Wydział Matematyki
i Nauk Informacyjnych

POLITECHNIKA WARSZAWSKA

Rachunek prawdopodobieństwa
i
Elementy statystyki matematycznej

Omówienie laboratorium

Prowadząca: dr Katarzyna Danielak

Paweł Szymański, Andrzej Wrzesiński, Julian Zalewski, Antoni Zasada, Dominik Zieliński

7 czerwca 2025

Spis treści

1. Estymacja punktowa, własności estymatorów

Zadanie 1.1. Wygenerować $N = 10000$ obserwacji $X_1, X_2, \ldots, X_n$ z rozkładu

(a) dwupunktowego Binom$(1, \frac{1}{4})$

(b) wykładniczego Exp$(\frac{1}{3})$

(c) Cauchy'ego $\mathcal{C}(0, 1)$.

Dla każdego z powyższych przypadków wyznaczyć wykres ciągu $\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_N$, gdzie $\bar{X}_n$ oznacza średnią z pierwszych n obserwacji, $n = 1, 2, \ldots, N$, czyli

$$\bar{X}_n = \frac{S_n}{n} = \frac{\sum_{i=1}^{n} X_i}{n}.$$

Wyciągnąć wnioski dotyczące zachowania się uzyskanych ciągów średnich.

Rozwiązanie: W języku R do generowania obserwacji z zadanego rozkładu służą funkcje o nazwach postaci `r<nazwa rozkładu>`. Ich pierwszym argumentem jest liczba obserwacji, a kolejne to parametry zależne od rozkładu. W tym zadaniu korzystamy zatem odpowiednio z funkcji `rbinom(N, 1, 1/4)`, `rexp(N, 1/3)` i `rcauchy(N, 0, 1)`.

Wektor $(\bar{X}_n)$ średnich z pierwszych n obserwacji otrzymamy, dzieląc (element po elemencie) wektor sum skumulowanych uzyskany funkcją `cumsum` przez wektor $[1, 2, \ldots, N]$, który w R zapiszemy jako `1:N`.

Następnie chcemy wyświetlić wykresy wyznaczonych ciągów. Wykorzystujemy do tego `plot`. Aby uzyskać wykres będący linią musimy ustawić parametr `type` na `l` (jak *line*).

Rozkłady z podpunktów (a) i (b) posiadają skończone wartości oczekiwane. Są one równe

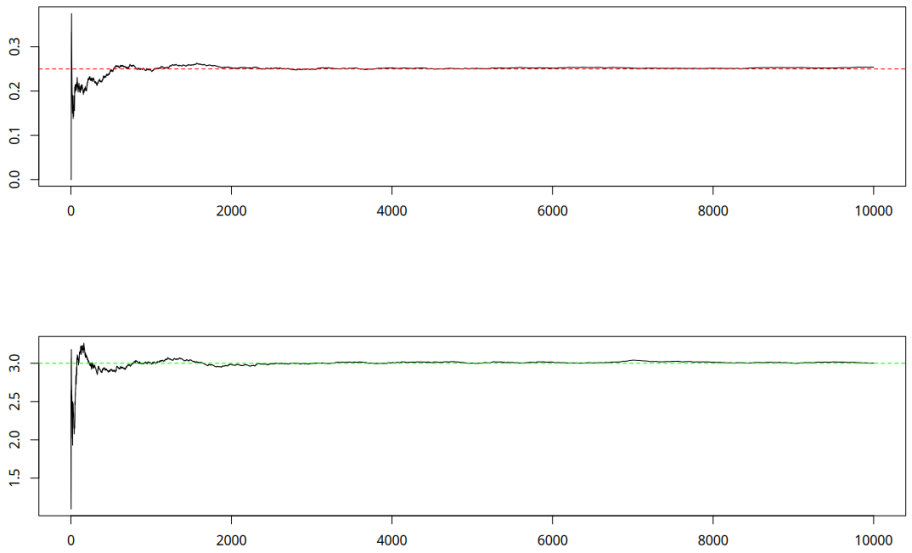
$$\mu_a = \frac{1}{4} \cdot 1 + \frac{3}{4} \cdot 0 = \frac{1}{4}$$
$$\mu_b = \frac{1}{\frac{1}{3}} = 3.$$

Spodziewamy się więc, że ich wykresy zobrazują działanie mocnego prawa wielkich liczb Kołmogorowa, tj. zauważymy zbieżność ciągu średnich do obliczonych wartości oczekiwanych. Do wykresów dodamy pomocnicze proste $y = \mu_a$ i $y = \mu_b$ korzystając z funkcji `abline`.

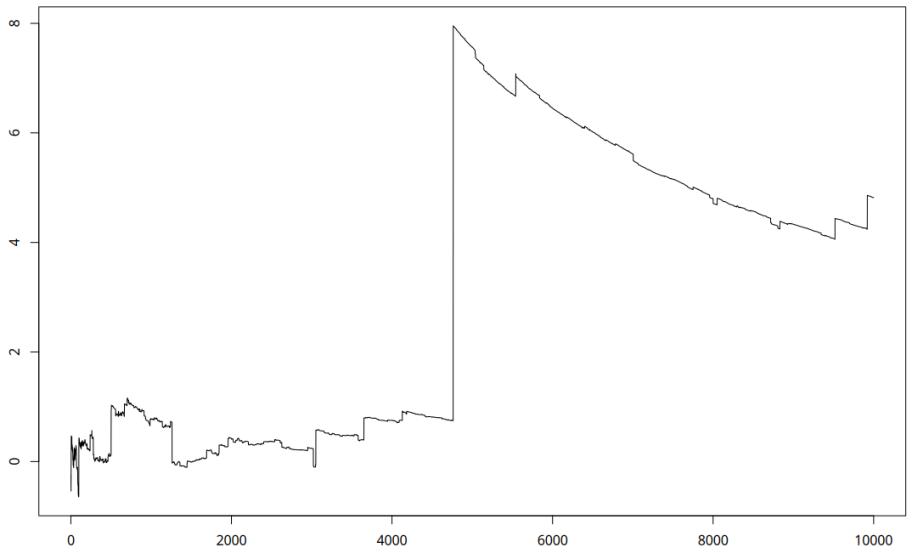
Dla rozkładu Cauchy'ego w podpunkcie (c) nie możemy spodziewać się zbieżności, ponieważ wartość oczekiwana jest w jego przypadku niezdefiniowana.

Pełny kod rozwiązania (*Zadaniem komendy* `rm(list = ls())` *w pierwszej linii jest usunięcie zalegających zmiennych z wcześniej uruchomionych skryptów w celu uniknięcia konfliktów.*):

```
rm(list = ls())

N = 1e4
n = 1:N
par(mfrow=c(3,1))

x_a = rbinom(N, size=1, prob=1/4)
S_a = cumsum(x_a)
M_a = S_a/n
plot(M_a, type="l", xlab="", ylab="")
abline(h = 1/4, lty = 2, col="red")

x_b = rexp(N, rate=1/3)
S_b = cumsum(x_b)
M_b = S_b/n
plot(M_b, type="l", xlab="", ylab="")
```



Rysunek 1: Wykresy (a) i (b), linie przerywane pokazują wartości oczekiwane rozkładów.



Rysunek 2: Wykres (c), rozkład nie spełnia MPWL ze względu na brak wartości oczekiwanej.

```
abline(h = 3, lty = 2, col="green")

x_c = rcauchy(N, location=0, scale=1)
S_c = cumsum(x_c)
M_c = S_c/n
plot(M_c, type="l", xlab="", ylab="")
```

Zadanie 1.2. Wygenerować próbę $Y = (Y_1, \ldots, Y_n),\ n = 500$ z rozkładu normalnego $N(\mu = 4, \sigma = 2)$. Utworzyć podpróby $X_i = (Y_1, \ldots, Y_i),\ i = 1, \ldots, n$ i wyznaczyć ciągi:

- średnich: $\{\bar{X}_i : i = 1, \ldots, n\}$,
- median: $\{\mathrm{Med}_i : i = 1, \ldots, n\}$,
- odchyleń standardowych: $\{S_i : i = 2, \ldots, n\}$,
- rozstępów międzykwartylowych podzielonych przez 1.35: $\{D_i = \mathrm{IQR}_i/1.35 : i = 2, \ldots, n\}$.

(a) Narysować na wspólnym wykresie ciągi średnich i median. Przeanalizować wpływ liczności próby na zachowanie się średniej i mediany z próby. Czy statystyki te wydają się być sensownymi estymatorami parametru położenia μ w tym modelu?

(b) Narysować na wspólnym wykresie ciągi odchyleń standardowych i rozstępów międzykwartylowych podzielonych przez 1.35. Przeanalizować wpływ liczności próby na zachowanie się tych statystyk. Czy wydają się one być sensownymi estymatorami parametru rozproszenia σ w tym modelu?

Rozwiązanie: Podobnie jak w zadaniu 1. generujemy próby z rozkładu normalnego $N(\mu = 4, \sigma = 2)$. Do obliczania ciągów w treści używamy funkcji R:

- `mean(wektor)` – oblicza średnią arytmetyczną:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

- `mean(wektor, na.rm=TRUE)` – oblicza średnią, pomijając wartości `NA` (*not available* = brak danych),
- `median(wektor)` – oblicza medianę (wartość środkowa),
- `quantile(wektor, 0.25)` – dolny kwartyl (Q_1),
- `quantile(wektor, 0.75)` – górny kwartyl (Q_3),
- `quantile(wektor, c(0.1, 0.99, 0.85))` – decyle, percentyle i kwantyle (np. 10%, 99%, 85%),
- `max(wektor) - min(wektor)` – rozstęp (zakres):
- `IQR(wektor)` – rozstęp międzykwartylowy:

$$\mathrm{IQR} = Q_3 - Q_1$$

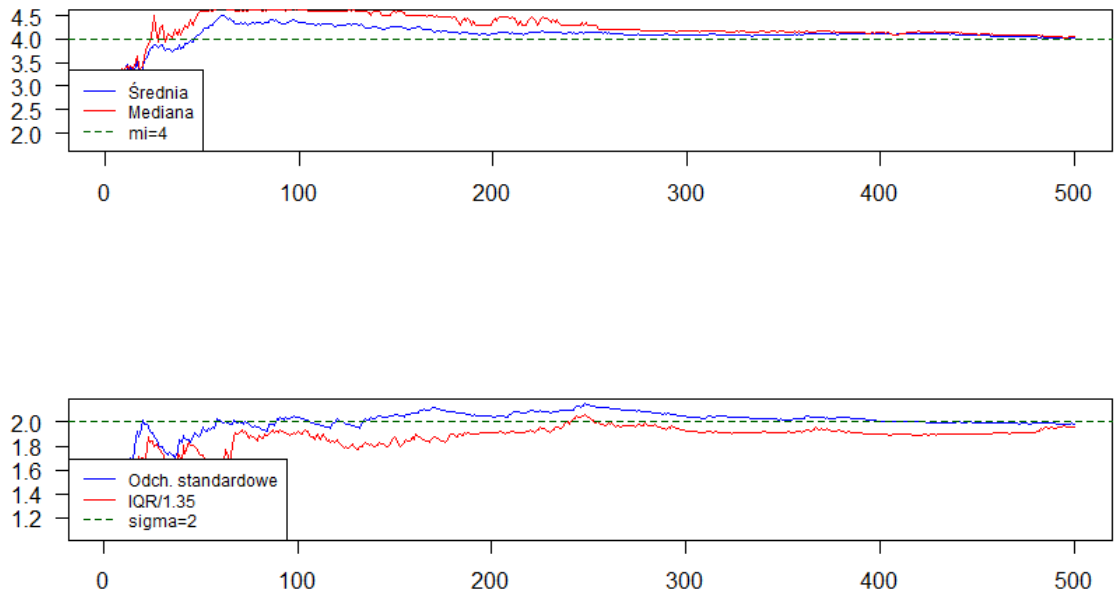
- `var(wektor)` – wariancja:

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

- `sd(wektor)` – odchylenie standardowe:

$$S = \sqrt{S^2}$$

W pętli `for` obliczamy podpróby używając `x = y[1:i]` oraz liczymy miary podane w zadaniu jednocześnie umieszczając je do wcześniej stworzonych wektorów. Do utworzenia wyżej wymienionych wektorów używamy `numeric`, która służy do tworzenia obiektów typu `numeric` o zadanej długości.



Rysunek 3: Wykresy dla podpunktów (a) i (b)

Z wykresów wynika, że wszystkie sprawdzane estymatory są sensowne.

Pełny kod rozwiązania:

```
rm(list = ls())
par(mfrow=c(2,1))
n = 500
mi = 4
sigma = 2
y = rnorm(n, mean=mi, sd=sigma)
mean_y = numeric(n)
median_y = numeric(n)
sd_y = numeric(n)
iqr_y = numeric(n)

for (i in 1:n)
{
  x = y[1:i]
  mean_y[i] = mean(x)
  median_y[i] = median(x)
  sd_y[i] = sd(x)
  iqr_y[i] = IQR(x)/1.35
}
plot(1:n, mean_y, type="l", col="blue", ylab="", las=1)
lines(1:n, median_y, col="red")
abline(h = mi, col="darkgreen", lty=2)
legend("bottomleft", legend=c("Średnia", "Mediana", "mi=4"),
       col=c("blue", "red", "darkgreen"), lty=c(1,1,2))

plot(2:n, sd_y[2:n], type="l", col="blue", ylab=" ", las=1)
lines(2:n, iqr_y[2:n], col="red")
abline(h=sigma, col="darkgreen", lty=2)
legend("bottomleft", legend=c("Odch. standardowe", "IQR/1.35", "sigma=2"),
       col=c("blue", "red", "darkgreen"), lty=c(1,1,2))
```

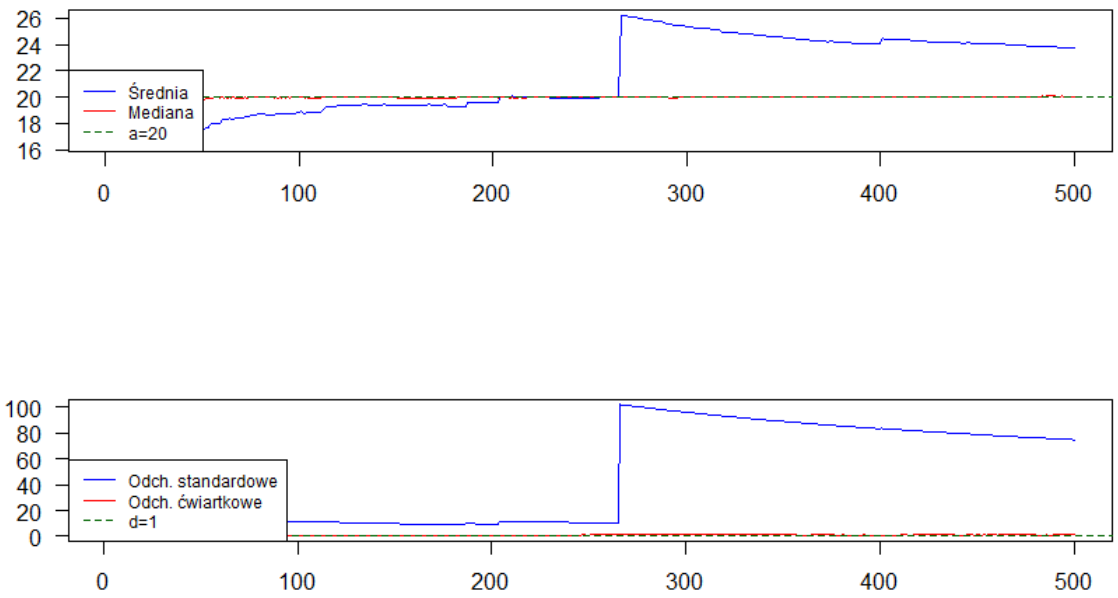
Zadanie 1.3. Wygenerować próbę $Y = (Y_1, \ldots, Y_n),\ n = 500$ z rozkładu Cauchy'ego $C(a = 20, d = 1)$. Utworzyć podpróby $X_i = (Y_1, \ldots, Y_i),\ i = 1, \ldots, n$ i wyznaczyć ciągi:

- średnich: $\{\bar{X}_i : i = 1, \ldots, n\}$,
- median: $\{\text{Med}_i : i = 1, \ldots, n\}$,
- odchyleń standardowych: $\{S_i : i = 2, \ldots, n\}$,
- odchyleń ćwiartkowych: $\{\text{SQR}_i = \text{IQR}_i/2 : i = 2, \ldots, n\}$.

(a) Narysować na wspólnym wykresie ciągi średnich i median. Przeanalizować wpływ liczności próby na zachowanie się średniej i mediany z próby. Czy statystyki te wydają się być sensownymi estymatorami parametru położenia a w tym modelu?

(b) Narysować na wspólnym wykresie ciągi odchyleń standardowych i ćwiartkowych. Przeanalizować wpływ liczności próby na zachowanie się tych statystyk. Czy wydają się one być sensownymi estymatorami parametru rozproszenia d w tym modelu?

Rozwiązanie: Zadanie technicznie całkowicie analogiczne do zadania 2.



Rysunek 4: Wykresy dla podpunktów (a) i (b)

Z pierwszego wykresu wynika, że mediana jest dobrym estymatorem parametru położenia a, podczas gdy średnia nie jest. Patrząc na drugi wykres wnioskujemy, że odchylenie ćwiartkowe jest sensownym estymatorem parametru rozproszenia d, w przeciwieństwie do odchylenia standardowego.

Pełny kod rozwiązania:

```
rm(list = ls())

par(mfrow=c(2,1))

n = 500
a = 20
d = 1
y = rcauchy(n, location=a, scale=d)
mean_y = numeric(n)
median_y = numeric(n)
sd_y = numeric(n)
sqr_y = numeric(n)
for (i in 1:n)
{
  x = y[1:i]
  mean_y[i] = mean(x)
  median_y[i] = median(x)
  sd_y[i] = sd(x)
  sqr_y[i] = IQR(x)/2
}

plot(1:n, mean_y, type="l", col="blue", xlab="", ylab="", las=1)
lines(1:n, median_y, col="red")
abline(h = a, col="darkgreen", lty=2)
legend("bottomleft", legend=c("Średnia", "Mediana", "a=20"),
       col=c("blue", "red", "darkgreen"), lty=c(1,1,2), cex=0.75)

plot(2:n, sd_y[2:n], type="l", col="blue", xlab="", ylab="", las=1)
lines(2:n, sqr_y[2:n], col="red")
abline(h = d, col="darkgreen", lty=2)
legend("bottomleft", legend=c("Odch. standardowe", "Odch. ćwiartkowe", "d=1"),
       col=c("blue", "red", "darkgreen"), lty=c(1,1,2), cex=0.75)
```

Zadanie 1.4. Niech $X_1, X_2, \ldots, X_n$ będzie prostą próbą losową z rozkładu wykładniczego Exp(λ), gdzie:

$$f_\lambda(x) = \begin{cases} \lambda e^{-\lambda x}, & x > 0, \\ 0, & x \leq 0, \end{cases} \quad \text{gdzie } \lambda > 0.$$

(a) W celu oszacowania czasu działania pewnych bateryjek, dział kontroli jakości zmierzył czas pracy 8 losowo wybranych bateryjek i otrzymał następujące wyniki (w godz.):

483, 705, 2623, 347, 620, 2719, 1035, 421

Wiadomo, że czas pracy tych bateryjek ma rozkład wykładniczy Exp(λ) z nieznanym $\lambda > 0$. Dla danych zebranych przez dział kontroli jakości, podać wartość estymatora największej wiarygodności parametru λ.

(b) Dla danych z pkt. (a) wyznaczyć estymator największej wiarygodności dla:

- średniego czasu działania bateryjki,
- prawdopodobieństwa, że bateryjka będzie działać krócej niż 1000 godz.

Rozwiązanie: Do zapisania danych używamy funkcji `c` (combine) aby stworzyć wektor poprzez połączenie wartości pomiarów. Do obliczenia estymatora największej wiarygodności używamy `fitdistr` z wcześniej załadowanej biblioteki *MASS*. Posłużyła nam do tego funkcja `library`. `fitdistr` jako argument `densfun` może przyjąć:

- *beta*,
- *cauchy*,
- *chi-squared*,
- *exponential*,
- *geometric*,
- *log-normal*,
- *lognormal*,
- *logistic*,
- *negative binomial*,
- *normal*,
- *Poisson*,
- *weibull*

Estymator średniego czasu działania bateryjki to $\frac{1}{\hat{\lambda}}$, z własności rozkładu wykładniczego. W języku R do obliczania wartości dystrybuanty zadanego rozkładu służą funkcje o nazwach postaci `p<nazwa rozkładu>` („p" pochodzi od *probability density function* (*PDF*)). W naszym przypadku używamy `pexp`.

Pełny kod rozwiązania:

```
rm(list = ls())
library(MASS)

results = c(483, 705, 2623, 347, 620, 2719, 1035, 421)
est_lambda = fitdistr(results, "exponential")$est
est_mean = 1/est_lambda
est_pdf = pexp(1000, est_lambda)
```

Zadanie 1.5. Niech Gamma(a, β) oznacza rozkład gamma z parametrem kształtu a i parametrem β, tzn.

$$f(x) = \begin{cases} \dfrac{\beta^a}{\Gamma(a)} x^{a-1} e^{-\beta x}, & x > 0, \\ 0, & x \leq 0, \end{cases} \quad \text{gdzie } a > 0, \beta > 0.$$

(a) Wygenerować $n = 100$ obserwacji z rozkładu Gamma$(3, 2)$.

(b) Przyjąć, że zapomniano wartości parametrów rozkładu gamma, z którego wygenerowano dane i, używając R, oszacować te parametry stosując metodę największej wiarygodności.

Rozwiązanie: Zadanie pominięte na zajęciach. Rozwiązanie nadesłane jako praca domowa.

```
# Autor: Dominik Zieliński
# Zestaw 1.
# Zadanie 5. Szacowanie parametrów rozkładu Gamma metodą Największej
# Wiarygodności

# Początkowo generujemy próbę losową z rozkładu Gamma z parametrami:
alpha = 3
beta = 2

# Liczba obserwacji:
n = 100

# Parametry wejściowe funkcji rgamma:
# n - liczba obserwacji
# shape - Alpha
# rate - Beta
# scale - 1 / rate
X = rgamma(n, shape=alpha, rate=beta) # Próba losowa z rokładu Gamma(3, 2)

# Należy załadować bibliotekę MASS m.in. żeby skorzystać z funkcji fitdistr
library("MASS")

# Estymujemy parametry przy użyciu funkcji fitdistr
est = fitdistr(X, densfun='gamma', lower=0.001)

# Wyniki (tutaj tak naprawdę jest właściwy koniec zadania, reszta to dodatek)
print(est)

# Porównanie unormowanego histogramu danych z dopasowaną dystrybuantą NW
# oraz dystrybuantą dla parametrów początkowych:
par(mfrow=c(2,3))
hist(X, probability = TRUE, breaks=20, col = "lightblue", main = "Dane vs.
     Dopasowany rozkład Gamma", las=1)
curve(dgamma(x, shape = est$estimate['shape'], rate = est$estimate['rate']),
      add = TRUE, col = "red", lwd = 2)
curve(dgamma(x, shape = alpha, rate = beta), add = TRUE, col = 'green', lwd = 2)
legend(x = 'topright', legend=c("Rozkład Gamma(3,2)", "Rozkład Gamma NW"),
       col=c("green", "red"), lty=c(1,1))

# Wektory estymowanych parametrów
alphas = c(0, 0)
betas = c(0, 0)

# Zbieżność estymatora na wygenerowanej próbie losowej
# W pętli wyznaczam estymatory NW dla pierwszych i wyrazów próby losowej, wyniki
# zapisuje w odpowiednich wektorach
for (i in 2:n) {
  est_v = fitdistr(X[1:i], densfun='gamma')
  alphas[i-1] = est_v$estimate['shape']
  betas[i-1] = est_v$estimate['rate']
}
```

```r

# Wykresy zbieżności
plot(alphas, type='l', col='brown', main="Zbieżność parametru alpha", las=1)
abline(h=3, col='blue')
legend(x='topright', legend=c("Estymacja alpha", "Rzeczywista alpha"),
       col=c("brown", "blue"), lty=1)
plot(betas, type='l', col='brown', main="Zbieżność parametru beta", las=1)
abline(h=2, col='yellow')
legend(x='topright', legend=c("Estymacja beta", "Rzeczywista beta"),
       col=c("brown", "yellow"), lty=1)


# Powtórzmy eksperyment B razy, aby zbadać rozkład estymatora NW parametrów
# alpha i beta
B <- 1000 # Liczba powtórzeń eksperymentu
alpha_est <- numeric(B) # Kolejne estymacje parametru alpha
beta_est <- numeric(B) # Kolejne estymacje parametru beta

for (i in 1:B) {
  Xi <- rgamma(n, shape=alpha, rate=beta)
  esti <- fitdistr(Xi, densfun='gamma')
  alpha_est[i] <- esti$estimate['shape']
  beta_est[i] <- esti$estimate['rate']
}

# Rozkłady estymatorów alpha i beta - wykresy
hist(alpha_est, probability = TRUE, main="Rozkład estymatora alpha",
     col="lightblue", las = 1)
abline(v=alpha, col="red", lwd=2)
legend(x='topright', legend=c('alpha'), col=c('red'), lty=1)
hist(beta_est, probability= TRUE, main="Rozkład estymatora beta",
     col="lightgreen", las = 1)
abline(v=beta, col="pink", lwd=2)
legend(x='topright', legend=c('beta'), col=c('pink'), lty=1)

# Estymatory wartości oczekiwanej estymatorów:
Ealpha = mean(alpha_est)
Ebeta = mean(beta_est)

cat("War. oczek. est. alphy: ", Ealpha, "War. oczek. est. bety: ",
    Ebeta, "\n")
```

Zadanie 1.6.

(a) Wybrać $\theta > 0$.

(b) Wygenerować $N = 10000$ k-elementowych próbek ($k = 20$) z rozkładu jednostajnego $\mathcal{U}([0, \theta])$.

(c) Porównać empirycznie obciążenie estymatora metody momentów i ENW parametru θ.

Rozwiązanie: W kodzie ustalamy $\theta = 2$. Tworzymy wektor `estimators` używając funkcji `replicate`, która jako pierwszy argument przyjmuje liczbę powtórzeń funkcji podanej jako drugi argument której wynik zostanie połączony do wektora `estimators`. Jak widać niżej, dla rozkładu jednostajnego estymatorem metody momentów jest podwojona średnia, a estymatorem największego wiarygodności jest wartość maksymalna z próbek. Wyniki rysujemy funkcją `plot`.

Wyznaczmy estymator metodą momentów:

$$M_1 = E(X) = \int_0^\theta \frac{x}{\theta}\,\mathrm{d}x = \frac{1}{2} \cdot \frac{\theta^2}{\theta} = \frac{\theta}{2} = \frac{X_1 + \cdots + X_k}{k} = \bar{X}$$

Zatem $\hat{\theta}_{\mathrm{MM}} = 2\bar{X}$.

Teraz skorzystamy z metody największej wiarygodności. Określmy najpierw funkcję gęstości rozkładu.

$$f(x) = \begin{cases} \frac{1}{\theta}, & x \in [0, \theta] \\ 0, & \text{wpp.} \end{cases}$$

Możemy teraz wyznaczyć funkcję wiarygodności.

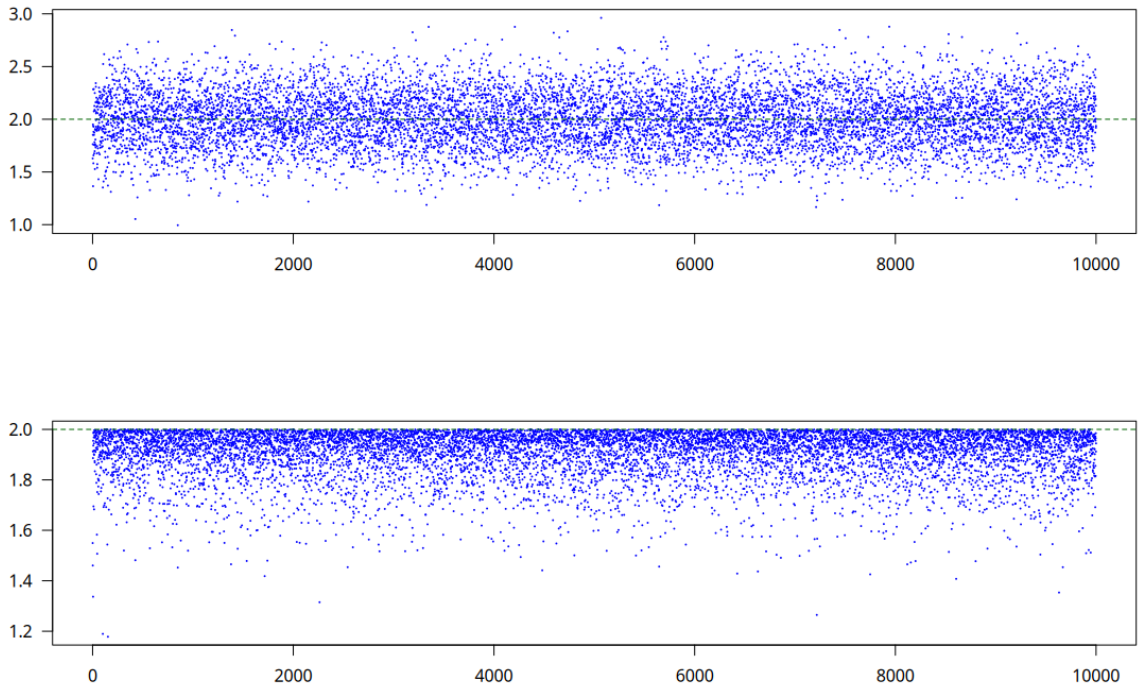
$$L(x_1, \ldots, x_k; \theta) = \prod_{i=1}^{k} f(x_i; \theta) = \prod_{i=1}^{k} \frac{1}{\theta} = \theta^{-k}, \text{ dla } x_i \in [0, \theta]\ (i = 1, 2, \ldots, k)$$

Ostatecznie chcemy znaleźć ekstremum funkcji L, więc możemy równie dobrze operować na jej logarytmie (ponieważ logarytm naturalny jest funkcją ściśle rosnącą).

$$\ln L(x_1, \ldots, x_k; \theta) = \ln\left(\theta^{-k}\right) = -k \ln(\theta)$$

Zatem pochodna funkcji wiarygodności po θ to $-k/\theta$. Jest to funkcja malejąca względem θ, więc maksimum spodziewamy się w lewym krańcu przedziału - czyli dla najmniejszej możliwej wartości $\theta = \max\{x_1, x_2, \ldots, x_k\}$. Wobec tego, $\hat{\theta}_{\mathrm{NW}} = \max\{X_1, X_2, \ldots, X_k\}$.

Na poniższych wykresach niebieskimi punktami zaznaczone są wyznaczone wartości $2\bar{X}$ oraz $\max\{X_1, \ldots, X_k\}$ dla każdej z 20-elementowych próbek.



Przypomnijmy definicję obciążenia estymatora $\hat{\theta}$: $B(\hat{\theta}) \coloneqq E(\hat{\theta}) - \theta$, gdzie $\theta \in \Theta$. Na wykresach wyraźnie widać, że estymator wyznaczony metodą momentów (górny wykres) ma wartości stosunkowo symetrycznie rozłożone wokół $\theta = 2$. Możemy się zatem spodziewać, że jest to estymator nieobciążony.

Natomiast estymator największej wiarygodności zawsze przyjmuje wartości nie większe od $\theta = 2$, więc jego wartość oczekiwana musi być mniejsza od $\theta = 2$. Jest to zatem estymator obciążony.

Pełny kod rozwiązania:

```
rm(list = ls())

par(mfrow=c(2,1))
N = 1e4
k = 20
theta = 2
estimators = replicate(N,
{
  x = runif(k, 0, theta)
  c(2 * mean(x), max(x))
})

plot(1:N, estimators[1,], cex=0.05, col="blue", ylab="", las=1)
abline(h = theta, col="darkgreen", lty=2)
plot(1:N, estimators[2,], cex=0.05, col="blue", ylab="", las=1)
abline(h = theta, col="darkgreen", lty=2)
```

Uwagi

- Do przeglądania dokumentacji używamy `?<nazwa komendy>`, a do wyszukiwania `??`. Przykładowo, aby wyszukać w dokumentacji wszystkie funkcje dotyczące rozkładu jednostajnego wpiszemy `??unif`.
- R jest „case sensitive”, tzn. funkcje lub zmienne różniące się tylko wielkością liter w nazwie zostaną uznane za różne.
- Nie należy nazywać zmiennych `c T F t dt df pt pf rt rf qt qf`, ponieważ są one zarezerwowane dla wbudowanych w R funkcji.
- W R są dwa operatory przypisania: `<-` i `=`. Jest pomiędzy nimi niewielka różnica - znak `=` powoduje przypisanie „lokalne” (dostępne tylko w ramach funkcji, w której zostało zapisane), a `<-` służy do przypisania globalnego, tj. takiego, które będzie dostępne w całej przestrzeni roboczej.
- Gdy chcemy wybrać tylko jedną kolumnę z ramki, to możemy również skorzystać z operatora `$`. Po operatorze `$` możemy podać całą nazwę zmiennej lub jej części. Jeżeli podamy część, to wynikiem będzie kolumna o nazwie rozpoczynającej się od wskazanego napisu.
- Opcja `las` w funkcji `par` kontroluje kierunek osi etykiet.
- Do wyświetlania zmiennych można:
  - otoczyć wyrażenie nawiasami np. `(x=2)`
  - użyć funkcji np. `print(x)`
  - napisać samą nazwę zmiennej w linijce np. `x`
- Przydatna „ściąga” wyjaśniająca możliwości programu RStudio

2. Własności rozkładów, statystyka opisowa, estymacja przedziałowa

Zadanie 2.1. Wygenerować dwie próby losowe: 20 i 100 elementową z rozkładu standardowego normalnego. Narysować dla obu prób dystrybuanty empiryczne i porównać je z odpowiednią dystrybuantą teoretyczną.

Rozwiązanie: W języku R do wygenerowania próby losowej z rozkładu normalnego służy funkcja o nazwie `rnorm`. Pierwszym argumentem jest liczba elementów próby, drugim wartość oczekiwana, a trzecim odchylenie standardowe. Aby otrzymać rozkład normalny standardowy za wartość średnią należy przyjąć $\mu = 0$, a za odchylenie $\sigma = 1$.

Dystrybuanta empiryczna jest to estymator dystrybuanty rozkładu z którego pochodzi próba. Można ją określić za pomocą wzoru:

$$\hat{F}_n(t) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(X_i \leq t)$$

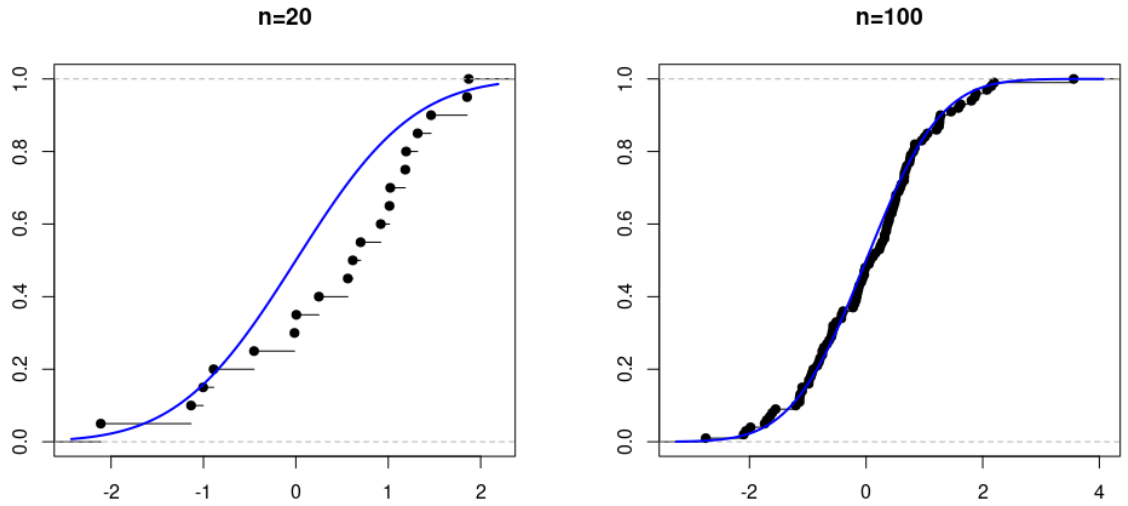
gdzie

$$\mathbb{1}(X_i \leq t) = \begin{cases} 1, X_i \leq t \\ 0, X_i > t \end{cases}$$

Do jej obliczania w R dostępna jest funkcja `ecdf`. Przyjmuje ona za argument wektor wartości próby.

Po obliczeniu dystrybuant rysujemy ich wykresy. W tym celu korzystamy z funkcji `plot`.

Ze względu na rozkład, który opisują, powinny być one zbliżone do dystrybuanty rozkładu normalnego standardowego. W celu porównania na wykresy nakładamy niebieskie krzywe obrazujące dokładną dystrybuantę rozkładu normalnego standardowego. Do uzyskania tego efektu służy funkcja `curve`, która potrzebuje wektora punktów według których ma być narysowana krzywa (w tym przypadku jest to wynik funkcji `pnorm` zwracającej dokładną dystrybuantę rozkładu normalnego) i dodatkowe argumenty, jak chociażby `col` w celu zmiany koloru linii.



Rysunek 5: Wykresy dystrybuant: po lewej z 20 elementami, po prawej z 100. Niebieska linia to spodziewany kształt.

Pełny kod rozwiązania:

```
# zad 2.1

mu = 0
sigma = 1
prob1 = rnorm(n=20, mean=mu, sd=sigma)
prob2 = rnorm(n=100, mean=mu, sd=sigma)

# to achieve two plots next to each other
par(mfrow=c(1,2))

plot(ecdf(prob1), main='n=20', xlab='', ylab='')
curve(pnorm(x, mu, sigma), add=TRUE, col='blue', lwd=2)

plot(ecdf(prob2), main='n=100', xlab='', ylab='')
curve(pnorm(x, mu, sigma), add=TRUE, col='blue', lwd=2)
```

Zadanie 2.2. Wygenerować $N = 1000$ obserwacji z rozkładu normalnego standardowego. Utworzyć histogram oraz estymator jądrowy dla tej próby. Nałożyć na uzyskany obraz wykres gęstości teoretycznej rozkładu normalnego.

Rozwiązanie: Dokładnie tak jak w zadaniu 1. generujemy próbę losową rozkładu normalnego standardowego, tyle że tym razem dla 1000 elementów.

Następnie tworzymy histogram. W R można to zrobić wywołując gotową funkcję `hist`, która przyjmuje wektor elementów, którego histogram ma zostać narysowany, wartość `br` (czyli *break length*) określającą grubość słupków histogramu i `freq` decydującą czy przedstawiona ma być gęstość czy częstotliwość rozkładu.

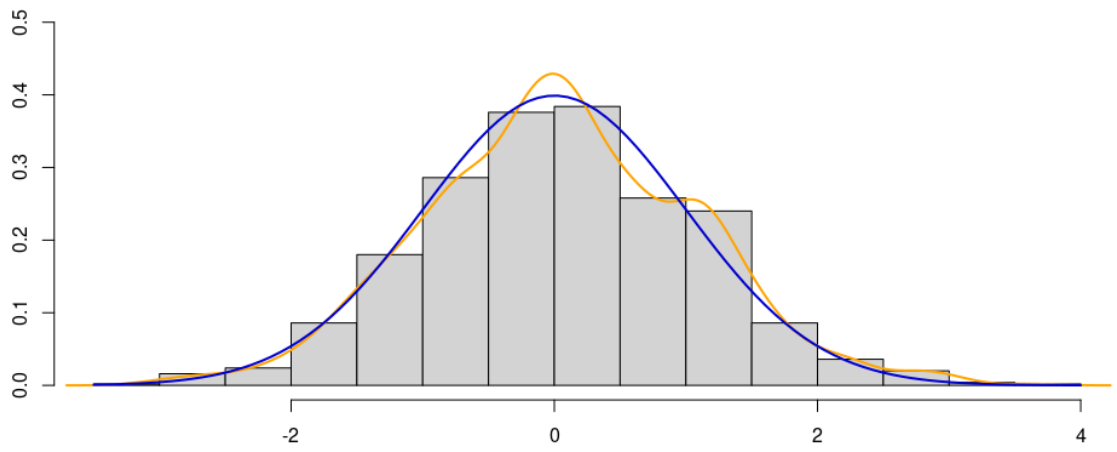
Estymator jądrowy jest to estymator służący wygładzaniu podanego rozkładu próby. Dla N elementowej realizacji próby losowej $x_1, x_2, ..., x_N$ jest on definiowany wzorem:

$$\hat{f}(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\Big(\frac{x - x_i}{h}\Big),$$

gdzie dodatni współczynnik h określa się mianem parametru wygładzania, a funkcja K jest mierzalną, symetryczną względem zera oraz posiadającą w tym punkcie słabe maksimum globalne funkcją $K : \mathbb{R} \to [0, \infty)$ spełniającą warunek $\int_{\mathbb{R}} K(x)dx = 1$ i jest ona nazywana jądrem.

W R do wyznaczania estymatora jądrowego istnieje funkcja `density`, która przyjmuje rozkład próby losowej oraz wartość `bw` (*smoothing bandwith*), czyli współczynnik wygładzenia (w poprzednim akapicie oznaczony jako h). Do jego narysowania można zastosować funkcji `lines`.

Do tego należy nakreślić wykres gęstości teoretycznej rozkładu normalnego. Podobnie jak w zadaniu 1. można do tego użyć funkcji `curve`, tyle że tym razem z funkcją `dnorm` dającą rozkład gęstości, a nie dystrybuantę.



Rysunek 6: Histogram losowej próby wraz z estymatorem jądrowym (`bw` $= 0.2$, na pomarańczowo) oraz teoretyczną gęstością rozkładu normalnego (na niebiesko).

Pełny kod rozwiązania:

```
# zad 2.2

mu = 0
sigma = 1
N = 1000
X = rnorm(N, mean=mu, sd=sigma)

(h = hist(X, br=20, freq=FALSE, ylim=c(0,0.5), main='Histogram losowej próby rozkładu
    normalnego standardowego', ylab='', xlab=''))
lines(density(X, bw=0.2), col='orange', lwd=2)
curve(dnorm(x, mu, sigma), add=TRUE, col='mediumblue', lwd=2)
```

Zadanie 2.3. Sporządzić wykresy funkcji prawdopodobieństwa następujących rozkładów dwumianowych: `binom(10, 0.5)`, `binom(10, 0.25)`, `binom(50, 0.25)`. Wyciągnąć wnioski.

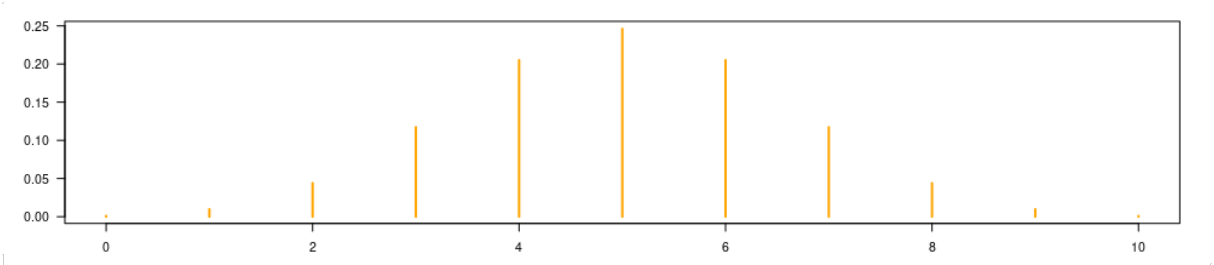
Rozwiązanie: W celu uzyskania trzech podanych rozkładów dwumianowych należy wykonać funkcję `dbinom` z podanymi w poleceniu parametrami.

Do narysowania wykresów tych rozkładów stosujemy funkcję `plot`. W celu uzyskania kilku wykresów jeden nad drugim przed użyciem plotów wywołujemy funkcję `par` z argumentem `mfrow` ustawionym na wartość dwuwymiarowego wektora $(3, 1)$.

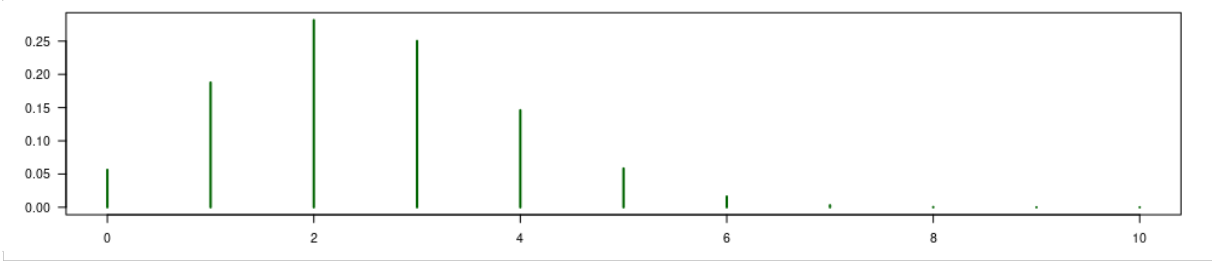
Jak można zauważyć na rysunku [9] wykres przypomina swoim wyglądem gęstość rozkładu normalnego. Można porównać go z tą funkcją rysując dodatkową krzywą na wykresie rozkładu dwumianowego korzystając z `curve`.

By otrzymać tą krzywą możemy obliczyć wartość oczekiwaną i odchylenie standardowe rozkładu dwumianowego i wziąć je jako te dla rozkładu normalnego. Wartość średnia rozkładu dwumianowego Z jest równa $EZ = np$, a odchylenie $\sigma_Z = \sqrt{np(1-p)}$, gdzie n jest liczbą elementów w rozkładzie, a p prawdopodobieństwem sukcesu.

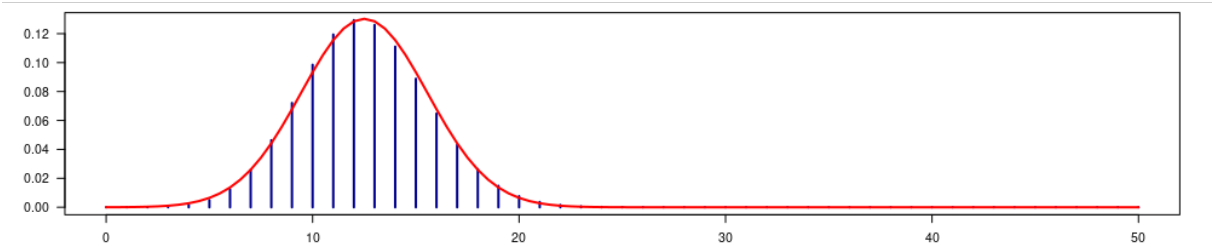
Dla naszego wykresu obliczenia wyglądają następująco: $\mu = 50 \cdot 0.25 = \frac{25}{2}$; $\sigma = \sqrt{50 \cdot 0.25 \cdot (1 - 0.25)} = \sqrt{\frac{150}{4}}$



Rysunek 7: Wykres funkcji prawdopodobieństwa rozkładu dwumianowego `binom(10, 0.5)`



Rysunek 8: Wykres funkcji prawdopodobieństwa rozkładu dwumianowego `binom(10, 0.25)`



Rysunek 9: Wykres funkcji prawdopodobieństwa rozkładu dwumianowego `binom(50, 0.25)` wraz z funkcją gęstości rozkładu normalnego $N(\frac{25}{2}, \frac{150}{4})$

Pełny kod rozwiązania:

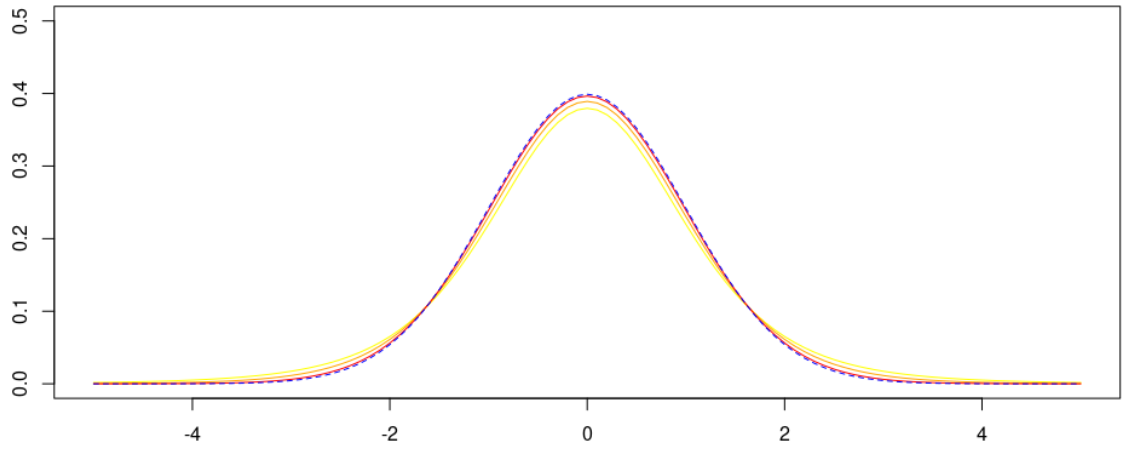
```
# zad 2.3

x = 0:10
y = 0:50

den_X = dbinom(x, 10, 0.5)
den_Y = dbinom(x, 10, 0.25)
den_Z = dbinom(y, 50, 0.25)

EZ = 50 * 0.25
VarZ = 50 * 0.25 * (1 - 0.25)

den_ZN = dnorm(y, mean=EZ, sd=sqrt(VarZ))
par(mfrow=c(3,1))
plot(x, den_X, pch=19, type='h', lwd=2, las=1, col='orange', lty=1, xlab='', ylab='',
    main='n=10, p=0.5')
plot(x, den_Y, pch=19, type='h', lwd=2, las=1, col='darkgreen', lty=1, xlab='', ylab='',
    main='n=10, p=0.25')
plot(y, den_Z, pch=19, type='h', lwd=2, las=1, col='darkblue', lty=1, xlab='', ylab='',
    main='n=50, p=0.25')
curve(dnorm(x, mean=EZ, sd=sqrt(VarZ)), add=TRUE, col='red', lwd=2, xlab='', ylab='')
```

Zadanie 2.4. Utworzyć wykresy gęstości zmiennych losowych o rozkładzie t-Studenta o 5, 10 oraz 40 stopniach swobody. Przeanalizować, jak zmienia się gęstość rozkładu t-Studenta wraz ze wzrostem liczby stopni swobody.

Rozwiązanie: W celu wygenerowania gęstości zmiennych losowych rozkładu t-Studenta w R istnieje funkcja `dt`. Przyjmuje ona argument `df`, który określa stopnie swobody rozkładu.

Aby narysować te rozkłady stosujemy funkcję `curve`. By uzyskać kilka wykresów na jednym wykresie należy ustawić argument `add` na wartość `TRUE`.

Wraz ze wzrostem stopni swobody rozkład t-Studenta zdaje się dążyć do rozkładu normalnego. Zaiste, faktycznie dla dużych ν t_ν ma w przybliżeniu rozkład $N(0,1)$. W celu porównania na wykres naniesiono niebieską, przerywaną linię reprezentującą rozkład normalny za pomocą `curve` z `dnorm`. Rzeczywiście, im większa liczba stopni, tym wykres bardziej się pokrywa z wykresem rozkładu normalnego.



Rysunek 10: Rozkłady t-Studenta o s stopniach swobody. Żółty: $s = 5$, pomarańczowy: $s = 10$, czerwony: $s = 40$.

Pełny kod rozwiązania:

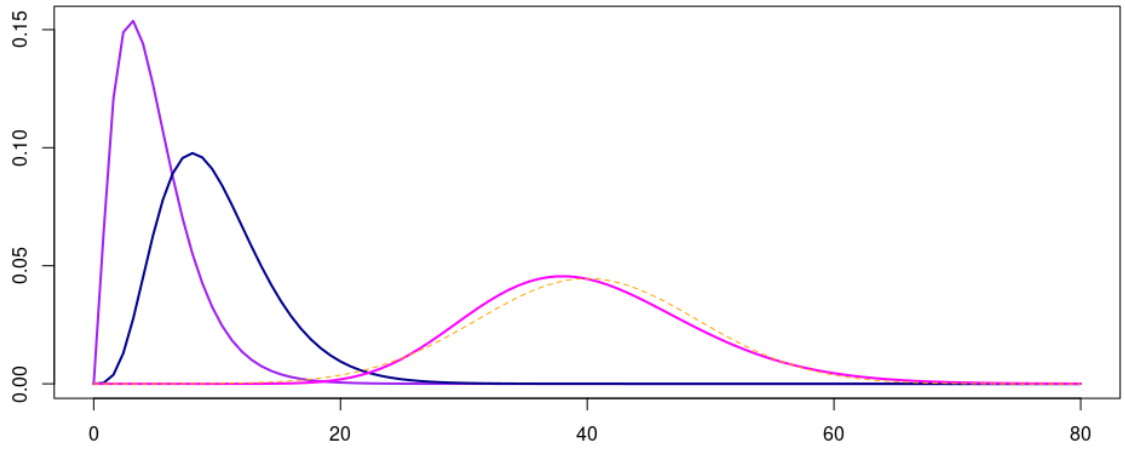
```
# zad 2.4

curve(dt(x, df=5), xlim=c(-5,5), ylim=c(0,0.5), col='yellow')
curve(dt(x, df=10), xlim=c(-5,5), ylim=c(0,0.5), col='orange', add=TRUE)
curve(dt(x, df=40), xlim=c(-5,5), ylim=c(0,0.5), col='red', add=TRUE)
curve(dnorm(x, mean=0, sd=1), xlim=c(-5,5), ylim=c(0,0.5), col='blue', add=TRUE, lty=2)
```

Zadanie 2.5. Utworzyć wykresy gęstości zmiennych losowych o rozkładzie chi-kwadrat o 5, 10 oraz 40 stopniach swobody. Przeanalizować, jak zmienia się gęstość rozkładu chi-kwadrat wraz ze wzrostem liczby stopni swobody.

Rozwiązanie: W celu wygenerowania gęstości zmiennych losowych rozkładu chi-kwadrat w R istnieje funkcja `dchisq`. Przyjmuje ona argument `df`, który określa stopnie swobody rozkładu.

Aby narysować te rozkłady tak samo jak w zadaniu 4 stosujemy funkcję `curve` z argumentem `add` ustawionym na wartość `TRUE`.

Na rysunku [11] można zauważyć, że wraz ze wzrostem stopni swobody wykres staje się coraz bardziej podobny do wykresu rozkładu normalnego. Rzeczywiście, dla dużych p χ_p ma w przybliżeniu rozkład $N(p, 2p)$. Rysujemy zatem pomarańczową, przerywaną linią rozkład normalny dla największej liczby stopni swobody, jakie mamy, czyli 40.



Rysunek 11: Rozkłady chi-kwadrat o s stopniach swobody. Na fioletowo: $s = 5$, niebiesko: $s = 10$, różowo: $s = 40$.

Pełny kod rozwiązania:

```
# zad 2.5

curve(dchisq(x, df=5), xlim=c(0,80), col='purple', lwd=2, xlab='', ylab='')
curve(dchisq(x, df=10), xlim=c(0,80), col='darkblue', lwd=2, add=TRUE)
curve(dchisq(x, df=40), xlim=c(0,80), col='magenta', lwd=2, add=TRUE)
curve(dnorm(x, mean=40, sd=sqrt(2*40)), xlim=c(0, 80), ylim=c(0,0.5), col='orange', add=
    TRUE, lty=2)
```

Zadanie 2.6. Zbiór `Cars93`, znajdujący się w bibliotece `MASS`, zawiera dane dotyczące różnych modeli samochodów osobowych.

a) Utworzyć nową zmienną o nazwie `zp.m` opisującą zużycie paliwa (mierzone w litrach na 100 km) podczas jazdy samochodu w mieście. Przyjąć, że 1 mila to 1.6 km; 1 galon amerykański to 3.8 litra. Odpowiednie dane wyrażone w milach na galon znajdują się w zmiennej `MPG.city`.

b) Wyznaczyć podstawowe statystyki próbkowe dla danych w zmiennej `zp.m`. Obliczyć kwantyl rzędu 0.95 dla tych danych i podać jego interpretację.

c) Sporządzić wykresy skrzynkowe dla zmiennej `zp.m` osobno dla samochodów amerykańskich i nieamerykańskich. Powtórzyć to samo dla zmiennej `MPG.city`.

d) Narysować wykres słupkowy i kołowy dla zmiennej `Type`. Ile spośród badanych samochodów zaliczono do kategorii sportowe?
Wskazówka: Aby uzyskać liczności poszczególnych grup użyć funkcji `table()`.

Rozwiązanie: Na początku ładujemy bibliotekę `MASS` za pomocą wcześniej poznanej funkcji `library`. Używając operatora `$` „doklejamy" zmienną `zp.m` do naszej zmiennej `x`. Kwantyl rzędu 0.95 liczymy za pomocą funkcji `quantile` - jego interpretacja to jakie jest minimalne zużycie paliwa dla 5% najbardziej ekonomicznych aut. Do zilustrowania danych używamy funkcji `boxplot` - stworzy ona wykres pudełkowy.

Zwraca ona listę z wartościami pozwalającymi przeanalizować sporządzony wykres:

- `stats` - każda kolumna zawiera: granicę dolną dolnego wąsa, dół pudełka, medianę, górę pudełka oraz granicę górną górnego wąsa
- `n` - wektor z obserwacjami (różnymi od NA) z danej grupy
- `conf` - macierz z granicami pudełek
- `out` - macierz, w której każda kolumna zawiera obserwacje odstające (`outliery`)
- `group` - wektor, który mówi do której grupy należą obserwacje odstające
- `names` - nazwy grup

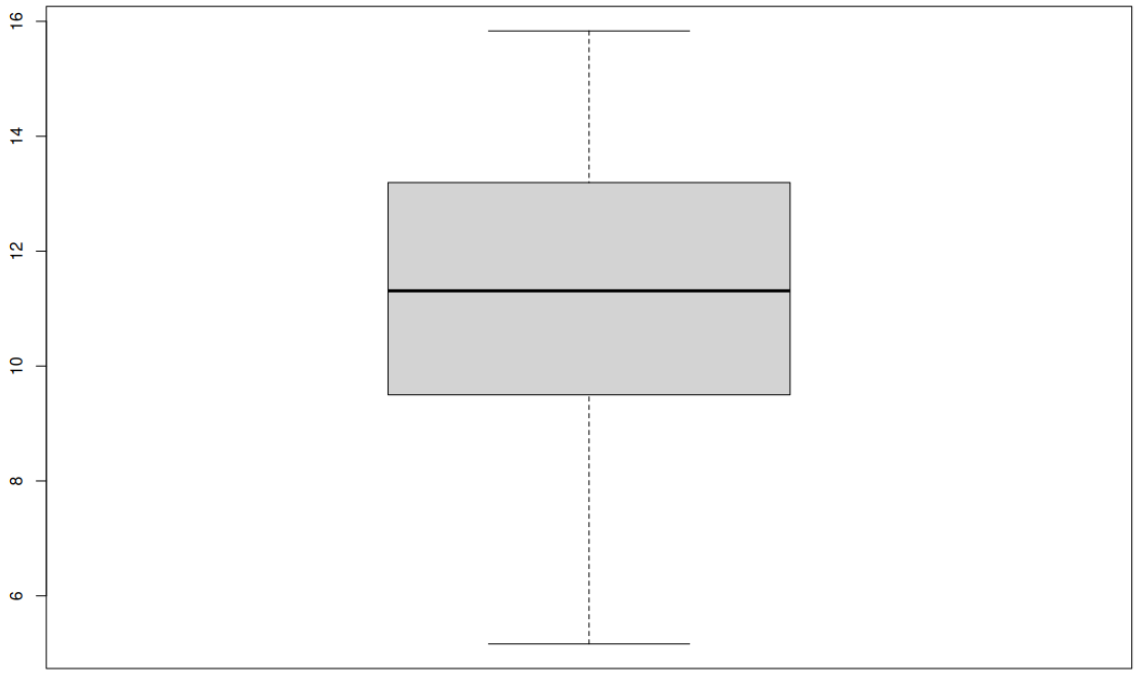
Wykres pudełkowy (przedstawiony poziomo) tworzy się odkładając na poziomej osi wartości niektórych parametrów rozkładu. Nad osią umieszczony jest prostokąt (pudełko), którego lewy bok jest wyznaczony przez pierwszy kwartyl, zaś prawy bok przez trzeci kwartyl. Szerokość pudełka odpowiada wartości rozstępu ćwiartkowego. Wewnątrz prostokąta znajduje się pionowa linia, określająca wartość mediany.

Po prawej i lewej stronie znajdują się odcinki zwane wąsami. Wąsy mają długość ostatniej wartości wewnątrz półtorej wartości rozstępu międzykwartylowego, zaś wartości leżące poza tym zakresem są reprezentowane przez punkty.

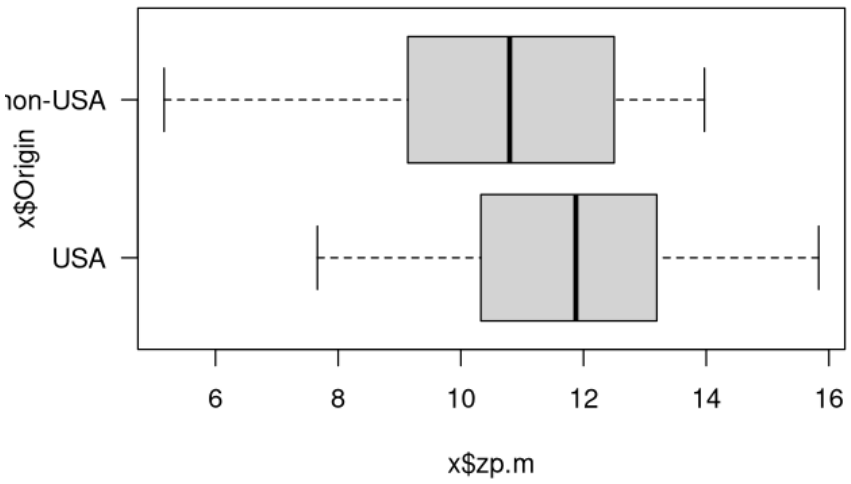
Następne 2 wywołania funkcji `boxplot` obrazują te same dane z podziałem na pochodzenie samochodów - służy do tego operator tyldy, który powoduje podział danych względem wszystkich kategorii typu czynnikowego `x$Origin` (typ czynnikowy to typ, który może przyjąć jedną z predefiniowanych wartości, w tym przypadku „USA" i „non-USA").

Funkcja ta wyznacza tablicę liczebności dla jednej, dwóch lub większej liczby zmiennych wyliczeniowych. W przypadku zmiennych jakościowych podobny efekt co funkcja `table()` ma funkcja `summary()`. Różnica polega na tym, że w razie występowania danych `NA` funkcja `table()` je ignoruje, a funkcja `summary()` wypisuje ich liczbę.

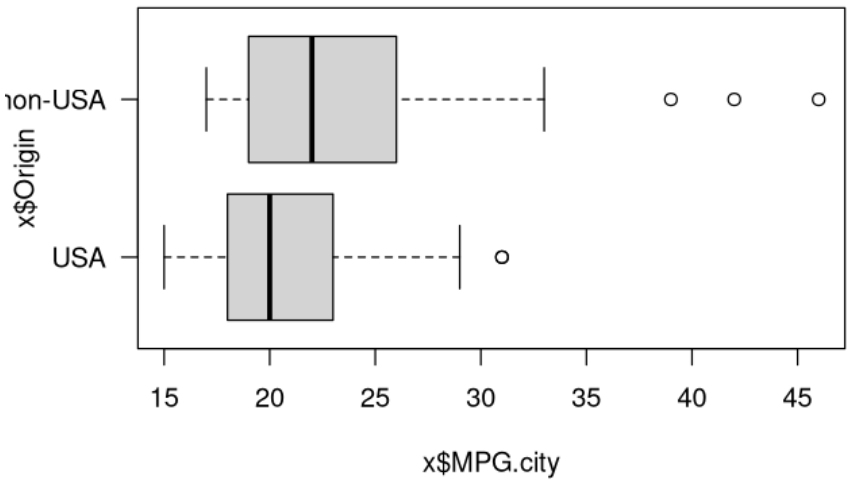
Dodatkowo możemy użyć `sort` aby posortować wartości, domyślnie rosnąco. Do narysowania wykresu słupkowego służy funkcja `barplot`, a kołowego `pie`.



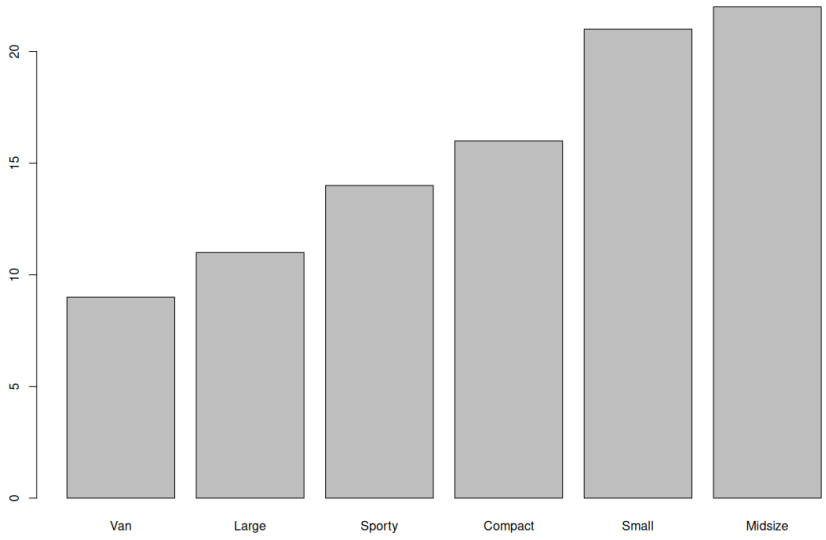
Rysunek 12: Wykres pudełkowy zużycia paliwa (w litrach na 100 km) wszystkich badanych aut



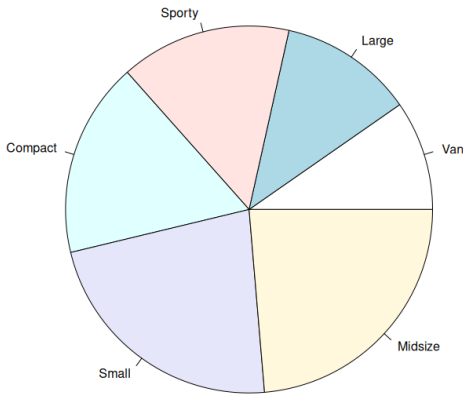
Rysunek 13: Wykres pudełkowy zużycia paliwa (w litrach na 100 km) wszystkich badanych aut z podziałem na pochodzenie aut



Rysunek 14: Wykres pudełkowy zużycia paliwa (w milach na galon) wszystkich badanych aut z podziałem na pochodzenie aut



Rysunek 15: Wykres słupkowy liczebności aut z poszczególnych kategorii



Rysunek 16: Wykres kołowy liczebności aut z poszczególnych kategorii

Pełny kod rozwiązania:

```
rm(list = ls())

library(MASS)
x = Cars93
dim(x)
head(x)
summary(x)

zp.m = 380/1.6/x$MPG.c
x$zp.m = 380/1.6/x$MPG.city

quantile(x$zp.m, 0.95)

boxplot(x$zp.m)

b = boxplot(x$zp.m~x$Origin, las=1, horizontal = TRUE)
b1 = boxplot(x$MPG.city~x$Origin, las=1, horizontal = TRUE)

t = table(x$Type)
t = sort(t)
barplot(t)

pie(t)
```

Zadanie 2.7. Wygenerować 10000 prób 10-elementowych z rozkładu normalnego. Następnie zakładając, iż o próbach wiemy tylko tyle, że pochodzą one z rozkładu normalnego o nieznanych parametrach, wyznaczyć dla każdej próby przedział ufności dla wartości oczekiwanej na poziomie ufności 0.95. Porównać frakcję pokryć przez przedziały ufności faktycznej wartości oczekiwanej z założonym poziomem ufności.

Rozwiązanie: W poniższym kodzie przedstawiono 2 sposoby rozwiązania tego zadania. W pierwszym, używamy `for`, w którym dla każdego `i` generujemy 10-elementową próbę z rozkładu normalnego używając funkcji `rnorm`, a następnie używamy `t.test`. Jednym z wyników tej funkcji jest pole `$conf`, które przechowuje lewy i prawy kraniec przedziału ufności. Jeżeli wartość oczekiwana znajduje się w przedziale ufności to zwiększamy zmienną `counter`. W drugim robimy to samo używając funkcji `replicate`, która jest znacząco szybsze. Funkcja `replicate` zwróci nam wektor wartości logicznych. Liczbę `TRUE` można policzyć poprzez użycie `sum` - traktuje ona `TRUE` jako 1, a `FALSE` jako 0.

Oba sposoby skutkują uzyskaniem wyniku 0.9523, co oznacza że około w 95.23% przypadków wartość oczekiwana μ trafiła do przedziału ufności. Jest to w dużej mierze zgodne z oczekiwanym wynikiem 95%.

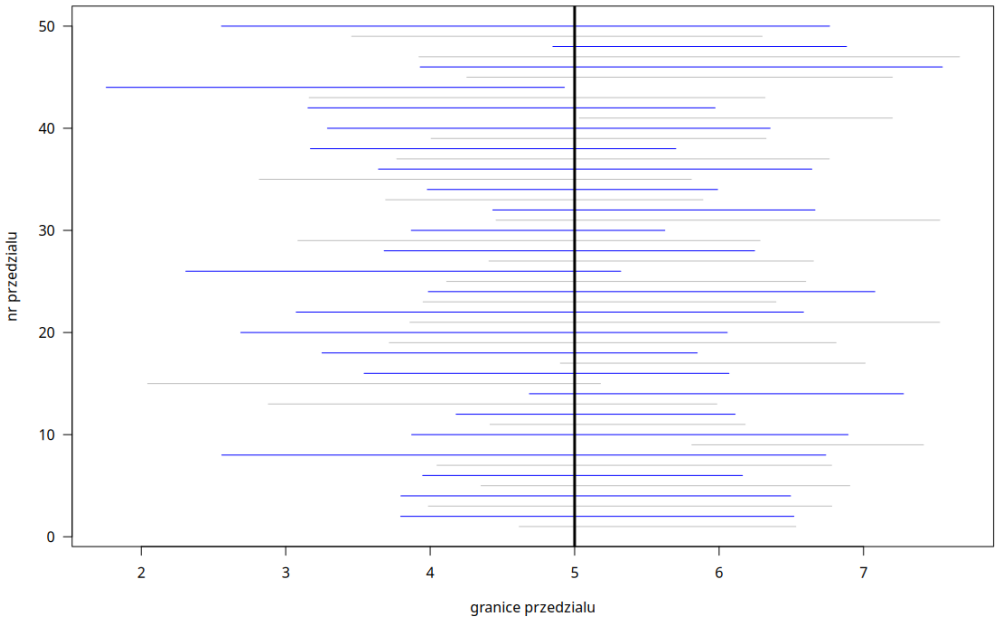
Pełny kod rozwiązania:

```
rm(list = ls())

N = 10^4
n = 10
mi = 5
sigma = 2
alpha = 0.05

counter = 0
for (i in 1:N) {
  x = rnorm(n, mean=mi, sd=sigma)
  range = t.test(x, conf.level=1-alpha)$conf
  if(mi < range[2] && mi > range[1]){
    counter = counter + 1
  }
}
result = counter/N

# alternative - use of replicate
hits = replicate(N,
                    {
                      x = rnorm(n, mean=mi, sd=sigma)
                      range = t.test(x, conf.level=1-alpha)$conf
                      mi < range[2] && mi > range[1]
                    })
result2 = sum(hits)/N
```

Zadanie 2.8. Wybrać $\mu \in \mathbb{R}$, $\sigma > 0$. Wygenerować $N = 50$ prób n-elementowych ($n = 10$) z rozkładu $N(\mu, \sigma)$ i dla każdej z nich utworzyć przedział ufności dla μ na poziomie ufności 0.95. Przedstawić na jednym wykresie uzyskane przedziały ufności. Ile z nich powinno zawierać μ?

Rozwiązanie: W tym zadaniu również używamy funkcji `replicate`, tym razem jednak zwróci ona wektor N przedziałów ufności dla 10-elementowych prób z rozkładu $N(\mu = 5, \sigma = 2)$. Warto zauważyć w jaki sposób wybieramy te wartości przedziałów - używamy operatora `$` oraz nazwy `conf` - odwoła się ona do `conf.int` - nie trzeba pisać pełnej nazwy komponentu, o ile `R` jest w stanie jednoznacznie stwierdzić o jaki komponent nam chodzi.

Dalej wizualizujemy wyniki przy pomocy funkcji `matplot`. Jako pierwszy argument (oś X) ustawiamy macierz $2 \times n$ pochodzącą z `result` (pierwszy wiersz to wektor początków przedziałów ufności, a drugi to wektor końców). Na osi Y odkładamy wektory $1 : N$. W ten sposób otrzymujemy poniższy wykres (kolory odcinków reprezentujących przedziały są naprzemienne dla czytelności - za co odpowiada ustawienie koloru (`col`) jako dwuelementowego wektora `('grey', 'blue')`):



Pionowa czarna linia odpowiada wartości $\mu = 5$. Spodziewamy się więc, że mniej-więcej 95% widocznych na wykresie poziomych odcinków przetnie się z tą linią. Patrząc na rysunek, widzimy, że 3 przedziały spośród 50 wygenerowanych nie zawierają wartości oczekiwanej, czyli 94% ją zawiera.

Pełny kod rozwiązania:

```
rm(list = ls())

N = 5e1
n = 1e1
mi = 5
sigma = 2
alpha = 5e-2

result = replicate(N,
                   {
                      x = rnorm(n, mi, sigma)
                      t.test(x, conf.level = 1-alpha)$conf
                   })
matplot(result, rbind(1:N, 1:N), type='l', lty=1,
        col=c('grey','blue'), las=1,
        xlab="granice przedzialu",
        ylab="nr przedzialu")
abline(v=mi, col=2)
```

Uwagi

- Opis wykresu pudełkowego wzięty z: https://pl.wikipedia.org/wiki/Wykres_pude%C5%82kowy
- Do generowania sekwencji można użyć funkcji `seq`, która jako parametry przyjmuje granice przedziału oraz opcjonalne argument `by` (krok) oraz `length.out` (pożądana długość sekwencji).

3. Estymacja przedziałowa, testy parametryczne dla jednej populacji

Zadanie 3.1. Ornitolog, badający określony gatunek, pobrał próbę losową 10 dorosłych ptaków i zmierzył ich wagę, otrzymując następujące wyniki (w kg): 5.21, 5.15, 5.20, 5.48, 5.19, 5.25, 5.09, 5.17, 4.94, 5.11. Zakładamy, że waga ptaków badanego gatunku ma rozkład normalny.

a) Utworzyć 95% przedział ufności dla średniej wagi ptaków badanego gatunku.

b) Czy na poziomie istotności 0,05 można stwierdzić, że średnia waga ptaków badanego gatunku wynosi 5,15 kg?

c) Czy na poziomie istotności 0,05 można stwierdzić, że średnia waga ptaków badanego gatunku jest mniejsza niż 5,20 kg?

d) Z jakim prawdopodobieństwem test, przeprowadzony w pkt. c), przyjmie na poziomie istotności 0,05 hipotezę, że średnia waga ptaków badanego gatunku jest mniejsza niż 5,20 kg, w sytuacji, gdy w rzeczywistości ta średnia waga wynosi 5,15 kg?

e) Ile by musiała wynosić średnia waga ptaków tego gatunku, by test z pkt. c) z prawdopodobieństwem 0,8, na poziomie istotności 0,05, przyjmował hipotezę, że średnia waga jest mniejsza niż 5,20 kg?

f) Załóżmy, że rzeczywista średnia waga ptaków jest równa 5,15 kg. Wyznaczyć minimalną liczność próby, która zagwarantuje, że test na poziomie istotności 0,05, z prawdopodobieństwem nie mniejszym niż 0,8, będzie przyjmował hipotezę, że średnia waga jest mniejsza niż 5,20 kg.

g) Utworzyć 95% przedział ufności dla wariancji wagi ptaków badanego gatunku.

h) Utworzyć 95% przedział ufności dla odchylenia standardowego wagi ptaków badanego gatunku.

i) Czy na poziomie istotności 0,05 można stwierdzić, że odchylenie standardowe wagi ptaków badanego gatunku wynosi 0,20 kg?

Rozwiązanie: Wektor `birds` tworzymy poprzez skopiowanie wartości wag z treści zadania i połączenie ich za pomocą funkcji `c`.

a) Przedział ufności dla średniej wagi tworzymy korzystając z funkcji `t.test` i podobnie jak na poprzednich laboratoriach przedział ufności pobieramy za pomocą `$conf`. Otrzymujemy $[5.08; 5.28]$.

b) Ponownie używamy funkcji `t.test`. Hipotezą zerową jest $\mu = \mu_0 = 5.15$, a alternatywną $\mu \neq \mu_0$. Jako argumenty przekazujemy wektor wartości, wartość średniej oraz hipotezę alternatywną - w tym przypadku `two.sided`, ponieważ w naszym przypadku hipoteza alternatywna nie rozróżnia, czy waga jest mniejsza, czy większa niż μ_0.

   Sytuacja w tym podpunkcie wpisuje się w model II (zmienna losowa dana rozkładem normalnym, nieznane μ, σ i hipoteza zerowa $\mu = \mu_0$). Za pomocą funkcji `t.test` wyznaczamy p-value, które wynosi około 0.5188 - więcej niż przyjęty poziom istotności, czyli nie ma podstaw do odrzucenia hipotezy zerowej.

c) Nadal korzystamy z modelu II, tym razem hipotezą alternatywną jest $\mu \geq 5.2$. Zmieniamy zatem parametr funkcji `t.test` na `less` i ponownie sprawdzamy `p-value`. Otrzymujemy `p-value` równe około 0.7, czyli zdecydowanie więcej niż α - więc nie ma podstaw do odrzucenia hipotezy zerowej.

d) W tym podpunkcie chcemy obliczyć wartość

$$\mathrm{P}(\text{odrzucamy } H_0 \,|\, \mu = 5.15).$$

   Używamy do tego `power.t.test` (obliczamy moc testu), jako argumenty podajemy kolejno: liczbę obserwacji, różnicę średnich ($\delta = 5.2 - 5.15$), typ testu, hipotezę alternatywną taką samą jak w

poprzednim podpunkcie oraz poziom istotności równy 0.05. Uzyskane prawdopodobieństwo wynosi około 0.28.

e) Korzystamy jeszcze raz z funkcji `power.t.test`, z taką różnicą, że tym razem dana jest moc testu równa 0.8, natomiast nieznana jest średnia waga ptaków. Jako parametry funkcji wstawiamy zatem znaną wartość `power`, a w jednym z pól wyniku otrzymujemy $\delta \approx 0.12$, czyli rzeczywista średnia waga musiałaby wynosić $\mu_0 - \delta \approx 5.08$ kg.

f) Podpunkt ten rozwiązujemy analogiczne do poprzedniego, tym razem dana jest moc testu i $\delta = 5.2 - 5.15 = 0.05$, a szukamy liczności próby n. Obliczenia zwracają $n \approx 47.51533$, zatem minimalna potrzebna liczba obserwacji wynosi 48.

g) Do wyznaczenia przedziału ufności dla wariancji danej populacji korzystamy z funkcji `sigma.test` z biblioteki `TeachingDemos`. Argumenty są podobne jak w przypadku użytej wcześniej funkcji `t.test`. Wyznaczony przedział to [0.00882574; 0.06217251].

h) Aby obliczyć 95% przedział ufności odchylenia standardowego wagi ptaków należy wziąć pierwiastki z obu krańców wyznaczonego w poprzednim punkcie przedziału ufności dla wariancji - wynik to [0.09394541; 0.24934416].

i) Hipoteza zerowa: $\sigma = 0.2$, alternatywna: $\sigma \neq 0.2$. Wykonujemy `sigma.test` z parametrem $\sigma = 0.2$. Uzyskane p-value wynosi 0.2041 - więcej niż $\alpha = 0.05$, więc nie mamy podstaw do odrzucenia hipotezy zerowej.

Pełny kod rozwiązania:

```
rm(list = ls())

alpha = 0.05

birds = c(5.21, 5.15, 5.20, 5.48, 5.19, 5.25, 5.09, 5.17, 4.94, 5.11)

range = t.test(birds, conf.level=1-alpha)$conf

t = t.test(birds, mu=5.15, alt="two.sided")
print(t$p.value)


t = t.test(birds, mu=5.2, alt="less")
print(t$p.value)

power.t.test(n=length(birds), delta=5e-2, sd=sd(birds),
             type="one.sample", alternative="one.sided", sig.level=alpha)$power


power.t.test(n=length(birds), power=8e-1, sd=sd(birds), type="one.sample", alternative="
    one.sided", sig.level=alpha)

power.t.test(delta=5e-2, power=8e-1, sd=sd(birds), type="one.sample", alternative="one.
    sided", sig.level=alpha)

library(TeachingDemos)
sigma.test(birds, conf.level=1-alpha)$conf
sqrt(sigma.test(birds, conf.level=1-alpha)$conf)
sigma.test(birds, sigma=2e-1)
```

Zadanie 3.2. W kolumnie `WeightInitial` w pliku `goats.txt` zapisano wagę (w kg) losowo wybranych młodych kóz hodowanych w Australii. Wiadomo, że rozkład badanej cechy jest normalny.

a) Utworzyć 95% przedział ufności dla wartości oczekiwanej wagi młodych kóz hodowanych w Australii.

b) Na poziomie istotności 0,05 przetestować hipotezę, że średnia waga młodych kóz hodowanych w Australii przekracza 23 kg.

c) Zakładając, że rzeczywista średnia waga młodych kóz hodowanych w Australii wynosi 24 kg, wyznaczyć prawdopodobieństwo, że przeprowadzając test na poziomie istotności 0,05 i na podstawie 40 obserwacji, błędnie uznamy, że średnia waga takich kóz nie przekracza 23 kg.

d) Załóżmy, że rzeczywista średnia waga młodych kóz hodowanych w Australii wynosi 24 kg. Ile trzeba by zebrać pomiarów wag takich kóz, by test (przeprowadzony na poziomie istotności 0,05) wykrywał, z prawdopodobieństwem nie mniejszym niż 0,8, że średnia waga takich kóz przekracza 23 kg?

e) Utworzyć 90% przedział ufności dla wariancji wagi młodych kóz hodowanych w Australii.

f) Czy można przyjąć, że wariancja wagi młodych kóz hodowanych w Australii wynosi 20 kg^2? Zweryfikować odpowiednią hipotezę na poziomie istotności 0,1.

g) Na poziomie istotności 0,1 zweryfikować hipotezę, że odchylenie standardowe wagi młodych kóz hodowanych w Australii przekracza 3 kg.

Rozwiązanie: Zbiór danych `goats.txt` (ze strony Pani Prof. Dembińskiej: Strona główna > Dydaktyka > Materiały dydaktyczne > Zbiory danych) ładujemy do R używając polecenia:

```
goats <- read.csv("ścieżka/do/pliku/goats.txt", sep="")
```

a) Na podstawie informacji z polecenia możemy wywnioskować, iż mamy do czynienia z **Modelem II** (rozkład normalny, σ i μ nieznane). Do wyznaczenia przedziału ufności używamy zatem funkcji `t.test`, podając jako argumenty kolumnę **WeightInitial** ze zbioru goats (`goats$WeightInitial`) oraz parametr `conf.level` równy 0.95. Aby wydobyć z wyniku działania tej funkcji jedynie przedział ufności, należy na końcu dopisać `$conf.int` (confidence interval).

b) Model zdefiniowaliśmy już w poprzednim podpunkcie. Przyjmujemy następujące hipotezy (oczywiście dla parametru μ):

$$H_0 : \mu = 23$$
$$H_1 : \mu > 23$$

Następnie używamy funkcji `t.test` (jak wcześniej) tylko zamiast podając argument `conf.level` podajemy `mu=23` oraz `alternative='greater'` (zgodnie z przyjętą hipotezą zerową i alternatywną). Aby wydobyć potrzebne nam p-value, na końcu dopisujemy `$p.value`. Przypomnijmy:

$$\begin{cases} \text{p-value} \leq \alpha : \text{Hipotezę zerową odrzucamy} \\ \text{p-value} > \alpha : \text{Nie ma podstaw do odrzucenia hipotezy zerowej} \end{cases}$$

Odczytujemy wyniki: p-value $\approx 0.39 > \alpha$ - zatem brak podstaw do odrzucenia hipotezy zerowej, więc brak podstaw do przyjęcia hipotezy, że średnia waga młodych kóz hodowlanych w Australii przekracza 23 kg.

c) Chcemy wyznaczyć:

$$P(\text{przyjmiemy } H_0 | \mu = 24) = 1 - P(\text{odrzucimy } H_0 | \mu = 24) = 1 - \text{moc testu}(24)$$

Użyjemy zatem funkcji `power.t.test`, która wyznacza właśnie wartość funkcji mocy testu (lub innych parametrów, w zależności który argument wejściowy pominiemy). W naszym przypadku musimy podać następujące parametry:

- `n = 40` - liczność próby (zgodnie z poleceniem)
- `detla = 1` - jest to różnica między hipotezą zerową (23kg) a wartością rzeczywistą (24kg)
- `sd = sd(goats$WeightInitial)` - odchylenie standardowe wagi, wyznaczamy je empirycznie (gdyż nie znamy dokładnej wartości σ) MUSIMY PODAĆ TEN ARGUMENT, GDYŻ DOMYŚLNIE JEST USTAWIONY NA WARTOŚĆ 1
- `type='one.sample'` - gdyż rozważamy testy dla jednej populacji
- `alternative='greater'` - hipotezą alternatywną jest $\mu > 23$
- `sig.level=0.05` - poziom istotności testu (nie trzeba było podawać, gdyż domyślnie jest ustawiony na 0.05)

Parametr, który pozostawiamy pusty to `power`, więc funkcja nam to wyznaczy. Aby wydobyć moc testu należy na końcu wywołania funkcji dopisać `$power`, oraz oczywiście wynik ten odejmujemy od jedynki (zgodnie ze wzorem powyżej). Odczytujemy wynik: 0.4460559.

d) Chcemy zbadać dla jakiego `n` moc testu dla wartości $\mu = 24$ jest większa niż 0.8. Wykorzystamy do tego ponownie funkcję `power.t.test` z argumentami jak wcześniej, zamieniając jedynie `n = 40` na `power = 0.8`. Skoro zostawiamy puste `n`, to funkcja właśnie tą wielkość nam wyznaczy - liczność próby taka, aby moc testu była równa 0.8. Aby wydobyć `n`, dopisujemy na końcu `$n`. Odczytujemy wynik: `76.68731`. Zatem potrzebaby zebrać przynajmniej 77 pomiarów.

e) Przechodząc do rozważań na temat wariancji cechy potrzebujemy załadować bibliotekę **TeachingDemos** poleceniem: `library(TeachingDemos)`. Zawiera ona potrzebną nam funkcję `sigma.test`. Jej wywołanie jest bardzo podobne do `t.test`, jako argumenty podajemy jak wcześniej kolumnę **WeightInitial** oraz poziom istotności `conf.level=0.9`. Aby wydobyć przedział ufności analogicznie dopisujemy na końcu `$conf.int$`. Wynik: $(8.705893, 18.489698)$.

f) Będziemy testować następujące hipotezy:

$$H_0 : \sigma = 20$$
$$H_1 : \sigma \neq 20$$

Wykorzystamy tą samą funkcję co w poprzednim podpunkcie, jednak nieco inaczej będzie wyglądać jej wywołanie. Podajemy następujące argumenty:

- `x = goats$WeightInitial` - dane z próby losowej
- `sigma=sqrt(20)` - odchylenie standardowe (KONIECZNIE PIERWIASTEK, BO BADAMY WARIANCJĘ 20 A DO FUNKCJI PODAJEMY ODCHYLENIE; można również podać wariancję przypisując ją do argumentu `sigmasq=20`)
- `alternative='two.sided'` - hipoteza alternatywna to $\neq$, dlatego `two.sided`

Odczytujemy p-value: 0.0519. Jest mniejsze od poziomu istotności testu, zatem hipotezę zerową odrzucamy.

g) Będziemy postępować analogicznie jak w poprzednim podpunkcie, zmieniając:

- `sigma = 3` - badamy odchylenie standardowe równe 3
- `alternative='greater'` - hipoteza alternatywna to $\sigma > 3$

Po wywołaniu funkcji `sigma.test` odczytujemy p-value: 0.06925. Jest ono mniejsze od poziomu istotności, zatem hipotezę zerową odrzucamy i przyjmujemy hipotezę, że waga mlodych kóz hodowlanych w Australii przekracza 3kg.

Pełny kod rozwiązania:

```
goats <- read.csv("C:/Users/Dominik Zieliński/Downloads/goats.txt", sep="")

goats

# Zadanie 2
# a)

t.test(goats$WeightInitial, conf.level=0.95)

# b)
# Model 2
# H0 - mi = 23
# H1 mi > 23

t.test(goats$WeightInitial, mu=23, alternative ="greater")

# c)

alpha = 0.05

1 - power.t.test(n=length(goats$WeightInitial), delta=1, sd=sd(goats$WeightInitial),
             type="one.sample", alternative="one.sided", sig.level=alpha, )$power

# d)
power.t.test(delta=1, sd=sd(goats$WeightInitial),
                 type="one.sample", alternative="one.sided", sig.level=alpha, power =
    0.8)

library(TeachingDemos)
# e
# conf range for variance
sigma.test(goats$WeightInitial, conf.level=0.9)$conf

# f
# H0 -> sigma = 2e1
# H1 -> sigma != 2e1
sigma.test(goats$WeightInitial, sigma=sqrt(2e1))
# pv < 0.1 = alpha - odrzucamy
# we rejest H0

# g
sigma.test(goats$WeightInitial, sigma=3, alternative='greater')
power.t.test()
```

Zadanie 3.3. Pełnomocnik rządu Alfalandii ds. równego statusu kobiet i mężczyzn podejrzewa, że udział mężczyzn wśród pracowników przedszkoli jest niższy niż minimum przewidziane w ustawie wynoszące 35%.

a) Czy na poziomie istotności 0,05 można uznać to stwierdzenie za uzasadnione, jeśli wśród losowo zbadanych 400 pracowników przedszkoli było 128 mężczyzn?

b) Utworzyć 95% przedział ufności dla odsetka mężczyzn wśród pracowników przedszkoli wykorzystując wyniki badania z punktu a).

c) Czy odpowiedź uzyskana w pkt. a) zmieniłaby się, gdyby pełnomocnik pobrał reprezentatywną próbkę 10 pracowników przedszkoli i 3 z nich okazałoby się mężczyznami?

d) Utworzyć 95% przedział ufności dla odsetka mężczyzn wśród pracowników przedszkoli dla danych z punktu c).

Rozwiązanie: Z treści zadania wynikają następujące hipotezy dla udziału mężczyzn wśród pracowników przedszkoli: **Hipoteza zerowa** $H_0 : p = 0.35$**, hipoteza alternatywna** $H_1 : p < 0.35$.

a) W tym celu korzystamy funkcji **`prop.test`**, podstawiając odpowiednie parametry: **`x = 128` - liczba sukcesów, `n = 400` - liczebność próby, `p = 0.35` - wartość stosunku w hipotezie zerowej, `alt = "less"` - rodzaj hipotezy alternatywnej - tutaj "less" oznacza, że rozpatrujemy hipotezę** $H_1 : p < 0.35$. Ta funkcja wypisze dokładny opis testu, a przede wszystkim wartość p-value, na podstawie której wnioskujemy, czy możemy odrzucić hipotezę. Dla tych danych $p - value = 0.114$, a więc większe od postawionego przedziału istotności wynoszącego 0.05, a więc **nie ma podstaw do odrzucenia hipotezy zerowej**.

b) Wywołanie funkcji bardzo podobne do a) - również korzystamy z funkcji **`prop.test`**, tym razem wykorzystujemy ją do utworzenia 95% przedziału ufności, korzystając z wyników z a) - a dokładniej z liczby sukcesów **`x = 128` i liczebności próby `n = 400`**. Należy pamiętać o ustawieniu **`conf.level = 0.95`**. Otrzymujemy **przedział ufności** $[0.2749928, 0.3685248]$.

c) W tym przypadku niestety **nie możemy skorzystać** ponownie z **`prop.test`**, ponieważ dla korzystania z tej funkcji liczba sukcesów oraz liczba porażek w próbie musi wynosić przynajmniej 5. W tym podpunkcie nie warunek nie jest spełniony. Należy więc skorzystać z innej funkcji, a dokładniej **`binom.test`**. Wywołanie tej funkcji oraz jej parametry są analogiczne do **`prop.test`** - wystarczy podmienić l**iczbę sukcesów `x = 3` i liczebność próby `n = 10`**. Otrzymujemy $p - value = 0.5138$, a więc podobnie **nie ma podstaw do odrzucenia hipotezy zerowej**.

d) Wywołanie analogiczne do b), tylko podobnie, jak w c) korzystamy z funkcji **`binom.test`**. Otrzymujemy **przedział ufności** $[0.06673951, 0.65245285]$

Pełny kod rozwiązania:

```
# Zadanie 3
# a)

prop.test(x=128, n=400, p=0.35, alternative = 'less')

# b)
prop.test(x=128, n=400, conf.level=0.95)$conf
binom.test(x=128, n=400, conf.level=0.95)$conf

# c)
binom.test(x=3, n=10, p=0.35, alternative='less')

# d)
binom.test(x=3, n=10, conf.level=0.95)
```

Zadanie 3.4. Badano staż pracy osób zatrudnionych w pewnej dużej sieci handlowej. Na 150 losowo wybranych pracowników, 118 pracowało w tej sieci mniej niż 5 lat. Czy na tej podstawie można twierdzić, że 80% pracowników tej sieci legitymuje się stażem pracy mniejszym niż 5 lat? Zweryfikować odpowiednią hipotezę przyjmując poziom istotności 0,05.

Rozwiązanie: Z treści zadania należy wywnioskować ponownie potrzebne dane: **liczba sukcesów** $x = 118$, **liczebność próby** $n = 150$, **hipoteza zerowa** $H_0 : p = 0.8$**, hipoteza alternatywna** $H_1 : p \neq 0.8$, **poziom istotności** $\alpha = 0.05$. Zadanie jest analogiczne do **3.1 a)**. Tu również korzystamy z funkcji **prop.test**, wstawiamy parametry **x = 118, n = 150, p = 0.8, alt = "two.sided"** - tutaj stosujemy "two.sided", gdyż sprawdzamy w hipotezie alternatywnej, czy szukana wartość jest różna od tej zadanej w p ($H_1 : p \neq 0.8$). Otrzymujemy wynik $p - value = 0.7595$, czyli nie ma podstaw do odrzucenia hipotezy zerowej.

Pełny kod rozwiązania:

```
# Zadanie 4
# H0 = p=0.8
# H1 : p != 0.8

prop.test(x=118, n=150, p=0.8, alternative='two.sided')
```

Zadanie 3.5.

a) Napisać funkcję, która dla dużej próby losowej ($n \geq 100$) z dowolnego rozkładu, zwraca przedział ufności dla średniej na zadanym poziomie ufności. Zadbać, by funkcja zwracała błąd w przypadku jej użycia do próby o liczności mniejszej niż 100.

b) W pakiecie **MASS** znajduje się zbiór danych **geyser** zawierający kolumnę **duration** z czasami trwania (w minutach) wybuchów gejzeru Old Faithful w Parku Narodowym Yellowstone w USA. Na poziomie ufności 0,95 wyznaczyć przedział ufności dla średniego czasu trwania wybuchu tego gejzera.

Rozwiązanie: Zadanie pominięte na zajęciach. Rozwiązanie nadesłane jako praca domowa.

```
rm(list = ls())

# a
# Compute Confidence Interval for the Mean (n >= 100)
#
# This function calculates a confidence interval for the mean of a numeric vector of an
    arbitrary distribution
#
# x             numeric vector representing the sample data, must contain at least 100
    observations
```

```r
# conf.level    confidence level for the interval, numeric value between 0 and 1

confidence_interval <- function(x, conf.level){
  n <- length(x)
  if (n < 100) {
    stop("Vector must be of length >= 100")
  }
  mean = mean(x)
  sd = sd(x)
  alpha = 1-conf.level
  margin = qnorm(1-alpha/2)*sd/sqrt(n)

  result <- list(
    conf.int = c(mean-margin, mean+margin),
    conf.level = conf.level,
    mean = mean
  )

  return (result)
}

# b
alpha = 0.05
library(MASS)
x = geyser$duration
confidence_interval(x=x, conf.level=1-alpha)
t.test(x=x, conf.level=1-alpha)


# Additional tests

# Compare two functions calculating confidence intervals
#
# This function calculates return the sum of differences of calculated intervals
#
# function1, function2        functions to be compared
# generating_data_function    function that generates data
# sample_size                 size of data generated
# conf.level                  confidence level for the interval, numeric value between 0 and 1

diff_confidence_interval <- function(function1, function2, generating_data_function, sample_size, conf.level){
  x = generating_data_function(n=sample_size)
  interval1 = function1(x=x, conf.level=conf.level)$conf.int
  interval2 = function2(x=x, conf.level=conf.level)$conf.int
  # print(interval1)
  # print(interval2)
  return (sum(abs(interval1-interval2)))
}

# diff_confidence_interval(confidence_interval, t.test, rnorm, 100, 0.95)


generators <- list(
  Normal = rnorm,
  Exponential = rexp
  # Uniform = runif
  # Gamma = function(n) rgamma(n, shape=2, scale=2)
  # Beta = function(n) rbeta(n, shape1=2, shape2=5)
  # Cauchy = rcauchy

  # be careful when using many, may cause plotting problems on small screens
)
```

```r

par(mfrow = c(length(generators), 1))
sample_sizes = seq(100, 1000, by = 5)

# for each function declared in generators
#   compares t.test with our function using diff_confidence_interval
#   for each sample size from sample_sizes and plots the outcome
for(i in 1:length(generators)){
  results = numeric(length=length(sample_sizes))
  for(j in 1:length(sample_sizes)){
    results[j] <- diff_confidence_interval(
      confidence_interval,
      t.test,
      generators[[i]],
      sample_sizes[j],
      conf.level = 0.95
    )
  }
  plot(sample_sizes, results, type = "o", col = "blue", pch = 16,
       ylim = range(unlist(results)),
       xlab = "sample Size", ylab = "(sum(abs(interval1-interval2)))",
       main = paste("Data generated using", names(generators)[i]))
}
```

Uwagi

- Różne funkcje mogą różnie reagować na wartość `NA`. `t.test` domyślnie je ignoruje, `sigma.test` zwróci `NA` jako wynik, podobnie jak `min`, `max`, `mean`. Niektóre funkcje posiadają parametry do usuwania wartości `NA` - `min`, `max`, `mean` posiadają parametr `na.rm`, który może przyjąć wartości `TRUE` albo `FALSE`. Warto też wspomnieć funkcje:
  - `na.omit` - usuwa wartości `NA`
  - `na.fail` zwróci dany objekt o ile ten nie posiada żadnych brakujących wartości

4. Testy parametryczne dla dwóch populacji

Zadanie 4.1. W losowej próbie 233 dorosłych mieszkańców Warszawy znalazło się 40 takich, które regularnie robią zakupy w sklepach sieci Żuczek. W Krakowie na 220 zapytane osoby, 31 okazało się klientami Żuczka.

a) Czy na podstawie powyższych danych można stwierdzić, że odsetek regularnych klientów Żuczka w Warszawie jest większy niż w Krakowie? Przyjąć poziom istotności $\alpha = 0{,}05$.

b) Przypuszczamy, że odsetek regularnych klientów Żuczka w Warszawie wynosi 17%, a w Krakowie - 14%.

   i) Jakie jest prawdopodobieństwo, że test z pkt. a) potwierdzi, że odsetek regularnych klientów Żuczka jest większy w Warszawie niż w Krakowie?

   ii) Ilu mieszkańców Warszawy i ilu mieszkańców Krakowa trzeba by wylosować do próby by, z prawdopodobieństwem nie mniejszym niż 0,8, jednostronny test o poziomie istotności 0,05 porównujący odsetek regularnych klientów Żuczka potwierdził, że odsetek ten jest większy w Warszawie niż w Krakowie?

Rozwiązanie: Mamy tutaj zadanie opisujące problem dwóch populacji. Na podstawie treści od razu możemy określić liczby sukcesów i liczebności prób obydwu populacji (1 - Warszawa, 2 - Kraków):

- liczby sukcesów: $x_1 = 40$, $x_2 = 31$
- liczebności prób $n_1 = 233$, $n_2 = 220$

a) Mamy dany poziom istotności $\alpha = 0.05$. Należy sprawdzić, czy na podstawie danych z treści zadania jesteśmy w stanie stwierdzić, czy odsetek klientów z Warszawy jest większy od tego w Krakowie. Określamy hipotezy. Oznaczmy p_1, p_2 - wartości odsetków klientów z Warszawy i Krakowa. Wówczas **hipoteza zerowa** $H_0 : p_1 = p_2$, **hipoteza alternatywna** $H_1 : p_1 > p_2$. Da się to zweryfikować, wykorzystując funkcję **`prop.test`**. Wprowadzamy do niej trzy parametry, **x - liczby sukcesów, n - liczebności prób, alternative(alt) - rodzaj hipotezy alternatywnej**. Tutaj najważniejszym punktem jest odpowiednie wprowadzenie parametrów x i n - trzeba mieć na uwadze, że mamy tutaj zadanie z dwoma populacjami - należy więc wprowadzić dane liczby sukcesów i liczebności prób, definiując wektory z tymi danymi **`x = c(x1, x2), n = c(n1, n2)`**. Koniecznie trzeba pamiętać o zachowaniu kolejności wprowadzania danych - w tym przypadku przypisaliśmy dane Warszawy jako te pierwsze, a Krakowa - drugie i **trzeba tę kolejność zachować wszędzie**! W ten sposób możemy bezproblemowo określić rodzaj hipotezy alternatywnej **`alt = "greater"`** (w kodzie "gr" to skrót od "greater" i ten sam parametr). Otrzymaliśmy p-value = 0.2204. Pamiętając, że przedział istotności określiliśmy na 0.05, czyli $p-value >= \alpha$ **nie mamy podstaw do odrzucenia hipotezy zerowej**

b) Przypuszczamy tutaj, że odsetki z Warszawy i Krakowa wynoszą $p_1 = 0.17, p_2 = 0.14$.

   i) Mamy określić, czy test a) potwierdzi, że odsetek z Warszawy jest większy od odsetku z Krakowa, czyli potwierdzi hipotezę alternatywną, czyli **odrzuci hipotezę zerową**, mówiącą, że te odsetki są równe. To zadanie sprowadza się więc do obliczenia **funkcji mocy testu**. Korzystamy z funkcji **`power.prop.test`**, do którem wprowadzamy następujące parametry: **`p1 = 0.17, p2 = 0.14, n = c(233, 220), sig.level = 0.05, alt="one.sided"`**. Oczywiście, `p1, p2` będą określały wartości zakładanych odsetek odpowiednio w Warszawie i Krakowie, `n` określa liczbności prób w obydwu miastach, a `sig.level` będzie przechowywał przedział istotności. Można przy wyświetlić tylko szukaną moc dopisując za funkcją `$power` Otrzymujemy dwie wartości funkcji mocy liczone osobno dla każdej z prób: 0.2263947 i 0.2188411 - można twierdzić, że ogólna wartość funkcji mocy znajduje się pomiędzy tymi dwiema wartościami, czyli tym samym prawdopodobieństwo, że test potwierdzi relację z hipotezy alternatywnej między tymi odsetkami wynosi **ok. 22%**.

ii) Mamy oczekiwane prawdopodobieństwo potwierdzenia tego samego testu wynoszące ≥ 0.8. Chcemy dowiedzieć się ile należy wylosować mieszkańców do obydwu prób, by dostać oczekiwane prawdopodobieństwo. Wywołujemy tę samą funkcję **`power.prop.test`**, tym razem jednak zamienić parametr liczebności prób na oczekiwane prawdopodobieństwo, czyli w tym przypadku moc **`power = 0.8`**. Wyświetlamy tylko liczebności prób **`$n`**. Otrzymany wynik **1799** (w przybliżeniu) oznacza, że **każda próba musi mieć liczebność wynoszącą przynajmniej 1799**, by wartość funkcji mocy testu przyjęła oczekiwany wynik.

