Summary of training Data

In [1]:
```
import pandas
from pandas import DataFrame
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

df = pandas.read_excel('run_report2.xlsx')
pace_filter = df['pace'] < 10
df = df[pace_filter]
df.describe()
```

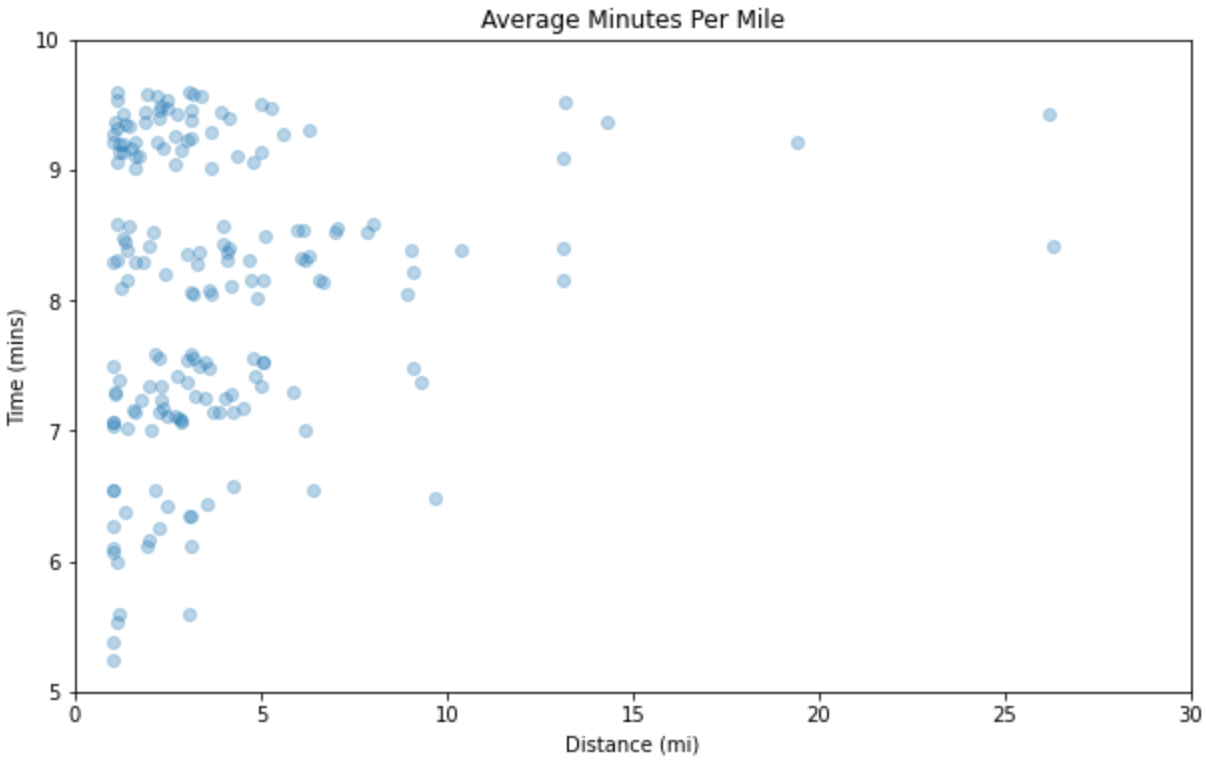
Out[1]:

	distance	pace
count	180.000000	180.000000
mean	3.850111	8.066000
std	3.740398	1.104205
min	1.000000	5.240000
25%	1.590000	7.247500
50%	3.015000	8.280000
75%	4.575000	9.142500
max	26.300000	9.590000

Plotting all training

In [2]:
```
X = DataFrame(df, columns=['distance'])
y = DataFrame(df, columns=['pace'])

plt.figure(figsize=(10,6))
plt.scatter(X, y, alpha=0.3)
plt.title('Average Minutes Per Mile')
plt.xlabel('Distance (mi)')
plt.ylabel('Time (mins)')
plt.ylim(5, 10)
plt.xlim(0, 30)
plt.show()
```



In [3]:
```
regression = LinearRegression()
regression.fit(X, y)
```

Out[3]:
```
LinearRegression()
```

Slope Coeffecient training data

In [4]:
```
regression.coef_ # theta_1
```

Out[4]:
```
array([[0.04833002]])
```

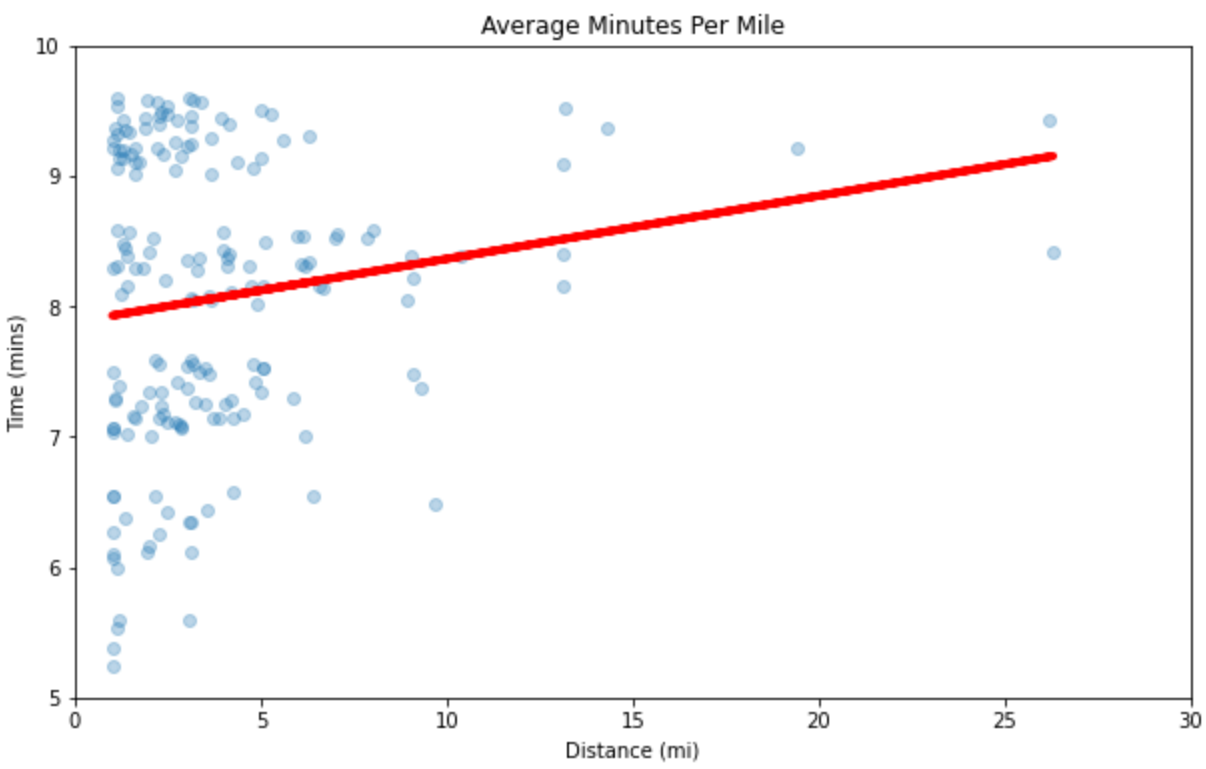
In [5]:
```
# Intercept
regression.intercept_
```

Out[5]:
```
array([7.87992406])
```

In [6]:
```
plt.figure(figsize=(10,6))
plt.scatter(X, y, alpha=0.3)

plt.plot(X, regression.predict(X), color='red', linewidth=4)

plt.title('Average Minutes Per Mile')
plt.xlabel('Distance (mi)')
plt.ylabel('Time (mins)')
plt.ylim(5, 10)
plt.xlim(0, 30)
plt.show()
```



In [7]:
```
#Getting r square from Regression
regression.score(X, y)
```

Out[7]:
```
0.026802195133993956
```

Data Summary from Races

In [8]:
```
df2 = pandas.read_excel('run_report_3.xlsx')
df2 = df2.drop(columns=['Unnamed: 2'])
df2.describe()
```

Out[8]:

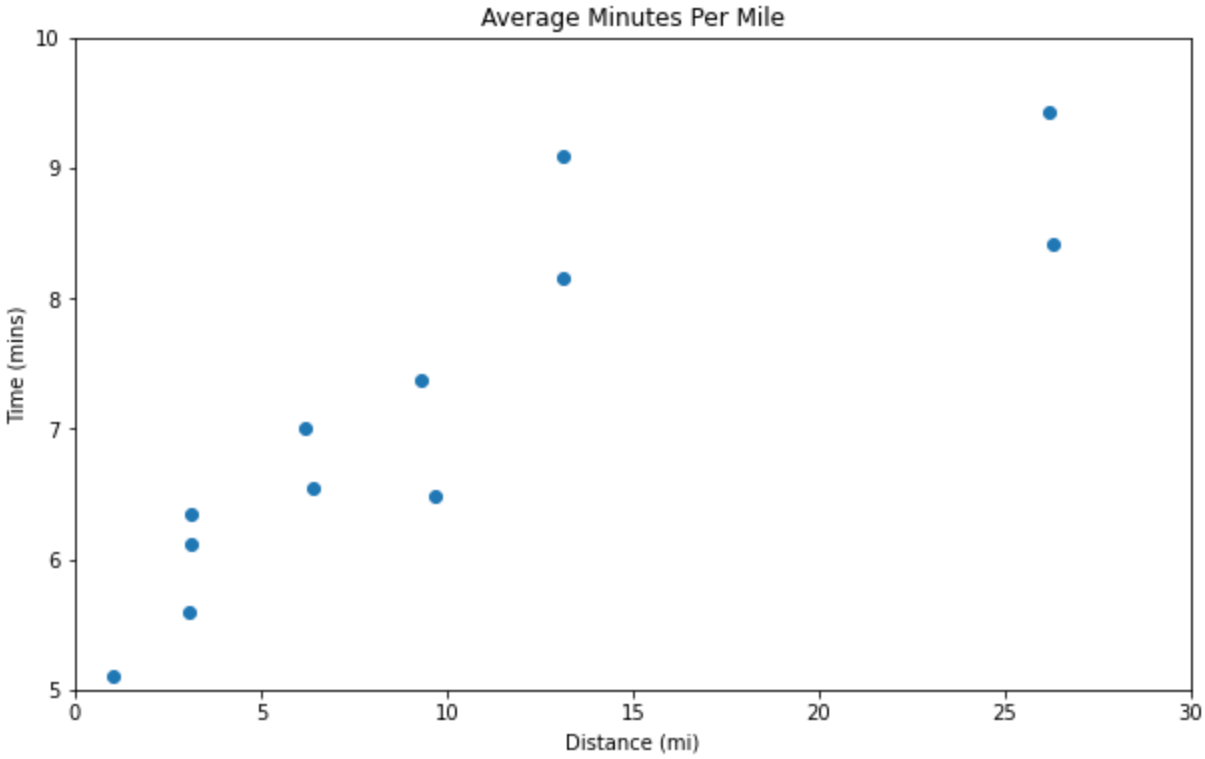
	distance	pace
count	12.000000	12.000000
mean	10.049167	7.137500
std	8.523352	1.374608
min	1.000000	5.110000
25%	3.110000	6.282500
50%	7.860000	6.775000
75%	13.100000	8.222500
max	26.300000	9.430000

Plotting Race Data

In [9]:
```
X2 = DataFrame(df2, columns=['distance'])
y2 = DataFrame(df2, columns=['pace'])

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

```
plt.scatter(X2, y2, alpha=1.0)
plt.title('Average Minutes Per Mile')
plt.xlabel('Distance (mi)')
plt.ylabel('Time (mins)')
plt.ylim(5, 10)
plt.xlim(0, 30)
plt.show()
```



In [10]:
```
regression2 = LinearRegression()
regression2.fit(X, y)

regression2.coef_ # theta_1
```

Out[10]:
```
array([[0.04833002]])
```

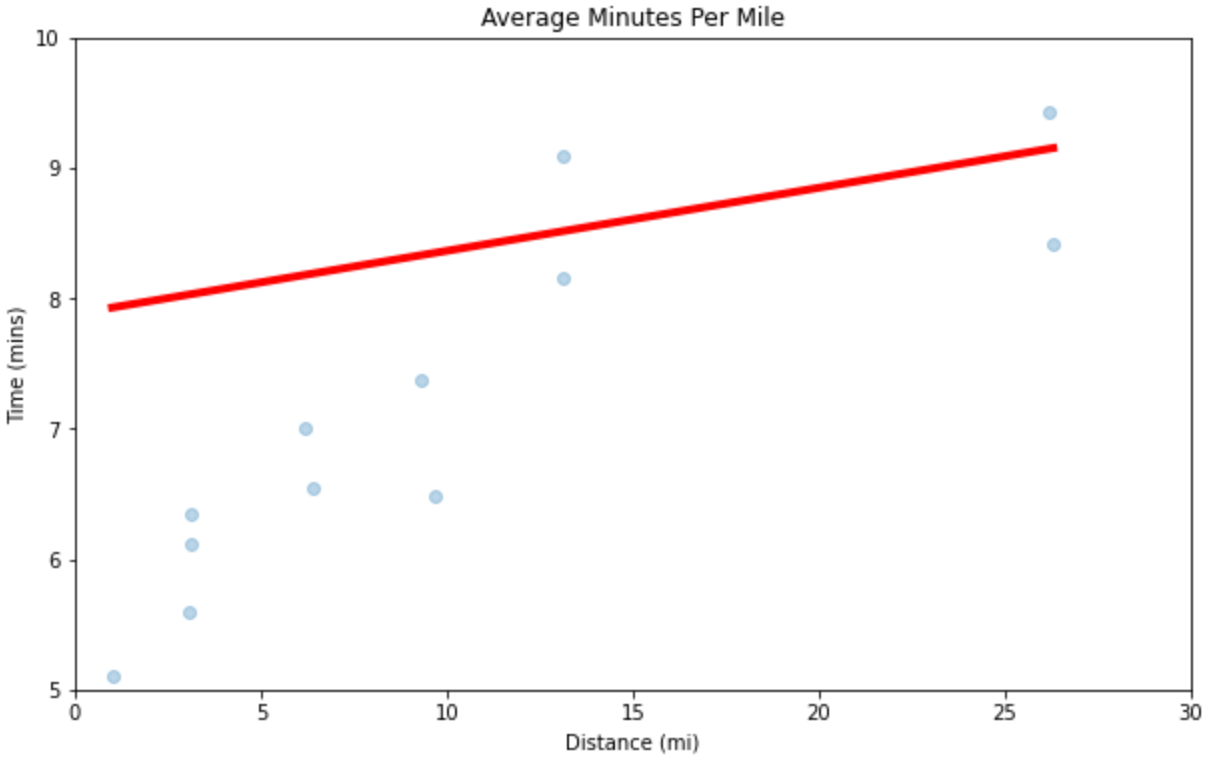
In [11]:
```
regression2.intercept_
```

Out[11]:
```
array([7.87992406])
```

In [12]:
```
plt.figure(figsize=(10,6))
plt.scatter(X2, y2, alpha=0.3)

plt.plot(X2, regression.predict(X2), color='red', linewidth=4)

plt.title('Average Minutes Per Mile')
plt.xlabel('Distance (mi)')
plt.ylabel('Time (mins)')
plt.ylim(5, 10)
plt.xlim(0, 30)
plt.show()
```


Average Minutes Per Mile
Time (mins)
Distance (mi)
10
9
8
7
6
5
0
5
10
15
20
25
30

In [ ]: