MSP Projekt - Adam Múdry (xmudry01)

December 11, 2022

[1]:
```
from io import StringIO
import numpy as np
import pandas as pd
from scipy.stats import chisquare, chi2, f, ttest_ind
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

---

1 1. úloha

1.1 Načítanie a úprava dát

Načítame si dáta.

[2]:
```
data_csv = """Praha,Brno,Znojmo,Tisnov,Paseky,Horni Lomna,Dolni Vestonice,okoli␣
 ↪studenta
1327,915,681,587,284,176,215,42
510,324,302,257,147,66,87,12
352,284,185,178,87,58,65,13
257,178,124,78,44,33,31,9
208,129,70,74,6,19,32,8
"""
df_input = pd.read_csv(StringIO(data_csv))
df_input.index = ['pocet respondentov', 'zimny cas', 'letny cas', 'striedanie␣
 ↪casu', 'nema nazor']
df_input
```

[2]:
```
                   Praha  Brno  Znojmo  Tisnov  Paseky  Horni Lomna  \
pocet respondentov   1327   915     681     587     284          176
zimny cas             510   324     302     257     147           66
letny cas             352   284     185     178      87           58
striedanie casu       257   178     124      78      44           33
nema nazor            208   129      70      74       6           19

                   Dolni Vestonice  okoli studenta
pocet respondentov              215              42
```

```
zimny cas                      87              12
letny cas                      65              13
striedanie casu                31               9
nema nazor                     32               8
```

Dáta si rozdelíme na 3 dátove rámce (data frames).

Prvý rámec obsahuje získané odpovede na jednotlivých miestach a ich okrajové početnosti.

[3]:
```
df = df_input.iloc[1:]
df
```

[3]:
```
                 Praha  Brno  Znojmo  Tisnov  Paseky  Horni Lomna  \
zimny cas          510   324     302     257     147           66
letny cas          352   284     185     178      87           58
striedanie casu    257   178     124      78      44           33
nema nazor         208   129      70      74       6           19

                 Dolni Vestonice  okoli studenta
zimny cas                     87              12
letny cas                     65              13
striedanie casu               31               9
nema nazor                    32               8
```

[4]:
```
df_column_sums = df.sum(axis=0)
df_column_sums
```

[4]:
```
Praha             1327
Brno               915
Znojmo             681
Tisnov             587
Paseky             284
Horni Lomna        176
Dolni Vestonice    215
okoli studenta      42
dtype: int64
```

[5]:
```
df_row_sums = df.sum(axis=1)
df_row_sums
```

[5]:
```
zimny cas          1705
letny cas          1222
striedanie casu     754
nema nazor          546
dtype: int64
```

Teoretické (očakávané) početnosti

[6]:
```
n = df.sum().sum()
column = []
for i in df_row_sums:
    row = []
    for j in df_column_sums:
        x = (i*j)/n
        row.append(x)
    column.append(row)
df_expected = pd.DataFrame(column, index=df.index, columns=df.columns)
df_expected
```

[6]:

	Praha	Brno	Znojmo	Tisnov	Paseky	\
zimny cas	535.257866	369.073811	274.687722	236.771942	114.554057	
letny cas	383.627632	264.520937	196.872960	169.698131	82.102673	
striedanie casu	236.706411	163.215046	121.474805	104.707357	50.659096	
nema nazor	171.408091	118.190206	87.964514	75.822569	36.684173	

	Horni Lomna	Dolni Vestonice	okoli studenta
zimny cas	70.991247	86.722262	16.941093
letny cas	50.880530	62.155193	12.141945
striedanie casu	31.394370	38.351076	7.491838
nema nazor	22.733854	27.771469	5.425124

1.2 Riešenie pre a), b), c)

Na riešenie použijeme **Test dobrej zhody** a funkciu `chisquare()`

1.2.1 Výpočet Chi^2, porovnanie s kritickým oborom a odpoveď

Stupne voľnosti a horná hranica doplnku kritického oboru pre a), b), c)

[7]:
```
alfa = 0.05
m = len(df.columns)
q = 1 # pocet odhadnutych tried
k = m - q -1
critical_value = chi2.ppf(1-alfa, df=k)
print(f"Stupne voľnosti = {k}")
print(f"Doplnok kritického oboru = <0, {critical_value}>")
```

```
Stupne voľnosti = 6
Doplnok kritického oboru = <0, 12.591587243743977>
```

1.2.2 Zimný čas

Hypotéza H0: V miestách, obciach a okolí študenta je rovnaké percentuálne zastúpenie obyvateľov preferujúcich zimný čas.

Ha: V miestách, obciach a okolí študenta nie je rovnaké percentuálne zastúpenie obyvateľov preferujúcich zimný čas.

[8]:
```
chi_sqrt, _ = chisquare(f_obs=df.loc['zimny cas'], f_exp=df_expected.loc['zimny␣
↪cas'])
print("Chi^2 =", chi_sqrt)
print("K =", critical_value)

if chi_sqrt >= critical_value:
    print("Chi^2 >= K")
else:
    print("Chi^2 < K")
```

```
Chi^2 = 22.123238111256924
K = 12.591587243743977
Chi^2 >= K
```

a) Chi^2 nepatrí do doplnku kritického oboru W

H0 zamietame

1.2.3 Letný čas

Hypotéza

H0: V miestách, obciach a okolí študenta je rovnaké percentuálne zastúpenie obyvateľov preferujúcich letný čas.

Ha: V miestách, obciach a okolí študenta nie je rovnaké percentuálne zastúpenie obyvateľov preferujúcich letný čas.

[9]:
```
chi_sqrt, _ = chisquare(f_obs=df.loc['letny cas'], f_exp=df_expected.loc['letny␣
↪cas'])
print("Chi^2 =", chi_sqrt)
print("K =", critical_value)

if chi_sqrt >= critical_value:
    print("Chi^2 >= K")
else:
    print("Chi^2 < K")
```

```
Chi^2 = 6.643240260656005
K = 12.591587243743977
Chi^2 < K
```

b) Chi^2 patrí do doplnku kritického oboru W

H0 nezamietame

1.2.4 Striedanie času

Hypotéza H0: V miestách, obciach a okolí študenta je rovnaké percentuálne zastúpenie obyvateľov preferujúcich striedanie času.

Ha: V miestách, obciach a okolí študenta nie je rovnaké percentuálne zastúpenie obyvateľov preferujúcich striedanie času.

[10]:
```
chi_sqrt, _ = chisquare(f_obs=df.loc['striedanie casu'], f_exp=df_expected.
↪loc['striedanie casu'])
print("Chi^2 =", chi_sqrt)
print("K =", critical_value)

if chi_sqrt >= critical_value:
    print("Chi^2 >= K")
else:
    print("Chi^2 < K")
```

```
Chi^2 = 12.613887594332608
K = 12.591587243743977
Chi^2 >= K
```

c) Chi^2 nepatrí do doplnku kritického oboru W

H0 zamietame

1.3 Riešenie d), e)

Na riešenie použijeme **Test dobrej zhody** a funkciu `chisquare()`

[11]:
```
df_d = pd.DataFrame()
df_d["velke mesta"] = df["Praha"] + df["Brno"]
df_d["male mesta"] = df["Znojmo"] + df["Tisnov"]
df_d["obce"] = df["Paseky"] + df["Horni Lomna"] + df["Dolni Vestonice"]
df_d
```

[11]:

	velke mesta	male mesta	obce
zimny cas	834	559	300
letny cas	636	363	210
striedanie casu	435	202	108
nema nazor	337	144	57

[12]:
```
df_d_column_sums = df_d.sum(axis=0)
df_d_column_sums
```

[12]:
```
velke mesta    2242
male mesta     1268
obce            675
dtype: int64
```

[13]:
```
df_d_row_sums = df_d.sum(axis=1)
df_d_row_sums
```

[13]:
```
zimny cas          1693
letny cas          1209
striedanie casu     745
nema nazor          538
dtype: int64
```

Teoretické (očakávané) početnosti

[14]:
```
n = df_d.sum().sum()
column = []
for i in df_d_row_sums:
    row = []
    for j in df_d_column_sums:
        x = (i*j)/n
        row.append(x)
    column.append(row)
df_d_expected = pd.DataFrame(column, index=df_d.index, columns=df_d.columns)
df_d_expected
```

[14]:

	velke mesta	male mesta	obce
zimny cas	906.978734	512.956750	273.064516
letny cas	647.688889	366.311111	195.000000
striedanie casu	399.113501	225.725209	120.161290
nema nazor	288.218877	163.006930	86.774194

1.3.1 Výpočet Chi^2, porovnanie s kritickým oborom a odpoveď

Stupne voľnosti a horná hranica doplnku kritického oboru pre d), e)

[15]:
```
alfa = 0.05
m = len(df_d.columns)
q = 1 # pocet odhadnutych tried
k = m - q - 1
critical_value = chi2.ppf(1-alfa, df=k)
print(f"Stupne voľnosti = {k}")
print(f"Doplnok kritického oboru = <0, {critical_value}>")
```

```
Stupne voľnosti = 1
Doplnok kritického oboru = <0, 3.841458820694124>
```

1.3.2 d)

Hypotéza pre d) H0: U väčších miest, menších miest a obcí je rovnaké percentuálne zastúpenie obyvateľov preferujúcich zimný čas.

Ha: U väčších miest, menších miest a obcí nie je rovnaké percentuálne zastúpenie obyvateľov preferujúcich zimný čas.

[16]:
```
chi_sqrt, _ = chisquare(f_obs=df_d.loc['zimny cas'], f_exp=df_d_expected.
 ↪loc['zimny cas'])
print("Chi^2 =", chi_sqrt)
print("K =", critical_value)

if chi_sqrt >= critical_value:
    print("Chi^2 >= K")
else:
    print("Chi^2 < K")
```

```
Chi^2 = 12.661948651569508
K = 3.841458820694124
Chi^2 >= K
```

d) Chi^2 nepatrí do doplnku kritického oboru W

H0 zamietame

1.3.3 e)

Hypotéza pre e) H0: U väčších miest, menších miest a obcí je rovnaké percentuálne zastúpenie nerozhodnutých obyvateľov.

Ha: U väčších miest, menších miest a obcí nie je rovnaké percentuálne zastúpenie nerozhodnutých obyvateľov.

[17]:
```
chi_sqrt, _ = chisquare(f_obs=df_d.loc['nema nazor'], f_exp=df_d_expected.
 ↪loc['nema nazor'])
print("Chi^2 =", chi_sqrt)
print("K =", critical_value)

if chi_sqrt >= critical_value:
    print("Chi^2 >= K")
else:
    print("Chi^2 < K")
```

```
Chi^2 = 20.688664757394136
K = 3.841458820694124
Chi^2 >= K
```

e) Chi^2 nepatrí do doplnku kritického oboru W

H0 zamietame

1.4 Riešenie f)

Porovnanie odpovedí z okolia študenta a odpovedí z veľkých, malých miest a obcí.

Používam Welchov T-test, lebo sa viac hodí na datasety s rôznou veľkosťou ako Studentov.

[18]:
```
df_d_s = df_d
df_d_s['okoli studenta'] = df['okoli studenta']

for col in list(df_d_s.columns):
    if col == "okoli studenta":
        continue

    t_statistic, p_value = ttest_ind(df_d_s['okoli studenta'], df_d_s[col],␣
 ↪equal_var=False)

    print(f"p-value for 'okoli studenta' and '{col}': {p_value}\n")
```

```
p-value for 'okoli studenta' and 'velke mesta': 0.015532486132489995

p-value for 'okoli studenta' and 'male mesta': 0.045893891082867136

p-value for 'okoli studenta' and 'obce': 0.06117252249298722
```

P-hodnota je najnižšia (najviac signifikantná) pri porovnaní s veľkým mestom. Tým pádom je najpravepodobnejšie, že sa prieskum uskutočnil vo veľkom meste, čo sa zhoduje s realitou.

---

2 Úloha 2

Hodnoty stĺpca Zi sú zo zadania číslo 16.

[19]:
```
data_csv = """
xi,yi,zi
0.00,0.00,-75.23
0.00,1.67,-133.49
0.00,3.33,221.49
0.00,5.00,-141.96
0.00,6.67,-150.27
0.00,8.33,-238.79
0.00,10.00,-338.73
2.22,0.00,61.52
2.22,1.67,79.17
2.22,3.33,136.48
2.22,5.00,43.3
2.22,6.67,101.39
2.22,8.33,-184.1
2.22,10.00,-53.95
4.44,0.00,164.96
4.44,1.67,305.02
4.44,3.33,36.75
4.44,5.00,-16.15
```

```
4.44,6.67,114.94
4.44,8.33,153.55
4.44,10.00,-14.72
6.67,0.00,28.79
6.67,1.67,172.22
6.67,3.33,188.25
6.67,5.00,74.83
6.67,6.67,295.69
6.67,8.33,291.03
6.67,10.00,162.86
8.89,0.00,299.93
8.89,1.67,422.69
8.89,3.33,426.75
8.89,5.00,422.44
8.89,6.67,279.62
8.89,8.33,593.67
8.89,10.00,367.81
11.11,0.00,508.42
11.11,1.67,731.32
11.11,3.33,678.25
11.11,5.00,816.06
11.11,6.67,732.05
11.11,8.33,704.33
11.11,10.00,659.81
13.33,0.00,761.02
13.33,1.67,992.01
13.33,3.33,1008.87
13.33,5.00,1058.73
13.33,6.67,1006.23
13.33,8.33,994.69
13.33,10.00,899.51
15.56,0.00,1028.39
15.56,1.67,1057.04
15.56,3.33,1394.85
15.56,5.00,1537.02
15.56,6.67,1317.17
15.56,8.33,1434.94
15.56,10.00,1530.17
17.78,0.00,1712.19
17.78,1.67,1585.74
17.78,3.33,1829.18
17.78,5.00,1688.55
17.78,6.67,1849.22
17.78,8.33,1807.13
17.78,10.00,1944.75
20.00,0.00,1893.26
20.00,1.67,1909.45
```

```
20.00,3.33,2014.4
20.00,5.00,2085.05
20.00,6.67,2280.92
20.00,8.33,2435.12
20.00,10.00,2369.4
"""
```

[20]:
```
df = pd.read_csv(StringIO(data_csv))
df
```

[20]:

	xi	yi	zi
0	0.0	0.00	-75.23
1	0.0	1.67	-133.49
2	0.0	3.33	221.49
3	0.0	5.00	-141.96
4	0.0	6.67	-150.27
..	…	…	…
65	20.0	3.33	2014.40
66	20.0	5.00	2085.05
67	20.0	6.67	2280.92
68	20.0	8.33	2435.12
69	20.0	10.00	2369.40

[70 rows x 3 columns]

2.1 Riešenie a), b)

Rovnicu regresnej funkcie použijeme v OLS a získame výsledky.

$$Z = \beta_1 + \beta_2 X + \beta_3 Y + \beta_4 (X^2) + \beta_5 (Y^2) + \beta_6 (X \cdot Y)$$

Na výpočet modelu regresie používam metódu OLS (metóda najmenších štvorcov), lebo je považovaná za najlepšiu, ak sú nasledujúce predpoklady pravdivé:

1. Parametre sú lineárne
2. Dáta sú náhodne vybraté
3. Neexistuje perfektná kolinearita
4. Nulou podmienená stredná hodnota
5. Homoskedasticita (podmienený rozptyl danej náhodnej veličiny je konštantný)
6. Chyby majú normálnu distribúciu

[21]:
```
# F = np.column_stack((df['xi'], df['yi'], df['xi']**2, df['yi']**2,␣
 ↪df['xi']*df['yi']))
# F = sm.add_constant(F)
# model = sm.OLS(endog=df['zi'], exog=F)
## The same as below:
```

```
formula = 'zi ~ 1 + I(xi) + I(yi) + I(xi**2) + I(yi**2) + I(xi*yi)'
model = smf.ols(formula=formula, data=df)
result = model.fit()
print(result.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                     zi   R-squared:                       0.983
Model:                            OLS   Adj. R-squared:                  0.982
Method:                 Least Squares   F-statistic:                     743.5
Date:                Sun, 11 Dec 2022   Prob (F-statistic):           2.89e-55
Time:                        21:00:52   Log-Likelihood:                -420.38
No. Observations:                  70   AIC:                             852.8
Df Residuals:                      64   BIC:                             866.3
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      4.8204     49.133      0.098      0.922     -93.334     102.974
I(xi)         -6.1581      7.665     -0.803      0.425     -21.471       9.154
I(yi)          6.9795     14.473      0.482      0.631     -21.933      35.892
I(xi ** 2)     4.9926      0.342     14.605      0.000       4.310       5.676
I(yi ** 2)    -3.4328      1.275     -2.692      0.009      -5.981      -0.885
I(xi * yi)     3.8198      0.577      6.624      0.000       2.668       4.972
==============================================================================
Omnibus:                        1.557   Durbin-Watson:                   1.920
Prob(Omnibus):                  0.459   Jarque-Bera (JB):                1.577
Skew:                           0.303   Prob(JB):                        0.455
Kurtosis:                       2.585   Cond. No.                         839.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

Zistenie ktoré premenné sú signifikantné.

[22]:
```
significant = [p_value < alfa for p_value in result.pvalues]
df_sig = pd.DataFrame(data = [x for x in result.summary().tables[1].data[1:] if␣
 ↪float(x[4]) < alfa], columns = result.summary().tables[1].data[0])
df_sig
```

[22]:

		coef	std err	t	P>|t|	[0.025	0.975]
0	I(xi ** 2)	4.9926	0.342	14.605	0.000	4.310	5.676
1	I(yi ** 2)	-3.4328	1.275	-2.692	0.009	-5.981	-0.885
2	I(xi * yi)	3.8198	0.577	6.624	0.000	2.668	4.972

Odstránenie nesignifikantných premenných z vzorcu regresie a výpočet nového modelu. Nový vzorec:

$$Z = \beta_1 + \beta_2(X^2) + \beta_3(Y^2) + \beta_4(X \cdot Y)$$

[23]:
```
formula_sig = 'zi ~ 1 + I(xi**2) + I(yi**2) + I(xi*yi)'
model_sig = smf.ols(formula=formula_sig, data=df)
result_sig = model_sig.fit()
print(result_sig.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                     zi   R-squared:                       0.983
Model:                            OLS   Adj. R-squared:                  0.982
Method:                 Least Squares   F-statistic:                     1258.
Date:                Sun, 11 Dec 2022   Prob (F-statistic):           3.80e-58
Time:                        21:00:52   Log-Likelihood:                -420.93
No. Observations:                  70   AIC:                             849.9
Df Residuals:                      66   BIC:                             858.9
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -5.8014     23.183     -0.250      0.803     -52.088      40.485
I(xi ** 2)     4.7195      0.146     32.246      0.000       4.427       5.012
I(yi ** 2)    -2.7400      0.574     -4.774      0.000      -3.886      -1.594
I(xi * yi)     3.7671      0.492      7.658      0.000       2.785       4.749
==============================================================================
Omnibus:                        1.539   Durbin-Watson:                   1.899
Prob(Omnibus):                  0.463   Jarque-Bera (JB):                1.540
Skew:                           0.335   Prob(JB):                        0.463
Kurtosis:                       2.718   Cond. No.                         389.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

Submodel - 95% intervaly spoľahlivosti

[24]:
```
alfa = 0.05
int_conf = result_sig.conf_int(alpha=alfa).rename(columns = {0: '<', 1: '>'})
int_conf
```

[24]:

	<	>
Intercept	-52.087784	40.485022

```
I(xi ** 2)   4.427255   5.011690
I(yi ** 2)  -3.885911  -1.593995
I(xi * yi)   2.784960   4.749337
```

Hodnoty R^2 (98,3%) pre originálny model a zjednodušeného submodelu sú približne rovnaké (zaokrúhlené na 3 desatinné miesta).

[25]:
```
assert round(result.rsquared, 3) == round(result_sig.rsquared, 3)
print(round(result_sig.rsquared, 3))
```

```
0.983
```

Ďalej riešim úlohu na submodeli.

2.2 Riešenie c)

Nestranný rozptyl závislej premennej.

Keďže bias = 0, môžeme použiť strednú kvadratickú chybu.

[26]:
```
print(result_sig.mse_resid)
```

```
10379.41299704382
```

2.3 Riešenie d)

Hypotéza H0: Dva mnou zvolené regresné parametry sú súčasne nulové

Ha: Dva mnou zvolené regresné parametry nie sú súčasne nulové

F-test

[27]:
```
f_test_result = result_sig.f_test('((I(xi ** 2)) = 0), ((I(yi ** 2)) = 0)')
print(f"{f_test_result.pvalue = }")
assert f_test_result.pvalue < alfa
print("f_test_result.pvalue < alfa")
```

```
f_test_result.pvalue = 3.293799470353056e-50
f_test_result.pvalue < alfa
```

d) p-hodnota je nižšia ako alfa

H0: zamietam

2.4 Riešenie e)

Hypotéza H0: Dva mnou zvolené regresné parametry sú rovnaké

Ha: Dva mnou zvolené regresné parametry nie sú rovnaké

T-test

[28]:
```
t_test_result = result_sig.t_test('(I(xi ** 2)) = (I(yi ** 2))')
print(t_test_result)
print(f"t_test_result.pvalue = {t_test_result.pvalue}")
assert t_test_result.pvalue < alfa
print("t_test_result.pvalue < alfa")
```

```
                             Test for Constraints
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
c0             7.4594      0.497     14.999      0.000       6.466       8.452
==============================================================================
t_test_result.pvalue = 6.063004275110807e-23
t_test_result.pvalue < alfa
```

e) p-hodnota je nižšia ako alfa

H0 zamietam