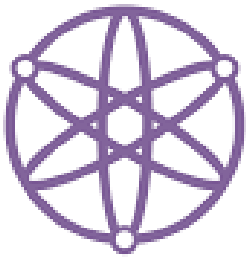


WYDZIAŁ
MATEMATYKI
I FIZYKI STOSOWANEJ
POLITECHNIKI RZESZOWSKIEJ

Projekt ze statystycznej analizy danych – ludzie bezrobotni uprzednio pracujący w latach 2014 i 2024

Stechnij Damian

Opiekun pracy:
dr Mariusz Startek

Rzeszów, 2024

Spis treści

1. Wprowadzenie

Projekt dotyczy statystycznej analizy danych ludzi bezrobotnych, którzy uprzednio pracowali. Dane zostały pobrane ze strony https://bdl.stat.gov.pl. Zawierają one lata od 2011 do 2024 z podziałem na miesiące oraz próbki ilości bezrobotnych pochodzą z wszystkich powiatów w Polsce. Wybrano te dane, aby przeprowadzić analizę różnicy w próbkach pomiędzy rokiem 2014 a 2024 w miesiącu styczeń. W projekcie wyznaczono parametry opisowe, przedstawiono graficznie zebrane dane oraz zbadano hipotezy statystyczne.

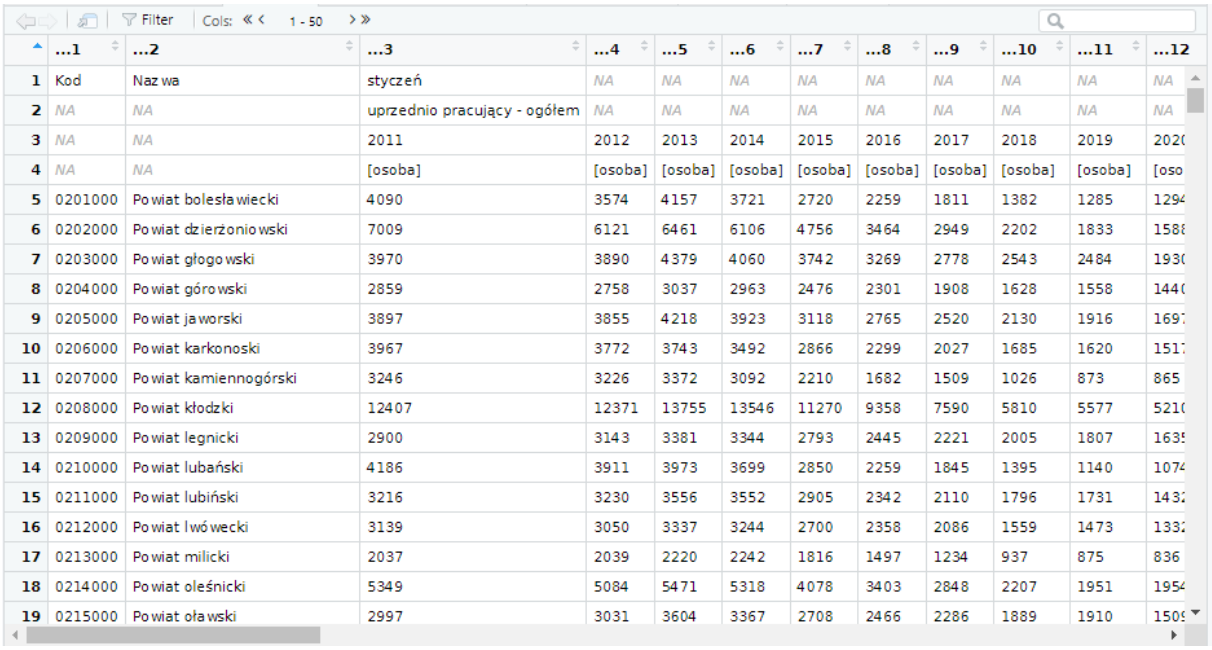
2. Wczytanie danych

Wczytywany plik jest w formacie .xlsx, więc do zaimportowania użyto biblioteki „readxl” i skorzystano z funkcji **read_excel**.

```
5  install.packages("readxl")
6  library(readxl)
7  dane ← read_excel('BEZROBOTNI_REJESTROWANI.xlsx', 2, col_names =  F)
8  View(dane)
```

Zrzut ekranu 2.1 Wczytanie danych

Po wczytaniu danych otrzymujemy taką tabele z ilością osób bezrobotnych wcześniej pracujących z poszczególnych powiatów od 2011 do 2024.



