Final Project Part 5

Table of Contents

EDA

In [1]:

```
import numpy as np
import pandas as pd
import xgboost as xgb
import statsmodels.api as sm
import matplotlib.pyplot as plt
import sklearn.linear_model as lm
from sklearn.model_selection import GridSearchCV, train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR


import seaborn as sns
sns.set(rc = {'axes.titlesize': 24,
             'axes.labelsize': 20,
             'xtick.labelsize': 12,
             'ytick.labelsize': 12,
             'figure.figsize': (12, 6)})
```

In [2]:

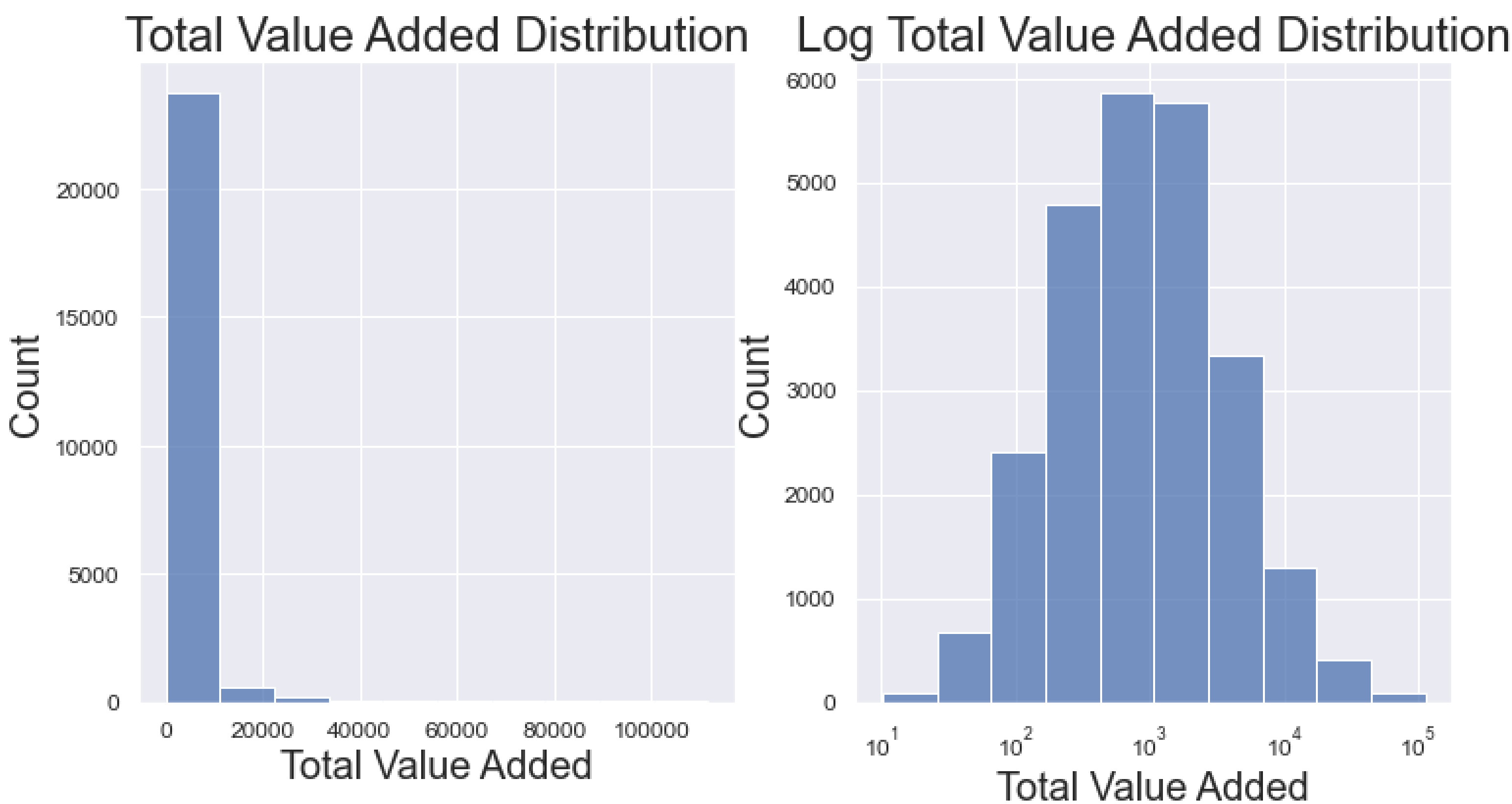
```
sic = pd.read_pickle('sic_fp2.pkl')
```

In [3]:

```
plt.figure()
plt.subplot(1,2,1)
plt.title('Total Value Added Distribution')
plt.xlabel('Total Value Added')
sns.histplot(x = 'vadd', data = sic, bins = 10) #checking to see if the label needs a transformation
```

```
plt.subplot(1,2,2)
plt.title('Log Total Value Added Distribution')
plt.xlabel('Total Value Added')
plt.semilogx()
sns.histplot(x = 'vadd', data = sic, bins = 10) #seeing if log transformation improves label distribution
```

Out[3]: <AxesSubplot:title={'center':'Log Total Value Added Distribution'}, xlabel='Total Value Added', ylabel='Count'>



In [4]:
```
#the label did indeed need a log transformation
sic['l_vadd'] = np.log(sic['vadd']) # creating new column with log transformation of label
sic = sic.drop('vadd', 1) # dropping old label
```

In [5]:
```
#checking to see if any of the features need transformations

# Histograms of emp
plt.figure(figsize = (15,80))
plt.subplot(8,2,1)
```

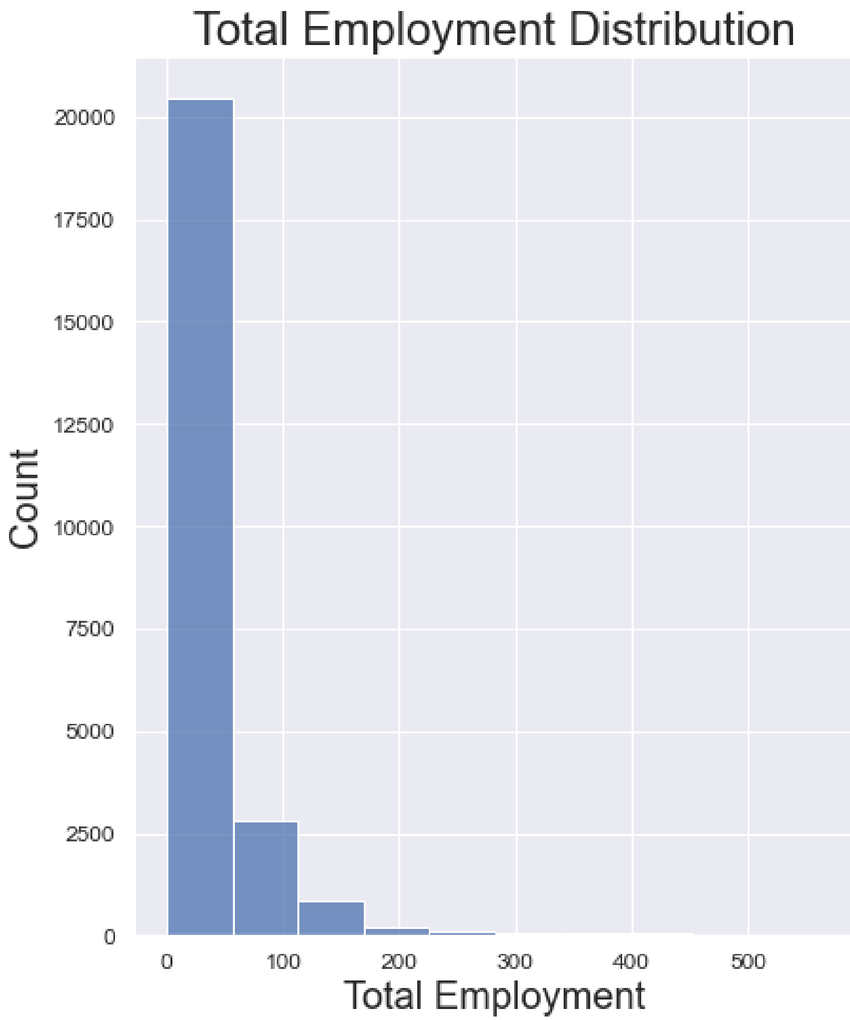
```python
plt.title('Total Employment Distribution')
plt.xlabel('Total Employment')
sns.histplot(x = 'emp', data = sic, bins = 10)

plt.subplot(8,2,2)
plt.title('Log Total Employment Distribution')
plt.xlabel('Total Employment')
plt.semilogx()
sns.histplot(x = 'emp', data = sic, bins = 10)

#Histograms of invest
plt.subplot(8,2,3)
plt.title('Total Capital Expenditure Dist.')
plt.xlabel('Total Capital Expenditure')
sns.histplot(x = 'invest', data = sic, bins = 10)

plt.subplot(8,2,4)
plt.title('Log Total Capital Expenditure Dist.')
plt.xlabel('Total Capital Expenditure')
plt.semilogx()
sns.histplot(x = 'invest', data = sic, bins = 10)


#Histograms of pay
plt.subplot(8,2,5)
plt.title('Total Payroll Distribution')
plt.xlabel('Total Payroll')
sns.histplot(x = 'pay', data = sic, bins = 10)

plt.subplot(8,2,6)
plt.title('Log Total Payroll')
plt.xlabel('Total Payroll')
plt.semilogx()
sns.histplot(x = 'pay', data = sic, bins = 10)

#Histograms of matcost
plt.subplot(8,2,7)
plt.title('Total Material Cost Dist.')
plt.xlabel('Total Material Costs')
sns.histplot(x = 'matcost', data = sic, bins = 10)

plt.subplot(8,2,8)
plt.title('Log Total Material Cost Dist.')
plt.xlabel('Total Material Costs')
plt.semilogx()
sns.histplot(x = 'matcost', data = sic, bins = 10)
```

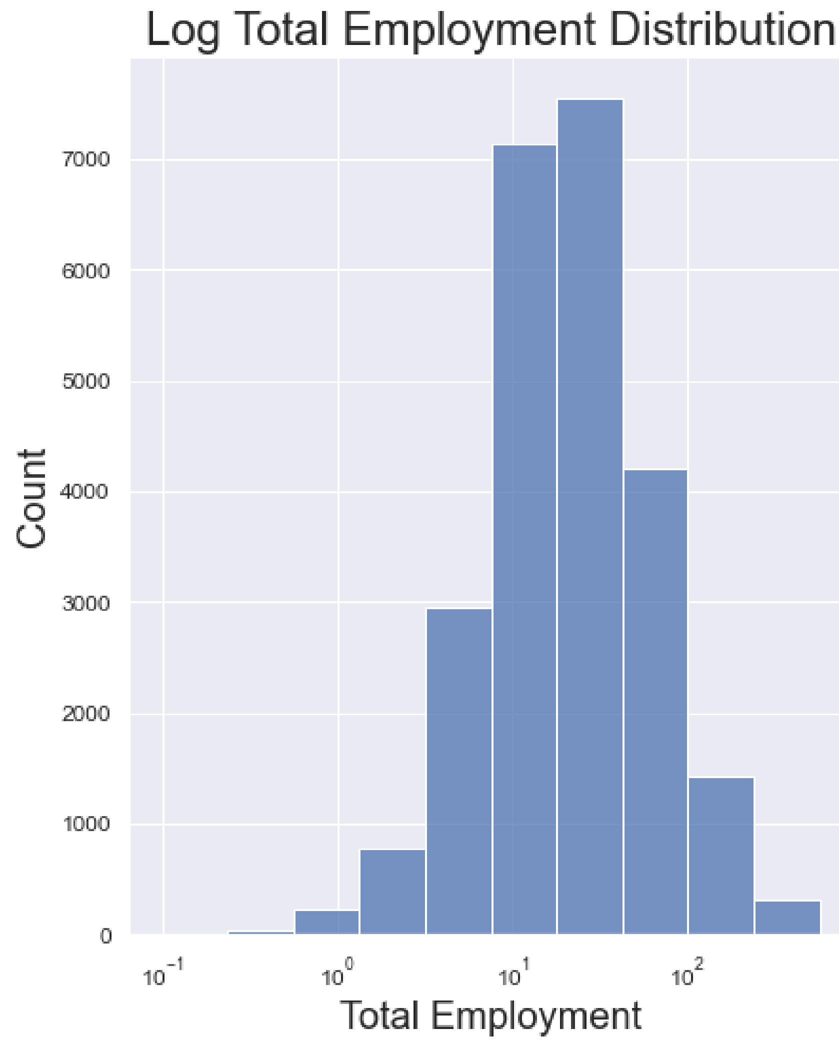
```
#Histograms of vship
plt.subplot(8,2,9)
plt.title('Total Value of Shipments Dist.')
plt.xlabel('Total Value of Shipments')
sns.histplot(x = 'vship', data = sic, bins = 10)

plt.subplot(8,2,10)
plt.title('Total Value of Shipments Dist.')
plt.xlabel('Total Value of Shipments')
plt.semilogx()
sns.histplot(x = 'vship', data = sic, bins = 10)

#Histograms of cap
plt.subplot(8,2,11)
plt.title('Total Real Capital Stock Dist.')
plt.xlabel('Total Real Capital Stock')
sns.histplot(x = 'cap', data = sic, bins = 10)

plt.subplot(8,2,12)
plt.title('Log Total Real Capital Stock Dist.')
plt.xlabel('Total Real Capital Stock')
plt.semilogx()
sns.histplot(x = 'cap', data = sic, bins = 10)

#Histograms of invent
plt.subplot(8,2,13)
plt.title('End of Year Inventory Dist.')
plt.xlabel('End of Year Inventories')
sns.histplot(x = 'invent', data = sic, bins = 10)

plt.subplot(8,2,14)
plt.title('Log End of Year Inventory Dist.')
plt.xlabel('End of Year Inventory')
plt.semilogx()
sns.histplot(x = 'invent', data = sic, bins = 10)

#Histograms of energy
plt.subplot(8,2,15)
plt.title('Electricity & Fuel Cost Dist.')
plt.xlabel('Cost of Electricity & Fuel')
sns.histplot(x = 'energy', data = sic, bins = 10)

plt.subplot(8,2,16)
plt.title('Log Electricity & Fuel Cost Dist. ')
plt.xlabel('Cost of Electricity & Fuel')
```

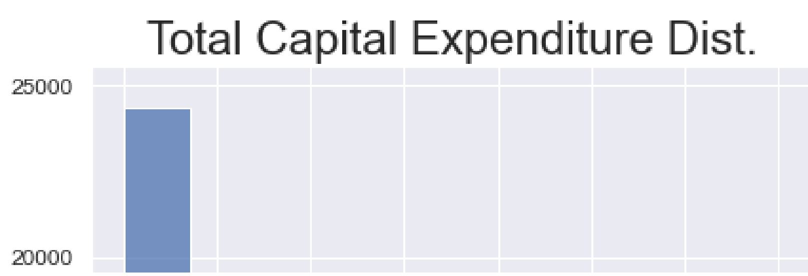
```
plt.semilogx()
sns.histplot(x = 'energy', data = sic, bins = 10)
```

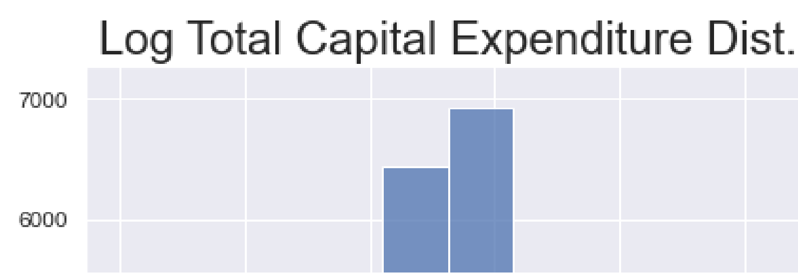
Out[5]:

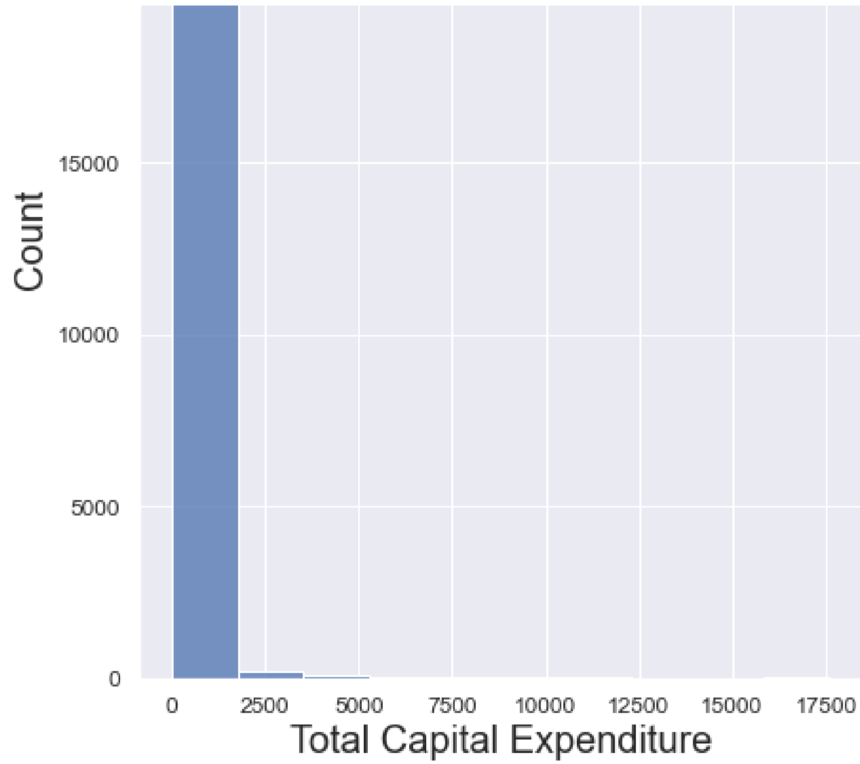
```
<AxesSubplot:title={'center':'Log Electricity & Fuel Cost Dist. '}, xlabel='Cost of Electricity & Fuel', ylabel='Count'>
```

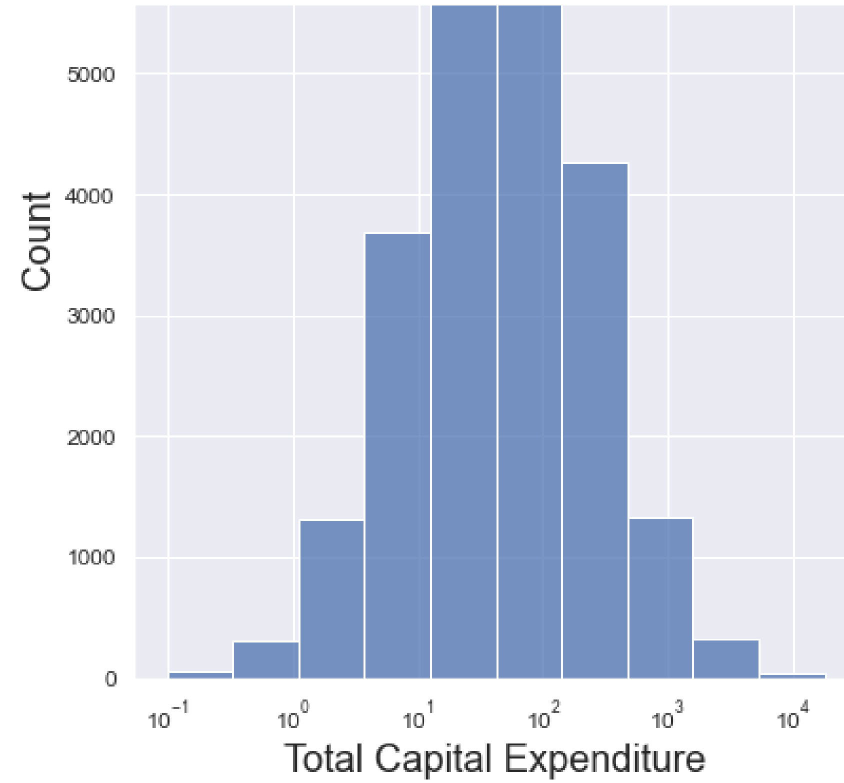




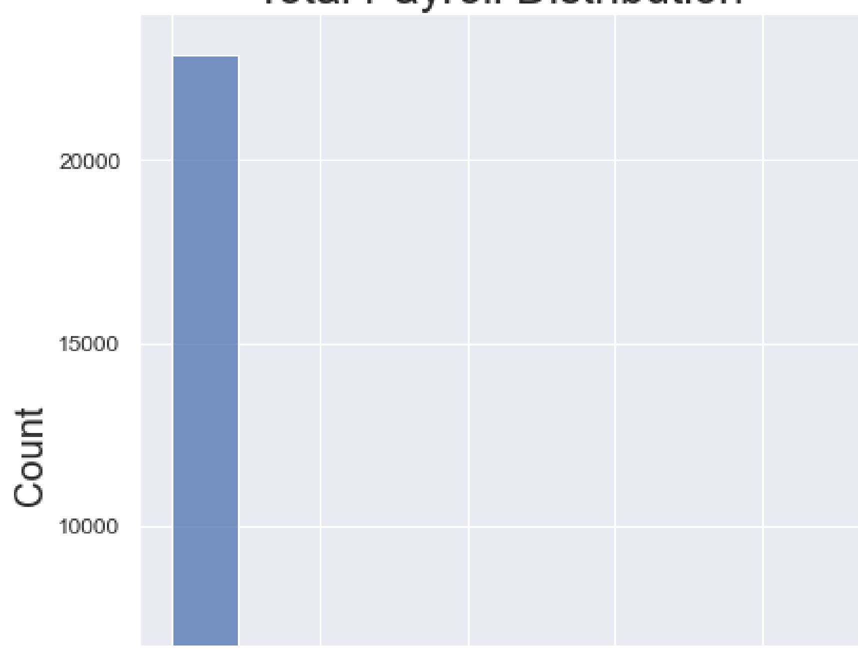




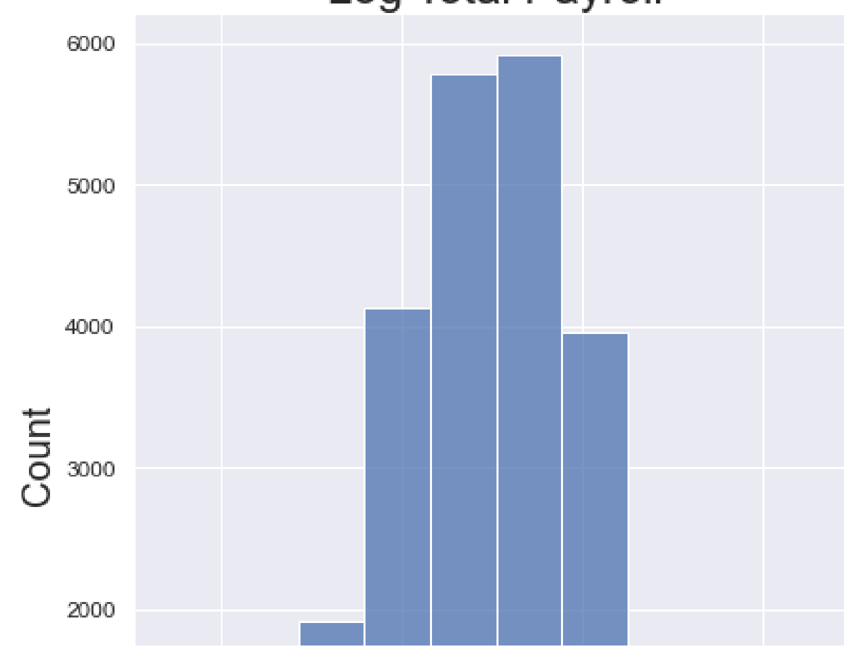


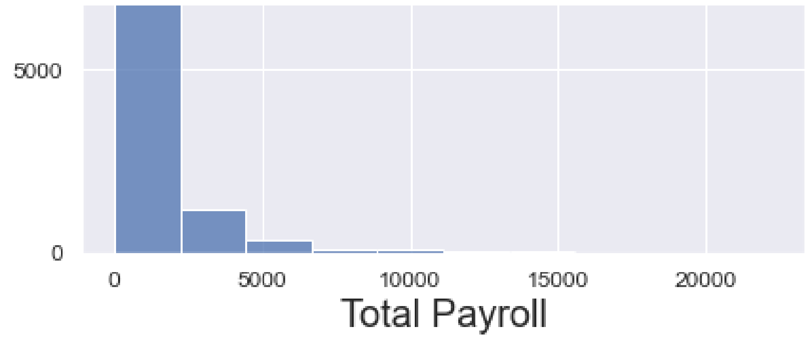


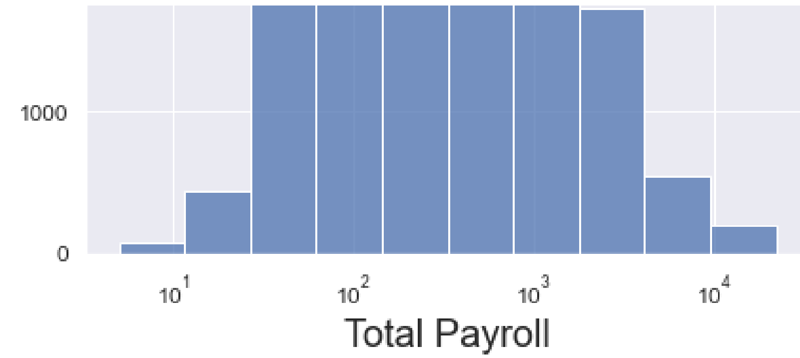
Total Payroll Distribution



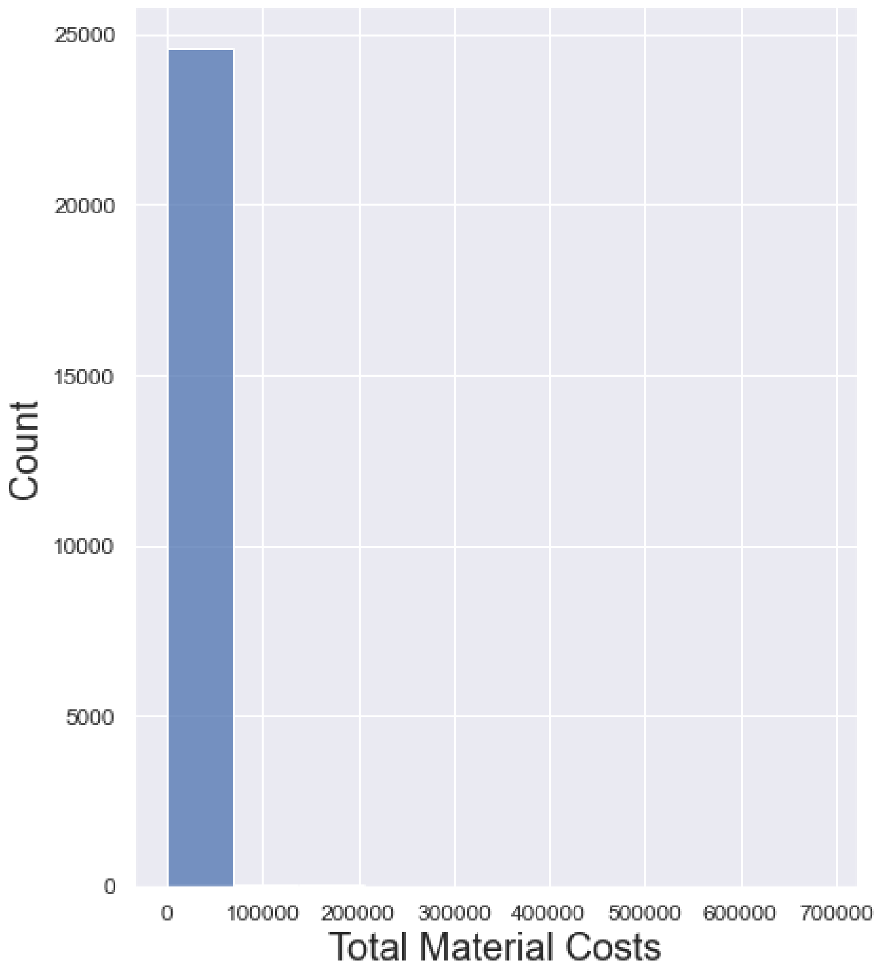
Log Total Payroll



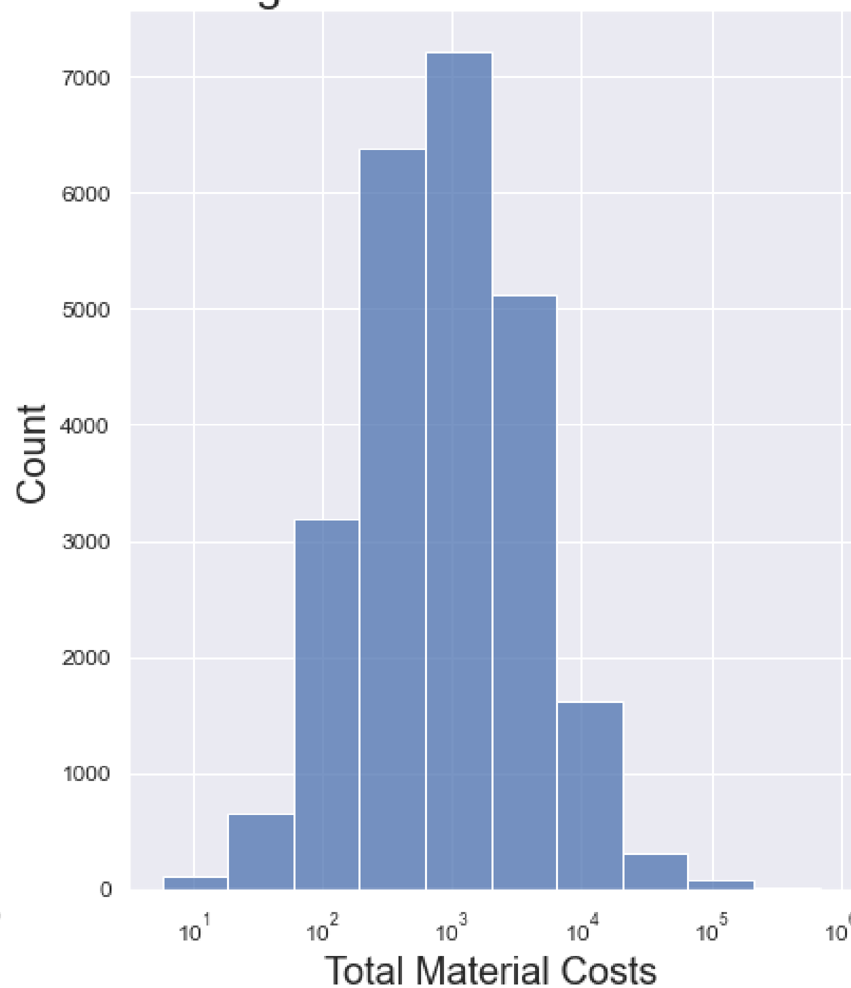




Total Material Cost Dist.

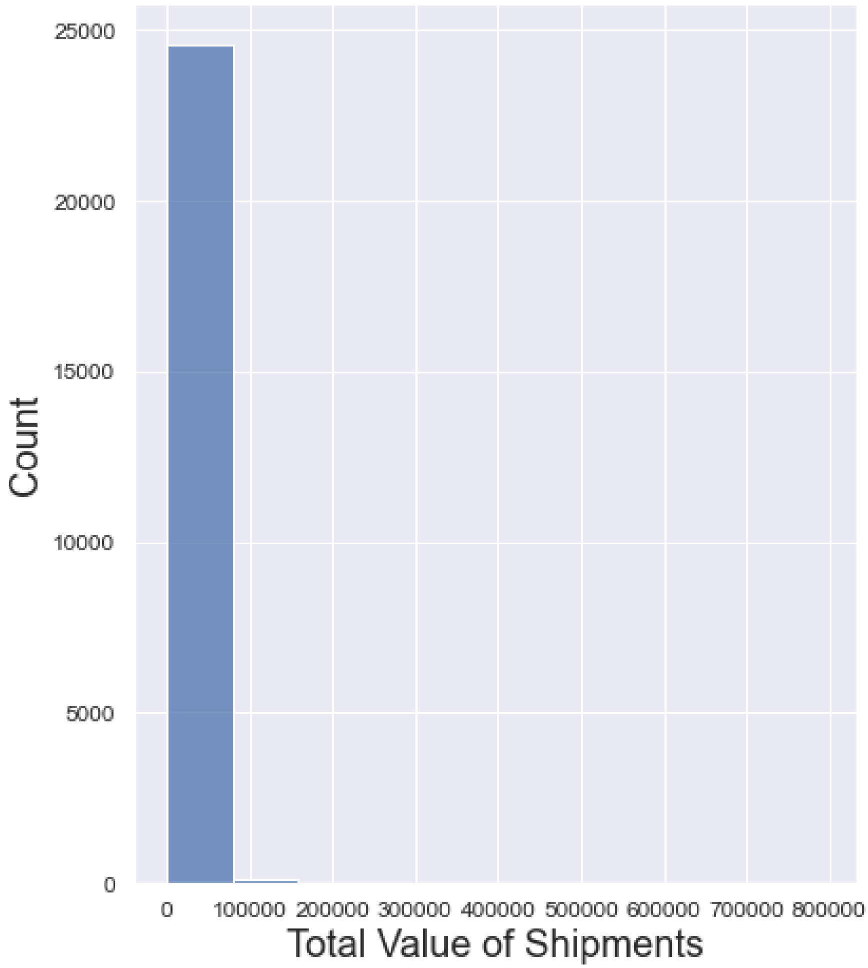


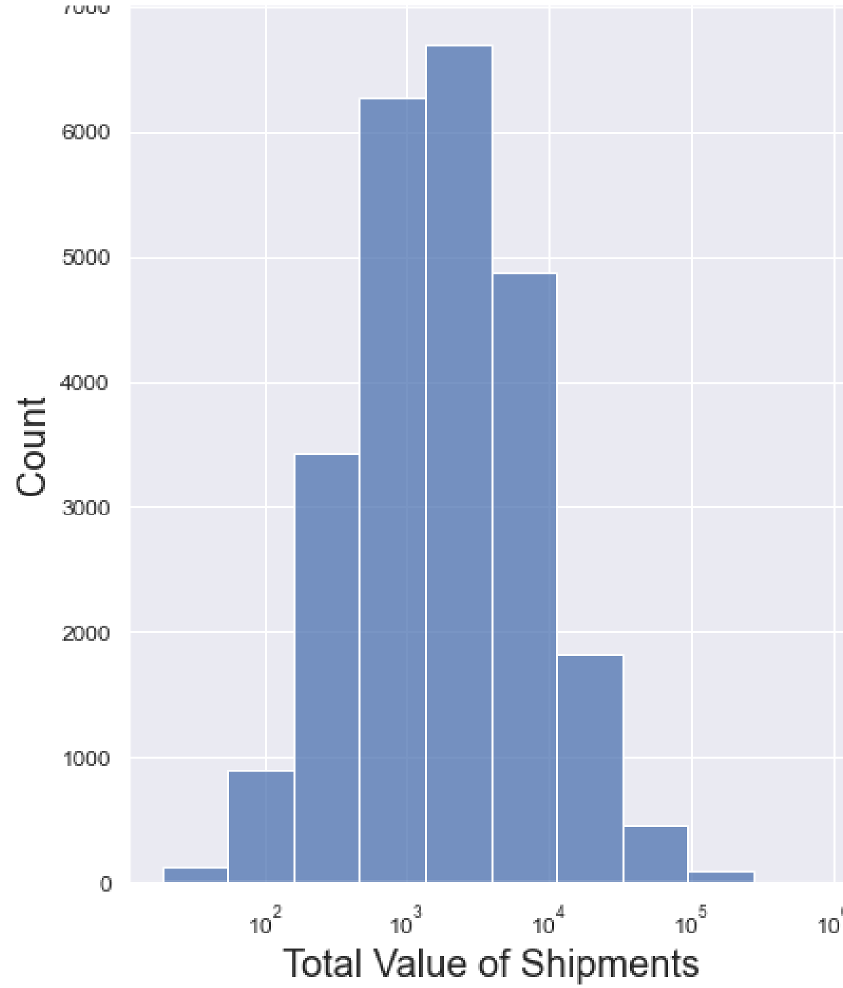
Log Total Material Cost Dist.



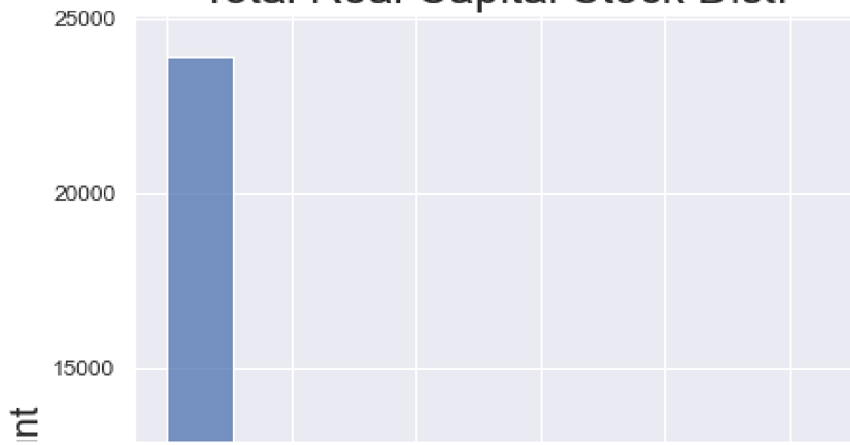
Total Value of Shipments Dist.

Total Value of Shipments Dist.

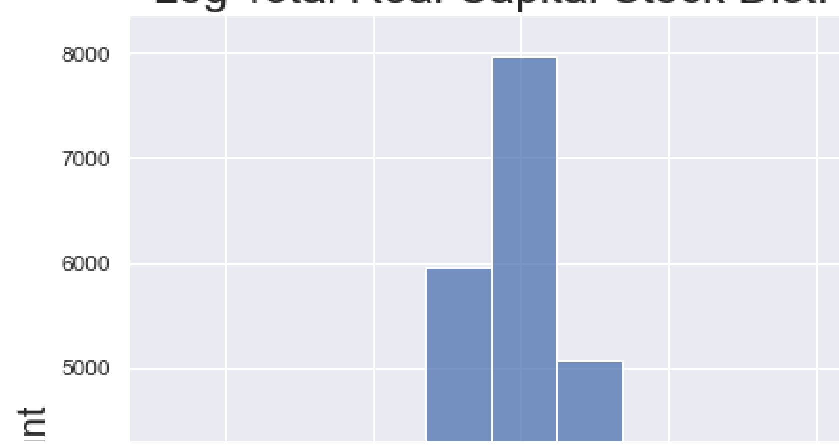


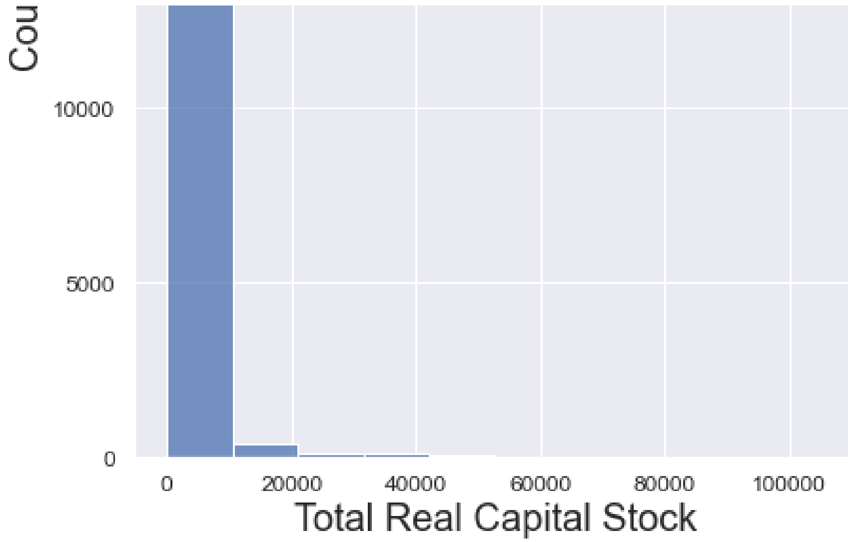


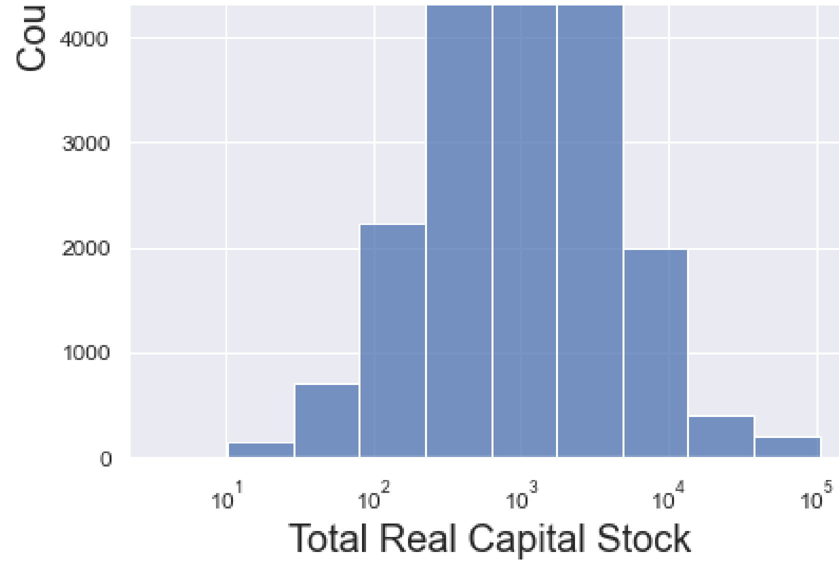
Total Real Capital Stock Dist.



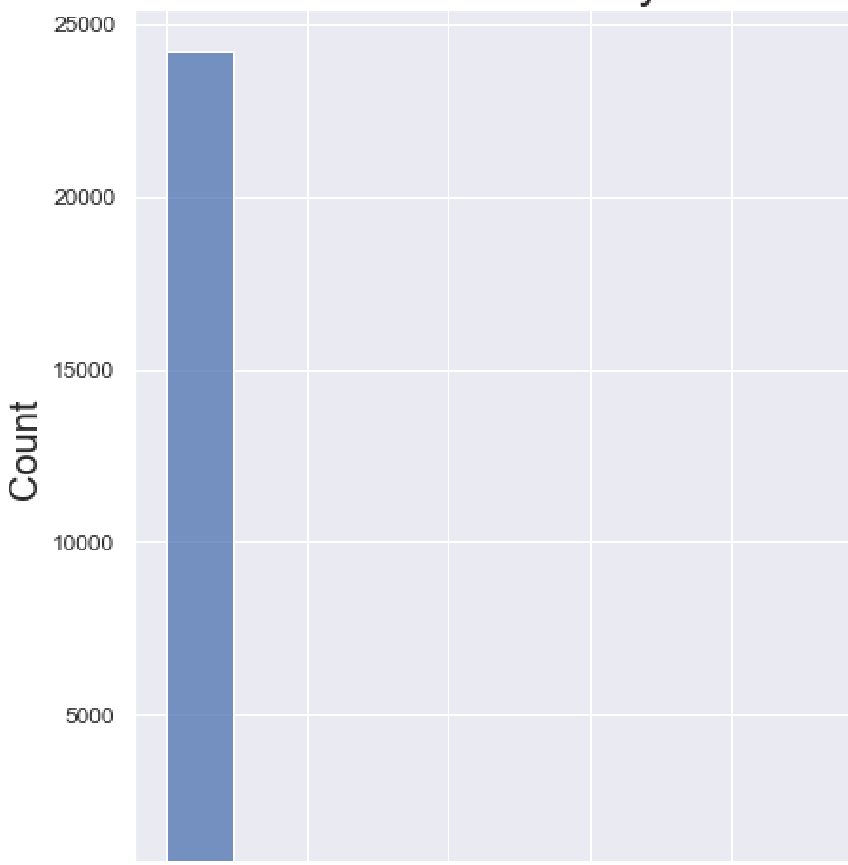
Log Total Real Capital Stock Dist.



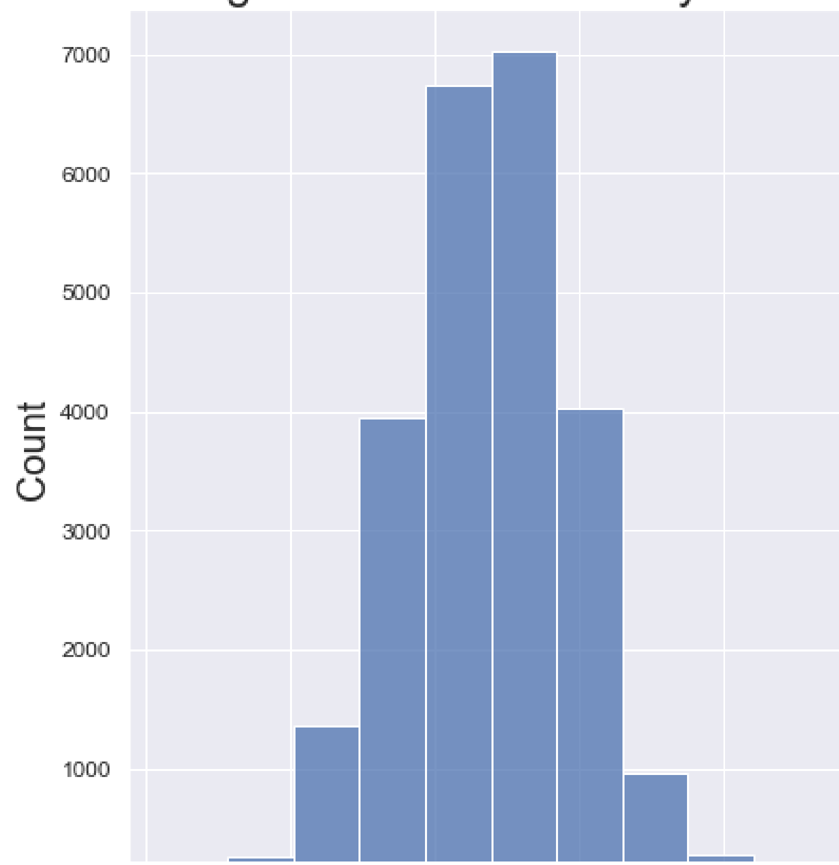


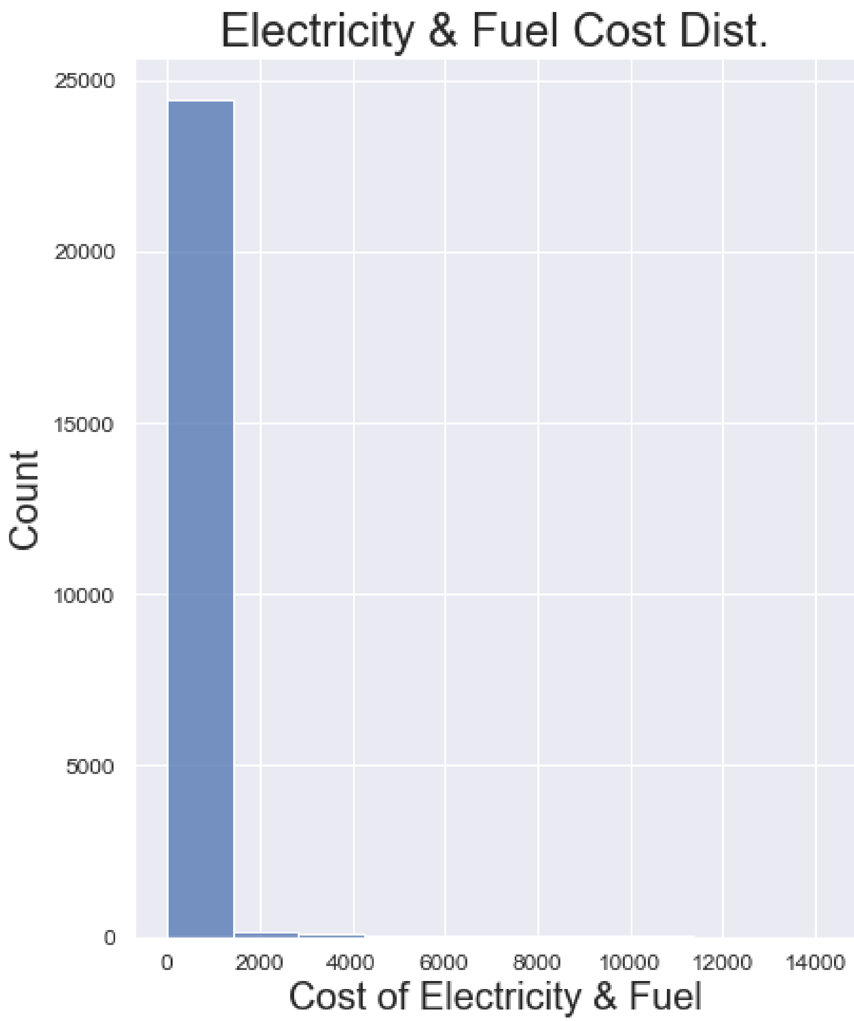


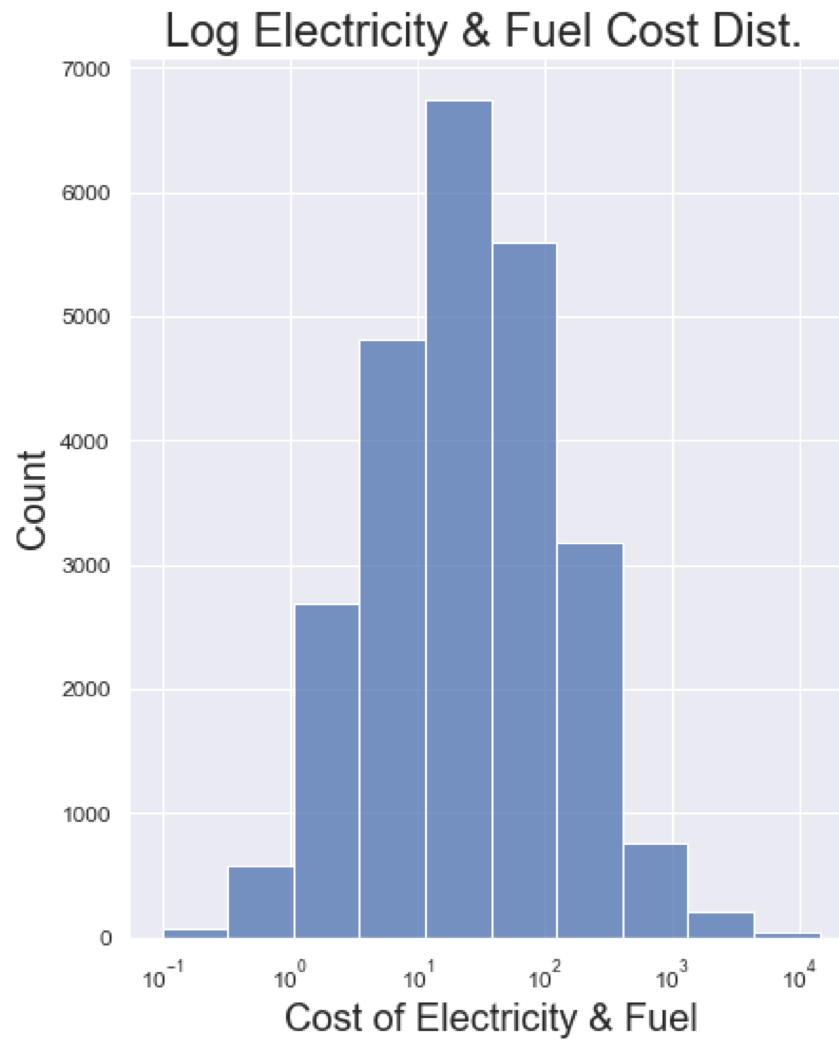
End of Year Inventory Dist.



Log End of Year Inventory Dist.







In [6]:

```
#all of the features required log transformations based off of the histograms

sic['l_emp'] = np.log(sic['emp']) # creating new column with log transformation
sic = sic.drop('emp', 1) # dropping old column

sic['l_invest'] = np.log(sic['invest']) # creating new column with log transformation
sic = sic.drop('invest', 1) # dropping old column
```

```
sic['l_pay'] = np.log(sic['pay']) # creating new column with log transformation
sic = sic.drop('pay', 1) # dropping old column

sic['l_matcost'] = np.log(sic['matcost']) # creating new column with log transformation
sic = sic.drop('matcost', 1) # dropping old column

sic['l_vship'] = np.log(sic['vship']) # creating new column with log transformation
sic = sic.drop('vship', 1) # dropping old column

sic['l_cap'] = np.log(sic['cap']) # creating new column with log transformation
sic = sic.drop('cap', 1) # dropping old column

sic['l_invent'] = np.log(sic['invent']) # creating new column with log transformation
sic = sic.drop('invent', 1) # dropping old column

sic['l_energy'] = np.log(sic['energy']) # creating new column with log transformation
sic = sic.drop('energy', 1) # dropping old column
```

In [7]:

```
sic.head()
```

Out[7]:

		l_vadd	l_emp	l_invest	l_pay	l_matcost	l_vship	l_cap	l_invent	l_energy
sic	**year**									
2011	**1958**	7.466571	5.302807	4.188138	6.973356	9.233090	9.388545	8.181860	6.011512	3.869116
	1959	7.513818	5.284218	4.210645	7.003974	9.204232	9.374871	8.220887	5.913773	3.899950
	1960	7.555225	5.268889	4.346399	7.037555	9.199360	9.376380	8.264441	5.944373	3.929863
	1961	7.543909	5.243333	4.322807	7.041587	9.215059	9.385704	8.299982	5.979645	3.958907
	1962	7.593928	5.223594	4.508659	7.057123	9.259968	9.430945	8.345598	6.018836	3.987130

In [8]:

```
sic.shape
```

Out[8]:

```
(24676, 9)
```

In [9]:

```
sic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
MultiIndex: 24676 entries, (2011, 1958) to (3999, 2011)
Data columns (total 9 columns):
l_vadd      24676 non-null float64
l_emp       24676 non-null float64
l_invest    24676 non-null float64
l_pay       24676 non-null float64
l_matcost   24676 non-null float64
l_vship     24676 non-null float64
l_cap       24676 non-null float64
l_invent    24676 non-null float64
l_energy    24676 non-null float64
dtypes: float64(9)
memory usage: 1.8 MB
```

Inferences

TOP

In [10]:

```
y = sic['l_vadd']
x = sic.drop(columns = 'l_vadd')

y_train, y_test = train_test_split(y, train_size = 3/4, random_state = 490)
x_train, x_test = train_test_split(x, train_size = 3/4, random_state = 490)

ss = StandardScaler()
x_train_std = pd.DataFrame(ss.fit(x_train).transform(x_train),
                          columns = x_train.columns,
                          index = x_train.index)
x_test_std = pd.DataFrame(ss.fit(x_test).transform(x_test),
                          columns = x_test.columns,
                          index = x_test.index)

x_train_std_c     = sm.add_constant(x_train_std)
x_test_std_c      = sm.add_constant(x_test_std)
x_train_c    = sm.add_constant(x_train)
x_test_c     = sm.add_constant(x_test)
```

```
C:\Users\tanse\Anaconda3\lib\site-packages\numpy\core\fromnumeric.py:2580: FutureWarning: Method .ptp is deprecated and w
ill be removed in a future version. Use numpy.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```

In [11]:

```
param_grid = [
    {'alpha': 10**np.linspace(-7, -4, num = 10)}
]
```

```
cv_lasso = lm.Lasso(fit_intercept = False, normalize = False,
                    random_state = 490)
grid_search = GridSearchCV(cv_lasso, param_grid, cv = 5,
                         scoring = 'neg_root_mean_squared_error')
grid_search.fit(x_train_std_c, y_train)
best = grid_search.best_params_['alpha']
best
```

Out[11]: 2.1544346900318822e-06

In [12]:

```
fit_lasso_tuned = sm.OLS(y_train, x_train_std_c).fit_regularized(alpha = best)
beta = fit_lasso_tuned.params #fitting on non regularized standardized model
beta.index[beta == 0]
x_train_trim = x_train_std_c.loc[:, ~x_train_std_c.columns.isin(beta.index[beta == 0])]
x_test_trim = x_test_std_c.loc[:, ~x_test_std_c.columns.isin(beta.index[beta == 0])]
```

In [13]:

