data_analysis

February 21, 2019

```
In [37]: cd
```

```
C:\Users\an-user
```

```
In [38]: cd C:\Users\an-user\Desktop\data study\graduate-admissions
```

```
C:\Users\an-user\Desktop\data study\graduate-admissions
```

1 Basic Data Information & Correction

```
In [53]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         import statsmodels.api as sm
```

```
In [40]: df = pd.read_csv('Admission_Predict_Ver1.1.csv')
```

```
In [41]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 9 columns):
Serial No.           500 non-null int64
GRE Score            500 non-null int64
TOEFL Score          500 non-null int64
University Rating    500 non-null int64
SOP                  500 non-null float64
LOR                  500 non-null float64
CGPA                 500 non-null float64
Research             500 non-null int64
Chance of Admit      500 non-null float64
dtypes: float64(4), int64(5)
memory usage: 35.2 KB
```

1.1 Two main Problems to be fixed.

1. column names are not neat
2. data types for serial number and research are not accurate.

```
In [42]: list1 = df.columns.tolist()
         print(list1)
```

```
['Serial No.', 'GRE Score', 'TOEFL Score', 'University Rating', 'SOP', 'LOR ', 'CGPA', 'Researc
```

```
In [43]: new_columns = []
         for i in range(0, len(list1)):
             new_column_name = list1[i].replace(' ', '_').lower()
             if new_column_name[-1] == '_' or new_column_name[-1] == '.':
                 new_column_name = new_column_name[0:-1]
             else:
                 new_column_name = new_column_name
             new_columns.append(new_column_name)
```

```
In [44]: print(new_columns)
```

```
['serial_no', 'gre_score', 'toefl_score', 'university_rating', 'sop', 'lor', 'cgpa', 'research
```

```
In [45]: df.columns = new_columns
```

```
In [69]: df['serial_no'] = df['serial_no'].astype('object')
         df['research'] = df['research'].astype('int')
```

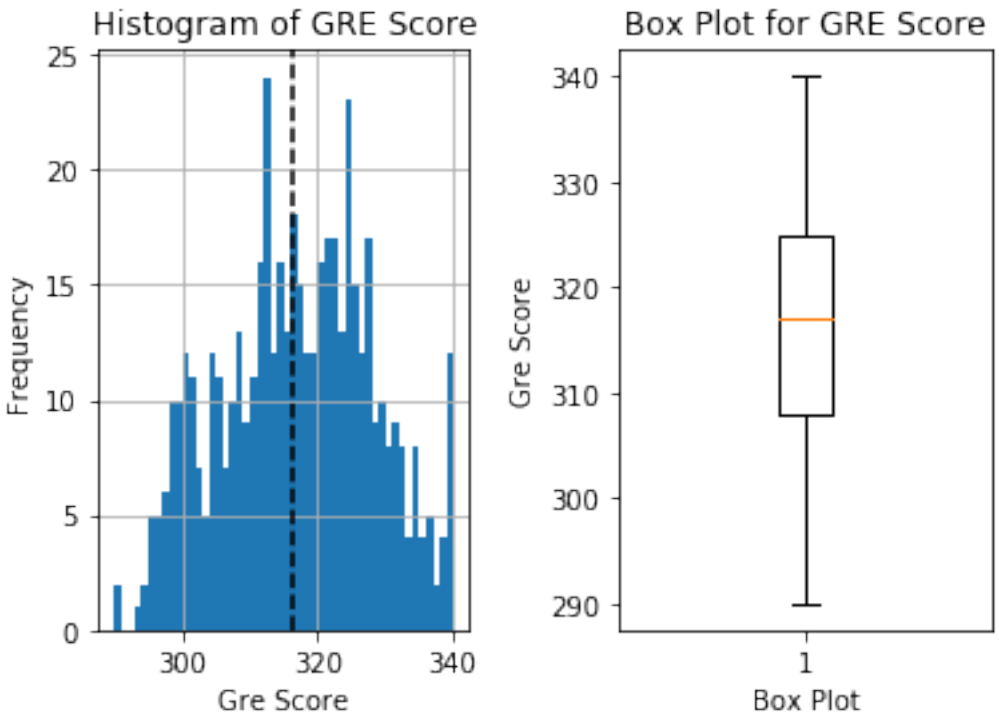
```
In [70]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 10 columns):
serial_no            500 non-null object
gre_score            500 non-null int64
toefl_score          500 non-null int64
university_rating    500 non-null int64
sop                  500 non-null float64
lor                  500 non-null float64
cgpa                 500 non-null float64
research             500 non-null int32
chance_of_admit      500 non-null float64
intercept            500 non-null int64
dtypes: float64(4), int32(1), int64(4), object(1)
memory usage: 37.2+ KB
```

```
In [33]: df.to_csv('clean_data.csv')
```

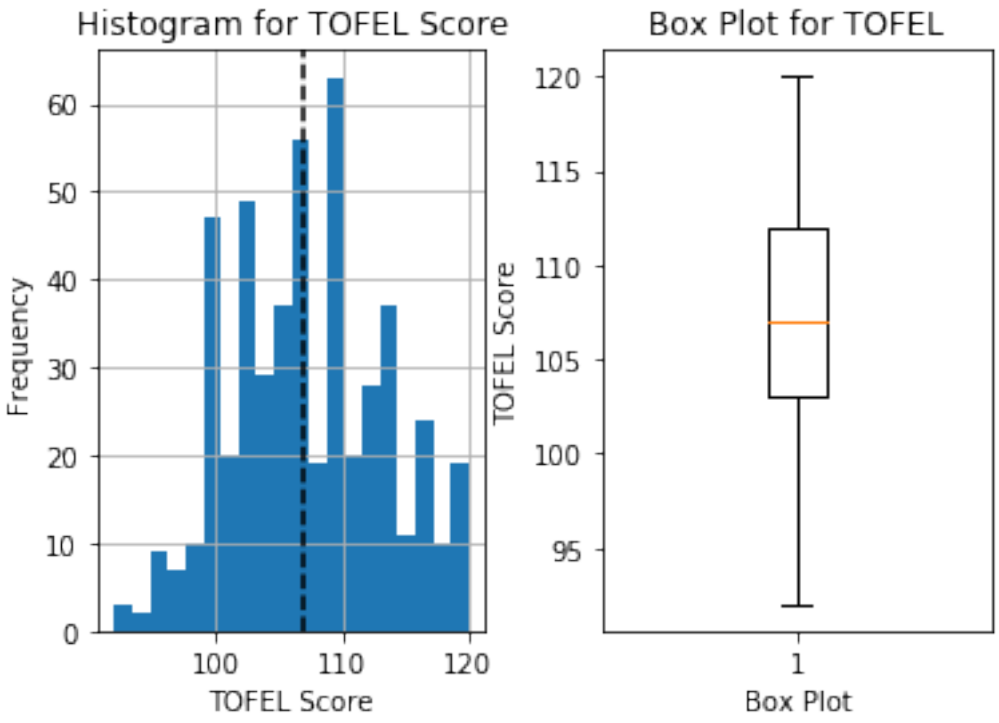
2 Univariate Analysis

In [19]:
```
##Univariate Analysis of GRE Score

plt.subplot(1,2,1)

plt.hist(x = df.gre_score, bins = 50)
plt.xlabel('Gre Score')
plt.ylabel('Frequency')
plt.title('Histogram of GRE Score')
plt.grid(True)
plt.axvline(df.gre_score.mean(), color = 'k', linestyle = 'dashed')

plt.subplot(1,2,2)
plt.boxplot(df.gre_score)
plt.xlabel('Box Plot')
plt.ylabel('Gre Score')
plt.title('Box Plot for GRE Score');

plt.subplots_adjust(wspace=0.4);
```



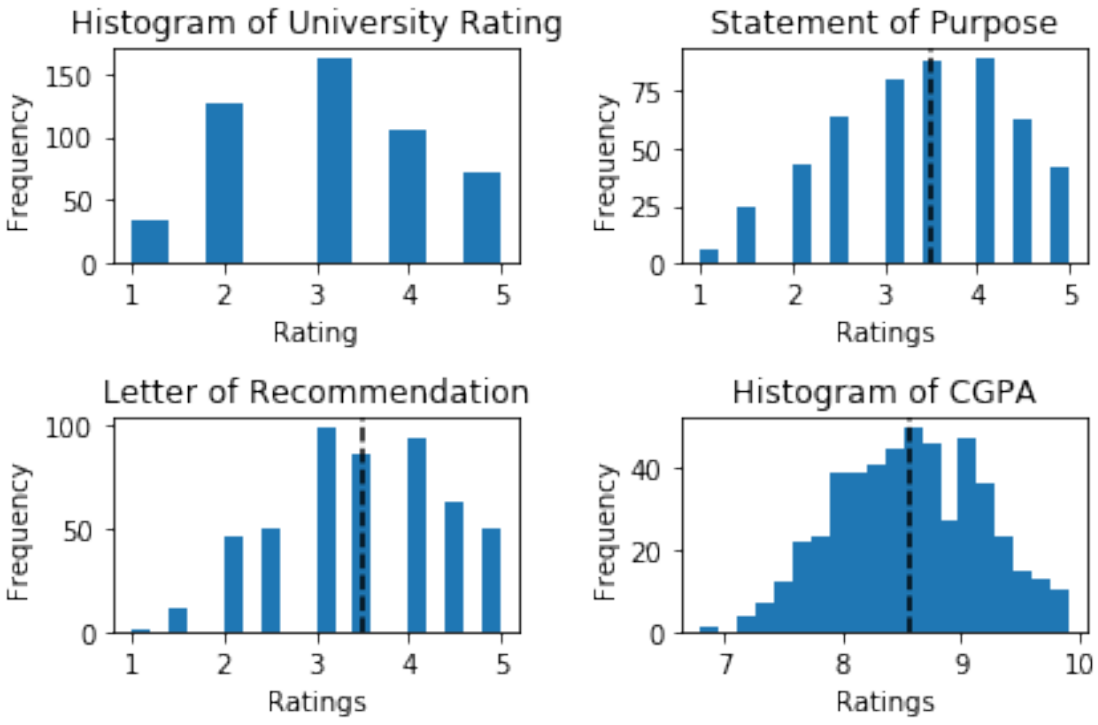
In [20]:
```
##Univariate Analysis of TOFEL Score

plt.subplot(1,2,1)
```

```
plt.hist(x = df.toefl_score, bins = 20)
plt.xlabel('TOFEL Score')
plt.ylabel('Frequency')
plt.title('Histogram for TOFEL Score')
plt.grid(True)
plt.axvline(df.toefl_score.median(), color = 'k', linestyle = 'dashed')

plt.subplot(1,2,2)
plt.boxplot(df.toefl_score)
plt.xlabel('Box Plot')
plt.ylabel('TOFEL Score')
plt.title('Box Plot for TOFEL')

plt.subplots_adjust(wspace = 0.3);
```



