importing pandas

```
import pandas as pd
df=pd.read_csv("temperatures.csv")
```

display data

```
df
```

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	D
0	1901	22.40	24.14	29.07	31.91	33.41	33.18	31.21	30.39	30.47	29.97	27.31	24.
1	1902	24.93	26.58	29.77	31.78	33.73	32.91	30.92	30.73	29.80	29.12	26.31	24.
2	1903	23.44	25.03	27.83	31.39	32.91	33.00	31.34	29.98	29.85	29.04	26.08	23.
3	1904	22.50	24.73	28.21	32.02	32.64	32.07	30.36	30.09	30.04	29.20	26.36	23.
4	1905	22.00	22.83	26.68	30.01	33.32	33.25	31.44	30.68	30.12	30.67	27.52	23.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
112	2013	24.56	26.59	30.62	32.66	34.46	32.44	31.07	30.76	31.04	30.27	27.83	25.
113	2014	23.83	25.97	28.95	32.74	33.77	34.15	31.85	31.32	30.68	30.29	28.05	25.
114	2015	24.58	26.89	29.07	31.87	34.09	32.48	31.88	31.52	31.55	31.04	28.10	25.
115	2016	26.94	29.72	32.62	35.38	35.72	34.03	31.64	31.79	31.66	31.98	30.11	28.
116	2017	26.45	29.46	31.60	34.95	35.84	33.82	31.88	31.72	32.22	32.29	29.60	27.

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.head()
```

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
0	1901	22.40	24.14	29.07	31.91	33.41	33.18	31.21	30.39	30.47	29.97	27.31	24.49
1	1902	24.93	26.58	29.77	31.78	33.73	32.91	30.92	30.73	29.80	29.12	26.31	24.04
2	1903	23.44	25.03	27.83	31.39	32.91	33.00	31.34	29.98	29.85	29.04	26.08	23.65
3	1904	22.50	24.73	28.21	32.02	32.64	32.07	30.36	30.09	30.04	29.20	26.36	23.63

Next steps: Generate code with df | View recommended plots | New interactive sheet

display all columns

```
df.columns
```

```
Index(['YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP',
       'OCT', 'NOV', 'DEC', 'ANNUAL', 'JAN-FEB', 'MAR-MAY', 'JUN-SEP',
       'OCT-DEC'],
      dtype='object')
```

importing required libraries

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

initialize x and y coordinates

```
X = df[['YEAR']]
y = df[['JAN']]
```

import library for splitting data

```
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

split the data

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42
```

prepare model

```
model = LinearRegression()
model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

predict target variable

```
y_pred = model.predict(X_test)
```

calculate mse

```
mse = mean_squared_error(y_test, y_pred)
```

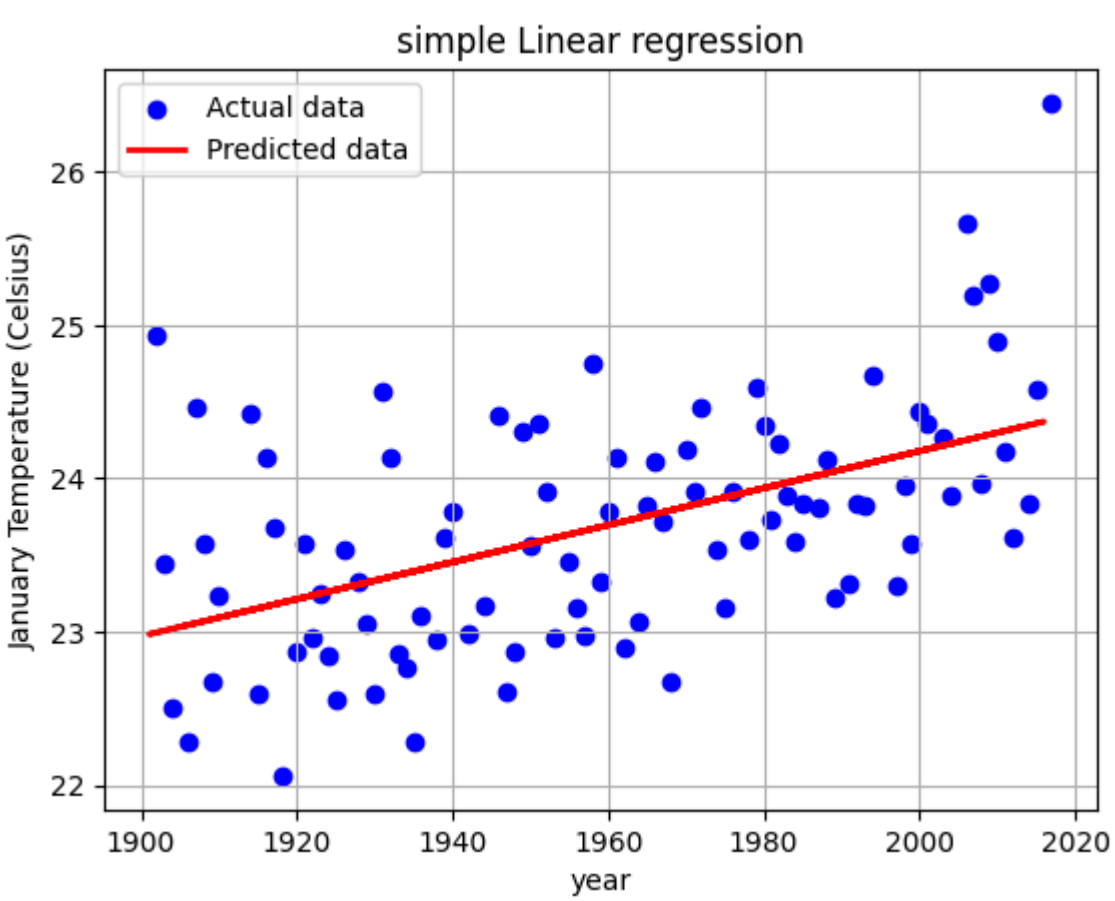
```
plt.figure(figsize=(10, 6))
```

```
<Figure size 1000x600 with 0 Axes>
<Figure size 1000x600 with 0 Axes>
```

```
import matplotlib.pyplot as plt
```

plot the graph

```
plt.scatter(X_train, y_train, color='blue', label='Actual data')
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Predicted data')
plt.title('simple Linear regression')
plt.xlabel('year')
plt.ylabel('January Temperature (Celsius)')
plt.legend()
plt.grid(True)
plt.show()
```




```
print(f'Mean Squared Error: {mse}')
```

```
Mean Squared Error: 0.5567563295289469
```

```
X = df[['YEAR']]
y = df[['MAY']]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=24
```

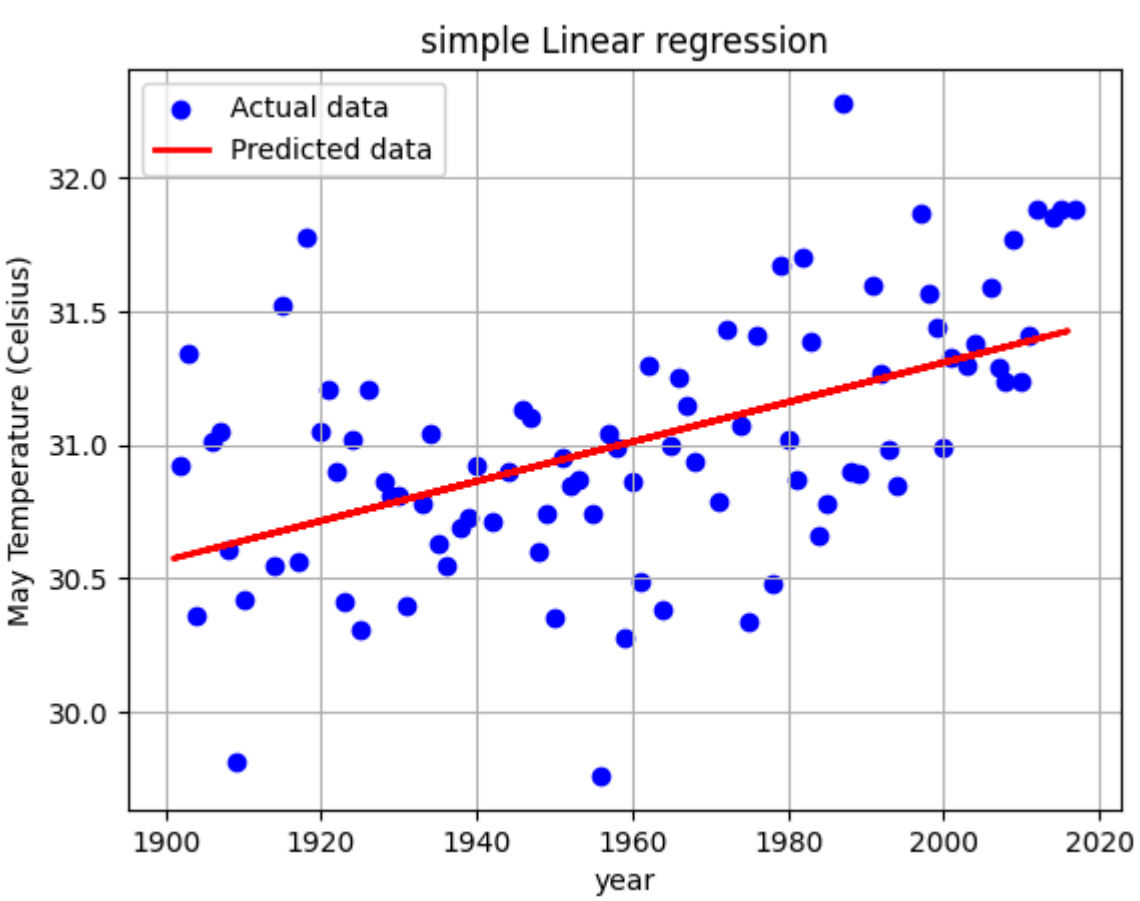
```
model = LinearRegression()
model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
y_pred = model.predict(X_test)
```

```
mse = mean_squared_error(y_test, y_pred)
```

```
plt.scatter(X_train, y_train, color='blue', label='Actual data')
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Predicted data')
plt.title('simple Linear regression')
plt.xlabel('year')
plt.ylabel('May Temperature (Celsius)')
plt.legend()
plt.grid(True)
plt.show()
```




```
print(mse)
```

```
0.47756077922995543
```

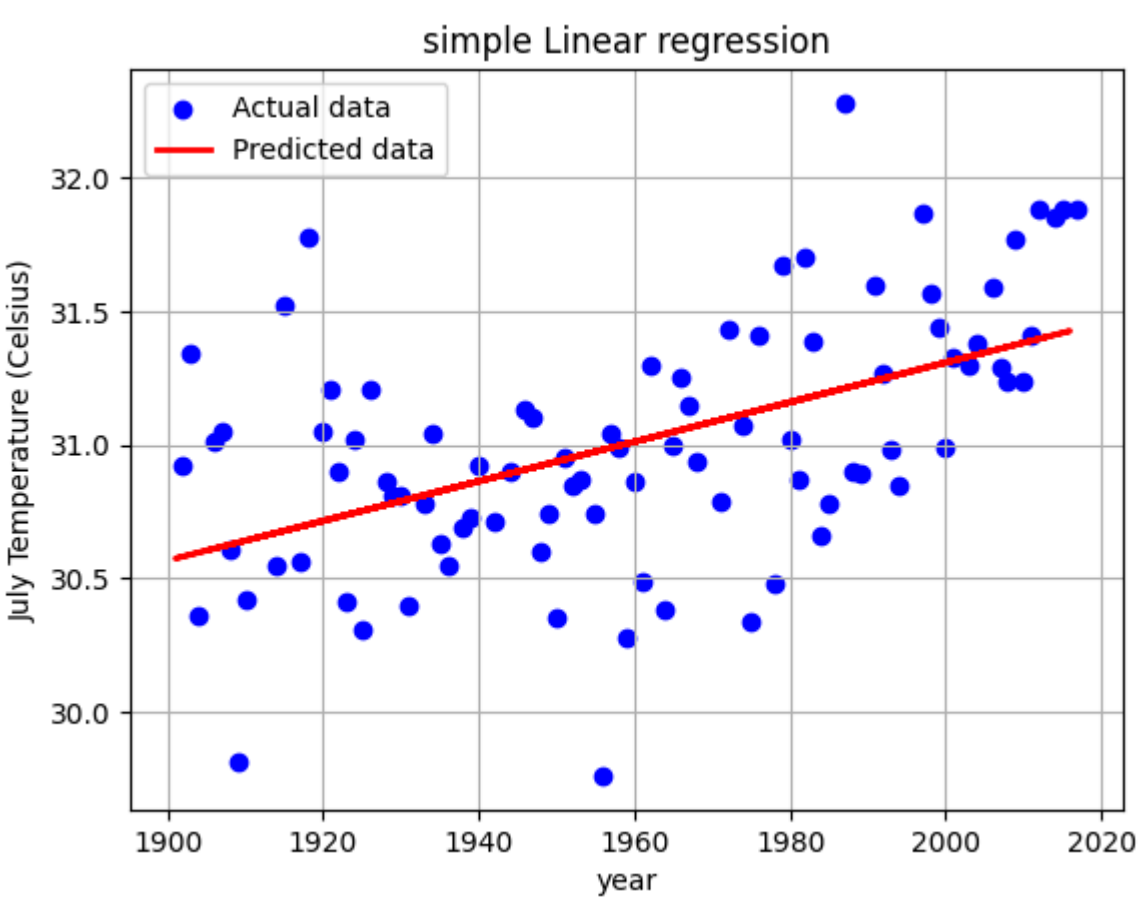
```
X = df[['YEAR']]
y = df[['JUL']]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=7
```

```
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

```
mse = mean_squared_error(y_test, y_pred)
```

```
plt.scatter(X_train, y_train, color='blue', label='Actual data')
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Predicted data')
plt.title('simple Linear regression')
plt.xlabel('year')
plt.ylabel('July Temperature (Celsius)')
plt.legend()
plt.grid(True)
plt.show()
```




```
print(mse)
```

```
0.20353474826878448
```

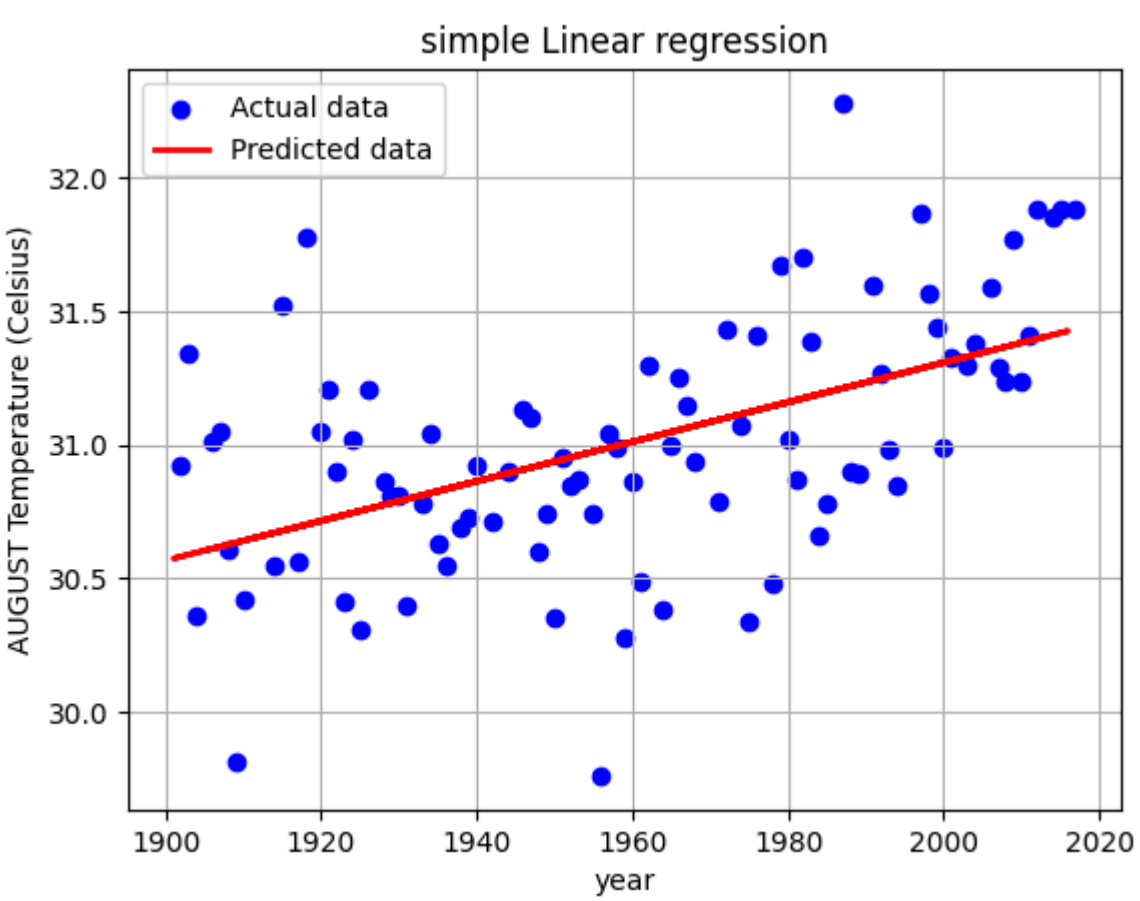
```
X = df[['YEAR']]
y = df[['AUG']]


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=27, random_state=42)


model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)


mse = mean_squared_error(y_test, y_pred)
```

```
plt.scatter(X_train, y_train, color='blue', label='Actual data')
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Predicted data')
plt.title('simple Linear regression')
plt.xlabel('year')
plt.ylabel('AUGUST Temperature (Celsius)')
plt.legend()
plt.grid(True)
plt.show()
```




```
print(mse)
```

```
0.19971785270280804
```

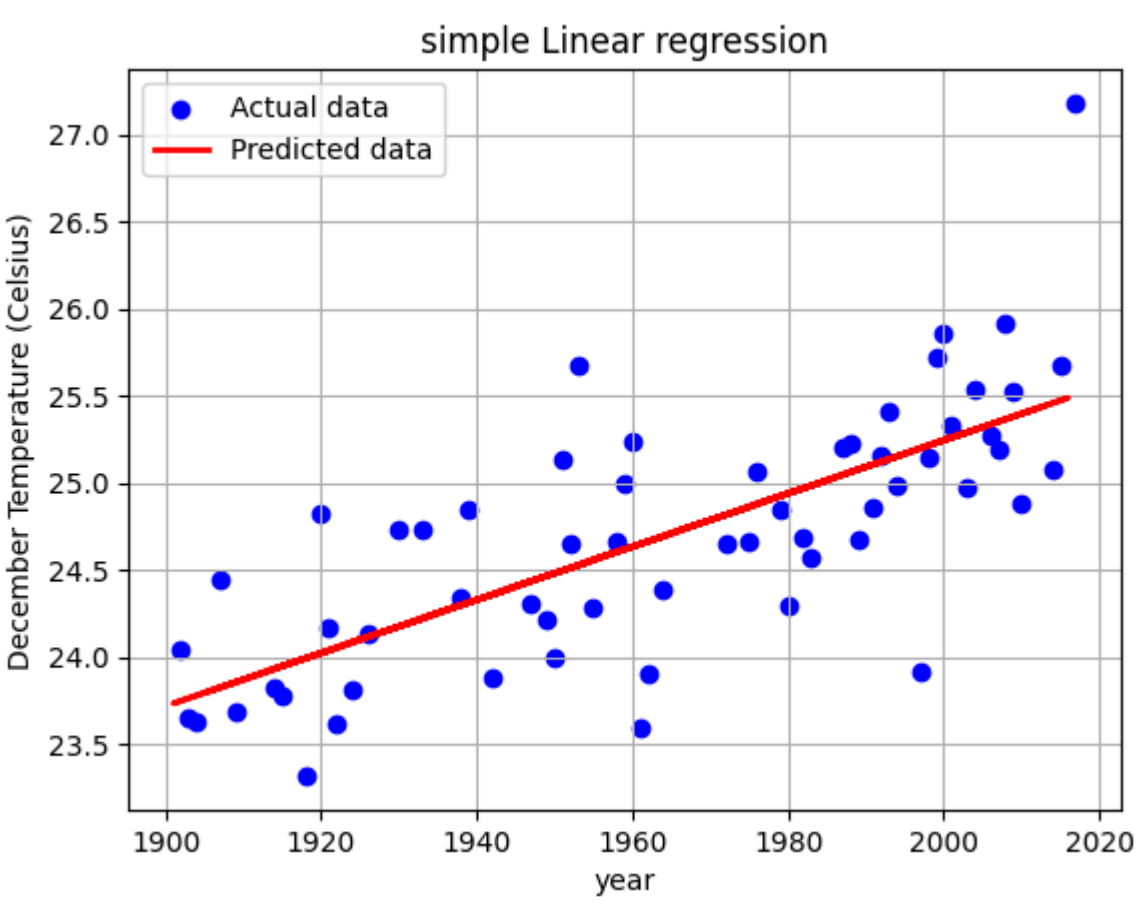
```
X = df[['YEAR']]
y = df[['DEC']]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=60, random_state=42
```

```
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

```
mse = mean_squared_error(y_test, y_pred)
```

```
plt.scatter(X_train, y_train, color='blue', label='Actual data')
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Predicted data')
plt.title('simple Linear regression')
plt.xlabel('year')
plt.ylabel('December Temperature (Celsius)')
plt.legend()
plt.grid(True)
plt.show()
```




```
print(mse)
```

```
0.3282960137590798
```

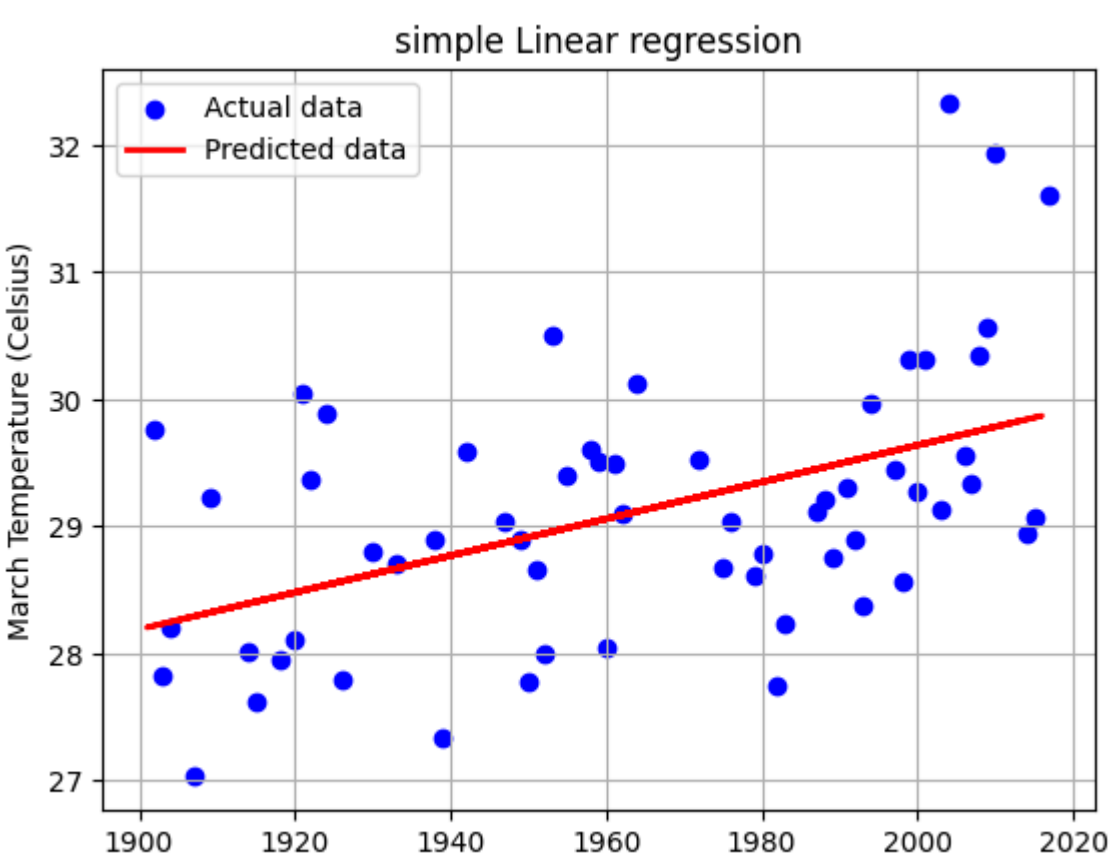
```
X = df[['YEAR']]
y = df[['MAR']]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=60, random_state=42
```

```
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

```
mse = mean_squared_error(y_test, y_pred)
```

```
plt.scatter(X_train, y_train, color='blue', label='Actual data')
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Predicted data')
plt.title('simple Linear regression')
plt.xlabel('year')
plt.ylabel('March Temperature (Celsius)')
plt.legend()
plt.grid(True)
plt.show()
```




```
print(mse)
```

```
0.7641028200376099
```