Laboratorijska vježba 3 - Poređenje više sistema : ANOVA i kontrasti

Zadatak: Napisati program za korištenje ANOVA tehnike i tehnike kontrasta.

Metodologija: Zadatak se radi samostalno. U proizvoljnom programskom jeziku napisati program za izračunavanje parametara ANOVA testa uz dozvoljavanje unosa proizvoljnog broja alternativa i proizvoljnog broja mjerenja svake od njih. Poželjno je da program ima grafički interfejs. Dodatno, implementirati tehniku kontrasta svake dvije poređene alternative.

Rješenje:

U Excel fajlu (alternative.xlsx) je data tabela mjerenja i alternativa. Nakon učitavanja podataka iz tabele vršićemo izračunavanje parametara ANOVA testa po koracima po kojima smo to radili na laboratorijskim vježbama, nakon čega ćemo sprovesti i tehniku kontrasta.

Napomena: Kod sproveden u nastavku "radi" i za drugačiji broj mjerenja, kao i za drugačiji broj alternativa (uz uslov da broj mjerenja po alternativama bude isti, kao što smo radili na vježbama). Tj. dozvoljen je proizvoljan broj alternativa i proizvoljan broj mjerenja po alternativama (uslovno rečeno), samo je potrebno u skladu s tim ažurirati dati Excel fajl - alternative.xlsx

In [129…

```
# Uvezimo prvo potrebne biblioteke
import numpy as np
import math
import pandas as pd
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.stats import t # za računanje kritične vrijednosti Studentove raspodjele
import seaborn as sns
from itertools import combinations # za dobijanje kombinacija parova alternativa
```

In [130…

```
# Čitanje .xlsx fajla (u kojem su zabilježena mjerenja alternativa) i prikaz njegovog sadržaja
df = pd.read_excel('./alternative.xlsx')
print("Matrica mjerenja po alternativama:")
df
```

```
Matrica mjerenja po alternativama:
```

Out[130…

	Alt 1	Alt 2	Alt 3	Alt 4
0	25	45	30	54
1	30	55	29	60
2	28	29	33	51
3	36	56	37	62
4	29	40	27	73

Prije same tehnike ANOVA čisto informativno pogledajmo grafički prikaz raspodjele mjerenja po alternativama.

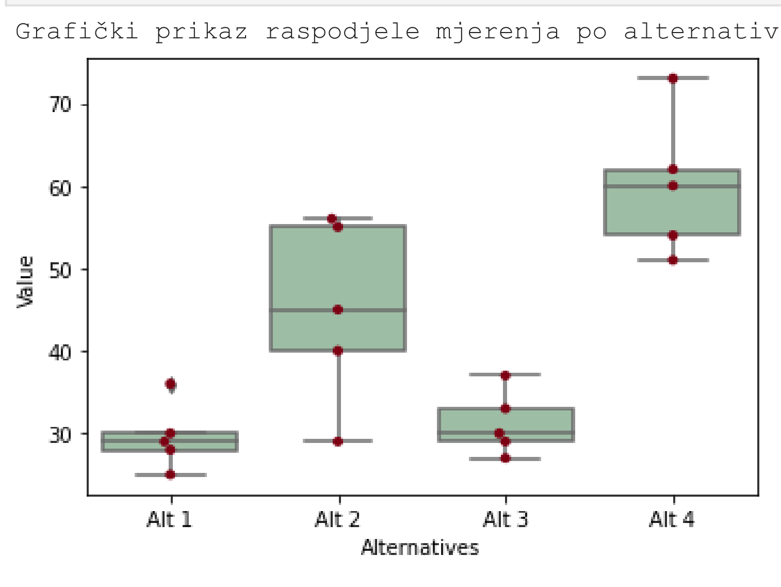
In [131…

```
# Preoblikujmo dataframe df kako bismo omogućili da bude pogodan za "statsmodels" paket
# koji će nam omogućiti da lakše vidimo razlike alternativa
df_melt = pd.melt(df.reset_index(), id_vars=['index'], value_vars=['Alt 1', 'Alt 2', 'Alt 3', 'Alt 4'])

# Premiještanje imena kolona
df_melt.columns = ['index', 'Alternatives', 'Value']

# Generišimo boxplot kako bismo vidjeli raspodjelu podataka po alternativama
# Korištenjem boxplot-a lakše detektujemo razlike između različitih alternativa
print("Grafički prikaz raspodjele mjerenja po alternativama:")
ax = sns.boxplot(x='Alternatives', y='Value', data=df_melt, color='#99c2a2')
ax = sns.swarmplot(x="Alternatives", y="Value", data=df_melt, color='#7d0013')
plt.show()
```

```
Grafički prikaz raspodjele mjerenja po alternativama:
```



Već na osnovu prethodnog dijagrama nam je jasno da se alternative 1 i 3 najmanje razlikuju.

ANOVA TEHNIKA:

In [132…

```
# Kreiranje ANOVA tabele i računanje parametara ANOVA tehnike
data = [['Alternatives', '', '', '', '', ''], ['Error', '', '', '', '', ''], ['Total', '', '', '', '', '']]
ANOVA_table = pd.DataFrame(data, columns = ['Variation', 'Sum of squares', 'Deg freedom', 'Mean square', 'Compu
ANOVA_table.set_index('Variation', inplace = True)

# Izračunaćemo prvo srednje vrijednosti kolona i to smjestiti u vektor y_col_ms
# Vektor y_col_ms ima onoliko elemenata koliko ima kolona, tj. alternativa (k)
y_col_ms = df.mean()
# Broj mjerenja n je broj vrsta matrice df
n = np.shape(df)[0]
# Broj alternativa k je broj kolona matrice df
k = np.shape(df)[1]
# Nađimo sada srednju vrijednost svih elemenata matrice df od k*n elemenata - y_total_ms
y_total_ms = np.average(df)

# Izračunavanje SSA i ažuriranje ANOVA tabele
SSA_vector = n * (y_col_ms - y_total_ms)**2 # vektor od k elemenata
SSA = SSA_vector.sum()
# Ažuriranje ANOVA tabele
ANOVA_table['Sum of squares']['Alternatives'] = SSA
#print(SSA)

# Izračunavanje SSE i ažuriranje ANOVA tabele
SSE_matrix = (df - y_col_ms)**2 # matrica od k*n elemenata
SSE_vector = SSE_matrix.sum()  # vektor od k elemenata
SSE = SSE_vector.sum()
# Ažuriranje ANOVA tabele
ANOVA_table['Sum of squares']['Error'] = SSE

# Izračunavanje SST i ažuriranje ANOVA tabele
SST = SSA + SSE
# Ažuriranje ANOVA tabele
ANOVA_table['Sum of squares']['Total'] = SST

# Izračunavanje stepeni slobode za SSA, SSE i SST i ažuriranje ANOVA tabele
df_SSA = k - 1 # zbog k alternativa
df_SSE = k * (n - 1) # zbog k alternativa, svaka sa n-1 stepeni slobode
df_SST = k * n - 1 # df_SST = df_SSA + df_SSE
# Ažuriranje ANOVA tabele
ANOVA_table['Deg freedom']['Alternatives'] = df_SSA
ANOVA_table['Deg freedom']['Error'] = df_SSE
ANOVA_table['Deg freedom']['Total'] = df_SST

# Izračunavanje varijanse sume kvadrata (srednje kvadratne vrijednosti) i ažuriranje ANOVA tabele
S_a = SSA / df_SSA # varijansa alternativa
S_e = SSE / df_SSE # varijansa greške
# Ažuriranje ANOVA tabele
ANOVA_table['Mean square']['Alternatives'] = S_a
ANOVA_table['Mean square']['Error'] = S_e

# Izračunavanje vrijednosti F_computed i ažuriranje ANOVA tabele
F_computed = S_a / S_e
# Ažuriranje ANOVA tabele
ANOVA_table['Computed F']['Alternatives'] = F_computed

# Izračunavanje vrijednosti F_tabulated i ažuriranje ANOVA tabele
# Ako pretpostavimo 95%-tni interval povjerenja, tada je 1-alpha=0.95 => alpha = 0.05
# F_tabulated se dobija iz tabele za vrijednosti [1-alpha;(k-1),k*(n-1)]
# U našem slučaju te vrijednosti su [0.95;3,16]
alpha = 0.05
# Izračunavanje F_tabulated
# Za ovo izračunavanje koristićemo funkciju stats.f.ppf() koja računa kritičnu vrijednost F raspodjele
# sa k-1 = 3 i k*(n-1) = 16 stepeni slobode sa 95%-nim intervalom povjerenja
# Ova funkcija zapravo daje ekvivalentnu vrijednost onoj iz naše skripte
F_tabulated = stats.f.ppf(1-alpha, ANOVA_table['Deg freedom']['Alternatives'], ANOVA_table['Deg freedom']['Erro
ANOVA_table['Tabulated F']['Alternatives'] = F_tabulated

print("ANOVA tabela:")
ANOVA_table
```

```
ANOVA tabela:
```

Out[132…

	Sum of squares	Deg freedom	Mean square	Computed F	Tabulated F
Variation					
Alternatives	3010.95	3	1003.65	17.4928	3.23887
Error	918	16	57.375		
Total	3928.95	19			

In [133…

```
# Zaključak korištenja F-testa za poređenje odnosa varijansi
# Pogledajmo najprije odnose SSA/SST i SSE/SST
# koji nam daju uticaje pojedinih izvora varijacija (alternative i grešaka) u ukupnoj varijaciji
ratio_a = SSA/SST # udio varijacije zbog razlika između alternativa u ukupnoj varijaciji
ratio_e = SSE/SST # udio varijacije zbog grešaka u mjerenjima u ukupnoj varijaciji
print("ZAKLJUČAK SPROVEDENE ANOVA TEHNIKE:")
print("-------------------------------------------------------------------------------------")
print(ratio_a * 100,"[%] ukupne varijacije u mjerenjima je zbog razlika između alternativa.")
print("-------------------------------------------------------------------------------------")
print(ratio_e * 100,"[%] ukupne varijacije u mjerenjima je zbog grešaka u mjerenjima.")
print("-------------------------------------------------------------------------------------")
conclusion = "Što znači da imamo 95%-tno povjerenje da razlike između alternativa nisu statistički značajne."
if ANOVA_table['Computed F']['Alternatives'] > ANOVA_table['Tabulated F']['Alternatives']:
    conclusion = "Što znači da imamo 95%-tno povjerenje da su razlike između alternativa statistički značajne.'
print("Rezultat F-testa:")
print("F izračunato je:", ANOVA_table['Computed F']['Alternatives'], " i F tabelarno je:", ANOVA_table['Tabulat
print(conclusion)
```

```
ZAKLJUČAK SPROVEDENE ANOVA TEHNIKE:
-------------------------------------------------------------------------------------
76.63497881113275 [%] ukupne varijacije u mjerenjima je zbog razlika između alternativa.
-------------------------------------------------------------------------------------
23.365021188867257 [%] ukupne varijacije u mjerenjima je zbog grešaka u mjerenjima.
-------------------------------------------------------------------------------------
Rezultat F-testa:
F izračunato je: 17.492810457516338  i F tabelarno je: 3.238871517453585
Što znači da imamo 95%-tno povjerenje da su razlike između alternativa statistički značajne.
```

TEHNIKA KONTRASTA:

Tehnika kontrasta nam daje informaciju o tome kako se pojedini parovi alternativa razlikuju, za razliku od ANOVA tehnike koja nam samo pokazuje da li postoji statistički značajna razlika između alternativa.

In [134…

```
# Pronađimo najprije vektor efekata alpha_effects
# Efekat svake kolone (alternative) dobijamo kao razliku srednje vrijednosti te kolone
# i ukupne srednje vrijednosti matrice df, y_col_ms i y_total_ms, respektivno, u našem slučaju
alpha_effects = y_col_ms - y_total_ms
print("Vektor efekata:")
alpha_effects
```

```
Vektor efekata:
```

Out[134…

```
Alt 1   -11.85
Alt 2     3.55
Alt 3   -10.25
Alt 4    18.55
dtype: float64
```

In [135…

```
# Pogledajmo sada sve moguće kombinacije parova alternativa čije ćemo intervale povjerenja računati
combs = list(combinations(df, 2))
print("Sve moguće kombinacije parova alternativa su:")
combs
```

```
Sve moguće kombinacije parova alternativa su:
```

Out[135…

```
[('Alt 1', 'Alt 2'),
 ('Alt 1', 'Alt 3'),
 ('Alt 1', 'Alt 4'),
 ('Alt 2', 'Alt 3'),
 ('Alt 2', 'Alt 4'),
 ('Alt 3', 'Alt 4')]
```

In [136…

```
# Računanje kritične vrijednosti Studentove t raspodjele
# Pretpostavimo 90%-tni interval povjerenja, tada je alpha=0.1 => alpha/2 = 0.05
# pa je a = 1-alpha/2 = 1-0.05 = 0.95
a = 0.95
# Stepen slobode je k*(n-1)
deg_freedom = k * (n-1)
# Pa se kritična vrijednost Studentove t raspodjele (t_value) dobija pozivom funkcije ppf():
t_value = t.ppf(a, deg_freedom)
print("Kritična vrijednost Studentove t raspodjele je:")
t_value
```

```
Kritična vrijednost Studentove t raspodjele je:
```

Out[136…

```
1.74588367627624
```

In [137…

```
# Računanje vrijednosti standardne devijacije Sc
# Standardna devijacija Se je kvadratni korijen varijanse S_e
Se = math.sqrt(S_e)
# Vrijednost ispod korijena je uvijek (2)/(k*n)
sq_rt = (2)/(k*n)
Sc = Se * math.sqrt(sq_rt)
print("Standardna devijacija Sc je:")
Sc
```

```
Standardna devijacija Sc je:
```

Out[137…

```
2.3953079134006967
```

Računanje kontrasta i intervala povjerenja za svaku kombinaciju alternativa koje se porede:

In [138…

```
groups = [] # vektor stringova parova alternativa koje se porede
contrast = [] # vektor vrijednosti kontrasta parova alternativa
c1 = [] # vektor donjih vrijednosti intervala povjerenja
c2 = [] # vektor gornjih vrijednosti intervala povjerenja
for comb in combs:
    # Izračunaćemo kontrast c za svaki par kao linaernu kombinaciju efekata alternativa
    c = alpha_effects[comb[0]] - alpha_effects[comb[1]]
    ci_lower = c - (t_value * Sc) # donja granica intervala povjerenja
    ci_upper = c + (t_value * Sc) # gornja granica intervala povjerenja
    # Dodavanje informacija o tekućem kontrastu na kraj odgovarajućeg vektora
    groups.append(str(comb[0]) + ' : ' + str(comb[1]))
    contrast.append(c)
    c1.append(ci_lower)
    c2.append(ci_upper)
```

Tabelarni prikaz rezultata dobijenih tehnikom kontrasta

In [139…

```
# Prikažimo dobijene rezultate tabelarno
result_table = pd.DataFrame({'groups': groups,
                             'c': contrast,
                             'c1': c1,
                             'c2': c2})
result_table
```

Out[139…

	groups	c	c1	c2
0	Alt 1 : Alt 2	-15.4	-19.581929	-11.218071
1	Alt 1 : Alt 3	-1.6	-5.781929	2.581929
2	Alt 1 : Alt 4	-30.4	-34.581929	-26.218071
3	Alt 2 : Alt 3	13.8	9.618071	17.981929
4	Alt 2 : Alt 4	-15.0	-19.181929	-10.818071
5	Alt 3 : Alt 4	-28.8	-32.981929	-24.618071

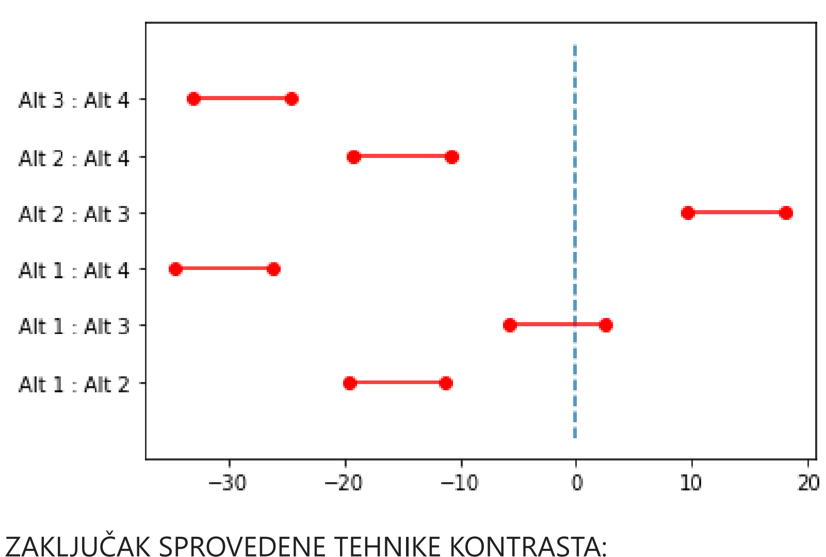
Grafički prikaz intervala povjerenja

In [140…

```
# Prikažimo dobijene intervale povjerenja grafički radi bolje vizuelizacije
data_dict = {}
data_dict['groups'] = groups
data_dict['ci_lower'] = c1
data_dict['ci_upper'] = c2
dataset = pd.DataFrame(data_dict)
# Grafički prikaz intervala povjerenja svih parova alternativa koje smo poredili
# Svaki interval povjerenja prikazan je jednom horizontalnom linijom u xy ravni
for ci_lower,ci_upper,y in zip(dataset['ci_lower'],dataset['ci_upper'],range(len(dataset))):
    plt.plot((ci_lower,ci_upper),(y,y),'ro-',color='red')
plt.yticks(range(len(dataset)),list(dataset['groups']))
# Predstavimo isprekidano osu x = 0 radi lakše detekcije intervala povjerenja koji (ne) sadrži 0
plt.vlines( 0, -1, len(groups), linestyles='dashed' )
```

Out[140…

```
<matplotlib.collections.LineCollection at 0x1f5ae0a0430>
```



ZAKLJUČAK SPROVEDENE TEHNIKE KONTRASTA:

Ukoliko interval povjerenja sadrži 0 ne postoji statistički značajna razlika između alternativa uključenih u kontrast. A ako interval povjerenja ne sadrži 0 tada postoji statistički značajna razlika između alternativa.

Na osnovu tabelarnog prikaza dobijenih rezultata, kao i grafičkog prikaza intervala povjerenja lako je zaključiti koji interval povjerenja sadrži/ne sadrži 0, a samim tim i koje alternative se statistički značajno ne razlikuju/razlikuju.

Za trenutni skup alternativa i mjerenja možemo zaključiti da samo za alternative Alt 1 i Alt 3 ne postoji statistički značajna razlika, dok se svi drugi parovi alternativa statistički značajno razlikuju.