone. Faktycznie, patrząc na histogram, możemy zauważać, że pierwszy bin jest wyższy od kolejnych binów, co zaburza rozkład. Jednakże w przypadku takiego wykresu nasuwa się pytanie: co to znaczy, że punkty są blisko prostej? Czy taki układ punktów na wykresie można przyjąć za normalny? Na udzielenie lepszych odpowiedzi na te pytania pozwala wykres `qqPlot` z biblioteki `car`, który poza linią z poprzedniego wykresu wyznacza także przedziały, w których muszą zawierać się punkty z wykresu kwantylowego, żeby można było przypuścić, że rozkład jest normalny.



Widzimy, że nie wszystkie punkty na wykresie zawierają się w przedziale wyznaczonym przez qqPlot. Możemy zatem z większą pewnością (oczywiście nie 100%, bo w statystyce niczego nie możemy być na 100% pewni!) powiedzieć, że rozkład wyników testu IQ uczniów nie jest rozkładem normalnym.

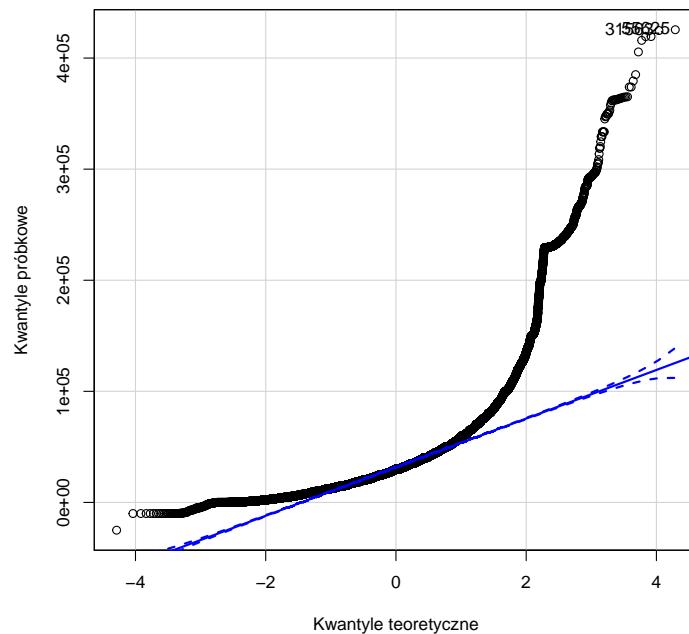
1.3 Podsumowanie i wnioski

Zadanie pokazuje, że reguła 68% - 95% - 99.7% nie jest najlepszą możliwą metodą sprawdzenia czy rozkład zmiennej ilościowej jest normalny. Znacznie lepsze jest wyznaczenie wykresu kwantylowego, w R najbardziej pomocny jest qqPlot z biblioteki car.

2 Zadanie 2

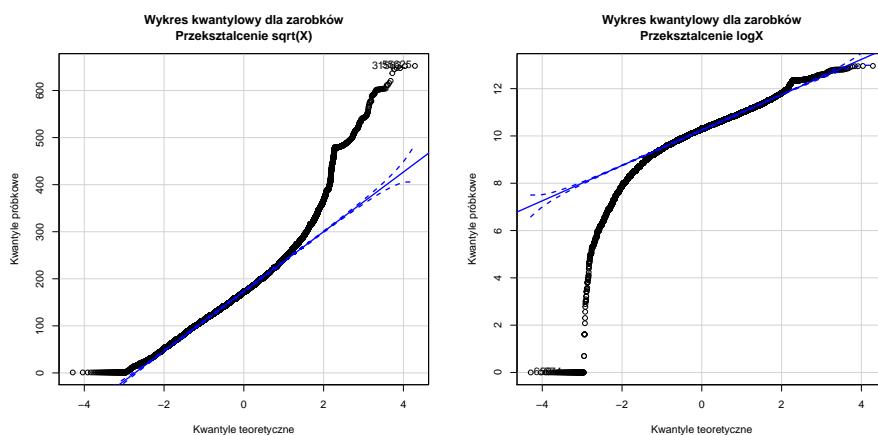
Korzystając ze zbioru danych `income.txt` sprawdzamy, czy rozkład zarobków jest normalny. W tym celu użyjemy wykresu qqPlot z biblioteki car wykorzystanego w poprzednim zadaniu. Dla zarobków mamy wykres pokazany poniżej.

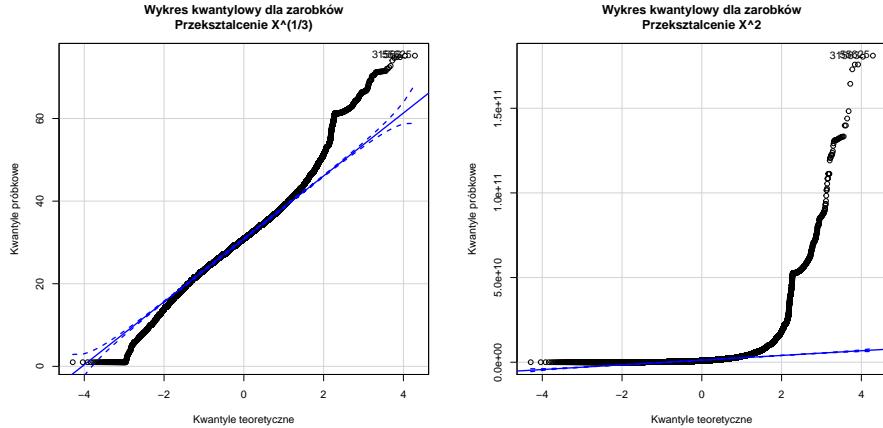
Wykres kwantylowy dla zarobków



Duża część punktów na wykresie nie zawiera się w przedziałach wyznaczonych przez qqPlot, zatem możemy stwierdzić, że rozkład zarobków nie jest normalny, co przykładowo mogłoby uniemożliwić przeprowadzenie pewnych testów statystycznych takich jak na przykład t-test.

Sprawdzmy zatem, czy przekształcając zmienne losowe w sposób nieliniowy, będziemy w stanie uzyskać rozkład normalny. Sprawdzimy cztery przekształcenia: \sqrt{X} , $\log X$, $\sqrt[3]{X}$ oraz X^2 .





Żaden z wykresów kwantylowych nie pokazuje rozkładu normalnego. Przy pomocy prostych transformacji nie udało się przekształcić danych, tak aby pochodziły z rozkładu normalnego.

2.1 Podsumowanie i wnioski

Przekształcenia przetestowane w zadaniu nie należą do zaawansowanych (takich jak na przykład transformacja Boxa-Coxa). Widzimy także, że nie pasują one do danych o zarobkach, między innymi dlatego, że przekształcenia takie jak $\log(X)$, $\log_{10} X$ oraz \sqrt{X} nie powinny być aplikowane do ujemnych danych, a takie znajdują się w naszym zbiorze. Przez to także widzimy, że stosunkowo „najlepiej” poradziła sobie transformacja pierwiastkiem sześciennym, która może być aplikowana do danych zawierających liczby ujemne. Podniesienie do kwadratu spowodowało zupełne zaburzenie rozkładu.

3 Zadanie 3

W zadaniu badamy trajektorie różnych statystyk tworzące się na skutek losowania danych ze zbioru `income.txt`. W obydwu podpunktach losujemy dziesięciokrotnie próbę róznej liczności.

3.1 Proste próbkowanie losowe

Losujemy 1000 elementów ze zbiorem danych `income.txt` zawierającym zarobki, wiek, wykształcenie i sektor zatrudnienia 55 899 osób ze Stanów Zjednoczonych. Zastosujemy proste próbkowanie losowe, zakładając, że każda osoba ma jednakowe prawdopodobieństwo zostania wylosowaną. Trajektorie statystyk: średnia, mediana, odchylenie standardowe i IQR zostaną wykreślone dla zmiennej ilościowej oznaczającej zarobki w dolarach. Ostatnia statystyka nie jest związana z zarobkami – wyznacza procent osób w próbie mających co najmniej licencję.

