In [45]:

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

In [46]:

```
df1 = pd.read_csv("Randomedi200.csv")
pd.set_option("display.max_columns", None)

df1
```

Out[46]:

	AnnuaL Mean Temperature	Precipitation Seasonality	Drainage Area (km^2)	Drainage Texture	Max Temperature of Warmest Month	Mean Annual Flow (m3/s)
0	22.307884	93.547569	192.502640	0.000087	29.887011	8.089800
1	17.263815	67.755791	213.932037	0.000065	25.001167	2.219200
2	25.653601	112.179329	18450.912110	0.000634	33.946518	274.760000
3	23.128416	158.140106	420.320160	0.000274	34.606106	510.265851
4	24.650948	116.591377	5066.837402	0.000580	33.433296	171.590008
5	26.049732	141.446732	350.433624	0.000232	31.109547	99.621960
6	24.212206	131.164093	2306.040039	0.000326	38.008038	285.111041
7	23.686493	138.930984	3779.519775	0.000440	38.434139	179.946829
8	25.385981	118.453384	35702.257810	0.000263	34.655426	884.470131
9	24.259298	115.949387	14968.109380	0.000361	38.112904	1155.533471
10	24.121849	131.046005	1942.562744	0.000220	35.905399	318.897775
11	25.424381	156.476242	791.283936	0.000142	36.807240	85.186700
12	25.237150	148.410629	541.287476	0.000218	35.212067	100.244678
13	24.768211	88.936455	69253.421880	0.000243	34.591045	300.008726
14	24.924210	98.838333	7870.645508	0.000265	36.391060	224.384353
15	25.415009	146.423004	300.548340	0.000035	33.123108	83.727390
16	25.485306	113.687416	15810.197270	0.000340	35.523151	535.794000
17	25.589821	134.401764	35479.253910	0.000129	34.859615	876.419578
18	24.263506	112.860062	12061.231450	0.000540	41.233913	799.351360
19	26.207943	79.652954	8584.285156	0.000360	35.699432	32.227258
20	25.652811	100.314293	4394.541504	0.000590	42.526558	161.717257
21	24.182590	154.482971	993.997376	0.000620	41.296951	236.560103
22	27.012070	126.736610	45774.484380	0.000294	37.217041	805.550047
23	24.751366	86.637672	2481.376465	0.000414	42.006042	19.927719
24	26.607559	124.961136	55306.183590	0.000246	34.648865	752.731794
25	24.400270	116.571381	2209.014648	0.000074	33.581619	63.086965
26	23.448601	136.135284	2821.115967	0.000119	34.769363	340.939798
27	24.699221	120.672516	6936.809570	0.000156	34.170368	269.255000
28	24.798716	96.240807	9891.224609	0.000116	33.098171	48.032168
29	25.451561	97.130249	9015.836914	0.000040	33.078529	121.885452
30	24.077030	91.857086	52122.167970	0.000156	32.179371	319.882146
31	25.490677	97.686272	66454.703130	0.000189	35.103558	302.238955
32	26.961117	124.130837	15843.124020	0.000176	32.562328	167.446797
33	25.921089	97.874069	6941.307617	0.000247	35.507923	121.528219
34	21.879065	141.480133	574.843384	0.000129	30.718559	62.144237

	Annual Mean Temperature	Precipitation Seasonality	Drainage Area (km^2)	Drainage Texture	Max Temperature of Warmest Month	Mean Annual Flow (m3/s)
35	24.137991	107.703888	14475.828130	0.000380	34.072067	457.387633
36	24.777304	99.339409	6891.486816	0.000174	32.926483	136.856902
37	25.242662	137.147522	3266.582031	0.000762	35.860931	718.753178
38	25.401436	97.490074	64073.714840	0.000126	37.038273	324.559416
39	22.698811	80.948708	5579.921387	0.000252	35.163727	23.735789
40	24.740263	109.545830	8494.139648	0.000749	35.971535	268.876866

In [47]:

```
#Preparing data for training
X = df1.iloc[:, 0:5].values
y = df1.iloc[:, 5].values
```

