correlations_calculations

November 17, 2022

1 Correlations Calculations

Below are the python-assisted calculations for Chi-Squared and Correlations

1.1 Chi-Squared on a Toy Data Set

[1]:
```
import pandas as pd
import numpy as np
```

First, read our data in. In this case, we only care about the first two columns of the dataset, as they pertain to `HourWatched` and `HourPlayed`

[2]:
```
df = pd.read_csv('responses.csv')

df = df.iloc[:, 74:76]
df = df.rename(columns={'74. How many hours do you play video games in an␣
 ↪average week? (PC, console, phone...)': 'HourPlayed', '75. How many hours do␣
 ↪you spend watching gaming streams in a week?': 'HourWatched'})
df.iloc[:, 0:2] = df.iloc[:, 0:2].replace(to_replace=r'[^\d.]+', value='',␣
 ↪regex=True)
df.iloc[:, 0:2] = df.iloc[:, 0:2].replace(to_replace='', value=0)
df.iloc[:, 0:2] = df.iloc[:, 0:2].replace(np.nan, 0)

df['HourPlayed'] = df['HourPlayed'].astype(float)
df['HourWatched'] = df['HourWatched'].astype(float)
df = df.drop(df[df.HourPlayed == 0].index)

df
```

[2]:

	HourPlayed	HourWatched
0	15.0	10.0
2	20.0	0.0
3	20.0	8.0
7	3.0	0.0
9	5.0	0.0
..	…	…
104	10.0	0.0
106	5.0	0.0

```
107         30.0          4.0
108         28.0          2.0
109         20.0          1.0

[70 rows x 2 columns]
```

Now, let's go ahead and calculate 3 random rows to create our toy dataframe, then do calculations on those. To avoid running into re-run errors, we'll just use google to randomly use three numbers between 0 and 70.

Random Number #1: 2

Random Number #2: 32

Random Number #3: 67

Then, let's reduce our dataframe to just those 3 rows:

[3]:
```
toy_index = [2, 32, 67]

df = df.iloc[toy_index]

df.index = ["Random #1", "Random #2", "Random #3"]

df
```

[3]:
```
           HourPlayed  HourWatched
Random #1        20.0          8.0
Random #2         8.0          4.0
Random #3        30.0          4.0
```

Add the row and column totals for easy visualization and plugging into our formula

[4]:
```
totals = df.copy()

totals.loc['col_total'] = totals.sum(axis=0)
totals['row_total'] = totals.sum(axis=1)

totals
```

[4]:
```
           HourPlayed  HourWatched  row_total
Random #1        20.0          8.0       28.0
Random #2         8.0          4.0       12.0
Random #3        30.0          4.0       34.0
col_total        58.0         16.0       74.0
```

[5]:
```
expected = np.outer(totals["row_total"][0:3], totals.loc['col_total'][0:2]) / 74

expected = pd.DataFrame(expected)
```

```
expected.columns = ['HourPlayed', 'HourWatched']
expected.index = ['Random #1', 'Random #2', 'Random #3']

expected
```

[5]:

	HourPlayed	HourWatched
Random #1	21.945946	6.054054
Random #2	9.405405	2.594595
Random #3	26.648649	7.351351

Now, we can write our chi-squared stat test:

[6]:
```
chi_squared_stat = (((df - expected)**2)/expected).sum().sum()
chi_squared_stat
```

[6]: 3.7185839853182 65

Now that we have our chi-squared value, we can do our critical value tests with a Confidence Interval of 95%. In this case, our degrees of freedom will be, `df = 2` since our table is 3x2, so we do 2 x 1.

[7]:
```
from scipy.stats import chi2

critical_value = chi2.ppf(q=0.95, df=2)
print("Critical Value:",critical_value)

p_value = 1 - (chi2.cdf(x=chi_squared_stat, df=2))
print("P Value:", p_value)
```

```
Critical Value: 5.991464547107979
P Value: 0.1557828867596316
```

Because our p-value is **greater than** the required threshold of 0.05, we can accept the null hypothesis that there is an independence between our two variables. It is important to note that the randomly selected elements all came from students who both play video games **AND** watch streams.

When it comes the the entire dataset, the majority of students tend to do one or the other, with most students playing video games and **NOT** watching streams.

1.2 Correlation on a Toy Data Set

Utilizing the same dataset as aforemention, we are just verifying correlation.

[8]:
```
corr1 = df['HourWatched']
corr2 = df['HourPlayed']

corr2.corr(corr1)
```

[8]: 0.05241424183609587

Once again, is important to note that since this particular toy dataset dealt with entries where students watched and played video games, it may appear that they do so independently of each other, and no clear correlation may be observed. A correlation coefficient of 0.05 is nearly neglible, although very slightly positive.

In comparison to entire dataset, where the correlation coefficient was closer to 0.30!