Zadania

May 2, 2025

1 Realizacja część 1

Przygotował: Paweł Jan Tłusty

IDE: jupyter studio + IRKernel

Export do pdf: pandoc

```
sudo apt-get install texlive texlive-latex-extra pandoc texlive-xetex
```

Wersja online: Notes dostępny również w serwisie GitHub (niektóre wykresy niewłaściwie dzedziczą style, zaleca zaleca się jasny motyw lub otwieranie poszczególnych wykresów jako obraz w nowej karcie) Link:

https://github.com/Kotmin/R-Planowanie-Eksperymentu

W związku z konwertowaniem notatnika jpt do innych formatów miejscami może zdarzyć się błędne kodowanie znaków specjalnych, w niniejszym dokumencie zakładamy, że wspominajac o **braku podstaw do odrzucenia hipotezy** …, przyjmujemy domyślnie, że mówimy o hipotezie zerowej.

1.1 Zadanie 1

Liczba strzelonych bramek w kolejnych meczach przez pewną drużynę piłkarską jest następująca:

2, 3, 0, 0, 1, 3, 1, 0, 2, 0, 1, 1, 1, 0, 3, 2, 0, 2, 0, 1, 1, 2, 0, 3, 1, 1, 0, 1, 0, 2, 4, 1, 0, 0, 1, 2, 0, 1, 1, 0, 1, 0, 2

Zbadać, czy ilość zdobytych goli w meczu jest zgodna z rozkładem Poissona. Parametry rozkładu oszacować na podstawie danych.

1.1.1 Hipotezy statystyczne:

- **H (hipoteza zerowa):** rozkład liczby goli jest zgodny z rozkładem Poissona.
- **H (hipoteza alternatywna):** rozkład liczby goli nie jest zgodny z rozkładem Poissona.

[1]:
```
gole <- c(2, 3, 0, 0, 1, 3, 1, 0, 2, 0, 1, 1, 1, 0, 3, 2, 0, 2, 0, 1, 1, 2, 0,␣
↪3,
          1, 1, 0, 1, 0, 2, 4, 1, 0, 0, 1, 2, 0, 1, 1, 0, 1, 0, 2)
```

[2]:
```
n <- length(gole)
```

[3]:
```
n
```

43

[4]:
```
# Oszacowanie parametru   dla rozkładu Poissona
lambda_hat <- mean(gole)
```

[5]:
```
lambda_hat
```

1.09302325581395

[6]:
```
## Dane empiryczne
### Ile razy występuje dana liczba goli
obs <- table(gole)
obs
```

```
gole
 0  1  2  3  4
15 15  8  4  1
```

[7]:
```
k <- 0:max(gole)
```

[8]:
```
k
```

1. 0 2. 1 3. 2 4. 3 5. 4

[9]:
```
# Teoretyczne prawdopodobieństwa z rozkładu Poissona
probs <- dpois(k, lambda_hat)
```

[10]:
```
probs
```

1. 0.335201560212229 2. 0.366383100697087 3. 0.200232624799571 4. 0.0729529718262003 5. 0.0199348236966943

[11]:
```
# Oczekiwane liczności
exp <- probs * n
names(exp) <- k

exp
```

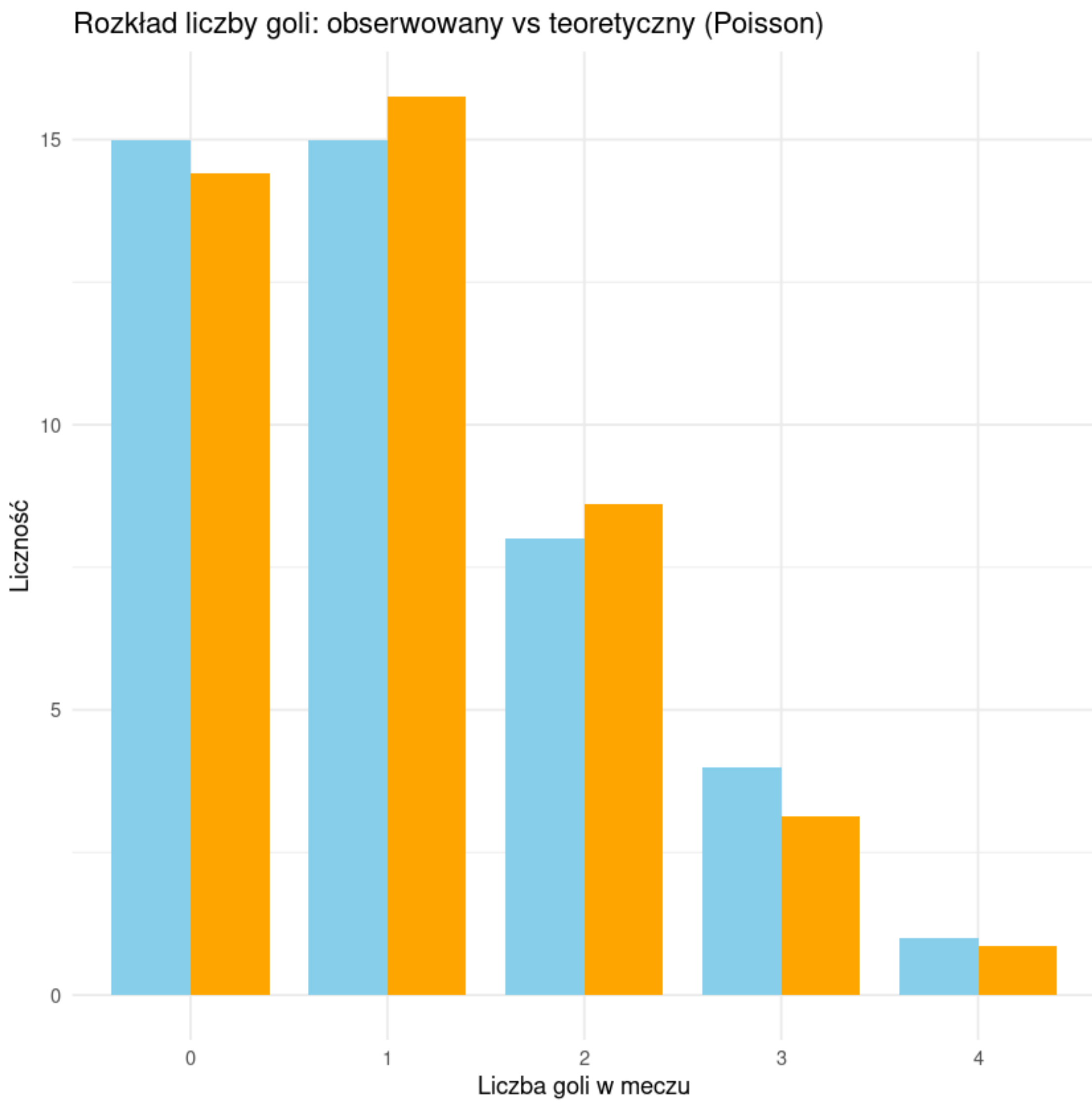
0 14.4136670891258 **1** 15.7544733299748 **2** 8.61000286638155 **3** 3.13697778852661 **4** 0.857197418957853

[12]:
```
## Wizualizacja
df <- data.frame(
  gole = factor(names(obs), levels = as.character(0:max(gole))),
  obserwowane = as.numeric(obs),
```

```
  oczekiwane = as.numeric(exp)
)
```

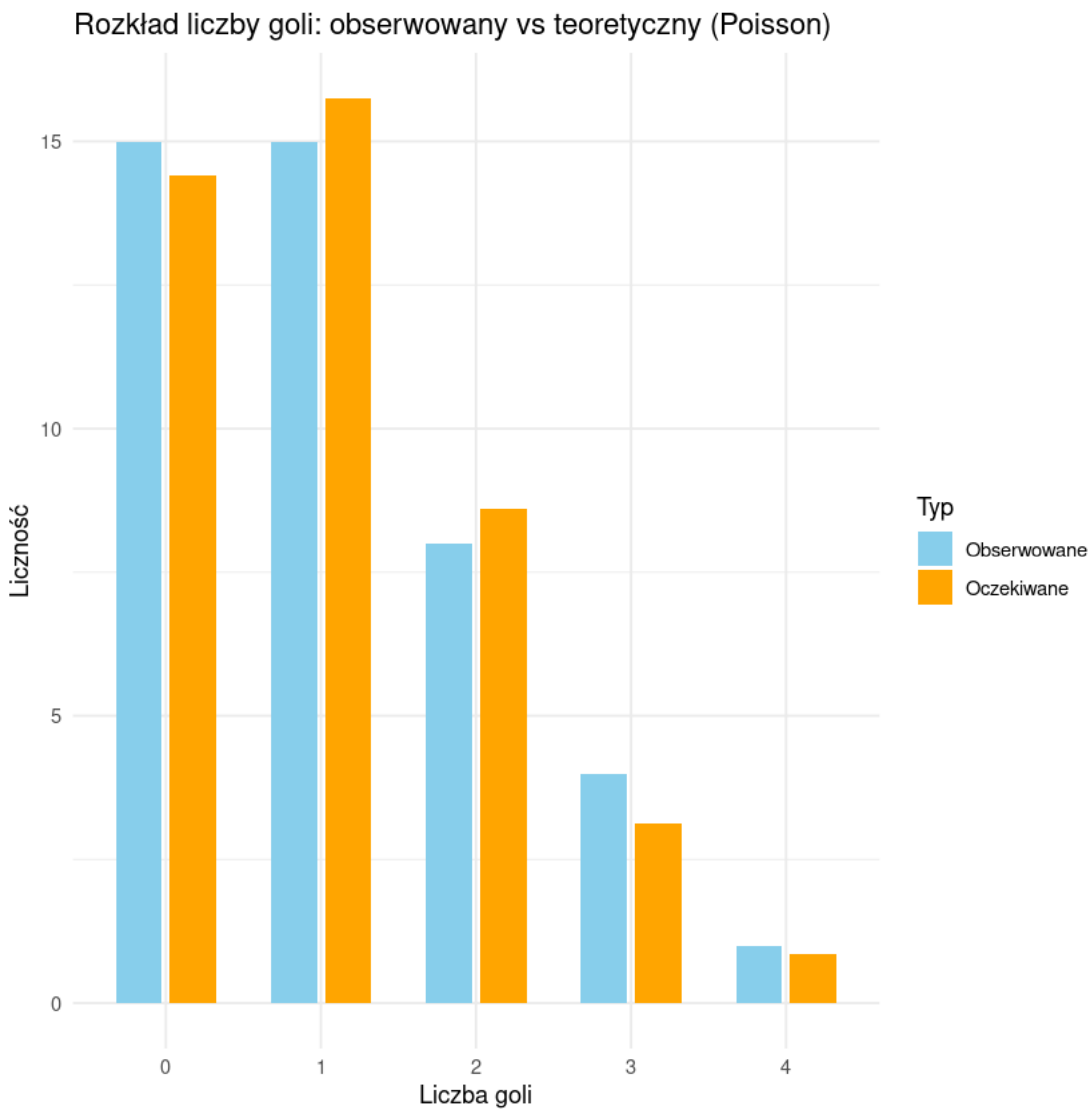
[13]:
```
# Załadowanie biblioteki
library(ggplot2)
```