In [34]:
```
'university_rating', 'sop', 'lor', 'cgpa'

plt.subplot(2, 2, 1)
plt.hist(df.university_rating)
plt.title('Histogram of University Rating')
plt.xlabel('Rating')
plt.ylabel('Frequency')
```

```
plt.subplot(2, 2, 2)
plt.hist(x = df.sop, bins = 20)
plt.title('Statement of Purpose')
plt.xlabel('Ratings')
plt.ylabel('Frequency')
plt.axvline(df.sop.quantile(0.5), color = 'k', linestyle = 'dashed')


plt.subplot(2, 2, 3)
plt.hist(x = df.lor, bins = 20)
plt.title('Letter of Recommendation')
plt.xlabel('Ratings')
plt.ylabel('Frequency')
plt.axvline(df.lor.quantile(0.5), color = 'k', linestyle = 'dashed')


plt.subplot(2, 2, 4)
plt.hist(x = df.cgpa, bins = 20)
plt.title('Histogram of CGPA')
plt.xlabel('Ratings')
plt.ylabel('Frequency')
plt.axvline(df.cgpa.quantile(0.5), color = 'k', linestyle = 'dashed')


plt.tight_layout();
```



3 Multivariate Analysis

```
In [52]: corr = df.corr(method = 'pearson')

In [49]: print(corr)
```

	gre_score	toefl_score	university_rating	sop	\
gre_score	1.000000	0.827200	0.635376	0.613498	
toefl_score	0.827200	1.000000	0.649799	0.644410	
university_rating	0.635376	0.649799	1.000000	0.728024	
sop	0.613498	0.644410	0.728024	1.000000	
lor	0.524679	0.541563	0.608651	0.663707	
cgpa	0.825878	0.810574	0.705254	0.712154	
chance_of_admit	0.810351	0.792228	0.690132	0.684137	

	lor	cgpa	chance_of_admit
gre_score	0.524679	0.825878	0.810351
toefl_score	0.541563	0.810574	0.792228
university_rating	0.608651	0.705254	0.690132
sop	0.663707	0.712154	0.684137
lor	1.000000	0.637469	0.645365
cgpa	0.637469	1.000000	0.882413
chance_of_admit	0.645365	0.882413	1.000000

```
In [64]: x = df[['gre_score', 'toefl_score']]
         y = df.chance_of_admit

In [65]: model = sm.OLS(y, dfx).fit()
         model.summary()

Out[65]: <class 'statsmodels.iolib.summary.Summary'>
         """
                                    OLS Regression Results
         ==============================================================================
         Dep. Variable:        chance_of_admit   R-squared:                       0.704
         Model:                            OLS   Adj. R-squared:                  0.703
         Method:                 Least Squares   F-statistic:                     590.3
         Date:                Wed, 20 Feb 2019   Prob (F-statistic):          5.16e-132
         Time:                        19:24:26   Log-Likelihood:                574.16
         No. Observations:                 500   AIC:                            -1142.
         Df Residuals:                     497   BIC:                            -1130.
         Df Model:                           2
         Covariance Type:            nonrobust
         ==============================================================================
```

```
                  coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          -2.1803      0.102    -21.284      0.000      -2.382      -1.979
gre_score       0.0061      0.001     11.300      0.000       0.005       0.007
toefl_score     0.0090      0.001      8.886      0.000       0.007       0.011
==============================================================================
Omnibus:                       74.774   Durbin-Watson:                   0.859
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              122.617
Skew:                          -0.926   Prob(JB):                     2.37e-27
Kurtosis:                       4.568   Cond. No.                     9.95e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spec
[2] The condition number is large, 9.95e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```