Міністерство освіти і науки України

Національний університет «Львівська політехніка»

ПРАКТИЧНЕ МАШИННЕ НАВЧАННЯ. ПЕРЕВІРКА СТАТИСТИЧНИХ ГІПОТЕЗ

Звіт

до лабораторної роботи №2

з курсу «Методи і засоби комп’ютерного навчання»

Виконав:

студент гр. СПКм-12

Сергієнко В.Р.

Прийняв

Пукач А.І.

Львів 2014

Мета: оволодіти навиками по перевірці статистичних гіпотез за допомогою тесту Шапіро-Уілка, критерія Колмогорова-Смирнова, t-тесту Стьюдента, F-тесту Фішера, критерія згоди x_2 Пірсона в системі R.

Короткі теоретичні відомості

Особливості використання тесту Шапіро-Уілки в системі R.

Функція shapiro.test(x) виконує тест Шапіро-Уілки. Нуль-гіпотеза полягає в тому, що випадкова величина, вибірка *x* якої відома, розподілена за нормальним законом. Обсяг вибірки повинен бути не менше 3 і не більше 5000.

Особливості використання критерія Колмогорова-Смирнова в системіR.

Якщо *y* – числовий вектор, тоді виконується двовибірковий тест Колмогорова-Смирнова, що перевіряє нуль-гіпотезу про те, що *x* і *y* вибрані з одного безперервного розподілу.

Якщо *y* – рядок, що містить ім'я безперервного розподілу, тоді виконується одновибірковий тест Колмогорова-Смирнова, що перевіряє нуль-гіпотезу про те, що *x* має заданий безперервний розподіл

Особливості використання t-тесту Стьюдента в системі R.

Виконує одно-або двовибірковий *t*-тест. Одновибірковий *t*-тест призначений для перевірки рівності середнього значення нормально розподіленої генеральної сукупності деякого заданого значення в припущенні, що дисперсія не відома. Двовибірковий тест служить для порівняння двох середніх значень з нормально розподілених генеральних сукупностей в припущенні, що їх дисперсії рівні, хоча і не відомі.

Якщо *y* і *formula* не задані, тоді виконується одновибірковий тест, який перевіряє, що вибірка *x* має середнє, що дорівнює 0.

Якщо *paired = TRUE*, тоді повинні бути визначені і мати однакову довжину вектори *x* та *y*.

Значення *NA* і *NaN* з даних видаляються (якщо *paired = TRUE*, тоді при цьому видаляється відповідне значення з другої вибірки).

Якщо *var.equal = TRUE*, тоді для оцінки відхилення використовується об'єднана вибірка. За замовчуванням *var.equal = FALSE* та відхилення оцінюється окремо для кожної вибірки. При цьому відбувається належне коригування числа ступенів свободи.

F-тест Фішера

Виконується F-тест Фішера для перевірки на рівність стандартних відхилень двох нормально розподілених сукупностей.

Критерій згоди x2 Пірсона

Якщо *x* – вектор або матриця з одним стовпцем або одним рядком, а вектор *y* не заданий, тоді *x* розглядається як статистичний ряд (одномірна таблиця спряженості ознак), тобто *i*-а компонента вектора *x* містить кількість точок, які потрапили в *i*-й інтервал групування. В цьому випадку виконується тест на перевірку згоди даних заданим ймовірностям *p*.

Індивідуальне завдання

1. Використовуючи тест Шапіро-Уілки, перевірити, чи є нормально розподіленими характеристики квітів ірису (фрейм даних *iris*). Рівень значущості *a = 0.05*.

```
cat("\014")
dani=read.table("C:\Program Files\LAB_YEAR_5\MZKN\Lab_02\iris.txt",header=TRUE)
dani
x1=c(dani [1,1]);
x2=c(dani [1,2]);
x3=c(dani [1,3]);
x4=c(dani [1,4]);
for (i in 2:150)
{  x1=c(x1, dani [i,1]);
   x2=c(x2, dani [i,2]);
   x3=c(x3, dani [i,3]);
   x4=c(x4, dani [i,4]);
}       x1
x2
x3
x4
set.seed(0)
shapiro.test(x1)
Shapiro-Wilk normality test
dani:  x1
W = 0.9761, p-value = 0.01018
//Гіпотеза відкидається, p-value < α
set.seed(0)
shapiro.test(x2)
Shapiro-Wilk normality test
dani:  x2
W = 0.9849, p-value = 0.1012
//Гіпотеза повинна бути прийнята, p-value > α
set.seed(0)
shapiro.test(x3)
Shapiro-Wilk normality test
dani:  x3
W = 0.8763, p-value = 7.412e-10
//Гіпотеза відкидається, p-value < α
set.seed(0)
shapiro.test(x4)
Shapiro-Wilk normality test
dani:  x4
W = 0.9018, p-value = 1.68e-08
//Гіпотеза відкидається - p-value <α
```

3. Завантажити таблицю з файлу *allcountries.txt*, що містить інформацію про населення, площі та ряд інших характеристик сучасних держав. Вибрати з таблиці ті країни, для яких доступна інформація про населення та площі (немає відсутніх значень *NA*) і площа більше 10.0. Нехай *area_log = log10 (log10 (area))*, *population_log = log10 (log10 (population))*. Побудувати лінійну регресію (використовуючи функцію *lm*) для залежності *population_log* від *area_log*. Використовуючи тест Колмогорова-Смирнова, перевірити гіпотезу про те, що *population_log* і *f(area_log)*, де *f()* – побудована регресійна функція, належать до одного безперервного розподілу. Рівень значущості *a = 0.05*.

```
cat("\014")
read=read.table("C:\Program
Files\LAB_YEAR_5\MZKN\Lab_02\allcountries.txt",header=TRUE)
names(read)=c("Num","Country","Population","Area","GDP","Index")
count=0;
l1=c(read[1,3]);
l2=c(read[1,4]);
k = 0;
for (i in 2:222)
{
 read[i,3]
 if((read[i,4]>10000)&&(read[i,4]!=0))
 {
  k=k+1;
  l1=c(l1,read[i,3]);
  l2=c(l2,read[i,4]);
 }
}
count
l1
```

```
[1] 1319498000 1319498000 1169016000  302425000  231627000  186736000
 [7]  160757000  158665000  148093000  142499000  127750000  103263388
[13]   88706300   87375000   82314900   77127000   75498000   74877000
[19]   71208000   64102140   62828706   62636000   60209500   59131287
[25]   48798000   48577000   48472700   46205000   45116894   43964602
[31]   40454000   39531000   38560000   38125479   37538000   33858000
[37]   32977500   31224000   30884000   28993000   28196000   27903000
[43]   27657000   27372000   27199388   27145000   24735000   23790000
[49]   23478000   22910000   22389000   21438000   21397000   21018897
[55]   19929000   19683000   19299000   19262000   18549000   17024000
[61]   16598074   16374716   15422000   14784000   14444000   14226000
[67]   13925000   13354000   13349000   13341000   12379000   12337000
[73]   11922000   11268000   11147000   10781000   10623000   10457000
[79]   10327000   10306709   10030000    9858000    9760000    9725000
[85]    9689000    9598000    9525000    9370000    9150000    9033000
[91]    8699000    8508000    8467000    8361000    7639000    7484000
[97]    7161000    7106000    6857000    6736000    6585000    6331000
[103]   6160000    6127000    5924000    5866000    5859000    5603000
[109]   5550000    5390000    5317000    5310000    4965000    4851000
[115]   4770000    4555000    4468000    4395000    4380000    4343000
[121]   4234925    4230000    4099000    3935000    3794000    3768000
```

```
[127]   3750000   3390000   3343000   3340000   3190000   3124000
[133]   3002000   2851000   2714000   2629000   2595000   2277000
[139]   2074000   2038000   2030000   2008000   1882000   1709000
[145]   1695000   1342409   1331000   1155000   1141000    841000
[151]    839000    833000    738000    658000    598000    507000
[157]    496000
```

l2

```
[1]  9640821  9640821  3287263  9629091  1904569  8514877   880254   143998
 [9]   923768 17098242   377873  1958201   300000   331689   357022  1104300
[17]  1001449   783562  1648195   551500   513115  2344858   242900   301318
[25]   676578  1221037    99538   603700   505992  1138914   945087  2780400
[33]  2505813   312685   580367  2381741  9970610   446550   241038   438317
[41]   147181  1285216   912050   447400   329847   652090  2149690   120538
[49]   238533    36188   527968   238391   801590    83858   185180   587041
[57]    65610   322463   475442  1246700   756096    41528  2724900   274000
[65]   181035  1267000   118484   108889   390757   283561   196722  1240192
[73]   752618   110861   131957  1284000    91982    30528   163610    78866
[81]    93032    88361    48671    26338   207600    27750  1098581   245857
[89]   449964   112622   637657    27834    86600    83858   110912    41284
[97]    22145   112088    21041   143100    56785   462840  1759540   406752
[105]   89342    71740   236800   130000    43094    49033   199900   338145
[113]  488100   117600   385155    56538    51100    69700    83600   622984
[121]   70273   270534    10400    51197    33851   342000   111369    65300
[129]   75517   175016    28748  1025520    29800    17818    10991  1564116
[137]  309500    64600   824292    25713    20256    30355   581730    11295
[145]   36125    45100   267668    14874    17364    11000    18274    23200
[153]  214969    47000    13812    28051    28896
```

area=log10(log10(l2))

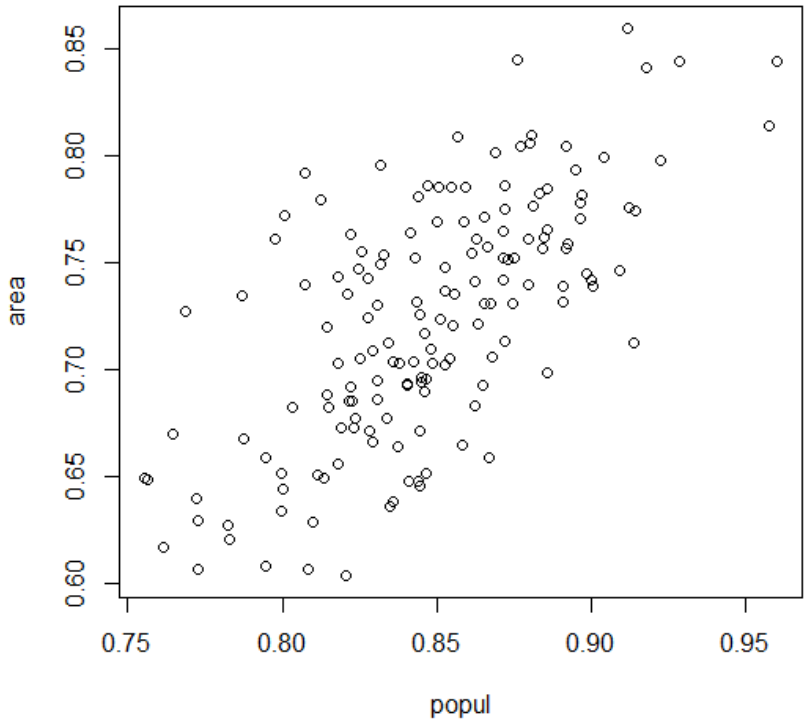
popul=log10(log10(l1))

area

popul

```
[1] 0.9600143 0.9600143 0.9575029 0.9284275 0.9224550 0.9175700 0.9141405
 [8] 0.9138393 0.9122505 0.9113607 0.9088259 0.9038464 0.9002554 0.8998964
[15] 0.8984772 0.8969232 0.8964124 0.8962146 0.8950095 0.8924771 0.8919920
[22] 0.8919177 0.8909609 0.8905226 0.8858361 0.8857247 0.8856720 0.8844945
[29] 0.8839077 0.8832697 0.8812112 0.8806386 0.8800207 0.8797388 0.8793523
[36] 0.8767754 0.8761149 0.8747420 0.8744664 0.8728723 0.8721672 0.8719027
[43] 0.8716783 0.8714157 0.8712552 0.8712044 0.8688390 0.8678441 0.8675064
[50] 0.8668793 0.8662894 0.8651741 0.8651249 0.8646659 0.8632923 0.8629712
[57] 0.8624614 0.8624118 0.8614341 0.8592021 0.8585407 0.8581866 0.8566166
[64] 0.8555066 0.8548941 0.8544933 0.8539290 0.8528221 0.8528122 0.8527964
[71] 0.8508107 0.8507203 0.8498093 0.8483029 0.8480140 0.8471196 0.8467234
[78] 0.8463004 0.8459642 0.8459113 0.8451787 0.8447125 0.8444430 0.8443460
[85] 0.8442459 0.8439911 0.8437848 0.8433412 0.8426979 0.8423491 0.8413263
[92] 0.8407224 0.8405909 0.8402478 0.8377801 0.8372180 0.8360058 0.8357936
[99] 0.8348106 0.8343191 0.8336924 0.8326029 0.8318430 0.8316937 0.8307564
[106] 0.8304823 0.8304490 0.8292021 0.8289364 0.8281176 0.8277353 0.8276984
[113] 0.8258102 0.8251554 0.8246801 0.8233756 0.8228290 0.8223616 0.8222645
[120] 0.8220235 0.8213069 0.8212737 0.8203774 0.8192112 0.8181663 0.8179691
[127] 0.8178318 0.8149265 0.8145230 0.8144971 0.8131666 0.8125598 0.8114014
[134] 0.8098961 0.8084547 0.8075209 0.8071383 0.8032770 0.8004977 0.7999746
[141] 0.7998570 0.7995310 0.7975874 0.7946792 0.7944302 0.7873106 0.7870478
[148] 0.7826576 0.7822781 0.7726734 0.7725976 0.7723690 0.7684943 0.7647906
[155] 0.7616799 0.7562563 0.7555305
```