	...1	...2	...3	...4	...5	...6	...7	...8	...9	...10	...11	...12
1	Kod	Nazwa	styczeń	NA	NA	NA	NA	NA	NA	NA	NA	NA
2	NA	NA	uprzednio pracujący - ogółem	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	NA	NA	2011	2012	2013	2014	2015	2016	2017	2018	2019	202(
4	NA	NA	[osoba]	[osoba]	[osoba]	[osoba]	[osoba]	[osoba]	[osoba]	[osoba]	[osoba]	[oso
5	0201000	Powiat bolesławiecki	4090	3574	4157	3721	2720	2259	1811	1382	1285	1294
6	0202000	Powiat dzierżoniowski	7009	6121	6461	6106	4756	3464	2949	2202	1833	158(
7	0203000	Powiat głogowski	3970	3890	4379	4060	3742	3269	2778	2543	2484	193(
8	0204000	Powiat górowski	2859	2758	3037	2963	2476	2301	1908	1628	1558	144(
9	0205000	Powiat jaworski	3897	3855	4218	3923	3118	2765	2520	2130	1916	169(
10	0206000	Powiat karkonoski	3967	3772	3743	3492	2866	2299	2027	1685	1620	151(
11	0207000	Powiat kamiennogórski	3246	3226	3372	3092	2210	1682	1509	1026	873	865
12	0208000	Powiat kłodzki	12407	12371	13755	13546	11270	9358	7590	5810	5577	521(
13	0209000	Powiat legnicki	2900	3143	3381	3344	2793	2445	2221	2005	1807	163(
14	0210000	Powiat lubański	4186	3911	3973	3699	2850	2259	1845	1395	1140	107(
15	0211000	Powiat lubiński	3216	3230	3556	3552	2905	2342	2110	1796	1731	143(
16	0212000	Powiat lwówecki	3139	3050	3337	3244	2700	2358	2086	1559	1473	133(
17	0213000	Powiat milicki	2037	2039	2220	2242	1816	1497	1234	937	875	836
18	0214000	Powiat oleśnicki	5349	5084	5471	5318	4078	3403	2848	2207	1951	195(
19	0215000	Powiat oławski	2997	3031	3604	3367	2708	2466	2286	1889	1910	150(

Zrzut ekranu 2.2 Tabela danych po wczytaniu

Postanowiono zebrać dane tylko ze stycznia, więc obcięto dane z pozostałych miesięcy oraz z pola z wartościami NA.

```
11 library(dplyr)
12 dane ← dane %>% slice(-c(2,4))
13
14 # Tworzenie tabeli tylko ze stycznia
15
16 dane_styczen ← dane[,2:16]
17
18 h1 ← c(as.character(dane_styczen[1,1]),
19         as.character(dane_styczen[2,2:length(dane_styczen[2,])]))
20 h1
21 dane_styczen ← dane_styczen %>% slice(-c(1:2))
22 colnames(dane_styczen) ← h1
23
24 View(dane_styczen)
```

Zrzut ekranu 2.3 Obcięcie danych tylko do stycznia

Po obcięciu danych wybrano tylko lata 2014 i 2024, by porównać 10 lat odstępu i zamieniono na typ liczbowy.

```
26  # Chcemy tylko 2014 i 2024
27
28  gen_dane ← dane_styczen[c(1, 5, 15) ]
29
30  # Zamiana na typ liczbowy
31
32  (typeof(gen_dane$`2014`))
33  gen_dane$`2014` ← as.numeric(gen_dane$`2014`)
34  gen_dane$`2024` ← as.numeric(gen_dane$`2024`)
35
36  View(gen_dane)
```

Zrzut ekranu 2.4 Wybranie tylko lat 2014 i 2024

Gotowe dane po obróbce wyglądają następująco:

	Nazwa	2014	2024
1	Powiat bolesławiecki	3721	916
2	Powiat dzierżoniowski	6106	1703
3	Powiat głogowski	4060	1666
4	Powiat górowski	2963	1195
5	Powiat jaworski	3923	1526
6	Powiat karkonoski	3492	1547
7	Powiat kamiennogórski	3092	768
8	Powiat kłodzki	13546	5012
9	Powiat legnicki	3344	1509
10	Powiat lubański	3699	919
11	Powiat lubiński	3552	1252
12	Powiat lwówecki	3244	882
13	Powiat milicki	2242	768
14	Powiat oleśnicki	5318	2107
15	Powiat oławski	3367	1380
16	Powiat polkowicki	2905	1336
17	Powiat strzeliński	2534	1351
18	Powiat średzki	2241	950
19	Powiat świdnicki	8070	3447

Zrzut ekranu 2.5 Dane po obróbce

Przekształcono również dane pod wykresy, gdzie kolumny to Rok i Bezrobotni, aby później było łatwiej zaimplementować.

```
#install.packages("reshape2")
library(reshape2)
gen_dane_long ← melt(gen_dane, variable.name = 'Rok', value.name = 'Bezrobotni')
View(gen_dane_long)
```

Zrzut ekranu 2.6 Przekształcenie danych pod wykresy

	Nazwa	Rok	Bezrobotni
1	Powiat bolesławiecki	2014	3721
2	Powiat dzierżoniowski	2014	6106
3	Powiat głogowski	2014	4060
4	Powiat górowski	2014	2963
5	Powiat jaworski	2014	3923
6	Powiat karkonoski	2014	3492
7	Powiat kamiennogórski	2014	3092
8	Powiat kłodzki	2014	13546
9	Powiat legnicki	2014	3344
10	Powiat lubański	2014	3699
11	Powiat lubiński	2014	3552
12	Powiat lwówecki	2014	3244
13	Powiat milicki	2014	2242
14	Powiat oleśnicki	2014	5318
15	Powiat oławski	2014	3367
16	Powiat polkowicki	2014	2905
17	Powiat strzeliński	2014	2534
18	Powiat średzki	2014	2241
19	Powiat świdnicki	2014	8070
20	Powiat trzebnicki	2014	3709
21	Powiat wałbrzyski	2014	3995
22	Powiat wołowski	2014	3264
23	Powiat wrocławski	2014	2844

Zrzut ekranu 2.7 Tabele po przekształceniu pod wykresy

3. Wyznaczenie parametrów

3.1. Średnia

Obliczono średnią arytmetyczną z obu lat. Służy ona do określania wartości centralnej zbioru danych.

```
> (srednia_2014 ← mean(gen_dane$`2014`))
[1] 4892.547
> (srednia_2024 ← mean(gen_dane$`2024`))
[1] 1933.742
```

Zrzut ekranu 3.1 Obliczenie średniej

Średnia na przestrzeni lat zmieniła się ponad dwukrotnie, co oznacza, że o wiele mniej ludzi traci swoją pracę lub rzadziej zmieniają. Może też oznaczać, że liczby te się ujednolicają.

3.2. Wariancja

Następnie obliczono wariancję, która jest miarą rozproszenia danych wokół średniej wartości, dla obu kolumn. Wyniki mogą wysokie z powodu dużego rozrzutu wartości w danych.

```
> (war_2014 ← var(gen_dane$`2014`))
[1] 16064535
> (war_2024 ← var(gen_dane$`2024`))
[1] 2260744
```

Zrzut ekranu 3.2 Obliczenie wariancji

3.3. Odchylenie standardowe

Odchylenie standardowe określa, ile wartość danej cechy odchyla się od obliczonej średniej arytmetycznej. W tym przypadku jej wartość jest bardzo zbliżone do samej średniej. Może to oznaczać, że wartości mogą wahać się nawet 2-krotnie od średniej.

```
> (odchylenie_2014 ← sd(gen_dane$`2014`))
[1] 4008.059
> (odchylenie_2024 ← sd(gen_dane$`2024`))
[1] 1503.577
```

Zrzut ekranu 3.3 Obliczenie odchylenia standardowego

3.4. Współczynnik zmienności

Poniżej są wyniki dla współczynnika zmienności, czyli dyspersji, dla obu lat. Dla 2014 wynosi blisko 82% a dla 2024 - 77,75%, co można zinterpretować jako znaczną dyspersję.

```
> (wspol_zm_2014 ← (odchylenie_2014 / srednia_2014) * 100)
[1] 81.92172
> (wspol_zm_2024 ← (odchylenie_2024 / srednia_2024) * 100)
[1] 77.75479
```

Zrzut ekranu 3.4 Obliczenie współczynnika zmienności

3.5. Kwartyle, mediana, minimum, maksimum

Kwartyle dzielą zbiory na ćwiartki. Przykładowo pierwszy kwartyl oznacza, że 25% danych jest mniejszych niż Q1, a pozostałych 75% jest większych niż Q1. Mediana jest drugim kwartylem i dzieli badany zbiór na połowy.

```
> (kwantyl1_2014 ← quantile(gen_dane$`2014`, probs = 0.25))
 25%
2963
> (mediana_2014 ← median(gen_dane$`2014`))
[1] 4003
> (kwantyl3_2014 ← quantile(gen_dane$`2014`, probs = 0.75))
   75%
5710.5
> (kwantyl1_2024 ← quantile(gen_dane$`2024`, probs = 0.25))
    25%
1179.75
> (mediana_2024 ← median(gen_dane$`2024`))
[1] 1611
> (kwantyl3_2024 ← quantile(gen_dane$`2024`, probs = 0.75))
    75%
2279.25
```

Zrzut ekranu 3.5 Obliczenie kwantyli

Poniżej znaleziono również wartości maksymalne jak i minimalne:

```
> # Wartość minimalna dla każdej kolumny
> (min_values ← sapply(gen_dane[, c("2014", "2024")], min))
2014 2024
 830  306
> # Wartość maksymalna dla każdej kolumny
> (max_values ← sapply(gen_dane[, c("2014", "2024")], max))
 2014  2024
49190 16961
```

Zrzut ekranu 3.6 Znalezienie wartości maksymalnej i minimalnej

3.6. Dominanta

Dominanta to wartość, którą wyznacza się poprzez znalezienie najczęściej występującej liczby w zbiorze danych.

```
> (dominanta_2014 ← as.numeric(names(which.max(table(gen_dane$`2014`)))))
[1] 2963
> (dominanta_2024 ← as.numeric(names(which.max(table(gen_dane$`2024`)))))
[1] 768
```

Zrzut ekranu 3.7 Obliczenie dominanty

Przedstawione wartości są całkiem niskie, więc może to oznaczać, że w wielu powiatach mało ludzi traci pracę, a na przestrzeni ostatnich 10 lat zmieniło się to niemal 4-krotnie.

3.7. Rozstęp

Rozstęp to miara odległości jaką dzieli pomiędzy wartością maksymalną a minimalną obranej cechy statystycznej.

```
> (rozstep_2014 ← max_values[1] - min_values[1])
 2014
48360
> (rozstep_2024 ← max_values[2]- min_values[2])
 2024
16655
```

Zrzut ekranu 3.8 Obliczenie rozstępu

W roku 2014 jest znacząco wyższy niż w 2024, co może oznaczać, że w o wielu mniej miejscach ludzie są bezrobotni i jest ich mniej.

3.8. Skośność

Skośność to miara asymetrii rozkładu zmiennej. Bada jak rozkład jest zbliżony do rozkładu normalnego. By móc użyć polecenia **skewness** użyto biblioteki „e1071”.

```
> (skosnosc_2014 ← round(skewness(gen_dane$`2014`), 3))
[1] 5.843
> (skosnosc_2024 ← round(skewness(gen_dane$`2024`), 3))
[1] 4.877
```

Zrzut ekranu 3.9 Obliczenie skośności

Dodatnie wyniki informują, że prawie ramię rozkładu jest wydłużone i wyniki poniżej średniej są przeważające w badanej próbce. Większa wartość w 2014 mówi o tym, że ramię jest bardziej wysunięte, niż 10 lat później.

3.9. Kurtoza

Kurtoza jest miarą występowania wartości odstających. Podobnie jak skośność wskazuje jak bardzo rozkład jest zbliżony do rozkładu normalnego. Im bliżej wartości 0, tym bardziej analizowany rozkład jest zbliżony do normalnego.

```
> (kurtoza_2014 ← kurtosis(gen_dane$`2014`))
[1] 52.44169
> (kurtoza_2014 ← kurtosis(gen_dane$`2024`))
[1] 38.12006
```

Zrzut ekranu 3.10 Wyniki dla kurtozy

Obliczona kurtoza jest większa od zera co wskazuje na leptokurtyczność oraz znacząco odbiega co sugeruje, że dane zasadniczo różnią się od rozkładu normalnego. Tak wysokie wyniki wskazują na bardzo spiczasty szczyt oraz dłuższe i cieńsze ogony w porównaniu do rozkładu normalnego. Oznacza to, że jest większe prawdopodobieństwo wystąpienia wartości odstających.

3.10. Zestawienie wyników

Poniżej zestawiono obliczone wartości parametrów w tabeli:

Rok	2014	2024
Średnia	4892,547	1933,742
Wariancja	16064535	2260744
Odchylenie standardowe	4008,059	1503,577
Współczynnik zmienności	81,922 %	77,755 %
Kwartyl 1. rzedu	2963	1179,75
mediana	4003	1611
Kwartyl 3. rzędu	5710,5	2279,25
Minimum	830	306
Maksymum	49190	16961
Dominanta	2963	768
Rozstęp	48360	16655
Skośność	5,843	4,877
Kurtoza	52,44169	38,12006

Tabela 3.1 Wartości parametrów

4. Graficzna prezentacja danych

4.1. Wykres pudełkowy

Na wykresie pudełkowy można przedstawić wiele informacji takie jak mediana, kwartyle, wartość minimum i maximum , przez co jest często używany.