[14]:
```
ggplot(df, aes(x = gole)) +
  geom_bar(aes(y = obserwowane), stat = "identity", fill = "skyblue", width = 0.
↪4, position = position_nudge(x = -0.2)) +
  geom_bar(aes(y = oczekiwane), stat = "identity", fill = "orange", width = 0.
↪4, position = position_nudge(x = 0.2)) +
  labs(
    title = "Rozkład liczby goli: obserwowany vs teoretyczny (Poisson)",
    x = "Liczba goli w meczu",
    y = "Liczność"
  ) +
  theme_minimal()
```



[20]:
```
gole_kategorie <- as.character(0:max(gole))

obserwowane <- as.numeric(table(factor(gole, levels = 0:max(gole))))
oczekiwane <- exp

df_obserw <- data.frame(gole = gole_kategorie, liczność = obserwowane, typ =
  "Obserwowane")
df_oczek <- data.frame(gole = gole_kategorie, liczność = oczekiwane, typ =
  "Oczekiwane")
df_final <- rbind(df_obserw, df_oczek)

ggplot(df_final, aes(x = gole, y = liczność, fill = typ)) +
```

```
  geom_bar(stat = "identity", position = position_dodge(width = 0.7), width = 0.
↪6) +
  scale_fill_manual(values = c("Obserwowane" = "skyblue", "Oczekiwane" =␣
↪"orange")) +
  labs(
    title = "Rozkład liczby goli: obserwowany vs teoretyczny (Poisson)",
    x = "Liczba goli",
    y = "Liczność",
    fill = "Typ"
  ) +
  theme_minimal()
```



[ ]:
```
## Teraz powinniśmy przeprowadzić test zgodności
```

[21]:
```
# Oczekiwane liczności
oczekiwane

# Warunki:
sum(oczekiwane < 1)                 # ile klas ma < 1
sum(oczekiwane < 5) / length(oczekiwane)   # jaki % ma < 5
```

0 14.4136670891258 **1** 15.7544733299748 **2** 8.61000286638155 **3** 3.13697778852661 **4** 0.857197418957853

1

0.4

[22]:
```
# bazując na tych danych dobrze by było połączyć 3 i 4
obs
```

```
gole
 0  1  2  3  4
15 15  8  4  1
```

[23]:
```
obs["3+"] <- sum(obs["3"], obs["4"])
obs <- obs[c("0", "1", "2", "3+")]
```

[24]:
```
obs
```

```
 0  1  2 3+
15 15  8  5
```

[25]:
```
oczekiwane
```

0 14.4136670891258 **1** 15.7544733299748 **2** 8.61000286638155 **3** 3.13697778852661 **4** 0.857197418957853

[27]:
```
oczekiwane["3+"] <- sum(oczekiwane[4:5])
```

[28]:
```
oczekiwane
```

0 14.4136670891258 **1** 15.7544733299748 **2** 8.61000286638155 **3** 3.13697778852661 **4** 0.857197418957853 **3+** 3.99417520748446

[29]:
```
oczekiwane <- oczekiwane[c(1:3, 6)]
names(oczekiwane) <- names(obs)
```

[30]:
```
oczekiwane
```

0 14.4136670891258 **1** 15.7544733299748 **2** 8.61000286638155 **3+** 3.99417520748446

[31]:
```
test_chikwadrat <- chisq.test(
  x = as.numeric(obs),
```

```
  p = oczekiwane / sum(oczekiwane),
  rescale.p = TRUE
)
```

```
Warning message in chisq.test(x = as.numeric(obs), p =
oczekiwane/sum(oczekiwane), :
"Chi-squared approximation may be incorrect"
```

[32]:
```
test_chikwadrat
```

```
        Chi-squared test for given probabilities

data:  as.numeric(obs)
X-squared = 0.3534, df = 3, p-value = 0.9497
```

[33]:
```
rozn_bezwzgl <-  abs(obs - oczekiwane)
procent_dopasowanych <-mean(rozn_bezwzgl <= 1) * 100
```

[34]:
```
cat("Dopasowanie (klas z różnicą   1):", round(procent_dopasowanych, 1), "%\n")
```

```
Dopasowanie (klas z różnicą   1): 75 %
```

1.1.2 Wnioski zadanie 1

Hipotezy statystyczne: - **H (hipoteza zerowa):** rozkład liczby goli jest zgodny z rozkładem Poissona. - **H (hipoteza alternatywna):** rozkład liczby goli nie jest zgodny z rozkładem Poissona.

Wniosek: Brak podstaw do odrzucenia hipotezy zerowej. p-value - bardzo duże / znacznie większe od 0.05.

Przemyślenia: Być może dodatkowa weryfiakcja przy pomocy Monte Carlo?

1.2 Zadanie 2: Weryfikacja zgodności z rozkładem chi-kwadrat

Na podstawie podanej próbki należy zweryfikować hipotezę, że cecha X ma rozkład chi-kwadrat.

1.0, 4.7, 5.2, 7.6, 2.9, 6.5, 4.3, 1.3, 1.6, 3.3, 0.5, 1.8, 15.4, 2.7, 9.6, 11.6, 23.2, 3.2, 3.4, 12.4, 19.5

Część (a): - Wykonać test Kołmogorowa-Smirnowa dla zgodności z rozkładem chi-kwadrat. - Porównać dystrybuantę empiryczną z teoretyczną (na wykresie).

Część (b): - Porównać kwantyle empiryczne i teoretyczne za pomocą wykresu Q-Q.

Hipotezy statystyczne: - **H (hipoteza zerowa):** próba pochodzi z rozkładu X^2. - **H (hipoteza alternatywna):** próba nie pochodzi z rozkładu X^2

[35]:
```
x <- c(1.0, 4.7, 5.2, 7.6, 2.9, 6.5, 4.3, 1.3, 1.6, 3.3,
       0.5, 1.8, 15.4, 2.7, 9.6, 11.6, 23.2, 3.2, 3.4, 12.4, 19.5)
```

[36]:
```
n <- length(x)
```

[37]:
```
n
```

21

1.2.1 Z2.a test Kołmogorowa-Smirnowa dla zgodności z rozkładem chi-kwadrat

[38]:
```
## est stopni swobody
df_hat <- mean(x)
```

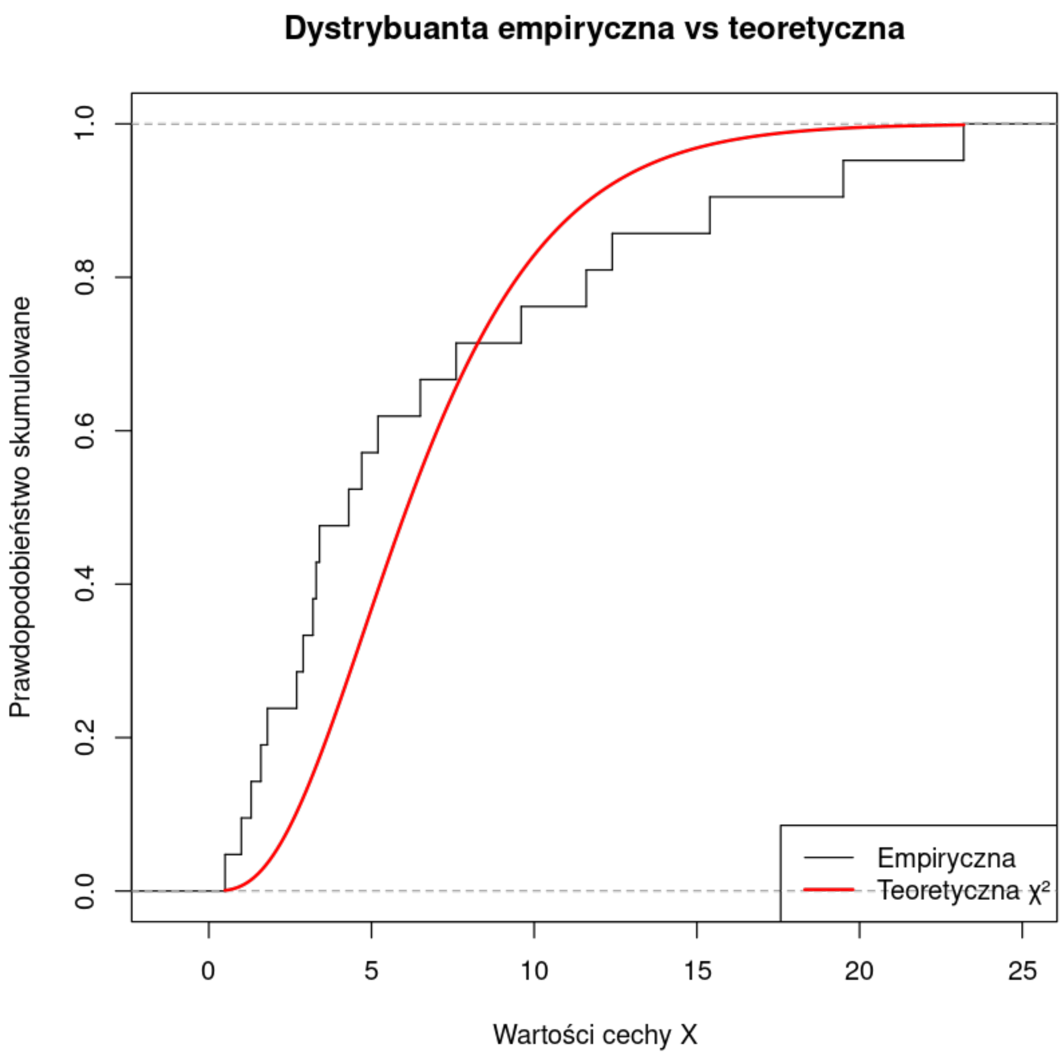
[39]:
```
df_hat
```

6.74761904761905

[40]:
```
ks.test(x, "pchisq", df = df_hat)
```

```

        Exact one-sample Kolmogorov-Smirnov test

data:  x
D = 0.30233, p-value = 0.03367
alternative hypothesis: two-sided
```

[41]:
```
dystr_empiryczna <- ecdf(x)

# Zakres wartości
x_wykres <- seq(min(x), max(x), length.out = 200)

# Rysowanie wykresu
plot(dystr_empiryczna, verticals = TRUE, do.points = FALSE,
     main = "Dystrybuanta empiryczna vs teoretyczna",
     xlab = "Wartości cechy X", ylab = "Prawdopodobieństwo skumulowane")

# Teoretyczna dystrybuanta chi-kwadrat
lines(x_wykres, pchisq(x_wykres, df = df_hat),
      col = "red", lwd = 2)

legend("bottomright", legend = c("Empiryczna", "Teoretyczna  ²"),
       col = c("black", "red"), lwd = c(1, 2))
```



[42]:
```
### Wnioski część (a)
Przy założeniu progu istotności p-value == 0.05.
Test Kołmogorowa-Smirnowa wykazał p-wartość 0.033, co oznacza, że istnieją
  ↪statystyczne podstawy do odrzucenia hipotezy zgodności z rozkładem
  ↪chi-kwadrat
```

1.2.2 Z2.b wykres kwantylowy (Q–Q plot)

[43]:
```
# asc sort (kwantyle empiryczne)
x_empiryczne <- sort(x)

# # Kwantyle teoretyczne (z rozkładu chi-kwadrat o df_hat)
kwantyle_teoretyczne <- qchisq(ppoints(n), df = df_hat)
```

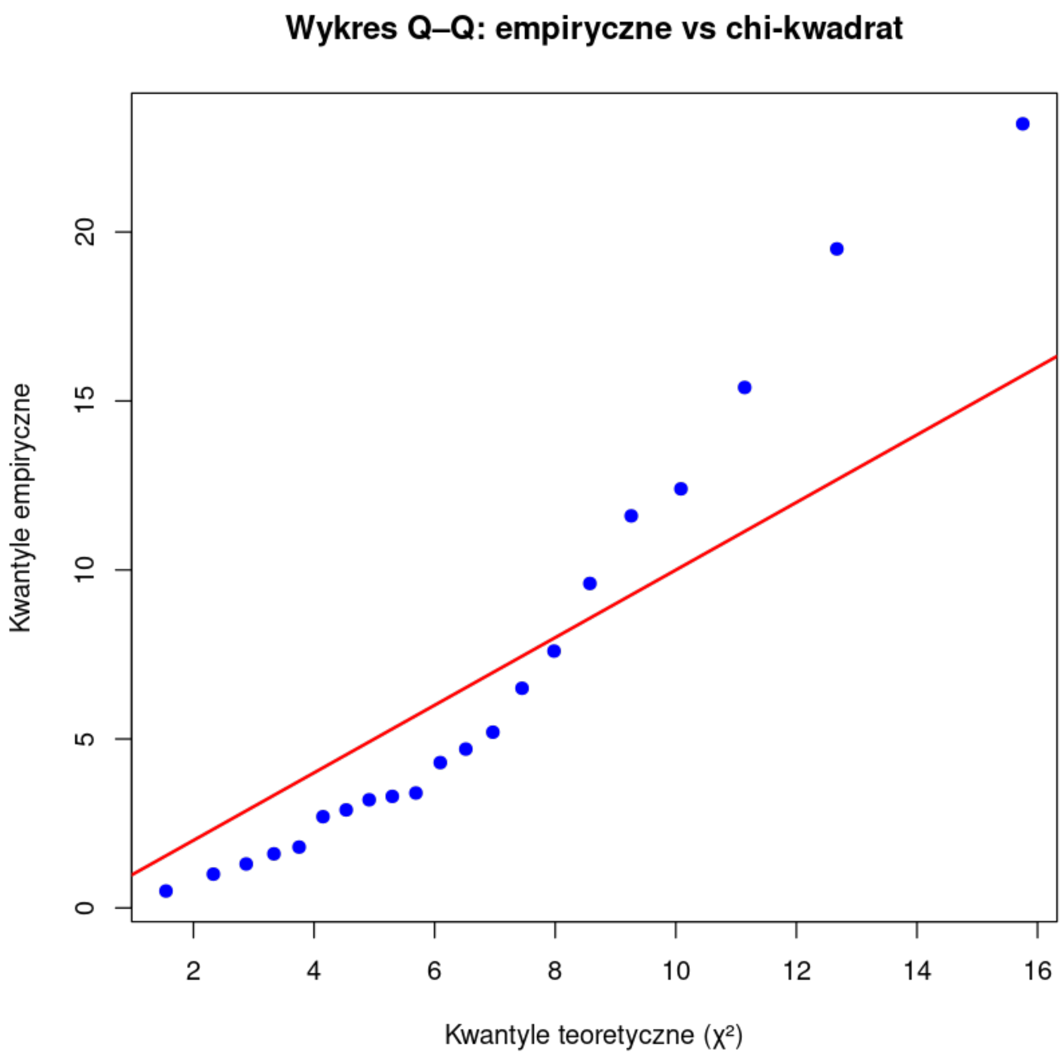
[45]:
```
x_empiryczne
```

1. 0.5 2. 1 3. 1.3 4. 1.6 5. 1.8 6. 2.7 7. 2.9 8. 3.2 9. 3.3 10. 3.4 11. 4.3 12. 4.7 13. 5.2 14. 6.5 15. 7.6 16. 9.6 17. 11.6 18. 12.4 19. 15.4 20. 19.5 21. 23.2

[44]:
```
kwantyle_teoretyczne
```

1. 1.54462427051942 2. 2.33096538125728 3. 2.8749947773806 4. 3.33476934436616 5. 3.75295516614992 6. 4.14883460199186 7. 4.53353548070311 8. 4.91466785143739 9. 5.29822204351978 10. 5.68954011714865 11. 6.09394293137503 12. 6.51725995266972 13. 6.96641529373264 14. 7.45022947387703 15. 7.98069095298224 16. 8.57521220448477 17. 9.26108836834429 18. 10.0855238579207 19. 11.1426621231644 20. 12.6717197778583 21. 15.755016482405

[46]:
```
# Wykres Q-Q
qqplot(kwantyle_teoretyczne, x_empiryczne,
       main = "Wykres Q-Q: empiryczne vs chi-kwadrat",
       xlab = "Kwantyle teoretyczne ( ²)", ylab = "Kwantyle empiryczne",
       pch = 19, col = "blue")

abline(0, 1, col = "red", lwd = 2)
```



[ ]:
```
### Z2.b Wnioski
Punkty znacząco odbiegają od linii idealnego dopasowania. Największe␣
↪rozbieżoności występują w górnych kwantylach
Wykres Q-Q wspiera wynik testu KS z punktu a.
```

1.3 Zadanie 3: Analiza wpływu nawozu na plony

1.3.1 Polecenie:

W pewnym doświadczeniu rolniczym bada się plony nowej odmiany pszenicy (w kwintalach na hektar) w zależności od rodzaju nawozu. Należy:

(a) Zweryfikować hipotezę H , że rozkłady plonów dla każdego typu nawozu są jednakowe, wykorzystując test Kruskala–Wallisa.

(b) Obliczyć średnią rangę dla każdej grupy.

Dane:

- n1 = c(35, 32, 33.5, 36, 38, 30, 32.5, 31, 34)
- n2 = c(28.5, 32, 33, 34, 28, 30.5, 30, 32)
- n3 = c(26.5, 29, 33, 31, 28, 25.5, 29, 32, 29.5, 32)
- n4 = c(30.5, 25.5, 32.5, 27, 34.5, 31)

1.3.2 Z3.a Kruskal-Wallis - rozkłady plonów dla każdego typu nawozu są jednakowe

[47]:
```
n1 <- c(35, 32, 33.5, 36, 38, 30, 32.5, 31, 34)
n2 <- c(28.5, 32, 33, 34, 28, 30.5, 30, 32)
n3 <- c(26.5, 29, 33, 31, 28, 25.5, 29, 32, 29.5, 32)
n4 <- c(30.5, 25.5, 32.5, 27, 34.5, 31)
```

[48]:
```
plony <- c(n1, n2, n3, n4)
```

[49]:
```
grupy <- factor(c(
  rep("n1", length(n1)),
  rep("n2", length(n2)),
  rep("n3", length(n3)),
  rep("n4", length(n4))
))
```

[50]:
```
grupy
```

1. n1 2. n1 3. n1 4. n1 5. n1 6. n1 7. n1 8. n1 9. n1 10. n2 11. n2 12. n2 13. n2 14. n2 15. n2 16. n2 17. n2 18. n3 19. n3 20. n3 21. n3 22. n3 23. n3 24. n3 25. n3 26. n3 27. n3 28. n4 29. n4 30. n4 31. n4 32. n4 33. n4

Levels: 1. 'n1' 2. 'n2' 3. 'n3' 4. 'n4'

[51]:
```
plony
```

1. 35 2. 32 3. 33.5 4. 36 5. 38 6. 30 7. 32.5 8. 31 9. 34 10. 28.5 11. 32 12. 33 13. 34 14. 28 15. 30.5 16. 30 17. 32 18. 26.5 19. 29 20. 33 21. 31 22. 28 23. 25.5 24. 29 25. 32 26. 29.5 27. 32 28. 30.5 29. 25.5 30. 32.5 31. 27 32. 34.5 33. 31

[52]:
```
test_kw <- kruskal.test(plony ~ grupy)
```

[53]:
```
test_kw
```

```
	Kruskal-Wallis rank sum test

data:  plony by grupy
Kruskal-Wallis chi-squared = 8.9766, df = 3, p-value = 0.0296
```

1.3.3 Z3.a Wnioski

Wyniki testu: - Statystyka testowa: $^2 = 8.9766$ - Stopnie swobody: df = 3 - p-wartość: 0.0296

Hipotezy: - **H :** Rozkłady plonów w grupach n1, n2, n3 i n4 są identyczne. - **H :** Co najmniej jedna grupa różni się pod względem rozkładu plonów.

Wniosek: Ponieważ p-wartość < 0.05, odrzucamy hipotezę zerową. Istnieją statystycznie istotne różnice w rozkładach plonów między co najmniej dwoma rodzajami nawozów

1.3.4 Z3.b średnia ranga dla każdej próbki

[55]:
```
# plony - wszystkie obserwacje
# grupy - wektor etykiet grupowych

rangi <- rank(plony)
```

[56]:
```
df_rangi <- data.frame(
  grupa = grupy,
  ranga = rangi
)
```

[57]:
```
df_rangi
```

A data.frame: 33 × 2

grupa	ranga
<fct>	<dbl>
n1	31.0
n1	20.0
n1	27.0
n1	32.0
n1	33.0
n1	11.5
n1	23.5
n1	16.0
n1	28.5
n2	7.0
n2	20.0
n2	25.5
n2	28.5
n2	5.5
n2	13.5
n2	11.5
n2	20.0
n3	3.0
n3	8.5
n3	25.5
n3	16.0
n3	5.5
n3	1.5
n3	8.5
n3	20.0
n3	10.0
n3	20.0
n4	13.5
n4	1.5
n4	23.5
n4	4.0
n4	30.0
n4	16.0

[58]:
```
srednie_rangi <- aggregate(ranga ~ grupa, data = df_rangi, FUN = mean)
```

[59]:
```
srednie_rangi
```

A data.frame: 4 × 2

grupa	ranga
<fct>	<dbl>
n1	24.72222
n2	16.43750
n3	11.85000
n4	14.75000

1.3.5 Z3.b Wnioski Średnie rangi dla każdej grupy nawozu

Najwyższą średnią rangę uzyskała grupa **n1**, co oznacza, że ta grupa miała generalnie **wyższe plony** niż pozostałe. Najniższą rangę uzyskała grupa **n3**, co sugeruje, że dawała najniższe plony.

Co potwierdza wynik testu Kruskala-Wallisa oraz jego interpretację z części Z3a Wynik

Wilcoxon - które grupy się istotnie różniły?

1.4 Zadanie 4 : Charakter losowości i niezależność cyfr

1.4.1 Polecenie:

(a) Zbadać, czy poniższa próbka ma charakter losowy.

(b) Niech X będzie pierwszą, a Y drugą cyfrą w rozważanych liczbach. Zbadać, czy X i Y są statystycznie niezależne.

Dane (próbka losowa):

35, 60, 148, 75, 92, 243, 37, 48, 95, 740, 154, 292, 334, 421, 15, 87, 36, 302, 250, 82, 101, 336, 230, 672, 55, 65, 17, 102, 21, 304, 640, 25, 354, 85, 340, 395, 720, 407, 230, 84, 14, 26, 35, 458, 370, 483, 310, 75, 300, 435, 92, 180, 405, 66, 315, 40, 532, 326, 604, 157, 640, 45, 31, 258, 625, 152, 193, 32, 488, 166, 10, 307, 260, 85, 450, 62, 345, 71, 165, 251, 236, 354, 58, 320, 81, 71, 45, 310, 345, 127, 476, 420, 150, 23, 48, 60, 95, 470, 92, 67, 325, 45, 157, 385, 125, 357, 582, 393, 175, 86, 830, 650, 40

1.4.2 Z4.a Zbadać czy próbka ma charakter losowy?

Benford? histogram?

[61]:
```r
x <- c(35, 60, 148, 75, 92, 243, 37, 48, 95, 740, 154, 292, 334, 421, 15,
       87, 36, 302, 250, 82, 101, 336, 230, 672, 55, 65, 17, 102, 21,
       304, 640, 25, 354, 85, 340, 395, 720, 407, 230, 84, 14, 26, 35,
       458, 370, 483, 310, 75, 300, 435, 92, 180, 405, 66, 315, 40, 532,
       326, 604, 157, 640, 45, 31, 258, 625, 152, 193, 32, 488, 166, 10,
       307, 260, 85, 450, 62, 345, 71, 165, 251, 236, 354, 58, 320, 81,
       71, 45, 310, 345, 127, 476, 420, 150, 23, 48, 60, 95, 470, 92,
       67, 325, 45, 157, 385, 125, 357, 582, 393, 175, 86, 830, 650, 40)
```

[62]:
```r
# Benford
# Pierwsza cyfra
pierwsze_cyfry <- as.numeric(substring(as.character(x), 1, 1))
```

[63]:
```r
pierwsze_cyfry
```

1. 3 2. 6 3. 1 4. 7 5. 9 6. 2 7. 3 8. 4 9. 9 10. 7 11. 1 12. 2 13. 3 14. 4 15. 1 16. 8 17. 3 18. 3 19. 2 20. 8 21. 1 22. 3 23. 2 24. 6 25. 5 26. 6 27. 1 28. 1 29. 2 30. 3 31. 6 32. 2 33. 3 34. 8 35. 3 36. 3 37. 7 38. 4 39. 2 40. 8 41. 1 42. 2 43. 3 44. 4 45. 3 46. 4 47. 3 48. 7 49. 3 50. 4 51. 9 52. 1 53. 4 54. 6 55. 3 56. 4 57. 5 58. 3 59. 6 60. 1 61. 6 62. 4 63. 3 64. 2 65. 6 66. 1 67. 1 68. 3 69. 4 70. 1 71. 1 72. 3 73. 2

74. 8 75. 4 76. 6 77. 3 78. 7 79. 1 80. 2 81. 2 82. 3 83. 5 84. 3 85. 8 86. 7 87. 4 88. 3 89. 3 90. 1 91. 4 92. 4 93. 1 94. 2 95. 4 96. 6 97. 9 98. 4 99. 9 100. 6 101. 3 102. 4 103. 1 104. 3 105. 1 106. 3 107. 5 108. 3 109. 1 110. 8 111. 8 112. 6 113. 4

[64]:
```
obs <- table(factor(pierwsze_cyfry, levels = 1:9))
```

[66]:
```
obs
```

```
 1  2  3  4  5  6  7  8  9 
19 13 28 18  4 12  6  8  5 
```

[67]:
```
# Warunki
# teoretyczne wystąpienie pierwszych cyfr
benford_probs <- log10(1 + 1 / (1:9))
```

[68]:
```
benford_probs
```

1. 0.301029995663981 2. 0.176091259055681 3. 0.1249387366083 4. 0.0969100130080564 5. 0.0791812460476248 6. 0.0669467896306132 7. 0.0579919469776867 8. 0.0511525224473813 9. 0.0457574905606751

[69]:
```
# liczności
exp <- benford_probs * length(x)
```

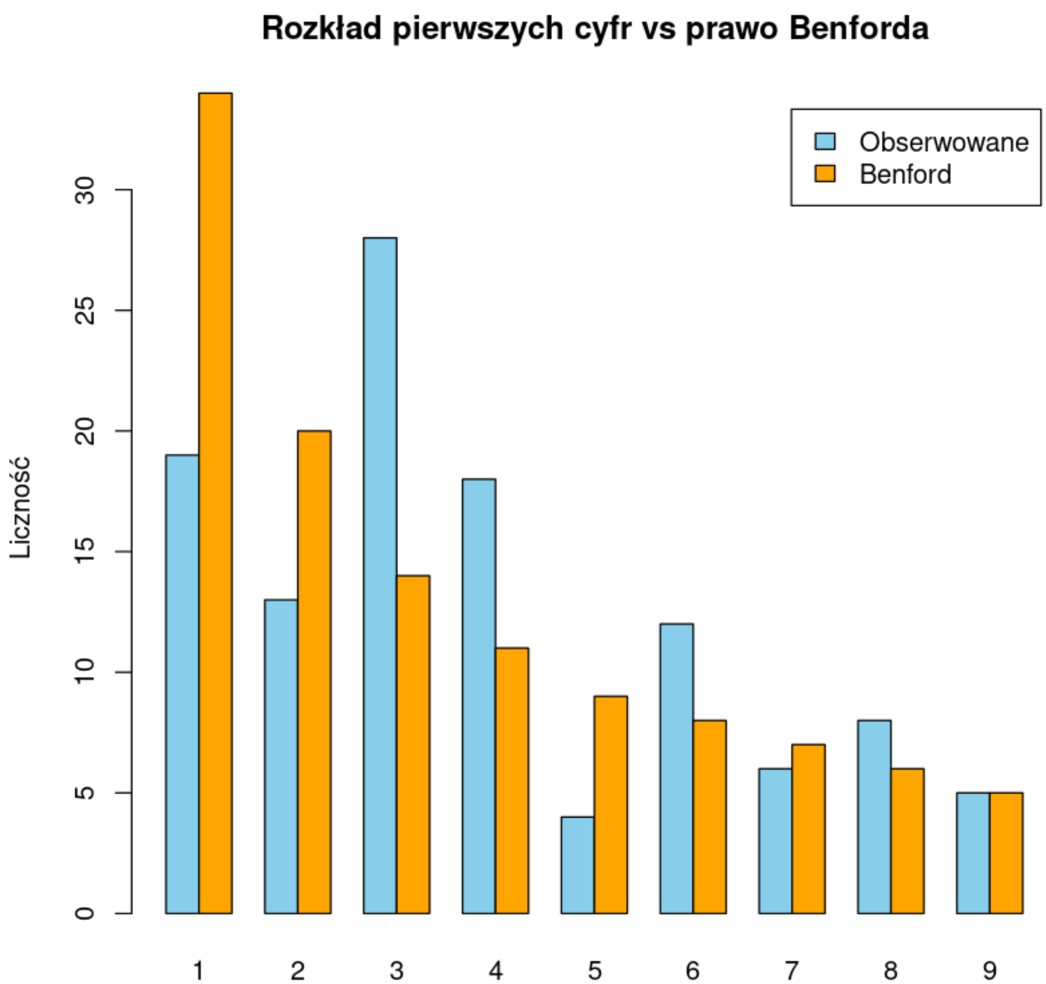
[70]:
```
exp
```

1. 34.0163895100299 2. 19.898312273292 3. 14.1180772367379 4. 10.9508314699104 5. 8.9474808033816 6. 7.56498722825929 7. 6.5530900084786 8. 5.78023503655409 9. 5.17059643335629

[71]:
```
chisq.test(x = obs, p = benford_probs, rescale.p = TRUE)
```

```
	Chi-squared test for given probabilities

data:  obs
X-squared = 33.448, df = 8, p-value = 5.112e-05
```

[72]:
```
barplot(rbind(obs, round(exp)),
        beside = TRUE, col = c("skyblue", "orange"),
        names.arg = 1:9, legend = c("Obserwowane", "Benford"),
        main = "Rozkład pierwszych cyfr vs prawo Benforda",
        ylab = "Liczność")
```



[77]:
```
install.packages("e1071")
```

```
Installing package into '/home/kotmin/R/x86_64-pc-linux-gnu-library/4.5'
(as 'lib' is unspecified)

also installing the dependency 'proxy'
```

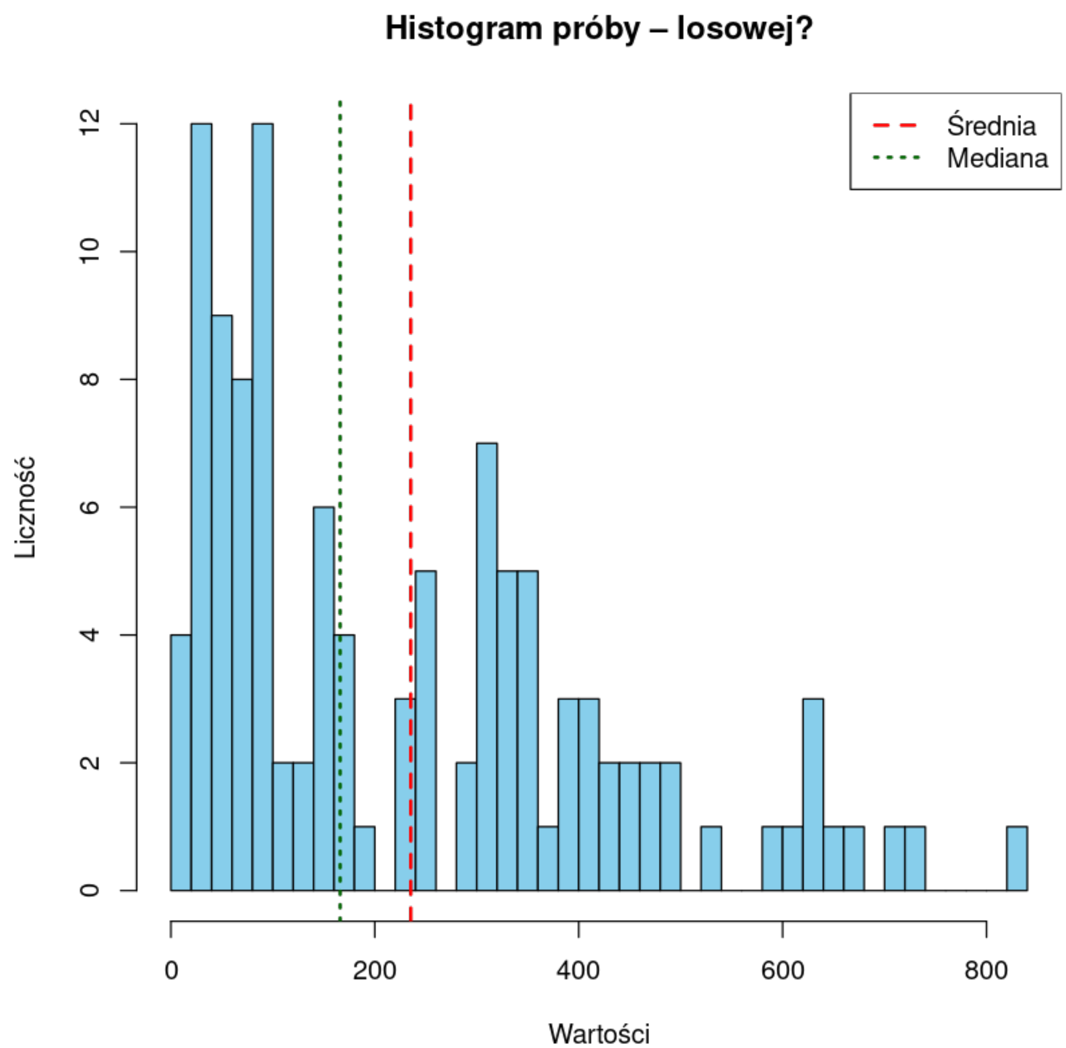
[78]:
```
# POM

srednia <- mean(x)
mediana <- median(x)
```

```
wariancja <- var(x)
odchylenie <- sd(x)


library(e1071)
skosnosc <- skewness(x)


pom <- mean(x, trim = 0.1)


data.frame(
  Średnia = round(srednia, 2),
  Mediana = round(mediana, 2),
  POM_10proc = round(pom, 2),
  Odchylenie = round(odchylenie, 2),
  Skośność = round(skosnosc, 2)
)
```

A data.frame: 1 × 5

Średnia	Mediana	POM_10proc	Odchylenie	Skośność
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
235.27	166	209.91	197.8	0.88

[80]:
```
## dodatkowo możemy zrobić histogram

hist(x,
     breaks = 30,
     col = "skyblue",
     main = "Histogram próby - losowej?",
     xlab = "Wartości",
     ylab = "Liczność")

# Dodajemy linię średniej
abline(v = mean(x), col = "red", lwd = 2, lty = 2)

# Dodajemy linię mediany
abline(v = median(x), col = "darkgreen", lwd = 2, lty = 3)

# Legenda
legend("topright",
       legend = c("Średnia", "Mediana"),
       col = c("red", "darkgreen"),
       lwd = 2,
       lty = c(2, 3))
```



1.4.3 4a Wnioski

Zostały sprawdzone warunki czy można wykonać testy zgodności z prawem Benforda.

Rozkład jest skośny dodatnio(prawostronnie) średnia > mediana.

Wysoka wartość współczynnika skośności i spora różnica między średnią, a medianą sugerują, że dane są **silnie niesymetryczne** i mogą pochodzić z próbki zdominowanej przez duże wartości.

Wniosek z testu zgodności z prawem Benforda: Test chi-kwadrat dał wynik: data: obs X-squared = 33.448, df = 8, p-value = 5.112e-05

Odrzucamy hipotezę zgodności z rozkładem Benforda – dane **nie mają losowego charakteru**

Histogram, linie statystyk, klasyczne statystyki opisowe, obserwowana dodatnia skośność oraz test Benforda jednoznacznie wskazują, że dane **nie są naturalnie rozłożone ani całkiem losowe**.

1.4.4 Z4.b XY statystycznie niezależne

[120]:
```
# polecenie mówiło o cyfrach - chcemy mieć możliwość uzyskania X oraz Y
x_filtr <- x[x >= 10]
```

[121]:
```
x_filtr
```

1. 35 2. 60 3. 148 4. 75 5. 92 6. 243 7. 37 8. 48 9. 95 10. 740 11. 154 12. 292 13. 334 14. 421 15. 15 16. 87 17. 36 18. 302 19. 250 20. 82 21. 101 22. 336 23. 230 24. 672 25. 55 26. 65 27. 17 28. 102 29. 21 30. 304 31. 640 32. 25 33. 354 34. 85 35. 340 36. 395 37. 720 38. 407 39. 230 40. 84 41. 14 42. 26 43. 35 44. 458 45. 370 46. 483 47. 310 48. 75 49. 300 50. 435 51. 92 52. 180 53. 405 54. 66 55. 315 56. 40 57. 532 58. 326 59. 604 60. 157 61. 640 62. 45 63. 31 64. 258 65. 625 66. 152 67. 193 68. 32 69. 488 70. 166 71. 10 72. 307 73. 260 74. 85 75. 450 76. 62 77. 345 78. 71 79. 165 80. 251 81. 236 82. 354 83. 58 84. 320 85. 81 86. 71 87. 45 88. 310 89. 345 90. 127 91. 476 92. 420 93. 150 94. 23 95. 48 96. 60 97. 95 98. 470 99. 92 100. 67 101. 325 102. 45 103. 157 104. 385 105. 125 106. 357 107. 582 108. 393 109. 175 110. 86 111. 830 112. 650 113. 40

[122]:
```
length(x_filtr)
```

113

[123]:
```
cyfra_X <- as.numeric(substr(as.character(x_filtr), 1, 1))
cyfra_Y <- as.numeric(substr(as.character(x_filtr), 2, 2))
```

[124]:
```
cyfra_X
```

1. 3 2. 6 3. 1 4. 7 5. 9 6. 2 7. 3 8. 4 9. 9 10. 7 11. 1 12. 2 13. 3 14. 4 15. 1 16. 8 17. 3 18. 3 19. 2 20. 8 21. 1 22. 3 23. 2 24. 6 25. 5 26. 6 27. 1 28. 1 29. 2 30. 3 31. 6 32. 2 33. 3 34. 8 35. 3 36. 3 37. 7 38. 4 39. 2 40. 8 41. 1 42. 2 43. 3 44. 4 45. 3 46. 4 47. 3 48. 7 49. 3 50. 4 51. 9 52. 1 53. 4 54. 6 55. 3 56. 4 57. 5 58. 3 59. 6 60. 1 61. 6 62. 4 63. 3 64. 2 65. 6 66. 1 67. 1 68. 3 69. 4 70. 1 71. 1 72. 3 73. 2 74. 8 75. 4 76. 6 77. 3 78. 7 79. 1 80. 2 81. 2 82. 3 83. 5 84. 3 85. 8 86. 7 87. 4 88. 3 89. 3 90. 1 91. 4 92. 4 93. 1 94. 2 95. 4 96. 6 97. 9 98. 4 99. 9 100. 6 101. 3 102. 4 103. 1 104. 3 105. 1 106. 3 107. 5 108. 3 109. 1 110. 8 111. 8 112. 6 113. 4

[125]:
```
cyfra_Y
```

1. 5 2. 0 3. 4 4. 5 5. 2 6. 4 7. 7 8. 8 9. 5 10. 4 11. 5 12. 9 13. 3 14. 2 15. 5 16. 7 17. 6 18. 0 19. 5 20. 2 21. 0 22. 3 23. 3 24. 7 25. 5 26. 5 27. 7 28. 0 29. 1 30. 0 31. 4 32. 5 33. 5 34. 5 35. 4 36. 9 37. 2 38. 0 39. 3 40. 4 41. 4 42. 6 43. 5 44. 5 45. 7 46. 8 47. 1 48. 5 49. 0 50. 3 51. 2 52. 8 53. 0 54. 6 55. 1 56. 0 57. 3 58. 2 59. 0 60. 5 61. 4 62. 5 63. 1 64. 5 65. 2 66. 5 67. 9 68. 2 69. 8 70. 6 71. 0 72. 0 73. 6 74. 5 75. 5 76. 2 77. 4 78. 1 79. 6 80. 5 81. 3 82. 5 83. 8 84. 2 85. 1 86. 1 87. 5 88. 1 89. 4 90. 2 91. 7 92. 2 93. 5 94. 3 95. 8 96. 0 97. 5 98. 7 99. 2 100. 7 101. 2 102. 5 103. 5 104. 8 105. 2 106. 5 107. 8 108. 9 109. 7 110. 6 111. 3 112. 5 113. 0

[126]:
```
# tablica liczności / kontyngencji
tablica_xy <- table(X = cyfra_X, Y = cyfra_Y)
```

[127]:
```
tablica_xy
```

```
   Y
X    0 1 2 3 4 5 6 7 8 9
  1  3 0 2 0 2 6 2 2 1 1
  2  0 1 0 4 1 4 2 0 0 1
  3  4 4 4 2 3 5 1 2 1 2
  4  4 0 2 1 0 5 0 2 4 0
  5  0 0 0 1 0 1 0 0 2 0
  6  3 0 2 0 2 2 1 2 0 0
  7  0 2 1 0 1 2 0 0 0 0
  8  0 1 1 1 1 2 1 1 0 0
  9  0 0 3 0 0 2 0 0 0 0
```

[128]:
```
## test wstpępny
test_chi <- chisq.test(tablica_xy)
```

```
Warning message in chisq.test(tablica_xy):
"Chi-squared approximation may be incorrect"
```

[129]:
```
test_chi
```

```
	Pearson's Chi-squared test

data:  tablica_xy
X-squared = 83.349, df = 72, p-value = 0.1698
```

[130]:
```
oczekiwane <- test_chi$expected
```

[131]:
```
oczekiwane
```

A matrix: 9 × 10 of type dbl

	0	1	2	3	4	5	6
1	2.3539823	1.3451327	2.5221239	1.5132743	1.6814159	4.876106	1.17699
2	1.6106195	0.9203540	1.7256637	1.0353982	1.1504425	3.336283	0.80530
3	3.4690265	1.9823009	3.7168142	2.2300885	2.4778761	7.185841	1.73451
4	2.2300885	1.2743363	2.3893805	1.4336283	1.5929204	4.619469	1.11504
5	0.4955752	0.2831858	0.5309735	0.3185841	0.3539823	1.026549	0.24778
6	1.4867257	0.8495575	1.5929204	0.9557522	1.0619469	3.079646	0.74336
7	0.7433628	0.4247788	0.7964602	0.4778761	0.5309735	1.539823	0.37168
8	0.9911504	0.5663717	1.0619469	0.6371681	0.7079646	2.053097	0.49557
9	0.6194690	0.3539823	0.6637168	0.3982301	0.4424779	1.283186	0.30973

[132]:
```
str(test_chi)
```

```
List of 9
 $ statistic: Named num 83.3
  ..- attr(*, "names")= chr "X-squared"
 $ parameter: Named int 72
```

```
  ..- attr(*, "names")= chr "df"
 $ p.value  : num 0.17
 $ method   : chr "Pearson's Chi-squared test"
 $ data.name: chr "tablica_xy"
 $ observed : 'table' int [1:9, 1:10] 3 0 4 4 0 3 0 0 0 0 …
  ..- attr(*, "dimnames")=List of 2
  .. ..$ X: chr [1:9] "1" "2" "3" "4" …
  .. ..$ Y: chr [1:10] "0" "1" "2" "3" …
 $ expected : num [1:9, 1:10] 2.354 1.611 3.469 2.23 0.496 …
  ..- attr(*, "dimnames")=List of 2
  .. ..$ X: chr [1:9] "1" "2" "3" "4" …
  .. ..$ Y: chr [1:10] "0" "1" "2" "3" …
 $ residuals: 'table' num [1:9, 1:10] 0.421 -1.269 0.285 1.185 -0.704 …
  ..- attr(*, "dimnames")=List of 2
  .. ..$ X: chr [1:9] "1" "2" "3" "4" …
  .. ..$ Y: chr [1:10] "0" "1" "2" "3" …
 $ stdres   : 'table' num [1:9, 1:10] 0.493 -1.441 0.351 1.381 -0.766 …
  ..- attr(*, "dimnames")=List of 2
  .. ..$ X: chr [1:9] "1" "2" "3" "4" …
  .. ..$ Y: chr [1:10] "0" "1" "2" "3" …
 - attr(*, "class")= chr "htest"
```

[133]:
```
sum(oczekiwane < 1)
```

48

[134]:
```
mean(oczekiwane < 5) * 100
```

98.8888888888889

[135]:
```
test_chi$stdres
```

```
   Y
X             0           1           2           3           4           5
  1  0.49321893 -1.31917508 -0.38707207 -1.40590728  0.28215151  0.64723735
  2 -1.44130973  0.09155254 -1.49948971  3.22830403 -0.15617009  0.44801254
  3  0.35117240  1.71413865  0.18186204 -0.18517663  0.40057528 -1.09045828
  4  1.38098686 -1.27721502 -0.29500894 -0.41171750 -1.44176693  0.22396027
  5 -0.76577712 -0.56209059 -0.79668852  1.28129396 -0.63450837 -0.03094415
  6  1.40249851 -1.01139197  0.36634231 -1.07788826  1.00849829 -0.75476039
  7 -0.94660627  2.57663005  0.25167546 -0.74050423  0.69283364  0.44201439
  8 -1.10340764  0.62009184 -0.06696362  0.49152489  0.37713332 -0.04458740
  9 -0.86011945 -0.63133910  3.14983353 -0.67284795 -0.71267863  0.75073855
   Y
X             6           7           8           9
  1  0.85877519  0.45219240 -0.33847255  0.44571472
  2  1.46116534 -1.12749723 -1.05794050  0.86129938
  3 -0.66393702 -0.18517663 -0.83451487  1.18963121
  4 -1.18907640  0.53775346  2.73182101 -0.88640137
```

```
5 -0.52330159 -0.59904653  3.40767418 -0.39009709
6  0.32507529  1.17769273 -1.01139197 -0.70191723
7 -0.64687303 -0.74050423 -0.69482159 -0.48221388
8  0.76748957  0.49152489 -0.80991587 -0.56209059
9 -0.58777138 -0.67284795 -0.63133910 -0.43815633
```

[136]:
```
# da się tu znaleźć wartości większe od 2
```

[137]:
```
# w każdytm razie liczności w klasach są mniejsze od 5, test może stracić na skuteczności. Test wykazał p-value < 0.05
# spróbujemy wykorzystać test Fishera

fisher.test(tablica_xy,simulate.p.value=TRUE)
```

```
        Fisher's Exact Test for Count Data with simulated p-value (based on
        2000 replicates)

data:  tablica_xy
p-value = 0.3628
alternative hypothesis: two.sided
```

[138]:
```
# można spróbować symulacji Monte Carlo
chisq.test(tablica_xy, simulate.p.value = TRUE, B = 10000)
```

```
        Pearson's Chi-squared test with simulated p-value (based on 10000
        replicates)

data:  tablica_xy
X-squared = 83.349, df = NA, p-value = 0.1646
```

1.4.5 4.b Wnioski

Elementy podzielono zgodnie z poleceniem, stworzono tablice kontyngencji.

Sprawdzono warunki zastosowania klasycznego testu chi-kwadrat: - **48 komórek** (spośród 90) miało oczekiwaną liczność **mniejszą niż 1**

- **98.9% wszystkich komórek** miało oczekiwaną liczność **mniejszą niż 5**

Wynik testu chi-kwadrat mógł zostać uznany za niewiarygodny. Sugerował **statystyczną niezależność**

Hipotezy:

- **H (hipoteza zerowa):** Cyfry X i Y są niezależne — rozkład drugiej cyfry nie zależy od pierwszej.

- **H (hipoteza alternatywna):** Cyfry X i Y są zależne — rozkład drugiej cyfry zależy od pierwszej.

Wykonano dwa alternatywne testy nieparametryczne: 1. fisher (2000 permutacji) p ~0.36

2. chi-sqrt z symulacją Monte Carlo, p~0.16

W obu przypadkach p-wartość jest większa niż 0.05, **brak podstaw do odrzucenia hipotezy zerowej**.

Cyfry X i Y mogą być uznane za **statystycznie niezależne** na podstawie dostępnych danych.

1.5 Zadanie 5: Piramida demograficzna USA 1948

1.5.1 Polecenie zad 5

W pakiecie latticeExtra znajduje się ramka danych USAge.df zawierająca wielkość populacji USA w latach 1900-1979 z podziałem na wiek i płeć.

Sporządzić wykres piramidowy/demograficzny przedstawiający strukturę wieku z podziałem na płeć dla roku 1948. Przyjąć następujący podział na kategorie wiekowe

---

wiek=c("0-5","6-11","12-17","18-23","24-29","30-35","36-41", "42-47","48-53","54-59","60-65","66-71","72 i więcej")

Uwaga. Wykorzystać funkcję **pyramid.plot** z pakietu **plotrix** lub też funkcje dostępne w innych pakietach.

[142]:
```
if (!require(plotrix)) install.packages("plotrix")
```

```
Loading required package: plotrix
```

Looks like it needs external dependencies to run latticeExtra

' ' 'bash sudo apt-get update sudo apt-get install -y
libjpeg-dev
libpng-dev
libeigen3-dev
libproj-dev
libgeos-dev
libgdal-dev
libtiff5-dev
libglpk-dev
libx11-dev
libxt-dev
' ' '

```
sudo apt update
sudo apt install -y build-essential \
```

```
    libjpeg-dev libpng-dev libtiff5-dev \
    libeigen3-dev libgdal-dev libgeos-dev \
    libproj-dev libglpk-dev \
    libxt-dev libx11-dev \
    libcurl4-openssl-dev libssl-dev \
    libxml2-dev

sudo apt update
sudo apt install -y build-essential \
  libeigen3-dev libgsl-dev libblas-dev liblapack-dev \
  libjpeg-dev libpng-dev libtiff5-dev libcurl4-openssl-dev libssl-dev libxml2-dev
```

Jak się okazuje brakującym elementem był kompilator fortrana

```
sudo apt install gfortran
```

[148]:
```
install.packages("latticeExtra", dependencies = TRUE)
```

```
Installing package into '/home/kotmin/R/x86_64-pc-linux-gnu-library/4.5'
(as 'lib' is unspecified)

also installing the dependencies 'RcppEigen', 'SparseM', 'MatrixModels',
'interp', 'maps', 'mapproj', 'deldir', 'quantreg', 'zoo'


Warning message in install.packages("latticeExtra", dependencies = TRUE):
"installation of package 'RcppEigen' had non-zero exit status"
Warning message in install.packages("latticeExtra", dependencies = TRUE):
"installation of package 'SparseM' had non-zero exit status"
Warning message in install.packages("latticeExtra", dependencies = TRUE):
"installation of package 'deldir' had non-zero exit status"
Warning message in install.packages("latticeExtra", dependencies = TRUE):
"installation of package 'interp' had non-zero exit status"
Warning message in install.packages("latticeExtra", dependencies = TRUE):
"installation of package 'quantreg' had non-zero exit status"
Warning message in install.packages("latticeExtra", dependencies = TRUE):
"installation of package 'latticeExtra' had non-zero exit status"
```

[146]:
```
if (!require(latticeExtra)) install.packages("latticeExtra", dependencies =
 TRUE)
```

```
Loading required package: latticeExtra

Warning message in library(package, lib.loc = lib.loc, character.only = TRUE,
logical.return = TRUE, :
"there is no package called 'latticeExtra'"
Installing package into '/home/kotmin/R/x86_64-pc-linux-gnu-library/4.5'
(as 'lib' is unspecified)
```

```
also installing the dependencies ‘deldir’, ‘RcppEigen’, ‘png’, ‘jpeg’, ‘interp’


Warning message in install.packages("latticeExtra"):
"installation of package ‘deldir’ had non-zero exit status"
Warning message in install.packages("latticeExtra"):
"installation of package ‘RcppEigen’ had non-zero exit status"
Warning message in install.packages("latticeExtra"):
"installation of package ‘interp’ had non-zero exit status"
Warning message in install.packages("latticeExtra"):
"installation of package ‘latticeExtra’ had non-zero exit status"
```

[150]:
```
install.packages(c('deldir', 'RcppEigen'))
```

```
Installing packages into ‘/home/kotmin/R/x86_64-pc-linux-gnu-library/4.5’
(as ‘lib’ is unspecified)

Warning message in install.packages(c("deldir", "RcppEigen")):
"installation of package ‘deldir’ had non-zero exit status"
Warning message in install.packages(c("deldir", "RcppEigen")):
"installation of package ‘RcppEigen’ had non-zero exit status"
```

[151]:
```
install.packages('SparseM')
```

```
Installing package into ‘/home/kotmin/R/x86_64-pc-linux-gnu-library/4.5’
(as ‘lib’ is unspecified)

Warning message in install.packages("SparseM"):
"installation of package ‘SparseM’ had non-zero exit status"
```

[144]:
```
library(plotrix)
```

[152]:
```
library(latticeExtra)
```

```
Loading required package: lattice


Attaching package: ‘latticeExtra’


The following object is masked from ‘package:ggplot2’:

    layer
```

[153]:
```
# wczytanie danych
data("USAge.df")
```

[154]:
```
dane_1948 <- subset(USAge.df, Year == 1948)
```

[155]:
```
dane_1948
```

A data.frame: 150 × 4

	Age	Sex	Year	Population
	<dbl>	<fct>	<dbl>	<dbl>
7201	0	Male	1948	1.622336
7202	1	Male	1948	1.795136
7203	2	Male	1948	1.359864
7204	3	Male	1948	1.401297
7205	4	Male	1948	1.424228
7206	5	Male	1948	1.469383
7207	6	Male	1948	1.291849
7208	7	Male	1948	1.206605
7209	8	Male	1948	1.144003
7210	9	Male	1948	1.131221
7211	10	Male	1948	1.151048
7212	11	Male	1948	1.094383
7213	12	Male	1948	1.098160
7214	13	Male	1948	1.091974
7215	14	Male	1948	1.065727
7216	15	Male	1948	1.073340
7217	16	Male	1948	1.110025
7218	17	Male	1948	1.137613
7219	18	Male	1948	1.130167
7220	19	Male	1948	1.124056
7221	20	Male	1948	1.136351
7222	21	Male	1948	1.146391
7223	22	Male	1948	1.165079
7224	23	Male	1948	1.183062
7225	24	Male	1948	1.206138
7226	25	Male	1948	1.226459
7227	26	Male	1948	1.223964
7228	27	Male	1948	1.203563
7229	28	Male	1948	1.184262
7230	29	Male	1948	1.159978
7321	45	Female	1948	0.929498
7322	46	Female	1948	0.912446
7323	47	Female	1948	0.899328
7324	48	Female	1948	0.886839
7325	49	Female	1948	0.869327
7326	50	Female	1948	0.849200
7327	51	Female	1948	0.830046
7328	52	Female	1948	0.807287
7329	53	Female	1948	0.786310
7330	54	Female	1948	0.762945
7331	55	Female	1948	0.739414
7332	56	Female	1948	0.717533
7333	57	Female	1948	0.694190
7334	58	Female	1948	0.670733
7335	59	Female	1948	0.645356
7336	60	Female	1948	0.618581
7337	61	Female	1948	0.599091
7338	62	Female	1948	0.582391
7339	63	Female	1948	0.565650
7340	64	Female	1948	0.550148

[156]:
```
przedzialy <- cut(
  dane_1948$Age,
  breaks = c(0, 6, 12, 18, 24, 30, 36, 42, 48, 54, 60, 66, 72, Inf),
  labels = c("0-5", "6-11", "12-17", "18-23", "24-29", "30-35", "36-41",
             "42-47", "48-53", "54-59", "60-65", "66-71", "72 i więcej"),
  right = FALSE
)
```

[157]:
```
dane_1948$wiek_kategoria <- przedzialy
```

[158]:
```
# Agregujemy populację według płci i kategorii wiekowej
tabela <- aggregate(Population ~ Sex + wiek_kategoria, data = dane_1948, sum)
```

[159]:
```
kobiety <- tabela$Population[tabela$Sex == "Female"]
mezczyzni <- tabela$Population[tabela$Sex == "Male"]
```

[160]:
```
kobiety
```

1. 8.732111 2. 6.787291 3. 6.377272 4. 6.98532 5. 7.389165 6. 6.900891 7. 6.455141 8. 5.659599 9. 5.029009 10. 4.230171 11. 3.452066 12. 2.634169 13. 0.932342

[161]:
```
# Upewniamy się, że obie grupy są w tej samej kolejności wiekowej
kategorie <- levels(tabela$wiek_kategoria)
```

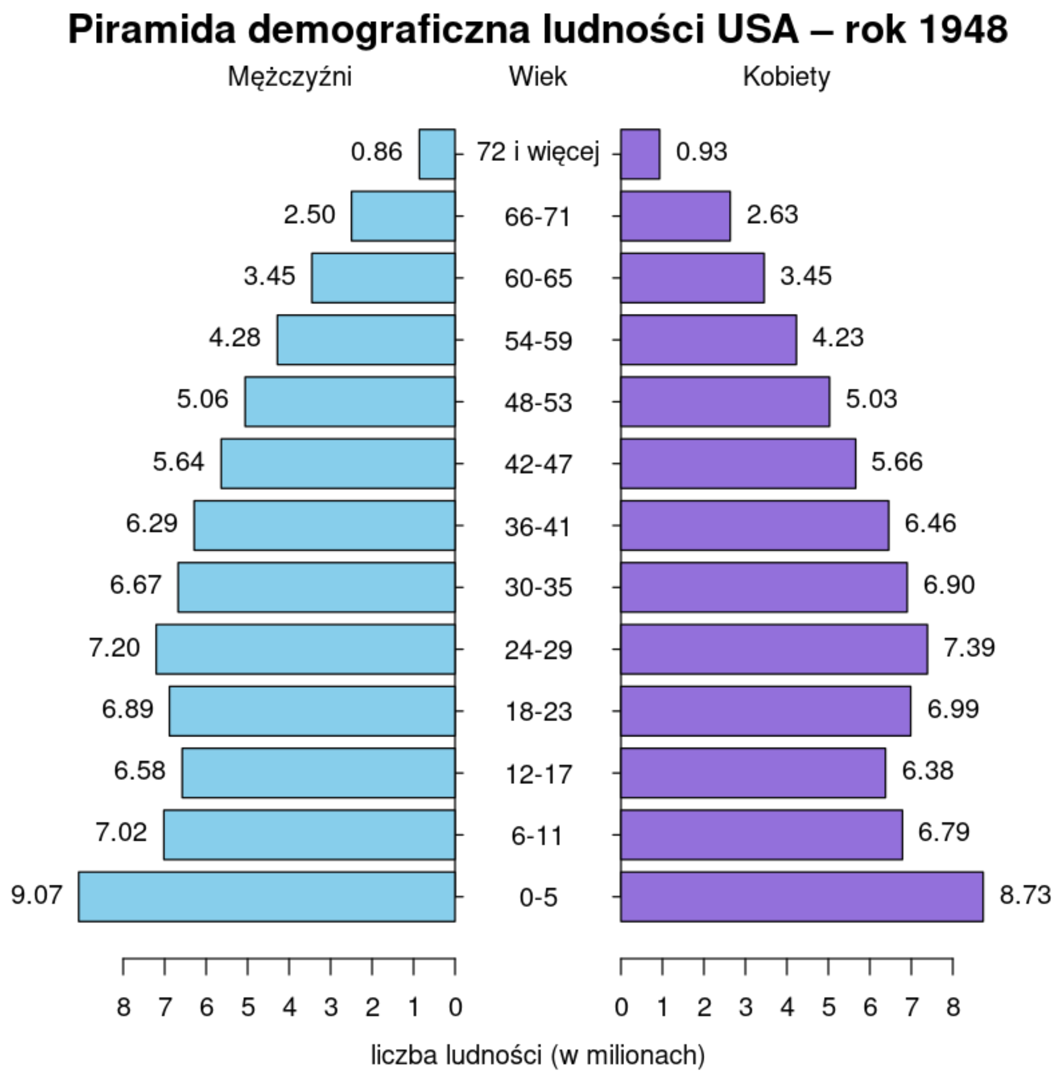
[163]:
```
# jeśli występuje N/A zastępujemy to 0
kobiety[is.na(kobiety)] <- 0
mezczyzni[is.na(mezczyzni)] <- 0
```

[177]:
```
pyramid.plot(
  lx = mezczyzni,
  rx = kobiety,
  labels = kategorie,
  main = "",  #
  lxcol = "#87CEEB",     # błękitny
  rxcol = "#9370DB",     # fioletowy
  unit = "liczba ludności (w milionach)",
  gap = 2,
  top.labels = c("Mężczyźni", "Wiek", "Kobiety"),
  labelcex = 1.0,
  ndig = 2,
  # xlim = maks  # przeskalowanie zakresu do danych - nawet w dokumentacji nie␣
↪ma sensownego opisu jak to symetrycznie ustawić
    show.values = TRUE

)
```

```
title(main = "Piramida demograficzna ludności USA - rok 1948", cex.main = 1.5)
```

```
10 10
```

1. 5.1 2. 4.1 3. 4.1 4. 2.1



1.5.2 Z5 Wnioski

Źródło: `USAge.df`, pakiet `latticeExtra`.

Zainstalowanie pakietu 'latticeExtra' wymagało niestandardowego podejscia. Być może korzystanie z **anaconda** ułatwiłoby ten proces. Głównym sprawcą błędów być brak kompilatora **fortrana**.

Widzimy tutaj specyficzną dysproporcję - szeroka podstawa wykresu. Może być to obraz początku tzw. **baby boom** będącego niejako naturalną reakcją na powojenną stabilizację społeczną.

Wąska grupa 6-11 - skutek WW2

Da się również zauważyć większą liczbę kobiet w grupach 65+

1.6 Zadanie 6: Zakład tkalniczy

Zakład tkalniczy produkuje tkaninę o nominalnej długości 50 m. Nabywca zmierzył długości i otrzymał dane w formie przedziałów:

Przedział długości (m)	Liczba obserwacji
45–47	4
47–49	10
49–51	38
51–53	20
53–55	12
55–57	8

Zbadać, czy długość tej tkaniny może być modelowana za pomocą **rozkładu normalnego**. Parametry rozkładu oszacować na podstawie danych.

Hipotezy statystyczne:

- **H (hipoteza zerowa):** Długość tkaniny ma rozkład zgodny z rozkładem normalnym.
- **H (hipoteza alternatywna):** Długość tkaniny nie ma rozkładu zgodnego z rozkładem normalnym.

[179]:
```
# Środki klas
srodki <- c(46, 48, 50, 52, 54, 56)
```

[180]:
```
licznosci <- c(4, 10, 38, 20, 12, 8)
```

[181]:
```
n <- sum(licznosci)
```

[182]:
```
n
```

92

[183]:
```
srednia_hat <- sum(srodki * licznosci) / n # ^
```

[184]:
```
srednia_hat
```

51.0869565217391

[185]:
```
# Odchylenie standardowe (^)
wariancja_hat <- sum(licznosci * (srodki - srednia_hat)^2) / n
odch_stand_hat <- sqrt(wariancja_hat)
```

[186]:
```
odch_stand_hat
```

2.45681029471072

[187]:
```
wariancja_hat
```

6.0359168241966

[188]:
```
srednia_hat
```

51.0869565217391

[189]:
```
# granice przedziałów
dolne <- c(45, 47, 49, 51, 53, 55)
gorne <- c(47, 49, 51, 53, 55, 57)
```

[190]:
```
# Teoretyczne prawdopodobieństwa dla każdego przedziału
proby <- pnorm(gorne, mean = srednia_hat, sd = odch_stand_hat) -
         pnorm(dolne, mean = srednia_hat, sd = odch_stand_hat)
```

[191]:
```
proby
```

1. 0.0414903110812614 2. 0.149709263254968 3. 0.288069437787548 4. 0.296030058952559 5. 0.162477164423549 6. 0.0475634632458484

[192]:
```
# Oczekiwane liczności
oczekiwane <- proby * n
```

[194]:
```
tabela <- data.frame(
  Przedzial = paste(dolne, gorne, sep = "-"),
  Prawdopodobienstwo = proby,
  Oczekiwane = oczekiwane
)
```

[208]:
```
tabela$Obserwowane <- licznosci
```

[205]:
```
install.packages("tidyr", dependencies = TRUE)
```

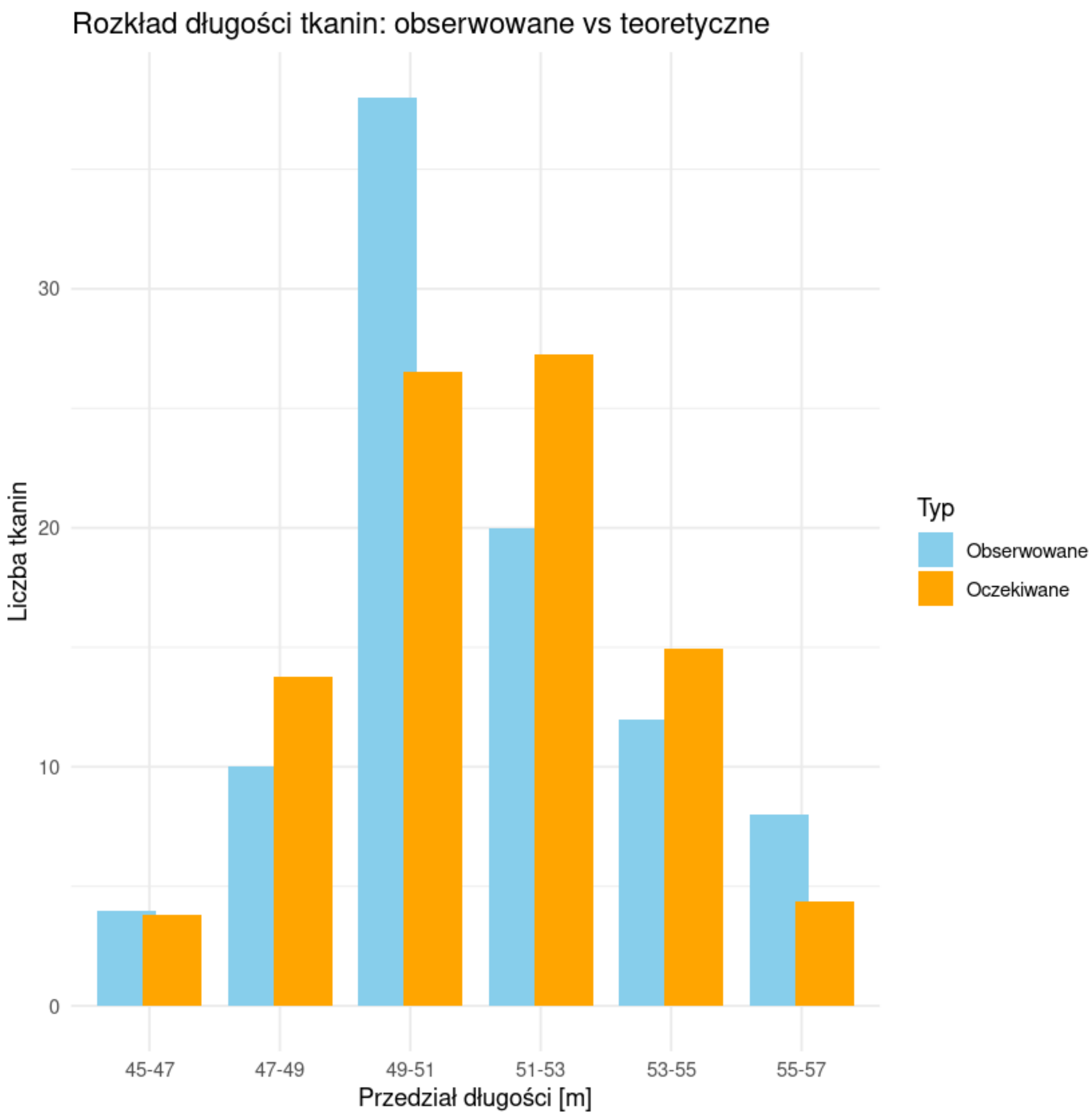
```
Installing package into '/home/kotmin/R/x86_64-pc-linux-gnu-library/4.5'
(as 'lib' is unspecified)

also installing the dependencies 'lazyeval', 'generics', 'rex', 'httr', 'dplyr',
'covr', 'data.table', 'repurrrsive'
```

[209]:
```
library(ggplot2)
library(tidyr)
```

[210]:
```
# Z długiego formatu do wykresu
tabela_long <- pivot_longer(
```

```
  tabela,
  cols = c("Obserwowane", "Oczekiwane"),
  names_to = "Typ",
  values_to = "Liczba"
)
```

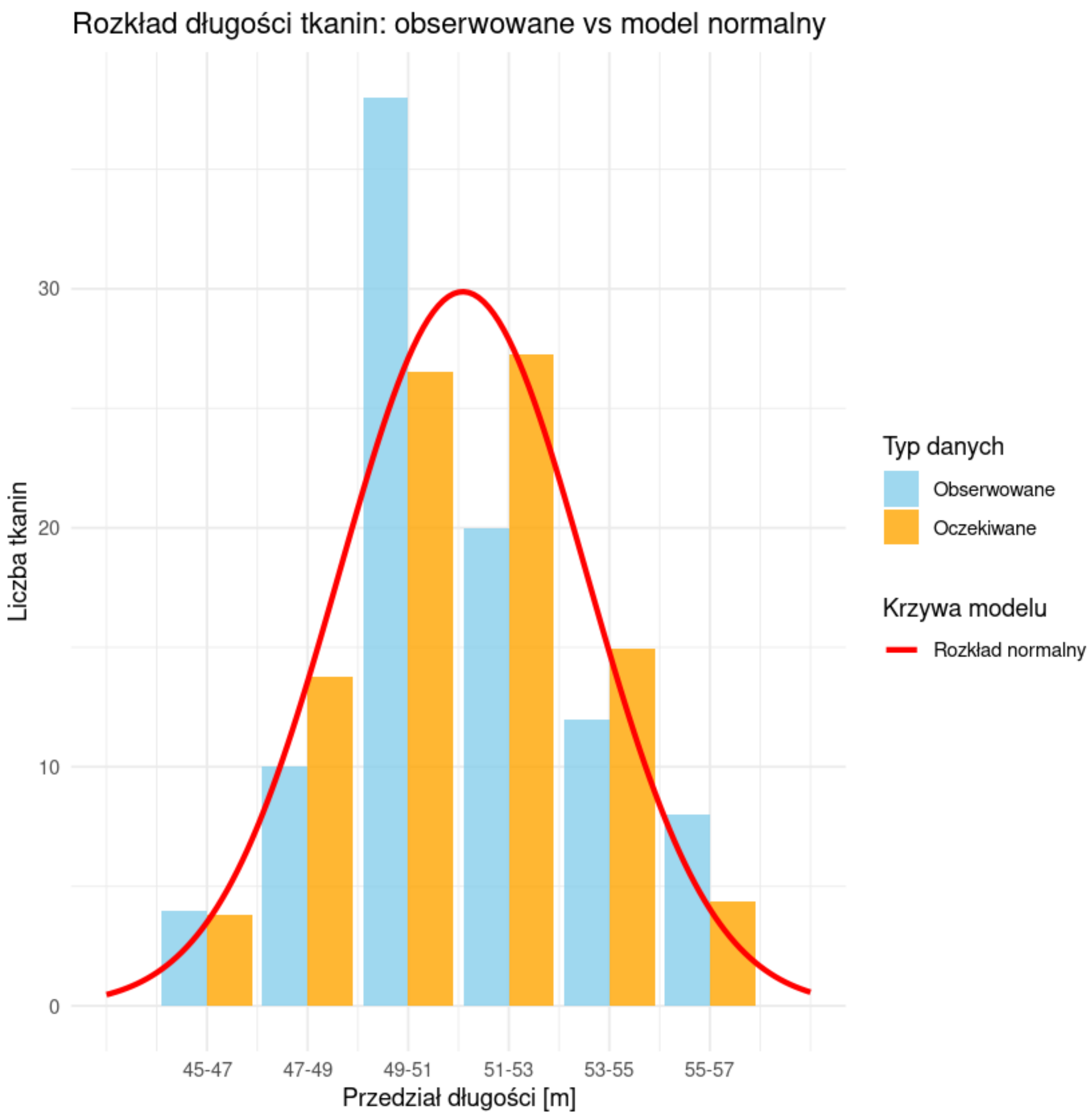
[211]:
```
ggplot(tabela_long, aes(x = Przedzial, y = Liczba, fill = Typ)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.7)) +
  scale_fill_manual(values = c("Obserwowane" = "skyblue", "Oczekiwane" =␣
 ↪"orange")) +
  labs(
    title = "Rozkład długości tkanin: obserwowane vs teoretyczne",
    x = "Przedział długości [m]",
    y = "Liczba tkanin",
    fill = "Typ"
  ) +
  theme_minimal()
```



[215]:
```
# spróbujmy jeszcze dodać poglądowo krzywą rozkładu normalnego
# Gęstość z rozkładu normalnego przeskalowana do sumy tkanin
gestosc_df <- data.frame(
  Przedzial = tabela$Przedzial,
  x = srodki,
  y = dnorm(srodki, mean = srednia_hat, sd = odch_stand_hat) * sum(licznosci) *
 ↪2  # 2 = szerokość klasy
)
```

[221]:
```
# Dane z etykietami
df_obs <- data.frame(x = srodki, y = licznosci, grupa = "Obserwowane",
 ↪Przedzial = tabela$Przedzial)
```

```r
df_exp <- data.frame(x = srodki, y = oczekiwane, grupa = "Oczekiwane",␣
↪Przedzial = tabela$Przedzial)
df_norm <- data.frame(x = x_gestosc, y = y_gestosc, grupa = "Rozkład normalny")

# słupki razem
df_bars <- rbind(df_obs, df_exp)


ggplot() +
  geom_col(data = df_bars, aes(x = x, y = y, fill = grupa), width = 1.8,␣
↪position = position_dodge(width = 1.8), alpha = 0.8) +
  geom_line(data = df_norm, aes(x = x, y = y, color = grupa), size = 1.2) +
  scale_fill_manual(
    values = c("Obserwowane" = "skyblue", "Oczekiwane" = "orange"),
    guide = guide_legend(order = 1)
  ) +
  scale_color_manual(
    values = c("Rozkład normalny" = "red"),
    guide = guide_legend(order = 2)
  ) +
  scale_x_continuous(
    name = "Przedział długości [m]",
    breaks = srodki,
    labels = tabela$Przedzial
  ) +
  labs(
    title = "Rozkład długości tkanin: obserwowane vs model normalny",
    y = "Liczba tkanin",
    fill = "Typ danych",
    color = "Krzywa modelu"
  ) +
  theme_minimal()
```



[195]:
```
# Zaokrąglenie tylko kolumn numerycznych
tabela$Prawdopodobienstwo <- round(tabela$Prawdopodobienstwo, 4)
tabela$Oczekiwane <- round(tabela$Oczekiwane, 2)
```

[196]:
```
tabela
```

A data.frame: 6 × 3

Przedzial <chr>	Prawdopodobienstwo <dbl>	Oczekiwane <dbl>
45-47	0.0415	3.82
47-49	0.1497	13.77
49-51	0.2881	26.50
51-53	0.2960	27.23
53-55	0.1625	14.95
55-57	0.0476	4.38

[197]:
```
sum(tabela$Prawdopodobienstwo)
```

0.9854

[199]:
```
# zakładamy że to błąd przybliżenia ~98% jest akceptowalne
```

[200]:
```
# chi-sqrt

## warunki
sum(tabela$Oczekiwane < 1)
mean(tabela$Oczekiwane < 5) * 100
```

0

33.3333333333333

[201]:
```
## przy chsqrt Pearsona możemy mieć problem z licznościami (>25% klas)
# pomimo tego zobaczmy co nam pokaże
test_chi <- chisq.test(x = oczekiwane, p = tabela$Oczekiwane /␣
 ↪sum(tabela$Oczekiwane), rescale.p = TRUE)
test_chi
```

```
Warning message in chisq.test(x = oczekiwane, p =
tabela$Oczekiwane/sum(tabela$Oczekiwane), :
"Chi-squared approximation may be incorrect"


	Chi-squared test for given probabilities

data:  oczekiwane
X-squared = 8.2373e-06, df = 5, p-value = 1
```

[203]:
```
# alt z MonteCarlo?
chisq.test(oczekiwane, p = tabela$Oczekiwane / sum(tabela$Oczekiwane),
           rescale.p = TRUE, simulate.p.value = TRUE, B = 10000)
```

```
Warning message in matrix(sample.int(nx, B * n, TRUE, prob = p), nrow = n):
"data length [906512] is not a sub-multiple or multiple of the number of rows
[90]"
```

```
        Chi-squared test for given probabilities with simulated p-value (based
        on 10000 replicates)

data:  oczekiwane
X-squared = 8.2373e-06, df = NA, p-value = 1.007
```

[ ]:
```r
# nie mamy podstaw do odrzucenia h0
```

[222]:
```r
# łączenie klas (pierwszych i ostatnich)
nowe_przedzialy <- c("45-49", "49-51", "51-53", "53-57")
licznosci_polaczone <- c(4 + 10, 38, 20, 12 + 8)
oczekiwane_polaczone <- c(3.82 + 13.77, 26.50, 27.23, 14.95 + 4.38)
```

[223]:
```r
# sprawdzenie warunków dla chi-kwadrat z nowymi przedziałami
sum(oczekiwane_polaczone < 1)
mean(oczekiwane_polaczone < 5) * 100
```

0

0

[224]:
```r
test_chikwadrat_polaczony <- chisq.test(
  x = licznosci_polaczone,
  p = oczekiwane_polaczone / sum(oczekiwane_polaczone),
  rescale.p = TRUE
)
```

[225]:
```r
test_chikwadrat_polaczony
```

```
        Chi-squared test for given probabilities

data:  licznosci_polaczone
X-squared = 7.5339, df = 3, p-value = 0.05669
```

[226]:
```r
# technicznie rzecz biorąc brakuje podstaw do odrzucenia hipotezy zerowej
```

1.6.1 Z6 Wnioski

ipotezy statystyczne:

- **H (hipoteza zerowa):** długość tkanin jest zgodna z rozkładem normalnym,
- **H (hipoteza alternatywna):** długość tkanin nie jest zgodna z rozkładem normalnym.

Nie mniej porównując wizualnie rozkłady teoretyczne z empirycznymi oraz krzywą normalną możemy zauważyć, że liczność przedziału 49-51 mogła by być głównym powodem **braku podstaw do odrzucenia h0**

Wartości testu chi-kwadrat Pearsona dla połączonych klasy przyjęły p-value bardzo bliskie progu odrzucenia.

Na poziomie istotności **= 0.05** brak podstaw do odrzucenia hipotezy H .
Rozkład długości tkanin można uznać za **zgodny z rozkładem normalnym**. Przy obecnym stopniu agregacji danych

Po połączeniu klas dla spełnienia warunków testu, brak podstaw do odrzucenia hipotezy o zgodności z rozkładem normalnym. Wynik sugeruje, że model normalny może być użytecznym przybliżeniem dla długości tkaniny — przy założeniu umiarkowanego poziomu szczegółowości