plot(popul,area)




```
regress=lm(popul~area)
regress
Call:
lm(formula = popul ~ area)
Coefficients:
(Intercept)         area
      0.502        0.478
Call:
lm(formula = popul ~ area)
Residuals:
      Min        1Q    Median        3Q       Max
-0.080946 -0.018869  0.002005  0.017865  0.071287
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.50199    0.02950   17.02   <2e-16 ***
area         0.47797    0.04059   11.78   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.02909 on 155 degrees of freedom
Multiple R-squared:  0.4722,    Adjusted R-squared:  0.4688
F-statistic: 138.7 on 1 and 155 DF,  p-value: < 2.2e-16
summary(regress)
Pol2=function(x1,y1,n)
{       a=(n*sum(x1*y1)-sum(x1)*sum(y1))/(n*sum(x1^2)-sum(x1)^2)
        b=(sum(y1)*sum(x1^2)-sum(x1*y1)*sum(x1))/(n*sum(x1^2)-sum(x1)^2)
        s=c(a,b)
return (s)
```

```
}

Regr=function(x,a,b)
{       return(a*x+b)
}       koef=Pol2(area,popul,k)
koef
[1]  1.101687627 -0.003668141
popul2=Regr(area,koef[1],koef[2])
popul popul

 [1] 0.9600143 0.9600143 0.9575029 0.9284275 0.9224550 0.9175700 0.9141405
 [8] 0.9138393 0.9122505 0.9113607 0.9088259 0.9038464 0.9002554 0.8998964
[15] 0.8984772 0.8969232 0.8964124 0.8962146 0.8950095 0.8924771 0.8919920
[22] 0.8919177 0.8909609 0.8905226 0.8858361 0.8857247 0.8856720 0.8844945
[29] 0.8839077 0.8832697 0.8812112 0.8806386 0.8800207 0.8797388 0.8793523
[36] 0.8767754 0.8761149 0.8747420 0.8744664 0.8728723 0.8721672 0.8719027
[43] 0.8716783 0.8714157 0.8712552 0.8712044 0.8688390 0.8678441 0.8675064
[50] 0.8668793 0.8662894 0.8651741 0.8651249 0.8646659 0.8632923 0.8629712
[57] 0.8624614 0.8624118 0.8614341 0.8592021 0.8585407 0.8581866 0.8566166
[64] 0.8555066 0.8548941 0.8544933 0.8539290 0.8528221 0.8528122 0.8527964
[71] 0.8508107 0.8507203 0.8498093 0.8483029 0.8480140 0.8471196 0.8467234
[78] 0.8463004 0.8459642 0.8459113 0.8451787 0.8447125 0.8444430 0.8443460
[85] 0.8442459 0.8439911 0.8437848 0.8433412 0.8426979 0.8423491 0.8413263
[92] 0.8407224 0.8405909 0.8402478 0.8377801 0.8372180 0.8360058 0.8357936
[99] 0.8348106 0.8343191 0.8336924 0.8326029 0.8318430 0.8316937 0.8307564
[106] 0.8304823 0.8304490 0.8292021 0.8289364 0.8281176 0.8277353 0.8276984
[113] 0.8258102 0.8251554 0.8246801 0.8233756 0.8228290 0.8223616 0.8222645
[120] 0.8220235 0.8213069 0.8212737 0.8203774 0.8192112 0.8181663 0.8179691
[127] 0.8178318 0.8149265 0.8145230 0.8144971 0.8131666 0.8125598 0.8114014
[134] 0.8098961 0.8084547 0.8075209 0.8071383 0.8032770 0.8004977 0.7999746
[141] 0.7998570 0.7995310 0.7975874 0.7946792 0.7944302 0.7873106 0.7870478
[148] 0.7826576 0.7822781 0.7726734 0.7725976 0.7723690 0.7684943 0.7647906
[155] 0.7616799 0.7562563 0.7555305
popul2
[1]  3.789512618  3.789512618  3.022134644  3.788673677  2.611557437
 [6]  3.703603313  2.003711048  0.431635512  2.042704483  4.177460537
[11]  1.297034800  2.632819157  1.096092319  1.183974725  1.247948563
[16]  2.185782272  2.107650703  1.909106546  2.500203537  1.618565265
[21]  1.557930218  2.769801086  0.909006196  1.099943480  1.788587717
[26]  2.265521706  0.081659369  1.694110448  1.546142319  2.210334060
[31]  2.061095504  2.897772555  2.819843859  1.132421739  1.661247053
[36]  2.781585706  3.812665781  1.440200973  0.902127370  1.424338437
[41]  0.452014882  2.305950817  2.032400910  1.441820705  1.179119119
[46]  1.758115897  2.703952609  0.264467337  0.892781948 -0.938134674
[51]  1.581954069  0.892249015  1.927664229 -0.084637762  0.663853454
[56]  1.670790105 -0.327137847  1.159356489  1.493475238  2.281955045
[61]  1.879933562 -0.793769249  2.882704166  1.016149473  0.643151665
[66]  2.294699519  0.248176618  0.167782403  1.325927479  1.046463783
[71]  0.718956614  2.277822171  1.876158425  0.184917671  0.349864800
[76]  2.305205078  0.005379571 -1.119166002  0.550126695 -0.144792141
[81]  0.016379140 -0.033628423 -0.629600714 -1.278734303  0.767777452
[86] -1.222025714  2.181646469  0.919814045  1.446686580  0.199941994
[91]  1.739574471 -1.218752524 -0.053232550 -0.084637762  0.185356423
[96] -0.799911673 -1.469138060  0.195413174 -1.525921704  0.425797591
[101] -0.472410082  1.470679161  2.550692440  1.360400465 -0.022891030
[106] -0.238256376  0.886254345  0.335812571 -0.755263558 -0.621995056
[111]  0.733515791  1.200766312  1.515727009  0.241070813  1.313492945
[116] -0.476823684 -0.579710124 -0.266884916 -0.087650094  1.720256484
[121] -0.258752386  1.004877015 -2.339530087 -0.577771903 -1.008838726
```

```
[126]  1.210628816  0.189278227 -0.331870624 -0.187519703  0.612158398
[131] -1.183822543  2.126683617 -1.145096805 -1.712575024 -2.273510201
[136]  2.459597484  1.123451563 -0.342648258  1.950405901 -1.304910873
[141] -1.568329914 -1.125266218  1.663205543 -2.241065241 -0.939974344
[146] -0.708116443  0.995437492 -1.918943337 -1.741834302 -2.272535451
[151] -1.684002546 -1.417711959  0.799302767 -0.665528609 -2.004716525
[156] -1.210346720 -1.178281286
ks.test(popul,popul2)
        Two-sample Kolmogorov-Smirnov test
data:  popul and popul2
D = 0.4777, p-value = 5.551e-16
alternative hypothesis: two-sided
```

Висновок

На лабораторній роботі я перевірив статистичні гіпотези використовуючи тест Шапіро-Уілки, для розподілу характеристик квітів ірису, та використовуючи тест Колмогорова-Смирнова перевірив чи побудована регресійна функція та *population_log* належать до одного безперервного розподілу в системі R. За отриманими даними результатами тесту Шапіро-Уілки бачимо що гіпотеза не відкидається лише для 2-го ряду даних. Тест Колмогорова-Смирнова показав що, функція розподілу вибірки не спів-падає з гіпотетичною.