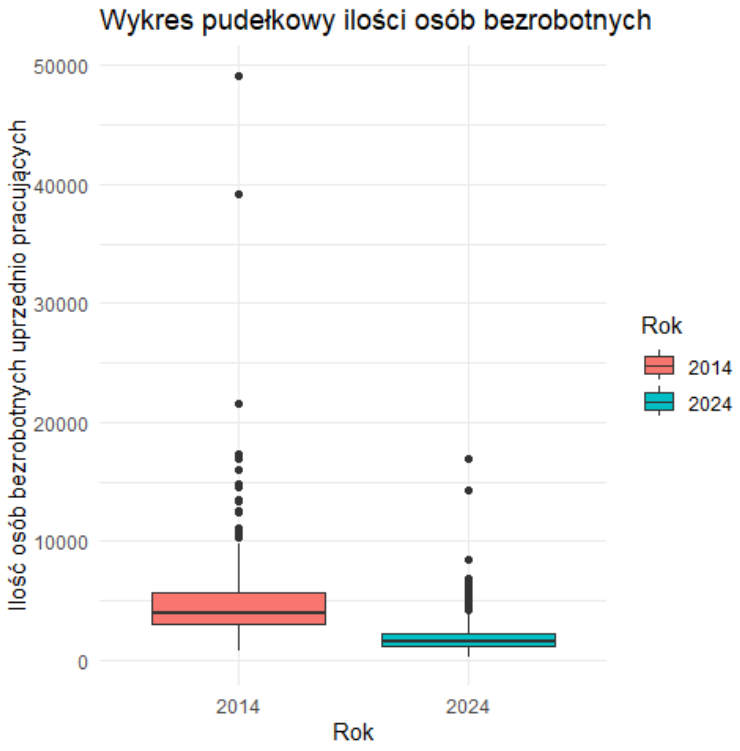
```
## Wykres pudełkowy

#install.packages("reshape2")
library(reshape2)
gen_dane_long ← melt(gen_dane, variable.name = 'Rok', value.name = 'Bezrobotni')
View(gen_dane_long)

ggplot(gen_dane_long, aes(x = Rok, y = Bezrobotni, fill = Rok)) +
  geom_boxplot() +
  labs(title = "Wykres pudełkowy ilości osób bezrobotnych",
       y = "Ilość osób bezrobotnych uprzednio pracujących") +
  theme_minimal()
```

Zrzut ekranu 4.1 Kod tworzenia wykresu pudełkowego

Na wykresach są bardzo duże wartości odstające. Tam, gdzie jest dolna ściana pudełka to jest to pierwszy kwartyl, a górna to trzeci kwartyl. Pozioma linia w środku pudełka to mediana. Wąsy to linie poza pudełkiem i ich końce mogą oznaczać od dołu wartość minimalna i od góry wartość maksymalną. W tym wypadku w górnej części wykresu są wartości odstające.



Rysunek 4-1 Wykres pudełkowy

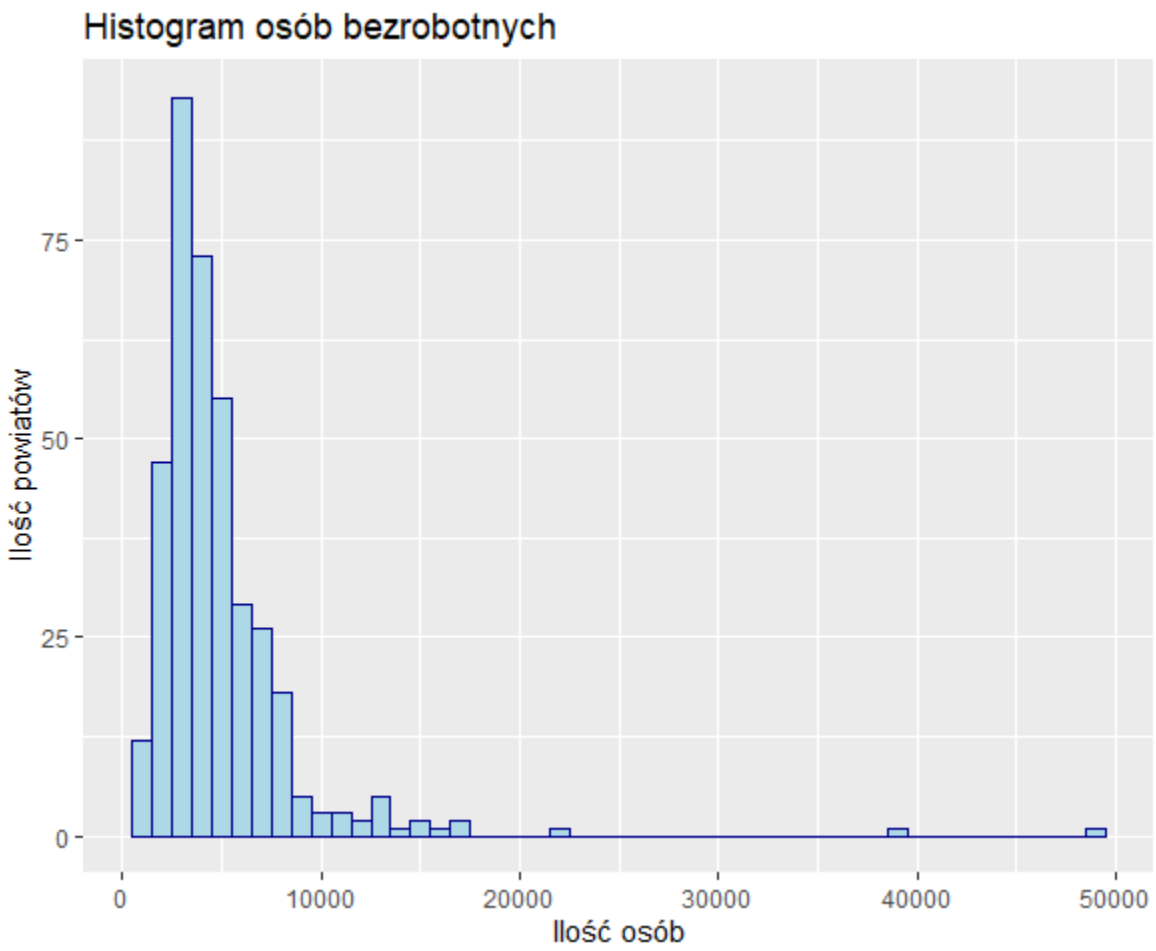
4.2. Histogram

Histogram to wykres przedstawiający rozkładu empirycznego cechy. Jest podobny do wykresu słupkowego. Wysokość słupków, to ile elementów znajduje się w przedziale a szerokość słupków to przedział, w którym mieszczą się elementy.

```
128 ## Histogram
129
130 ggplot(gen_dane, aes(x=`2014`)) +
131   geom_histogram( binwidth=1000,
132                   color="darkblue", fill="lightblue") +
133   labs(title="Histogram osób bezrobotnych(przedział 1000 osób)",
134        x="Ilość osób", y="Ilość powiatów")
```

Zrzut ekranu 4.2 Kod tworzenia histogramu

W kodzie ustawiono przedział, czyli szerokość słupka, na 1000. Poniżej znajduje się histogram.



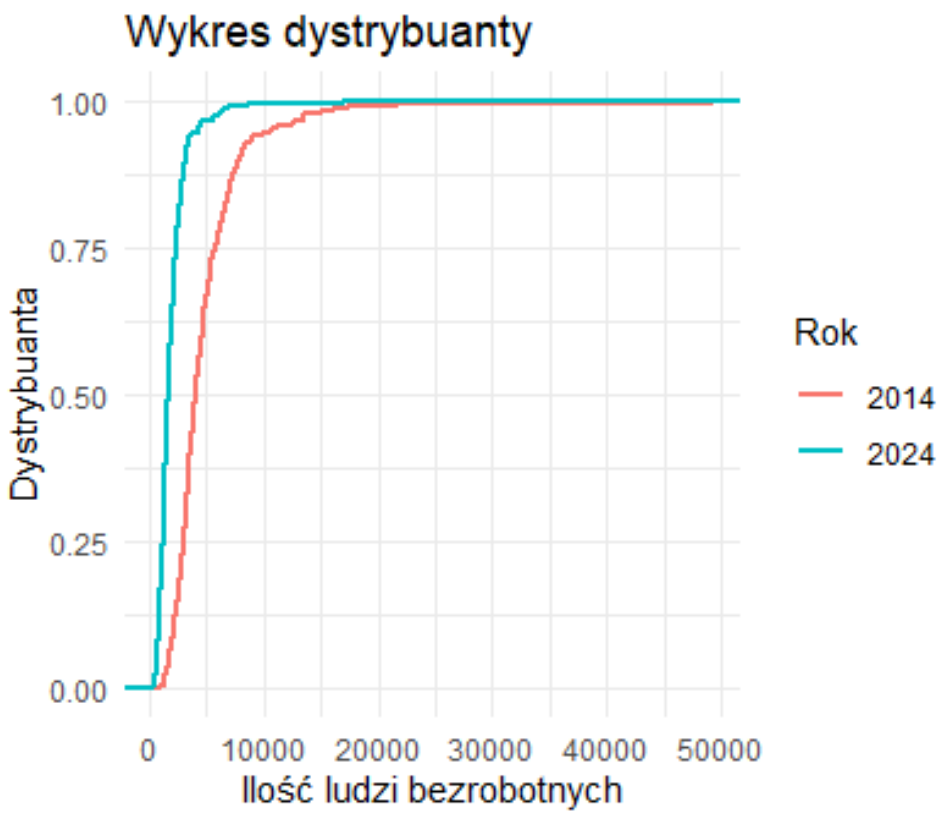
Rysunek 4-2 Histogram

4.3. Wykres dystrybuanty

Wykres dystrybuanty przedstawia rozkład prawdopodobieństwa występowania określonych wartości.

```
135 ## Wykres dystrybuanty
136
137 ggplot(gen_dane_long, aes(x = Bezrobotni, color = Rok)) +
138   stat_ecdf(size = 1) +
139   labs(title = "Wykres dystrybuanty",
140        x = "Ilość ludzi bezrobotnych",
141        y = "Dystrybuanta",
142        color = "Rok") +
143   theme_minimal() +
144   theme(legend.position = "right")
```

Zrzut ekranu 4.3 Kod tworzenia wykresu dystrybuanty



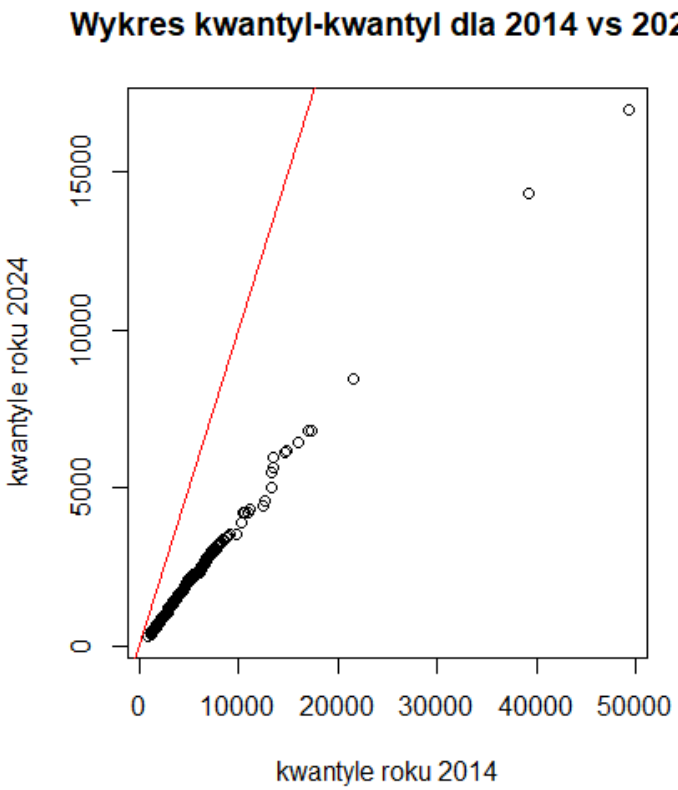
Rysunek 4-3 Wykres dystrybuanty

4.4. Wykres kwantyl-kwantyl

Wykres kwantyl-kwantyl można użyć do porównania danych z jakimś teoretycznym rozkładem. Służy do przedstawienia porównania kwantyli dwóch prób danych. W tym przypadku użyto do porównania obu prób danych i wizualizacji podobieństwa ich rozkładów co pozwala na ich ocenę.

```
147 ## Wykres kwantyl - kwantyl
148
149 qqplot(gen_dane$`2014`, gen_dane$`2024`,
150         main = "Wykres kwantyl-kwantyl dla 2014 vs 2024",
151         xlab = "kwantyle roku 2014", ylab = "kwantyle roku 2024")
152
153 # Dodanie linii referencyjnej
154 abline(0, 1, col = "red")
```

Zrzut ekranu 4.4 Kod tworzenia wykresu kwantyl-kwantyl



Rysunek 4-4 Wykres kwantyl-kwantyl

Czerwona linia to linia referencyjna o równaniu $y = x$. Służy jako odniesienie do porównania rozkładów, pokazuje, gdzie leżałyby punkty, jeśli oba rozkłady były identyczne.

Na powyższym wykresie punkty leżą poniżej tej linii, co wskazuje, że wartości z roku 2024 są z reguły mniejsze niż te z roku 2014 dla odpowiadających kwantyli.

5. Hipotezy statystyczne

5.1. Hipoteza dotycząca rozkładu bezrobotnych

Drugą, poddaną testowi, zostaje hipoteza dotyczy tego czy dane posiadają rozkład normalny. Został do tego wykorzystany test Shapiro-Wilka.

H0: Dane pochodzą z rozkładu normalnego.

H1: Dane nie pochodzą z rozkładu normalnego.

```
> shapiro_test_2014 ← shapiro.test(gen_dane$`2014`)
> print(shapiro_test_2014)

        Shapiro-Wilk normality test

data:  gen_dane$`2014`
W = 0.57501, p-value < 2.2e-16

> shapiro_test_2014$p.value
[1] 2.146899e-29
> shapiro_test_2024 ← shapiro.test(gen_dane$`2024`)
> print(shapiro_test_2024)

        Shapiro-Wilk normality test

data:  gen_dane$`2024`
W = 0.63525, p-value < 2.2e-16

> shapiro_test_2024$p.value
[1] 1.206036e-27
```

Zrzut ekranu 5.1 Test Shapiro-Wilka

Wartość p wynosi 4.525356e-10 co znaczy ze jest bardzo bliskie 0 i jest znacznie mniejsze niż poziom istotności 0.05. Na tej podstawie odrzucamy hipotezę zerową i przyjmujemy hipotezę alternatywną, która przyjmuje, że dane nie pochodzą z rozkładu normalnego.

