In [ ]:

```
# Maha Ebrahim mohammed al juhdali
# 4051350
# IA8G
# lab 10 : Linear Regression & Logistic Regression
```

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

In [3]:

```
from sklearn import datasets
dataset= datasets.load_iris()
df= pd.DataFrame(dataset['data'], columns=['petal length (cm)', 'petal width(cm)','sepal le
df
```

Out[3]:

	petal length (cm)	petal width(cm)	sepal length (cm)	sepal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

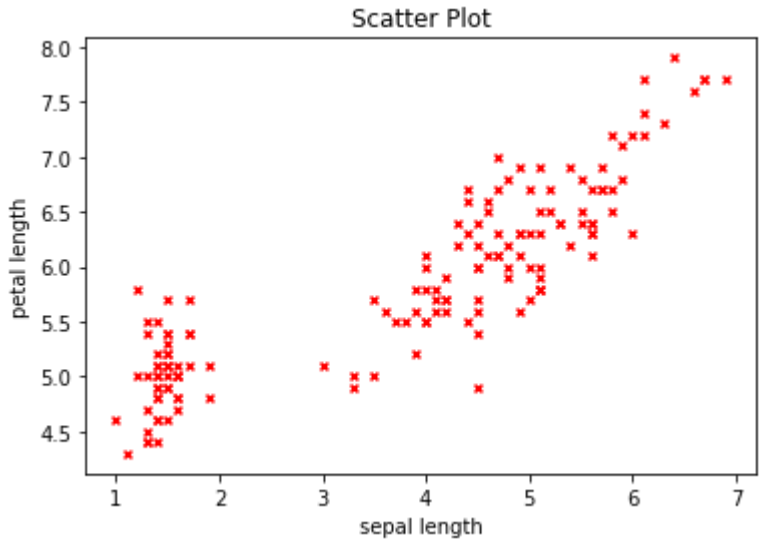
In [4]:

```
X = df['petal length (cm)']
Y = df['sepal length (cm)']
slic_df = pd.DataFrame({'petal length': X, 'sepal length': Y})
print(slic_df)
```

```
     petal length  sepal length
0             5.1           1.4
1             4.9           1.4
2             4.7           1.3
3             4.6           1.5
4             5.0           1.4
..            ...           ...
145           6.7           5.2
146           6.3           5.0
147           6.5           5.2
148           6.2           5.4
149           5.9           5.1

[150 rows x 2 columns]
```

In [5]:

```
plt.scatter(slic_df[['sepal length' ]], slic_df[['petal length']], color = "r", marker = "x
plt.xlabel('sepal length')
plt.ylabel('petal length')
plt.title('Scatter Plot')
plt.show()
```



In [6]:

```
from sklearn.linear_model import LinearRegression

classifier=LinearRegression()
model=classifier.fit(slic_df[['sepal length' ]], slic_df[['petal length']])
```

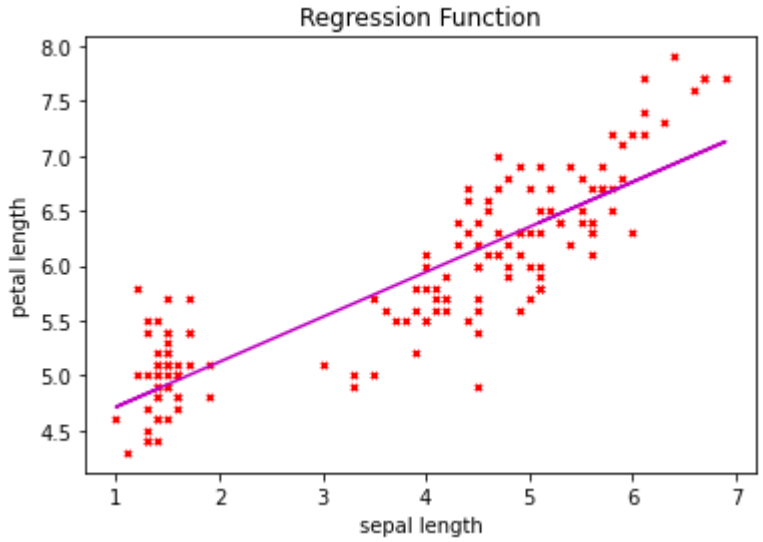
In [13]:

```
y_pred=classifier.predict(slic_df[['sepal length']])
print(y_pred)

print('Coefficients: \n',classifier.coef_)
print('Intercept: \n' ,classifier.intercept_)
```

```
 [6.59656817]
 [6.80102931]
 [6.59656817]
 [6.55567594]
 [6.26943035]
 [6.51478371]
 [6.59656817]
 [6.39210703]
 [6.39210703]
 [6.71924485]
 [6.6374604 ]
 [6.43299926]
 [6.3512148 ]
 [6.43299926]
 [6.51478371]
 [6.39210703]]
Coefficients:
 [[0.40892228]]
Intercept:
 [4.30660342]
```

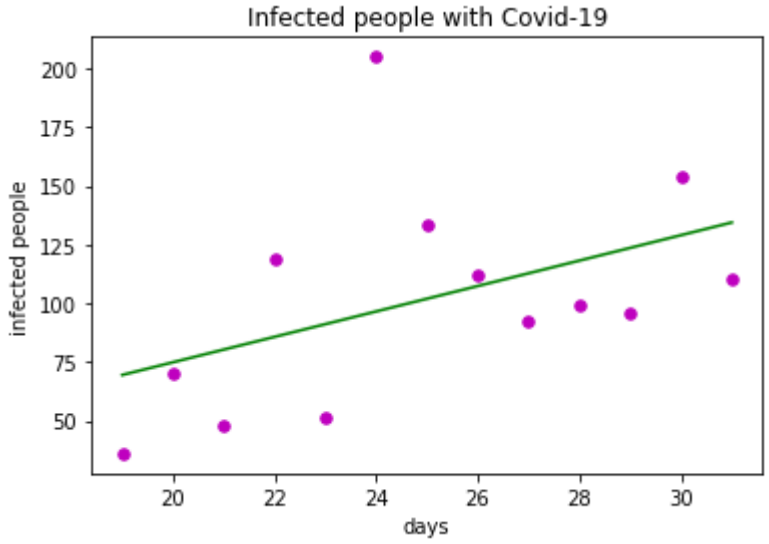
In [14]:

```
plt.scatter(slic_df[['sepal length' ]], slic_df[['petal length']], color = "r", marker = "x

plt.plot(slic_df['sepal length'], y_pred, color ="m")
plt.xlabel('sepal length')
plt.ylabel('petal length')
plt.title('Regression Function')
plt.show()
```



In [57]:

```
df2 = pd.DataFrame({'col1': [19, 20,21,22,23,24,25,26,27,28,29,30,31],
                    'col2': [36,70,48,119,51,205,133,112,92,99,96,154,110]})

classifier=LinearRegression()
classifier.fit(df2[['col1']], df2[['col2']])
y_predict = classifier.predict(df2[['col1']])

plt.scatter(df2[['col1']], df2[['col2']], color = "m", marker = 'o', s = 30)

plt.plot(df2[['col1']], y_predict, color= "g")
plt.xlabel('days')
plt.ylabel('infected people')
plt.title('Infected people with Covid-19')
plt.show()

print('Coefficients: \n',classifier.coef_)
print('Intercept: \n' ,classifier.intercept_)
```




```
Coefficients:
 [[5.41208791]]
Intercept:
 [-33.37912088]
```

In [62]:

```
Day_18 = classifier.intercept_ + classifier.coef_ *18
print(Day_18)
```

```
[[64.03846154]]
```

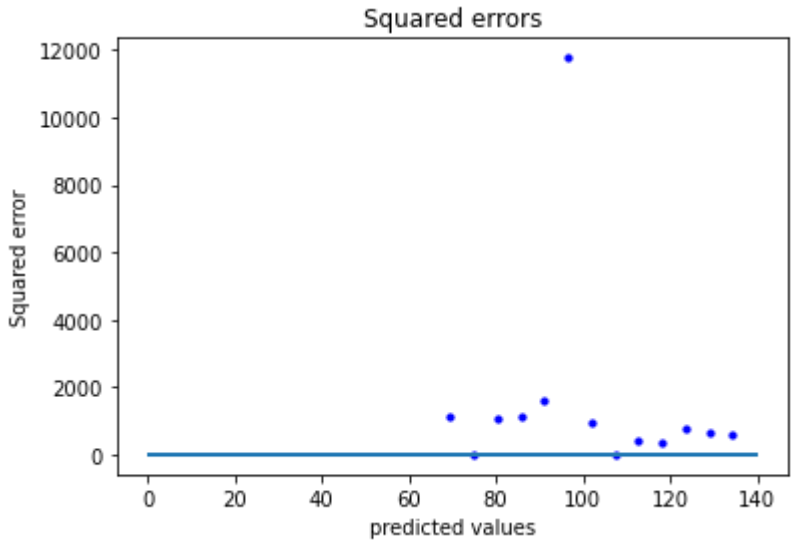
In [65]:

```
from sklearn.metrics import mean_squared_error, mean_absolute_error
print ("Mean squared error:")
print(mean_squared_error(df2[['col2']], y_predict))
```

```
Mean squared error:
1572.1551141166526
```

In [68]:

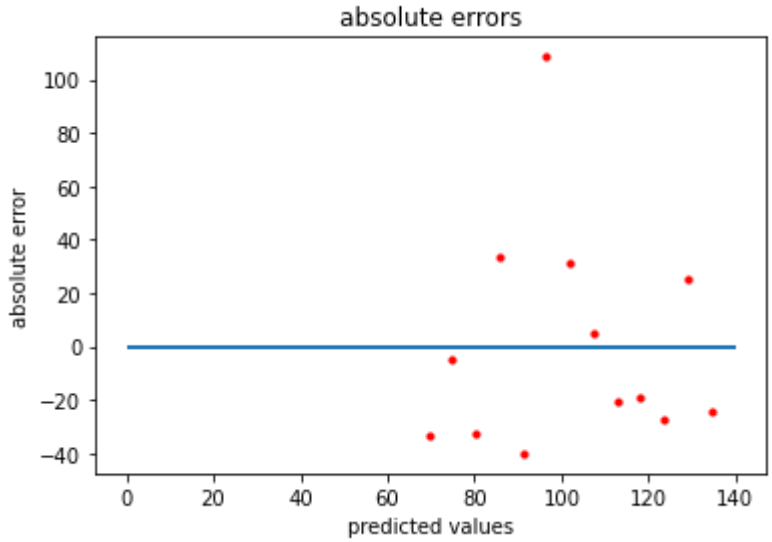
```
plt.scatter(y_predict, (df2[['col2' ]] - y_predict) ** 2, color = "blue", s=10)
plt.title('Squared errors')
plt.hlines(y = 0, xmin = 0, xmax = 140, linewidth = 2)
plt.xlabel('predicted values')
plt.ylabel('Squared error')
plt.show()
```



In [73]:

```
print("Mean absolute error: ")
print(mean_absolute_error(df2[['col2']], y_predict))

plt.scatter(y_predict, (df2[['col2' ]] - y_predict), color = "red", s=10)
plt.title('absolute errors')
plt.hlines(y = 0, xmin = 0, xmax = 140, linewidth = 2)
plt.xlabel('predicted values')
plt.ylabel('absolute error')
plt.show()
```

```
Mean absolute error:
31.163144547759927
```



In [9]:

```
df['Species' ]= dataset['target']
df['Species']=df['Species'].apply(lambda x:dataset['target_names'][x])
df['Species']
```

Out[9]:

```
0         setosa
1         setosa
2         setosa
3         setosa
4         setosa
         ...
145    virginica
146    virginica
147    virginica
148    virginica
149    virginica
Name: Species, Length: 150, dtype: object
```

In [10]:

```
from sklearn.linear_model import LogisticRegression
print ('All features: \n', df.columns.tolist())
X = df[['sepal length (cm)']]
y = df['Species']

print('X: \n', X.head(),'\n')
print('y: \n', y.head(),'\n')

classifier3 = LogisticRegression(solver='liblinear', multi_class = 'ovr')
classifier3.fit(X, y)
pred = classifier3.predict(X)

print('Score: \n', classifier3.score(X, y))
print('Coefficients: \n',classifier3.coef_)
print('Intercept: \n' ,classifier3.intercept_)
```

```
All features:
 ['petal length (cm)', 'petal width(cm)', 'sepal length (cm)', 'sepal width
(cm)', 'Species']
X:
    sepal length (cm)
0                 1.4
1                 1.4
2                 1.3
3                 1.5
4                 1.4

y:
 0    setosa
1    setosa
2    setosa
3    setosa
4    setosa
Name: Species, dtype: object

Score:
 0.7933333333333333
Coefficients:
 [[-1.72964826]
 [ 0.19387808]
 [ 0.98677372]]
Intercept:
 [ 4.28475916 -1.39893216 -4.70469008]
```

In [11]:

```
X = df.iloc[:, 0:4]
y = df['Species']

print('X: \n', X.head(),'\n')
print('y: \n', y.head(),'\n')

classifier3 = LogisticRegression(solver='liblinear', multi_class = 'ovr')
classifier3.fit(X, y)
pred = classifier3.predict(X)

print('Score: \n', classifier3.score(X, y))
print('Coefficients: \n',classifier3.coef_)
print('Intercept: \n' ,classifier3.intercept_)
```

```
X:
    petal length (cm)  petal width(cm)  sepal length (cm)  sepal width (cm)
0                 5.1              3.5                1.4               0.2
1                 4.9              3.0                1.4               0.2
2                 4.7              3.2                1.3               0.2
3                 4.6              3.1                1.5               0.2
4                 5.0              3.6                1.4               0.2

y:
 0    setosa
1    setosa
2    setosa
3    setosa
4    setosa
Name: Species, dtype: object

Score:
 0.96
Coefficients:
 [[ 0.41021713  1.46416217 -2.26003266 -1.02103509]
 [ 0.4275087  -1.61211605  0.5758173  -1.40617325]
 [-1.70751526 -1.53427768  2.47096755  2.55537041]]
Intercept:
 [ 0.26421853  1.09392467 -1.21470917]
```

In [ ]: