	MSP projekt Matej Mištík - xmisti00 - streda 8:00
3	<pre>import pandas as pd from scipy import stats pd.set_option('display.max_columns', 500) pd.set_option('display.width', 1000) df = pd.read_excel('./MSP_Projekt_2022-23_Zadani_St_8 (1).xlsx', sheet_name='Úkol 1') df = df.dropna()</pre>
	<pre>df = df.rename(columns={'Unnamed: 0': ''}) df2 = df.copy() # # merge columns of Praha and Brno df2['Praha'] = df2['Praha'].astype(int) df2['Brno'] = df2['Brno'].astype(int) df2['Praha'] + df2['Brno']</pre>
	<pre>df2 = df2.drop(columns=['Brno']) df2 = df2.rename(columns={'Praha': 'Vetsi_mesta'}) # merge columns of Znojmo Tišnov df2['Znojmo'] = df2['Znojmo'].astype(int) df2['Tišnov'] = df2['Tišnov'].astype(int) df2['Znojmo'] = df2['Znojmo'] + df2['Tišnov']</pre>
i	<pre>df2 = df2.drop(columns=['Tišnov']) df2 = df2.rename(columns={'Znojmo': 'Mala_mesta'}) # merge forth, fifth, sixth column df2['Rokytnice nad Jizerou'] = df2['Rokytnice nad Jizerou'].astype(int) df2['Jablunkov'] = df2['Jablunkov'].astype(int) df2['Jolni Věstonice'] = df2['Dolni Věstonice'].astype(int) df2['Dolni Věstonice'] = df2['Dolni Věstonice'].astype(int)</pre>
]: [df2['Rokytnice nad Jizerou'] = df2['Rokytnice nad Jizerou'] + df2['Jablunkov'] + df2['Dolní Věstonice'] df2 = df2.drop(columns=['Jablunkov', 'Dolní Věstonice']) df2 = df2.rename(columns={'Rokytnice nad Jizerou': 'obce'}) print('Datova sada, s ktorou budeme pracovať v ulohách 1a až 1c:') df Datova sada, s ktorou budeme pracovať v ulohách 1a až 1c:
:	Praha Brno Znojmo Tišnov Rokytnice nad Jizerou Jablunkov Dolní Věstonice okolí studenta o počet respondentů 1327.0 915.0 681.0 587.0 284.0 176.0 215.0 20.0 zimní čas 510.0 324.0 302.0 257.0 147.0 66.0 87.0 4.0 ziřídání časů 257.0 178.0 178.0 58.0 65.0 9.0 ziřídání časů 257.0 178.0 78.0 44.0 33.0 4.0
	4 nemá názor 208.0 129.0 70.0 74.0 6.0 19.0 32.0 2.0 print('Datova sada, s ktorou budeme pracovať v ulohách 1d až 1f:') df2 Datova sada, s ktorou budeme pracovať v ulohách 1d až 1f: Vetsi_mesta Mala_mesta obce okolí studenta
;	0 počet respondentů 2242 1268 675 20.0 1 zimní čas 834 559 300 4.0 2 letní čas 636 363 210 9.0 3 střídání časů 435 202 108 4.0 4 nemá názor 337 144 57 2.0
	BOD 1 Hypotézy
ı	1.a H0 : percentuálne zastúpenie obyvateľov pre zimný čas je rovnaký pre všetky mesta H1 : percentuálne zastúpenie obyvateľov pre zimný čas je rôzny pre všetky mesta
I	 1.b H0: percentuálne zastúpenie obyvateľov pre letný čas je rovnaký pre všetky mestá H1: percentuálne zastúpenie obyvateľov pre letný čas je rôzny pre všetky mestá 1.c
: :	H0 : percentuálne zastúpenie obyvateľov pre striedanie časov je rovnaký pre všetky mestá H1 : percentuálne zastúpenie obyvateľov pre striedanie časov je rôzny pre všetky mestá # A - zimny, B - letny , C - striedanie casov alpha = 0.05
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	<pre># sum of first row sumForEveryRow = df.sum(axis=1,numeric_only=True) print('SUM of every row:') print(sumForEveryRow) print('') # cycle trough list of sumForEveryRow optionsList = ['A','B','C']</pre>
	<pre>for i in range(1,4): print('Option:', optionsList[i-1]) p = sumForEveryRow[i] / sumForEveryRow[0] print('probability:', p) values = df.iloc[0,1:].values print('Observed frequencies in each category:', values) # create list of expected values for every column</pre>
	<pre>expectedValues = [] for j in range(0,8): expectedValues.append(values[j] * p) print('Expected frequencies in each category:',expectedValues) T, p = stats.chisquare(f_obs = df.iloc[i].values[1:,], f_exp = expectedValues,ddof=1) print('chisq:',T)</pre>
(<pre>print('The p-value of the test:',p) print('H0 is rejected' if p < alpha else 'H0 is not rejected') print('') SUM of every row: 0 4205.0 1 1697.0 2 1218.0</pre>
() () () ()	3 749.0 4 540.0 dtype: float64
 	The p-value of the test: 0.0008957117744714648 H0 is rejected
 	The p-value of the test: 0.2133672661064076 H0 is not rejected
	The p-value of the test: 0.05403021733299772 H0 is not rejected
	H0 : percentuálne zastúpenie obyvateľov pre zimný čas je rovnaký pre všetky mestá - zamieta sa 1.b H0 : percentuálne zastúpenie obyvateľov pre letný čas je rovnaký pre všetky mestá - nezamieta sa
ı	1.c H0 : percentuálne zastúpenie obyvateľov pre striedanie časov je rovnaký pre všetky mestá - nezamieta sa Tabuľka hodnôt pre úlohy 1.d až 1.f
1	# Tabulka pre ulohy 1d až 1f df2 Vetsi_mesta Mala_mesta obce okolí studenta 0 počet respondentů 2242 1268 675 20.0 1 zimní čas 834 559 300 4.0
;	2 letní čas 636 363 210 9.0 3 střídání časů 435 202 108 4.0 4 nemá názor 337 144 57 2.0
:	Hypotézy D, E 1.d H0: percentuálne zastúpenie obyvateľov pre zimný čas je rovnaký pre vsetky zastúpenie miest H1: percentuálne zastúpenie obyvateľov pre zimný čas je rôzny pre vsetky zastúpenie miest
:	1.e H0 : percentuálne zastúpenie nerozhodnutných obyvateľov je rovnaký pre všetky zastúpenie miest H1 : percentuálne zastúpenie nerozhodnutných obyvateľov je rôzny pre všetky zastúpenie miest
:	#D, E nova tabula bez okolia studenta , len 3 hodnoty sumForEveryRow = df2.sum(axis=1,numeric_only=True) print('SUM of every row:') print(sumForEveryRow) print('
	<pre>expectedValues = [] for j in range(0,8): expectedValues.append(values[j] * p) print('Expected frequencies in each category:',expectedValues) T, p = stats.chisquare(f_obs = df.iloc[i].values[1:,], f_exp = expectedValues,ddof=1)</pre>
	<pre>print('chisq:',T) print('The p-value of the test:',p) print('H0 is rejected' if p < alpha else 'H0 is not rejected') print('') SUM of every row: 0</pre>
(((2 1218.0 3 749.0 4 540.0 dtype: float64
 - 	chisq: 22.720476798835556 The p-value of the test: 0.0008957117744714648 H0 is rejected
	chisq: 39.78016680405891 The p-value of the test: 5.031424182862881e-07 H0 is rejected Vyhodnotenie Hypotéz 1.d
	1.u H0 : percentuálne zastúpenie obyvateľov pre zimný čas je rovnaký pre vsetky zastúpenie miest - zamieta sa 1.e H0 : percentuálne zastúpenie nerozhodnutných obyvateľov je rovnaký pre všetky zastúpenie miest - zamieta sa
	Hypotéza F pre alfa = 0.1 1.f H0: Okolie študenta je podobné veľkým mestám
	H1: Okolie študenta je podobné malým mestám H2: Okolie študenta je podobné obciam sumForEveryRow = df2.sum(axis=1, numeric_only=True) print('SUM of every row:') print(sumForEveryRow)
i	<pre>print('') # create variable for p1, p2 in cycle # cycle trough list of sumForEveryRow alpha = 0.1 for i in range(1,4): p = sumForEveryRow[i]/ sumForEveryRow[0] print('probability:',p)</pre>
	<pre>values = df.iloc[0,1:].values print(values) # create list of expected values for every column expectedValues = [] for j in range(0,8): expectedValues.append(values[j] * p)</pre>
	<pre>print(expectedValues) T, p = stats.chisquare(f_obs = df.iloc[i].values[1:,], f_exp = expectedValues,ddof=1) print('chisq:',T) print('The p-value of the test:',p) print('H{} is rejected'.format(i) if p < alpha else 'H{} is not rejected'.format(i))</pre>
(print('') SUM of every row: 0 4205.0 1 1697.0 2 1218.0 3 749.0 4 540.0 dtype: float64
 	The p-value of the test: 0.0008957117744714648 Has rejected
(-	probability: 0.2896551724137931 [1327.0 915.0 681.0 587.0 284.0 176.0 215.0 20.0] [384.3724137931035, 265.0344827586207, 197.25517241379313, 170.02758620689656, 82.26206896551724, 50.97931034482759, 62.27586206896552, 5.793103448275863] chisq: 8.3529705718337 The p-value of the test: 0.2133672661064076 H2 is not rejected
- - -	[1327.0 915.0 681.0 587.0 284.0 176.0 215.0 20.0] [236.36694411414985, 162.98097502972652, 121.30059453032105, 104.55719381688466, 50.58644470868015, 31.34934601664685, 38.2960760998811, 3.5624256837098693] chisq: 12.378916461699824 The p-value of the test: 0.05403021733299772 H3 is rejected
(H2 is not rejected
	Vyhodnotenie Hypotézy F 1.f H0 : Okolie študenta je podobné veľkým mestám - zamieta sa
ı	H1 : Okolie študenta je podobné malým mestám - nezamieta sa H2 : Okolie študenta je podobné obciam - zamieta sa Okolie študenta je podobné malým mestám
	import numpy as np from sklearn.linear_model import LinearRegression from sklearn.preprocessing import PolynomialFeatures import matplotlib.pyplot as plt from mpl toolkits mplot3d import Axes3D
	<pre>from mpl_toolkits.mplot3d import Axes3D import statsmodels.api as sm import scipy.stats as stats from statsmodels.formula.api import ols data = pd.read_excel('./MSP_Projekt_2022-23_Zadani_St_8 (1).xlsx', sheet_name='Úkol 2 - Data') xi = np.array(pd.DataFrame(data, columns=['xi'])) yi = np.array(pd.DataFrame(data, columns=['yi'])) zi = np.array(pd.DataFrame(data, columns=['zi']))</pre>
	xi = xi.ravel() yi = yi.ravel() zi = zi.ravel() 2.a
; ; ;	<pre>Určenie vhodného modelu #ANOVA model = ols('zi ~ yi + I(yi**2) + xi + I(xi**2) + (xi*yi)', data = data).fit() at = sm.stats.anova_lm(model, typ=2) print(at) print(model.summary()) model.pvalues for i in range(0.6):</pre>
	<pre>for i in range(0,6): print('H{} is rejected'.format(i) if model.pvalues[i] < 0.05 else 'H{} is not rejected'.format(i))</pre>
2	xi:yi 8469.528042 1.0 0.860815 0.356997
: :: :: :: :: ::	Residual 629693.666965 64.0 NaN NaN OLS Regression Results ===================================
= = = = = = = = = = = = = = = = = = =	OLS Regression Results Section Results Dep. Variable: zi R-squared: 0.580 Model: OLS Adj. R-squared: 0.547 Method: Least Squares F-statistic: 17.64 Date: Sat, 10 Dec 2022 Prob (F-statistic): 6.00e-11 Time: 16:07:08 Log-Likelihood: -417.98 No. Observations: 70 AIC: 848.0 Df Residuals: 64 BIC: 861.5 Df Model: 5 Covariance Type: nonrobust The color of std err The policy of the pol
	Dep. Variable: Zi R-squared: 0.590 Model: 0LS Adj R-squared: 0.590 Model: 0LS Squared: 0.547 Method: Least Squared: 15:07:08 F-statistic: 17:64 Prior (F-statistic): 6.09e-11 Time: 16:07:08 Log-Likelihood: 417.98 Mot. 0Bservations: 70 Atc: 848.0 Prof Residuals: 64 BIC: 861.5 F-statistic: 17:04 Mot. 15:07 Model: 0.500
	Dep. Variable:
	OLS Megression Results Dep. Variable:
	Cut Part Cut
	Color
	Section Sect
	Section Sect
	Company Comp
	Part
	1
	The content of the
	The content of the
	The content of the
	Part
	Company Comp
	The content
	The content of the
	The state of the s
	The state of the s
	The state of the s
	The content of the
	The content of the co