Homework2-Copy1

February 19, 2024

1 Homework 2

Name/Znumber: Jordan Small / Z23465928 Professor : Juan Yepes Date : 17 FEB, 2024

Dataset: USGS EARTHQUAKES DATASET

Description: This data is related anticipating seismic tremors in order to predict where likely in the world and on what dates the earthquake will happen.

https://www.kaggle.com/datasets/rupindersinghrana/usgs-earthquakes-2024

[147]:
```
# Import necessary libraries
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt
import statsmodels.stats.outliers_influence as inf
import seaborn as sns
```

1.1 1. Setup

1.1.1 Load the dataset into a pandas Dataframe and display the first 5 rows

[267]:
```
eq = pd.read_csv('earthquakes.csv')
eq = eq.dropna()
eq.head(n=5)
```

[267]:
```
                       time   latitude   longitude    depth   mag magType  \
0  2024-01-26T04:52:42.967Z  31.604000 -104.213000   4.4198  1.70      ml
3  2024-01-26T04:29:01.180Z  38.833168 -122.797165   1.7300  0.40      md
6  2024-01-26T04:01:35.366Z  31.738000 -104.124000   4.5509  2.00      ml
7  2024-01-26T04:00:02.000Z  38.821999 -122.796997   2.0800  0.74      md
8  2024-01-26T03:55:54.430Z  19.704500  -64.647100  59.0000  3.93      md

    nst   gap      dmin   rms  …                   updated  \
0  18.0  69.0  0.100000  0.50  …  2024-01-26T05:08:27.774Z
3   9.0  65.0  0.007468  0.02  …  2024-01-26T04:46:12.828Z
6  19.0  63.0  0.000000  0.30  …  2024-01-26T04:29:32.597Z
7   7.0  90.0  0.010310  0.01  …  2024-01-26T04:29:13.730Z
```

```
8  10.0  309.0  1.387000  0.41  …  2024-01-26T04:30:44.357Z

                                   place        type horizontalError  \
0                   51 km NW of Toyah, Texas  earthquake            0.00
3                         6 km W of Cobb, CA  earthquake            0.34
6                  49 km W of Mentone, Texas  earthquake            0.00
7                 6 km NW of The Geysers, CA  earthquake            0.45
8  152 km N of Cruz Bay, U.S. Virgin Islands  earthquake            5.60

  depthError  magError  magNst     status  locationSource magSource
0   1.292059      0.10    13.0  automatic              tx        tx
3   0.970000      0.31    10.0  automatic              nc        nc
6   1.199743      0.20    14.0  automatic              tx        tx
7   1.050000      0.17     8.0  automatic              nc        nc
8  25.700000      0.13     6.0   reviewed              pr        pr

[5 rows x 22 columns]
```

1.1.2 Use .info() method to gather info about the predictors

[201]:
```
eq.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9451 entries, 0 to 9450
Data columns (total 22 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   time             9451 non-null   object
 1   latitude         9451 non-null   float64
 2   longitude        9451 non-null   float64
 3   depth            9451 non-null   float64
 4   mag              9451 non-null   float64
 5   magType          9451 non-null   object
 6   nst              7319 non-null   float64
 7   gap              7319 non-null   float64
 8   dmin             6175 non-null   float64
 9   rms              9451 non-null   float64
 10  net              9451 non-null   object
 11  id               9451 non-null   object
 12  updated          9451 non-null   object
 13  place            9451 non-null   object
 14  type             9451 non-null   object
 15  horizontalError  6462 non-null   float64
 16  depthError       9451 non-null   float64
 17  magError         7294 non-null   float64
 18  magNst           7318 non-null   float64
 19  status           9451 non-null   object
```

```
 20  locationSource   9451 non-null   object
 21  magSource        9451 non-null   object
dtypes: float64(12), object(10)
memory usage: 1.6+ MB
```

1.1.3 Construct an Evaluation Set with 5 datapoints and 3 predictors

[269]:
```python
# Evaluation set with 5 rows represented as a dictionary

data_points = [
    {'mag': 1.7, 'nst': 18, 'dmin': 0.1000, 'gap': 69, 'depth':4.419800},
    {'mag': 0.4, 'nst': 9, 'dmin': 0.007468, 'gap': 65, 'depth':1.73},
    {'mag': 2, 'nst': 19, 'dmin': 0, 'gap': 63, 'depth':4.551},
    {'mag': 0.74, 'nst': 7, 'dmin': 0.010310, 'gap': 90, 'depth':2.08},
    {'mag': 2.38, 'nst': 31, 'dmin': 0.025160, 'gap': 57, 'depth':11.37}
]
```

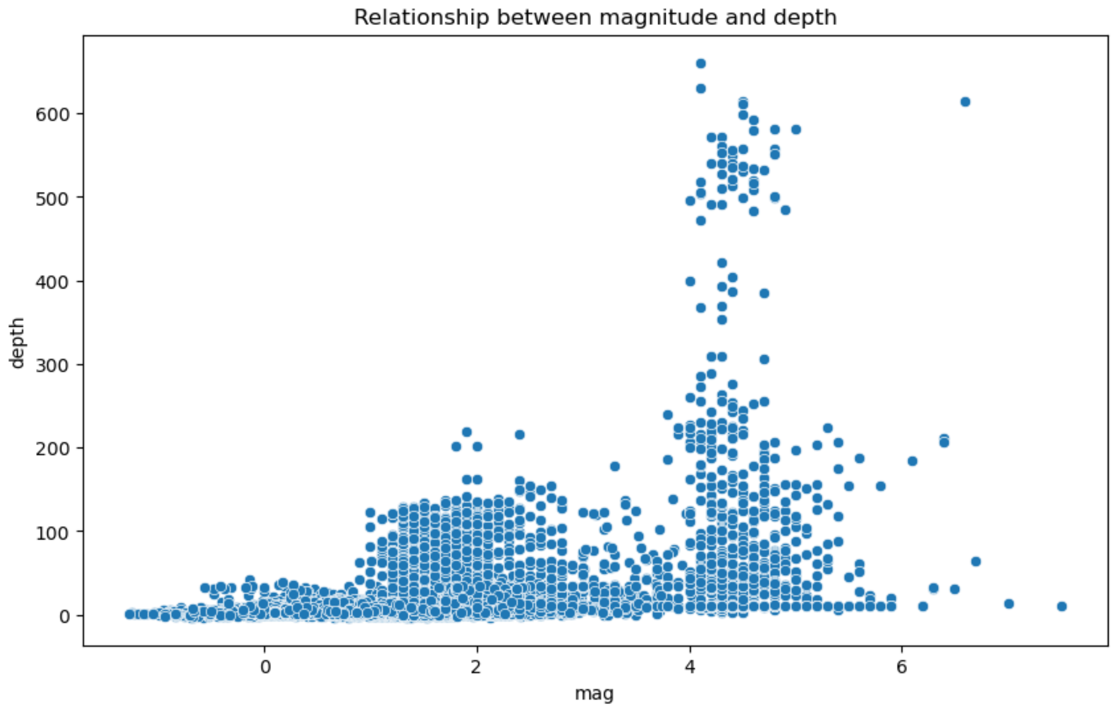
1.2 2. Simple Linear Regression

1.2.1 Predict a continuous target variable using one predictor

[181]:
```python
# Using 'Magnitude' as predictor and 'Depth' as response

X = eq['mag']
y = eq['depth']
```

1.2.2 Visualize the relationship with a scatter plot

[183]:
```python
plt.figure(figsize=(10,6))
sns.scatterplot(x='mag', y='depth', data=eq)
plt.title('Relationship between magnitude and depth')
plt.show()
```



1.2.3 Fit a linear regression model and print the model's summary

[185]:
```
# Fitting a Linear Model
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()

# Printing a summary
model.summary()
```

[185]:

Dep. Variable:	depth	**R-squared:**	0.150
Model:	OLS	**Adj. R-squared:**	0.150
Method:	Least Squares	**F-statistic:**	1665.
Date:	Sun, 18 Feb 2024	**Prob (F-statistic):**	0.00
Time:	05:53:14	**Log-Likelihood:**	-49767.
No. Observations:	9451	**AIC:**	9.954e+04
Df Residuals:	9449	**BIC:**	9.955e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P> |t|	[0.025	0.975]
const	-4.5457	0.797	-5.703	0.000	-6.108	-2.983
mag	15.8810	0.389	40.803	0.000	15.118	16.644

Omnibus:	11037.060	**Durbin-Watson:**	1.965
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1314855.040
Skew:	6.164	**Prob(JB):**	0.00
Kurtosis:	59.454	**Cond. No.**	3.94

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

1.2.4 Analyze the model summary

[109]:
```
# Intercept coefficient: -4.5457

# Magnitude coefficient: 15.881. For every one-unit increase in Magnitude, the␣
↪price is expected to increase by approximately 15.881 units

# R-squared: 0.150 -> This suggests that approximately 15% of the variability␣
↪in the dependent variable, price, can be explained by the model.
```

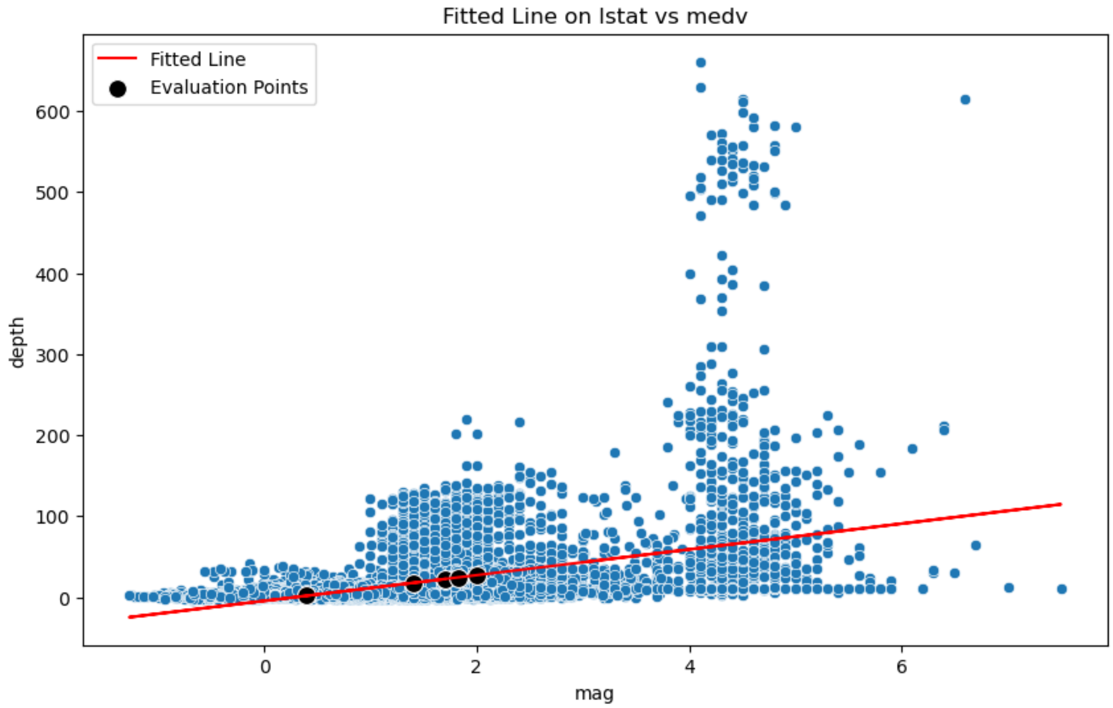
1.2.5 Model Equation

depth = -4.5457 + 15.881 * magnitude

1.2.6 Plotting the regression line and evaluation points

[187]:
```
# Get a list of the evaluation points for mileage
mag_values = []
for point in data_points:
    mag_values.append(point['mag'])

# Least squares coefficients
beta_1 = model.params['mag']
beta_0 = model.params['const']

# Extracting price values using the regression equation for these mileage values
depth_values = beta_0 + beta_1 * np.array(mag_values)

# Original plot with scatter points and regression line
plt.figure(figsize=(10,6))
sns.scatterplot(x=eq['mag'], y=eq['depth'])
plt.plot(eq['mag'], beta_0 + beta_1 * eq['mag'], color='red', label="Fitted␣
↪Line")

# Adding the points from the dictionary with a different color and size
sns.scatterplot(x=mag_values, y=depth_values, color='black', s=100,␣
↪label="Evaluation Points")

# Title and legend
plt.title('Fitted Line on lstat vs medv')
```

```
plt.legend()
plt.show()
```



1.3 3. Multicollinearity

1.3.1 Choose predictors and create a subset of your dataset

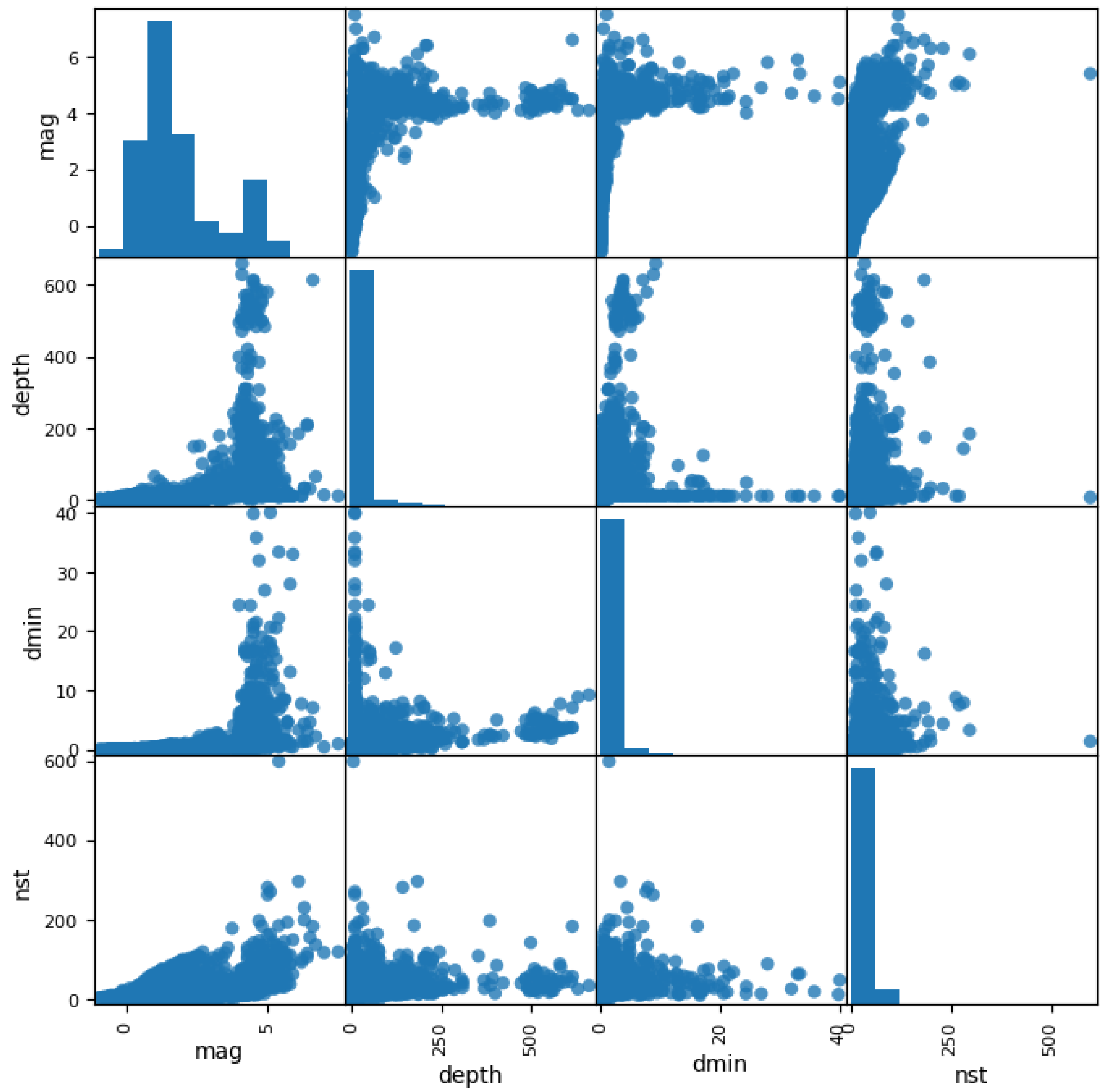
[225]:
```
selected_predictors = ['mag', 'depth', 'dmin', 'nst']
X = eq[selected_predictors]
```

1.3.2 Visualize the relationships using a scatter plot matrix

[227]:
```
pd.plotting.scatter_matrix(X, figsize=(8, 8), alpha=0.8,␣
↪marker='o',diagonal='hist')
```

[227]:
```
array([[<Axes: xlabel='mag', ylabel='mag'>,
        <Axes: xlabel='depth', ylabel='mag'>,
        <Axes: xlabel='dmin', ylabel='mag'>,
        <Axes: xlabel='nst', ylabel='mag'>],
       [<Axes: xlabel='mag', ylabel='depth'>,
        <Axes: xlabel='depth', ylabel='depth'>,
        <Axes: xlabel='dmin', ylabel='depth'>,
        <Axes: xlabel='nst', ylabel='depth'>],
       [<Axes: xlabel='mag', ylabel='dmin'>,
```

```
 <Axes: xlabel='depth', ylabel='dmin'>,
 <Axes: xlabel='dmin', ylabel='dmin'>,
 <Axes: xlabel='nst', ylabel='dmin'>],
[<Axes: xlabel='mag', ylabel='nst'>,
 <Axes: xlabel='depth', ylabel='nst'>,
 <Axes: xlabel='dmin', ylabel='nst'>,
 <Axes: xlabel='nst', ylabel='nst'>]], dtype=object)
```



[307]:
```
# Visual Observations:
# - mag: Appears to be correlated with dmin (horizontal distance from the
 ↪epicenter to the nearest station in degrees), depth of the event, and nst
#         (the total number of seismic stations used to determine earthquake
 ↪location)
```

```
# - depth: Appears to have a correlation with the magnitude
# - dmin: Shows no clear correlation with other predictors
# - nst: Shows no clear correlation with other predictors
```

1.3.3 Calculate the VIF (Variance Inflation Factor) for these predictors

[228]:
```
for i in range(X.shape[1]):
    vif = inf.variance_inflation_factor(X.values, i) # Compute VIF
    print(f"VIF for {X.columns[i]}: \t{vif:10.3f}")
```

```
VIF for mag:         3.529
VIF for depth:       1.328
VIF for dmin:        1.378
VIF for nst:         2.679
```

[319]:
```
# Observations from VIF:

# mag - Relative high VIF; some multicollinearity with other predictors
# depth - Relative low VIF; some multicollinearity is present
# dmin - Relative low VIF; some multicollinearity is present
# nst - Moderate VIF; some multicollinearity is present
```

1.4 4. Multiple Linear Regression

[261]:
```
# Subset of the data with 4 predictors
X = pd.DataFrame(eq[['mag','dmin','nst','gap']])
y = eq['depth']

# Fitting a Linear Model
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()

#Print a summary
model.summary()
```

[261]:

Dep. Variable:	depth	**R-squared:**	0.179
Model:	OLS	**Adj. R-squared:**	0.178
Method:	Least Squares	**F-statistic:**	288.1
Date:	Sun, 18 Feb 2024	**Prob (F-statistic):**	1.80e-224
Time:	06:17:54	**Log-Likelihood:**	-28854.
No. Observations:	5300	**AIC:**	5.772e+04
Df Residuals:	5295	**BIC:**	5.775e+04
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P> |t|	[0.025	0.975]
const	-8.8859	2.125	-4.182	0.000	-13.051	-4.721
mag	18.1389	0.736	24.643	0.000	16.696	19.582
dmin	0.7266	0.381	1.908	0.056	-0.020	1.473
nst	-0.0462	0.041	-1.124	0.261	-0.127	0.034
gap	-0.0394	0.016	-2.412	0.016	-0.071	-0.007

Omnibus:	6051.511	**Durbin-Watson:**	1.922
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	520911.075
Skew:	6.026	**Prob(JB):**	0.00
Kurtosis:	50.049	**Cond. No.**	308.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[244]:
```
# Observations:
# The R-squared value for the multiple linear regression model is higher (0.150␣
↪vs. 0.179),
# indicating that this model explains a greater proportion of the variance in␣
↪the dependent variable (depth).
```

1.4.1 Model equation

depth = -8.8859 + (18.1389 * mag) + (0.7266 * dmin) - (0.0461 * nst) - (0.0394 * gap)

[271]:
```
# Get lists of the evaluation points for all predictors
mag_values = []
dmin_values = []
nst_values = []
gap_values = []

for point in data_points:
    mag_values.append(point['mag'])
    dmin_values.append(point['dmin'])
    nst_values.append(point['nst'])
    gap_values.append(point['gap'])

# Extracting coefficients from the model
beta_0 = model.params['const']
beta_mag = model.params['mag']
beta_dmin = model.params['dmin']
beta_nst = model.params['nst']
beta_gap = model.params['gap']

# Calculating depth values using the multiple regression equation
depth_values = (beta_0 +
                beta_mag * np.array(mag_values) +
                beta_dmin * np.array(dmin_values) +
```

```
               beta_nst * np.array(nst_values) +
               beta_gap * np.array(gap_values))

# Printing the depth values along with the predictor values
for i, (mag, dmin, nst, gap, depth) in enumerate(zip(mag_values, dmin_values,␣
 ↪nst_values, gap_values, depth_values)):
    print(f"Data Point {i+1} - mag: {mag:.2f}, dmin: {dmin:.2f}, nst: {nst:.
 ↪2f}, gap: {gap:.2f}, Estimated depth: {depth:.2f}")
```

```
Data Point 1 - mag: 1.70, dmin: 0.10, nst: 18.00, gap: 69.00, Estimated depth:
18.47
Data Point 2 - mag: 0.40, dmin: 0.01, nst: 9.00, gap: 65.00, Estimated depth:
-4.60
Data Point 3 - mag: 2.00, dmin: 0.00, nst: 19.00, gap: 63.00, Estimated depth:
24.03
Data Point 4 - mag: 0.74, dmin: 0.01, nst: 7.00, gap: 90.00, Estimated depth:
0.68
Data Point 5 - mag: 2.38, dmin: 0.03, nst: 31.00, gap: 57.00, Estimated depth:
30.63
```