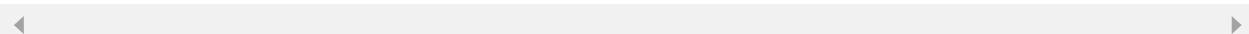


In [109...]

```
!pip install TensorFlow
!pip install keras
!pip install sklearn
```

```
Requirement already satisfied: TensorFlow in c:\users\matt\miniconda3\lib\site-packages (2.3.0)
Requirement already satisfied: six>=1.12.0 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (1.15.0)
Requirement already satisfied: grpcio>=1.8.6 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (1.41.1)
Requirement already satisfied: numpy<1.19.0,>=1.16.0 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (1.18.5)
Requirement already satisfied: wheel>=0.26 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (0.35.1)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (3.3.0)
Requirement already satisfied: google-pasta>=0.1.8 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (0.2.0)
Requirement already satisfied: gast==0.3.3 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (0.3.3)
Requirement already satisfied: keras-preprocessing<1.2,>=1.1.1 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (1.1.2)
Requirement already satisfied: wrapt>=1.11.1 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (1.12.1)
Requirement already satisfied: protobuf>=3.9.2 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (3.19.1)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (1.1.0)
Requirement already satisfied: h5py<2.11.0,>=2.10.0 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (2.10.0)
Requirement already satisfied: scipy==1.4.1 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (1.4.1)
Requirement already satisfied: tensorflow-estimator<2.4.0,>=2.3.0 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (2.3.0)
Requirement already satisfied: tensorboard<3,>=2.3.0 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (2.7.0)
Requirement already satisfied: astunparse==1.6.3 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (1.6.3)
Requirement already satisfied: absl-py>=0.7.0 in c:\users\matt\miniconda3\lib\site-packages (from TensorFlow) (0.15.0)
Requirement already satisfied: setuptools>=41.0.0 in c:\users\matt\miniconda3\lib\site-packages (from tensorflow<3,>=2.3.0->TensorFlow) (50.3.1.post20201107)
Requirement already satisfied: werkzeug>=0.11.15 in c:\users\matt\miniconda3\lib\site-packages (from tensorboard<3,>=2.3.0->TensorFlow) (2.0.2)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in c:\users\matt\miniconda3\lib\site-packages (from tensorboard<3,>=2.3.0->TensorFlow) (1.8.0)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in c:\users\matt\miniconda3\lib\site-packages (from tensorboard<3,>=2.3.0->TensorFlow) (0.6.1)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\matt\miniconda3\lib\site-packages (from tensorboard<3,>=2.3.0->TensorFlow) (2.25.1)
Requirement already satisfied: markdown>=2.6.8 in c:\users\matt\miniconda3\lib\site-packages (from tensorboard<3,>=2.3.0->TensorFlow) (3.3.4)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in c:\users\matt\miniconda3\lib\site-packages (from tensorboard<3,>=2.3.0->TensorFlow) (0.4.6)
Requirement already satisfied: google-auth<3,>=1.6.3 in c:\users\matt\miniconda3\lib\site-packages (from tensorboard<3,>=2.3.0->TensorFlow) (1.35.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\matt\miniconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<3,>=2.3.0->TensorFlow) (1.25.11)
```

```
Requirement already satisfied: idna<3,>=2.5 in c:\users\matt\miniconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<3,>=2.3.0->TensorFlow) (2.10)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\matt\miniconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<3,>=2.3.0->TensorFlow) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\matt\miniconda3\lib\site-packages (from requests<3,>=2.21.0->tensorboard<3,>=2.3.0->TensorFlow) (2021.10.8)
Requirement already satisfied: requests-oauthlib>=0.7.0 in c:\users\matt\miniconda3\lib\site-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard<3,>=2.3.0->TensorFlow) (1.3.0)
Requirement already satisfied: rsa<5,>=3.1.4; python_version >= "3.6" in c:\users\matt\miniconda3\lib\site-packages (from google-auth<3,>=1.6.3->tensorboard<3,>=2.3.0->TensorFlow) (4.7.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\matt\miniconda3\lib\site-packages (from google-auth<3,>=1.6.3->tensorboard<3,>=2.3.0->TensorFlow) (0.2.8)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in c:\users\matt\miniconda3\lib\site-packages (from google-auth<3,>=1.6.3->tensorboard<3,>=2.3.0->TensorFlow) (4.2.4)
Requirement already satisfied: oauthlib>=3.0.0 in c:\users\matt\miniconda3\lib\site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard<3,>=2.3.0->TensorFlow) (3.1.1)
Requirement already satisfied: pyasn1>=0.1.3 in c:\users\matt\miniconda3\lib\site-packages (from rsa<5,>=3.1.4; python_version >= "3.6"->google-auth<3,>=1.6.3->tensorboard<3,>=2.3.0->TensorFlow) (0.4.8)
Requirement already satisfied: keras in c:\users\matt\miniconda3\lib\site-packages (2.4.3)
Requirement already satisfied: pyyaml in c:\users\matt\miniconda3\lib\site-packages (from keras) (6.0)
Requirement already satisfied: scipy>=0.14 in c:\users\matt\miniconda3\lib\site-packages (from keras) (1.4.1)
Requirement already satisfied: h5py in c:\users\matt\miniconda3\lib\site-packages (from keras) (2.10.0)
Requirement already satisfied: numpy>=1.9.1 in c:\users\matt\miniconda3\lib\site-packages (from keras) (1.18.5)
Requirement already satisfied: six in c:\users\matt\miniconda3\lib\site-packages (from h5py->keras) (1.15.0)
Requirement already satisfied: sklearn in c:\users\matt\miniconda3\lib\site-packages (0.0)
Requirement already satisfied: scikit-learn in c:\users\matt\miniconda3\lib\site-packages (from sklearn) (0.24.2)
Requirement already satisfied: scipy>=0.19.1 in c:\users\matt\miniconda3\lib\site-packages (from scikit-learn->sklearn) (1.4.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\matt\miniconda3\lib\site-packages (from scikit-learn->sklearn) (2.2.0)
Requirement already satisfied: joblib>=0.11 in c:\users\matt\miniconda3\lib\site-packages (from scikit-learn->sklearn) (1.0.1)
Requirement already satisfied: numpy>=1.13.3 in c:\users\matt\miniconda3\lib\site-packages (from scikit-learn->sklearn) (1.18.5)
```



In [44]:

#Import Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import tensorflow as tf
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
```

In [3]:

```
#Data Exploration

global_temperature = pd.read_csv("GlobalTemperatures.csv")

print(global_temperature.head(5))
print(global_temperature.shape)
```

	dt	LandAverageTemperature	LandAverageTemperatureUncertainty	\
0	1750-01-01	3.034	3.574	
1	1750-02-01	3.083	3.702	
2	1750-03-01	5.626	3.076	
3	1750-04-01	8.490	2.451	
4	1750-05-01	11.573	2.072	
	LandMaxTemperature	LandMaxTemperatureUncertainty	LandMinTemperature	\
0	NaN	NaN	NaN	
1	NaN	NaN	NaN	
2	NaN	NaN	NaN	
3	NaN	NaN	NaN	
4	NaN	NaN	NaN	
	LandMinTemperatureUncertainty	LandAndOceanAverageTemperature	\	
0	NaN	NaN		
1	NaN	NaN		
2	NaN	NaN		
3	NaN	NaN		
4	NaN	NaN		
	LandAndOceanAverageTemperatureUncertainty			
0	NaN			
1	NaN			
2	NaN			
3	NaN			
4	NaN			
	(3192, 9)			

In [4]:

```
global_temperature.columns
```

```
Out[4]: Index(['dt', 'LandAverageTemperature', 'LandAverageTemperatureUncertainty',
       'LandMaxTemperature', 'LandMaxTemperatureUncertainty',
       'LandMinTemperature', 'LandMinTemperatureUncertainty',
       'LandAndOceanAverageTemperature',
       'LandAndOceanAverageTemperatureUncertainty'],
      dtype='object')
```

In [5]:

```
#Identify Missing Data
```

```
global_temperature.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3192 entries, 0 to 3191
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype 

```

```
---      -----
0  dt                      3192 non-null  object
1  LandAverageTemperature    3180 non-null  float64
2  LandAverageTemperatureUncertainty  3180 non-null  float64
3  LandMaxTemperature       1992 non-null  float64
4  LandMaxTemperatureUncertainty  1992 non-null  float64
5  LandMinTemperature       1992 non-null  float64
6  LandMinTemperatureUncertainty  1992 non-null  float64
7  LandAndOceanAverageTemperature  1992 non-null  float64
8  LandAndOceanAverageTemperatureUncertainty  1992 non-null  float64
dtypes: float64(8), object(1)
memory usage: 224.6+ KB
```

In [6]: `global_temperature.isnull().sum()`

```
Out[6]: dt          0
LandAverageTemperature   12
LandAverageTemperatureUncertainty  12
LandMaxTemperature      1200
LandMaxTemperatureUncertainty  1200
LandMinTemperature      1200
LandMinTemperatureUncertainty  1200
LandAndOceanAverageTemperature  1200
LandAndOceanAverageTemperatureUncertainty  1200
dtype: int64
```

In [7]: *#Drop missing values from dataframe*

```
global_temperature.dropna(inplace=True)
global_temperature.isnull().sum()
```

```
Out[7]: dt          0
LandAverageTemperature   0
LandAverageTemperatureUncertainty  0
LandMaxTemperature      0
LandMaxTemperatureUncertainty  0
LandMinTemperature      0
LandMinTemperatureUncertainty  0
LandAndOceanAverageTemperature  0
LandAndOceanAverageTemperatureUncertainty  0
dtype: int64
```

In [8]: *#Create a function that transforms data from Celsius to Fahrenheit and apply*

```
def temp_convert(x):
    x = (x * 1.8) + 32
    return float(x)
```

```
global_temperature['LandAverageTemperature'] = global_temperature['LandAverageTemperature'].apply(temp_convert)
global_temperature['LandMaxTemperature'] = global_temperature['LandAverageTemperature'].apply(temp_convert)
global_temperature['LandMinTemperature'] = global_temperature['LandMinTemperature'].apply(temp_convert)
```

In [9]: *# Extract Year from date*

```
#global_temperature['dt'] = pd.to_datetime(global_temperature['dt'], format='%d-%m-%Y')
global_temperature['year'] = pd.DatetimeIndex(global_temperature['dt']).year
```

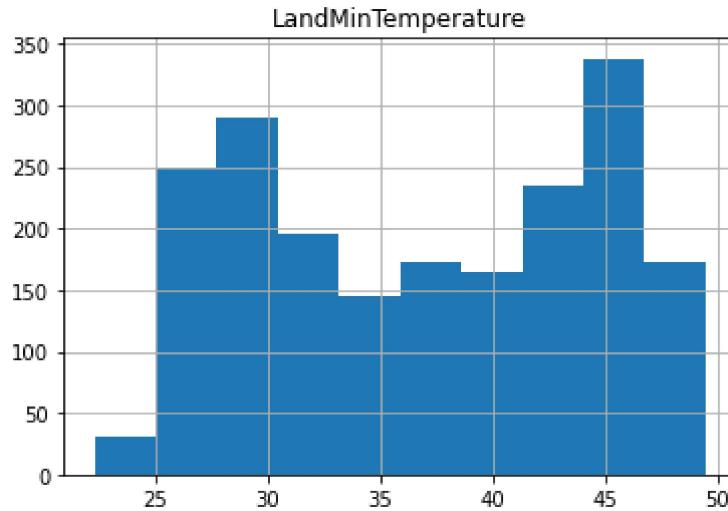
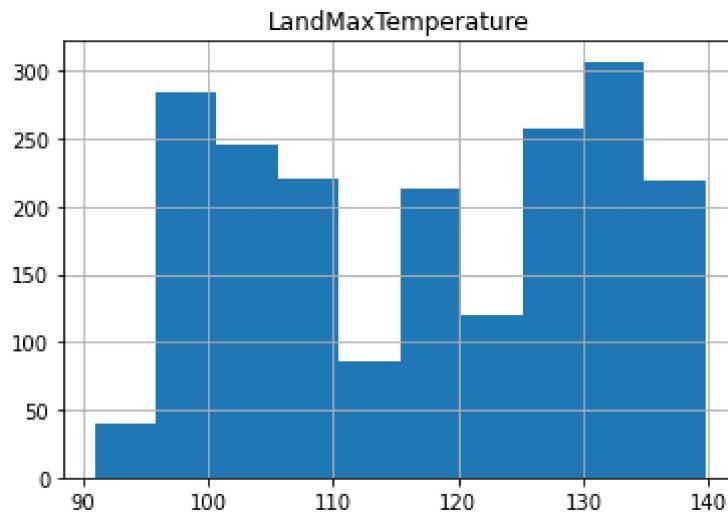
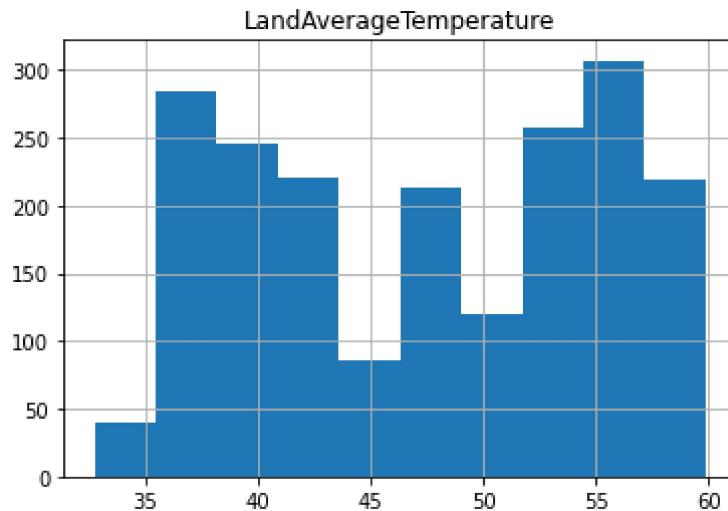
In [10]:

#Visualizations

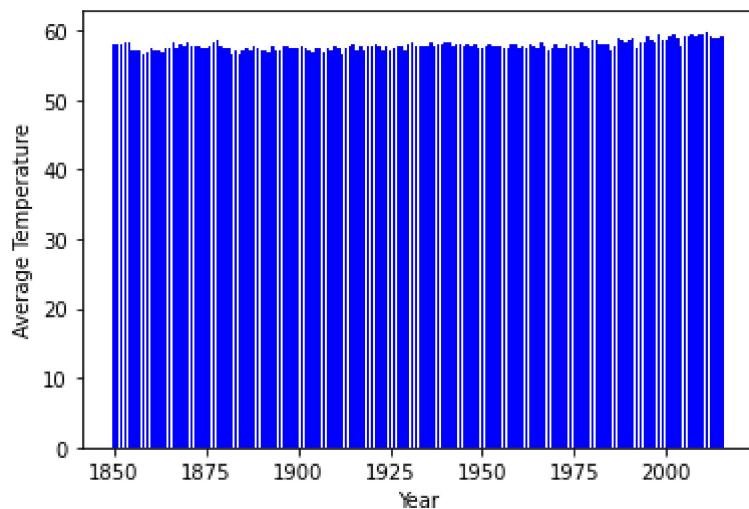
```
global_temperature.hist('LandAverageTemperature')
global_temperature.hist('LandMaxTemperature')
global_temperature.hist('LandMinTemperature')
```

Out[10]:

```
array([<AxesSubplot:title={'center':'LandMinTemperature'}>],  
      dtype=object)
```



```
In [12]: plt.xlabel("Year")
plt.ylabel("Average Temperature")
plt.bar(global_temperature[ 'year' ], global_temperature[ 'LandAverageTemperature' ], color
plt.show()
```



```
In [13]: fig = px.scatter(global_temperature[ 'LandAverageTemperature' ], x=global_temperature[ 'La
fig.show()
```

In []:

In []:

In [15]:

```
#Fit the model
X = global_temperature["year"].to_numpy().reshape(-1,1)
y = global_temperature["LandAverageTemperature"].to_numpy().reshape(-1,1)
model = LinearRegression()
model.fit(X,y)
```

Out[15]:

LinearRegression()

In []:

In [16]:

```
#R squared score
model.score(X,y)
```

Out[16]:

0.009200756797997922

In [17]:

```
#slope parameters
model.coef_
```

Out[17]:

array([[0.0153568]])

In [18]:

```
#intercept
model.intercept_
```

Out[18]:

array([17.75184048])

In [19]:

