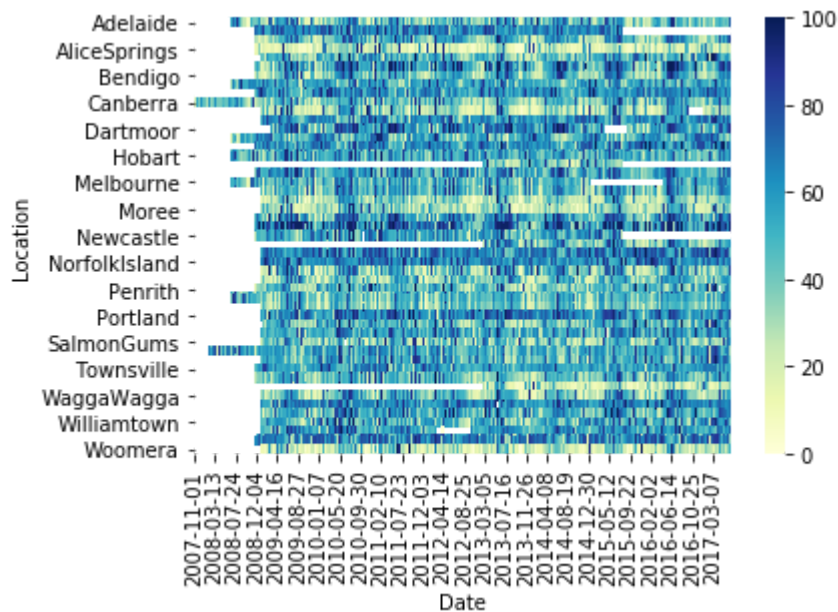


## Seaborn heatmap for rainfall prediction

```
In [28]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df=pd.read_csv("weatherAUS.csv")
```

```
In [29]: custom_pivot = df.pivot(index="Location", columns="Date", values="Humidity3p")
sns.heatmap(custom_pivot,cmap="YlGnBu")
plt.show()
```

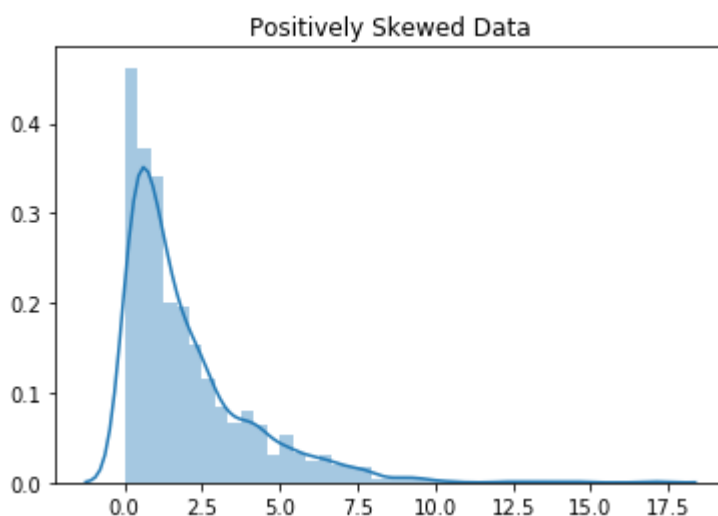


## Visualisation and removal of Skewness

```
In [20]: import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

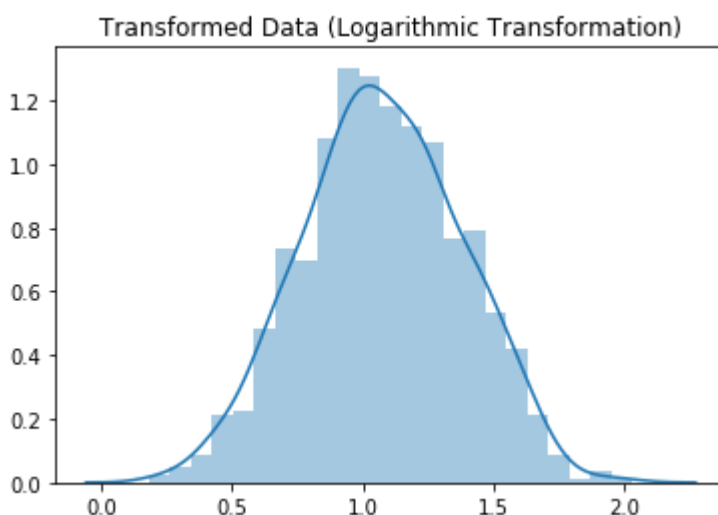
# Generate a positively skewed dataset
np.random.seed(0)
skewed_data = np.random.exponential(scale=2, size=1000)

# Plot the histogram to visualize the skewness
sns.distplot(skewed_data, kde=True)
plt.title("Positively Skewed Data")
plt.show()
```



```
In [27]: # Apply a logarithmic transformation to the skewed data
log_transformed_data = np.power(skewed_data, 1/4)

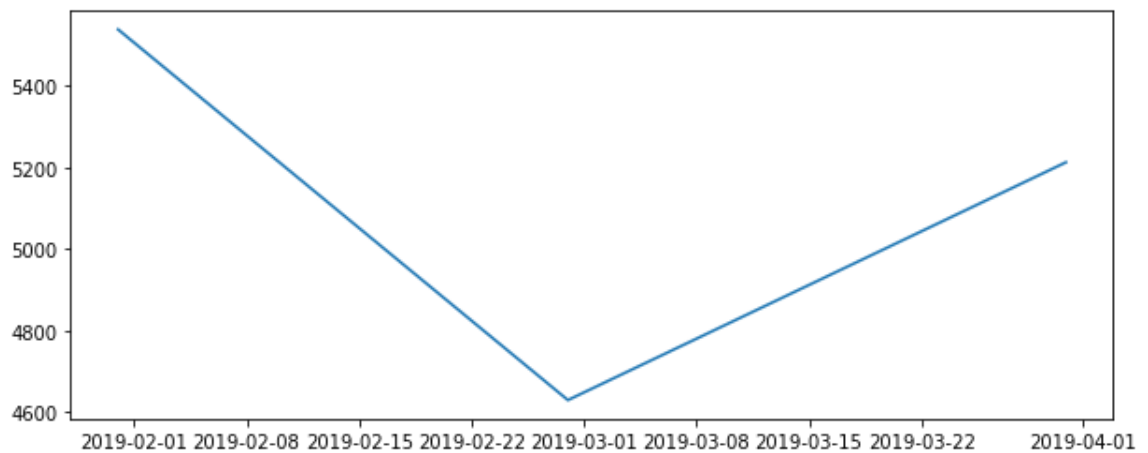
# Plot the histogram of the transformed data
sns.distplot(log_transformed_data, kde=True)
plt.title("Transformed Data (Logarithmic Transformation)")
plt.show()
```



## Time Series Visualisation of Sales month wise

```
In [49]: data2=pd.read_csv("supermarket_sales - Sheet1.csv")
data2['Date'] = pd.to_datetime(data2['Date'])
xx = data2.groupby(pd.Grouper(key='Date', freq='M')).sum()
xx = xx.reset_index()
plt.figure(figsize=(10, 4))
plt.plot(xx["Date"],xx["gross income"])
```

Out[49]: [<matplotlib.lines.Line2D at 0x1edac278320>]



## Build a scatter plot and perform Dimensity reduction

```
In [53]: import pandas as pd
import numpy as np
from sklearn.feature_selection import SelectKBest, f_regression
import matplotlib.pyplot as plt

# Create a small random dataset with 3 features and a target variable
np.random.seed(0)
data = pd.DataFrame({
    'Feature1': np.random.rand(50),
    'Feature2': np.random.rand(50),
    'Feature3': np.random.rand(50),
    'Target': 2 * np.random.rand(50) + 1
})

# Select 2 best features based on f_regression scores
X = data[['Feature1', 'Feature2', 'Feature3']]
y = data['Target']
selector = SelectKBest(score_func=f_regression, k=2)
X_new = selector.fit_transform(X, y)

# Scatter plot of the selected features
plt.scatter(X_new[:, 0], X_new[:, 1])
plt.show()
```