In [48]:

```
y
```

Out[48]:

```
array([   8.0898    ,    2.2192    ,  274.76      ,  510.2658515 ,
        171.5900081 ,   99.6219603 ,  285.1110414 ,  179.9468294 ,
        884.470131  , 1155.533471  ,  318.8977745 ,   85.1866996 ,
        100.2446782 ,  300.0087258 ,  224.3843529 ,   83.72738969,
        535.794     ,  876.4195783 ,  799.3513597 ,   32.22725828,
        161.7172566 ,  236.5601029 ,  805.5500465 ,   19.92771889,
        752.7317945 ,   63.08696516,  340.9397982 ,  269.255     ,
         48.03216765,  121.8854523 ,  319.8821463 ,  302.238955  ,
        167.4467966 ,  121.5282188 ,   62.14423729,  457.3876331 ,
        136.8569022 ,  718.7531779 ,  324.5594165 ,   23.73578949,
        268.8768656 ])
```

In [49]:

```
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 [50]:

```
kf = KFold(n_splits=2, random_state = 0)
kf.get_n_splits(X)
```

Out[50]:

```
2
```

In [51]:

```
kf
```

Out[51]:

```
KFold(n_splits=2, random_state=0, shuffle=False)
```

In [52]:

```
for train_index, test_index in kf.split(X):
    print("TRAIN:", train_index, "TEST:", test_index)
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]
```

```
TRAIN: [21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40] TEST: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 1
9 20]
TRAIN: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20] TEST: [21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 3
9 40]
```

In [53]:

In [53]:

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

In [54]:

```
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators =10, random_state = 0)
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)
```

In [55]:

```
y_pred
```

Out[55]:

```
array([311.06951073, 892.02863045, 187.49704741, 701.90681376,
       110.75410277, 160.97170792, 224.89120042, 135.19036563,
       109.48353263, 381.73329968, 338.95519519, 276.43393712,
       192.49997404, 189.08509616, 240.56335486, 153.98028044,
       187.20896247, 424.50766971, 162.77456193, 178.63358714])
```

In [56]:

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

df5.to_csv("Annualflow2.csv")
df5
```

Out[56]:

	Real Values	Predicted Values
0	236.560103	311.069511
1	805.550047	892.028630
2	19.927719	187.497047
3	752.731794	701.906814
4	63.086965	110.754103
5	340.939798	160.971708
6	269.255000	224.891200
7	48.032168	135.190366
8	121.885452	109.483533
9	319.882146	381.733300
10	302.238955	338.955195
11	167.446797	276.433937
12	121.528219	192.499974
13	62.144237	189.085096
14	457.387633	240.563355
15	136.856902	153.980280
16	718.753178	187.208962
17	324.559416	424.507670
18	23.735789	162.774562
19	268.876866	178.633587

In [57]:

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

Out[57]:

0.5506270929065115

In [58]:

```
from sklearn import metrics
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[58]:

(112.55653851894999, 24679.268034681, 10.609266634360266)

In [59]:

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

In [60]:

```
y_pred1
```

Out[60]:

```
array([  37.99647745,   37.47492328,  415.0441471 ,  332.60337862,
        157.80582878,   91.88287778,  247.46443813,  194.88233213,
        730.29507105, 1155.533471  ,  251.79604775,  214.21624303,
        117.38439147,  373.52151661,  165.94655717,   87.11450659,
        578.62077093,  727.07484997,  755.93585699,  146.12081733,
        195.47295562])
```

In [32]:

```
df6 = pd.DataFrame({'Real Values':y_train, 'Predicted Values':y_pred1})
df6.to_csv("Annualflow1.csv")
df6
```

Out[32]:

	Real Values	Predicted Values
0	8.089800	37.996477
1	2.219200	37.474923
2	274.760000	415.044147
3	510.265851	332.603379
4	171.590008	157.805829
5	99.621960	91.882878
6	285.111041	247.464438
7	179.946829	194.882332
8	884.470131	730.295071
9	1155.533471	1155.533471
10	318.897775	251.796048
11	85.186700	214.216243
12	100.244678	117.384391
13	300.008726	373.521517
14	224.384353	165.946557
15	83.727390	87.114507
16	535.794000	578.620771
17	876.419578	727.074850

	Real Values	Predicted Values
18	799.351360	755.935857
19	32.227258	146.120817
20	161.717257	195.472956

In [61]:

```
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[61]:

(63.96354263980955, 7116.706822088488, 7.9977210903987865)

In [62]:

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

Out[62]:

0.9320923424863184

In [63]:

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

In [64]:

```
importance=regressor.feature_importances_
```

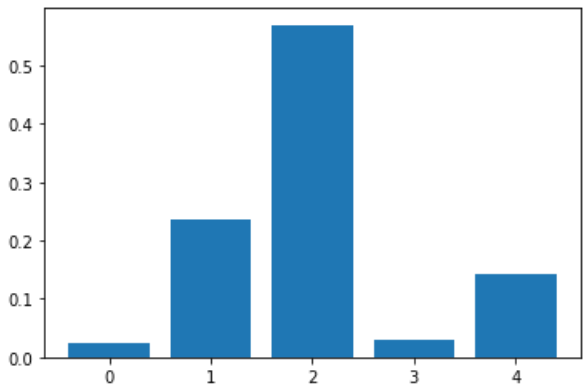
In [65]:

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

In [66]:

```
# 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: 0.02303
Feature: 1, Score: 0.23538
Feature: 2, Score: 0.56985
Feature: 3, Score: 0.02961
Feature: 4, Score: 0.14214



In [44]:

```
#Correlation Matrix

corr = df1.corr()
corr.style.background_gradient(cmap='YlGnBu')
```

Out[44]:

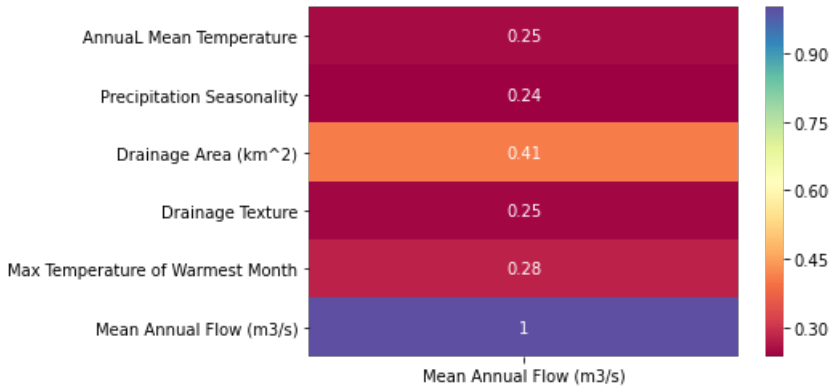
	AnnuaL Mean Temperature	Precipitation Seasonality	Drainage Area (km^2)	Drainage Texture	Max Temperature of Warmest Month	Mean Annual Flow (m3/s)
AnnuaL Mean Temperature	1	0.216688	0.325187	0.178263	0.430429	0.249977
Precipitation Seasonality	0.216688	1	-0.262082	0.0888714	0.175863	0.237923
Drainage Area (km^2)	0.325187	-0.262082	1	-0.138631	0.0174227	0.408584
Drainage Texture	0.178263	0.0888714	-0.138631	1	0.549694	0.245898
Max Temperature of Warmest Month	0.430429	0.175863	0.0174227	0.549694	1	0.282174
Mean Annual Flow (m3/s)	0.249977	0.237923	0.408584	0.245898	0.282174	1

In [67]:

```
import seaborn as sns
```

In [75]:

```
x = corr[['Mean Annual Flow (m3/s)']]
y=sns.heatmap(x,annot=True,cmap="Spectral")
y.plot()
pyplot.savefig("aa2.png", dpi=400, bbox_inches='tight')
```



In [ ]:

In [ ]: