STATYSTYCZNA ANALIZA DANYCH

Projekt - Bezrobotni i oferty pracy

Adrian Jakubowski
Inżynieria i Analiza Danych, II rok

Spis treści

1. Wstęp 2
   - Wstęp teoretyczny 2
   - Cel projektu 2
   - Dane 2
2. Przygotowanie pracy 2
   - Wczytanie bibliotek i ustawienie working directory 2
   - Import oraz przygotowanie danych 3
3. Obliczanie parametrów 4
   - Średnia 4
   - Maksymalne wartości 4
   - Minimalne wartości 5
   - Rozstęp 5
   - Odchylenie standardowe 6
   - Wariancja 6
   - Mediana 7
   - Kwantyle 7
   - Rozstęp międzykwartylowy 8
   - Współczynnik zmienności 8
4. Wizualizacja 9
   - Histogramy 9
   - Wykresy pudełkowe 12
   - Wykresy liniowe 15
   - Wykres kołowy 17
5. Hipotezy 19
   - Hipoteza pierwsza 19
   - Hipoteza druga 20
6. Podsumowanie i wnioski 21
7. Lista użytych komend 21

1. Wstęp

➢ Wstęp teoretyczny

Bezrobocie to sytuacja, w której osoby zdolne i gotowe do podjęcia pracy nie mają zatrudnienia. Występowanie bezrobocia jest uważane za jeden z kluczowych wskaźników ekonomicznych, ponieważ ma wpływ na sytuację finansową i jakość życia jednostek oraz na rozwój gospodarki kraju. Bezrobocie jest zwykle mierzone jako stosunek liczby osób bezrobotnych do ogólnej liczby osób w wieku produkcyjnym (w Polsce zazwyczaj jest to osoby w wieku od 15 do 64 lat), wyrażony w procentach i nazywany stopą bezrobocia. Występowanie bezrobocia jest niekorzystne zarówno dla jednostek, które pozostają bez pracy, jak i dla całego społeczeństwa, ponieważ prowadzi do marnotrawienia zasobów ludzkich, spadku produkcji, a także wzrostu kosztów związanych z zasiłkami dla bezrobotnych.

➢ Cel projektu

Celem projektu jest przeanalizowanie danych dotyczących bezrobocia za pomocą pakietów R oraz identyfikacja zależności między różnymi zmiennymi związanych z bezrobociem, takimi jak liczba osób bezrobotnych, a liczba ofert pracy czy liczba nowo zarejestrowanych osób bezrobotnych w korelacji z liczbą wyrejestrowanych osób bezrobotnych. Analiza ta ma na celu zwiększenie zrozumienia sytuacji na rynku pracy oraz poprawę zdolności wyciągania wniosków idących z analizy.

➢ Dane

Dane poddane dalszej obróbce oraz analizie zostały pobrane z oficjalnej strony Głównego Urzędu Statystycznego. Na potrzeby projektu został pobrany plik z rozszerzeniem .xlsx zawierający miesięczne dane związane z bezrobociem z lat 2010-2022. Plik ten zawierał kolumny dotyczące m.in. ilości osób bezrobotnych, ogólnie, ale również w różnych kategoriach (np. płeć, osoby dotychczas niepracujące, bez prawa do zasiłku itp.). Z pliku można odczytać również stopę bezrobocia w danym miesiącu oraz liczbę zgłoszonych ofert pracy. W projekcie skupiam się jednak na kolumnach zawierających ogólną liczbę osób bezrobotnych, liczbę kobiet bezrobotnych, stopę bezrobocia, liczbę osób nowo zarejestrowanych, wyrejestrowanych oraz także liczbę zgłoszonych ofert pracy.

2. Przygotowanie pracy

➢ Wczytanie bibliotek i ustawienie working directory

- **readxl** – biblioteka do importu plików excel
- **lubridate** - umożliwia manipulację danymi dotyczącymi dat i czasu. W projekcie jest wykorzystywana do przekształcenia daty na liczbę dziesiętną, która reprezentuje datę w formacie rozumianym przez obiekt szeregów czasowych ts
- **plotly** – do tworzenia bardziej zaawansowanych wykresów
- **graphics** – do tworzenia wykresów oraz dodawania różnych elementów do utworzonych wykresów

```
#_____________________ BIBLIOTEKI
library(readxl)
library(lubridate)
library(plotly)
library(graphics)
```

➢ Import oraz przygotowanie danych

Zaimportowałem dane za pomocą polecenia read_excel, a następnie zapisałem je jako ramkę danych. Następnym krokiem był wybór odpowiednich kolumn i wierszy, aby w ramce zawrzeć te dane, które są gotowe do dalszej analizy. Zmieniłem nazwy poszczególnych kolumn oraz zmieniłem ich typ na numeryczny, aby możliwe było obliczanie wszelakich statystyk na ich podstawie. Początkowe dane były podane w tysiącach, więc aby nie zaburzać wyników postanowiłem przemnożyć wszystkie takie kolumny przez 1000. Po takim zabiegu wszystkie wyniki będą bardziej dokładne.

```
#Wczytanie danych do zmiennej dane_start
dane_start <- (read_excel("tabl12_bezrobotni_zarejestrowani_oferty_pracy.xlsx"))

#Zmiana wczytanych danych na ramkę danych
dane_start <- as.data.frame(dane_start)

#Przycięcie ramki danych do łatwiejszej analizy
dane <- dane_start[-c(1:5),c(1,2,4,20,22,26,30)]

#Zmiana nazw kolumn
colnames(dane) <- c("Rok/Miesiac",
                    "Liczba_bezrobotnych_ogolem",
                    "Liczba_bezrobotnych_kobiet",
                    "Stopa_bezrobocia",
                    "Bezrobotni_nowo_zarejestrowani",
                    "Bezrobotni_wyrejestrowani",
                    "Liczba_zgloszonych_ofert_pracy")

#zmiana typu danych w kolumnach
dane$Liczba_bezrobotnych_ogolem <- as.numeric(dane$Liczba_bezrobotnych_ogolem)
dane$Liczba_bezrobotnych_kobiet <- as.numeric(dane$Liczba_bezrobotnych_kobiet)
dane$Stopa_bezrobocia <- as.numeric(dane$Stopa_bezrobocia)
dane$Bezrobotni_nowo_zarejestrowani <- as.numeric(dane$Bezrobotni_nowo_zarejestrowani)
dane$Bezrobotni_wyrejestrowani <- as.numeric(dane$Bezrobotni_wyrejestrowani)
dane$Liczba_zgloszonych_ofert_pracy <- as.numeric(dane$Liczba_zgloszonych_ofert_pracy)

#przemnożenie razy 1000
dane$Liczba_bezrobotnych_ogolem <- dane$Liczba_bezrobotnych_ogolem * 1000
dane$Liczba_bezrobotnych_kobiet <- dane$Liczba_bezrobotnych_kobiet * 1000
dane$Bezrobotni_nowo_zarejestrowani <- dane$Bezrobotni_nowo_zarejestrowani * 1000
dane$Bezrobotni_wyrejestrowani <- dane$Bezrobotni_wyrejestrowani * 1000
dane$Liczba_zgloszonych_ofert_pracy <- dane$Liczba_zgloszonych_ofert_pracy * 1000
```

3. Obliczanie parametrów

➢ Średnia

Obliczam średnią z danych kolumn za pomocą polecenia **mean**.

```
#1 - średnia
(srBO <- mean(dane$Liczba_bezrobotnych_ogolem))
(srBK <- mean(dane$Liczba_bezrobotnych_kobiet))
(srSB <- mean(dane$Stopa_bezrobocia))
(srBNZ <- mean(dane$Bezrobotni_nowo_zarejestrowani))
(srBW <- mean(dane$Bezrobotni_wyrejestrowani))
(srOP <- mean(dane$Liczba_zgloszonych_ofert_pracy))
```

```
> #1 - średnia
> (srBO <- mean(dane$Liczba_bezrobotnych_ogolem))
[1] 1463928
> (srBK <- mean(dane$Liczba_bezrobotnych_kobiet))
[1] 771834.2
> (srSB <- mean(dane$Stopa_bezrobocia))
[1] 9.117722
> (srBNZ <- mean(dane$Bezrobotni_nowo_zarejestrowani))
[1] 171732.3
> (srBW <- mean(dane$Bezrobotni_wyrejestrowani))
[1] 178243
> (srOP <- mean(dane$Liczba_zgloszonych_ofert_pracy))
[1] 99684.81
```

➢ Maksymalne wartości

Największa wartość w kolumnie jest otrzymywana dzięki funkcji **max**.

```
#2 - maksymalne wartości
(maxBO <- max(dane$Liczba_bezrobotnych_ogolem))
(maxBK <- max(dane$Liczba_bezrobotnych_kobiet))
(maxSB <- max(dane$Stopa_bezrobocia))
(maxBNZ <- max(dane$Bezrobotni_nowo_zarejestrowani))
(maxBW <- max(dane$Bezrobotni_wyrejestrowani))
(maxOP <- max(dane$Liczba_zgloszonych_ofert_pracy))
```

```
> #2 - maksymalne wartości
> (maxBO <- max(dane$Liczba_bezrobotnych_ogolem))
[1] 2336700
> (maxBK <- max(dane$Liczba_bezrobotnych_kobiet))
[1] 1161900
> (maxSB <- max(dane$Stopa_bezrobocia))
[1] 14.4
> (maxBNZ <- max(dane$Bezrobotni_nowo_zarejestrowani))
[1] 317900
> (maxBW <- max(dane$Bezrobotni_wyrejestrowani))
[1] 306100
> (maxOP <- max(dane$Liczba_zgloszonych_ofert_pracy))
[1] 169900
```

➢ Minimalne wartości

Najmniejsza wartość w kolumnie – polecenie **min**.

```
#3 - minimalne wartości
(minBO <- min(dane$Liczba_bezrobotnych_ogolem))
(minBK <- min(dane$Liczba_bezrobotnych_kobiet))
(minSB <- min(dane$Stopa_bezrobocia))
(minBNZ <- min(dane$Bezrobotni_nowo_zarejestrowani))
(minBW <- min(dane$Bezrobotni_wyrejestrowani))
(minOP <- min(dane$Liczba_zgloszonych_ofert_pracy))
```

```
> #3 - minimalne wartości
> (minBO <- min(dane$Liczba_bezrobotnych_ogolem))
[1] 796000
> (minBK <- min(dane$Liczba_bezrobotnych_kobiet))
[1] 433800
> (minSB <- min(dane$Stopa_bezrobocia))
[1] 5
> (minBNZ <- min(dane$Bezrobotni_nowo_zarejestrowani))
[1] 84100
> (minBW <- min(dane$Bezrobotni_wyrejestrowani))
[1] 43600
> (minOP <- min(dane$Liczba_zgloszonych_ofert_pracy))
[1] 35100
```

➢ Rozstęp

Rozstęp jest obliczany jako różnica wartości maksymalnej od wartości minimalnej.

```
#4 - rozstęp
(rozBO <- maxBO - minBO)
(rozBK <- maxBK - minBK)
(rozSB <- maxSB - minSB)
(rozBNZ <- maxBNZ - minBNZ)
(rozBW <- maxBW - minBW)
(rozOP <- maxOP - minOP)
```

```
> #4 - rozstęp
> (rozBO <- maxBO - minBO)
[1] 1540700
> (rozBK <- maxBK - minBK)
[1] 728100
> (rozSB <- maxSB - minSB)
[1] 9.4
> (rozBNZ <- maxBNZ - minBNZ)
[1] 233800
> (rozBW <- maxBW - minBW)
[1] 262500
> (rozOP <- maxOP - minOP)
[1] 134800
```

➢ Odchylenie standardowe

Wartość odchylenia standardowego jest obliczana za pomocą polecenia **sd**.

```
#5 - odchylenie standardowe
(sdBO <- sd(dane$Liczba_bezrobotnych_ogolem))
(sdBK <- sd(dane$Liczba_bezrobotnych_kobiet))
(sdSB <- sd(dane$Stopa_bezrobocia))
(sdBNZ <- sd(dane$Bezrobotni_nowo_zarejestrowani))
(sdBW <- sd(dane$Bezrobotni_wyrejestrowani))
(sdOP <- sd(dane$Liczba_zgloszonych_ofert_pracy))
```

```
> #5 - odchylenie standardowe
> (sdBO <- sd(dane$Liczba_bezrobotnych_ogolem))
[1] 496450.6
> (sdBK <- sd(dane$Liczba_bezrobotnych_kobiet))
[1] 240983.4
> (sdSB <- sd(dane$Stopa_bezrobocia))
[1] 3.14121
> (sdBNZ <- sd(dane$Bezrobotni_nowo_zarejestrowani))
[1] 56215.42
> (sdBW <- sd(dane$Bezrobotni_wyrejestrowani))
[1] 58275.48
> (sdOP <- sd(dane$Liczba_zgloszonych_ofert_pracy))
[1] 28627.2
```

➢ Wariancja

Wariancję obliczyłem za pomocą funkcji **var**.

```
#6 - wariancja
(warBO <- var(dane$Liczba_bezrobotnych_ogolem))
(warBK <- var(dane$Liczba_bezrobotnych_kobiet))
(warSB <- var(dane$Stopa_bezrobocia))
(warBNZ <- var(dane$Bezrobotni_nowo_zarejestrowani))
(warBW <- var(dane$Bezrobotni_wyrejestrowani))
(warOP <- var(dane$Liczba_zgloszonych_ofert_pracy))
```

```
> #6 - wariancja
> (warBO <- var(dane$Liczba_bezrobotnych_ogolem))
[1] 246463205207
> (warBK <- var(dane$Liczba_bezrobotnych_kobiet))
[1] 58072982009
> (warSB <- var(dane$Stopa_bezrobocia))
[1] 9.8672
> (warBNZ <- var(dane$Bezrobotni_nowo_zarejestrowani))
[1] 3160173920
> (warBW <- var(dane$Bezrobotni_wyrejestrowani))
[1] 3396031130
> (warOP <- var(dane$Liczba_zgloszonych_ofert_pracy))
[1] 819516583
```

➢ Mediana

Wartość środkowa – mediana jest wyliczana w języku R za pomocą polecenia **median**.

```
#7 - mediana
(medianBO <- median(dane$Liczba_bezrobotnych_ogolem))
(medianBK <- median(dane$Liczba_bezrobotnych_kobiet))
(medianSB <- median(dane$Stopa_bezrobocia))
(medianBNZ <- median(dane$Bezrobotni_nowo_zarejestrowani))
(medianBW <- median(dane$Bezrobotni_wyrejestrowani))
(medianOP <- median(dane$Liczba_zgloszonych_ofert_pracy))
```

```
> #6 - wariancja
> (warBO <- var(dane$Liczba_bezrobotnych_ogolem))
[1] 246463205207
> (warBK <- var(dane$Liczba_bezrobotnych_kobiet))
[1] 58072982009
> (warSB <- var(dane$Stopa_bezrobocia))
[1] 9.8672
> (warBNZ <- var(dane$Bezrobotni_nowo_zarejestrowani))
[1] 3160173920
> (warBW <- var(dane$Bezrobotni_wyrejestrowani))
[1] 3396031130
> (warOP <- var(dane$Liczba_zgloszonych_ofert_pracy))
[1] 819516583
```

➢ Kwantyle

Do obliczenia kwantyli użyłem funkcji **quantile**.

```
#8 - kwantyle
(quantileBO <- quantile(dane$Liczba_bezrobotnych_ogolem))
(quantileBK <- quantile(dane$Liczba_bezrobotnych_kobiet))
(quantileSB <- quantile(dane$Stopa_bezrobocia))
(quantileBNZ <- quantile(dane$Bezrobotni_nowo_zarejestrowani))
(quantileBW <- quantile(dane$Bezrobotni_wyrejestrowani))
(quantileOP <- quantile(dane$Liczba_zgloszonych_ofert_pracy))
```

```
> #8 - kwantyle
> (quantileBO <- quantile(dane$Liczba_bezrobotnych_ogolem))
     0%     25%     50%     75%    100%
 796000  977350 1387950 1944575 2336700
> (quantileBK <- quantile(dane$Liczba_bezrobotnych_kobiet))
     0%     25%     50%     75%    100%
 433800  544050  734400 1018775 1161900
> (quantileSB <- quantile(dane$Stopa_bezrobocia))
   0%   25%   50%   75%  100%
 5.00  6.10  8.50 12.25 14.40
> (quantileBNZ <- quantile(dane$Bezrobotni_nowo_zarejestrowani))
    0%    25%    50%    75%   100%
 84100 120275 169950 212750 317900
> (quantileBW <- quantile(dane$Bezrobotni_wyrejestrowani))
    0%    25%    50%    75%   100%
 43600 128175 173350 230050 306100
> (quantileOP <- quantile(dane$Liczba_zgloszonych_ofert_pracy))
    0%    25%    50%    75%   100%
 35100  77875  98300 119900 169900
```

➢ Rozstęp międzykwartylowy

Rozstęp międzykwartylowy, inaczej rozstęp ćwiartkowy w języku R obliczany jest za pomocą funkcji IQR (interquartile range), a jest to różnica miedzy trzecim, a pierwszym kwartylem.

```
#9 - rozstep miedzykwartylowy
(iqrBO <- IQR(dane$Liczba_bezrobotnych_ogolem))
(iqrBK <- IQR(dane$Liczba_bezrobotnych_kobiet))
(iqrSB <- IQR(dane$Stopa_bezrobocia))
(iqrBNZ <- IQR(dane$Bezrobotni_nowo_zarejestrowani))
(iqrBW <- IQR(dane$Bezrobotni_wyrejestrowani))
(iqrOP <- IQR(dane$Liczba_zgloszonych_ofert_pracy))
```

```
> #9 - rozstęp międzykwartylowy
> (iqrBO <- IQR(dane$Liczba_bezrobotnych_ogolem))
[1] 967225
> (iqrBK <- IQR(dane$Liczba_bezrobotnych_kobiet))
[1] 474725
> (iqrSB <- IQR(dane$Stopa_bezrobocia))
[1] 6.15
> (iqrBNZ <- IQR(dane$Bezrobotni_nowo_zarejestrowani))
[1] 92475
> (iqrBW <- IQR(dane$Bezrobotni_wyrejestrowani))
[1] 101875
> (iqrOP <- IQR(dane$Liczba_zgloszonych_ofert_pracy))
[1] 42025
>
```

➢ Współczynnik zmienności

Współczynnik zmienności wyraża się jako iloraz odchylenia standardowego i średniej.

```
#10 - współczynnik zmienności
(VBO <- sdBO/srBO)
(VBK <- sdBK/srBK)
(VSB <- sdSB/srSB)
(VBNZ <- sdBNZ/srBNZ)
(VBW <- sdBW/srBW)
(VOP <- sdOP/srOP)
```

```
> #10 - współczynnik zmienności
> (VBO <- sdBO/srBO)
[1] 0.3391223
> (VBK <- sdBK/srBK)
[1] 0.3122217
> (VSB <- sdSB/srSB)
[1] 0.344517
> (VBNZ <- sdBNZ/srBNZ)
[1] 0.3273434
> (VBW <- sdBW/srBW)
[1] 0.3269439
> (VOP <- sdOP/srOP)
[1] 0.2871772
```

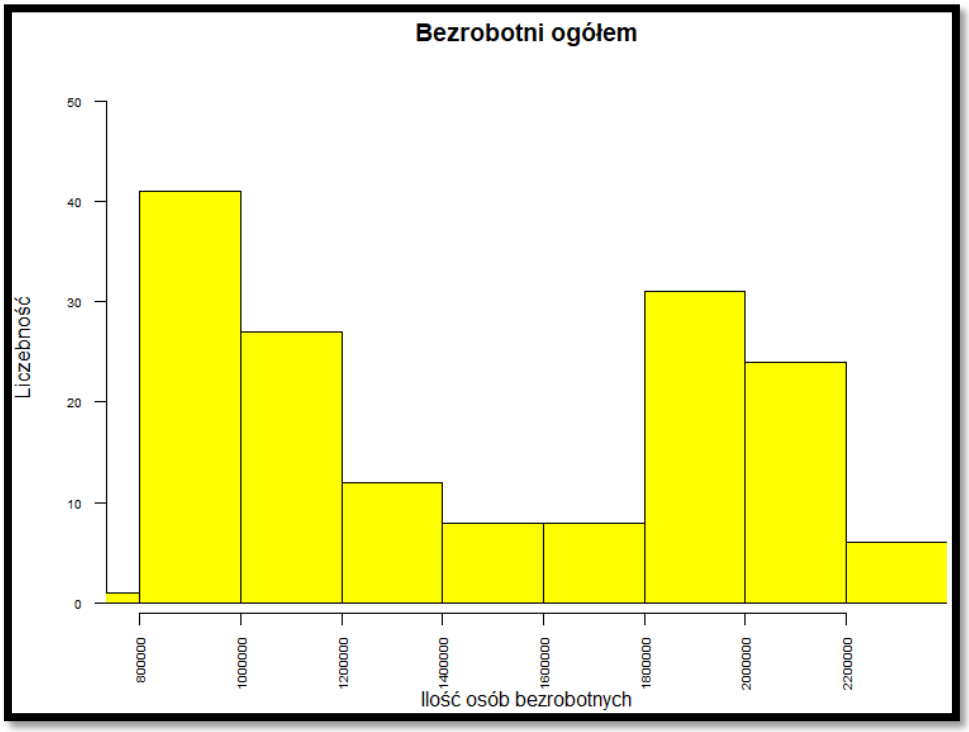
4. Wizualizacja

➢ Histogramy

Histogramy informują o rozkładzie danych lub częstości występowania wartości w danym zbiorze. Są to wykresy, które przedstawiają ilość wystąpień poszczególnych wartości w danym zakresie.

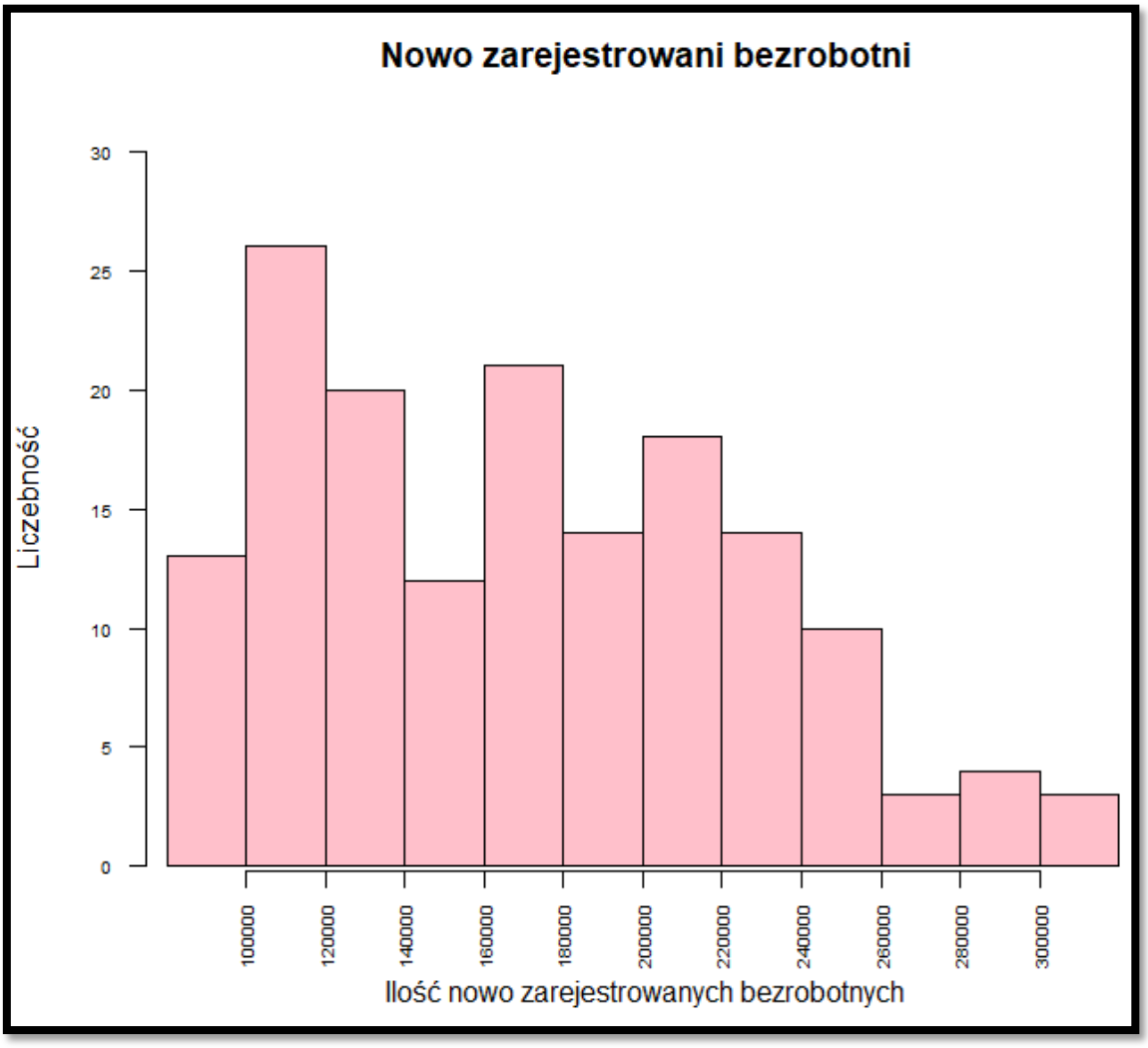
Pierwsze dwa histogramy, które utworzyłem dotyczą ogólnej liczby bezrobotnych oraz także bezrobotnych nowo zarejestrowanych. Do tego celu zastosowałem funkcję hist, w której zawarłem dane (zmienione na format szeregu czasowego – as.ts). Dodatkowo, dla przejrzystości wykresu zmieniłem nazwę oraz parametry dotyczące odpowiednich osi – nazwę oraz zakres wartości. Zmiany zostały także dokonane jeśli chodzi o kolor słupków w histogramie. Końcowym etapem tworzenia histogramu było polecenie axis, dzięki któremu dopisuje dodatkowe wartości na osi X, aby ułatwić odczyt oraz analizę wykresu. Przed tym jednak, użyłem argumentu xaxt = „n”, aby wyłączyć pierwotnie tworzone wartości na osi X (nie były dopasowane dokładnie do początków zbiorów). Warto dodać, że zarówno w funkcji hist, jak i w poleceniu axis używam dodatkowych dwóch argumentów: cex.axis, która zmniejsza wielkość liczb oraz las = 2, które ustawia liczby pionowo.

```
#Histogram ogólnej liczby bezrobotnych
hist(as.ts(dane$Liczba_bezrobotnych_ogolem),
     main = "Bezrobotni ogółem",
     xlab = "Ilość osób bezrobotnych",
     ylab = "Liczebność",
     ylim = c(1,50),
     xlim = c(minBO,maxBO),
     col = "yellow",
     las = 2,
     cex.axis = 0.6,
     xaxt = "n")
axis(side = 1,
     at = c(800000, 1000000, 1200000, 1400000, 1600000, 1800000, 2000000, 2200000),
     labels = c("800000", "1000000", "1200000", "1400000", "1600000", "1800000", "2000000", "2200000"),
     las = 2,
     cex.axis = 0.6)
```



Histogram przedstawia, że w latach, które są poddane analizie ogólna liczba bezrobotnych była bardzo zróżnicowana, szczególnie często występowały wartości w okolicach miliona oraz dwóch milionów. Zdecydowanie mniej można było odnotować wartości z przedziału 1200000-1800000.

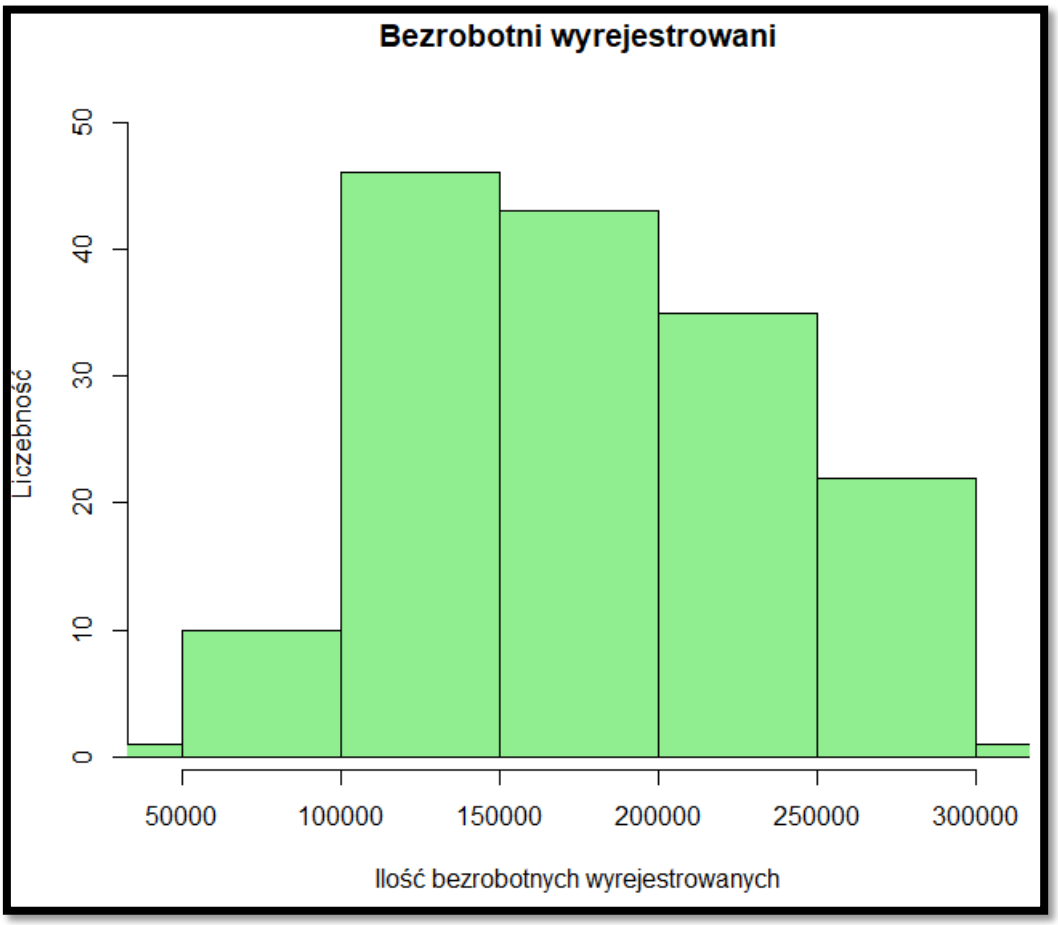
```
#Histogram liczby bezrobotnych nowo zarejestrowanych
hist(as.ts(dane$Bezrobotni_nowo_zarejestrowani),
     main = "Nowo zarejestrowani bezrobotni",
     xlab = "Ilość nowo zarejestrowanych bezrobotnych",
     ylab = "Liczebność",
     ylim = c(1,30),
     xlim = c(minBNZ,maxBNZ),
     col = "pink",
     cex.axis = 0.6,
     las = 2,
     xaxt = "n")
axis(side = 1,
     at = c(100000, 120000, 140000, 160000, 180000, 200000, 220000, 240000, 260000, 280000, 300000),
     labels = c("100000","120000","140000","160000","180000","200000","220000","240000","260000","280000","300000"),
     las = 2,
     cex.axis = 0.6)
```



Można wysnuć ogólny wniosek z tego histogramu, ktory ukazuje, że wraz ze wzrostem wartości oznaczającej ilość nowo zarejestrowanych bezrobotnych, liczebność tej wartości malała.

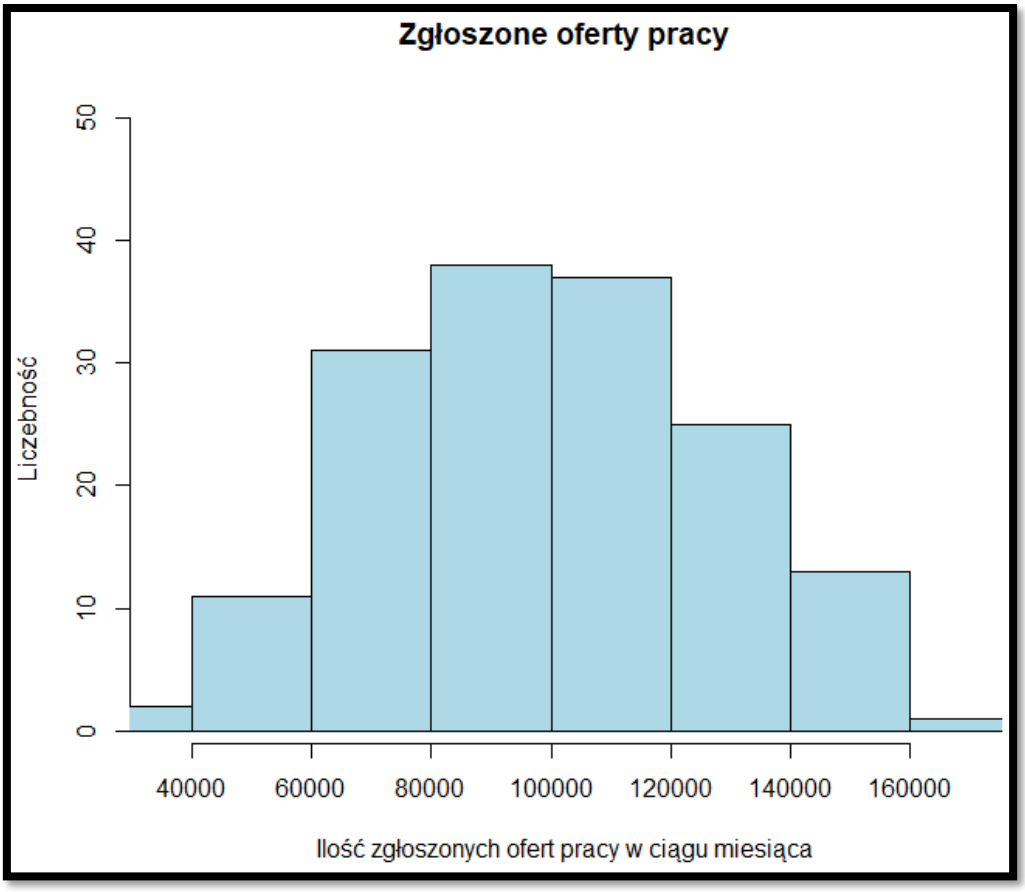
Następnie utworzyłem histogramy dotyczące liczby bezrobotnych wyrejestrowanych oraz miesięcznej liczby nowo ogłoszonych ofert pracy.

```r
#Histogram liczby bezrobotnych wyrejestrowanych
hist(as.ts(dane$Bezrobotni_wyrejestrowani),
     main = "Bezrobotni wyrejestrowani",
     xlab = "Ilość bezrobotnych wyrejestrowanych",
     ylab = "Liczebność",
     ylim = c(1,50),
     xlim = c(minBW,maxBW),
     col = "lightgreen")
```



Wśród miesięcznych ilości bezrobotnych wyrejestrowanych przeważały wartości średnie, najwięcej z przedziału 100000-150000.