```
fit_std_final = sm.OLS(y_train, x_train_trim).fit() #testing on non-regularized values
fit_std_final.summary2()
```

Out[13]:

Model:	OLS	Adj. R-squared:	0.993
Dependent Variable:	l_vadd	AIC:	-26272.1018
Date:	2021-05-14 15:57	BIC:	-26201.6686
No. Observations:	18507	Log-Likelihood:	13145.
Df Model:	8	F-statistic:	3.253e+05
Df Residuals:	18498	Prob (F-statistic):	0.00
R-squared:	0.993	Scale:	0.014151

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
const	6.7547	0.0009	7724.5430	0.0000	6.7530	6.7564
l_emp	-0.0319	0.0019	-16.7239	0.0000	-0.0356	-0.0281
l_invest	0.0604	0.0032	19.0809	0.0000	0.0542	0.0666
l_pay	0.2437	0.0042	57.7686	0.0000	0.2354	0.2519
l_matcost	-1.2478	0.0064	-195.6550	0.0000	-1.2603	-1.2353

l_vship	2.3447	0.0089	264.7884	0.0000	2.3273	2.3620
l_cap	-0.0217	0.0025	-8.6475	0.0000	-0.0266	-0.0167
l_invent	0.0372	0.0025	14.7130	0.0000	0.0322	0.0421
l_energy	-0.0127	0.0024	-5.3406	0.0000	-0.0173	-0.0080

Omnibus:	12515.174	Durbin-Watson:	2.025
Prob(Omnibus):	0.000	Jarque-Bera (JB):	310124.657
Skew:	-2.905	Prob(JB):	0.000
Kurtosis:	22.194	Condition No.:	33

In [14]:

```
rmse_ols = np.sqrt(np.mean((y_test - fit_std_final.predict(x_test_trim))**2))
rmse_ols #metric for comparing the three models
```

Out[14]: 0.11483983674229031

Predictions

TOP

Extreme Gradient Boosting

In [15]:

```
x_train_train, x_train_test, y_train_train, y_train_test = train_test_split(x_train_std, y_train,
                                                                            train_size = 1/2,
                                                                           random_state = 490)
# setting up double train test split for gradient boosting
```

In [16]:

```
clf_xgb = xgb.XGBRegressor(n_estimators = 750, max_depth = 6, learning_rate = 0.1,
                          random_state = 490, use_label_encoder = False)
clf_xgb.fit(x_train_train, y_train_train, eval_set = [(x_train_test, y_train_test)],
           early_stopping_rounds = 5) # fitting on train train, testing on train tests
```

```
[0]	validation_0-rmse:5.77151
[1]	validation_0-rmse:5.19649
[2]	validation_0-rmse:4.67859
[3]	validation_0-rmse:4.21337
```

```
[4]	validation_0-rmse:3.79459
[5]	validation_0-rmse:3.41731
[6]	validation_0-rmse:3.07821
[7]	validation_0-rmse:2.77316
[8]	validation_0-rmse:2.49860
[9]	validation_0-rmse:2.25194
[10]	validation_0-rmse:2.02966
[11]	validation_0-rmse:1.82977
[12]	validation_0-rmse:1.65021
[13]	validation_0-rmse:1.48836
[14]	validation_0-rmse:1.34290
[15]	validation_0-rmse:1.21192
[16]	validation_0-rmse:1.09413
[17]	validation_0-rmse:0.98799
[18]	validation_0-rmse:0.89279
[19]	validation_0-rmse:0.80693
[20]	validation_0-rmse:0.72975
[21]	validation_0-rmse:0.66041
[22]	validation_0-rmse:0.59834
[23]	validation_0-rmse:0.54284
[24]	validation_0-rmse:0.49299
[25]	validation_0-rmse:0.44822
[26]	validation_0-rmse:0.40824
[27]	validation_0-rmse:0.37259
[28]	validation_0-rmse:0.34087
[29]	validation_0-rmse:0.31251
[30]	validation_0-rmse:0.28722
[31]	validation_0-rmse:0.26500
[32]	validation_0-rmse:0.24527
[33]	validation_0-rmse:0.22787
[34]	validation_0-rmse:0.21245
[35]	validation_0-rmse:0.19890
[36]	validation_0-rmse:0.18724
[37]	validation_0-rmse:0.17668
[38]	validation_0-rmse:0.16756
[39]	validation_0-rmse:0.15960
[40]	validation_0-rmse:0.15268
[41]	validation_0-rmse:0.14658
[42]	validation_0-rmse:0.14145
[43]	validation_0-rmse:0.13668
[44]	validation_0-rmse:0.13262
[45]	validation_0-rmse:0.12930
[46]	validation_0-rmse:0.12645
[47]	validation_0-rmse:0.12388
[48]	validation_0-rmse:0.12168
[49]	validation_0-rmse:0.11968
[50]	validation_0-rmse:0.11808
[51]	validation_0-rmse:0.11654
[52]	validation_0-rmse:0.11528
```

```
[53]	validation_0-rmse:0.11399
[54]	validation_0-rmse:0.11282
[55]	validation_0-rmse:0.11200
[56]	validation_0-rmse:0.11112
[57]	validation_0-rmse:0.11018
[58]	validation_0-rmse:0.10934
[59]	validation_0-rmse:0.10885
[60]	validation_0-rmse:0.10813
[61]	validation_0-rmse:0.10752
[62]	validation_0-rmse:0.10698
[63]	validation_0-rmse:0.10650
[64]	validation_0-rmse:0.10615
[65]	validation_0-rmse:0.10578
[66]	validation_0-rmse:0.10538
[67]	validation_0-rmse:0.10488
[68]	validation_0-rmse:0.10459
[69]	validation_0-rmse:0.10419
[70]	validation_0-rmse:0.10377
[71]	validation_0-rmse:0.10341
[72]	validation_0-rmse:0.10303
[73]	validation_0-rmse:0.10257
[74]	validation_0-rmse:0.10224
[75]	validation_0-rmse:0.10184
[76]	validation_0-rmse:0.10138
[77]	validation_0-rmse:0.10104
[78]	validation_0-rmse:0.10074
[79]	validation_0-rmse:0.10045
[80]	validation_0-rmse:0.09997
[81]	validation_0-rmse:0.09977
[82]	validation_0-rmse:0.09935
[83]	validation_0-rmse:0.09915
[84]	validation_0-rmse:0.09882
[85]	validation_0-rmse:0.09851
[86]	validation_0-rmse:0.09834
[87]	validation_0-rmse:0.09790
[88]	validation_0-rmse:0.09756
[89]	validation_0-rmse:0.09718
[90]	validation_0-rmse:0.09712
[91]	validation_0-rmse:0.09688
[92]	validation_0-rmse:0.09660
[93]	validation_0-rmse:0.09633
[94]	validation_0-rmse:0.09599
[95]	validation_0-rmse:0.09587
[96]	validation_0-rmse:0.09565
[97]	validation_0-rmse:0.09539
[98]	validation_0-rmse:0.09501
[99]	validation_0-rmse:0.09468
[100]	validation_0-rmse:0.09463
[101]	validation_0-rmse:0.09444
```

```
[102]	validation_0-rmse:0.09412
[103]	validation_0-rmse:0.09387
[104]	validation_0-rmse:0.09371
[105]	validation_0-rmse:0.09348
[106]	validation_0-rmse:0.09342
[107]	validation_0-rmse:0.09322
[108]	validation_0-rmse:0.09303
[109]	validation_0-rmse:0.09272
[110]	validation_0-rmse:0.09266
[111]	validation_0-rmse:0.09246
[112]	validation_0-rmse:0.09219
[113]	validation_0-rmse:0.09206
[114]	validation_0-rmse:0.09179
[115]	validation_0-rmse:0.09165
[116]	validation_0-rmse:0.09164
[117]	validation_0-rmse:0.09157
[118]	validation_0-rmse:0.09145
[119]	validation_0-rmse:0.09122
[120]	validation_0-rmse:0.09109
[121]	validation_0-rmse:0.09093
[122]	validation_0-rmse:0.09071
[123]	validation_0-rmse:0.09056
[124]	validation_0-rmse:0.09031
[125]	validation_0-rmse:0.09015
[126]	validation_0-rmse:0.08990
[127]	validation_0-rmse:0.08968
[128]	validation_0-rmse:0.08953
[129]	validation_0-rmse:0.08939
[130]	validation_0-rmse:0.08914
[131]	validation_0-rmse:0.08907
[132]	validation_0-rmse:0.08890
[133]	validation_0-rmse:0.08874
[134]	validation_0-rmse:0.08854
[135]	validation_0-rmse:0.08850
[136]	validation_0-rmse:0.08846
[137]	validation_0-rmse:0.08838
[138]	validation_0-rmse:0.08827
[139]	validation_0-rmse:0.08810
[140]	validation_0-rmse:0.08804
[141]	validation_0-rmse:0.08796
[142]	validation_0-rmse:0.08785
[143]	validation_0-rmse:0.08772
[144]	validation_0-rmse:0.08758
[145]	validation_0-rmse:0.08753
[146]	validation_0-rmse:0.08745
[147]	validation_0-rmse:0.08731
[148]	validation_0-rmse:0.08716
[149]	validation_0-rmse:0.08701
[150]	validation_0-rmse:0.08687
```

```
[151]	validation_0-rmse:0.08678
[152]	validation_0-rmse:0.08675
[153]	validation_0-rmse:0.08665
[154]	validation_0-rmse:0.08656
[155]	validation_0-rmse:0.08643
[156]	validation_0-rmse:0.08632
[157]	validation_0-rmse:0.08625
[158]	validation_0-rmse:0.08610
[159]	validation_0-rmse:0.08598
[160]	validation_0-rmse:0.08590
[161]	validation_0-rmse:0.08584
[162]	validation_0-rmse:0.08576
[163]	validation_0-rmse:0.08563
[164]	validation_0-rmse:0.08554
[165]	validation_0-rmse:0.08539
[166]	validation_0-rmse:0.08534
[167]	validation_0-rmse:0.08530
[168]	validation_0-rmse:0.08523
[169]	validation_0-rmse:0.08508
[170]	validation_0-rmse:0.08498
[171]	validation_0-rmse:0.08483
[172]	validation_0-rmse:0.08474
[173]	validation_0-rmse:0.08468
[174]	validation_0-rmse:0.08467
[175]	validation_0-rmse:0.08459
[176]	validation_0-rmse:0.08447
[177]	validation_0-rmse:0.08438
[178]	validation_0-rmse:0.08432
[179]	validation_0-rmse:0.08424
[180]	validation_0-rmse:0.08418
[181]	validation_0-rmse:0.08413
[182]	validation_0-rmse:0.08407
[183]	validation_0-rmse:0.08399
[184]	validation_0-rmse:0.08399
[185]	validation_0-rmse:0.08391
[186]	validation_0-rmse:0.08380
[187]	validation_0-rmse:0.08372
[188]	validation_0-rmse:0.08362
[189]	validation_0-rmse:0.08358
[190]	validation_0-rmse:0.08354
[191]	validation_0-rmse:0.08352
[192]	validation_0-rmse:0.08342
[193]	validation_0-rmse:0.08329
[194]	validation_0-rmse:0.08324
[195]	validation_0-rmse:0.08318
[196]	validation_0-rmse:0.08311
[197]	validation_0-rmse:0.08301
[198]	validation_0-rmse:0.08295
[199]	validation_0-rmse:0.08292
```

```
[200]	validation_0-rmse:0.08280
[201]	validation_0-rmse:0.08272
[202]	validation_0-rmse:0.08262
[203]	validation_0-rmse:0.08252
[204]	validation_0-rmse:0.08246
[205]	validation_0-rmse:0.08244
[206]	validation_0-rmse:0.08235
[207]	validation_0-rmse:0.08231
[208]	validation_0-rmse:0.08229
[209]	validation_0-rmse:0.08223
[210]	validation_0-rmse:0.08217
[211]	validation_0-rmse:0.08213
[212]	validation_0-rmse:0.08206
[213]	validation_0-rmse:0.08199
[214]	validation_0-rmse:0.08191
[215]	validation_0-rmse:0.08184
[216]	validation_0-rmse:0.08183
[217]	validation_0-rmse:0.08175
[218]	validation_0-rmse:0.08175
[219]	validation_0-rmse:0.08172
[220]	validation_0-rmse:0.08167
[221]	validation_0-rmse:0.08160
[222]	validation_0-rmse:0.08155
[223]	validation_0-rmse:0.08156
[224]	validation_0-rmse:0.08151
[225]	validation_0-rmse:0.08144
[226]	validation_0-rmse:0.08138
[227]	validation_0-rmse:0.08135
[228]	validation_0-rmse:0.08129
[229]	validation_0-rmse:0.08126
[230]	validation_0-rmse:0.08122
[231]	validation_0-rmse:0.08117
[232]	validation_0-rmse:0.08110
[233]	validation_0-rmse:0.08105
[234]	validation_0-rmse:0.08102
[235]	validation_0-rmse:0.08099
[236]	validation_0-rmse:0.08091
[237]	validation_0-rmse:0.08085
[238]	validation_0-rmse:0.08079
[239]	validation_0-rmse:0.08073
[240]	validation_0-rmse:0.08073
[241]	validation_0-rmse:0.08063
[242]	validation_0-rmse:0.08061
[243]	validation_0-rmse:0.08054
[244]	validation_0-rmse:0.08052
[245]	validation_0-rmse:0.08046
[246]	validation_0-rmse:0.08046
[247]	validation_0-rmse:0.08046
[248]	validation_0-rmse:0.08041
```

```
[249]	validation_0-rmse:0.08038
[250]	validation_0-rmse:0.08037
[251]	validation_0-rmse:0.08034
[252]	validation_0-rmse:0.08030
[253]	validation_0-rmse:0.08021
[254]	validation_0-rmse:0.08019
[255]	validation_0-rmse:0.08015
[256]	validation_0-rmse:0.08011
[257]	validation_0-rmse:0.08005
[258]	validation_0-rmse:0.08001
[259]	validation_0-rmse:0.07998
[260]	validation_0-rmse:0.07992
[261]	validation_0-rmse:0.07987
[262]	validation_0-rmse:0.07987
[263]	validation_0-rmse:0.07984
[264]	validation_0-rmse:0.07982
[265]	validation_0-rmse:0.07978
[266]	validation_0-rmse:0.07974
[267]	validation_0-rmse:0.07971
[268]	validation_0-rmse:0.07968
[269]	validation_0-rmse:0.07966
[270]	validation_0-rmse:0.07962
[271]	validation_0-rmse:0.07958
[272]	validation_0-rmse:0.07952
[273]	validation_0-rmse:0.07951
[274]	validation_0-rmse:0.07951
[275]	validation_0-rmse:0.07946
[276]	validation_0-rmse:0.07944
[277]	validation_0-rmse:0.07941
[278]	validation_0-rmse:0.07940
[279]	validation_0-rmse:0.07937
[280]	validation_0-rmse:0.07934
[281]	validation_0-rmse:0.07932
[282]	validation_0-rmse:0.07927
[283]	validation_0-rmse:0.07924
[284]	validation_0-rmse:0.07924
[285]	validation_0-rmse:0.07923
[286]	validation_0-rmse:0.07918
[287]	validation_0-rmse:0.07914
[288]	validation_0-rmse:0.07911
[289]	validation_0-rmse:0.07911
[290]	validation_0-rmse:0.07905
[291]	validation_0-rmse:0.07902
[292]	validation_0-rmse:0.07898
[293]	validation_0-rmse:0.07894
[294]	validation_0-rmse:0.07894
[295]	validation_0-rmse:0.07890
[296]	validation_0-rmse:0.07887
[297]	validation_0-rmse:0.07886
```

```
[298]	validation_0-rmse:0.07882
[299]	validation_0-rmse:0.07881
[300]	validation_0-rmse:0.07881
[301]	validation_0-rmse:0.07878
[302]	validation_0-rmse:0.07876
[303]	validation_0-rmse:0.07875
[304]	validation_0-rmse:0.07873
[305]	validation_0-rmse:0.07869
[306]	validation_0-rmse:0.07867
[307]	validation_0-rmse:0.07865
[308]	validation_0-rmse:0.07861
[309]	validation_0-rmse:0.07858
[310]	validation_0-rmse:0.07858
[311]	validation_0-rmse:0.07854
[312]	validation_0-rmse:0.07850
[313]	validation_0-rmse:0.07849
[314]	validation_0-rmse:0.07848
[315]	validation_0-rmse:0.07845
[316]	validation_0-rmse:0.07844
[317]	validation_0-rmse:0.07842
[318]	validation_0-rmse:0.07841
[319]	validation_0-rmse:0.07838
[320]	validation_0-rmse:0.07836
[321]	validation_0-rmse:0.07832
[322]	validation_0-rmse:0.07827
[323]	validation_0-rmse:0.07823
[324]	validation_0-rmse:0.07820
[325]	validation_0-rmse:0.07818
[326]	validation_0-rmse:0.07816
[327]	validation_0-rmse:0.07813
[328]	validation_0-rmse:0.07812
[329]	validation_0-rmse:0.07813
[330]	validation_0-rmse:0.07811
[331]	validation_0-rmse:0.07810
[332]	validation_0-rmse:0.07807
[333]	validation_0-rmse:0.07805
[334]	validation_0-rmse:0.07805
[335]	validation_0-rmse:0.07802
[336]	validation_0-rmse:0.07800
[337]	validation_0-rmse:0.07800
[338]	validation_0-rmse:0.07797
[339]	validation_0-rmse:0.07795
[340]	validation_0-rmse:0.07793
[341]	validation_0-rmse:0.07792
[342]	validation_0-rmse:0.07790
[343]	validation_0-rmse:0.07788
[344]	validation_0-rmse:0.07786
[345]	validation_0-rmse:0.07786
[346]	validation_0-rmse:0.07784
```

```
[347]	validation_0-rmse:0.07783
[348]	validation_0-rmse:0.07781
[349]	validation_0-rmse:0.07777
[350]	validation_0-rmse:0.07776
[351]	validation_0-rmse:0.07772
[352]	validation_0-rmse:0.07771
[353]	validation_0-rmse:0.07770
[354]	validation_0-rmse:0.07769
[355]	validation_0-rmse:0.07767
[356]	validation_0-rmse:0.07767
[357]	validation_0-rmse:0.07767
[358]	validation_0-rmse:0.07764
[359]	validation_0-rmse:0.07764
[360]	validation_0-rmse:0.07763
[361]	validation_0-rmse:0.07762
[362]	validation_0-rmse:0.07758
[363]	validation_0-rmse:0.07757
[364]	validation_0-rmse:0.07756
[365]	validation_0-rmse:0.07752
[366]	validation_0-rmse:0.07752
[367]	validation_0-rmse:0.07746
[368]	validation_0-rmse:0.07743
[369]	validation_0-rmse:0.07743
[370]	validation_0-rmse:0.07741
[371]	validation_0-rmse:0.07741
[372]	validation_0-rmse:0.07739
[373]	validation_0-rmse:0.07738
[374]	validation_0-rmse:0.07737
[375]	validation_0-rmse:0.07736
[376]	validation_0-rmse:0.07736
[377]	validation_0-rmse:0.07735
[378]	validation_0-rmse:0.07735
[379]	validation_0-rmse:0.07734
[380]	validation_0-rmse:0.07733
[381]	validation_0-rmse:0.07733
[382]	validation_0-rmse:0.07734
[383]	validation_0-rmse:0.07734
[384]	validation_0-rmse:0.07732
[385]	validation_0-rmse:0.07730
[386]	validation_0-rmse:0.07728
[387]	validation_0-rmse:0.07726
[388]	validation_0-rmse:0.07724
[389]	validation_0-rmse:0.07723
[390]	validation_0-rmse:0.07722
[391]	validation_0-rmse:0.07720
[392]	validation_0-rmse:0.07718
[393]	validation_0-rmse:0.07718
[394]	validation_0-rmse:0.07714
[395]	validation_0-rmse:0.07712
```

```
[396]	validation_0-rmse:0.07712
[397]	validation_0-rmse:0.07709
[398]	validation_0-rmse:0.07707
[399]	validation_0-rmse:0.07705
[400]	validation_0-rmse:0.07705
[401]	validation_0-rmse:0.07705
[402]	validation_0-rmse:0.07703
[403]	validation_0-rmse:0.07702
[404]	validation_0-rmse:0.07700
[405]	validation_0-rmse:0.07698
[406]	validation_0-rmse:0.07697
[407]	validation_0-rmse:0.07695
[408]	validation_0-rmse:0.07693
[409]	validation_0-rmse:0.07693
[410]	validation_0-rmse:0.07691
[411]	validation_0-rmse:0.07690
[412]	validation_0-rmse:0.07687
[413]	validation_0-rmse:0.07687
[414]	validation_0-rmse:0.07687
[415]	validation_0-rmse:0.07687
[416]	validation_0-rmse:0.07686
[417]	validation_0-rmse:0.07686
[418]	validation_0-rmse:0.07685
[419]	validation_0-rmse:0.07684
[420]	validation_0-rmse:0.07681
[421]	validation_0-rmse:0.07680
[422]	validation_0-rmse:0.07679
[423]	validation_0-rmse:0.07679
[424]	validation_0-rmse:0.07677
[425]	validation_0-rmse:0.07677
[426]	validation_0-rmse:0.07676
[427]	validation_0-rmse:0.07676
[428]	validation_0-rmse:0.07673
[429]	validation_0-rmse:0.07673
[430]	validation_0-rmse:0.07672
[431]	validation_0-rmse:0.07671
[432]	validation_0-rmse:0.07670
[433]	validation_0-rmse:0.07669
[434]	validation_0-rmse:0.07668
[435]	validation_0-rmse:0.07667
[436]	validation_0-rmse:0.07666
[437]	validation_0-rmse:0.07666
[438]	validation_0-rmse:0.07666
[439]	validation_0-rmse:0.07664
[440]	validation_0-rmse:0.07662
[441]	validation_0-rmse:0.07660
[442]	validation_0-rmse:0.07660
[443]	validation_0-rmse:0.07659
[444]	validation_0-rmse:0.07658
```

```
[445]	validation_0-rmse:0.07657
[446]	validation_0-rmse:0.07657
[447]	validation_0-rmse:0.07656
[448]	validation_0-rmse:0.07655
[449]	validation_0-rmse:0.07654
[450]	validation_0-rmse:0.07654
[451]	validation_0-rmse:0.07652
[452]	validation_0-rmse:0.07651
[453]	validation_0-rmse:0.07650
[454]	validation_0-rmse:0.07650
[455]	validation_0-rmse:0.07649
[456]	validation_0-rmse:0.07647
[457]	validation_0-rmse:0.07646
[458]	validation_0-rmse:0.07646
[459]	validation_0-rmse:0.07645
[460]	validation_0-rmse:0.07643
[461]	validation_0-rmse:0.07643
[462]	validation_0-rmse:0.07641
[463]	validation_0-rmse:0.07639
[464]	validation_0-rmse:0.07639
[465]	validation_0-rmse:0.07639
[466]	validation_0-rmse:0.07638
[467]	validation_0-rmse:0.07638
[468]	validation_0-rmse:0.07638
[469]	validation_0-rmse:0.07636
[470]	validation_0-rmse:0.07636
[471]	validation_0-rmse:0.07635
[472]	validation_0-rmse:0.07635
[473]	validation_0-rmse:0.07635
[474]	validation_0-rmse:0.07634
[475]	validation_0-rmse:0.07634
[476]	validation_0-rmse:0.07634
[477]	validation_0-rmse:0.07633
[478]	validation_0-rmse:0.07633
[479]	validation_0-rmse:0.07632
[480]	validation_0-rmse:0.07632
[481]	validation_0-rmse:0.07631
[482]	validation_0-rmse:0.07629
[483]	validation_0-rmse:0.07628
[484]	validation_0-rmse:0.07627
[485]	validation_0-rmse:0.07626
[486]	validation_0-rmse:0.07625
[487]	validation_0-rmse:0.07624
[488]	validation_0-rmse:0.07624
[489]	validation_0-rmse:0.07623
[490]	validation_0-rmse:0.07622
[491]	validation_0-rmse:0.07620
[492]	validation_0-rmse:0.07618
[493]	validation_0-rmse:0.07617
```

```
[494]	validation_0-rmse:0.07617
[495]	validation_0-rmse:0.07615
[496]	validation_0-rmse:0.07615
[497]	validation_0-rmse:0.07614
[498]	validation_0-rmse:0.07614
[499]	validation_0-rmse:0.07613
[500]	validation_0-rmse:0.07612
[501]	validation_0-rmse:0.07611
[502]	validation_0-rmse:0.07611
[503]	validation_0-rmse:0.07610
[504]	validation_0-rmse:0.07610
[505]	validation_0-rmse:0.07609
[506]	validation_0-rmse:0.07609
[507]	validation_0-rmse:0.07608
[508]	validation_0-rmse:0.07608
[509]	validation_0-rmse:0.07607
[510]	validation_0-rmse:0.07607
[511]	validation_0-rmse:0.07607
[512]	validation_0-rmse:0.07607
[513]	validation_0-rmse:0.07606
[514]	validation_0-rmse:0.07605
[515]	validation_0-rmse:0.07605
[516]	validation_0-rmse:0.07605
[517]	validation_0-rmse:0.07604
[518]	validation_0-rmse:0.07603
[519]	validation_0-rmse:0.07603
[520]	validation_0-rmse:0.07602
[521]	validation_0-rmse:0.07601
[522]	validation_0-rmse:0.07601
[523]	validation_0-rmse:0.07601
[524]	validation_0-rmse:0.07600
[525]	validation_0-rmse:0.07599
[526]	validation_0-rmse:0.07599
[527]	validation_0-rmse:0.07599
[528]	validation_0-rmse:0.07599
[529]	validation_0-rmse:0.07598
[530]	validation_0-rmse:0.07597
[531]	validation_0-rmse:0.07597
[532]	validation_0-rmse:0.07597
[533]	validation_0-rmse:0.07597
[534]	validation_0-rmse:0.07596
[535]	validation_0-rmse:0.07595
[536]	validation_0-rmse:0.07594
[537]	validation_0-rmse:0.07594
[538]	validation_0-rmse:0.07593
[539]	validation_0-rmse:0.07592
[540]	validation_0-rmse:0.07592
[541]	validation_0-rmse:0.07592
[542]	validation_0-rmse:0.07590
```

```
[543]	validation_0-rmse:0.07590
[544]	validation_0-rmse:0.07589
[545]	validation_0-rmse:0.07588
[546]	validation_0-rmse:0.07588
[547]	validation_0-rmse:0.07587
[548]	validation_0-rmse:0.07587
[549]	validation_0-rmse:0.07587
[550]	validation_0-rmse:0.07587
[551]	validation_0-rmse:0.07586
[552]	validation_0-rmse:0.07586
[553]	validation_0-rmse:0.07584
[554]	validation_0-rmse:0.07584
[555]	validation_0-rmse:0.07584
[556]	validation_0-rmse:0.07584
[557]	validation_0-rmse:0.07584
[558]	validation_0-rmse:0.07583
[559]	validation_0-rmse:0.07583
[560]	validation_0-rmse:0.07583
[561]	validation_0-rmse:0.07583
[562]	validation_0-rmse:0.07583
[563]	validation_0-rmse:0.07583
[564]	validation_0-rmse:0.07583
[565]	validation_0-rmse:0.07582
[566]	validation_0-rmse:0.07582
[567]	validation_0-rmse:0.07582
[568]	validation_0-rmse:0.07581
[569]	validation_0-rmse:0.07580
[570]	validation_0-rmse:0.07580
[571]	validation_0-rmse:0.07579
[572]	validation_0-rmse:0.07579
[573]	validation_0-rmse:0.07579
[574]	validation_0-rmse:0.07579
[575]	validation_0-rmse:0.07578
[576]	validation_0-rmse:0.07577
[577]	validation_0-rmse:0.07576
[578]	validation_0-rmse:0.07576
[579]	validation_0-rmse:0.07576
[580]	validation_0-rmse:0.07575
[581]	validation_0-rmse:0.07575
[582]	validation_0-rmse:0.07575
[583]	validation_0-rmse:0.07575
[584]	validation_0-rmse:0.07574
[585]	validation_0-rmse:0.07574
[586]	validation_0-rmse:0.07573
[587]	validation_0-rmse:0.07573
[588]	validation_0-rmse:0.07573
[589]	validation_0-rmse:0.07572
[590]	validation_0-rmse:0.07571
[591]	validation_0-rmse:0.07570
```

```
[592]	validation_0-rmse:0.07570
[593]	validation_0-rmse:0.07570
[594]	validation_0-rmse:0.07569
[595]	validation_0-rmse:0.07569
[596]	validation_0-rmse:0.07569
[597]	validation_0-rmse:0.07569
[598]	validation_0-rmse:0.07569
[599]	validation_0-rmse:0.07569
[600]	validation_0-rmse:0.07568
[601]	validation_0-rmse:0.07567
[602]	validation_0-rmse:0.07566
[603]	validation_0-rmse:0.07566
[604]	validation_0-rmse:0.07566
[605]	validation_0-rmse:0.07566
[606]	validation_0-rmse:0.07566
[607]	validation_0-rmse:0.07566
[608]	validation_0-rmse:0.07565
[609]	validation_0-rmse:0.07566
[610]	validation_0-rmse:0.07565
[611]	validation_0-rmse:0.07565
[612]	validation_0-rmse:0.07565
[613]	validation_0-rmse:0.07565
[614]	validation_0-rmse:0.07565
[615]	validation_0-rmse:0.07564
[616]	validation_0-rmse:0.07564
[617]	validation_0-rmse:0.07564
[618]	validation_0-rmse:0.07563
[619]	validation_0-rmse:0.07563
[620]	validation_0-rmse:0.07563
[621]	validation_0-rmse:0.07563
[622]	validation_0-rmse:0.07562
[623]	validation_0-rmse:0.07562
[624]	validation_0-rmse:0.07562
[625]	validation_0-rmse:0.07561
[626]	validation_0-rmse:0.07560
[627]	validation_0-rmse:0.07561
[628]	validation_0-rmse:0.07561
[629]	validation_0-rmse:0.07560
[630]	validation_0-rmse:0.07560
[631]	validation_0-rmse:0.07559
[632]	validation_0-rmse:0.07559
[633]	validation_0-rmse:0.07559
[634]	validation_0-rmse:0.07559
[635]	validation_0-rmse:0.07558
[636]	validation_0-rmse:0.07558
[637]	validation_0-rmse:0.07558
[638]	validation_0-rmse:0.07557
[639]	validation_0-rmse:0.07557
[640]	validation_0-rmse:0.07558
```

```
[641]	validation_0-rmse:0.07557
[642]	validation_0-rmse:0.07557
[643]	validation_0-rmse:0.07557
[644]	validation_0-rmse:0.07557
[645]	validation_0-rmse:0.07557
[646]	validation_0-rmse:0.07555
[647]	validation_0-rmse:0.07555
[648]	validation_0-rmse:0.07555
[649]	validation_0-rmse:0.07555
[650]	validation_0-rmse:0.07554
[651]	validation_0-rmse:0.07554
[652]	validation_0-rmse:0.07553
[653]	validation_0-rmse:0.07553
[654]	validation_0-rmse:0.07553
[655]	validation_0-rmse:0.07552
[656]	validation_0-rmse:0.07552
[657]	validation_0-rmse:0.07552
[658]	validation_0-rmse:0.07550
[659]	validation_0-rmse:0.07550
[660]	validation_0-rmse:0.07549
[661]	validation_0-rmse:0.07548
[662]	validation_0-rmse:0.07548
[663]	validation_0-rmse:0.07547
[664]	validation_0-rmse:0.07546
[665]	validation_0-rmse:0.07546
[666]	validation_0-rmse:0.07546
[667]	validation_0-rmse:0.07546
[668]	validation_0-rmse:0.07546
[669]	validation_0-rmse:0.07546
```

Out[16]:

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
             importance_type='gain', interaction_constraints='',
             learning_rate=0.1, max_delta_step=0, max_depth=6,
             min_child_weight=1, missing=nan, monotone_constraints='()',
             n_estimators=750, n_jobs=12, num_parallel_tree=1, random_state=490,
             reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
             tree_method='exact', use_label_encoder=False,
             validate_parameters=1, verbosity=None)
```

In [17]:

```
xgb_n_est = clf_xgb.best_iteration #storing best iteration
xgb_n_est
```

Out[17]:

```
664
```

In [18]:

```
clf_xgb = xgb.XGBRegressor(n_estimators = xgb_n_est, max_depth = 5,
                           learning_rate = 0.1, random_state = 490,
```

```
                          use_label_encoder = False)
clf_xgb.fit(x_train_std, y_train) #training on full with best iteration
```

Out[18]:
```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
             importance_type='gain', interaction_constraints='',
             learning_rate=0.1, max_delta_step=0, max_depth=5,
             min_child_weight=1, missing=nan, monotone_constraints='()',
             n_estimators=664, n_jobs=12, num_parallel_tree=1, random_state=490,
             reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
             tree_method='exact', use_label_encoder=False,
             validate_parameters=1, verbosity=None)
```

In [19]:
```
rmse_xgb = np.sqrt(np.mean((y_test - clf_xgb.predict(x_test_std))**2))
rmse_xgb  #metric for comparing the three models
```

Out[19]:
```
0.06475533035641769
```

Random Forest

In [20]:
```
clf_rf = RandomForestRegressor(n_estimators = 1000,
                          random_state = 490,
                          max_features = 'sqrt',
                          oob_score = True)
clf_rf.fit(x_train_std, y_train) # creating 1000 independent decision trees and averaging
```

Out[20]:
```
RandomForestRegressor(max_features='sqrt', n_estimators=1000, oob_score=True,
                      random_state=490)
```

In [24]:
```
rmse_rf = np.sqrt(np.mean((y_test - clf_rf.predict(x_test_std))**2))
rmse_rf #metric for comparing the three models
```

Out[24]:
```
0.10929388013954283
```