5.2. Hipoteza dotycząca mediany liczby bezrobotnych

Do sprawdzenia hipotezy użyto nieparametrycznego testu Wilcoxona, który sprawdza, czy różnica pomiędzy medianami w dwóch próbach jest taka sama. Nasze dane są zależne, ponieważ zostały wykonane w tych samych powiatach, tyle że w odstępie czasowym 10 lat.

H0: Mediana ludzi bezrobotnych w powiatach nie różni się istotnie po 10 latach.

$$m_0 = m_1 \qquad \text{Wzór 1}$$

H1: Mediana ludzi bezrobotnych w powiatach różni się istotnie po 10 latach.

$$m_0 \neq m_1 \qquad \text{Wzór 2}$$

```
> wilc ← wilcox.test(gen_dane$`2014`, gen_dane$`2024`,
+                    paired = TRUE)
> print(wilc)

        Wilcoxon signed rank test with continuity correction

data:  gen_dane$`2014` and gen_dane$`2024`
V = 72390, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

> wilc$p.value
[1] 5.08303e-64
```

Zrzut ekranu 5.2 Test Wilcoxona

Wartość p jest bardzo bliska zeru i jest mniejsza niż poziom istotności 0,05. Na tej podstawie odrzucamy hipotezę zerową i możemy uznać hipotezę alternatywną za prawdziwą.

6. Użyte biblioteki i polecenia

6.1. Do wczytania danych

Do wczytania danych użyto biblioteki „readxl”, dzięki której łatwiej wyciągnąć dane z Excela do R. Przy pomocy polecenia **read_excel()** można odcztać pliki o rozszerzeniu .xls, .xlsx.

Przy obróbce danych wykorzystano bibliotekę „dplyr”, by skorzystać z operatora **%>%**, by łatwiej było odcinać niepotrzebne kolumny i wiersze.

As.numeric() – zamienia dane na typ liczbowy

6.2. Do obliczeń parametrów

mean() – oblicza średnią arytmetyczną

var() – oblicza wariancję

sd() – oblicza odchylenie standardowe

quantile() – oblicza kwantyle, dzięki **probs** określa, jak mają zostać podzielone dane

mediane() – znajduje medianę

sapply(dane, min/max) – znajduje wartości minimalne lub maksymalne

skewness() – oblicza skośność, pochodzi z biblioteki „e1071”

kurtosis() – oblicza kurtozę, również z biblioteki „e1071”

Dzięki **which.max()** znaleziono dominantę.

6.3. Do wykresów

Użyto bibliotekę „reshape2”, by przy pomocy polecenia melt() móc przekształcić dane, dzięki czemu łatwiej można było zaimplementować je do wykresów.

Aby utworzyć wykresy wykorzystano bibliotekę „ggplot2”.

ggplot(`data = df, mapping = aes(x, y, other aesthetics`**)** – inicjuje obiekt wykres,

aes() – określa co jest jako x i co jako y

geom_boxplot() – tworzy wykres pudełkowy

geom_histogram() – nakłada warstwę z histogramem

stat_ecdf() – tworzy wykres dystrybuanty

labs() – tworzy etykiety do legend

theme_minimal() – ustawia minimalny styl wykresu

6.4. Do hipotez

wilcox.test() – wykonuje test Wilcoxona, parametr paired = TRUE dla danych sparowanych

shapiro.test() – wykonuje test Shapiro-Wilka, by zbadać, czy dane mają rozkład normalny

7. Wnioski

Analizując dane o bezrobotnych, którzy uprzednio pracowali po odstępie 10 lat, pomiędzy rokiem 2014 a 2024 w miesiącu styczeń, można wyciągnąć pewne wnioski.

Rynek pracy się zmienił na tyle, by o wiele mniej ludzi traci pracę lub o wiele chętniej zostają w swojej pracy. Po wszelkich parametrach było widać, że te dane są znacząco mniejsze, bo nawet ponad dwukrotnie. Średnie po takim odstępie czasu znacząco się różniły.

Po graficznym przedstawieniu danych poprzez histogram czy wykres pudełkowy można zaobserwować asymetrie lewostronną, która na przestrzeni lat jeszcze bardziej się zmieniła. Można również stwierdzić, że rozrzut danych się istotnie zmniejszył. Analiza ta pomaga w dostrzeżeniu zmian na polskim rynku pracy.