Pełny kod rozwiązania:

```
# 4.1

# a)
prop.test(x = c(40, 31), n = c(233, 220), alt = "gr")

# bi)
power.prop.test(p1 = 0.17, p2 = 0.14, n = c(233, 220), sig.level = 0.05,
                alt="one.sided")$power

# bii)
power.prop.test(p1 = 0.17, p2 = 0.14, power = 0.8, sig.level = 0.05,
                alt="one.sided")$n
```

Zadanie 4.2. Pomiary dokonane niezależnie na próbach losowych dwóch gatunków papierosów dały następujące wyniki zawartości nikotyny (w miligramach):

Gatunek A: 26,4, 22,5, 24,9, 23,7, 21,5
Gatunek B: 25,1, 29,0, 23,4, 27,6, 22,3

Przyjmujemy, że w przypadku obu badanych gatunków papierosów zawartość nikotyny ma rozkład normalny.

a) Na poziomie istotności $\alpha = 0{,}05$ zweryfikować hipotezę, że gatunek B ma wyższą zawartość nikotyny niż gatunek A.

b) Zakładając, że gatunek B ma zawartość nikotyny średnio o 2 miligramy większą niż gatunek A, obliczyć prawdopodobieństwo, że test z pkt. a) da błędną odpowiedź.

c) Załóżmy, że gatunek B ma zawartość nikotyny średnio o 2 miligramy większą niż gatunek A. Jak liczne próby losowe tych gatunków papierosów trzeba by pobrać, by na ich podstawie test z pkt. a), z prawdopodobieństwem nie mniejszym niż 0,75, dawał poprawną odpowiedź?

Rozwiązanie: Rozważamy problem dwóch populacji, a dokładniej hipotezy dotyczące ich wartości średnich. Dane mamy z dwóch niezależnych prób losowych z obu populacji (nie są to pary) o rozkładzie normalnym i nieznanych wszystkich parametrach, sugeruje to zatem użycie **Modelu II**. Na początku jednak musimy zbadać, czy obie populacje mają te same odchylenia standardowe (potrzebne jest to do jednego z parametrów funkcji `t.test` - `var.equal`. Badamy następujące hipotezy:

$$H_0 : \sigma_a^2 = \sigma_b^2$$
$$H_1 : \sigma_a^2 \neq \sigma_b^2$$

Wywołujemy `var.test(A,B)` (domyślnie `alternative='two.sided"`, reszta argumentów nas nie interesuje) i odczytujemy p-value: 0.4891. Zalecane jest wykonywanie tego testu na poziomie istotności 0.1 $\implies$ nie ma podstaw do odrzucenia hipotezy zerowej, i używamy **Modelu II** z równymi wariancjami.

a) Badamy hipotezy:

$$H_0 : \mu_a = \mu_b$$
$$H_1 : \mu_a < \mu_b$$

Aby zweryfikować hipotezę użyjemy funkcji `t.test` z odpowiednimi argumentami:

- `A, B` - pierwsze dwa argumenty to uzyskane próby losowe
- `alternative='less'` - zgodnie z hipotezą alternatywną, kolejność podanych zbiorów ma znaczecnie
- `var.equal=TRUE` - równość wariancji (zweryfikowana wcześniej), domyślnie jest FALSE

Dodatkowo korzystamy z domyślnego argumentu `paired = FALSE` zgodnie z modelem. Odczytujemy p-value: 0.1511. Jest ono większe niż poziom istotności testu (zadany w poleceniu), zatem nie ma podstaw do odrzucenia hipotezy zerowej. Stąd nie ma podstaw do przyjęcia hipotezy, że gatunek B ma wyższą zawartośc nikotyny niż A.

b) Chcemy wyznaczyć:

$$P(\text{test przyjmie } H_0 \mid \mu_b - \mu_a = 2) = 1 - P(\text{test odrzuci } H_0 \mid \mu_b - \mu_a = 2) = 1 - \text{moc testu}(2)$$

Wykorzystamy znaną już funkcję `power.t.test`. Do jej wywołania potrzebujemy jednak podać argument `sd`, który musimy wyznaczyć empirycznie na podstawie dwóch prób losowych. Używamy do tego poniższego wzoru:

$$\sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}$$

Który - zważając na to, iż $n_1 = n_2$, możemy na szczęście nieco uprościć:

$$\sqrt{\frac{s_1^2+s_2^2}{2}}$$

s_1^2 i s_2^2 wyznaczamy empirycznie jako wariancje z prób. Ostatecznie wywołujemy funkcję `power.t.test` z następującymi argumentami:

- `n=length(A)` - liczność prób
- `delta=2` - różnica obu wartości średnich (zgodnie z poleceniem)
- `sd=sqrt((var(A) + var(B))/2)` - zgodnie ze wzorem powyżej
- `type='two.sample'` - test dla dwóch populacji (inna opcja to `paired`, co oczywiście w naszym przypadku nie ma miejsca)
- `alternative='one.sided'` - gdyż hipoteza alternatywna to relacja mniejszości (a nie nierówność)

Pomijamy oczywiście parametr power, który to funkcja ta nam wyznaczy. Z wyniku tej funkcji wydobywamy `$power` i oczywiście odejmujemy tą wartość od jedynki, zgodnie z tym co chcemy wyznaczyć. Odczytujemy wynik: 0.6710692.

c) Chcemy oczywiście, aby:

$$P(\text{test odrzuci } H_0 \mid \mu_b - \mu_a = 2) \geq 0.75$$

Co jest równoważne temu, aby moc testu była większa od 0.75. Wywołujemy zatem tą samą funkcję, co w podpunkcie b) z tymi samymi argumentami, zamieniając `n=length(A)` na `power=0.75`. Pomijamy wtedy argument `n`, który to funkcja ta nam wyznaczy. Odczytujemy wynik: n = 16.32933. Zatem potrzebowalibyśmy obu prób o liczności przynajmniej 17.

Pełny kod rozwiązania:

```
# 4.2

gat.a = c(26.4, 22.5, 24.9, 23.7, 21.5)
gat.b = c(25.1, 29.0, 23.4, 27.6, 22.3)

var.test(x = gat.a, y = gat.b)

# a)
t.test(x = gat.a, y = gat.b, alt="less", var.equal = T)

# b)
 p = 1 - power.t.test(n = 5, delta = 2, sd = sqrt((var(gat.a) + var(gat.b)) / 2), type = "two.sample",
                alt = "one.sided")$power
 p

 # c)
 power.t.test(delta = 2, p = 0.75, sd = sqrt((var(gat.a) + var(gat.b)) / 2), type = "two.sample",
              alt = "one.sided")$n
```

Zadanie 4.3. Do badania wybrano w sposób losowy 15 dzieci chorych na cukrzycę. Poddano ich kuracji podając nowo opracowany lek. W pliku `hemoglobina` zapisano poziom hemoglobiny glikowanej (w %) u tych dzieci przed (zmienna *przed*) oraz po kuracji (zmienna *po*). Wiadomo, że poziomy te mają łączny rozkład normalny.

a) Czy dane te potwierdzają, że nowy lek obniża poziom hemoglobiny glikowanej? Przyjąć poziom istotności 0,05.

b) Zakładając, że nowy lek obniża poziom hemoglobiny glikowanej o średnio 1,5 pp. (punktu procentowego), wyznaczyć moc testu z punktu a) i podać interpretację otrzymanego wyniku.

Rozwiązanie: Zbiór danych `hemoglobina.txt` (ze strony Pani Prof. Dembińskiej: Strona główna > Dydaktyka > Materiały dydaktyczne > Zbiory danych) ładujemy do R używając polecenia:

```
hemoglobina <- read.csv("/path/to/file/hemoglobina.txt" , sep="")
```

Następnie przypisujemy wyniki badań przed i po do zmiennych pomocniczych używając operatora $.

a) W tym zadaniu dysponujemy wzajemnie niezależnymi parami obserwacji (wyniki przed i po kuracji u każdego z dzieci). Ponadto wiemy, że różnice wyników mają rozkład normalny. Korzystamy więc z modelu III.

   Hipoteza zerowa H_0: $\mu_{\text{przed}} = \mu_{\text{po}}$, czyli terapia nie obniżyła średniego poziomu hemoglobiny glikowanej.

   Hipoteza alternatywna H_1: $\mu_{\text{przed}} > \mu_{\text{po}}$, czyli terapia działa skutecznie.

   W celu wyznaczenia p-value skorzystamy z funkcji `t.test` z parametrem `paired` ustawionym na `TRUE`. Odczytujemy wynik ≈ 0.029, czyli mniej niż ustalony poziom istotności. Odrzucamy zatem H_0 i wobec tego stwierdzamy, że nowy lek obniża poziom hemoglobiny glikowanej.

b) Aby wyznaczyć moc testu, używamy funkcji `power.t.test` z odpowiednimi argumentami: `delta` $= 1.5$, liczność próby $n = 15$, odchylenie standardowe z różnicy wyników, typ testu `paired` i rodzaj hipotezy alternatywnej `one.sided`.

   W wyniku otrzymujemy moc ≈ 0.936, czyli po wykonaniu testu z punktu a) słusznie odrzucimy hipotezę zerową z prawdopodobieństwem ≈ 0.936.

Pełny kod rozwiązania:

```
rm(list = ls())

hemoglobina <- read.csv("/path/to/file/hemoglobina.txt", sep="")

before = hemoglobina$przed
after = hemoglobina$po

t.test(before, after, alt="gr", paired=TRUE)

power.t.test(n=length(before), delta=1.5, sd=sd(before-after), type="p", alt="o", sig.
    level=5e-2)
```

Zadanie 4.4. Zbiór `nlschools`, znajdujący się w bibliotece `MASS`, zawiera dane dotyczące wybranych uczniów szkół holenderskich kończących ósmą klasę:

- **IQ** – wynik testu na IQ werbalne (w punktach),
- **SES** – społeczno-ekonomiczny status rodziny ucznia.

Czy na podstawie powyższych danych możemy stwierdzić na poziomie istotności 0,05, że wśród uczniów kończących ósmą klasę ci pochodzący z domów o społeczno-ekonomicznym statusie powyżej mediany mają wyższy poziom inteligencji werbalnej niż pozostali?

Rozwiązanie: W tym zadaniu nie posiadamy żadnych informacji co do rozkładu, z którego pochodzą wyniki IQ. Mamy natomiast niezależne próby losowe z dwóch populacji (wyniki uczniów pochodzących z domów o statusie społeczno-ekonomicznym powyżej mediany `x` i pozostałych `y`). Liczności n_1 i n_2 tych populacji wynoszą ponad 1000, więc korzystamy z modelu IV.

Niech μ_1 oznacza średni wynik uczniów „bogatszych”, a μ_2 średni wynik „biedniejszych”. Hipoteza zerowa H_0: $\mu_1 = \mu_2$, alternatywna H_1: $\mu_1 > \mu_2$.

Ponieważ nie istnieje funkcja wbudowana w R, która oblicza statystykę testową w tym przypadku, robimy to korzystając ze wzoru:

$$U = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{s_2^1}{n_1}} + \sqrt{\frac{s_2^2}{n_2}}} \approx 12.137.$$

Zmienna losowa U ma w przybliżeniu rozkład normalny, więc nawet na oko widać, że prawdopodobieństwo wystąpienia wartości statystyki testowej większej niż otrzymana (czyli p-value) jest bliskie zeru. Dokładniej jest ono rzędu 10^{-34}, możemy to obliczyć funkcją `pnorm`.

Wobec tego odrzucamy hipotezę zerową i stwierdzamy, że uczniowie pochodzący z bogatszych domów mają wyższy poziom inteligencji werbalnej niż pozostali.

Pełny kod rozwiązania:

```
rm(list = ls())

library(MASS)

alpha = 5e-2
med = median(nlschools$SES)

x = nlschools$IQ[nlschools$SES>med]
y = nlschools$IQ[nlschools$SES<=med]
n1 = length(x)
n2 = length(y)

U = (mean(x)-mean(y))/(sqrt(var(x)/n1+var(y)/n2))
1 - pnorm(U)
pnorm(U, lower.tail=F)
```

5. Testowanie zgodności

Zadanie 5.1. Losową próbę studentów spytano o ich ulubiony przedmiot. Otrzymano następujące odpowiedzi:

Przedmiot	Fizyka	WF	Mechanika	Statystyka
Liczba studentów	380	340	380	500

Na poziomie istotności 0.05 sprawdzić hipotezę, że rozkład preferencji jest równomierny.

Rozwiązanie: W tym zadaniu badamy, czy dane liczbowe pochodzą z rozkładu równomiernego. Dane to liczby osób wybierających cztery różne przedmioty. Hipoteza zerowa H_0: rozkład jest równomierny, czyli każdy z czterech przedmiotów ma prawdopodobieństwo $\frac{1}{4}$ bycia ulubionym.

W celu zbadania zgodności skorzystamy z testu χ^2. Możemy go użyć, ponieważ $np \geq 5$, gdzie n to suma po licznościach wszystkich grup, a p to prawdopodobieństwo przynależności do danej grupy (tutaj równe $\frac{1}{4}$ dla wszystkich grup).

Jako dane podajemy wartości skopiowane z treści, domyślnie `chisq.test` zbada czy faktycznie pochodzą one z rozkładu równomiernego. Można to wywnioskować po spojrzeniu do dokumentacji - domyślny wektor prawdopodobieństw p ma n elementów, każdy równy $\frac{1}{n}$, czyli reprezentuje on rozkład równomierny:

```
chisq.test(x, y = NULL, correct = TRUE,
       p = rep(1/length(x), length(x)), rescale.p = FALSE,
       simulate.p.value = FALSE, B = 2000)
```

Ponieważ wartość p-value jest bardzo mała (poniżej poziomu istotności $\alpha = 0.05$), odrzucamy hipotezę zerową. Wniosek: dane nie pochodzą z rozkładu równomiernego.

Pełny kod rozwiązania:

```
rm(list = ls())

alpha = 0.05
data = c(380, 340, 380, 500)

sum(data)/4

chisq.test(data)
```

Zadanie 5.2. Zbadano grupę krwi 100 osób. Grupę 0 miało 36 osób, A – 42 osoby, B – 14 osób i grupę AB – 8 osób. Zweryfikować hipotezę, że prawdopodobieństwa wystąpienia grup krwi 0, A, B, AB w populacji są równe odpowiednio: 0.4; 0.4; 0.1; 0.1. Przyjąć poziom istotności 0,05.

Rozwiązanie: W tym zadaniu testujemy zgodność danych z rozkładem teoretycznym o zadanych prawdopodobieństwach: $p = [0.4, 0.4, 0.1, 0.1]$. Ponownie należy sprawdzić czy możemy skorzystać z `chisq.test`, robimy to za pomocą funkcji `sum` i mnożąc jej wynik przez wektor prawdopodobieństw. Każda z wartości jest ≥ 5, więc odpowiedź jest twierdząca. Sprawdzamy, czy obserwowane liczności są zgodne z oczekiwanymi za pomocą funkcji `chisq.test`. Wartość p-value jest większa od poziomu istotności $\alpha = 0.05$, więc nie mamy podstaw do odrzucenia hipotezy zerowej. Wniosek: dane mogą pochodzić z danego rozkładu.

Pełny kod rozwiązania:

```
rm(list = ls())

data = c(36, 42, 14, 8)
p = c( 0.4, 0.4, 0.1, 0.1)
alpha = 5e-2
n = 1e2

sum(data)*p
chisq.test(x=data, p=p)
```

Zadanie 5.3. Aby zaliczyć programowanie, Maciek musi napisać program generujący liczby losowe z rozkładu dwumianowego o parametrach $n = 3$ i $p = 0.5$. Aby udowodnić, że jego program działa poprawnie, wygenerował 200 liczb i uzyskał następujące wyniki:

Wartość	0	1	2	3
Liczność	24	73	77	26

Czy na poziomie istotności 0.05 można stwierdzić, że generator Maćka działa prawidłowo?

Rozwiązanie: Ponownie używamy `chisq.test`, tym razem jednak wektor `p` generujemy używając `dbinom` oraz możliwości tej funkcji obliczenia prawdopodobieństw dla danych wektorowych.
Wartość p-value jest większa niż poziom istotności, więc nie odrzucamy H_0. Wniosek: próbka jest zgodna z rozkładem dwumianowym.

Pełny kod rozwiązania:

```
rm(list = ls())

p = 5e-1
n = 3
data = c(24, 73, 77, 26)
p_data = dbinom(0:3, size=n, prob=p)
chisq.test(x=data, p=p_data)
```

Zadanie 5.4. Naukowiec chce sprawdzić, czy liczba cząstek emitowanych przez pewną substancję promieniotwórczą w ciągu 10 sekund ma rozkład Poissona. W tym celu zbadał liczbę cząstek emitowanych w dziesięciosekundowych odcinkach czasu:

Liczba cząstek	0	1	2	3	4	5
Liczba przypadków	140	280	235	200	100	45

Jakie wnioski można wyciągnąć na poziomie istotności 0.1?

Rozwiązanie: Dane zawierają liczby obserwacji dla wartości od 0 do 5. Najpierw estymujemy parametr λ rozkładu Poissona. Możemy to zrobić zarówno funkcją `fitdistr` poznaną na wcześniejszych laboratoriach, lub w tym przypadku poprzez policzenie średniej używając funkcji `mean`. Następnie obliczamy teoretyczne prawdopodobieństwa i wartości oczekiwane, pamiętając że suma wszystkich dostarczonych prawdopodobieństw musi być równa 1 - musimy wziąć pod uwagę przypadki, których nie zaobserwowaliśmy: liczba cząstek wyemitowanych wyniosła 6 lub więcej.
Ponownie należy sprawdzić czy możemy skorzystać z `chisq.test`, robimy to za pomocą funkcji `sum` i mnożąc jej wynik przez wektor prawdopodobieństw. Każda z wartości jest ≥ 5, więc odpowiedź jest twierdząca.
Uwaga: nie możemy bezpośrednio odczytać p-value, bo ono odpowiada prostej hipotezie H_0: badana próba losowa pochodzi z rozkładu Poissona o średniej, która estymowaliśmy. Aby przetestować złożoną hipotezę H_0: badana próba losowa pochodzi z rozkładu Poissona, możemy jedynie wykorzystać obliczoną w ten sposób wartość statystyki testowej, ale zbiór krytyczny musimy wyznaczyć sami, co robimy dalej.

Zbiór krytyczny wyrażany jest wzorem $W = \left\langle \chi^2_{1-\alpha,\, k-1-r}, +\infty \right)$, gdzie k jest liczbą klas, r jest liczbą parametrów szacowanych z próby, zaś $\chi^2_{1-\alpha,\, k-1-r}$ to kwantyl rzędu $1-\alpha$ rozkładu chi-kwadrat o $k-1-r$ stopniach swobody. Szacowaliśmy jeden parametr i mamy 6 klas, zatem $k-1-r = 6-1-1 = 4$. Kwantyl obliczamy za pomocą funkcji `qchisq` i zauważamy, że statystyka testowa należy do zbioru krytycznego - należy odrzucić H_0.

Ponieważ p-value $= 0.04511184 < \alpha = 0.1$, odrzucamy hipotezę zerową. Wniosek: dane nie pochodzą z rozkładu Poissona. Zauważmy ponadto, że ta p-value jest inna niż ta wyznaczona przez `chisq.test` (równa 0.08306). Pokazuje to, że nie mogliśmy użyć wbudowanej funkcji R, ponieważ jeden z parametrów rozkładu estymujemy.

Pełny kod rozwiązania:

```
rm(list = ls())

library(MASS)
alpha = 1e-1
data = c(140, 280, 235, 200, 100, 45)
data2 = rep(0:5, data)
lambda_estimator <- fitdistr(data2,"Poisson")$estimate

p_data = c(dpois(0:4, lambda=lambda_estimator), ppois(4, lambda_estimator, lower.tail=
    FALSE))

sum(data)*p_data

chisq.test(x=data, p=p_data)
1-pchisq(9.7363, 4)
```

Zadanie 5.5. W kolumnie *czas* w pliku `infolinia.txt` zapisano czasy oczekiwania (w minutach) na połączenia z pewną infolinią. Używając testu Kołmogorowa-Smirnowa, sprawdzić czy można uznać, że prezentowane czasy pochodzą z rozkładu gamma Gamma(a, β) z parametrem kształtu $a = 4.5$ i drugim parametrem $\beta = 4$. Przyjąć poziom istotności $\alpha = 0.05$.

Rozwiązanie: Dane wczytujemy z pliku za pomocą polecenia `read.csv` lub za pomocą interfejsu graficznego. Następnie używamy `ks.test` aby przeprowadzić test Kołmogorowa-Smirnowa. Jako argument `x` podajemy nasze dane, jako `y` nazwę rozkładu, z którym nasze dane będą porównywane w hipotezach. Parametry `shape` oraz `rate` to parametry, które opisują nasz rozkład. Jest to bardzo istotne, ponieważ podobnie jak wcześniej `ks.test` obsługuje test Kołmogorowa-Smirnowa jedynie dla prostej H_0. W przypadku złożonej H_0 (np. H_0: rozkład badanej cechy jest wykładniczy z nieznanym parametrem λ) pozostaje nam samemu wyznaczyć przybliżoną wartość p-value metodą symulacji.
Ponieważ p-value jest większe niż $\alpha = 0.05$, nie ma podstaw do odrzucenia hipotezy zerowej. Wniosek: dane są zgodne z rozkładem gamma.

Pełny kod rozwiązania:

```
rm(list = ls())

time <- read.csv("~/Documents/coding/R/lab5/infolinia.txt", sep="")

alpha = 5e-2
a = 4.5
beta = 4

ks.test(x=time, y="pgamma", alternative="t", shape=4.5, rate=4)
```

Zadanie 5.6. Wygenerować po $N = 100$ liczb z następujących rozkładów:

(i) normalnego o średniej $= 20$ i odchyleniu standardowym $= 5$,

(ii) jednostajnego na przedziale $(-1, 1)$,

(iii) wykładniczego o średniej $= 5$,

(iv) Poissona o średniej $= 3$.

a) Dla każdej próbki sporządzić wykres normalności i wyświetlić je w jednym oknie. Przeanalizować ich kształt.

b) Dla każdej próbki sporządzić wykres skrzynkowy i wyświetlić je w jednym oknie. Przeanalizować ich kształt.

c) Dla każdej próbki sporządzić histogram częstości i nanieść na niego jądrowy estymator gęstości. Przeanalizować kształty tych wykresów.

d) Dla danych z punktów (i) i (ii) przeprowadzić test normalności Shapiro-Wilka na poziomie istotności 0,05.

Rozwiązanie: Do sporządzenia wykresów normalności (kwantylowych) używamy `qqnorm` oraz `qqline` do naniesienia linii przechodzącej przez kwartyle.
Niech $x_1, x_2, \ldots, x_n$ oznacza realizację próby losowej. Po jej uporządkowaniu (od obserwacji najmniejszej do największej) otrzymujemy tzw. statystyki porządkowe z próby, oznaczane $x_{1:n}, x_{2:n}, \ldots, x_{n:n}$. Wykres kwantylowy to zbiór punktów o współrzędnych $(u_{(i-0.5)/n}, x_{i:n})$, gdzie $i = 1, 2, \ldots, n$ zaś $u_{(i-0.5)/n}$ to kwantyl standardowego rozkładu normalnego rzędu $(i - 0.5)/n$. Jeśli próba losowa pochodzi z rozkładu normalnego $\mathcal{N}(\mu, \sigma^2)$, to wykres kwantylowy jest zbiorem punktów leżących mniej-więcej na prostej $y = \sigma x + \mu$. Wykresy skrzynkowe oraz histogramy częstości były opisywane na wcześniejszych laboratoriach.
Do przeprowadzenia testów Shapiro-Wilka używamy funkcji `shapiro.test`. Przyjmuje ona jeden argument, wektor z danymi. Wartości p-value jakie kolejno otrzymujemy to: 0.2834, 0.001053, $1.994 \cdot 10^{-10}$, 0.001613. Pokrywa się to z naszymi oczekiwaniami, ponieważ faktycznie tylko `x1` była wygenerowana za pomocą `rnorm`.

Pełny kod rozwiązania:

```
rm(list = ls())

par("mfcol"=c(4,3))
N = 100
x1 = rnorm(n=N, 20, 5)
x2 = runif(n=N, -1, 1)
x3 = rexp(n=N, 5)
x4 = rpois(n=N, 3)

qqnorm(x1, main="Norm")
qqline(x1)
qqnorm(x2, main="Unif")
qqline(x2)
qqnorm(x3, main="Exp")
qqline(x3)
qqnorm(x4, main="Pois")
qqline(x4)

boxplot(x1, main="Norm")
boxplot(x2, main="Unif")
boxplot(x3, main="Exp")
boxplot(x4, main="Pois")

hist(x1, main="Norm", freq=FALSE)
lines(density(x1))
```

```
hist(x2, main="Unif", freq=FALSE)
lines(density(x2))
hist(x3, main="Exp", freq=FALSE)
lines(density(x3))
hist(x4, main="Pois", freq=FALSE)
lines(density(x4))

alpha = 5e-2
shapiro.test(x1)
shapiro.test(x2)
shapiro.test(x3)
shapiro.test(x4)
```

Zadanie 5.7. * Posługując się pakietem R i ustalając ziarno generatora równe 4411 wygenerować 200 liczb z rozkładu wykładniczego o parametrze $\lambda = 2$. Następnie, na poziomie istotności 0.05, sprawdzić, używając testu zgodności chi-kwadrat, czy liczby te rzeczywiście pochodzą z rozkładu wykładniczego.

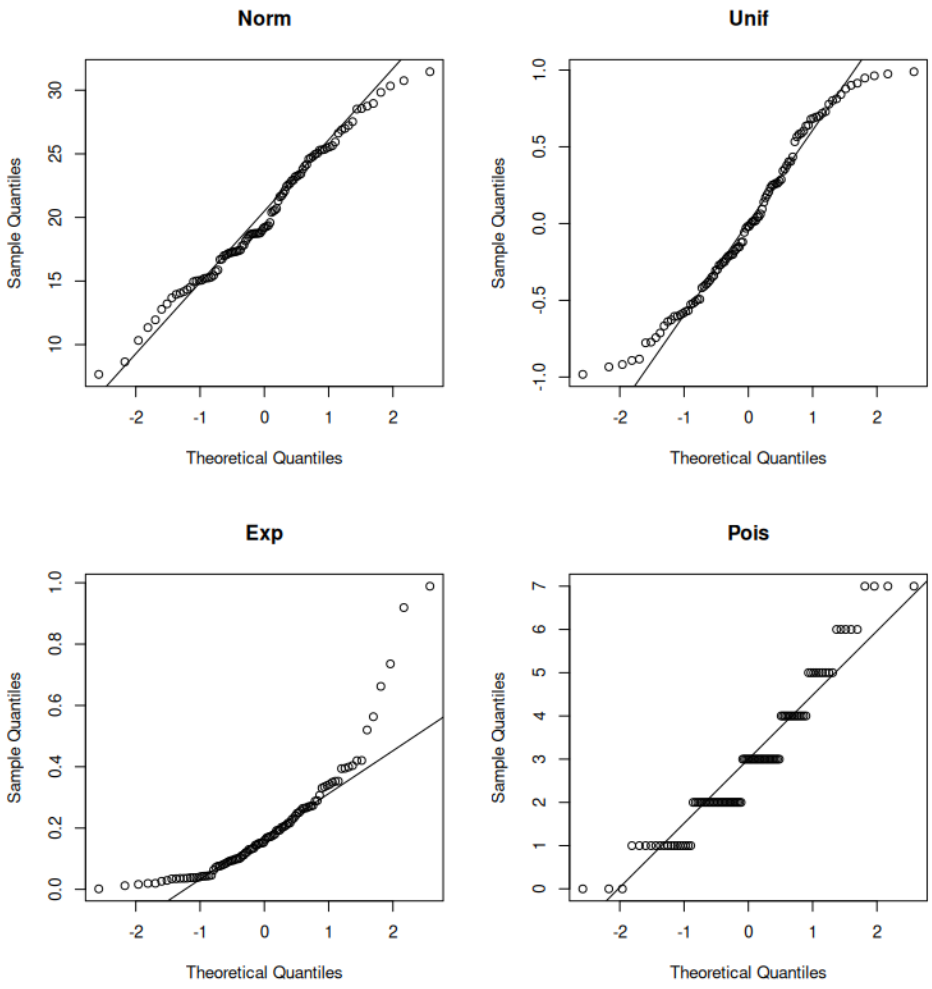
WSKAZÓWKA: Aby przeprowadzić test zgodności chi-kwadrat musimy najpierw dane zdyskretyzować, rozdzielając je do odpowiedniej liczby klas. Wiemy, że pożądanym jest, aby prawdopodobieństwa klas p_j^0 były przynajmniej w przybliżeniu równe i by był spełniony warunek, że wszystkie $np_j^0 \geq 5$. Zdecydujmy się na równe prawdopodobieństwa wszystkich klas i wynoszące $\frac{1}{20}$; zagwarantuje to, że $np_j^0 = 10 \geq 5$. Zatem chcemy mieć 20 klas. Jeśli za końce klas (skoro chcemy 20 klas, to potrzebujemy 21 punktów końcowych) przyjmiemy odpowiednie kwantyle rozkładu wykładniczego z parametrem λ, wyznaczonym metodą największej wiarygodności,

```
> konce.przedzialow = qexp(seq(0,1,length.out=21), estymator.lambdy)
```

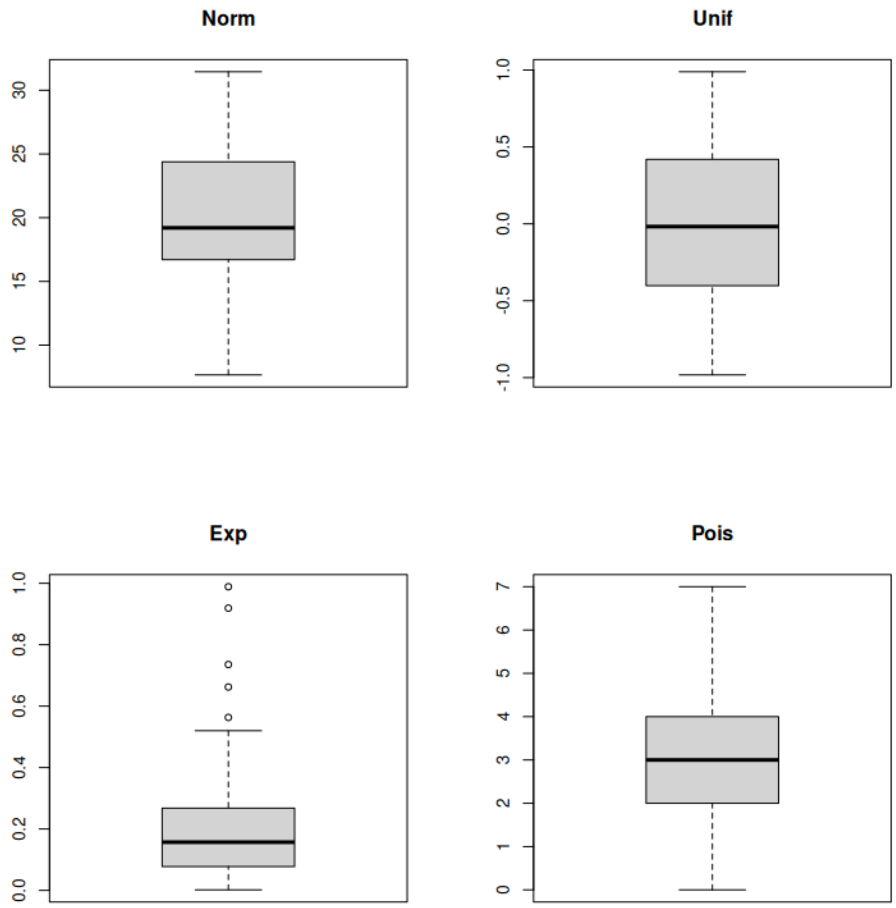
to dla każdej klasy rzeczywiście będziemy mieć $p_j^0 = \frac{1}{20}$. Pozostaje zliczyć ile obserwacji wpadło do poszczególnych klas. Możemy to zrobić używając funkcji `cut()` i `table()`:

```
> licznosci.klas = table(cut(x=probka, breaks=konce.przedzialow))
```

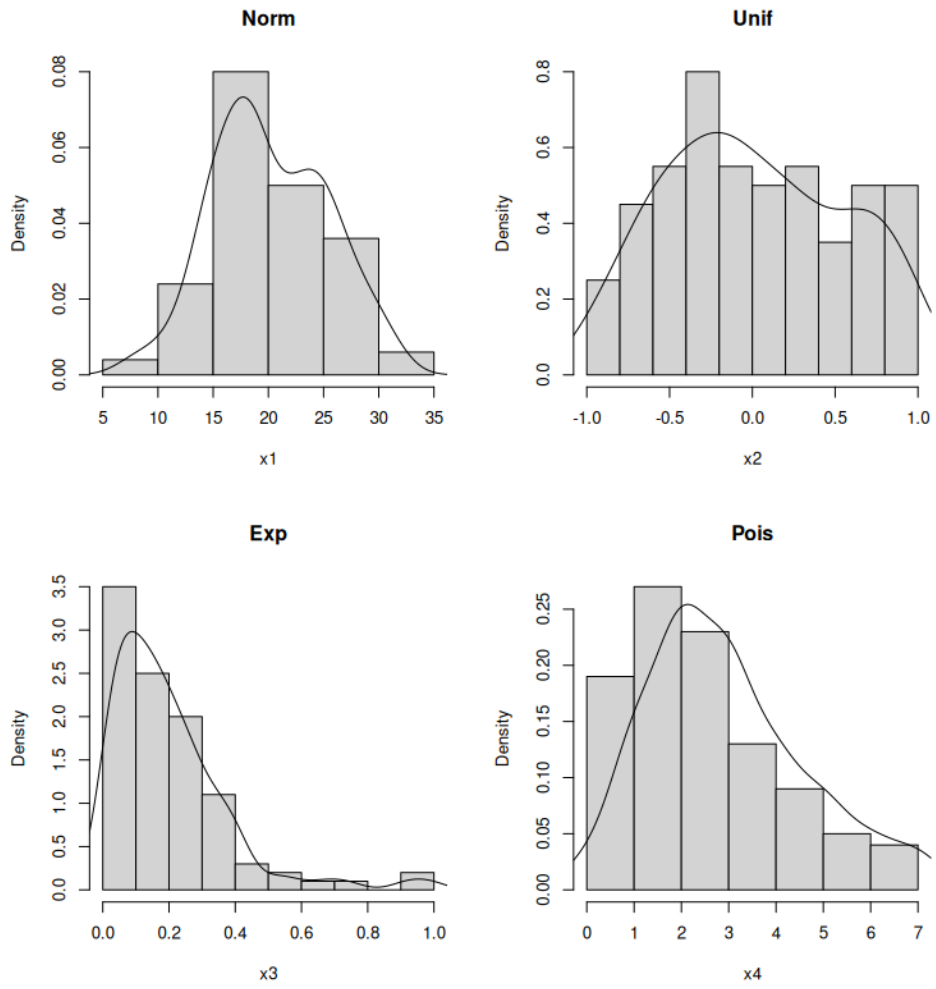
Następnie, używając testu Kołmogorowa-Smirnowa, sprawdzić czy wygenerowane liczby pochodzą z rozkładu wykładniczego o parametrze $\lambda = 2$. Przyjąć poziom istotności $\alpha = 0.01$.



Rysunek 17: Wykresy normalności



Rysunek 18: Wykresy skrzynkowe



Rysunek 19: Histogramy częstości z naniesionymi jądrowymi estymatorami gęstości

Uwagi

- Komentarze dotyczące funkcji `chisq.test`, `ks.test` zaczerpnięte ze skrótów wykładu
- Opisy wykresów zaczerpnięte z skrótu wykładu 13

6. Jednoczynnikowa analiza wariancji

Zadanie 6.1. W katalogu bazowym base znajduje się plik iris a w nim między innymi następujące zmienne:
Sepal.Width - szerokość działki kielicha,
Spieces - odmiana irysa.

a) Chcemy zweryfikować czy szerokość działki kielicha zależy od odmiany. Jakie postawimy hipotezy do testowania? Wykonać wykresy skrzynkowe w grupach by wstępnie ocenić sytuację.

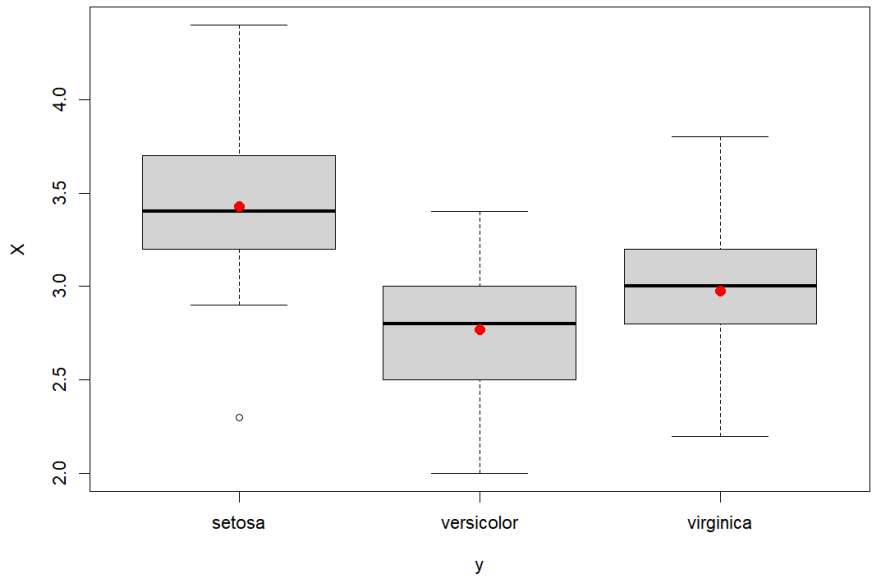
b) Sprawdzić czy są spełnione założenia analizy wariancji.

c) Stwierdzić, czy szerokość działki kielicha irysa zależy od jego odmiany. Jeśli tak, przeprowadzić testy porównań wielokrotnych.

Rozwiązanie: Oznaczmy wstępnie potrzebne zmienne pochodzące z pliku `iris`: `X = iris$Sepal.Width`, `y = iris$Species`. Przed przejściem do testowania warto sprawdzić, czy **obiekt y jest typu factor**. Stosujemy do tego funkcję logiczną `is.factor(y)`. Jeżeli uzyskamy wynik negatywny, możemy łatwo zmapować ten obiekt na pożądany typ za pomocą wywołania funkcji `y = as.factor(y)`. W ten sposób będziemy mieli pewność, że y zawiera dane kategoryczne reprezentujące zmienne objaśniające.

a) Zdefiniujmy najpierw odpowiednie hipotezy. Z danych z pliku `iris` wynika, że mamy do czynienia z trzema różnymi odmianami irysa. Są to zmienne objaśniające (czynniki). Szerokość działki kielicha natomiast jest zmienną objaśnianą. Hipoteza zerowa zakłada, że średnie szerokości kielicha (ozn. odpowiednio μ_1, μ_2, μ_3) są **takie same** dla każdej odmiany irysa: $H_0 : \mu_1 = \mu_2 = \mu_3$. Hipoteza alternatywna natomiast zakłada, że istnieją $i, j \in \{1, 2, 3\}$ takie, że $\mu_i \neq \mu_j$, czyli formalnie: $H_1 : \exists_{i,j \in \{1,2,3\}} \mu_i \neq \mu_j$.
Wykonujemy wstępną analizę danych, posługując się wykresami skrzynkowymi. Najpierw jednak musimy obliczyć średnie próbkowe we wszystkich grupach. Stosujemy tu funkcję `tapply`, a jej wywołanie wygląda w następujący sposób: `srednie = tapply(X, y, mean)`. Funkcja ta odpowiednio grupuje wartości z X według czynników z y. W ten sposób będziemy mieli policzone średnie szerokości działek kielicha dla każdej odmiany. Możemy teraz przystąpić do narysowania wykresów skrzynkowych. Stosujemy funkcję `boxplot` i wprowadzamy do niej formułę `X~y`. Informujemy w ten sposób, że chcemy narysować wykresy skrzynkowe osobno dla każdej z grup. Dorysowujemy również punkty reprezentujące średnie wartości szerokości działek kielicha dla każdej odmiany irysa za pomocą funkcji `lines(1:3, srednie, pch=20, type="p", col='red', cex=2)`.
Na powstałym wykresie widać, że dla odmiany irysa *setosa* kwiaty osiągają **większą szerokość** działki kielicha niż dla pozostałych odmian.