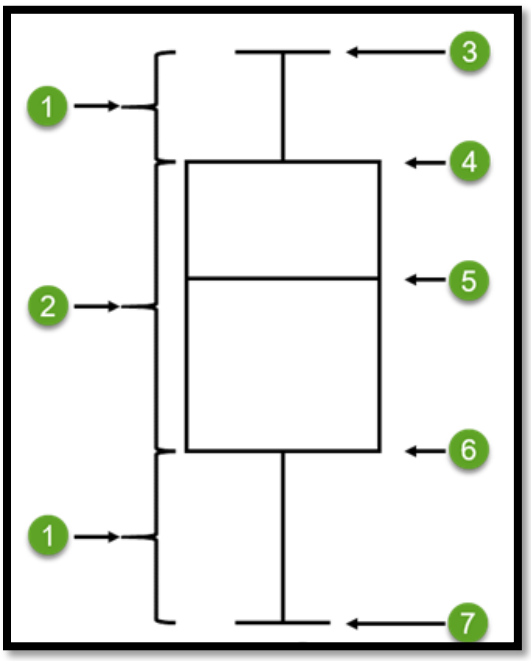
```r
#Histogram ofert pracy
hist(as.ts(dane$Liczba_zgloszonych_ofert_pracy),
     main = "Zgłoszone oferty pracy",
     xlab = "Ilość zgłoszonych ofert pracy w ciągu miesiąca",
     ylab = "Liczebność",
     ylim = c(1,50),
     xlim = c(minOP,maxOP),
     col = "lightblue")
```



Podobnie jak w przypadku poprzedniego histogramu, tak również w przypadku zgłoszonych ofert pracy, przeważały wartości średnie.

➢ Wykresy pudełkowe

Przed rozpoczęciem pracy nad wykresami pudełkowymi warto najpierw zagłębić się najpierw w to, jak działają tego typu wykresy.



1 – WĄS - Przedział danych mniejszych od pierwszego kwartylu lub większych od trzeciego kwartylu. Każdy wąs zawiera 25% danych. Typowo wąsy nie powinny przekraczać wartości 1,5 razy większej od IQR, co określa wartość progową dla elementów odstających.

2 – PROSTOKĄT - Przedział danych między pierwszym i trzecim kwartylem. 50 procent danych leży w tym przedziale.

3 – MAKSIMUM - Największa wartość w zestawie danych lub największa wartość niewykraczająca poza wartość progową wyznaczoną przez wąsy.

4 – TRZECI KWARTYL - Wartość, od której 75% danych ma mniejszą wartość, a 25% danych wartość większą.

5 – MEDIANA - Środkowa liczba w zestawie danych. Połowa liczb ma większą wartość niż mediana, a połowa wartość mniejszą. Medianę można także nazwać drugim kwartylem.

6 – PIERWSZY KWARTYL - Wartość, od której 25% danych ma mniejszą wartość, a 75% danych wartość większą

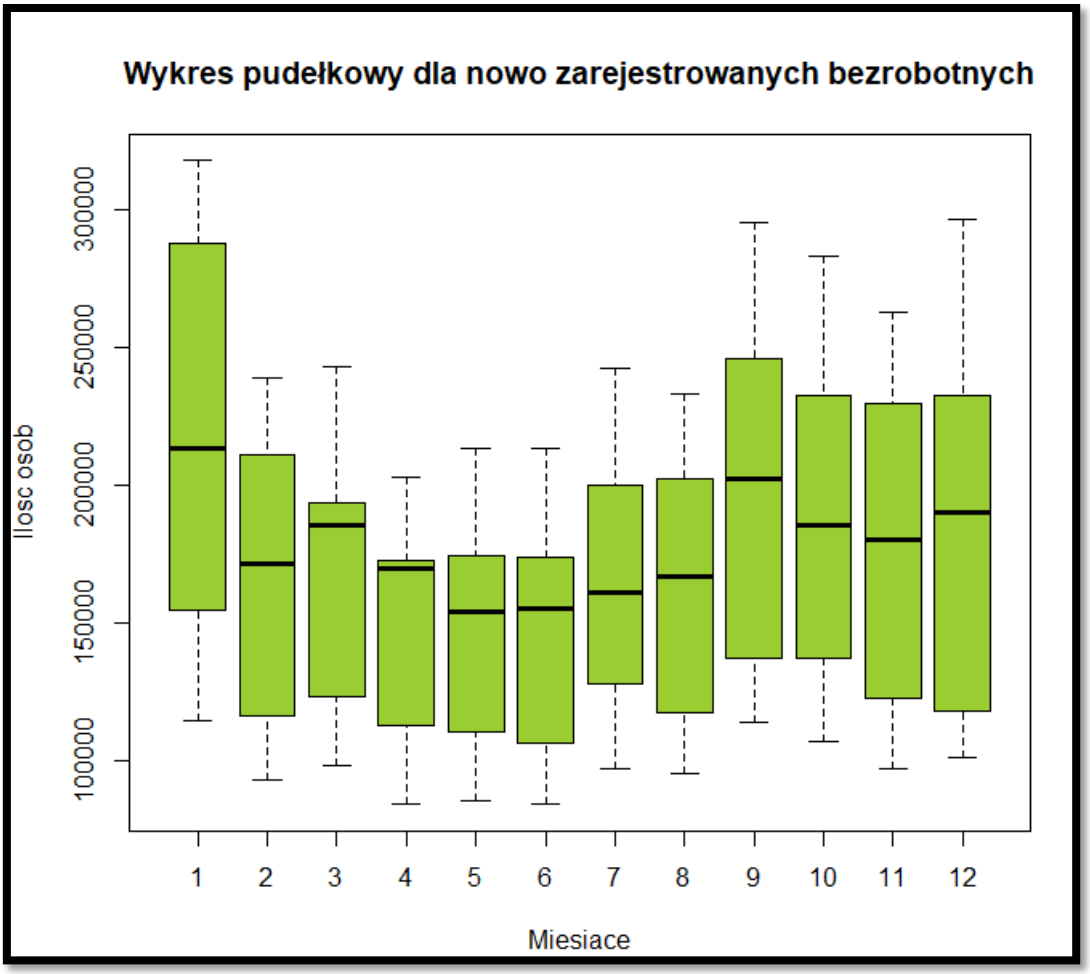
7 – MINIMUM - Najmniejsza wartość w zestawie danych lub najmniejsza wartość niewykraczająca poza wartość progową wyznaczoną przez wąsy.

Na potrzeby projektu zdecydowałem się utworzyć wykres pudełkowy dotyczący osób bezrobotnych nowo zarejestrowanych oraz wyrejestrowanych, aby zobaczyć czy są widoczne zależności pomiędzy tymi wielkościami.

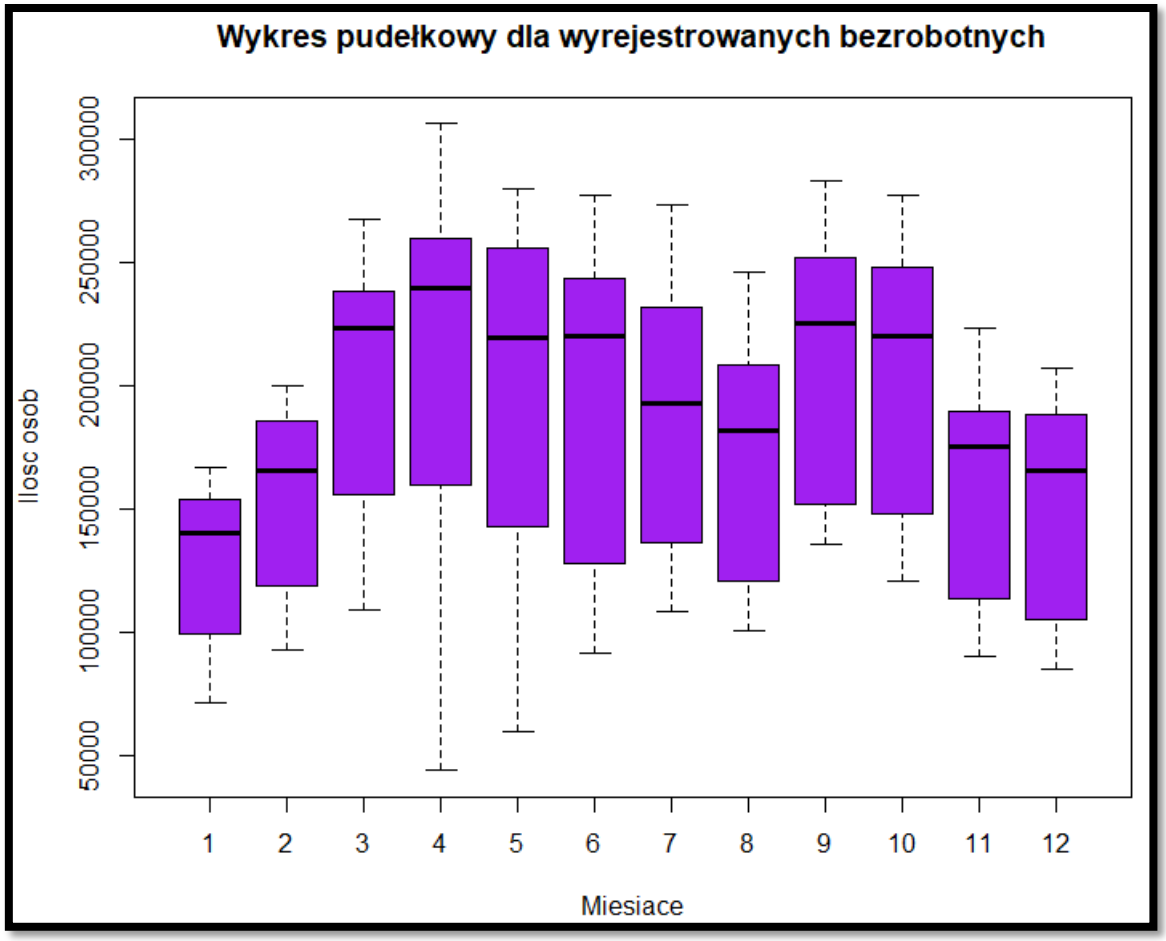
Aby tego dokonać utworzyłem najpierw szereg czasowy, stosując do tego celu funkcję ts. W niej użyłem argumentu start z funkcją decimal_date z pakietu lubridate, która jest wykorzystywana do przekształcenia daty na liczbę dziesiętną reprezentującą datę w formacie rozumianym przez obiekt szeregów czasowych ts. Podałem start jako pierwszy miesiąc z moich danych oraz wartość frequency równą 12, czyli liczbę miesięcy.

Do stworzenia wykresu pudełkowego zastosowałem funkcję boxplot, a w niej zawarłem szereg czasowy oraz argumenty związane z obróbką graficzną wykresu, podobnie jak przy histogramach.

```
tsBNZ <- ts(data = dane$Bezrobotni_nowo_zarejestrowani,
            start = decimal_date(ymd("2010-01-01")),
            frequency = 12)
boxplot(tsBNZ~ cycle(tsBNZ),
        col = "yellowgreen",
        main = "Wykres pudełkowy dla nowo zarejestrowanych bezrobotnych",
        xlab = "Miesiace",
        ylab = "Ilosc osob")
```




```r
tsBW <- ts(data = dane$Bezrobotni_wyrejestrowani,
           start = decimal_date(ymd("2010-01-01")),
           frequency = 12)
boxplot(tsBW~ cycle(tsBW),
        col = "purple",
        main = "Wykres pudełkowy dla wyrejestrowanych bezrobotnych",
        xlab = "Miesiace",
        ylab = "Ilosc osob")
```



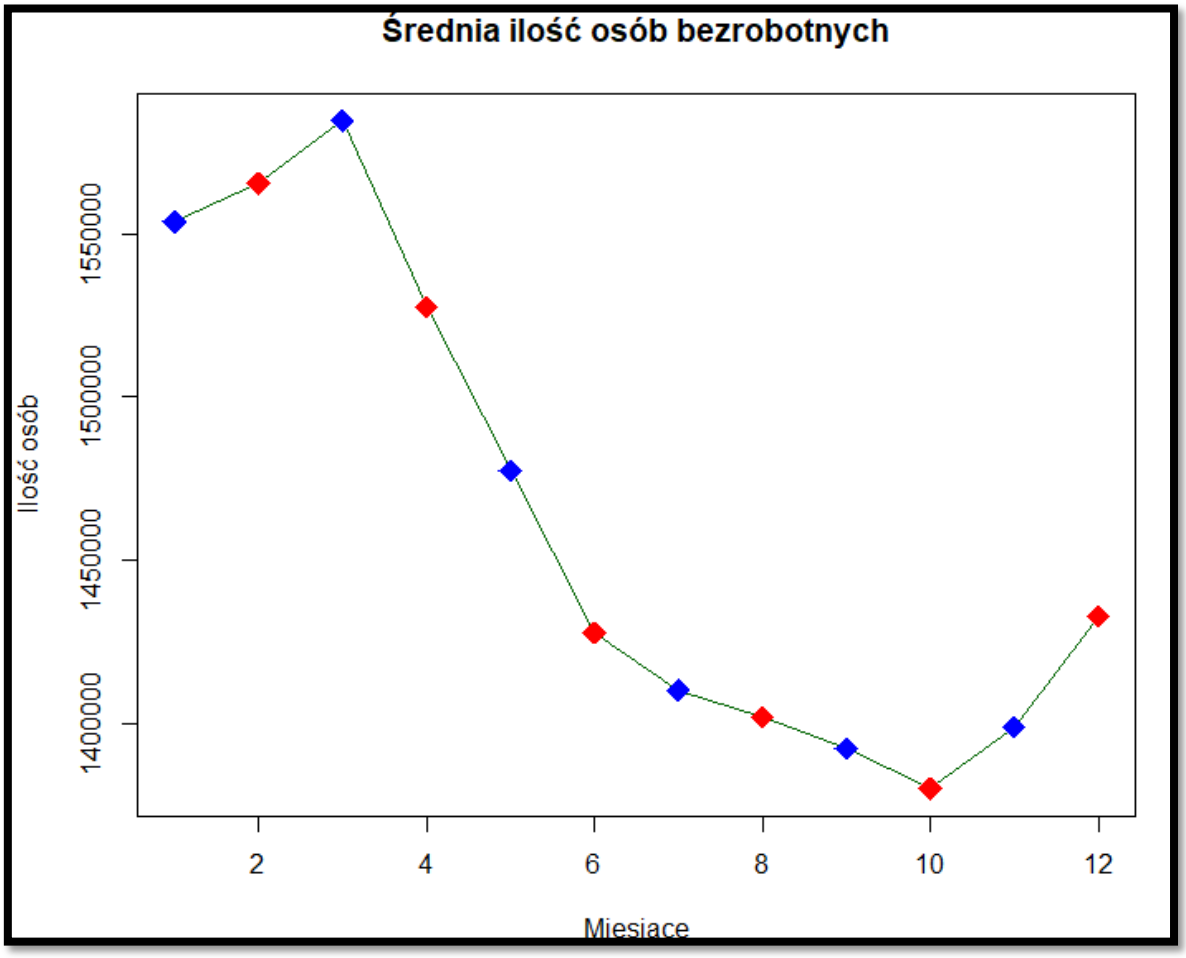
Utworzone wykresy pudełkowe ukazują pewną zależność między miesięczną ilością osób nowo zarejestrowanych jako bezrobotni oraz osób wyrejestrowanych. Możemy stwierdzić, że wykresy te są swego rodzaju swoją wzajemną odwrotnością. Ilość nowo zarejestrowanych wzrasta głównie w miesiącach jesienno-zimowych, a maleje w lecie. Odwrotną sytuację prezentuje wykres osób wyrejestrowanych. Wyniki tego wykresu mogą wskazywać na znajdywanie pracy przez osoby bezrobotne głównie w okresie letnim, co często jest związane z popularną w Polsce pracą sezonową.

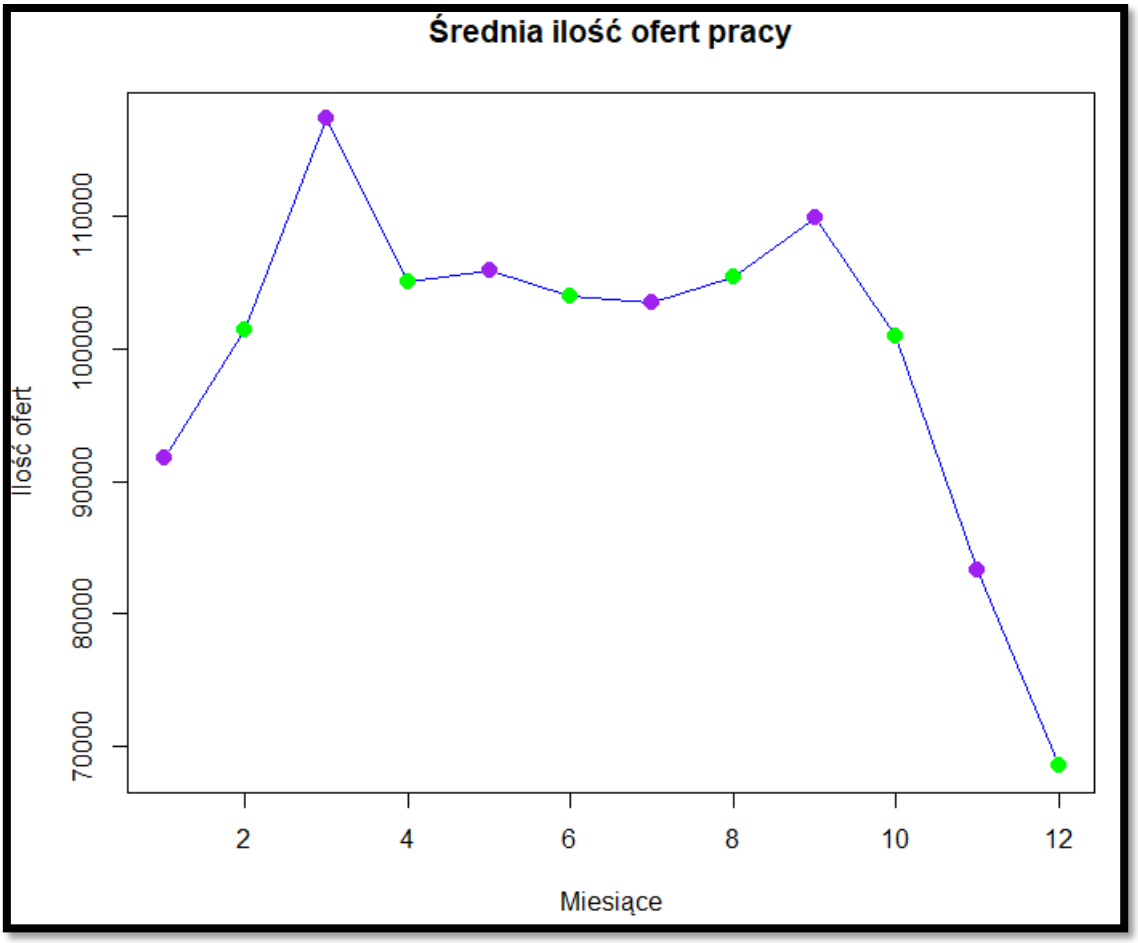
➢ Wykresy liniowe

Kolejnym typem wizualizacji, jaki zastosowałem, były wykresy liniowe. Prezentują one średnią ilość osób bezrobotnych oraz średnią liczbę ofert pracy w danych miesiącach.

Podobnie jak w przypadku wykresów pudełkowych, pracę rozpocząłem od utworzenia szeregów czasowych z danych wielkości, następnie za pomocą funkcji tapply wyliczyłem średnią z każdego miesiąca na podstawie wcześniej utworzonych szeregów czasowych. Same wykresy utworzyłem za pomocą polecenia plot, określając typ jako liniowy (type='l'). Z rzeczy nowych, dodałem do wykresu punkty za pomocą funkcji points oraz zmieniłem ich kolor, wielkość i kształt.

```r
### WYKRES LINIOWY ###

# Tworzenie szeregów czasowych
tsBO <- ts(data = dane$Liczba_bezrobotnych_ogolem, start = decimal_date(ymd("2010-01-01")), frequency = 12)
tsOP <- ts(data = dane$Liczba_zgloszonych_ofert_pracy, start = decimal_date(ymd("2010-01-01")), frequency = 12)

# Obliczanie średnich wartości dla każdego miesiąca
sr_miesieczna_BO <- tapply(tsBO, cycle(tsBO), mean)
sr_miesieczna_OP <- tapply(tsOP, cycle(tsOP), mean)

plot(sr_miesieczna_BO, type="l", col="darkgreen",
     main="Średnia ilość osób bezrobotnych",
     xlab="Miesiące", ylab="Ilość osób")
points(sr_miesieczna_BO, col=c("blue","red"), pch=18, cex=2)

plot(sr_miesieczna_OP, type="l", col="blue",
     main="Średnia ilość ofert pracy",
     xlab="Miesiące", ylab="Ilość ofert")
points(sr_miesieczna_OP, col=c("purple","green"), pch=20, cex=2)
```





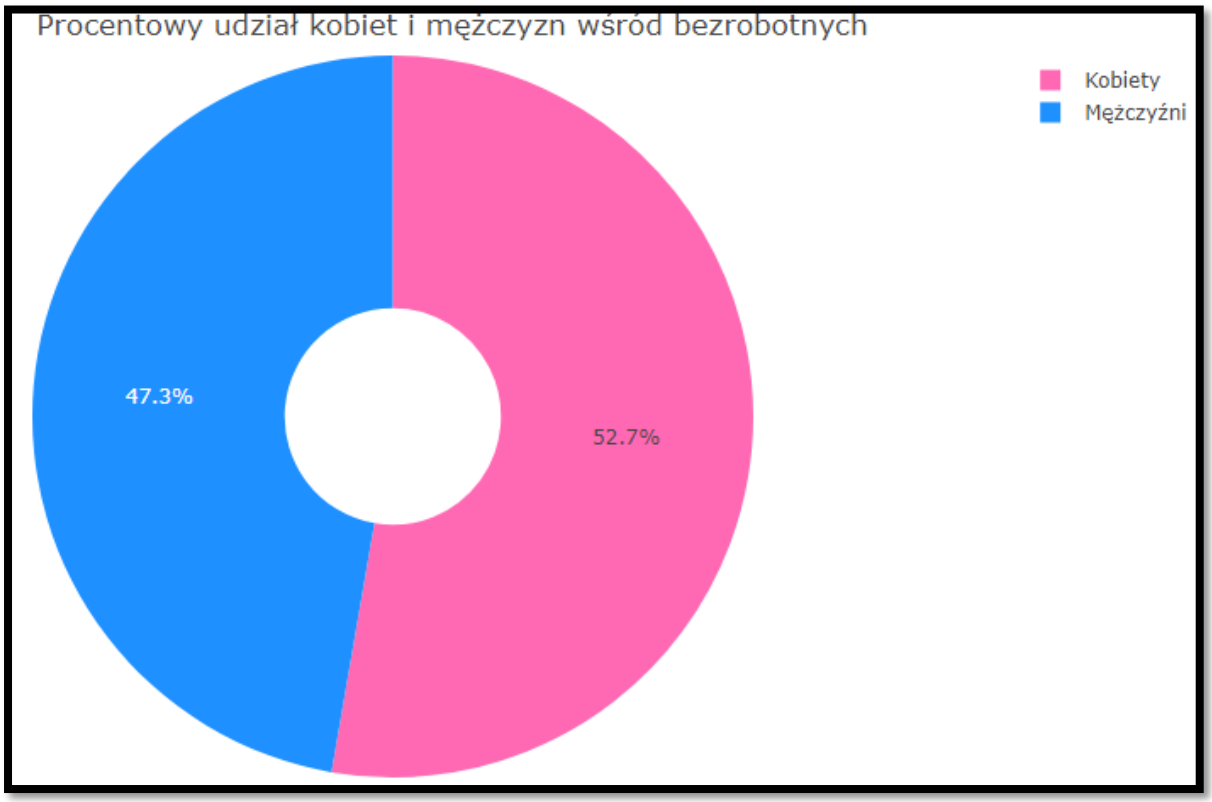
Wykresy liniowe mogą posłużyć do wyciągnięcia wniosków analogicznych do wykresów pudełkowych. Latem, liczba osób bezrobotnych diametralnie spada, zaś liczba ofert w tamtym okresie rośnie. Zimą natomiast, ilość osób zarejestrowanych jako bezrobotni zaczyna się zwiększać, natomiast swoje najniższe wartości obejmuje wtedy wartości zgłoszonych ofert pracy.

➢ Wykres kołowy

Najbardziej pasującą kategorią wśród moich danych do utworzenia wykresu kołowego jest płeć. Postanowiłem więc utworzyć wykres kołowy prezentujący procentowy udział kobiet i mężczyzn wśród osób bezrobotnych.

Aby tego dokonać obliczyłem najpierw sumę wszystkich bezrobotnych mężczyzn (odjąłem od ogólnej liczby osób bezrobotnych liczbę bezrobotnych kobiet). Następnie, obliczyłem wartości procentowe udziału kobiet i mężczyzn wśród wszystkich osób bezrobotnych. Kolejnym krokiem było utworzenie ramki danych, zawierających płeć oraz procent płci (ramka danych była kluczowa do wykonania wykresu za pomocą funkcji plot_ly). Kolejnym etapem było zastosowanie funkcji plot_ly z pakietu plotly do wyświetlenia wykresu kołowego. Jako argumenty funkcji podałem wcześniej utworzoną ramkę danych, etykiety (labels), wartości (values), typ wykresu (type), rozmiar dziury w środku wykresu (hole), a także parametr hoverinfo, który decyduje w jakim formacie mają być wyświetlone wartości na wykresie (w moim przypadku wartość+procent). Ponadto wybrałem kolory wykresu, a za pomocą funkcji layout dodałem tytuł wykresu.

```r
### WYKRES KOŁOWY ###

suma_ogolnie <- sum(dane$Liczba_bezrobotnych_ogolem)
suma_kobiet <- sum(dane$Liczba_bezrobotnych_kobiet)
suma_mezczyzn <- suma_ogolnie - suma_kobiet


# Obliczenie procentowego udziału kobiet i mężczyzn wśród bezrobotnych
procent_kobiet <- (suma_kobiet / suma_ogolnie) * 100
procent_mezczyzn <- (suma_mezczyzn / suma_ogolnie) * 100

# Tworzenie obiektu dataframe z wartościami procentowymi
dane_procentowe <- data.frame(
  płeć = c("Kobiety", "Mężczyźni"),
  procent = c(procent_kobiet, procent_mezczyzn)
)

# Tworzenie wykresu kołowego 3D
wykres <- plot_ly(dane_procentowe, labels = ~płeć, values = ~procent, type = "pie",
                  hole = 0.3, hoverinfo = "label+percent",
                  marker = list(colors = c("#FF69B4", "#1E90FF")))

# Dodanie tytułu wykresu i opisu osi Y
wykres <- layout(wykres, title = "Procentowy udział kobiet i mężczyzn wśród bezrobotnych")

# Wyświetlenie wykresu
wykres
```



Wykres okazuje niewielką przewagę kobiet w ilości osób bezrobotnych.

5. Hipotezy

➢ Hipoteza pierwsza

Załóżmy, że chcemy zbadać czy średnia stopa bezrobocia jest równa 10%.

Hipoteza zerowa H0: średnia stopa bezrobocia jest równa 10%.

Hipoteza alternatywna H1: średnia stopa bezrobocia jest mniejsza niż 10%.

Do rozpatrzenia hipotezy posłużyłem się funkcją t.test, w której zawarłem odpowiednie dane, wartość hipotezy zerowej oraz rodzaj hipotezy alternatywnej.

```
#hipoteza pierwsza
H1 <- t.test(x=dane$Stopa_bezrobocia,
             mu = 10,
             alternative="less")
H1
```

```
        One Sample t-test

data:  dane$Stopa_bezrobocia
t = -3.5305, df = 157, p-value = 0.0002723
alternative hypothesis: true mean is less than 10
95 percent confidence interval:
     -Inf 9.531212
sample estimates:
mean of x
 9.117722
```

Aby zdecydować, czy odrzucić hipotezę zerową na rzecz hipotezy alternatywnej, musiałem ocenić istotność statystyczną wyniku.

Wartość p (p-value) wynosi 0.0002723. Jest to bardzo mała wartość, znacznie mniejsza od poziomu istotności alfa = 0.05. Oznacza to, że mamy wystarczające dowody statystyczne, aby odrzucić hipotezę zerową. Hipoteza zerowa w tym przypadku głosi, że średnia stopa bezrobocia wynosi 10. Hipoteza alternatywna sugeruje, że średnia stopa bezrobocia jest mniejsza niż 10.

Przedział ufności 95% (-Inf, 9.531212) wskazuje, że z 95% pewnością różnica między średnią stopą bezrobocia a wartością 10 mieści się w zakresie od ujemnej nieskończoności do 9.531212.

Podsumowując, na podstawie wyników testu t-studenta i wartości p-value, możemy odrzucić hipotezę zerową na rzecz hipotezy alternatywnej.

➢ Hipoteza druga

Załóżmy, że chcemy zbadać czy wariancje liczby osób bezrobotnych nowo zarejestrowanych oraz wyrejestrowanych są istotnie różne.

Hipoteza zerowa H0: Wariancje liczby zgłoszonych ofert pracy oraz liczby osób wyrejestrowanych są równe

Hipoteza alternatywna H1: Wariancje liczby zgłoszonych ofert pracy oraz liczby osób wyrejestrowanych są różne

Do rozpatrzenia hipotezy posłużyłem się funkcją var.test, w której zawarłem odpowiednie grupy danych.

```
#hipoteza druga
H2 <- var.test(dane$Liczba_zgloszonych_ofert_pracy,
               dane$Bezrobotni_wyrejestrowani)
H2
```

```
        F test to compare two variances

data:  dane$Liczba_zgloszonych_ofert_pracy and dane$Bezrobotni_wyrejestrowani
F = 0.24132, num df = 157, denom df = 157, p-value < 2.2e-16
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1762889 0.3303294
sample estimates:
ratio of variances
          0.241316
```

Wartość p (p-value) jest bardzo mała, mniejsza niż 2.2e-16. Oznacza to, że mamy wystarczające dowody statystyczne, aby odrzucić hipotezę zerową.

Hipoteza zerowa w tym przypadku głosi, że stosunek wariancji między dwiema grupami wynosi 1. Hipoteza alternatywna sugeruje, że stosunek wariancji jest różny od 1. Na podstawie wyników testu var.test() oraz wartości p-value, mamy wystarczające dowody, aby odrzucić hipotezę zerową. Możemy przyjąć, że istnieje istotna statystyczna różnica między wariancjami dla tych dwóch grup. Wartość F jest mniejsza niż 1, co sugeruje, że wariancja liczby zgłoszonych ofert pracy jest mniejsza niż wariancja liczby bezrobotnych wyrejestrowanych.

Przedział ufności 95% (0.1762889, 0.3303294) wskazuje, że z 95% pewnością stosunek wariancji mieści się w zakresie od 0.1762889 do 0.3303294.

Podsumowując, na podstawie wyników testu var.test() i wartości p-value, możemy odrzucić hipotezę zerową na rzecz hipotezy alternatywnej, co sugeruje, że istnieje istotna statystyczna różnica między wariancjami dla tych dwóch grup.

6. Podsumowanie i wnioski

Stworzony przeze mnie projekt stanowił solidne podwaliny pod dalszą naukę związaną ze statystyką w języku R. Podczas zajęć projektowych oraz samodzielnego tworzenia projektu nabyłem wiedzę dotyczącą tego jakich funkcji użyć, aby obliczyć daną statystykę. Zanim do tego jednak przystąpiłem, kluczowym elementem było prawidłowe przygotowanie danych, co stanowiło delikatny problem. Nieobce mi jest również wizualizowanie otrzymanych danych oraz ich analizę poprzez wykresy pudełkowe, kołowe, liniowe czy histogramy. Najtrudniejszym, ponieważ również zupełnie dla mnie wcześniej obcym zagadnieniem były hipotezy. Istotną rzeczą przy rozpatrywaniu hipotez było zagłębienie się w dokumentację danej funkcji, aby lepiej zrozumieć jej działanie. Podsumowując, projekt ten stanowił solidną dawkę wiedzy z zakresu statystyki oraz programowania, którą z pewnością wykorzystam w przyszłości na studiach oraz być może w potencjalnej przyszłej pracy.

7. Lista użytych komend

Funkcja	Pakiet	Opis
setwd	base	Ustawia bieżący katalog roboczy
read_excel	readxl	Wczytuje dane z pliku Excela
as.data.frame	base	Konwertuje obiekt na ramkę danych
colnames	base	Ustawia nazwy kolumn w ramce danych
as.numeric	base	Konwertuje obiekt na typ liczbowy
mean	base	Oblicza średnią wartość
max	base	Znajduje maksymalną wartość
min	base	Znajduje minimalną wartość
sd	base	Oblicza odchylenie standardowe
var	base	Oblicza wariancję
median	base	Znajduje medianę
quantile	base	Oblicza kwantyle
IQR	base	Oblicza rozstęp międzykwartylowy
hist	graphics	Tworzy histogram
axis	graphics	Dodaje osie do wykresu

Funkcja	Pakiet	Opis
ts	base	Tworzy szereg czasowy
decimal_date	lubridate	Konwertuje daty na liczbę dziesiętną w formacie R lubridate
boxplot	graphics	Tworzy wykres pudełkowy
tapply	base	Aplikuje funkcję do podzbiorów danych określonych przez czynniki (w projekcie konkretnie do obliczania średnich wartości każdego miesiąca)
plot	graphics	Tworzy wykres
points	graphics	Dodaje punkty do istniejącego wykresu
sum	base	Oblicza sumę wartości
data.frame	base	Tworzy ramkę danych
plot_ly	plotly	Tworzy interaktywne, bardziej zaawansowane wykresy
layout	graphics	Ustawia układ i atrybuty dla wykresów
t.test	base	Przeprowadza test t-studenta
var.test	base	Przeprowadza test na zgodność wariancji między dwoma grupami