31-07-newspaper

February 13, 2025

[ ]:
```
from google.colab import files
 uploaded = files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving NewspaperData.csv to NewspaperData.csv
```

[ ]:
```
import pandas as pd
 df = pd.read_csv('NewspaperData.csv')
 df.head()
```

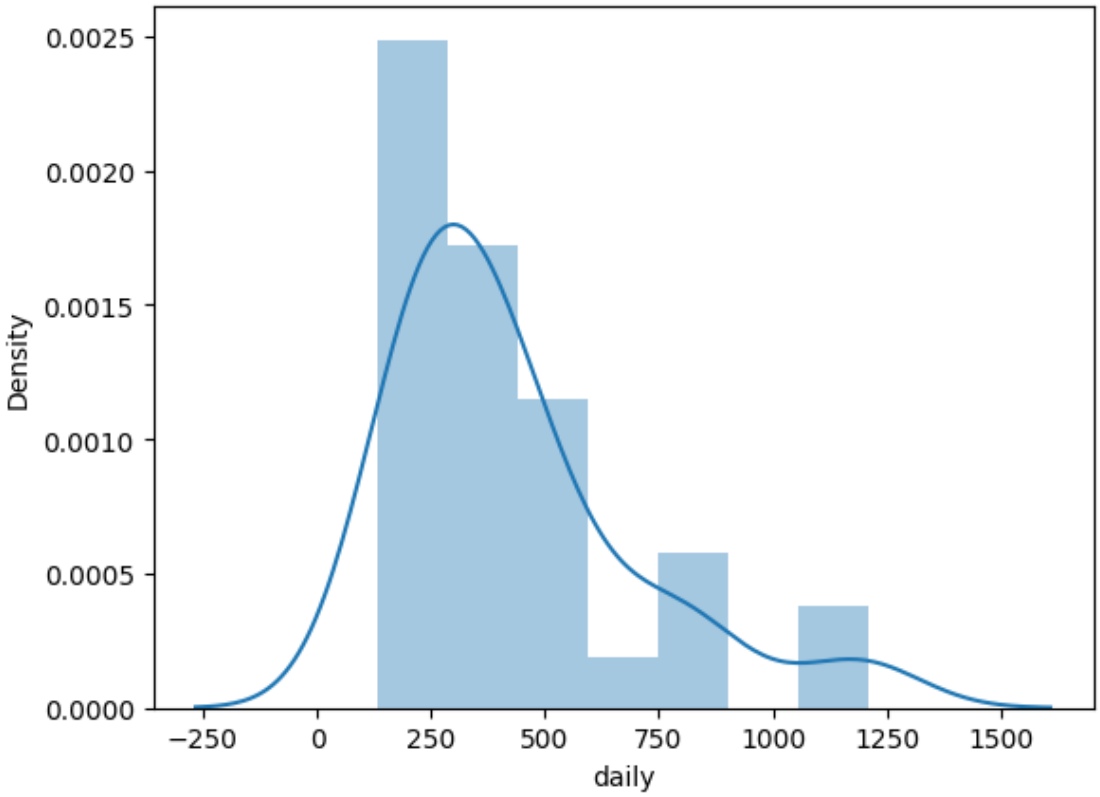
[ ]:

	Newspaper	daily	sunday
0	Baltimore Sun	391.952	488.506
1	Boston Globe	516.981	798.298
2	Boston Herald	355.628	235.084
3	Charlotte Observer	238.555	299.451
4	Chicago Sun Times	537.780	559.093

[ ]:
```
import seaborn as sns
sns.distplot(df['daily'])
```

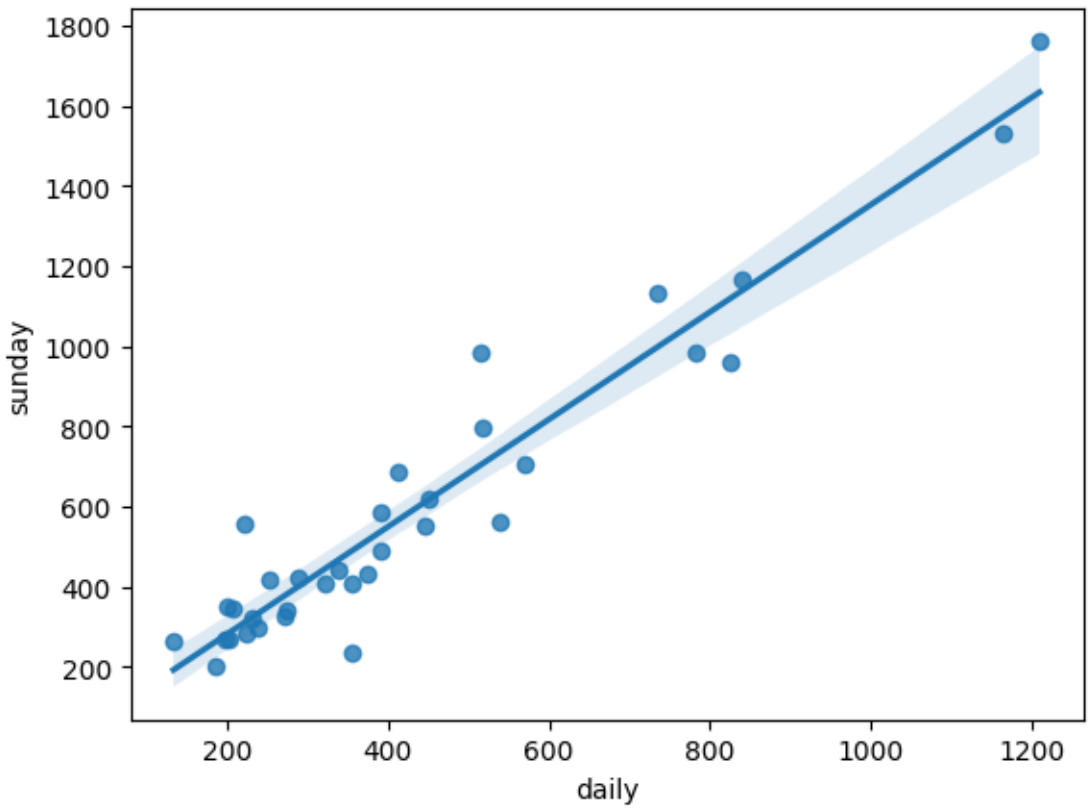
```
<ipython-input-3-60e927544913>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['daily'])
```

[ ]:
```
<Axes: xlabel='daily', ylabel='Density'>
```



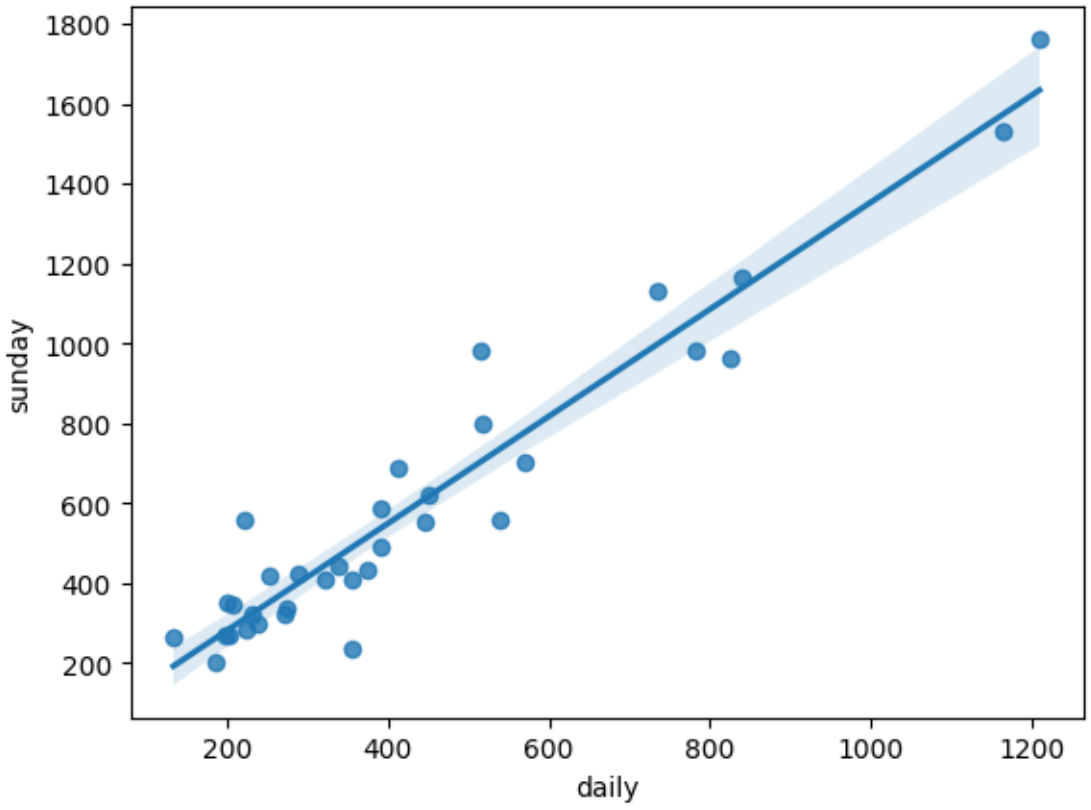
[ ]:
```
#fitting a linear regression model
 import statsmodels.formula.api as smf
 model = smf.ols('sunday~daily',data=df).fit()
 sns.regplot(x='daily',y='sunday',data=df)
```

[ ]: <Axes: xlabel='daily', ylabel='sunday'>



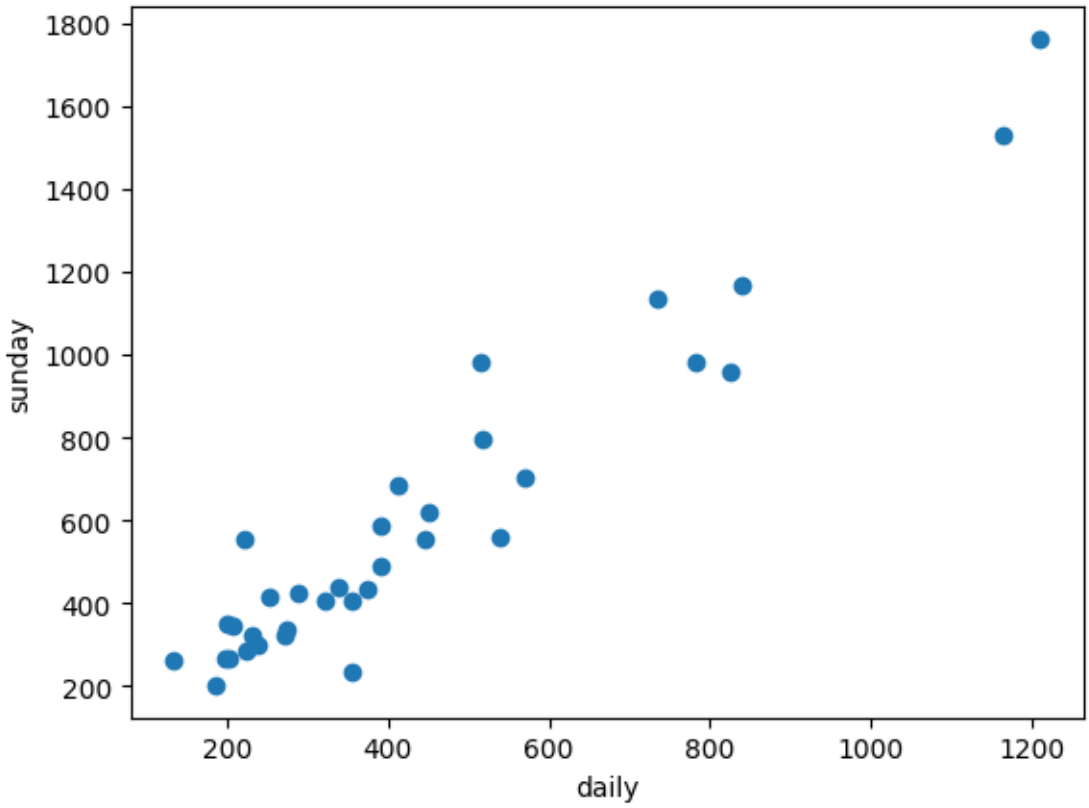
[ ]:
```
import seaborn as sns
sns.regplot(x='daily',y='sunday',data=df)
```

[ ]: <Axes: xlabel='daily', ylabel='sunday'>



[ ]:
```
import matplotlib.pyplot as plt
plt.scatter(df.daily,df.sunday)
plt.xlabel('daily')
plt.ylabel('sunday')
```

[ ]:
```
Text(0, 0.5, 'sunday')
```



[ ]:
```
from sklearn.metrics import accuracy_score
```

[ ]:
```
# prompt: define smf

# You have already imported statsmodels.formula.api as smf in your provided␣
↪code.
# So, smf is already defined as an alias for statsmodels.formula.api.

# Example usage (from your provided code):
# model = smf.ols('sunday~daily',data=df).fit()
```

[ ]:
```
# prompt: define model

model = smf.ols('sunday~daily',data=df).fit()
print(model.summary())
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-34-4c50d6aa6d80> in <cell line: 0>()
      1 # prompt: define model
      2
```

```
----> 3 model = smf.ols('sunday~daily',data=df).fit()
      4 print(model.summary())

NameError: name 'smf' is not defined
```

[ ]:
```
# prompt: R square value

model.rsquared
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-30-1ed9329df096> in <cell line: 0>()
      1 # prompt: R square value
      2
----> 3 model.rsquared

NameError: name 'model' is not defined
```

[ ]:
```
#R Squared value
print(model.tvalues, '\n',model.pvalues)
(model.rsquared,model.rsquared_adj)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-24-fbe62d8a75a3> in <cell line: 0>()
      1 #R Squared value
----> 2 print(model.tvalues, '\n',model.pvalues)
      3 (model.rsquared,model.rsquared_adj)

NameError: name 'model' is not defined
```

[ ]:
```
newdata=pd.Series([200,300])
```

[ ]:
```
data_pred=pd.DataFrame(newdata,columns=['daily'])
```

[ ]:
```

```

[ ]:
```
model.predict(data_pred)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-29-025ab8a983ec> in <cell line: 0>()
----> 1 model.predict(data_pred)
```

```
NameError: name 'model' is not defined
```

[ ]:
```
# Import necessary libraries
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd
import re
from nltk.corpus import stopwords
import nltk

# Download NLTK stopwords if not already available
nltk.download('stopwords')

# Load your dataset
url = 'path_to_your_newspaper_data.csv'  # Replace with the actual path or URL␣
 ↪to your dataset
df = pd.read_csv(url)

# Preprocess the text data
stop_words = set(stopwords.words('english'))

def preprocess_text(text):
    text = re.sub(r'[^a-zA-Z0-9\s]', '', text)  # Remove non-alphanumeric␣
 ↪characters
    text = text.lower()  # Convert to lowercase
    text = ' '.join([word for word in text.split() if word not in stop_words]) ␣
 ↪# Remove stopwords
    return text

df['processed_text'] = df['Text'].apply(preprocess_text)

# Vectorize the text data
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(df['processed_text'])

# Assuming 'Length' is the continuous target variable you want to predict
y = df['Length']  # Replace 'Length' with your actual target column name

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Train a Linear Regression model
model = LinearRegression()
```

```
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Calculate R² score
r2 = r2_score(y_test, y_pred)
print(f'R² Score: {r2:.4f}')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-26-0a9ca8bbd19c> in <cell line: 0>()
     14 # Load your dataset
     15 url = 'path_to_your_newspaper_data.csv'  # Replace with the actual path
 ↪or URL to your dataset
---> 16 df = pd.read_csv(url)
     17
     18 # Preprocess the text data

/usr/local/lib/python3.11/dist-packages/pandas/io/parsers/readers.py in␣
 ↪read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col,␣
 ↪usecols, dtype, engine, converters, true_values, false_values,␣
 ↪skipinitialspace, skiprows, skipfooter, nrows, na_values, keep_default_na,␣
 ↪na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format,␣
 ↪keep_date_col, date_parser, date_format, dayfirst, cache_dates, iterator,␣
 ↪chunksize, compression, thousands, decimal, lineterminator, quotechar,␣
 ↪quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect␣
 ↪on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision,␣
 ↪storage_options, dtype_backend)
   1024     kwds.update(kwds_defaults)
   1025
-> 1026     return _read(filepath_or_buffer, kwds)
   1027
   1028


/usr/local/lib/python3.11/dist-packages/pandas/io/parsers/readers.py in␣
 ↪_read(filepath_or_buffer, kwds)
    618
    619     # Create the parser.
--> 620     parser = TextFileReader(filepath_or_buffer, **kwds)
    621
    622     if chunksize or iterator:

/usr/local/lib/python3.11/dist-packages/pandas/io/parsers/readers.py in␣
 ↪__init__(self, f, engine, **kwds)
   1618
   1619         self.handles: IOHandles | None = None
```

```
-> 1620         self._engine = self._make_engine(f, self.engine)
   1621
   1622     def close(self) -> None:

/usr/local/lib/python3.11/dist-packages/pandas/io/parsers/readers.py in␣
 ↪_make_engine(self, f, engine)
   1878                 if "b" not in mode:
   1879                     mode += "b"
-> 1880             self.handles = get_handle(
   1881                 f,
   1882                 mode,

/usr/local/lib/python3.11/dist-packages/pandas/io/common.py in␣
 ↪get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text,␣
 ↪errors, storage_options)
    871         if ioargs.encoding and "b" not in ioargs.mode:
    872             # Encoding
--> 873             handle = open(
    874                 handle,
    875                 ioargs.mode,

FileNotFoundError: [Errno 2] No such file or directory:␣
 ↪'path_to_your_newspaper_data.csv'
```