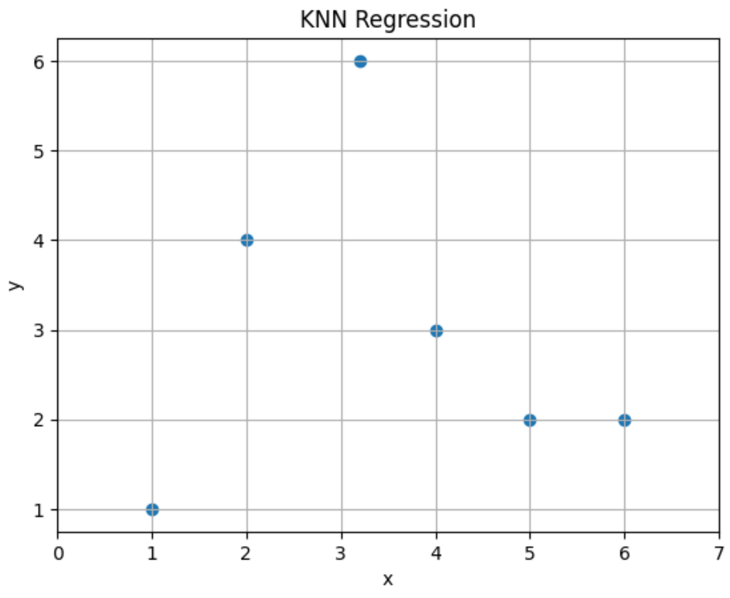
Question 1: Root Mean Squared Error for KNN

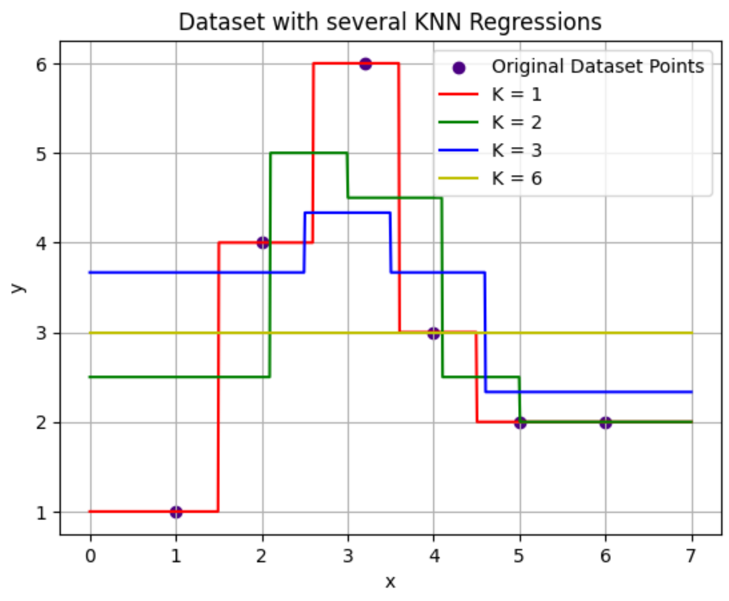
```
from matplotlib import pyplot as plt
from sklearn.neighbors import KNeighborsRegressor
import numpy as np


x = [[1], [2], [3.2], [4], [5], [6]]
y = [1, 4, 6, 3, 2, 2]
plt.scatter(x, y)
plt.grid()
plt.title("KNN Regression")
plt.xlabel("x")
plt.ylabel("y")
plt.xlim([0, 7])
plt.show()
```



```
k1 = KNeighborsRegressor(n_neighbors = 1)
k1.fit(x, y)
k2 = KNeighborsRegressor(n_neighbors = 2)
k2.fit(x, y)
k3 = KNeighborsRegressor(n_neighbors = 3)
k3.fit(x, y)
k6 = KNeighborsRegressor(n_neighbors =6)
k6.fit(x, y)
```

```
▾        KNeighborsRegressor
KNeighborsRegressor(n_neighbors=6)
```

```
pred_y = np.linspace(0, 7, 700)
k1list = []
k2list = []
k3list = []
k6list = []

for i in range (len(pred_y)):
  k1list.append(k1.predict([[pred_y[i]]]))
  k2list.append(k2.predict([[pred_y[i]]]))
  k3list.append(k3.predict([[pred_y[i]]]))
  k6list.append(k6.predict([[pred_y[i]]]))


plt.title("Dataset with several KNN Regressions")
plt.xlabel("x")
plt.ylabel("y")
plt.scatter(x, y, color = "indigo", label = "Original Dataset Points")
plt.plot(pred_y, k1list, label = "K = 1", color = "r")
plt.plot(pred_y, k2list, label = "K = 2", color = "g")
plt.plot(pred_y, k3list, label = "K = 3", color = "b")
plt.plot(pred_y, k6list, label = "K = 6", color = "y")
plt.legend()
plt.grid()
plt.show()
```



```
def rmse(x, y, largex, knn):
  sum = 0
  length = len(y)
  current = 0
  for i in range (len(largex)):
    if(current >= length):
      return np.sqrt(sum/length)
    if round(largex[i],2) == x[current]:
      sum += (y[current]-knn[i])**2
      current += 1

  return np.sqrt(sum/length)


x_original = [1, 2, 3.2, 4, 5, 6]
y_original = [1, 4, 6, 3, 2, 2]

x_new = [1.25, 3.4, 4.25]
y_new = [2, 5, 2.5]


arr = [k1list, k2list, k3list, k6list]

print("original data")
for k in arr:
  print(rmse(x_original, y_original, pred_y, k))

print("new data")
for k in arr:
  print(rmse(x_new, y_new, pred_y, k))
```

```
    original data
    [0.]
    [1.2416387]
    [1.33333333]
    [1.63299316]
    new data
    [0.8660254]
    [0.40824829]
    [1.23603308]
    [1.32287566]
```

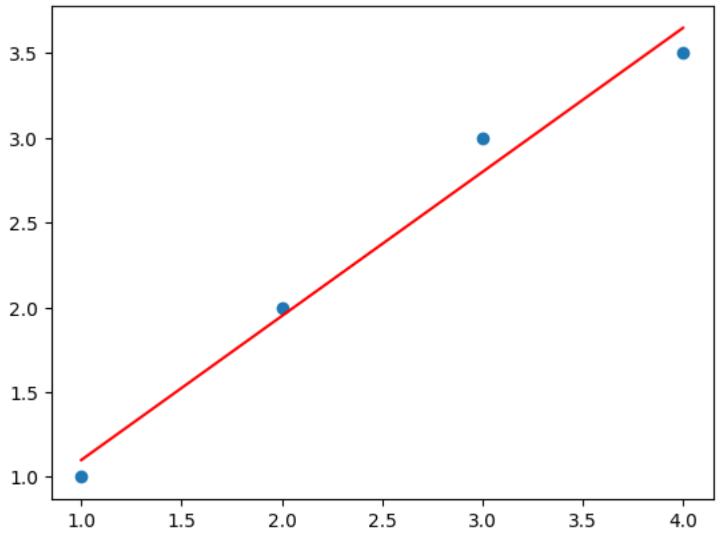
## Question 2. Mean Square Error

```
x = [1, 2, 3, 4]
y = [1, 2, 3, 3.5]


def b1(x, y):
  xavg = sum(x) / len(x)
  yavg = sum(y) / len(y)
  length = len(x)
  numerator = 0.0
  denominator = 0.0
  for i in range(length):
    numerator += (x[i]-xavg)*(y[i]-yavg)
    denominator += (x[i]-xavg)**2

  return numerator/denominator

def b0(x,y):
  xavg = sum(x) / len(x)
  yavg = sum(y) / len(y)
  return yavg - b1(x,y)*xavg
```

```
print(b1(x,y), b0(x,y))
```

```
    0.85 0.25
```

```
largex = np.linspace(1, 4, 300)
largey = largex * b1(x, y) + b0(x, y)
plt.scatter(x, y)
plt.plot(largex, largey, color = "red")
plt.show()
```



```
def r2(x, y, largex, largey):
  numerator = 0.0
  denominator = 0.0
  xavg = np.average(x)
  yavg = np.average(y)
  curr = 0

  for i in range(len(y)):
    denominator += (yavg - y[i])**2

  for j in range(len(largex)):
    if curr >= len(x):
      return 1-(numerator/denominator)
    if largex[j] == x[curr]:
      numerator += (largey[j]-y[curr])**2
      curr += 1

  return 1-(numerator/denominator)
```

```
print(r2(x,y,largex,largey))
```

```
    0.9972881355932204
```

✓ 0s completed at 10:02 AM