Pierwsze cztery wykresy trajektorii zostały wyznaczone przy pomocy następującej funkcji, której kod został podany poniżej.

```
library(ggplot2)
library(dplyr)
library(reshape2)
set.seed(1)

income_sampling <- function(statistic, statistic_name) {
  # pusty wektor dla pierwszych t elementów
  t_elem <- c()
  # pusty wektor dla losowania
  sample_income <- c()
  # pusta ramka danych dla późniejszego zapisywania wyników
  df <- data.frame(matrix(NA, nrow = 1000, ncol = 10))

  for(i in 1:10) {
    sample_income <- sample(income$zarobki, 1000)
    for (j in 1:1000) {
      #obliczenie wartosci odpowiedniej statystyki
      t_elem[j] <- statistic(sample_income[1:j])
    }
    #zapisywanie wynikow jako kolumn
    df[,i] <- t_elem
  }

  df <- df %>%
    mutate(times = 1:nrow(df))
  #funkcja melt 'skleja' ramke
  #dzięki czemu możemy narysować wykres
  df <- melt(df, id = 'times')

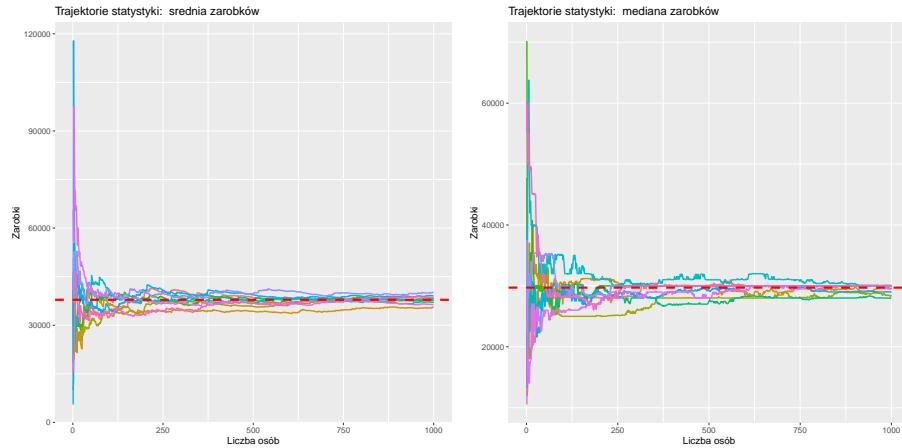
  g <- ggplot(df, aes(times, value)) +
    geom_line(aes(colour = variable), show.legend = F,
              size = 0.75) +
    labs(title = paste("Trajektorie statystyki: ",
                      statistic_name)) +
    labs(x="Liczba osób", y="Zarobki") +
    geom_hline(yintercept = statistic(income$zarobki),
               linetype = "dashed",
               color = "red",
               size = 1.15)

    return(g)
}
```

Funkcja przyjmuje interesującą nas statystykę i jej polską nazwę, dzięki czemu automatyzujemy generowanie wykresów. Zwraca wykres trajektorii statystyki liczonej w oparciu o pierwszych t elementów ($t \in [1, 1000]$). W przypadku naszych danych tymi „elementami” są osoby. Czerwoną przerywaną linią zaznaczono wartości statystyk dla całej próby.

Wykresy oraz wywołania dla pierwszych dwóch statystyk:

```
income_sampling(mean, "średnia zarobków")
income_sampling(median, "medianana zarobków")
```



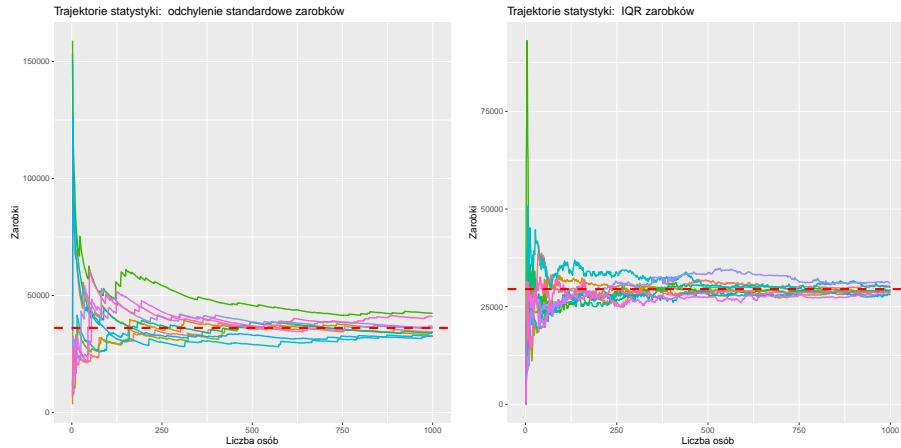
Obydwa wykresy stabilizują się na poziomie wartości statystyk dla całej próby, wtedy gdy liczba osób jest większa niż około 250 (w przypadku średniej) oraz większa niż około 325 (w przypadku mediany). Dla małej próby, czyli mniejszej niż 250 osób, możemy zauważać, że mediana osiąga wyniki bardziej oddalone od swojej wartości dla całej próby niż średnia.

Dla rozstępu międzykwartylowego (IQR) zaimplementowałam następującą funkcję:

```
income_sampling(sd, "odchylenie standardowe zarobków")

IQR <- function(x) {
  return(quantile(x, 0.75) - quantile(x, 0.25))
}

income_sampling(IQR, "IQR zarobków")
```



Odchylenie standarowe, podobnie jak mediana, dla małej liczby osób (< 250) osiąga wyniki bardziej oddalone od wartości dla całej próby niż, przykładowo, IQR czy średnia. To zjawisko można wytlumaczyć analizując wzór na odchylenie standardowe (σ) wyglądający następująco:

$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N}},$$

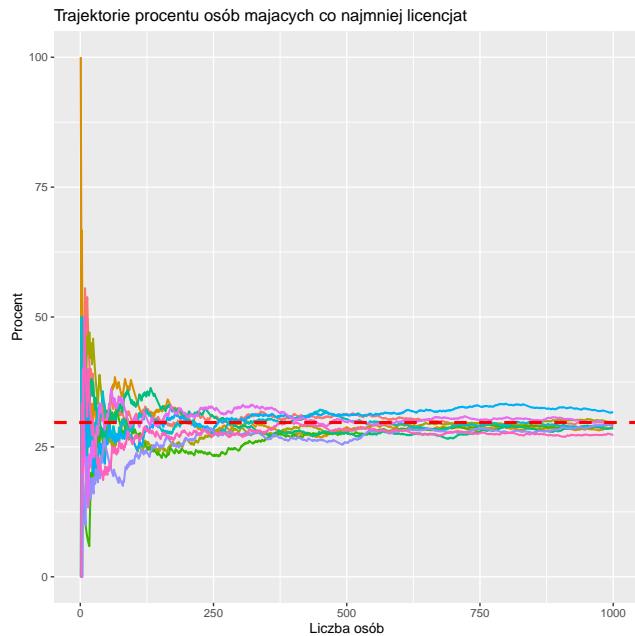
gdzie N to rozmiar próby, x_i – wartość zmiennej w próbie, a μ to wartość średnia próby. Ponieważ N jest w mianowniku, im większa jego wartość, tym mniejsza wartość odchylenia standardowego. Na wykresie możemy zauważać, że dla dużych ($250 <$) N oznaczonych jako liczba osób, trajektorie odchylenia standardowego „spłaszczały” się. Podobne zjawisko możemy zaobserwować dla średniej próbkoowej (\bar{x}), której wzór jest następujący:

$$\bar{x} = \frac{\sum x_i}{N}.$$

W przypadku tej statystyki szybciej osiągamy poziom wyliczony dla całej próby, ponieważ w mianowniku znajduje się N , a nie \sqrt{N} .

Jeśli chodzi o medianę i rozstęp międzykwartylowy, większa liczba osób gwarantuje większe zróżnicowanie pomiędzy zarobkami. Wartość środkowa zbliża się wtedy do wartości dla całej próby, a pomiędzy kwartylami pierwszym i trzecim zawiera się więcej wartości.

Ostatnią omawianą statystyką jest procent osób mających co najmniej licencję. Wykres narysowałam, stosując bardzo podobny kod do funkcji `income_sampling` zamieniając losowanie z wektora zarobków na losowanie z wektora wykształcenia, czyli `sample(income$wykszt, 1000)`. Wyliczenie procentowych wartości wykonałam, sumując liczbę wszystkich osób z wykształceniem większym lub równym 5 (czyli wykształceniem wyższym i równym licencjatowi), dzieląc przez obecną wartość t ($t \in [1, 1000]$) i mnożąc przez 100. Pozostałe elementy kodu zostały takie jak we wcześniej omawianej funkcji. Powstały wykres został zaprezentowany poniżej.



Trajektorie tej statystyki stabilizują się w przypadku liczby osób większej niż około 325.

3.1.1 Podsumowanie i wnioski

Związek między liczbą osób a definicjami odpowiednich statystyk został omówiony wcześniej. Porównując narysowane wykresy z dokładnymi wartościami dla całej próby:

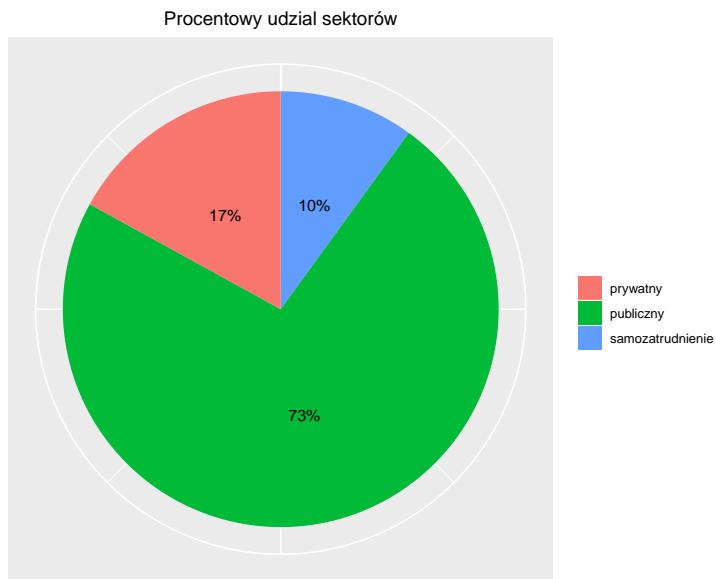
Sr.	Medianą	Odczytanie std.	IQR	% osób z min. licencjatem
37864.61	29717	36158.03	29503.5	≈ 29.7

możemy zauważyć, że dla odpowiedniej liczby osób (zazwyczaj większej niż około 250) otrzymujemy wartości statystyk zbliżone do wartości wyliczonej dla całej próby. Oznacza to, że w przypadku posiadanych przez nas danych, aby próbkiowanie miało sens i dawało odpowiednie, czyli oddające charakter wszystkich danych, rezultaty dla obliczanych statystyk należy wylosować grupę większą niż 250 osób. W przypadku wylosowania mniejszej grupy osób moglibyśmy po pełnić duży *błąd próbkiowania* wynikający z niewystarczającej reprezentacji całej próby.

3.2 Próbkowanie warstwowe

Zastosujemy próbkiowanie warstwowe ze względu na sektor zatrudnienia, czyli będziemy dziesięć razy losować grupę 100-osobową ze zbioru `income.txt`, gdzie

w każdej takiej grupie liczba osób z danego sektora będzie odpowiadać procentowi osób z całego sektora. W zbiorze danych mamy trzy sektory oznaczone liczbami 5, 6, 7, gdzie 5 oznacza sektor prywatny, 6 sektor publiczny, a 7 samozatrudnienie. Procentowy udział każdego z sektorów w zbiorze danych jest następujący:



W każdej 100-elementowej próbie będziemy losować 73 osoby z sektora publicznego, 17 z sektora prywatnego oraz 10 z samozatrudniających. Naszym zadaniem jest wykonanie poprawnego losowania, a następnie wyznaczenie trajektorii wybranych statystyk. W moim przypadku będą to: średnia, mediana, odchylenie standardowe oraz rozstęp międzykwartylowy (IQR). Funkcja do wykonania zadania jest następująca.

```
library(plyr)
library(dplyr)
library(ggplot2)
library(reshape2)
#ziarno, zeby za kazdym razem nie dostawac innych wykresow
set.seed(1)

strat_income_sampling <- function(statistic, statistic_name) {
  temp_income <- income
```

```

#sektory procentowo
sizes <- round(table(income$sektor)/nrow(income)*100)
df <- data.frame(matrix(NA, nrow = 10, ncol = 10))

# petla powtarzajaca eksperyment
for(k in 1:10) {
  temp_income <- income
  sektory_total <- data.frame()
  stat_value <- c()
  for (i in 1:10) {
    # losowanie dla kazdego z sektorow autorstwa
    # Pani Giniewicz
    sektor5 <-temp_income[temp_income$sektor==5,]
    idx5 <-sample(nrow(sektor5),sizes[1])
    sektor5_df <- sektor5[idx5,]

    sektor6 <-temp_income[temp_income$sektor==6,]
    idx6 <-sample(nrow(sektor6),sizes[2])
    sektor6_df <- sektor6[idx6,]

    sektor7 <-temp_income[temp_income$sektor==7,]
    idx7 <-sample(nrow(sektor7),sizes[3])
    sektor7_df<- sektor7[idx7,]

    #polaczenie wylosowanych obserwacji z trzech sektorow
    sektory <- rbind(sektor5_df, sektor6_df, sektor7_df)

    #zapamietywania poprzedniego losowania
    #dopisywanie kolejnego
    sektory_total <- rbind(sektory_total, sektory)

    #wyliczenie trajektorii
    stat_value[i] <- statistic(sektory_total$zarobki)

    temp_income <- setdiff(temp_income, sektory_total)
  }
  #wyliczone wartosci statystyki
  df[,k] <- stat_value
}

#nowa ramka danych potrzebna do rysowania wykresow
df <- df %>%
  mutate(number = (1:nrow(df))*100)
#'sklejenie' ramki do dwoch kolumn
# w celu narysowania wykresow

```

```

df <- melt(df, id="number")

#wykres trajektorii statystyki
g <- ggplot(data=df, aes(x=number, y=value)) +
  geom_line(aes(color=variable), show.legend = F,
            size = 0.75) +
  labs(title = paste("Trajektoria statystyki:",
                     statistic_name)) +
  labs(x="Liczba osób", y="Zarobki") +
  geom_hline(yintercept = statistic(income$zarobki),
             linetype = "dashed", color = "red",
             size = 1.15)

return(g)
}

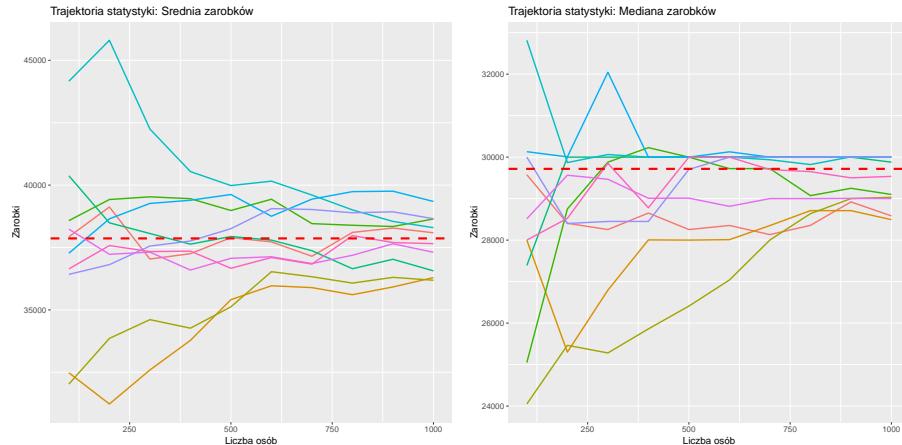
```

Tak jak poprzednio, funkcja przyjmuje nazwę po angielsku i po polsku odpowiedniej statystyki, której trajektorię chcemy badać i zwraca wykres. Wywołania funkcji dla dwóch pierwszych statystyk:

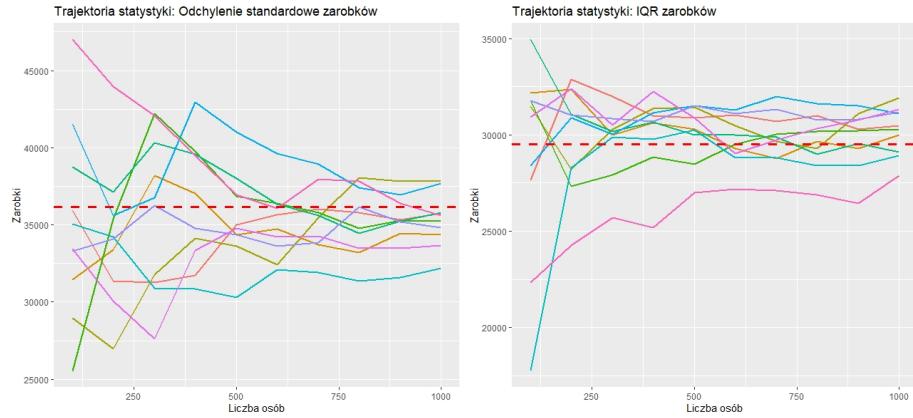
```

strat_income_sampling(mean, "Średnia zarobków")
strat_income_sampling(median, 'Medianą zarobków')

```



Dwa pozostałe wykresy trajektorii:



Trajektorie stabilizują się na poziomie zbliżonym do wartości statystyki dla całej próby, jednakże możemy zauważać różnice wynikające z faktu, że losujemy więcej osób z konkretnego sektora. Przykładem takich różnic jest wykres trajektorii dla odchylenia standardowego, gdzie 7 z 10 linii wyznaczających wartości odchylenia w każdym z eksperymentów stabilizuje się poniżej wartości dla całej próby. Ponieważ na 170 przypadków zarobków ujemnych aż 163 ma miejsce w sektorze 7, czyli w sektorze wyznaczającym osoby z samozatrudnienia, odchylenie od wartości średniej jest mniejsze, gdy losujemy za każdym razem tylko 10 osób z tego sektora.

3.2.1 Podsumowanie i wnioski

Patrząc na wykresy trajektorii w przypadku próbkowania warstwowego, możemy zauważyć, że potrzeba większej liczby osób niż w prostym próbkowaniu losowym, aby trajektorie były stabilne. Wynika to z dużych różnic w udziale sektorów zatrudnienia w zbiorze danych. Jednakże takie próbkowanie lepiej oddaje stan rzeczywisty danych, ponieważ nieprzypadkowo w zbiorze jest 73% osób pracujących w sektorze publicznym.

4 Zadanie 4

W zadaniu symulujemy dziesięciokrotnie eksperyment polegający na 1000 rzutów monetą. Dla każdego z eksperymentów wykreślmy trajektorie liczby frakcji orłów w kolejnych k rzutach, gdzie k należy do przedziału $[1, 1000]$. Zadanie rozwiązałam implementując następującą funkcję:

```
library(ggplot2)
library(reshape2)
```

```

# Oznaczenia: 1 - orzel, 0 - reszka

coin_tossing <- function(p) {
  #pusty wektor dla pierwszych k elementów
  k_elem <- c()
  #pusty wektor dla symulacji rzutu monetą
  tossing <- c()
  #pusta ramka danych dla późniejszego zapisywania wyników
  df <- data.frame(matrix(NA, nrow = 1000, ncol = 10))

  for(i in 1:10) {
    tossing <- rbinom(1000, 1, prob=p)
    for (j in 1:1000) {
      # frakcja
      k_elem[j] <- sum(tossing[1:j])/j
    }
    # zapisywanie wyników frakcji jako kolumn w ramce
    df[,i] <- k_elem
  }

  df <- df %>%
    mutate(times = 1:nrow(df))

  # sklejamy ramkę w celu wykonania wykresu
  df <- melt(df, id = 'times')

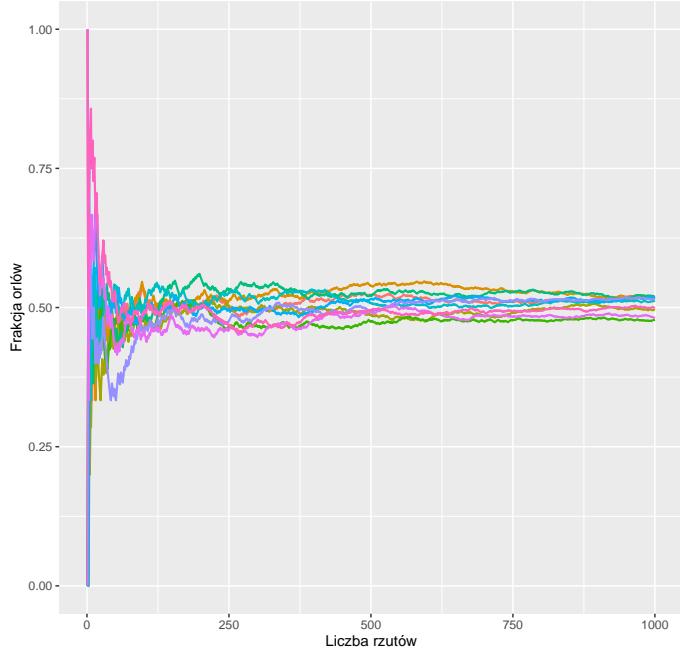
  # wykres trajektorii
  g <- ggplot(df, aes(times, value)) +
    geom_line(aes(colour = variable), show.legend = F,
              size = 0.75) +
    labs(title = paste("Trajektorie frakcji orłów w 10 eksperymentach rzutu monetą, p = ", p)) +
    labs(x="Liczba rzutów", y="Frakcja orłów")

  return(g)
}

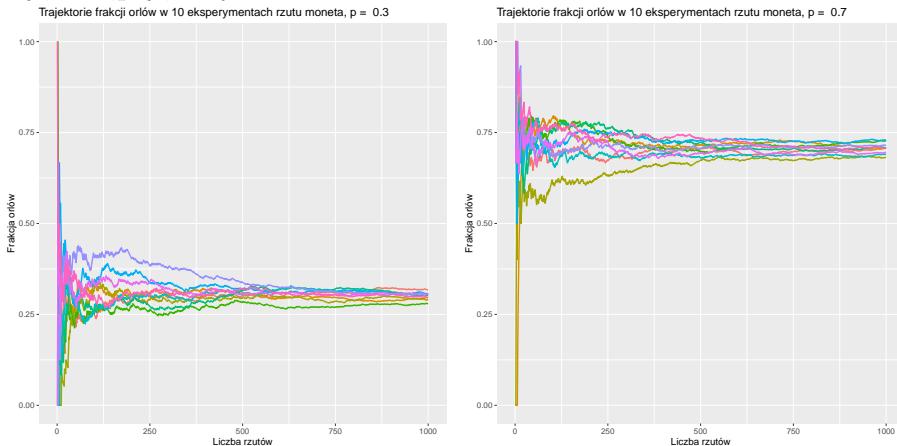
```

Funkcja `coin_tossing()` przyjmuje jako argument p czyli prawdopodobieństwo sukcesu. Zwraca wykres z trajektoriami frakcji orłów. Wywołanie funkcji dla $p = 0.5$, czyli „sprawiedliwego” rzutu monetą pokazano poniżej.

Trajektorie frakcji orłów w 10 eksperymentach rzutu monetą, $p = 0.5$



Wywołania funkcji dla niesymetrycznego rzutu monetą ($p = 0.3$ i $p = 0.7$) dają następujące wyniki.



Na wykresach można zauważać, że trajektorie po około 250 rzutach zaczynają się stabilizować na poziomie wartości p oznaczającej prawdopodobieństwo sukcesu. Jest to ilustracja Prawa Wielkich Liczb (konkretniej Mocnego Prawa Wielkich Liczb (1)), mówiącego, że średni wynik eksperymentu powtarzanego w dużej liczbie prób powinien zbliżać się do wartości oczekiwanej. W przypadku gdy zmienna losowa X pochodzi z rozkładu dwumianowego wzór na wartość oczekiwana to $E(X) = np$. W naszym przypadku liczba prób n jest równa 1,

zatem $E(X) = p$. Zatem na wykresach widzimy, że dla dużej liczby rzutów trajektorie stabilizują się na poziomie $p = 0.5, 0.3, 0.7$, czyli na poziomie odpowiednich wartości oczekiwanych.

4.1 Podsumowanie i wnioski

Omawiane w zadaniu zdarzenia są niezależne. Nie możemy przewidzieć wyniku rzutu monetą, ale dzięki Prawu Wielkich Liczb wiemy, że z czasem, konsekwentnie rzucając monetą, $p\%$ wyników będą stanowić orły, czyli odniesione sukcesy.

Literatura

- [1] Law of Large Numbers, https://en.wikipedia.org/wiki/Law_of_large_numbers