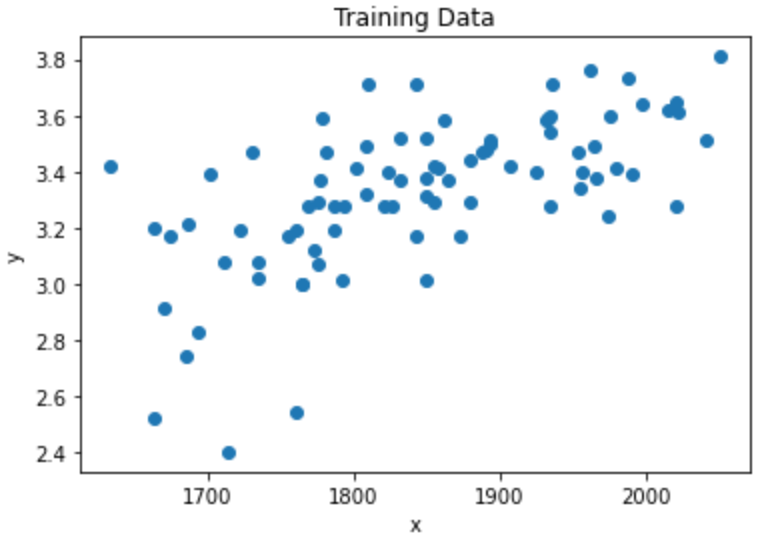
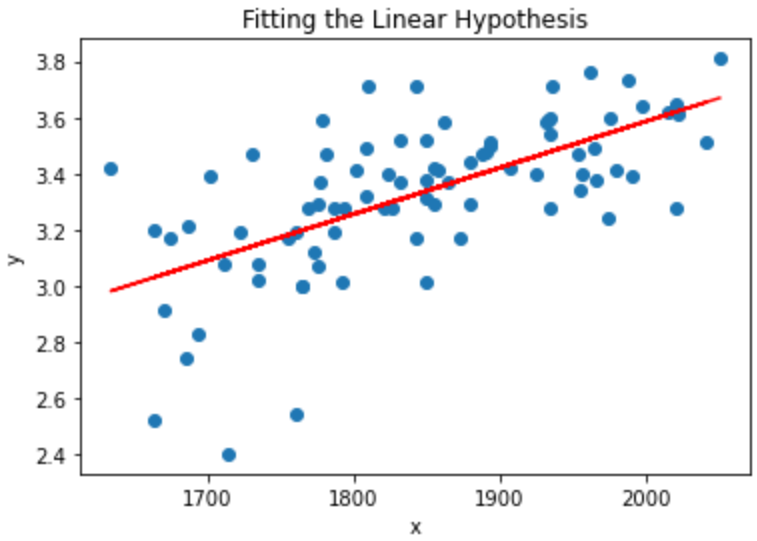
In [11]:

```
#Making the import
import numpy as np
from sklearn.linear_model import LinearRegression
import pandas as pd
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (6.0, 4.0)
from sklearn.metrics import mean_squared_error # for calculating mean_squared error
#Preprocessing Input data
data = pd.read_csv('1.01. Simple linear regression.csv')
X = data.iloc[:, 0]
Y = data.iloc[:, 1]
plt.scatter(X, Y)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Training Data')
plt.show()
X = np.array(X).reshape((-1, 1))

#Building the model
model = LinearRegression()
model.fit(X, Y)
print('theta0 = ', model.intercept_, 'theta1 = ', model.coef_)

#Making predictions for Training Data
# Making predictions for Training Data
Y_pred = model.predict(X)

#Computing mean_squared error
print('RMSE error for Regression=>',np.sqrt(mean_squared_error(Y,Y_pred)))
plt.scatter(X, Y)
plt.plot(X, Y_pred, color = 'red')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Fitting the Linear Hypothesis')
plt.show()

#Making predictions
test_X = 1.2
Y_pred = model.predict(np.array(test_X).reshape((-1, 1)))
#predict() is equivalent to below
Y_pred2 = model.intercept_ + model.coef_ * test_X
print("STA Prediction for Experience of ", str(test_X), " GPA =", Y_pred, Y_pred2)
```




```
theta0 =  0.2750402996602803 theta1 =  [0.00165569]
RMSE error for Regression=> 0.20808860291153056
```




```
STA Prediction for Experience of  1.2  GPA = [0.27702713] [0.27702713]
```

In [10]:
```
data
```

Out[10]:

	SAT	GPA
0	1714	2.40
1	1664	2.52
2	1760	2.54
3	1685	2.74
4	1693	2.83
...	...	...
79	1936	3.71
80	1810	3.71
81	1987	3.73
82	1962	3.76
83	2050	3.81

84 rows × 2 columns

In [ ]:
```

```