```
#Create train, testing data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

print(X_train.shape, y_train.shape)
print(X_test.shape, y_test.shape)
```

(1593, 1) (1593, 1)
(399, 1) (399, 1)

In [21]:

#Goodness of fit - R^2 value

```
print("Training set score: {:.2f} ".format(model.score(X_train, y_train)))
print("Test set score {:.2f} ".format(model.score(X_test, y_test)))
```

```
Training set score: 0.01
Test set score -0.00
```

In [42]:

```
#Establish MAE
```

```
y_pred = [y_train.mean()] * len(y_train)
print("MAE", round(mean_absolute_error(y_train, y_pred), 5))
```

```
MAE 6.87739
```

In [48]:

```
#Establish MSE
```

```
mse = mean_squared_error(y_train, y_pred, sample_weight=None, multioutput='uniform_aver
print("MSE", mse)
```

```
MSE 58.74890720985924
```

In [30]:

```
#Fit the training data
```

```
model2 = model.fit(X_train, y_train)
prediction2 = model.predict(X_test)
print(prediction2)
```

```
[[48.72981962]
 [46.74309471]
 [47.36605082]
 [47.51758069]
 [47.19768431]
 [48.02268024]
 [47.6522739 ]
 [47.38288748]
 [47.43339743]
 [48.29206667]
 [47.43339743]
 [48.61196306]
 [48.49410649]
 [47.18084765]
 [46.03595533]
 [47.24819426]
 [46.54105489]
 [47.38288748]
 [47.71962051]
 [46.01911868]
 [48.47726984]
 [45.95177207]
 [47.93849698]
 [48.5277798 ]
 [47.26503091]
 [46.38952502]
 [46.40636167]
 [48.05635355]
 [46.89462457]
 [45.98544538]
 [47.88798703]
 [46.38952502]
 [47.92166033]
 [46.27166846]
 [47.6186006 ]
```

```
[46.6420748 ]  
[48.5277798 ]  
[47.28186756]  
[47.07982774]  
[46.47370828]  
[47.60176395]  
[46.99564448]  
[46.32217841]  
[46.40636167]  
[46.01911868]  
[47.78696712]  
[47.92166033]  
[47.51758069]  
[47.07982774]  
[47.70278386]  
[46.44003498]  
[47.33237752]  
[47.26503091]  
[46.10330194]  
[47.63543725]  
[47.50074404]  
[47.23135761]  
[48.14053681]  
[47.90482368]  
[47.45023408]  
[46.91146123]  
[46.91146123]  
[48.1068635 ]  
[47.14717435]  
[46.94513453]  
[48.0395169 ]  
[46.96197118]  
[47.1303377 ]  
[48.1068635 ]  
[47.95533364]  
[46.10330194]  
[46.92829788]  
[48.27523002]  
[48.72981962]  
[47.39972413]  
[46.87778792]  
[47.73645716]  
[46.91146123]  
[46.33901507]  
[47.0966644 ]  
[46.08646529]  
[46.57472819]  
[47.51758069]  
[46.87778792]  
[47.90482368]  
[48.47726984]  
[46.00228203]  
[46.86095127]  
[46.86095127]  
[47.56809065]  
[47.6186006 ]  
[48.44359654]  
[46.1538119 ]  
[46.65891145]  
[47.11350105]
```

```
[47.68594721]
[48.30890332]
[47.93849698]
[48.15737346]
[48.20788341]
[47.70278386]
[48.5277798 ]
[48.15737346]
[48.51094315]
[46.49054493]
[47.60176395]
[46.30534176]
[47.41656078]
[47.19768431]
[47.87115038]
[48.39308658]
[46.20432185]
[47.24819426]
[46.84411462]
[47.0966644 ]
[48.27523002]
[47.19768431]
[46.74309471]
[46.6757481 ]
[48.02268024]
[48.24155672]
[46.30534176]
[48.46043319]
[46.30534176]
[47.28186756]
[48.64563636]
[47.78696712]
[47.41656078]
[48.39308658]
[47.80380377]
[47.85431373]
[48.57828975]
[47.5849273 ]
[46.50738158]
[48.29206667]
[48.15737346]
[46.7094214 ]
[47.50074404]
[48.5277798 ]
[46.6420748 ]
[47.164011 ]
[47.07982774]
[47.80380377]
[47.60176395]
[47.43339743]
[47.68594721]
[46.69258475]
[48.5614531 ]
[47.06299109]
[45.95177207]
[48.51094315]
[47.70278386]
[48.0395169 ]
[46.1538119 ]
[47.73645716]
```

```
[48.57828975]
[47.46707073]
[46.75993136]
[46.54105489]
[48.72981962]
[46.47370828]
[47.56809065]
[48.32573998]
[47.66911056]
[47.77013047]
[48.14053681]
[46.74309471]
[47.02931779]
[46.2211585 ]
[48.61196306]
[47.31554087]
[48.5614531 ]
[47.39972413]
[47.38288748]
[48.67930966]
[48.17421011]
[47.14717435]
[46.50738158]
[47.18084765]
[46.05279198]
[47.90482368]
[46.03595533]
[47.04615444]
[47.01248114]
[48.02268024]
[47.82064042]
[48.51094315]
[46.1538119 ]
[46.57472819]
[46.79360466]
[46.77676801]
[48.14053681]
[46.13697524]
[47.19768431]
[46.35585172]
[46.28850511]
[47.28186756]
[47.85431373]
[46.86095127]
[46.52421823]
[46.32217841]
[46.28850511]
[46.38952502]
[46.42319832]
[48.32573998]
[47.33237752]
[47.06299109]
[46.10330194]
[46.6420748 ]
[46.23799515]
[46.06962864]
[48.19104676]
[48.02268024]
[47.6186006 ]
[46.12013859]
```

```
[46.52421823]
[46.77676801]
[48.30890332]
[46.20432185]
[48.37624993]
[46.12013859]
[46.03595533]
[47.39972413]
[47.164011 ]
[47.93849698]
[48.40992323]
[46.38952502]
[46.49054493]
[48.09002685]
[48.69614631]
[46.89462457]
[48.44359654]
[46.37268837]
[47.36605082]
[46.06962864]
[47.71962051]
[48.42675989]
[46.74309471]
[48.0395169 ]
[47.28186756]
[47.71962051]
[46.37268837]
[46.10330194]
[48.47726984]
[46.12013859]
[46.50738158]
[46.97880783]
[45.96860873]
[46.72625806]
[48.19104676]
[47.04615444]
[48.46043319]
[46.13697524]
[47.38288748]
[46.87778792]
[46.60840149]
[45.96860873]
[47.34921417]
[46.1874852 ]
[46.69258475]
[48.66247301]
[47.51758069]
[47.55125399]
[47.87115038]
[47.63543725]
[48.67930966]
[46.74309471]
[48.5614531 ]
[48.25839337]
[46.06962864]
[47.11350105]
[48.54461645]
[46.12013859]
[47.45023408]
[47.50074404]
```

```
[46.40636167]
[47.50074404]
[47.85431373]
[46.10330194]
[46.69258475]
[48.49410649]
[48.42675989]
[47.88798703]
[46.57472819]
[46.38952502]
[47.33237752]
[48.54461645]
[46.40636167]
[48.57828975]
[46.94513453]
[46.79360466]
[47.95533364]
[47.1303377 ]
[46.6757481 ]
[46.05279198]
[47.80380377]
[46.49054493]
[48.09002685]
[47.75329381]
[47.43339743]
[48.0731902 ]
[46.94513453]
[46.2211585 ]
[48.57828975]
[48.5614531 ]
[47.51758069]
[47.34921417]
[46.20432185]
[47.45023408]
[48.42675989]
[46.77676801]
[47.98900694]
[47.63543725]
[48.66247301]
[48.17421011]
[48.0731902 ]
[47.83747707]
[47.75329381]
[48.35941328]
[47.6186006 ]
[47.98900694]
[45.95177207]
[47.0966644 ]
[46.23799515]
[46.75993136]
[47.63543725]
[47.48390739]
[46.03595533]
[47.31554087]
[47.75329381]
[47.83747707]
[48.64563636]
[46.28850511]
[47.95533364]
[48.14053681]
```

```
[46.52421823]
[47.51758069]
[47.43339743]
[47.6522739 ]
[46.10330194]
[47.18084765]
[46.30534176]
[48.32573998]
[46.45687163]
[47.39972413]
[46.37268837]
[47.71962051]
[47.88798703]
[46.77676801]
[46.52421823]
[47.6186006 ]
[47.83747707]
[46.47370828]
[46.55789154]
[47.73645716]
[48.67930966]
[46.25483181]
[47.95533364]
[46.00228203]
[47.78696712]
[46.27166846]
[46.7094214 ]
[46.54105489]
[46.97880783]
[47.06299109]
[47.18084765]
[46.38952502]
[48.69614631]
[47.68594721]
[47.45023408]
[46.01911868]
[46.27166846]
[48.17421011]
[48.5277798 ]
[46.89462457]
[48.40992323]
[46.87778792]
[47.41656078]
[47.36605082]
[48.22472006]
[47.85431373]
[47.95533364]
[47.88798703]
[48.1068635 ]
[46.1538119 ]
[48.37624993]
[48.5614531 ]
[46.6420748 ]
[47.29870422]
[47.36605082]
[47.77013047]
[48.27523002]
[48.35941328]
[46.81044131]
[48.64563636]
```

```
[46.6757481 ]  
[47.18084765]  
[47.87115038]  
[46.50738158]]
```

In [31]:

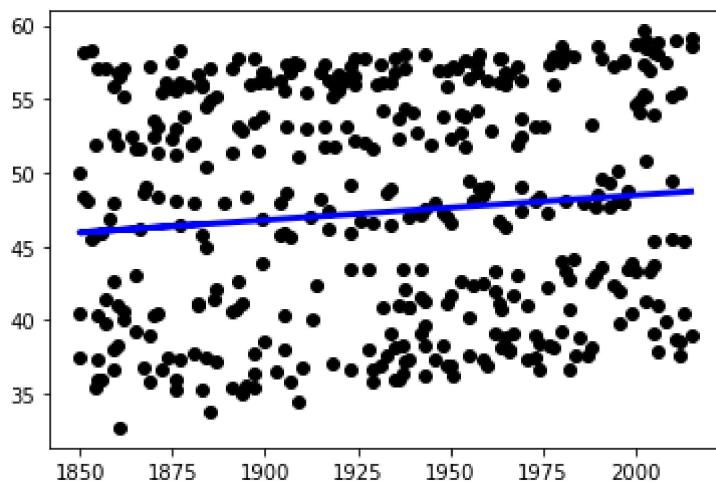
```
#Accuracy  
  
errors = abs(y_pred - y_train)  
mape = 100 * (errors / y_pred)  
accuracy = 100 - np.mean(mape)  
print("Model Accuracy ", round(accuracy, 2), "%")
```

Model Accuracy 85.47 %

In [34]:

```
#Linear regression plot  
  
plt.scatter(X_test,y_test, color="black")  
plt.plot(X_test,prediction2, color="blue", linewidth=3)
```

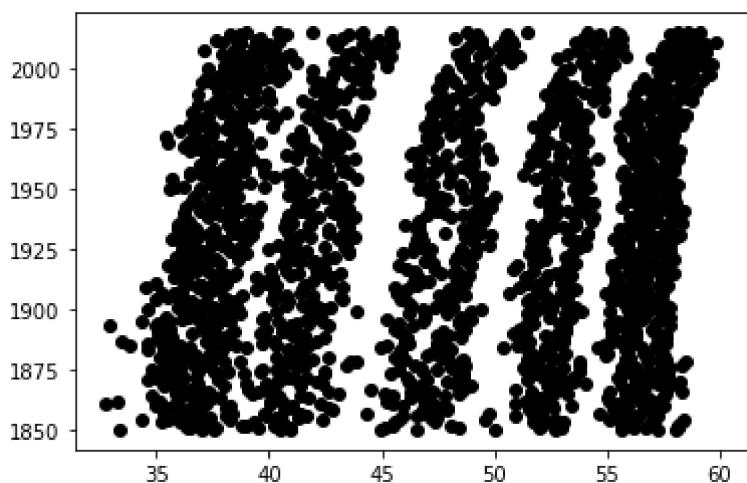
Out[34]:



In [41]:

```
plt.scatter(global_temperature['LandAverageTemperature'],global_temperature['year'],color='black')
```

Out[41]:



In []: