CS 3300 Data Science - Lab 3: Exploratory Data Analysis with Statistical Testing

Stuart Harley

Introduction

In this lab, we are exploring statistical tests. We first go through an introductory problem that explores the Two-Sample T-Test. We then identify some of the important information about a test for correlation based on linear regression, the Kruskal-Wallis test, and the Ch-Squared test of Goodness of Fit. Finally, we explore the San Francisco real estate data that we have cleaned in Lab 1. In Lab 2 we made predictions about correlations and associations between features which we will confirm or deny in this lab using the appropriate statistical tests.

Importing Libraries

In [1]:
```
import pandas as pd
from scipy import stats
```

Part I: Review of Statistical Tests

Let' say you decide you want to know if playing video games impacts students' grades. You set up a survey which asks students two questions:

1. Do you play video games regularly? Yes / No
2. What is your GPA?

a. Hypothesis: Students' playing of video games will not affect their grades.

You now decide to look at the survey results. You have 100 responses! 68 students said they play video games regularly, while 32 students said they did not. The 68 gamers have an average GPA of 3.4 with a standard deviation of 1.2, while the 32 non-gamers have an average GPA of 3.3 with a standard deviation of 1.1.

b. You use a Two-Sample T-Test when you have a boolean variable and a numerical variable. The situation above meets these criteria. The T-Test assumes that the data comes from a normal distribution which may not hold here.

c. Null Hypothesis: The means of the student's GPAs are equal for the group that plays video games and the group that doesn't.

Alternative Hypothesis: The means of the student's GPAs are not equal for the group that plays video games and the group that doesn't.

d. Performing a T-Test on the data from above.

```
In [2]: stats.ttest_ind_from_stats(mean1=3.4, std1=1.2, nobs1=68, \
                                  mean2=3.3, std2=1.1, nobs2=32)

Out[2]: Ttest_indResult(statistic=0.39893881176878243, pvalue=0.6908062583072547)
```

Our calculated p-value is 0.69. We are interpreting our p-value with a significance threshold of 0.01. Therefore, we are not able to reject the null hypothesis because our p-value is greater than the threshold. This means that the differences in GPAs of the two groups are not statistically significant.

e. The evidence from the T-Test supports my original hypothesis, that students who play video games will not have their grades affected.

Part II: Exploring Additional Statistical Tests

1. Test for Correlation based on Linear Regression

a. Used for 2 numerical variables

b. Null Hypothesis: The slope of the best-fit line is equal to zero.

Alternative Hypothesis: The slope of the best-fit line not is equal to zero.

c. If the test indicates statistical significance, then the two variables are correlated. If the test does not indicate statistical significance, then the two variables are not correlated.

1. Kruskal-Wallis Test

a. Used for 1 categorical variable and 1 numerical variable. You convert the numerical variables to relative ranked value. This test does not assume that the data comes from normal distributions.

b. Null Hypothesis: The mean ranks of the groups are the same.

c. If the test indicates statistical significance, then the variables are associated. If the test does not indicate statistical significance, then the variables are not associated.

1. Chi-Squared Test of Goodness of Fit

a. Used for 1 categorical variable when you want to see if the number of observations in each category fits a theoretical expectation. The sample size is assumed to be large.

b. Null Hypothesis: The number of observations in each category is sampled form the expected distribution.

Alternative Hypothesis: The number of observations in each category is not sampled from the expected distribution.

c. If the test indicates statistical significance, then the data is not sampled from the expected distribution. If the test does not indicate statistical significance, then the data is sampled from the expected distribution.

Part III: Regression on Price

a. Loading in the cleaned data set from Lab 1.

In [3]:
```
df = pd.read_csv('CleanedSacramentorealestatetransactions.csv', \
                 dtype={'city': 'category', 'zip': 'category', \
                        'state': 'category', 'beds': 'category', \
                        'baths': 'category', 'type': 'category', \
                        'street_type': 'category'})
df.head()
```

Out[3]:

	street	city	zip	state	beds	baths	sq__ft	type	sale_date	price
0	3526 HIGH ST	SACRAMENTO	95838	CA	2	1	836	Residential	Wed May 21 00:00:00 EDT 2008	59222
1	51 OMAHA CT	SACRAMENTO	95823	CA	3	1	1167	Residential	Wed May 21 00:00:00 EDT 2008	68212
2	2796 BRANCH ST	SACRAMENTO	95815	CA	2	1	796	Residential	Wed May 21 00:00:00 EDT 2008	68880
3	2805 JANETTE WAY	SACRAMENTO	95815	CA	2	1	852	Residential	Wed May 21 00:00:00 EDT 2008	69307
4	6001 MCMAHON DR	SACRAMENTO	95824	CA	2	1	797	Residential	Wed May 21 00:00:00 EDT 2008	81900

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 984 entries, 0 to 983
Data columns (total 14 columns):
street         984 non-null object
city           984 non-null category
zip            984 non-null category
state          984 non-null category
beds           984 non-null category
baths          984 non-null category
sq__ft         984 non-null int64
type           984 non-null category
sale_date      984 non-null object
price          984 non-null int64
latitude       984 non-null float64
longitude      984 non-null float64
empty_lot      984 non-null bool
street_type    984 non-null category
dtypes: bool(1), category(7), float64(2), int64(2), object(2)
memory usage: 60.0+ KB
```

b. For each continuous variable, fitting a simple linear regression model, estimating the Pearson correlation coefficient (r-value), and the statistical significance (p-value) of the correlation against the price of the property.

In [5]:
```
slope, intercept, r, sq_ft_p, stderr = stats.linregress(df['price'], df['sq__ft'])
slope, intercept, r, lat_p, stderr = stats.linregress(df['price'], df['latitude'])
slope, intercept, r, long_p, stderr = stats.linregress(df['price'], df['longitude'])
```

Variable	p-value	Statistically Significant Relationship at Threshold = 0.01
Sq Ft	3.386e-27	Yes
Latitude	.214	No
Longitude	7.281e-20	Yes

c. For each categorical variable, using a Kruskal-Wallis test to test for an association between the categorical varible and the price of the property.

In [6]:
```
samples_by_city = []
for value in set(df['city']):
    mask = df['city'] == value
    samples_by_city.append(df['price'][mask])
stat, city_p = stats.kruskal(*samples_by_city)
```

In [7]:
```
samples_by_zip = []
for value in set(df['zip']):
    mask = df['zip'] == value
    samples_by_zip.append(df['price'][mask])
stat, zip_p = stats.kruskal(*samples_by_zip)
```

Can't compute for State because there is only 1 category in state. Therefore, there is also no association because every price has the same state value.

In [8]:
```
samples_by_beds = []
for value in set(df['beds']):
    mask = df['beds'] == value
    samples_by_beds.append(df['price'][mask])
stat, beds_p = stats.kruskal(*samples_by_beds)
```

In [9]:
```
samples_by_baths = []
for value in set(df['baths']):
    mask = df['baths'] == value
    samples_by_baths.append(df['price'][mask])
stat, baths_p = stats.kruskal(*samples_by_baths)
```

In [10]:
```
samples_by_type = []
for value in set(df['type']):
    mask = df['type'] == value
    samples_by_type.append(df['price'][mask])
stat, type_p = stats.kruskal(*samples_by_type)
```

In [11]:
```
samples_by_empty_lot = []
for value in set(df['empty_lot']):
    mask = df['empty_lot'] == value
    samples_by_empty_lot.append(df['price'][mask])
stat, empty_lot_p = stats.kruskal(*samples_by_empty_lot)
```

In [12]:
```
samples_by_street_type = []
for value in set(df['street_type']):
    mask = df['street_type'] == value
    samples_by_street_type.append(df['price'][mask])
stat, street_type_p = stats.kruskal(*samples_by_street_type)
```

Variable	p-value	Statistically Significant Relationship at Threshold = 0.01
City	3.714e-49	Yes
Zip	2.793e-65	Yes
State	NA	No
Beds	7.751e-38	Yes
Baths	9.614e-51	Yes
Type	3.207e-07	Yes
Empty_Lot	.043	No
Street_Type	1.435e-15	Yes

d. The results of the statistical tests mostly line up with the results from my analysis of the visualizations of variable pairs from Lab 2. The only result that disagrees is that the street type is shown to have a significant association with price in the Kruskal-Wallis Test. But I did not determine that association from looking at the visualization in Lab 2. However, I was unsure about this relationship while completing Lab 2, so evidently I was incorrect at the time.

Part IV: Classification of Property Type

a. For each continuous variable, using a Kruskal-Wallis test to test for an association between the continuous variable and the type of the property.

In [13]:
```
samples_by_type = []
for value in set(df['type']):
    mask = df['type'] == value
    samples_by_type.append(df['sq__ft'][mask])
stat, sq_ft_p = stats.kruskal(*samples_by_type)
```

In [14]:
```
samples_by_type = []
for value in set(df['type']):
    mask = df['type'] == value
    samples_by_type.append(df['price'][mask])
stat, price_p = stats.kruskal(*samples_by_type)
```

In [15]:
```
samples_by_type = []
for value in set(df['type']):
    mask = df['type'] == value
    samples_by_type.append(df['latitude'][mask])
stat, latitude_p = stats.kruskal(*samples_by_type)
```

In [16]:
```
samples_by_type = []
for value in set(df['type']):
    mask = df['type'] == value
    samples_by_type.append(df['longitude'][mask])
stat, longitude_p = stats.kruskal(*samples_by_type)
```

Variable	p-value	Statistically Significant Relationship at Threshold = 0.01
Sq Ft	1.694e-12	Yes
Price	3.207e-07	Yes
Latitude	.306	No
Longitude	.802	No

b. For each categorical variable, using a Chi-Squared Test of Independence to test for an association between the categorical variable and the type of the property.

In [17]:
```
combination_counts = df[['type', 'city']]\
                    .groupby(by=['type', 'city'])\
                    .size().unstack(level=0).fillna(0)
chi2, city_p, dof, expected = stats.chi2_contingency(combination_counts)
```

In [18]:
```
combination_counts = df[['type', 'zip']]\
                    .groupby(by=['type', 'zip'])\
                    .size().unstack(level=0).fillna(0)
chi2, zip_p, dof, expected = stats.chi2_contingency(combination_counts)
```

In [19]:
```
combination_counts = df[['type', 'state']]\
                     .groupby(by=['type', 'state'])\
                     .size().unstack(level=0).fillna(0)
chi2, state_p, dof, expected = stats.chi2_contingency(combination_counts)
```

In [20]:
```
combination_counts = df[['type', 'beds']]\
                     .groupby(by=['type', 'beds'])\
                     .size().unstack(level=0).fillna(0)
chi2, beds_p, dof, expected = stats.chi2_contingency(combination_counts)
```

In [21]:
```
combination_counts = df[['type', 'baths']]\
                     .groupby(by=['type', 'baths'])\
                     .size().unstack(level=0).fillna(0)
chi2, baths_p, dof, expected = stats.chi2_contingency(combination_counts)
```

In [22]:
```
combination_counts = df[['type', 'empty_lot']]\
                     .groupby(by=['type', 'empty_lot'])\
                     .size().unstack(level=0).fillna(0)
chi2, empty_lot_p, dof, expected = stats.chi2_contingency(combination_counts)
```

In [23]:
```
combination_counts = df[['type', 'street_type']]\
                     .groupby(by=['type', 'street_type'])\
                     .size().unstack(level=0).fillna(0)
chi2, street_type_p, dof, expected = stats.chi2_contingency(combination_counts
)
```

Variable	p-value	Statistically Significant Relationship at Threshold = 0.01
City	.969	No
Zip	1.069e-4	Yes
State	1.0	No
Beds	1.817e-67	Yes
Baths	6.407e-43	Yes
Empty_Lot	.162	No
Street_Type	2.361e-16	Yes

c. The results of the statistical tests mostly line up with the results from my analysis of the visualizations of variable pairs from Lab 2. The only result that disagrees is that the city is shown to not have a significant association with type in the Chi-Squared Test of Independence. But I did determine that association from looking at the visualization in Lab 2. However, I was unsure about this relationship while completing Lab 2, becuase the heatmaps were difficult to read since the amount of types was very uneven, so evidently I was incorrect at the time.

Conclusion

Visualizing your data in plots and graphs is helpful to better understand your data and make some preliminary assumptions about correlations and associations between features. However, it is also important to run statistical tests on the data in order to confirm what your visualizations appear to tell you.