Wykład 8 - Zadania 2 i 5

Krystian Baran 145000

27 kwietnia 2021

Spis treści

1 Zadanie 2 - Funkcje testów w R

W języku programowania R istnieją różne funkcji do wykonywania testów, natomiast zatrzymamy się na najważniejszych.

1.1 t.test()

Funkcja do wykonania testu t-Studenta dla którego zakłada się że dane pochodzą z rozkładu normalnego z parametrami nieznanymi. Parametry tej funkcj są następujące:

- **x** - wektor wartości próby
- **y** - wektor wartości próby do porównania z próbą x. Opcjonalny
- **alternative** - specyfikacja hipotezy alternatywnej przyjmujące wartości: "two.sided", "greater", "less"
- **mu** - wartość prawdziwej wartości oczekiwanej lub różnica między wartościami oczekiwanymi dla dwóch prób
- **paired** - logiczna wartość dla "paired" testu
- **var.equal** - logiczna mówiąca czy traktować wariancje jako takie same lub nie. Gdy FALSE korzysta się z aproksymacja Welcha.
- **conf.equal** - confidence level
- **formula** - "lhs" dla numerycznej zmiennej oddającej wartości, "rhs" dla two pionową korespondencji grup.
- **data** - Opcjonalna macierz z wartościami użytych do polu *formula*
- **subset** - Opcjonalny wektor z podzbiorem obserwacji do wykorzystania w teście
- **na.action** - funkcja definiująca co się dzieje gdy napotkane zostają wartości *NA*

Oddawane przez tą funkcje wartości są następujące:

- **statistic** - wartość statystyki t
- **parameter** - stopnie swobody statystyki t
- **p.value** - *p value* wykonanego testu
- **conf.int** - przedział ufności dla wartości oczekiwanej
- **estimate** - estymowana wartość oczekiwana lub różnica wartości oczekiwanych dla testu dwóch prób

- **null.value** - podana wartość *mu*
- **stderr** - standardowy błąd wartości oczekiwanej, używany jako mianownik statystyki t
- **alternative** - opis hipotezy alternatywnej
- **method** - typ wykonanego testu t
- **data.name** - imię podanej macierz pod *data*

1.2 wilcox.test()

Test Wilcoxona wykorzystany jest do badania wartości oczekiwanej jak w teście t-Studenta ale nie zakłada się że próba ma rozkład normalny. Funkcja ta przyjmuje następujące parametry:

- **x** - wektor liczb na podstawie której będzie test prowadzony
- **y** - wektor liczb w przypadku testu dwóch prób
- **mu** - wartość oczekiwana hipotezy zerowej
- **paired** - logiczna dla testu "paired"
- **exact** - logiczna mówiąca czy dokładna wartość *p value* powinna być liczona
- **correct** - logiczna mówiąca czy *p value* powinna być poprawiona ze względu na ciągłość
- **conf.int** - logiczna mówiąca czy powinien zostać liczony przedział ufności
- **conf.level** - poziom ufności testu
- **formula** - "lhs" dla numerycznej zmiennej oddającej wartości, "rhs" dla two pionową korespondencji grup.
- **data** - Opcjonalna macierz z wartościami użytych do polu *formula*
- **subset** - Opcjonalny wektor z podzbiorem obserwacji do wykorzystania w teście
- **na.action** - funkcja definiująca co się dzieje gdy napotkane zostają wartości *NA*

Funkcja ta oddaje podobne wartości jak test t-Studenta:

- **statistic** - wartość statystyki z imieniem opisującym
- **parameter** - parametry dokładnego rozkładu statystyki testowej
- **p.value** - *p value* wykonanego testu

- **null.value** - podana wartość *mu*
- **alternative** - opis hipotezy alternatywnej
- **method** - typ wykonanego testu
- **data.name** - imię podanej macierz pod *data*
- **conf.int** - przedział ufności dla wartości oczekiwanej
- **estimate** - estymowana wartość oczekiwana lub różnica wartości oczekiwanych dla testu dwóch prób

1.3 var.test()

Test ten jest testem F Snedecora dla porównania wariancji pomiędzy dwoma populacjami. Funkcja ta przyjmuje następujące wartości:

- **x, y** - wektory liczb dla których przeprowadzony jest test
- **ratio** - hipotetyczny "ratio" pomiędzy badanymi wariancjami
- **alternative** - hipoteza alternatywna, przyjmuje wartości: "two.sided", "greater", "less"
- **conf.level** - poziom ufności testu
- **formula** - "lhs" dla numerycznej zmiennej oddającej wartości, "rhs" dla two pionową korespondencji grup.
- **data** - Opcjonalna macierz z wartościami użytych do polu *formula*
- **subset** - Opcjonalny wektor z podzbiorem obserwacji do wykorzystania w teście
- **na.action** - funkcja definiująca co się dzieje gdy napotkane zostają wartości *NA*

Funkcja ta zwraca listę typu "htest" zawierająca następujące komponenty:

- **statistic** - wartość statystyki F-test
- **parameter** - stopnie swobody rozkładu F dla testu
- **p.value** - *p value* wykonanego testu
- **conf.int** - przedział ufności dla wartości oczekiwanej
- **estimate** - estymowana wartość oczekiwana lub różnica wartości oczekiwanych dla testu dwóch prób
- **null.value** - "ratio" wariancji populacji podane
- **alternative** - opis hipotezy alternatywnej
- **method** - typ wykonanego testu
- **data.name** - imię podanej macierz pod *data*

1.4 ks.test()

Funckja do wykonania testu na dwóch próbach w celu sprawdzenia czy mają ten sam rozkład. Funkcja ta przyjmuje następujące wartosci:

- **x** - wektor wartości
- **y** - wektor wartości lub łańcuch opisujący dystrybuantę rozkładu
- **alternative** - hipoteza alternatywna, przyjmuje wartości: "two.sided", "greater", "less"
- **exact** - logiczna mówiąca czy dokładna wartość *p value* powinna być liczona
- **tol** - górny koniec przedziału dla błędu zaokrąglania
- **simulate.p.value** - logiczna mówiąca czy symulować *p value* według Monte Carlo
- **B** - liczba replikatów dla testu Monte Carlo

Funkcja ta zwraca listę typu "htest" zawierająca następujące komponenty:

- **statistic** - wartość statystyki testowej
- **p.value** - *p value* wykonanego testu
- **alternative** - opis hipotezy alternatywnej
- **method** - typ wykonanego testu
- **data.name** - imię podanej macierz pod *data*

2 Zadanie 4 - Studium przypadku

Pascal jest językiem programowania wysokiego poziomu, stosowanym często do oprogramowywania mikrokomputerów. W celu zbadania wskaźnika p zmiennych pascalowych typu tablicowego został przeprowadzony eksperyment. Dwadzieścia zmiennych zostało losowo wybranych ze zbioru programów pascalowych i liczba X zmiennych typu tablicowego została odnotowana. Celem poznawczym jest zweryfikowanie hipotezy, że pascal jest językiem o większej wydolności (tj. ma większy udział zmiennych typu tablicowego) niż algol, dla którego, jak pokazało doświadczenie, jedynie 20% zmiennych jest typu tablicowego.

a) Skonstruować test statystyczny do zweryfikowania postawionej hipotezy.

b) Znaleźć α dla zbioru odrzuceń $X \geqslant 8$.

c) Znaleźć α dla zbioru odrzuceń $X \geqslant 5$.

d) Znaleźć β dla zbioru odrzuceń $X \geqslant 8$, jeżeli $p = 0,5$ (doświadczenie pokazuje, że około połowa zmiennych w programach pascalowskich jest typu tablicowego).

e) Znaleźć β dla zbioru odrzuceń $X \geqslant 5$, jeżeli $p = 0,5$.

f) Który ze zbiorów odrzuceń $X \geqslant 8$ czy $X \geqslant 5$ jest bardziej pożądany, jeżeli minimalizowany jest:

   A) błąd I rodzaju?

   B) błąd II rodzaju?

g) Znaleźć jednostronny zbiór odrzuceń postaci $X \geqslant a$, tak aby poziom ufności był w przybliżeniu równy $\alpha = 0,01$.

h) Dla zbioru odrzuceń wyznaczonego w poprzednim punkcie znaleźć moc testu, jeżeli $p = 0,4$.

i) Dla zbioru odrzuceń wyznaczonego w punkcie g) znaleźć moc testu, jeżeli $p = 0,7$.

2.1 a)

Oznaczmy jako X liczbę zmiennych typu tablicowego w Pascal z losowo wybranych, wtedy X ma rozkład dwumianowy z nieznanym parametrem p. Załóżmy że liczby typu tablicowego wylosowane z programu Algog ma także rozkład dwumianowy gdzie natomiast jest znany parametr $p = 0.2$. Wtedy hipoteza że Pascal jest językiem programowania o większej zdolności niż Algol, czyli że p liczb typu tablicowego w Pascal jest większa niż p dla liczb tablicowych w Algol. Jest to hipoteza alternatywa ponieważ wstępuje ostra nierówność, zatem stosując normy statystyki można wyznaczyć hipotezę zerowa.

H_0	$p \leqslant p_0 = 0.2$
H_1	$p > p_0 = 0.2$

Znany jest rozkład ale nie znany jest parametr p, musimy sprawdzić poniższy warunek aby móc zastosować statystykę.

$$0 < p_0 \mp \sqrt{\frac{p_0(1-p_0)}{n}} = 0.2 \mp \sqrt{\frac{0.16}{20}} < 1$$

Spełniony jest warunek zatem możemy zastosować statystykę która jest zbliżona do rozkładu $N(0,1)$:

$$Z = \frac{\overline{P}_n - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$$

2.2 b)

Zbiór odrzuceń jest zbiorem dla którego, jeżeli liczba liczb tablicowych się znajdzie odrzucamy hipotezę zerową tj że Pascal jest mniej wydajny. Zatem dla $X \geqslant 8$ zbiór krytyczny jest następujący:

$$R = \{8, 9, \ldots, 20\}$$

Wtedy błąd pierwszego rodzaju, zakładając że dla H_0 $p = 0.2$ jest następujący:

$$\alpha = P(U_n \in R|H_0) = P(X \geqslant 8) = 1 - P(X \leqslant 7)$$
$$\overset{R}{=} 1 - pbinom(7, 20, 0.2) \approx 0.03214266$$

2.3 c)

Podobnie jak w poprzednim podpunkcie wyznaczymy zbiór krytyczny:

$$R = \{5, 6, \ldots, 20\}$$

Wtedy, zakładając jak poprzednio, błąd pierwszego rodzaju wynosi:

$$\alpha = P(U_n \in R|H_0) = P(X \geqslant 5) = 1 - P(X \leqslant 4)$$
$$\overset{R}{=} 1 - pbinom(4, 20, 0.2) \approx 0.3703517$$

2.4 d)

Jak dla podpunktu b, zbiór wartości krytycznych jest następujący:

$$R = \{8, 9, \ldots, 20\}$$

Natomiast potrzebujemy obliczyć błąd drugiego rodzaju, to znaczy że zakładamy że hipoteza alternatywna jest prawdziwa, czyli że Pascal jest bardziej wydajnym programem, i zakładamy że $p = 0.5$. Wtedy szykane β wyraża się następującym wzorem:

$$\begin{aligned}\beta &= 1 - P(U_n \in R|H_1) = 1 - P(X \geqslant 8) = P(X \leqslant 7)\\ &\overset{R}{=} pbinom(7, 20, 0.5) \approx 0.131588\end{aligned}$$

Zatem β wynosi około 0.1326.

2.5 e)

Zbiór wartości krytycznych jest jak w podpunkcie c:

$$R = \{5, 6, \dots, 20\}$$

Natomiast obliczamy błąd drugiego rodzaju jak dla poprzedniego podpunktu, czyli:

$$\begin{aligned}\beta &= 1 - P(U_n \in R|H_1) = 1 - P(X \geqslant 5) = P(X \leqslant 4)\\ &\overset{R}{=} pbinom(4, 20, 0.5) \approx 0.005908966\end{aligned}$$

Zatem β wynosi około 0.0059.

2.6 f)

Nie rozwiązane.

2.7 g)

Szukamy wartość a taka aby błąd pierwszego rodzaju był równy 0.01. Zatem, korzystając z poprzednich podpunktów można tę wartość wyznaczyć:

$$\begin{aligned}\alpha = 0.01 &= 1 - P(X \leqslant a - 1)\\ P(X \leqslant a - 1) &= 0.99\\ a - 1 &= F^{-1}(0.99) \overset{R}{=} qbinom(0.99, 20, 0.2) = 8\\ a &= 9\end{aligned}$$

Zatem dla $X \geqslant 9$ błąd pierwszego rodzaju wynosi 0.01.

2.8 h)

Jak w poprzednich podpunktach obliczymy błąd drugiego rodzaju i na jego podstawie moc testu. Przyjmujemy wartość $p = 0.4$.

$$\beta = 1 - P(X \leqslant 8) \overset{R}{=} 1 - pbinom(8, 20, 0.4) \approx 0.4044013$$

Wtedy moc testu wynosi $1 - \beta = 0.5955987$.

2.9 i)

Nie rozwiązane. Jak w poprzednim podpunkcie obliczymy moc testu na podstawie błędu drugiego rodzaju przyjmując $p = 0.7$.

$$\beta = 1 - P(X \leqslant 8) \stackrel{R}{=} 1 - pbinom(8, 20, 0.7) \approx 0.9948618$$

Wtedy moc testu wynosi $1 - \beta = 0.005138162$.

3 Bibliografia

- https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/t.test
- https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/wilcox.test
- https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/var.test
- https://www.rdocumentation.org/packages/dgof/versions/1.2/topics/ks.test