```
"""
fedhah Almutairi
4050392 , IA8G"""
```

In [1]:

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

In [78]:

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

Out[78]:

	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 [15]:

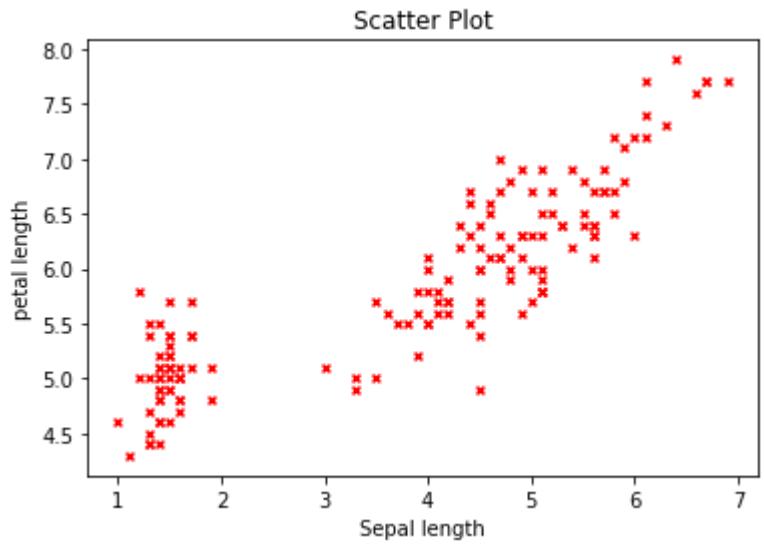
```
X = df['petal length (cm)']
Y = df['sepal length (cm)']
Slic_df= pd.DataFrame({'petal length':X,'sepal length':Y})
print(Slic_df)
```

```
     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 [20]:

```
plt.scatter(Slic_df[['sepal length']], Slic_df[['petal length']], color = "r", marke
plt.xlabel('Sepal length')
plt.ylabel('petal length')
plt.title('Scatter Plot')
plt.show()
```



In [22]:

```
from sklearn.linear_model import LinearRegression
classifier = LinearRegression()
model = classifier.fit(Slic_df[['sepal length']],Slic_df[['petal length']])
```

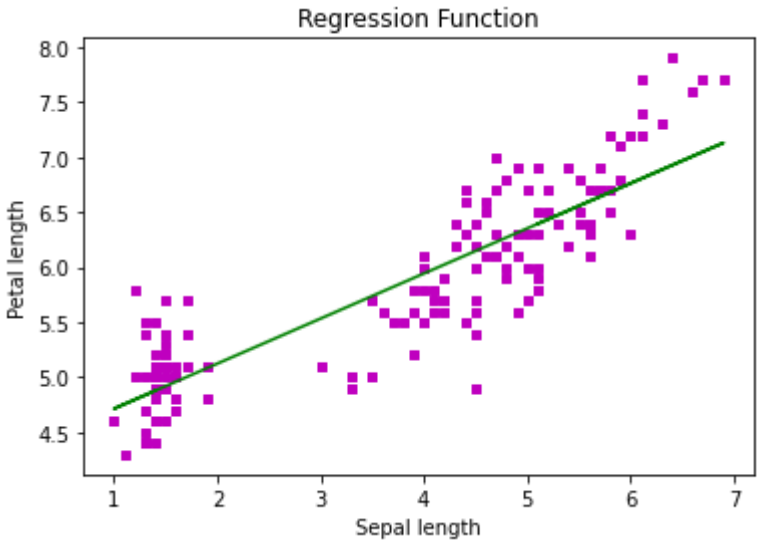
In [23]:

```
y_pred = classifier.predict(Slic_df[['sepal length']])
print(y_pred)
print('Coefficient: \n', classifier.coef_)
print('Intercept: \n', classifier.intercept_)
```

```
[[4.8790946 ]
 [4.8790946 ]
 [4.83820238]
 [4.91998683]
 [4.8790946 ]
 [5.00177129]
 [4.8790946 ]
 [4.91998683]
 [4.8790946 ]
 [4.91998683]
 [4.91998683]
 [4.96087906]
 [4.8790946 ]
 [4.75641792]
 [4.79731015]
 [4.91998683]
 [4.83820238]
 [4.8790946 ]
 [5.00177129]
```

In [24]:

```
plt.scatter(Slic_df[['sepal length']], Slic_df[['petal length']], color= 'm',
           marker='s',s=10)
plt.plot(Slic_df['sepal length'],y_pred, color='g')
plt.xlabel('Sepal length')
plt.ylabel('Petal length')
plt.title('Regression Function')
plt.show()
```



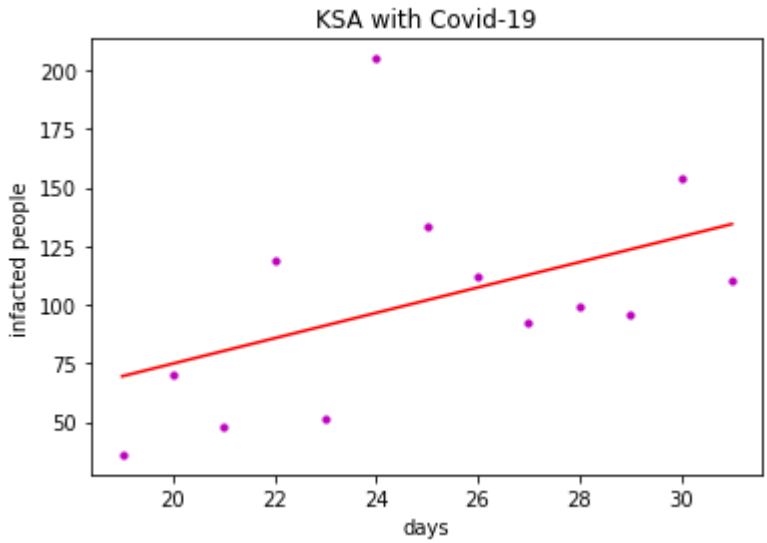
In [46]:

```
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()

model = classifier.fit(df2[['col1']],df2[['col2']])

y_predic = classifier.predict(df2[['col1']])
plt.scatter(df2[['col1']],df2[['col2']], color= 'm', marker= 'o', s=10)
plt.plot(df2['col1'],y_predic,color='r')
plt.xlabel('days')
plt.ylabel('infacted people')
plt.title('KSA with Covid-19')
plt.show()
print('coefficients: \n',classifier.coef_)
print('intercept: \n', classifier.intercept_)
```




```
coefficients:
 [[5.41208791]]
intercept:
 [-33.37912088]
```

In [48]:

```
day_18= classifier.intercept_+ classifier.coef_*18
print(day_18)
```

```
[[64.03846154]]
```

In [28]:

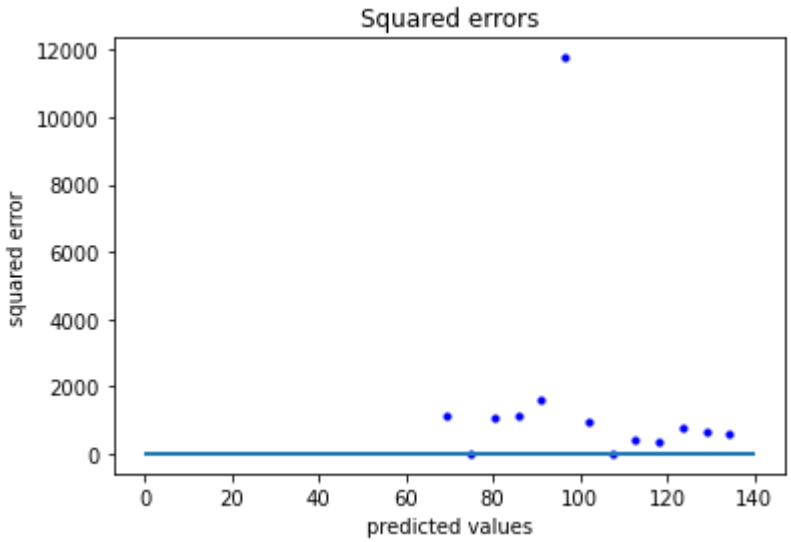
```
from sklearn.metrics import mean_squared_error, mean_absolute_error

print('Mean squared error:')
print(mean_squared_error(df2['col2'],y_predic))
```

```
Mean squared error:
1572.1551141166524
```

In [30]:

```
plt.scatter(y_predic, (df2[['col2']] - y_predic)**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()
```

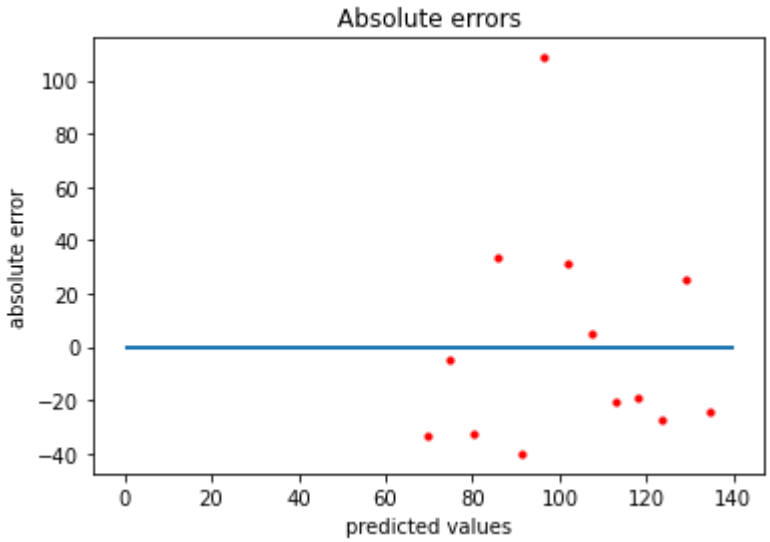


In [49]:

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

plt.scatter(y_predic, (df2[['col2']] - y_predic),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 [89]:

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

Out[89]:

```
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 [92]:

```
df
```

Out[92]:

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

150 rows × 5 columns

In [100]:

```
from sklearn.linear_model import LogisticRegression
print(df.columns.tolist())
z= df[['sepal length (cm)']]
v= df['species']
print('z:',z.head(),'\n')
print('v:',v.head(),'\n')

classifier3= LogisticRegression(solver = 'liblinear',multi_class='ovr')
classifier3.fit(z,v)
pred= classifier3.predict(z)

print('score:',classifier3.score(z,v))
print('Coefficient: ',classifier3.coef_)
print('inercept:',classifier3.intercept_)
```

```
['petal length (cm)', 'petal width (cm)', 'sepal length (cm)', 'sepal
width (cm)', 'species']
z:    sepal length (cm)
0                 1.4
1                 1.4
2                 1.3
3                 1.5
4                 1.4

v: 0    setosa
1    setosa
2    setosa
3    setosa
4    setosa
Name: species, dtype: object

score: 0.7933333333333333
Coefficient:  [[-1.72964826]
 [ 0.19387808]
 [ 0.98677372]]
inercept: [ 4.28475916 -1.39893216 -4.70469008]
```

In [101]:

```
print(df.columns.tolist(),'\n')
z= df.iloc[:,0:4]
v= df['species']
print('z:',z.head(),'\n')
print('v:',v.head(),'\n')
from sklearn.linear_model import LogisticRegression
classifier4 = LogisticRegression()
classifier4= LogisticRegression(solver = 'liblinear',multi_class='ovr')
classifier4.fit(z,v)
pred= classifier4.predict(z)

print('score:',classifier4.score(z,v))
print('Coefficient: ',classifier4.coef_)
print('inercept:',classifier4.intercept_)
```

```
['petal length (cm)', 'petal width (cm)', 'sepal length (cm)', 'sepal
width (cm)', 'species']

z:    petal length (cm)  petal width (cm)  sepal length (cm)  sepal wi
dth (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

v: 0    setosa
1    setosa
2    setosa
3    setosa
4    setosa
Name: species, dtype: object

score: 0.96
Coefficient:  [[ 0.41021713  1.46416217 -2.26003266 -1.02103509]
 [ 0.4275087  -1.61211605  0.5758173  -1.40617325]
 [-1.70751526 -1.53427768  2.47096755  2.55537041]]
inercept: [ 0.26421853  1.09392467 -1.21470917]
```

In [ ]: