Technical Test - Product Data Analyst

Section 1 - SQL

Table Name: Business

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **id** | **Business Number** | **Business Start Date** | **Region Id** | **Industry** | **Staff Numbers** | **Annual Net Profit** |
| 1 | 9429041352323 | 2016-01-28 | 3 | Catering | 53 | 218,214.25 |
| 2 | 4537898722468 | 2012-05-15 | 7 | Construction | 148 | 1,024,489.29 |
| 3 | ... |  |  |  |  |  |

Table Name: Region

|  |  |  |
| --- | --- | --- |
| **Region Id** | **Region Name** | **Region Code** |
| 1 | United States | US |
| 2 | United Kingdom | UK |
| 3 | Australia | AU |
| 4 | ... |  |

Part A:

Given the tables above, write a SQL query to return the Business Number, Business Start Date, Region Name, Industry, and Annual Net Profit for businesses created in the years 2012 to 2015.

SELECT

b.Business\_Number,

b.Business\_Start\_Date,

r.RegionName,

b.Industry,

b.Annual\_Net\_profit

FROM business b

INNER JOIN region r

ON b.RegionId = r.﻿RegionId

WHERE b.Business\_Start\_Date between CAST('2012-01-01' as date) and CAST('2016-01-01' as date)

Part B:

Write a SQL query that returns, for each region and industry combination, the *Average* *Annual Net Profit,* and an extra columnwith the number of businesses considered in the calculation*.*

SELECT

r.RegionName,

b.Industry,

COUNT(distinct b.Business\_Number) as Number\_of\_Business,

AVG(b.Annual\_Net\_Profit) as Avg\_Annual\_Net\_profit

FROM business b

INNER JOIN Region r

ON b.RegionId = r.﻿RegionId

GROUP BY r.RegionName , b.Industry

Section 2 - R / Python

For this question use your preference of either R or Python with the included CSV file. Code and document your process (if you create a script that’s ok, just provide relevant documentation as part of your script or as a separate file). Include any library dependencies and setup needed, as well as any assumptions you established beforehand. Start from reading the file:

Part A:

Continue the notebook to show the following values of the Total income: Minimum, Mean, Maximum.

Python:

|  |
| --- |
| ## Import data and produce outputs  import pandas as pd  import numpy as np  from sklearn.linear\_model import LinearRegression  from sklearn import preprocessing  # For statistics. Requires statsmodels 5.0 or more  from statsmodels.formula.api import ols  # Analysis of Variance (ANOVA) on linear models  from statsmodels.stats.anova import anova\_lm  import seaborn as sns  ## Read in the Annual Enterprise Survey data  df = pd.read\_csv('aes-2015-csv.csv')  #### Add your code here ####  ### Checking and cleaning data  # the value column is string type and must be changed to integer.  df.Value = pd.to\_numeric(df.Value, errors='coerce')  df = df.dropna(subset=['Value'])  df.Value = df.Value.astype(int)  # Changing year column to string format. I made the decision to treat Year column as categorical variable.  df['Year']=df['Year'].astype(str)  # check again whether Value column is transformed to integer.  print(len(df),df.isnull().values.any())  print(df.describe())  ## Part A - Calculation of mean, max, and min of Value column, filtering for Total income  #filtering entry rows to only Total income variable name  df=df[df.Variable\_name=='Total income']  print(len(df))  # description of Value, mean, max, and min value of Total income.  print('Value mean=', round(df.Value.describe().mean(),2))  print('Value max=', round(df.Value.describe().max(),2))  print('Value min=', round(df.Value.describe().min(),2)) |

Part B:

Continue the notebook to perform a linear regression and analysis.

1. Run a linear regression with Total Income as the dependent variable and your choice of four other variables as the independent variables (NOTE: Don’t limit your analysis to the variables presented; feel free to try aggregations or categorical variables from the original variables). Explain why you chose those independent variables, as well as the what the R-squared value for this regression represents.

Literature review in accounting and variables that influence Total income of businesses shows that **Firm size** , **Productivity** , **Liquidity**, and **Leverage ratio** are appropriate choice of independent variables to predict the Total income. ['Total assets', 'Return on total assets', 'Quick ratio’, ‘Leverage ratio'] can be an appropriate choice for independent variables to predict the Total income.

**Firm size (**'Total assets') is calculated as the logarithm of Fixed tangible assets.

**Productivity (**'Return on total assets') of industry is given as the column of Return on total assets.

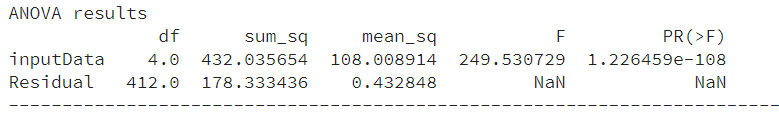
**Liquidity** ('Quick ratio’) of industry is industry is given as the column of Quick ratio

**Leverage ratio** is calculated by (Current liability+Other liabilities)/Shareholders funds or owners equity

Other variables such as 'Surplus before income tax',, 'Total expenditure', etc are either Multicollinear or conceptually, do not influence the industry Total income.

R-squared of regression explains the percentage of variance explained by independent variables in the model. It tells us the proportion of variation in the dependent (Total income) variable that has been explained by the model, and it can range between 0 and 1. In current model, **R²: 0.707** which means that **70.7%** of variations in Total income is explained by independent variables.

1. Run an ANOVA on the linear regression result and explain what the *p*-values and residuals are indicating. Give a general analysis of the results.



The null hypothesis is that all the coefficients of independent variables are zero. The P-value for the F test statistic is less than 0.001, providing strong evidence against the null hypothesis. This means that there is a very low probability that all of the coefficients of independent variables are zero.

The squared multiple correlation is equal to the ratio of the model sum of squares to the total sum of squares: R² = SSM/SST = 432.03/(432.03+178.33) = 0.707, indicating that 70.7% of the variability in the "Total income" variable is explained by the "Return on total assets", "Quick ratio", "Total assets", and "Leverage ratio" variables.

1. Include a chart of the correlation between each independent variable you chose and the dependent variable Total income. Document your observations.

|  |
| --- |
|  |
| Fig.1 Correlation between the logarithm of Total income (M$) and the logarithm of Total assets (M$). |

|  |
| --- |
|  |
| Fig.2 Correlation between the logarithm of Total income (M$) and the Leverage ratio. |

|  |
| --- |
|  |
| Fig.3 Correlation between the logarithm of Total income (M$) and the Quick ratio. |

|  |
| --- |
|  |
| Fig.4 Correlation between the logarithm of the Total income (M$) and the Return on total assets. |

The above graphs show the correlation between each independent variable and independent variables (Total income).

There is a strong positive correlation between Total assets and Total income (R-sq =0.68). When the Total assets of industry increases, Total income tend to increase.

There is a weak positive correlation between Leverage ratio and Total income (R-sq = 0.03). When the Leverage ratio of industry increases, Total income tend to increase.

There is a weak negative correlation between Quick ratio and Total income (R-sq = 0.06). When the Quick ratio of industry increases, Total income tend to decrease.

There is a weak negative correlation between Return on total assets and Total income (R-sq = 0.01). When the Return on total assets of industry increases, Total income tend to decrease.

Executive Summary:

|  |
| --- |
| ## Import data and produce outputs  import pandas as pd  import numpy as np  from sklearn.linear\_model import LinearRegression  from sklearn import preprocessing  # For statistics. Requires statsmodels 5.0 or more  from statsmodels.formula.api import ols  # Analysis of Variance (ANOVA) on linear models  from statsmodels.stats.anova import anova\_lm  import seaborn as sns  ## Read in the Annual Enterprise Survey data  df = pd.read\_csv('aes-2015-csv.csv')  #### Add your code here ####  # Part B- Linear regression  ### Checking and cleaning data  #importing the table  # the value column is string type and must be changed to integer.  df.Value = pd.to\_numeric(df.Value, errors='coerce')  df = df.dropna(subset=['Value'])  df.Value = df.Value.astype(int)  # The Year is integer type which I decided to transform it to string.  df['Year']=df['Year'].astype(str)  # pivoting data to extract variables  df=pd.pivot\_table(df,index=["Industry\_code\_NZSIOC",'Year'],columns=['Variable\_name'], values='Value')  df=df.reset\_index(level=['Year'])  ### Dropping columns with more than 60% data missing. Filling the other null values by average of each column.  df=df.drop(['Margin on sales of goods for resale',  'Salaries and wages to self employed commission agents','Sales of goods and services',  'Sales, government funding, grants and subsidies',  'Surplus per employee count(3)'],axis=1)  df=df.fillna(df.mean())  ##### Literature review in accounting and vriables that influence Total income of businesses shows that ['Total assets', 'Return on total assets', 'Quick ratio','Leverage ratio'] can be selected for independent variables.  ##### Firm size is calculated as logarithm of Fixed tangible assets.  ##### Productivity of industry are obtained from Return on total assets.  ##### Liquidity of industry is obtained from Quick ratio  ##### Leverage ratio is calculated by (Current liability+Other liabilities)/Shareholders funds or owners equity  ##### (Variables such as [ 'Surplus before income tax','Total expenditure'] are either Multicollinear or conceptually, they are not influencing the Total income.  # calculation of total Liability  df['Liability']=df['Current liabilities']+df['Other liabilities']  # calculation of leverage ratio = Liability/Shareholders funds or owners equity  df['Leverage ratio']=df['Liability']/df['Shareholders funds or owners equity']  # Logarithmic of total asset  df['Total assets']=np.log(df['Total assets'])  #resetting index  df=df.reset\_index()  ## Linear regression pipeline  # selecting below variables  inputData=df[['Total assets', 'Quick ratio','Leverage ratio','Return on total assets']] #  inputData=df[['Return on total assets']] #  outputData=np.log(df['Total income'])  # scaling the variables  # Get column names first  names = inputData.columns  # Create the Scaler object  scaler = preprocessing.StandardScaler()  # Fit your data on the scaler object  scaled\_df = scaler.fit\_transform(inputData)  scaled\_df = pd.DataFrame(scaled\_df, columns=names)  model\_1=LinearRegression()  model\_1.fit(scaled\_df,outputData)  coefficients=pd.DataFrame({'name':list(scaled\_df),'value':model\_1.coef\_})  print(coefficients)  print('Mean squared error:', np.mean((model\_1.predict(scaled\_df) - outputData) \*\* 2))  print('R²:',model\_1.score(scaled\_df, outputData))  import statsmodels.api as sm  from statsmodels.formula.api import ols  outputData=np.log(df[['Total income']])  inputData=scaled\_df[['Total assets', 'Quick ratio','Leverage ratio','Return on total assets']]  moore\_lm = ols('outputData ~ inputData',  data=scaled\_df).fit()  table = sm.stats.anova\_lm(moore\_lm) # ANOVA DataFrame  print('\nANOVA results' )  print(table)  print('-------------------------------------------------------------------------')  # print('\nModel summary' )  # print(moore\_lm.summary())  ##### the F statistic is equal to sum of square of model divide by total sum of squares (108.00/0.432 = 249.53) . The distribution is F(4, 412),and the probability of observing a value greater than or equal to 249.53 is less than 0.001. The P-value for the F test statistic is is less than 0.001, providing strong evidence against the null hypothesis. This means that there is very low probability that all of the coefficients of independent variables are zero.  ##### The squared multiple correlation R² = SSM/SST = 432.03/(432.03+178.33) = 0.707, indicating that 70.7% of the variability in the "Total income" variable is explained by the "Return on total assets", "Quick ratio", "Total assets", and "Leverage ratio" variables.  # Plot of independent variable versus dependent variable  inputData=['Total assets', 'Leverage ratio','Quick ratio','Return on total assets'] #'Return on total assets',  # df=df[(df['Quick ratio']<200)& (df['Return on total assets']<30)&(df['Leverage ratio']<6) ] # filtering out the extreme values to show correlation more clearly.  df=df[(df['Return on total assets']>-2)& (df['Return on total assets']<30)&(df['Leverage ratio']<6)&(df['Quick ratio']<250) ] # filtering out the extreme values to show correlation more clearly.  for x in inputData:  ax=sns.regplot(x=df[x],y=np.log(df['Total income']))  ax.set(xlabel=x, ylabel='Total income')  plt.savefig(x+".png", dpi =150)  plt.show() |