Data Analysis Tools

Running an analysis of variance

Run an analysis of variance.

You will need to analyze and interpret post hoc paired comparisons in instances where your original statistical test was significant, and you were examining more than two groups (i.e. more than two levels of a categorical, explanatory variable).

In [1]:

```
import numpy as np
from numpy import count_nonzero
from numpy import median
from numpy import mean
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import random

import scipy

import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.formula.api import ols

%matplotlib inline
#sets the default autosave frequency in seconds
%autosave 60
sns.set_style('dark')
sns.set(font_scale=1.2)

plt.rc('axes', titlesize=9)
plt.rc('axes', labelsize=14)
plt.rc('xtick', labelsize=12)
plt.rc('ytick', labelsize=12)

import warnings
warnings.filterwarnings('ignore')

pd.set_option('display.max_columns',None)
#pd.set_option('display.max_rows',None)
pd.set_option('display.width', 1000)
pd.set_option('display.float_format','{:.2f}'.format)

random.seed(0)
np.random.seed(0)
np.set_printoptions(suppress=True)
```

```
Autosaving every 60 seconds
```

Exploratory Data Analysis

In [2]:

```
df = pd.read_csv("gapminderfinal.csv")
```

In [3]:

```
df
```

Out[3]:

	lifeexpectancy	polityscore	employrate	urbanrate	suicideper100th	hivrate	democorecat	incomecat
0	48.67	0	55.70	24.04	6.68	0.40	1	3
1	76.92	9	51.40	46.72	7.70	0.40	3	1
2	73.13	2	50.50	65.22	4.85	0.10	2	2
3	73.13	6	58.70	88.92	5.36	0.40	3	4
4	51.09	-2	75.70	56.70	14.55	2.00	1	1
...	...	...	...	...	...	...	...	...
208	75.18	-7	71.00	27.84	11.65	0.40	0	1
209	72.83	6	32.00	71.90	8.26	0.40	3	3
210	65.49	-2	39.00	30.64	6.27	0.40	1	0
211	49.02	7	61.00	35.42	12.02	13.50	3	0
212	51.38	1	66.80	37.34	13.91	14.30	2	0

213 rows × 8 columns

ANOVA

We test democracy score with life expectency here.

State the hypothesis

- H_0 : μ_1 = μ_2 (the two means are equal)
- H_1 : At least one of the group means differ

One Way ANOVA

In [4]:

```
df.columns
```

Out[4]:

```
Index(['lifeexpectancy', 'polityscore', 'employrate', 'urbanrate', 'suicideper100th', 'hivrate', 'democoreca
t', 'incomecat'], dtype='object')
```

In [5]:

```
anova_df = df[["lifeexpectancy",'democorecat']]
```

In [6]:

```
anova_df
```

Out[6]:

	lifeexpectancy	democorecat
0	48.67	1
1	76.92	3
2	73.13	2
3	73.13	3
4	51.09	1
...	...	...
208	75.18	0
209	72.83	3
210	65.49	1
211	49.02	3
212	51.38	2

213 rows × 2 columns

In [7]:

```
mod = ols('lifeexpectancy~C(democorecat)', data=anova_df).fit()
```

In [8]:

```
aov_table = sm.stats.anova_lm(mod,typ=2)
```

In [9]:

```
aov_table
```

Out[9]:

	sum_sq	df	F	PR(>F)
C(democorecat)	4389.50	3.00	22.25	0.00
Residual	13744.43	209.00	NaN	NaN

The p value obtained from ANOVA analysis is significant (p < 0.05), and therefore, we conclude that there are significant differences among democratic scores.

Running a Chi-Square Test of Independence

Run a Chi-Square Test of Independence.

You will need to analyze and interpret post hoc paired comparisons in instances where your original statistical test was significant, and you were examining more than two groups (i.e. more than two levels of a categorical, explanatory variable).

Chi-square

State the hypothesis:

- H_0 : The proportion of democracy score is independent of income category
- H_1 : The proportion of democracy score is associated with income category

In [10]:

```
df.columns
```

Out[10]:

```
Index(['lifeexpectancy', 'polityscore', 'employrate', 'urbanrate', 'suicideper100th', 'hivrate', 'democoreca
t', 'incomecat'], dtype='object')
```

In [11]:

```
#Create a Cross-tab table

cont_table  = pd.crosstab(df['democorecat'], df['incomecat'])
cont_table
```

Out[11]:

incomecat	**0**	**1**	**2**	**3**	**4**
democorecat					
0	3	7	5	7	3
1	15	5	4	2	1
2	9	6	4	0	0
3	16	24	30	33	39

In [12]:

```
chi_square = scipy.stats.chi2_contingency(cont_table, correction = True)
```

In [13]:

```
print(f"Chi score is", chi_square[0])
```

```
Chi score is 52.99091365364315
```

In [14]:

```
print("P-value is", chi_square[1])
```

```
P-value is 4.1347271278321373e-07
```

In [15]:

```
print("Degrees of freedom is", chi_square[2])
```

```
Degrees of freedom is 12
```

The p value obtained from Chi Square Test is significant (p < 0.05), and therefore, we conclude that there are significant differences among both groups.

Generating a Correlation Coefficient

Generate a correlation coefficient.

Note 1: Two 3+ level categorical variables can be used to generate a correlation coefficient if the the categories are ordered and the average (i.e. mean) can be interpreted. The scatter plot on the other hand will not be useful. In general the scatterplot is not useful for discrete variables (i.e. those that take on a limited number of values).

Note 2: When we square r, it tells us what proportion of the variability in one variable is described by variation in the second variable (a.k.a. RSquared or Coefficient of Determination).

State the hypothesis:

- H_0 : Total Employees is not correlated with urban population
- H_1 : Total Employees is correlated with urban population

In [16]:

```
df.columns
```

Out[16]:

```
Index(['lifeexpectancy', 'polityscore', 'employrate', 'urbanrate', 'suicideper100th', 'hivrate', 'democoreca
t', 'incomecat'], dtype='object')
```

In [17]:

```
pearson_correlation = scipy.stats.pearsonr(df['employrate'], df['urbanrate'])
```

In [18]:

```
print("Pearson's correlation coefficient is", pearson_correlation[0])
```

```
Pearson's correlation coefficient is -0.2897618551651312
```

In [19]:

```
print("P-value is", pearson_correlation[1])
```

```
P-value is 1.7344734052971086e-05
```

The p value obtained from Pearson Correlation is significant (p < 0.05), and therefore, we conclude that there are significant differences among both groups.

Testing a Potential Moderator

Run an ANOVA, Chi-Square Test or correlation coefficient that includes a moderator.

In [20]:

```
df.columns
```

Out[20]:

```
Index(['lifeexpectancy', 'polityscore', 'employrate', 'urbanrate', 'suicideper100th', 'hivrate', 'democoreca
t', 'incomecat'], dtype='object')
```

In [21]:

```
pearson_correlation1 = scipy.stats.pearsonr(df['lifeexpectancy'], df['suicideper100th'])
pearson_correlation1
```

Out[21]:

```
(-0.10752768587698334, 0.11767201554751555)
```

In [22]:

```
pearson_correlation2 = scipy.stats.pearsonr(df['suicideper100th'], df['employrate'])
pearson_correlation2
```

Out[22]:

```
(0.030017726820558456, 0.6631164170147359)
```

In [23]:

```
pearson_correlation3 = scipy.stats.pearsonr(df['lifeexpectancy'], df['employrate'])
pearson_correlation3
```

Out[23]:

```
(-0.3210761202073892, 1.7039660975096335e-06)
```

It seems that suicide cases has no impact on total employees and life expectancy.

Python code done by Dennis Lam

In [ ]: