Analyzing Where Do People Drink?

Description

This Dataset is from the story Dear Mona Followup: Where Do People Drink The Most Beer, Wine And Spirits? (https://fivethirtyeight.com/features/dear-mona-followup-where-do-people-drink-the-most-beer-wine-and-spirits/) The dataset contains Average serving sizes per person such as average wine, spirit, beer servings. As well as several other metrics. You will be asked to analyze the data and predict the total liters served given the servings. See how to share your lab at the end.

You will need the following libraries:

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
```

1.0 Importing the Data

Load the csv:

In [2]:
```
df= pd.read_csv('https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/CognitiveClass/DA0101EN/edx/project/drinks.csv')
```

We use the method `head()` to display the first 5 columns of the dataframe:

In [3]:

```
df.head()
```

Out[3]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	contine
0	Afghanistan	0	0	0	0.0	As
1	Albania	89	132	54	4.9	Europ
2	Algeria	25	0	14	0.7	Afric
3	Andorra	245	138	312	12.4	Europ
4	Angola	217	57	45	5.9	Afric

Question 1: Display the data types of each column using the attribute dtype.

In [4]:

```
df.dtypes
```

Out[4]:

```
country                          object
beer_servings                     int64
spirit_servings                   int64
wine_servings                     int64
total_litres_of_pure_alcohol    float64
continent                        object
dtype: object
```

Question 2 use the method `groupby` to get the number of wine servings per continent:

In [5]:

```
#To find the average of wine_servings per continent and to find out which cont
inent is the largest consumer:

df_wine = df[["wine_servings","continent"]]

df_wine_grp = df_wine.groupby(["continent"], as_index=False).mean()

#The wine_servings are all to be in integer format per the actual table. Ther
e're no decimals.
df_wine_grp["wine_servings"] = df_wine_grp["wine_servings"].astype(int)

df_wine_grp
```

Out[5]:

	continent	wine_servings
0	Africa	16
1	Asia	9
2	Europe	142
3	North America	24
4	Oceania	35
5	South America	62

Question 3: Perform a statistical summary and analysis of beer servings for each continent:

In [6]:

```
df_beer = df[["beer_servings","continent"]]

df_beer_grp = df_beer.groupby(["continent"], as_index=False).mean()

#The wine_servings are all to be in integer format per the actual table. There're no decimals.
df_beer_grp["beer_servings"] = df_beer_grp["beer_servings"].astype(int)

df_beer_grp
```

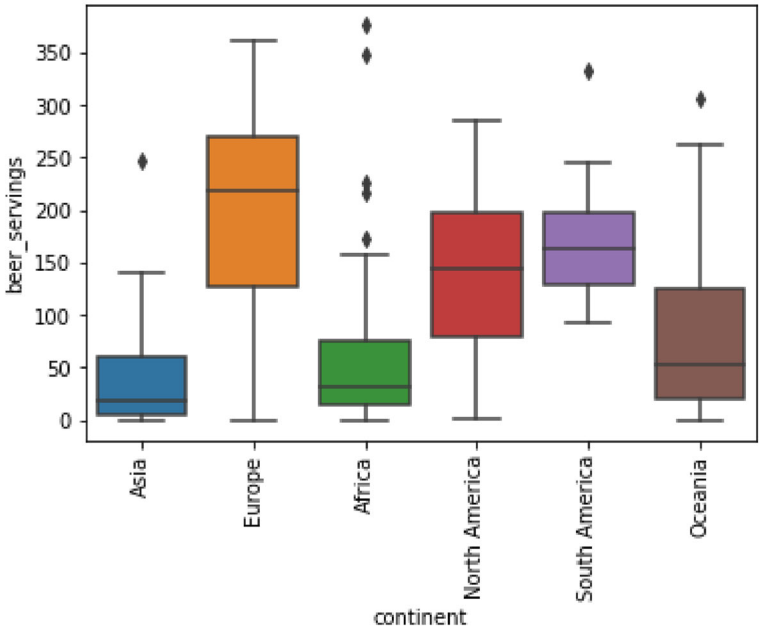
Out[6]:

	continent	beer_servings
0	Africa	61
1	Asia	37
2	Europe	193
3	North America	145
4	Oceania	89
5	South America	175

Question 4: Use the function boxplot in the seaborn library to produce a plot that can be used to show the number of beer servings on each continent.

```
In [7]: import seaborn as sns

        sns.boxplot(x = "continent", y = "beer_servings", data = df)
        plt.xticks(rotation=90)
```

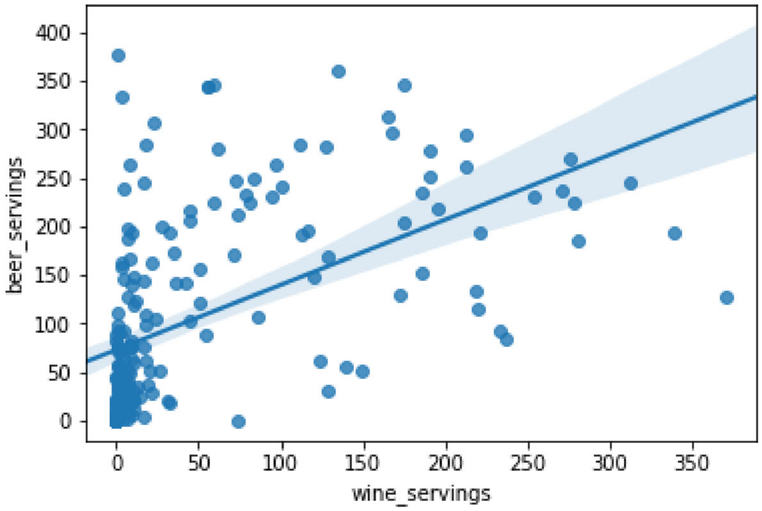
```
Out[7]: (array([0, 1, 2, 3, 4, 5]), <a list of 6 Text xticklabel objects>)
```



Question 5: Use the function `regplot` in the seaborn library to determine if the number of wine servings is negatively or positively correlated with the number of beer servings.

```
In [8]: import seaborn as sns

        sns.regplot(x = "wine_servings", y = "beer_servings", data = df)
```

```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9ee23880f0>
```



Question 6: Fit a linear regression model to predict the `'total_litres_of_pure_alcohol'` using the number of `'wine_servings'` then calculate R^2:

In [9]:
```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()

X = df[['wine_servings']]
Y = df['total_litres_of_pure_alcohol']

lm.fit(X,Y)

Yhat=lm.predict(X)
Yhat[0:4]
```

Out[9]:
```
array([ 3.15407943,  4.86088833,  3.59658545, 13.01564196])
```

R^2, Slope, and Intercept calculations for Question - 6:

In [10]:
```
# Intercept and Slope are not required by problem statement, but I did it here
just to check if it makes sense, knowing that wine_servings does NOT affect to
tal alcohol.
# Therefore, we should expect a lower slope value for the regression line.

print("Intercept:           ", lm.intercept_, "\nCoefficient (Slope): ", lm.co
ef_, "\nR-Squared:           ", lm.score(X,Y))
```

```
Intercept:            3.1540794346874996
Coefficient (Slope):  [0.03160757]
R-Squared:            0.4456875459787605
```

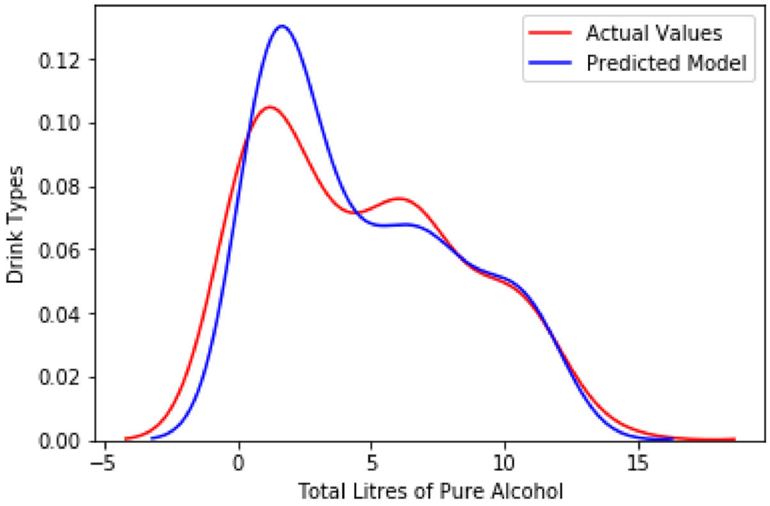
Question 7

Use the list of features to predict the `'total_litres_of_pure_alcohol'` , split the data into training and testing and determine the R^2 on the test data, using the provided code:

Answer 7 (Part 1 - Prediction):

In [11]:
```
from sklearn.linear_model import LinearRegression
lm1 = LinearRegression()

x_data = df[["beer_servings", "wine_servings", "spirit_servings"]]
y_data = df["total_litres_of_pure_alcohol"]

lm1.fit(x_data, y_data)

Yhat = lm1.predict(x_data)

# Plotting predicted vs. actual values
ax1 = sns.distplot(y_data, hist=False, color="r", label="Actual Values")
sns.distplot(Yhat, hist=False, color="b", label="Predicted Model" , ax=ax1)
plt.xlabel('Total Litres of Pure Alcohol')
plt.ylabel('Drink Types')
```

Out[11]:
```
Text(0, 0.5, 'Drink Types')
```



Answer 7 (Part 2 - Splitting):

In [12]:
```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size=
0.15, random_state=1)

print("number of test samples (15%):     ", x_test.shape[0])
print("number of training samples (85%):",x_train.shape[0])
```

```
number of test samples (15%):     29
number of training samples (85%): 164
```

Answer 7 (Part 3 - R^2 Calculation):

```
In [13]: lm1.fit(x_test, y_test)

         Yhat1 = lm1.predict(x_test)

         R_sqrd = lm1.score(x_test, y_test)

         print("R_Squared = ", R_sqrd)
```

```
R_Squared =  0.6968078799715802
```

Question 8 : Create a pipeline object that scales the data, performs a polynomial transform and fits a linear regression model. Fit the object using the training data in the question above, then calculate the R^2 using. the test data. Take a screenshot of your code and the R^2. There are some hints in the notebook:

```
'scale'

'polynomial'

'model'
```

The second element in the tuple contains the model constructor

```
StandardScaler()

PolynomialFeatures(include_bias=False)

LinearRegression()
```

```
In [25]: from sklearn.pipeline import Pipeline
         from sklearn.preprocessing import StandardScaler,PolynomialFeatures
         from sklearn.metrics import r2_score

         # Creating a pipeline object that scales the data, performs a polynomial trans
         form and fits a linear regression model:

         Input=[('scale',StandardScaler()), ('polynomial', PolynomialFeatures(include_b
         ias=False)), ('model',LinearRegression())]

         pipe=Pipeline(Input)
         print(pipe)

         # Fit the object using the training data in the question-7:

         pipe.fit(x_train,y_train)

         # calculate the R^2 using the test data:

         ypipe=pipe.predict(x_test)
         r_squared = r2_score(y_test, ypipe)

         print('The R-square value is: ', r_squared)
```

```
Pipeline(memory=None,
     steps=[('scale', StandardScaler(copy=True, with_mean=True, with_std=Tru
e)), ('polynomial', PolynomialFeatures(degree=2, include_bias=False, interact
ion_only=False)), ('model', LinearRegression(copy_X=True, fit_intercept=True,
n_jobs=None,
         normalize=False))])
The R-square value is:  0.6569256100620995

/opt/conda/envs/Python36/lib/python3.6/site-packages/sklearn/preprocessing/da
ta.py:645: DataConversionWarning: Data with input dtype int64 were all conver
ted to float64 by StandardScaler.
  return self.partial_fit(X, y)
/opt/conda/envs/Python36/lib/python3.6/site-packages/sklearn/base.py:467: Dat
aConversionWarning: Data with input dtype int64 were all converted to float64
by StandardScaler.
  return self.fit(X, y, **fit_params).transform(X)
/opt/conda/envs/Python36/lib/python3.6/site-packages/sklearn/pipeline.py:331:
DataConversionWarning: Data with input dtype int64 were all converted to floa
t64 by StandardScaler.
  Xt = transform.transform(Xt)
```

Question 9: Create and fit a Ridge regression object using the training data, setting the regularization parameter to 0.1 and calculate the R^2 using the test data. Take a screenshot of your code and the R^2

```
In [35]: from sklearn.linear_model import Ridge

         RigeModel = Ridge(alpha=0.1)

         RigeModel.fit(x_train, y_train)

         print("R-Squared = ", RigeModel.score(x_test, y_test))

         R-Squared =  0.617605921385483
```

Question 10 : Perform a 2nd order polynomial transform on both the training data and testing data. Create and fit a Ridge regression object using the training data, setting the regularization parameter to 0.1. Calculate the R^2 utilizing the test data provided. Take a screen-shot of your code and the R^2.

```
In [37]: from sklearn.linear_model import Ridge

         pr=PolynomialFeatures(degree=2)
         x_train_pr=pr.fit_transform(x_train)
         x_test_pr=pr.fit_transform(x_test)

         RigeModel = Ridge(alpha=0.1)

         RigeModel.fit(x_train, y_train)

         print("R-Squared = ", RigeModel.score(x_test, y_test))

         R-Squared =  0.617605921385483
```

CLICK HERE (https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/share-notebooks.html\) to see how to share your notebook

Sources

Dear Mona Followup: Where Do People Drink The Most Beer, Wine And Spirits? (https://fivethirtyeight.com/features/dear-mona-followup-where-do-people-drink-the-most-beer-wine-and-spirits/) by By Mona Chalabi , you can download the dataset here (https://github.com/fivethirtyeight/data/tree/master/alcohol-consumption).