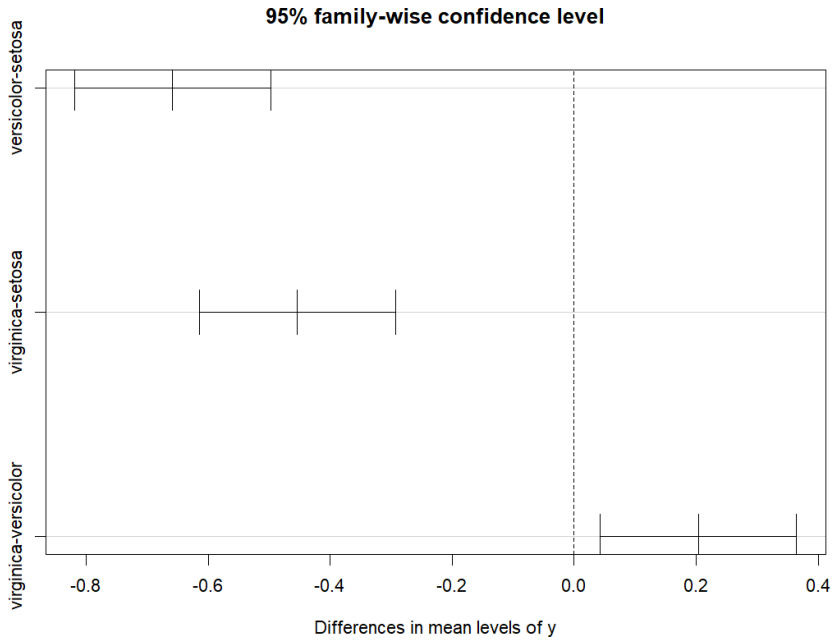
b) Chcemy teraz sprawdzić, czy spełnione są założenia analizy wariancji, czyli, czy wariancje dla każdej z grup są takie same oraz czy wszystkie rozkłady szerokości działki kielicha są normalne. Wpierw wykonajmy dla danych **test Shapiro-Wilka**, który zbada czy rozkłady są normalne. Definiujemy hipotezy zerową i alternatywną:
H_0 : *wszystkie rozkłady są normalne*, H_1 : *istnieje grupa, dla której rozkład nie jest normalny.* Test można wywołać na kilka sposobów. Interesuje nas tylko p-value, więc najbardziej praktycznym rozwiązaniem będzie wyodrębnienie z wyników testów samej wartości p-value. Wywołanie to ma postać `tapply(X, y, function(x)shapiro.test(x)$p.value)`. Tutaj również stosujemy funkcję `tapply`, która podobnie jak w przypadku obliczania średnich, również rozdzieli dane X na 3 odrębne grupy i na każdej wykona test Shapiro-Wilka. Domyślnie testy są przeprowadzane na **poziomie istotności 0.05**. Otrzymujemy następujące wyniki p-value: $0.2715264, 0.3379951, 0.1808960$. Dla żadnej grupy, czyli dla żadnej odmiany irysa **nie mamy podstaw do odrzucenia hipotezy zerowej**.
Przejdźmy teraz do sprawdzenia, czy wariancje każdej z grup są takie same. Czyli hipoteza zerowa wygląda w następujący sposób: $H_0 : \sigma_1^2 = \sigma_2^2 = \sigma_3^2$, a alternatywna: H_1 : istnieje $i, j \in \{1, 2, 3\}$



Rysunek 20: Wykresy skrzynkowe szerokości działki kielicha dla każdej odmiany irysa)

$\sigma_i^2 \neq \sigma_j^2$. Przeprowadzamy w tym celu test Bartletta na poziomie istotności 0.01. Jego wywołanie ma postać **`bartlett.test(X, y)`**. Z testu otrzymujemy p-value na poziomie 0.3515, czyli nie mamy podstaw do odrzucenia hipotezy zerowej.

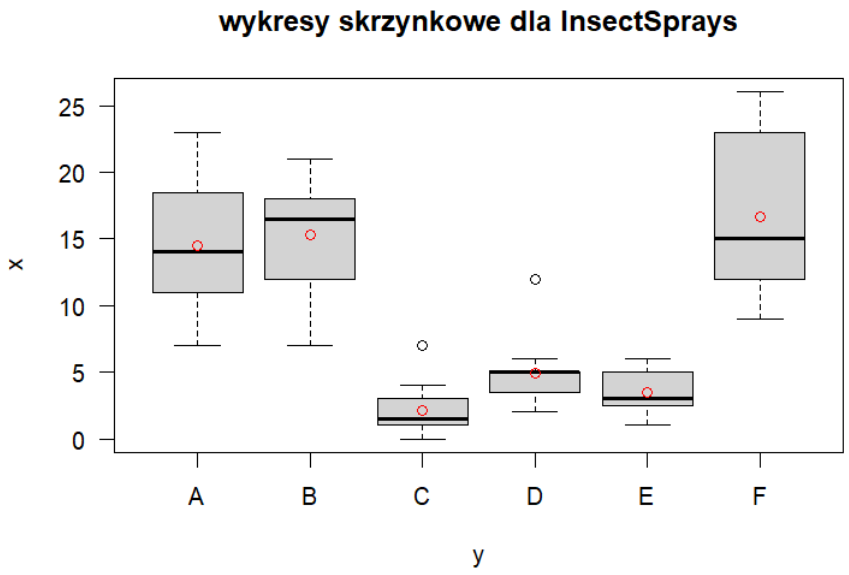
c) Na podstawie przeprowadzonych testów stwierdzamy, że są spełnione założenia analizy wariancji. Możemy więc przejść do sprawdzenia, czy szerokość działki kielicha zależy od jego odmiany, czyli, czy odrzucamy hipotezę zerową zdefiniowaną w pkt. a).
Najpierw konstruujemy model liniowy na podstawie danych X i y za pomocą funkcji lm: **`model = lm(X~y)`**. Dla tak skonstruowanego modelu przeprowadzam analizę wariancji: **`anova(model)`**. Szukane p-value mówiące, czy odrzucamy hipotezę zerową znajdujemy pod "Pr(>F)". Jak widać, jego wartość jest ekstremalnie mała, wynosi ona bowiem 2.2e-16, czyli tym samym **odrzucamy hipotezę zerową, a więc szerokość działki kielicha zależy od jego odmiany**. Przeprowadzamy więc porównania wielokrotne. Można to zrobić na dwa sposoby: używając funkcji **`pairwise.t.test(X, y, p.adjust="bonf")`**, czyli zastosowanie procedury Bonferroniego lub użycie procedury Tukey'a **`tukey = TukeyHSD(aov(model), conf.level = 0.95)`**, która jest szczególnie polecana dla równolicznych grup. Dla drugiej procedury wyniki możemy również zaprezentować na wykresie **`plot(tukey`**, z którego łatwo możemy określić różnice pomiędzy średnimi. Przez wykres w punkcie 0 przechodzi pionowa linia. Zaznaczone są również przedziały ufności różnic średnich dla każdej z par. Jeżeli nie przecinają opisanej pionowej linii, to różnica pomiędzy średnimi dla rozpatrywanego przedziału jest istotna. Jak widać na załączonym wykresie, żaden przedział ufności **nie przecina tej linii, w związku z tym średnie szerokości dla każdej z odmian istotnie się od siebie różnią**.



Rysunek 21: Wykres różnic średnich dla każdej pary odmian irysa

Pełny kod rozwiązania:

```
# 6.1

X = iris$Sepal.Width
y = iris$Species

is.factor(y) # sprawdzam, czy y to zmienne typu factor - bardzo ważne!!!
# y = as.factor(y) - w razie co, jakby nie była factor, to rzutujemy na niego

srednie = tapply(X, y, mean)

boxplot(X~y)
lines(1:3, srednie, pch=20, type="p", col='red', cex=2)

tapply(X, y, shapiro.test)
tapply(X, y, function(x)shapiro.test(x)$p.value)
simplify2array(tapply(X, y, function(x)shapiro.test(x)[1:2]))

bartlett.test(X, y)

model = lm(X ~ y)
anova(model)

pairwise.t.test(X, y, p.adjust="bonf")

tukey = TukeyHSD(aov(model), conf.level = 0.95)
tukey
plot(tukey)
```



Rysunek 22: Wykresy skrzynkowe dla każdego rodzaju Spray'u (dla każdego czynnika)

Zadanie 6.2. W katalogu bazowym `base` znajduje się plik `InsectSprays`, zawierający następujące zmienne:

- `count` – liczba insektów w wydzielonych obszarach eksperymentalnych,
- `spray` – rodzaj stosowanego w danym obszarze środka owadobójczego.

a) Chcemy zweryfikować, czy przeżywalność insektów zależy od rodzaju stosowanego środka owadobójczego. Jakie postawimy hipotezy do testowania? Wykonać wykres średnich w grupach, by wstępnie ocenić sytuację.

b) Sprawdzić, czy są spełnione założenia analizy wariancji.

c) Jeśli założenia nie są spełnione, to zaproponować odpowiednie przekształcenie zmiennej `count`, tak by w nowym modelu przynajmniej w przybliżeniu założenia analizy wariancji były spełnione. **Wskazówka:** Zalecane przekształcenie to pierwiastkowanie, ponieważ zmienna `count` jest typu zliczającego.

d) Na podstawie nowego modelu stwierdzić, czy liczba insektów zależy od rodzaju stosowanego środka owadobójczego (opisać dokładnie używany model). Jeśli zależy, to przeprowadzić testy porównań wielokrotnych.

Rozwiązanie: Pierwszym krokiem było załadowanie kolumny identyfikującej czynniki (*sprays*) oraz zmienną objaśnianą (*count*) ze zbioru *InsectSprays*.

a) W tym podpunkcie wstępnie zbadaliśmy zależność zmiennej objaśnianej od czynników, analizując średnie dla różnych grup oraz wykresy skrzynkowe. Do wyznaczenia średnich użyliśmy omówionej wcześniej funkcji `tapply`, a następnie nakreśliliśmy wykresy skrzynkowe dla każdego typu Spray'a. Wynik przedstawiamy na Rysunku 22, wraz z zaznaczonymi średnimi na czerwono. Na podstawie wykresu nietrudno jest wywnioskować, iż są podstawy do badania zależności przeżywalności insektów od rodzaju stosowanego środka. W tym celu posłużymy się **jednoczynnikową analizą wariancji (ANOVA)**.

b) Najpierw jednak musimy sprawdzić, czy są spełnione założenia analizy wariancji. Jak w poprzednim zadaniu, analizować będziemy normalność rozkładu zmiennej odpowiedzi oraz równość wariancji w

	A	B	C	D	E	F
statistic	0.9576	0.9503	0.8591	0.7506	0.9213	0.8848
p.value	0.7487	0.6415	0.0476	0.0027	0.2967	0.1009

Tabela 1: Wyniki testu normalności (statystyki i wartości p dla grup A–F)

	A	B	C	D	E	F
statistic	0.9679	0.9319	0.9391	0.8491	0.9131	0.9071
p.value	0.8876	0.4003	0.4868	0.0358	0.2341	0.1957

Tabela 2: Wyniki testów normalności dla poszczególnych grup (po transformacji)

grupach.

- **Normalność rozkładu** - zważając na to, iż n nie jest aż tak bardzo małe, wykonaliśmy test **Shapiro-Wilka** dla każdego czynnika (na poziomie istotności 0.05). Wyniki przedstawiamy w tabeli 1. Jak widać, nie wszystkie grupy spełniają założenia o normalności, ale możemy przymknąć na to oko, o ile drugie z założeń będzie spełnione.
- **Równość wariancji** - Skoro nie wszystkie wartości zmiennej odpowiedzi mają rozkład normalny. wykonamy test Levene'a (alternatywa to test Bartletta przy założeniu, że $n \geq 10$ oraz normalności rozkładów w grupach). Przekazujemy argument `center = 'mean'`, gdyż za punkt odniesienia odchyleń absolutnych przyjmujemy wartość średnią (inne opcje to `'median'`, `'trimmed'`). Wyniki poniżej:

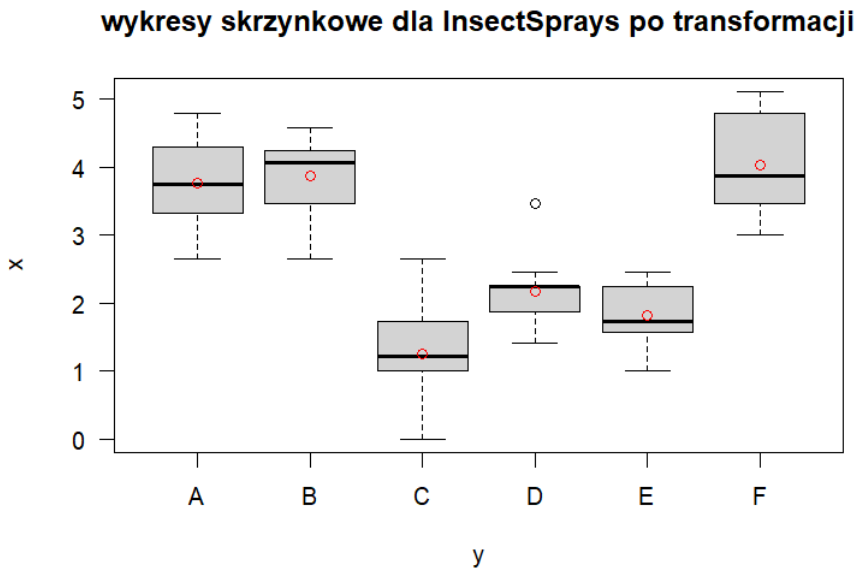
```
Levene's Test for Homogeneity of Variance (center = mean)
      Df F value    Pr(>F)
group  5  6.4554 6.104e-05 ***
      66
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

  - Statystyka testowa: $F = 6{,}455$
  - Stopnie swobody: $df_1 = 5$, $df_2 = 66$
  - Wartość $p = 6{,}104 \times 10^{-5}$

  Ponieważ $p < 0{,}01$, odrzucamy hipotezę zerową o równości wariancji w grupach. Oznacza to, że założenie homogeniczności wariancji nie jest spełnione, co może wpływać na poprawność klasycznej analizy wariancji (ANOVA). Zastosujemy zatem transformację danych (pierwiastkowanie).

c) W tym podpunkcie spróbowaliśmy zastosować przekształcenie pierwiastkowania wartości zmiennej odpowiedzi. Dla stransformowanych danych naszkicowaliśmy wykres skrzynkowy (Rysunek 23). Ponadto przeprowadziliśmy ponownie test Shapiro-Wilka. Wyniki przedstawiamy w Tabeli 2. Porównując p-value przed transformacją i po nietrudno stwierdzić, iż normalność rozkładów poszczególnych grup nieco się polepszyła. Nadal jednak dla grupy D musimy odrzucić hipotezę o normalności rozkładu zmiennej odpowiedzi dla tego czynnika, zatem do analizy równości wariancji poszczególnych grup użyjemy ponownie testu Levene'a. Wyniki predstawiwamy poniżej:

```
Levene's Test for Homogeneity of Variance (center = mean)
      Df F value Pr(>F)
group  5  1.0683  0.386
```



Rysunek 23: Wykres skrzynkowy dla każdego rodzaju Spray'u, po transformacji

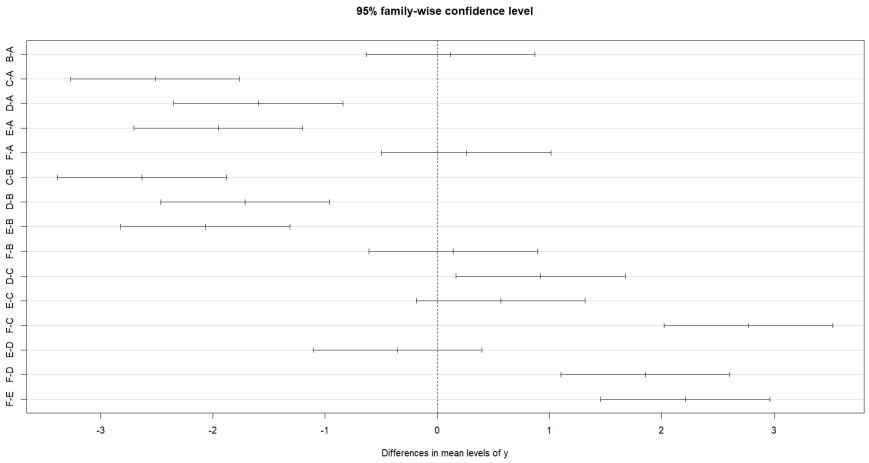
66

p-value jest większe od 0.01, zatem nie mamy podstaw do odrzucenia hipotezy zerowej równości wariancji wszystkich grup. Więc po transformacji (w przybliżeniu) wszystkie założenia analizy wariancji są spełnione i możemy przejść do konstrukcji modelu.

d) Aby w R przeprowadzić jednoczynnikową analizę wariancji, tak jak w poprzednim zadaniu, przypisujemy do jakiejś zmiennej obiekt `lm(zmienna.odpowiedzi~czynnik)` i przekazujemy ją do funkcji `anova`. Wyniki poniżej:

```
Analysis of Variance Table

Response: x
          Df Sum Sq Mean Sq F value    Pr(>F)
y          5 88.438 17.6876  44.799 < 2.2e-16 ***
Residuals 66 26.058  0.3948
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- Liczba obserwacji: $n = 72$
- Liczba grup (rodzajów sprayów): $k = 6$
- Stopnie swobody:
  - dla czynnika (sprayu): $df_1 = k - 1 = 5$
  - dla reszt: $df_2 = n - k = 66$
  - łącznie: $df = df_1 + df_2 = 71$
- **Suma kwadratów:**
  - SSA – suma kwadratów między grupami (czynnik y): $SSA = 88{,}438$



Rysunek 24: Wykres przedstawiający przedziały ufności różnicy średnich dla każdej pary czynników

  - SSE – suma kwadratów błędu (reszt): $SSE = 26{,}058$
  - SST – całkowita suma kwadratów: $SST = SSA + SSE = 114{,}496$
- **Średnie kwadraty:**
  - $MSA = \frac{SSA}{df_1} = \frac{88{,}438}{5} = 17{,}6876$
  - $MSE = \frac{SSE}{df_2} = \frac{26{,}058}{66} = 0{,}3948$
- **Statystyka F:**

$$F = \frac{MSA}{MSE} = \frac{17{,}6876}{0{,}3948} = 44{,}799$$

- **Wartość p:** $p < 2{,}2 \times 10^{-16}$ (bardzo istotna statystycznie)

Zatem odrzucamy hipotezę zerową, mówiącą że liczba insektów nie zależy od rodzaju środka. Z tego powodu wykonujemy test porównań wielokrotnych, używając procedury Tukeya (gdyż rozważamy średnią w grupach o tej samej liczności, co sprawdziliśmy używając funkcji `table` na zmiennej objaśnianej). Analizę wyników tego testu przeprowadziliśmy na podstawie wykresu, który przedstawiliśmy na Rysunku 24. Z wykresu można odczytać, iż jedynie B-A, F-A, F-B, E-C, E-D to pary czynników, dla których 95% przedział ufności różnicy ich wartości średnich zawiera $0 \implies$ można stwierdzić, iż tylko one w podobny sposób wpływają na przeżywalność owadów.

Pełny kod rozwiązania:

```
X = InsectSprays$count
y = InsectSprays$spray

srednie = tapply(X, y, mean)

par(mfrow=c(1, 6))

boxplot(X~y)
lines(1:6, srednie, pch=20, type="b", cex=2)

simplify2array(tapply(X, y, function(x)shapiro.test(x)[1:2]))

leveneTest(X~y, center=mean) # nie potrzebuje założenia o normalności


# przekształcanie - pierwiastkowanie danych
X = sqrt(X)
srednie = tapply(X, y, mean)
boxplot(X~y)
lines(1:6, srednie, pch=20, type="b", cex=2)
simplify2array(tapply(X, y, function(x)shapiro.test(x)[1:2]))
leveneTest(X~y, center=mean)

model = lm(X ~ y)
anova(model)

pairwise.t.test(X, y, p.adjust="bonf")

tukey = TukeyHSD(aov(model), conf.level = 0.95)
tukey
plot(tukey)

tapply(X, y, function(x){qqnorm(x); qqline(x)})
```