6.5.2 Forward and Backward Stepwise Selection

May 18, 2018

```
In [44]: # conventional way to import pandas
         import pandas as pd
         # conventional way to import seaborn
         import seaborn as sns
         # conventional way to import numpy
         import numpy as np

         from sklearn import metrics
         import matplotlib.pyplot as plt

         data = pd.read_csv("https://vincentarelbundock.github.io/Rdatasets/csv/ISLR/Hitters.cs

         data.head()
```

```
Out[44]:                    AtBat  Hits  HmRun  Runs  RBI  Walks  Years  CAtBat  CHits  \
         -Andy Allanson       293    66      1    30   29     14      1     293     66
         -Alan Ashby          315    81      7    24   38     39     14    3449    835
         -Alvin Davis         479   130     18    66   72     76      3    1624    457
         -Andre Dawson        496   141     20    65   78     37     11    5628   1575
         -Andres Galarraga    321    87     10    39   42     30      2     396    101

                            CHmRun  CRuns  CRBI  CWalks League Division  PutOuts  \
         -Andy Allanson          1     30    29      14      A        E      446
         -Alan Ashby            69    321   414     375      N        W      632
         -Alvin Davis           63    224   266     263      A        W      880
         -Andre Dawson         225    828   838     354      N        E      200
         -Andres Galarraga      12     48    46      33      N        E      805

                            Assists  Errors  Salary NewLeague
         -Andy Allanson          33      20     NaN         A
         -Alan Ashby             43      10   475.0         N
         -Alvin Davis            82      14   480.0         A
         -Andre Dawson           11       3   500.0         N
         -Andres Galarraga       40       4    91.5         N
```

After listing the data we can see that some have missing data for their Salary. Next drop all the rows that contain NaN data.

```
In [45]: data = data.dropna()
         data.index.name = 'Player'
         data.head()

Out[45]:                     AtBat  Hits  HmRun  Runs  RBI  Walks  Years  CAtBat  CHits  \
         Player
         -Alan Ashby           315    81      7    24   38     39     14    3449    835
         -Alvin Davis          479   130     18    66   72     76      3    1624    457
         -Andre Dawson         496   141     20    65   78     37     11    5628   1575
         -Andres Galarraga     321    87     10    39   42     30      2     396    101
         -Alfredo Griffin      594   169      4    74   51     35     11    4408   1133

                             CHmRun  CRuns  CRBI  CWalks League Division  PutOuts  \
         Player
         -Alan Ashby             69    321   414     375      N        W      632
         -Alvin Davis            63    224   266     263      A        W      880
         -Andre Dawson          225    828   838     354      N        E      200
         -Andres Galarraga       12     48    46      33      N        E      805
         -Alfredo Griffin        19    501   336     194      A        W      282

                             Assists  Errors  Salary NewLeague
         Player
         -Alan Ashby              43      10   475.0         N
         -Alvin Davis             82      14   480.0         A
         -Andre Dawson            11       3   500.0         N
         -Andres Galarraga        40       4    91.5         N
         -Alfredo Griffin        421      25   750.0         A
```

We will have to do this by hand using the following code and display infomation about the variables we converted from strings to numbers.

```
In [46]: dummieVariables = pd.get_dummies(data[['League', 'Division', 'NewLeague']])
         dummieVariables.info()
         print(dummieVariables.head())

<class 'pandas.core.frame.DataFrame'>
Index: 263 entries, -Alan Ashby to -Willie Wilson
Data columns (total 6 columns):
League_A       263 non-null uint8
League_N       263 non-null uint8
Division_E     263 non-null uint8
Division_W     263 non-null uint8
NewLeague_A    263 non-null uint8
NewLeague_N    263 non-null uint8
dtypes: uint8(6)
memory usage: 3.6+ KB
                   League_A  League_N  Division_E  Division_W  NewLeague_A  \
Player
-Alan Ashby               0         1           0           1            0
```

```
-Alvin Davis              1         0           0           1            1
-Andre Dawson             0         1           1           0            0
-Andres Galarraga         0         1           1           0            0
-Alfredo Griffin          1         0           0           1            1

                   NewLeague_N
Player
-Alan Ashby                  1
-Alvin Davis                 0
-Andre Dawson                1
-Andres Galarraga            1
-Alfredo Griffin             0
```

Next we must remove the columns with our independent variable (Salary), and columns for which we created dummy variables and reinterduce them into our predictors.

```
In [47]: dummieVariables = pd.get_dummies(data[['League', 'Division', 'NewLeague']])
         dummieVariables.info()
         print(dummieVariables.head())

         y = data.Salary
         X_ = data.drop(['Salary', 'League', 'Division', 'NewLeague'], axis=1).astype('float64

         X = pd.concat([X_, dummieVariables[['League_N', 'Division_W', 'NewLeague_N']]], axis=
         X.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 263 entries, -Alan Ashby to -Willie Wilson
Data columns (total 6 columns):
League_A       263 non-null uint8
League_N       263 non-null uint8
Division_E     263 non-null uint8
Division_W     263 non-null uint8
NewLeague_A    263 non-null uint8
NewLeague_N    263 non-null uint8
dtypes: uint8(6)
memory usage: 3.6+ KB
                   League_A  League_N  Division_E  Division_W  NewLeague_A  \
Player
-Alan Ashby               0         1           0           1            0
-Alvin Davis              1         0           0           1            1
-Andre Dawson             0         1           1           0            0
-Andres Galarraga         0         1           1           0            0
-Alfredo Griffin          1         0           0           1            1

                   NewLeague_N
Player
```

```
-Alan Ashby                  1
-Alvin Davis                 0
-Andre Dawson                1
-Andres Galarraga            1
-Alfredo Griffin             0
```

```
Out[47]:                   AtBat   Hits  HmRun  Runs   RBI  Walks  Years  CAtBat  \
         Player
         -Alan Ashby        315.0   81.0    7.0  24.0  38.0   39.0   14.0  3449.0
         -Alvin Davis       479.0  130.0   18.0  66.0  72.0   76.0    3.0  1624.0
         -Andre Dawson      496.0  141.0   20.0  65.0  78.0   37.0   11.0  5628.0
         -Andres Galarraga  321.0   87.0   10.0  39.0  42.0   30.0    2.0   396.0
         -Alfredo Griffin   594.0  169.0    4.0  74.0  51.0   35.0   11.0  4408.0

                             CHits  CHmRun  CRuns   CRBI  CWalks  PutOuts  Assists  \
         Player
         -Alan Ashby         835.0    69.0  321.0  414.0   375.0    632.0     43.0
         -Alvin Davis        457.0    63.0  224.0  266.0   263.0    880.0     82.0
         -Andre Dawson      1575.0   225.0  828.0  838.0   354.0    200.0     11.0
         -Andres Galarraga   101.0    12.0   48.0   46.0    33.0    805.0     40.0
         -Alfredo Griffin   1133.0    19.0  501.0  336.0   194.0    282.0    421.0

                            Errors  League_N  Division_W  NewLeague_N
         Player
         -Alan Ashby          10.0         1           1            1
         -Alvin Davis         14.0         0           1            0
         -Andre Dawson         3.0         1           0            1
         -Andres Galarraga     4.0         1           0            1
         -Alfredo Griffin     25.0         0           1            0
```

bla bla

```
In [48]: import statsmodels.api as sm
         import itertools
         max_predictors = 3

         # Functions found at "https://github.com/qx0731/ISL_python/blob/master/Chapter_6_sec_
         def CalulateRSS(y, X, predictors_list):
             model = sm.OLS(y, X[list(predictors_list)]).fit()
             RSS   = ((model.predict(X[list(predictors_list)]) - y) ** 2)
             RSS   = RSS.sum()
             return {'Model':model, "RSS":RSS}

         def forwardStepwiseSelection(y, X, predictors_list):
             remaining_predictors = [p for p in X.columns if p not in predictors_list]
             results = []
             for p in remaining_predictors:
```

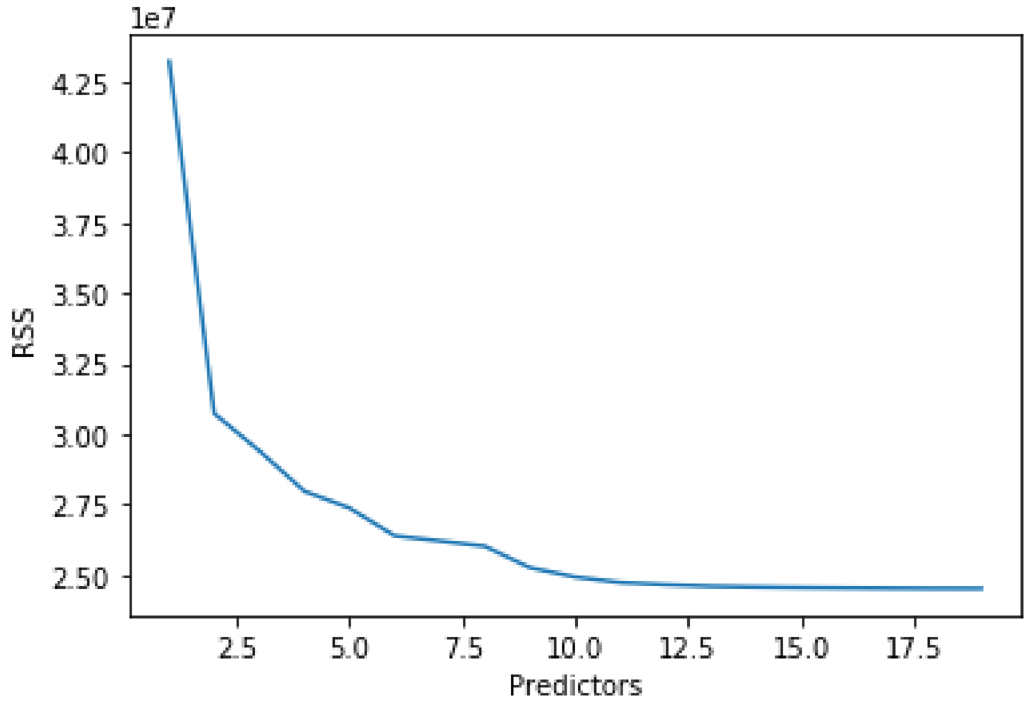
```
        results.append(CalulateRSS(y, X, feature_list+[p]))

    models = pd.DataFrame(results)
    best_model_forwardsetwise = models.loc[models['RSS'].argmin()]
    return best_model_forwardsetwise

def backwardStepwiseSelection(y, X, predictors_list):
    results = []
    for combo in itertools.combinations(predictors_list, len(predictors_list)-1):
        results.append(CalulateRSS(y, X, combo))

    models = pd.DataFrame(results)
    best_model = models.loc[models['RSS'].argmin()]
    return best_model
```

bla bla

```
In [49]: forwardModels = pd.DataFrame(columns=["RSS", "Model"])
         feature_list = []
         for i in range(1,len(X.columns)+1):
             forwardModels.loc[i] = forwardStepwiseSelection(y, X, feature_list)
             feature_list = forwardModels.loc[i]["Model"].model.exog_names
```

```
C:\Users\au479931\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launcher.py:19
```

```
In [50]: plt.figure()
         plt.plot(forwardModels["RSS"])
         plt.xlabel('Predictors')
         plt.ylabel('RSS')
         plt.show()
```



```
In [51]: print(forwardModels.loc[max_predictors, 'Model'].params)
```

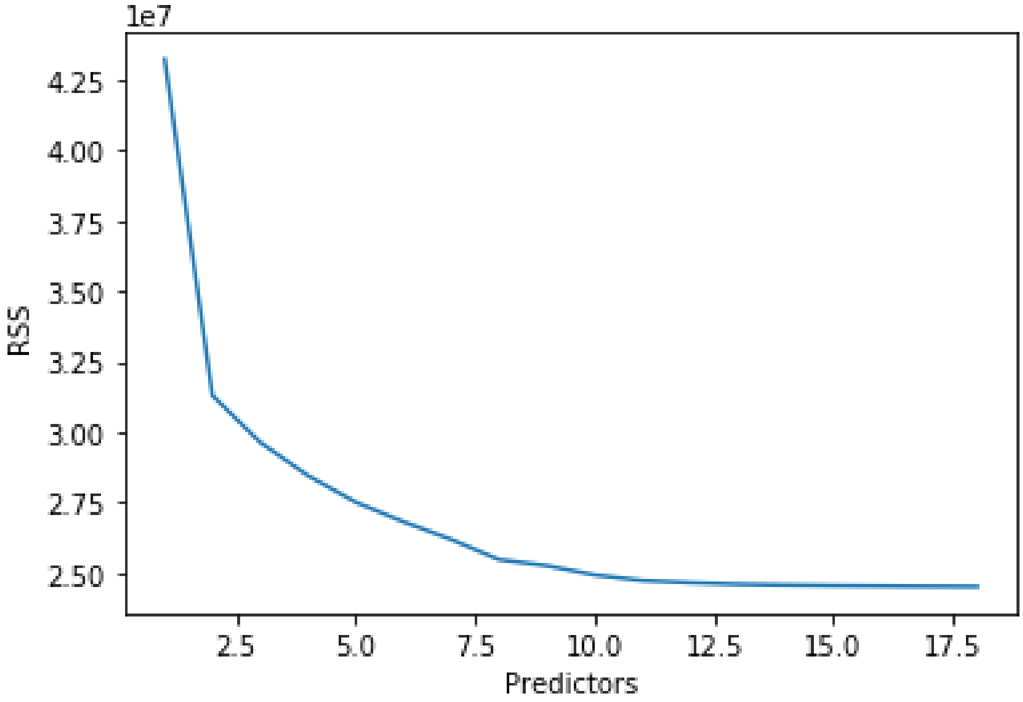
```
Hits             3.405706
CRBI             0.696362
Division_W    -129.160367
dtype: float64
```

bla bla

```
In [52]: backwardModels = pd.DataFrame(columns=["RSS", "Model"], index = range(1,len(X.columns)
         feature_list = X.columns

         while(len(feature_list) > 1):
             backwardModels.loc[len(feature_list)-1] = backwardStepwiseSelection(y, X, feature_
             feature_list = backwardModels.loc[len(feature_list)-1]["Model"].model.exog_names
```

```
C:\Users\au479931\AppData\Local\Continuum\anaconda3\lib\site-packages\ipykernel_launcher.py:28
```

```
In [53]: plt.figure()
         plt.plot(backwardModels["RSS"])
         plt.xlabel('Predictors')
         plt.ylabel('RSS')
         plt.show()
```

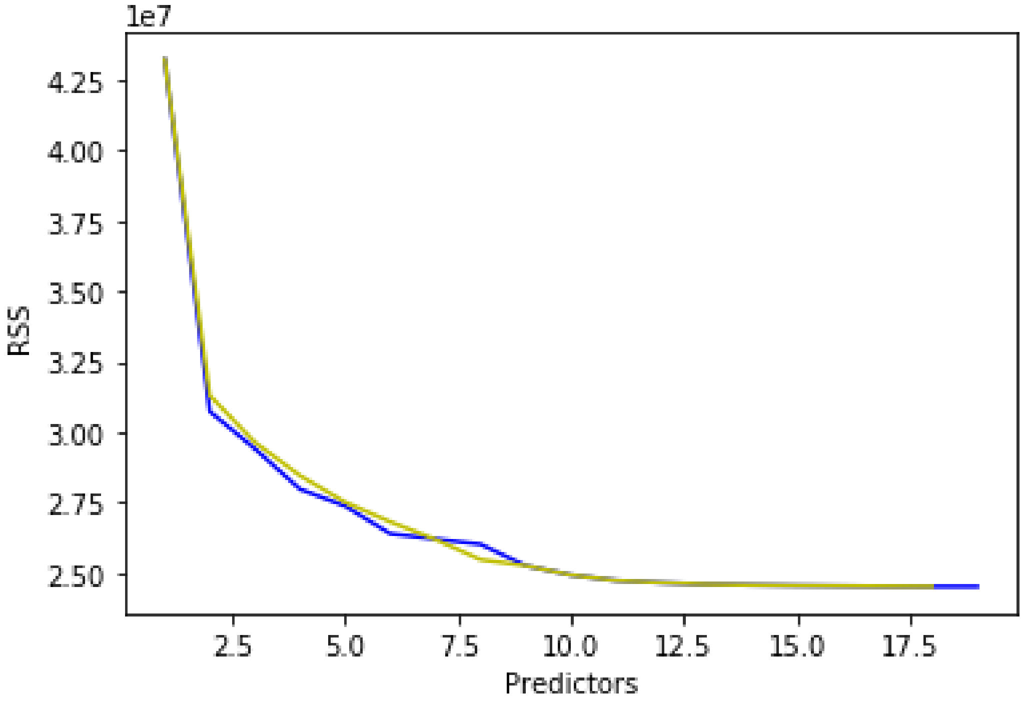


```
In [54]: print(backwardModels.loc[max_predictors, "Model"].params)
```

```
Hits       2.111712
CRuns      0.646149
PutOuts    0.295625
dtype: float64
```

Plotting both Forward and Backward Stepwise Selection in a single plot, to see the difference. Forward is blue, Backward is yellow.

```
In [56]: plt.figure()
         plt.plot(forwardModels["RSS"],'b')
         plt.plot(backwardModels["RSS"],'y')
         plt.xlabel('Predictors')
         plt.ylabel('RSS')
         plt.show()
```


1e7
4.25
4.00
3.75
3.50
3.25
3.00
2.75
2.50
RSS
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Predictors