In [69]:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot
from sklearn.model_selection import KFold
```

In [70]:

```
df1 = pd.read_csv("Randomedi202.csv")
pd.set_option("display.max_columns", None)

df1
```

Out[70]:

	GaugeID	AnnuaL Mean Temperature	Drainage Area (km^2)	Max Temperature of Warmest Month	Maximal Flow Length	Slope of FDC
0	IWM-gauge-0100	22.307884	192.502640	29.887011	20429.95117	1.827450
1	IWM-gauge-0248	17.263815	213.932037	25.001167	22990.45508	5.978000
2	IWM-gauge-0387	25.653601	18450.912110	33.946518	619559.56250	5.678000
3	IWM-gauge-0636	23.128416	420.320160	34.606106	179310.10940	3.204780
4	IWM-gauge-0763	24.650948	5066.837402	33.433296	548838.00000	3.678942
5	IWM-gauge-0877	26.049732	350.433624	31.109547	171629.01560	4.600424
6	IWM-gauge-0880	24.212206	2306.040039	38.008038	409606.46880	3.070795
7	IWM-gauge-0908	23.686493	3779.519775	38.434139	634150.81250	5.693498
8	IWM-gauge-1061	25.385981	35702.257810	34.655426	157997.68750	3.923497
9	IWM-gauge-1089	24.259298	14968.109380	38.112904	297125.25000	3.245973
10	IWM-gauge-1169	24.121849	1942.562744	35.905399	141983.37500	3.407380
11	IWM-gauge-1442	25.424381	791.283936	36.807240	105572.35160	2.826344
12	IWM-gauge-1553	25.237150	541.287476	35.212067	131485.95310	3.326440
13	IWM-gauge-1602	24.768211	69253.421880	34.591045	128035.28130	7.556237
14	IWM-gauge-1642	24.924210	7870.645508	36.391060	233908.21880	6.817048
15	IWM-gauge-1784	25.415009	300.548340	33.123108	41151.16406	4.126189
16	IWM-gauge-2037	25.485306	15810.197270	35.523151	280266.00000	3.537800
17	IWM-gauge-2039	25.589821	35479.253910	34.859615	129727.49220	6.163512
18	IWM-gauge-2104	24.263506	12061.231450	41.233913	401571.68750	3.146119
19	IWM-gauge-2113	26.207943	8584.285156	35.699432	257194.57810	5.333939
20	IWM-gauge-2205	25.652811	4394.541504	42.526558	651244.93750	6.928748
21	IWM-gauge-2257	24.182590	993.997376	41.296951	515685.56250	4.964025
22	IWM-gauge-2293	27.012070	45774.484380	37.217041	148609.93750	5.985873

	GaugeID	AnnuaL Mean Temperature	Drainage Area (km^2)	Max Temperature of Warmest Month	Maximal Flow Length	Slope of FDC
23	IWM-gauge-2353	24.751366	2481.376465	42.006048	238973.25940	3.293072
24	IWM-gauge-2459	26.607559	55306.183590	34.648865	171939.62500	4.843241
25	IWM-gauge-2509	24.400270	2209.014648	33.581619	23044.41602	4.627342
26	IWM-gauge-2553	23.448601	2821.115967	34.769363	112966.04690	5.115323
27	IWM-gauge-2784	24.699221	6936.809570	34.170368	127708.65630	4.252300
28	IWM-gauge-2914	24.798716	9891.224609	33.098171	40775.80469	3.026108
29	IWM-gauge-2984	25.451561	9015.836914	33.078529	35003.85156	4.850712
30	IWM-gauge-3060	24.077030	52122.167970	32.179371	56171.39844	5.339107
31	IWM-gauge-3088	25.490677	66454.703130	35.103558	91831.54688	8.178789
32	IWM-gauge-3181	26.961117	15843.124020	32.562328	127295.21090	5.324360
33	IWM-gauge-3273	25.921089	6941.307617	35.507923	202064.34380	5.867786
34	IWM-gauge-3289	21.879065	574.843384	30.718559	41930.72656	4.406474
35	IWM-gauge-3333	24.137991	14475.828130	34.072067	299903.46880	3.715099
36	IWM-gauge-3369	24.777304	6891.486816	32.926483	104589.42970	4.791435
37	IWM-gauge-3643	25.242662	3266.582031	35.860931	576530.68750	3.983119
38	IWM-gauge-3744	25.401436	64073.714840	37.038273	46129.55859	5.076878
39	IWM-gauge-3812	22.698811	5579.921387	35.163727	210640.28130	1.068297
40	IWM-gauge-3825	24.740263	8494.139648	35.971535	640989.93750	6.124544

In [71]:

```
#Preparing data for training
X = df1.iloc[:, 1:5].values
y = df1.iloc[:, 5].values
```

In [72]:

```
y
```

Out[72]:

```
array([1.82745   , 5.978     , 5.678     , 3.20477954, 3.67894194,
       4.60042414, 3.07079512, 5.6934975 , 3.92349718, 3.24597329,
       3.40737951, 2.82634433, 3.32644029, 7.55623697, 6.81704797,
       4.12618936, 3.5378    , 6.16351157, 3.1461192 , 5.33393939,
       6.92874803, 4.96402504, 5.9858729 , 3.2930722 , 4.84324098,
       4.62734168, 5.1153229 , 4.2523    , 3.02610843, 4.85071183,
       5.33910707, 8.17878899, 5.32435972, 5.8677865 , 4.40647388,
       3.71509903, 4.79143506, 3.98311926, 5.07687786, 1.06829748,
       6.12454353])
```

In [73]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test= train_test_split(X,y, test_size = 0.20, random_state = 0)
```

In [74]:

```
#Feature scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)
```

In [75]:

```
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators =9, random_state = 0)
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)
```

In [76]:

```
y_pred
```

Out[76]:

```
array([4.45042577, 2.79367376, 4.63081856, 5.05845202, 3.80022316,
       6.82369107, 3.80599226, 5.41660342, 4.02980524])
```

In [77]:

```
df5 = pd.DataFrame({'Real Values':y_test, 'Predicted Values':y_pred})
df5.to_csv("SlopeofFDC1.csv")
df5
```

Out[77]:

	Real Values	Predicted Values
0	4.627342	4.450426
1	3.715099	2.793674
2	4.850712	4.630819
3	3.678942	5.058452
4	3.407380	3.800223
5	8.178789	6.823691
6	4.252300	3.805992
7	2.826344	5.416603
8	4.791435	4.029805

In [78]:

```
regressor.score(X_test,y_test)
```

Out[78]:

```
0.3554714647665742
```

In [79]:

```
a=metrics.mean_absolute_error(y_test,y_pred)
b=metrics.mean_squared_error(y_test,y_pred)
c=np.sqrt(metrics.mean_absolute_error(y_test,y_pred))
a,b,c
```

Out[79]:

```
(0.9159869732098765, 1.3678949028825274, 0.9570720836017925)
```

In [80]:

```
y_pred1 = regressor.predict(X_train)
```

In [81]:

```
y_pred1
```

Out[81]:

```
array([4.85709709, 3.37333311, 6.6497799 , 5.32997408, 5.53565191,
       5.01428602, 3.73439307, 4.04986577, 5.75377701, 3.73737104,
       4.50045337, 4.44353822, 6.15667586, 4.40617763, 5.42800968,
       5.48009406, 4.31380532, 5.59924348, 5.70051045, 4.04896083,
       4.52873715, 3.60979436, 5.13865489, 5.17199305, 3.3636157 ,
       4.13700041, 4.35440627, 5.1757303 , 3.55557883, 2.79959291,
       3.1931931 , 2.49083015])
```

In [82]:

```
df6 = pd.DataFrame({'Real Values':y_train, 'Predicted Values':y_pred1})
df6.to_csv("SlopeofFDC.csv")
df6
```

Out[82]:

	Real Values	Predicted Values
0	5.339107	4.857097
1	3.026108	3.373333
2	6.928748	6.649780
3	5.076878	5.329974
4	5.678000	5.535652
5	6.124544	5.014286
6	3.146119	3.734393
7	4.126189	4.049866
8	5.985873	5.753777
9	3.537800	3.737371
10	3.983119	4.500453
11	3.923497	4.443538
12	7.556237	6.156676
13	4.600424	4.406178
14	6.163512	5.428010
15	6.817048	5.480094
16	4.406474	4.313805
17	5.693498	5.599243
18	5.867786	5.700510
19	5.978000	4.048961
20	5.115323	4.528737
21	3.326440	3.609794
22	5.324360	5.138655
23	4.843241	5.171993
24	3.070795	3.363616
25	3.293072	4.137000
26	4.964025	4.354406
27	5.333939	5.175730
28	3.245973	3.555579
29	1.068297	2.799593
30	3.204780	3.193193
31	1.827450	2.490830

In [83]:

```
from sklearn import metrics
a1=metrics.mean_absolute_error(y_train,y_pred1)
b1=metrics.mean_squared_error(y_train,y_pred1)
c1=np.sqrt(metrics.mean_absolute_error(y_train,y_pred1))
a1,b1,c1
```

Out[83]:

(0.5219339465833333, 0.503685745846248, 0.7224499613006656)

In [84]:

```
regressor.score(X_train,y_train)
```

Out[84]:

0.7707066733679699

In [85]:

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import SelectFromModel
```

In [86]:

```
importance=regressor.feature_importances_
```

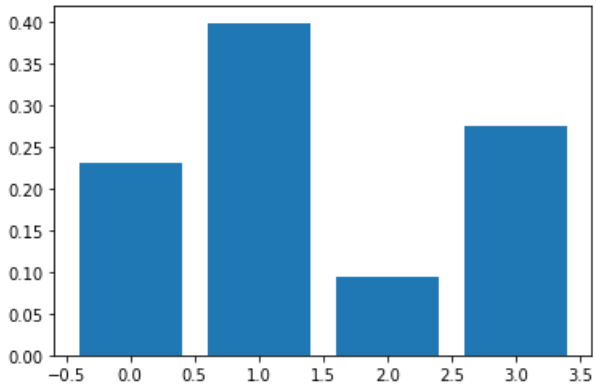
In [87]:

```
from sklearn.datasets import make_regression
from sklearn.ensemble import RandomForestRegressor
```

In [88]:

```
# summarize feature importance
for i,v in enumerate(importance):
    print('Feature: %0d, Score: %.5f' % (i,v))
# plot feature importance
pyplot.bar([x for x in range(len(importance))], importance)
pyplot.show()
```

Feature: 0, Score: 0.23130
Feature: 1, Score: 0.39906
Feature: 2, Score: 0.09443
Feature: 3, Score: 0.27521



In [89]:

```
#Correlation Matrix

corr = df1.corr()
```

```
corr.style.background_gradient(cmap='RdYlGn')
```

Out[89]:

	AnnuaL Mean Temperature	Drainage Area (km^2)	Max Temperature of Warmest Month	Maximal Flow Length	Slope of FDC
AnnuaL Mean Temperature	1	0.325187	0.430429	0.143433	0.200205
Drainage Area (km^2)	0.325187	1	0.0174227	-0.2223	0.483996
Max Temperature of Warmest Month	0.430429	0.0174227	1	0.544049	0.00676977
Maximal Flow Length	0.143433	-0.2223	0.544049	1	0.102959
Slope of FDC	0.200205	0.483996	0.00676977	0.102959	1

In [102]:

```
x = corr[['Slope of FDC']]


x.style.background_gradient(cmap='hsv')
```

Out[102]:

	Slope of FDC
AnnuaL Mean Temperature	0.200205
Drainage Area (km^2)	0.483996
Max Temperature of Warmest Month	0.00676977
Maximal Flow Length	0.102959
Slope of FDC	1

In [93]:

```
import seaborn as sns
```

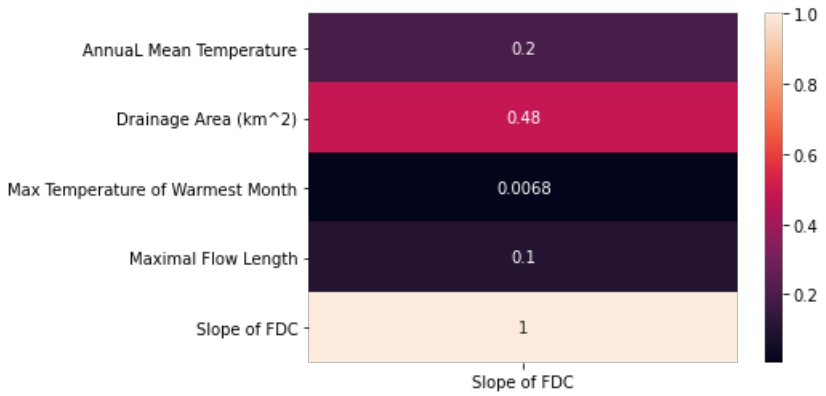
In [103]:

```
corr = df1.corr()

x = corr[['Slope of FDC']]
sns.heatmap(x,annot=True)
```

Out[103]:

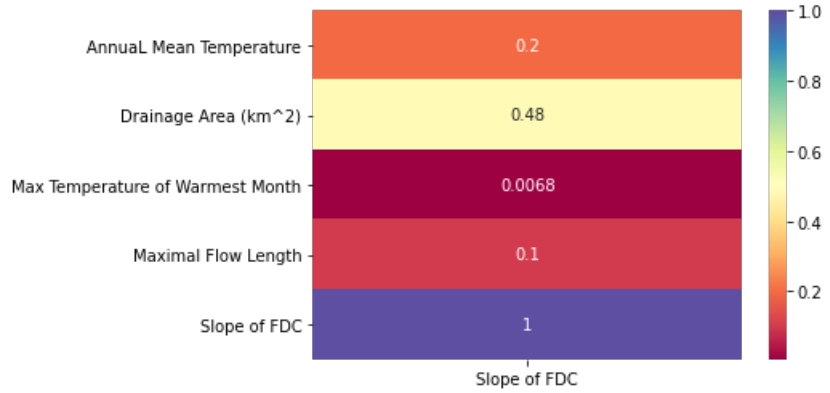
<matplotlib.axes._subplots.AxesSubplot at 0xc2c2e48>



In [122]:

```
corr = df1.corr()

x = corr[['Slope of FDC']]
sns.heatmap(x,annot=True,cmap="Spectral")
pyplot.savefig("aa5.png", dpi=400, bbox_inches='tight')
```



In [ ]:

```
```