

```
import warnings
warnings.simplefilter("ignore")
```

```
import numpy as np
import pandas as pd
from matplotlib.pyplot import subplots
from statsmodels.api import OLS
import sklearn.model_selection as skm
import sklearn.linear_model as skl
from sklearn.preprocessing import StandardScaler #used to help scale the predictors#
!pip install ISLP
from ISLP import load_data
from ISLP.models import ModelSpec as MS
from functools import partial
```

```
Collecting ISLP
  Downloading ISLP-0.4.0-py3-none-any.whl.metadata (7.0 kB)
Requirement already satisfied: numpy>=1.7.1 in /usr/local/lib/python3.10/dist-packages (from ISLP) (1.26.4)
Requirement already satisfied: scipy>=0.9 in /usr/local/lib/python3.10/dist-packages (from ISLP) (1.13.1)
Requirement already satisfied: pandas>=0.20 in /usr/local/lib/python3.10/dist-packages (from ISLP) (2.2.2)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packages (from ISLP) (4.9.4)
Requirement already satisfied: scikit-learn>=1.2 in /usr/local/lib/python3.10/dist-packages (from ISLP) (1.5.2)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from ISLP) (1.4.2)
Requirement already satisfied: statsmodels>=0.13 in /usr/local/lib/python3.10/dist-packages (from ISLP) (0.14.4)
Collecting lifelines (from ISLP)
  Downloading lifelines-0.30.0-py3-none-any.whl.metadata (3.2 kB)
Collecting pygam (from ISLP)
  Downloading pygam-0.9.1-py3-none-any.whl.metadata (7.1 kB)
Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (from ISLP) (2.5.0+cu121)
Collecting pytorch-lightning (from ISLP)
  Downloading pytorch_lightning-2.4.0-py3-none-any.whl.metadata (21 kB)
Collecting torchmetrics (from ISLP)
  Downloading torchmetrics-1.5.1-py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.20->ISLP) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.20->ISLP) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.20->ISLP) (2024.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.2->ISLP) (3.5.0)
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.10/dist-packages (from statsmodels>=0.13->ISLP) (0.5.6)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.10/dist-packages (from statsmodels>=0.13->ISLP) (24.1)
Requirement already satisfied: matplotlib>=3.0 in /usr/local/lib/python3.10/dist-packages (from lifelines->ISLP) (3.7.1)
Requirement already satisfied: autograd>=1.5 in /usr/local/lib/python3.10/dist-packages (from lifelines->ISLP) (1.7.0)
Collecting autograd-gamma>=0.3 (from lifelines->ISLP)
  Downloading autograd-gamma-0.5.0.tar.gz (4.0 kB)
  Preparing metadata (setup.py) ... done
Collecting formulaic>=0.2.2 (from lifelines->ISLP)
  Downloading formulaic-1.0.2-py3-none-any.whl.metadata (6.8 kB)
Requirement already satisfied: progressbar<5.0.0,>=4.2.0 in /usr/local/lib/python3.10/dist-packages (from pygam->ISLP) (4.5.0)
Collecting scipy>=0.9 (from ISLP)
  Downloading scipy-1.11.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (60 kB)
  60.4/60.4 kB 2.4 MB/s eta 0:00:00
Requirement already satisfied: tqdm>=4.57.0 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning->ISLP) (4.66.5)
Requirement already satisfied: PyYAML>=5.4 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning->ISLP) (6.0.2)
Requirement already satisfied: fsspec>=2022.5.0 in /usr/local/lib/python3.10/dist-packages (from fsspec[http]>=2022.5.0->pytorch-lightning->ISLP) (2024.10.1)
Requirement already satisfied: typing-extensions>=4.4.0 in /usr/local/lib/python3.10/dist-packages (from pytorch-lightning->ISLP) (4.12.0)
Collecting lightning-utilities>=0.10.0 (from pytorch-lightning->ISLP)
  Downloading lightning_utilities-0.11.8-py3-none-any.whl.metadata (5.2 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch->ISLP) (3.16.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch->ISLP) (3.4.2)
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.10/dist-packages (from torch->ISLP) (3.1.4)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-packages (from torch->ISLP) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch->ISLP) (1.3.0)
Collecting interface-meta>=1.2.0 (from formulaic>=0.2.2->lifelines->ISLP)
  Downloading interface_meta-1.3.0-py3-none-any.whl.metadata (6.7 kB)
Requirement already satisfied: wrapt>=1.0 in /usr/local/lib/python3.10/dist-packages (from formulaic>=0.2.2->lifelines->ISLP) (1.16.0)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in /usr/local/lib/python3.10/dist-packages (from fsspec[http]>=2022.5.0->pytorch-lightning->ISLP) (4.0.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from lightning-utilities>=0.10.0->pytorch-lightning->ISLP) (68.1.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0->lifelines->ISLP) (1.1.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0->lifelines->ISLP) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0->lifelines->ISLP) (4.53.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0->lifelines->ISLP) (1.4.5)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0->lifelines->ISLP) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0->lifelines->ISLP) (3.2.0)
```

```
from sklearn.pipeline import Pipeline
from sklearn.decomposition import PCA
from sklearn.cross_decomposition import PLSRegression
from ISLP.models import \
    (Stepwise,
```

```

sklearn_selected,
sklearn_selection_path)

!pip install l0bnb

from l0bnb import fit_path

Collecting l0bnb
  Downloading l0bnb-1.0.0.tar.gz (79 kB)
    79.8/79.8 kB 2.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: numpy>=1.18.1 in /usr/local/lib/python3.10/dist-packages (from l0bnb) (1.26.4)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from l0bnb) (1.11.4)
Requirement already satisfied: numba>=0.53.1 in /usr/local/lib/python3.10/dist-packages (from l0bnb) (0.60.0)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.10/dist-packages (from numba>=0.53.1->l0bnb) (0.43.0)
Building wheels for collected packages: l0bnb
  Building wheel for l0bnb (setup.py) ... done
  Created wheel for l0bnb: filename=l0bnb-1.0.0-py3-none-any.whl size=22295 sha256=951e97bebb44e6d705043fc6abbdf8ce6dbff7887aa62a4ef4813
  Stored in directory: /root/.cache/pip/wheels/db/c7/a3/a2e42159dcc85eeb004a84772e1884c45f88d43fac0075de19
Successfully built l0bnb
Installing collected packages: l0bnb
Successfully installed l0bnb-1.0.0

```

✓ Load the data

```

Hitters = load_data('Hitters')
print(Hitters)

```

```

AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun \
0      293    66     1    30    29    14     1    293     66     1
1      315    81     7    24    38    39    14    3449    835    69
2      479   130    18    66    72    76     3    1624    457    63
3      496   141    20    65    78    37    11    5628   1575   225
4      321    87    10    39    42    30     2     396    101    12
..      ...    ...    ...    ...    ...    ...    ...    ...    ...    ...
317    497   127     7    65    48    37     5    2703    806    32
318    492   136     5    76    50    94    12    5511   1511    39
319    475   126     3    61    43    52     6    1700    433     7
320    573   144     9    85    60    78     8    3198    857    97
321    631   170     9    77    44    31    11    4908   1457    30

CRuns CRBI CWalks League Division PutOuts Assists Errors Salary \
0      30    29    14     A     E    446    33    20     NaN
1      321   414   375     N     W    632    43    10    475.0
2      224   266   263     A     W    880    82    14    480.0
3      828   838   354     N     E    200    11     3    500.0
4       48    46    33     N     E    805    40     4    91.5
..      ...    ...    ...    ...    ...    ...    ...    ...    ...
317    379   311   138     N     E    325     9     3    700.0
318    897   451   875     A     E    313    381    20    875.0
319    217    93   146     A     W     37   113     7    385.0
320    470   420   332     A     E   1314   131    12   960.0
321    775   357   249     A     W    408     4     3  1000.0

NewLeague
0      A
1      N
2      A
3      N
4      N
..      ...
317    N
318    A
319    A
320    A
321    A

```

[322 rows x 20 columns]


```
np.isnan(Hitters['Salary']).sum()
```

```
59
```

```

Hitters = Hitters.dropna()
Hitters.shape

```

 (263, 20)

✓ OLS Regression

```
design = MS(Hitters.columns.drop('Salary')).fit(Hitters)
Y = np.array(Hitters['Salary'])
X = design.transform(Hitters)
ols_results = OLS(Y,X).fit()
print(ols_results.summary())
```



OLS Regression Results

```
=====
Dep. Variable:          y      R-squared:          0.546
Model:                  OLS    Adj. R-squared:      0.511
Method:                 Least Squares  F-statistic:    15.39
Date:                   Wed, 30 Oct 2024  Prob (F-statistic): 7.84e-32
Time:                   19:00:51  Log-Likelihood: -1876.2
No. Observations:      263      AIC:             3792.
Df Residuals:          243      BIC:             3864.
Df Model:              19
Covariance Type:       nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
intercept	163.1036	90.779	1.797	0.074	-15.710	341.917
AtBat	-1.9799	0.634	-3.123	0.002	-3.229	-0.731
Hits	7.5008	2.378	3.155	0.002	2.818	12.184
HmRun	4.3309	6.201	0.698	0.486	-7.885	16.546
Runs	-2.3762	2.981	-0.797	0.426	-8.248	3.495
RBI	-1.0450	2.601	-0.402	0.688	-6.168	4.078
Walks	6.2313	1.829	3.408	0.001	2.630	9.833
Years	-3.4891	12.412	-0.281	0.779	-27.938	20.960
CAtBat	-0.1713	0.135	-1.267	0.206	-0.438	0.095
Chits	0.1340	0.675	0.199	0.843	-1.195	1.463
CHmRun	-0.1729	1.617	-0.107	0.915	-3.358	3.013
CRuns	1.4543	0.750	1.938	0.054	-0.024	2.933
CRBI	0.8077	0.693	1.166	0.245	-0.557	2.172
CWalks	-0.8116	0.328	-2.474	0.014	-1.458	-0.165
League[N]	62.5994	79.261	0.790	0.430	-93.528	218.727
Division[W]	-116.8492	40.367	-2.895	0.004	-196.363	-37.335
PutOuts	0.2819	0.077	3.640	0.000	0.129	0.434
Assists	0.3711	0.221	1.678	0.095	-0.065	0.807
Errors	-3.3608	4.392	-0.765	0.445	-12.011	5.290
NewLeague[N]	-24.7623	79.003	-0.313	0.754	-180.380	130.855

```
=====
Omnibus:                 87.414  Durbin-Watson:          2.018
Prob(Omnibus):           0.000  Jarque-Bera (JB):        452.923
Skew:                    1.236  Prob(JB):                4.46e-99
Kurtosis:                8.934  Cond. No.:               2.09e+04
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
 [2] The condition number is large, 2.09e+04. This might indicate that there are strong multicollinearity or other numerical problems.

```
# Convert X into NumPy array so we can calculate the standardize
D = design.fit_transform(Hitters)
D = D.drop('intercept', axis=1)
X = np.asarray(D)
```

```
Xs = X - X.mean(0)[None,:]
# 0 means calculate the means along the colum; 1 indicate
# None for making one more dimension to the data array and
x_scale = X.std(0) #scale the predictors#
Xs = Xs/x_scale[None,:]
```

```
print(Xs)
```



```
[[[-0.6029005 -0.59567545 -0.5285512 ... -0.52319133  0.21335208
  1.07505242]
 [ 0.51254171  0.49225957  0.72996619 ... -0.25386267  0.81996395
 -0.9301872 ]
 [ 0.62816682  0.73648988  0.95878753 ... -0.74417894 -0.84821868
  1.07505242]
 ...
 [ 0.48533581  0.40344855 -0.98619389 ... -0.03978092 -0.24160682
```

```
-0.9301872 ]
[ 1.15188054  0.80309815 -0.29972986 ...  0.08452461  0.51665801
-0.9301872 ]
[ 1.54636621  1.38036979 -0.29972986 ... -0.79251998 -0.84821868
-0.9301872 ]]
```

✓ Create values for lambdas

```
# Create values for lambdas
lambdas = 10**np.linspace(8,-2,100)/Y.std()
# lambdas will contain 100 values, scaled by the standard

print(lambdas)
```

```
[2.22093791e+05 1.76005531e+05 1.39481373e+05 1.10536603e+05
 8.75983676e+04 6.94202082e+04 5.50143278e+04 4.35979140e+04
 3.45506012e+04 2.73807606e+04 2.16987845e+04 1.71959156e+04
 1.36274691e+04 1.07995362e+04 8.55844774e+03 6.78242347e+03
 5.37495461e+03 4.25955961e+03 3.37562814e+03 2.67512757e+03
 2.11999285e+03 1.68005808e+03 1.33141730e+03 1.05512544e+03
 8.36168866e+02 6.62649526e+02 5.25138417e+02 4.16163215e+02
 3.29802231e+02 2.61362628e+02 2.07125413e+02 1.64143347e+02
 1.30080796e+02 1.03086806e+02 8.16945308e+01 6.47415185e+01
 5.13065463e+01 4.06595605e+01 3.22220063e+01 2.55353890e+01
 2.02363591e+01 1.60369685e+01 1.27090233e+01 1.00716836e+01
 7.98163700e+00 6.32531083e+00 5.01270066e+00 3.97247954e+00
 3.14812210e+00 2.49483293e+00 1.97711243e+00 1.56682779e+00
 1.24168423e+00 9.84013516e-01 7.79813883e-01 6.17989166e-01
 4.89745845e-01 3.88115207e-01 3.07574664e-01 2.43747661e-01
 1.93165853e-01 1.53080635e-01 1.21313785e-01 9.61391003e-02
 7.61885928e-02 6.03781569e-02 4.78486568e-02 3.79192422e-02
 3.00503510e-02 2.38143892e-02 1.88724962e-02 1.49561305e-02
 1.18524776e-02 9.39288583e-03 7.44370138e-03 5.89900605e-03
 4.67486141e-03 3.70474772e-03 2.93594921e-03 2.32668954e-03
 1.84386167e-03 1.46122884e-03 1.15799887e-03 9.17694298e-04
 7.27257037e-04 5.76338765e-04 4.56738615e-04 3.61957541e-04
 2.86845161e-04 2.27319885e-04 1.80147121e-04 1.42763513e-04
 1.13137642e-04 8.96596467e-05 7.10537367e-05 5.63088712e-05
 4.46238174e-05 3.53636122e-05 2.80250579e-05 2.22093791e-05]
```

```
design = MS(Hitters.columns.drop('Salary')).fit(Hitters)
Y = np.array(Hitters['Salary'])
X = design.transform(Hitters)
ols_results = OLS(Y,Xs).fit()
print(ols_results.summary())
```

```
OLS Regression Results
=====
Dep. Variable:          y      R-squared (uncentered):          0.226
Model:                  OLS    Adj. R-squared (uncentered):      0.166
Method:                 Least Squares      F-statistic:              3.749
Date:                   Wed, 30 Oct 2024    Prob (F-statistic):       7.05e-07
Time:                   19:14:50           Log-Likelihood:           -2062.4
No. Observations:       263               AIC:                    4163.
Df Residuals:           244               BIC:                    4231.
Df Model:                19
Covariance Type:        nonrobust
=====
               coef      std err          t      P>|t|      [0.025      0.975]
-----
x1            -291.0946    188.841     -1.541     0.124    -663.062     80.873
x2             337.8305    216.943     1.557     0.121    -89.490    765.151
x3              37.8538    109.813     0.345     0.731   -178.448    254.156
x4            -60.5725    153.937    -0.393     0.694   -363.787    242.642
x5            -26.9950    136.122    -0.198     0.843   -295.119    241.129
x6             135.0739     80.300     1.682     0.094    -23.096    293.244
x7            -16.6934    120.312    -0.139     0.890   -253.677    220.290
x8            -391.0387    625.289    -0.625     0.532  -1622.692    840.614
x9              86.6876    884.144     0.098     0.922  -1654.842    1828.217
x10           -14.1817    268.801    -0.053     0.958   -543.649    515.286
x11            480.7471    502.590     0.957     0.340   -509.221    1470.715
x12           260.6899    452.887     0.576     0.565   -631.377    1152.757
x13          -213.8923    175.177    -1.221     0.223   -558.945    131.160
x14             31.2488     80.159     0.390     0.697   -126.642    189.140
x15           -58.4141     40.883    -1.429     0.154   -138.943     22.115
x16             78.7613     43.835     1.797     0.074    -7.583    165.105
x17            53.7325     64.892     0.828     0.408    -74.087    181.552
x18           -22.1609     58.668    -0.378     0.706   -137.721     93.399
x19           -12.3488     79.818    -0.155     0.877   -169.569    144.872
```

```
=====
Omnibus:            87.414    Durbin-Watson:           0.490
Prob(Omnibus):      0.000    Jarque-Bera (JB):        452.923
Skew:               1.236    Prob(JB):               4.46e-99
Kurtosis:           8.934    Cond. No.                78.3
=====
```

Notes:

[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
 [2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
# run ridge regression l1_ratio = 0, 1 for lasso
# skl.ElasticNet.path returns [0] alphas(lambda); [1]
soln_array = skl.ElasticNet.path(Xs,
                                Y,
                                l1_ratio = 0.,
                                alphas = lambdas)[1] #[1]
```

soln_array

```
array([[ 8.00310363e-04,  1.00986867e-03,  1.27429625e-03, ...,
        -2.90923382e+02, -2.90958537e+02, -2.90986528e+02],
       [ 8.89318672e-04,  1.12218451e-03,  1.41602291e-03, ...,
        3.37260446e+02,  3.37377455e+02,  3.37470648e+02],
       [ 6.95410715e-04,  8.77500619e-04,  1.10726717e-03, ...,
        3.75180641e+01,  3.75871222e+01,  3.76420770e+01],
       ...,
       [ 5.15666738e-05,  6.50694021e-05,  8.21076995e-05, ...,
        5.36633570e+01,  5.36777586e+01,  5.36891521e+01],
       [-1.09496252e-05, -1.38170424e-05, -1.74354208e-05, ...,
        -2.21910710e+01, -2.21848930e+01, -2.21799639e+01],
       [-5.74274914e-06, -7.24531429e-06, -9.14061794e-06, ...,
        -1.23832054e+01, -1.23761905e+01, -1.23705867e+01]])
```

Organize the results

```
soln_path = pd.DataFrame(soln_array.T,
                        columns = D.columns,
                        index = np.log(lambdas))
```

```
soln_path.index.name = 'log(lambda)'
```

```
soln_path
```

#Each columns shows a solution path as lambda changes

```

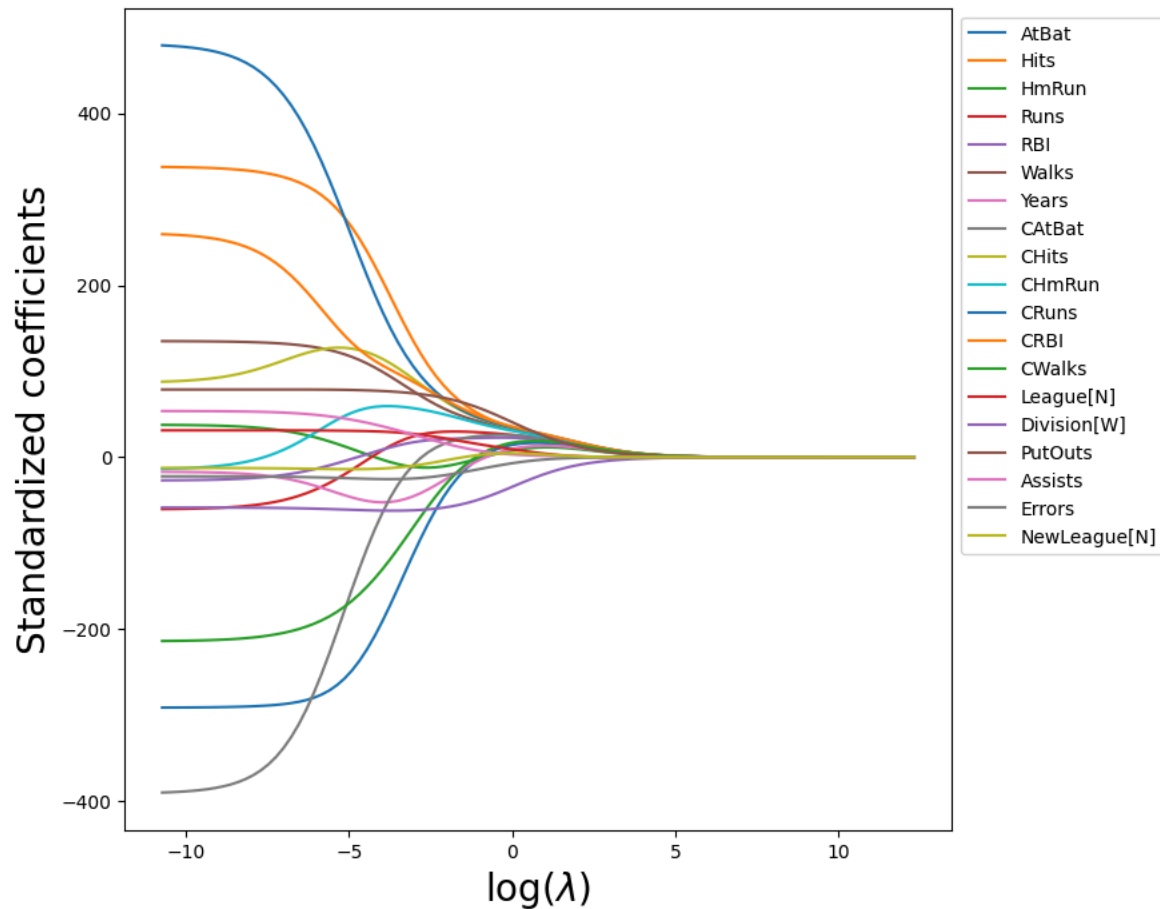
AtBat    Hits    HmRun    Runs    RBI    Walks    Years    CAtBat    CHits    CHmRun    CRuns
log(lambda)
12.310855    0.000800    0.000889    0.000695    0.000851    0.000911    0.000900    0.000812    0.001067    0.001113    0.001064    0.001141
12.078271    0.001010    0.001122    0.000878    0.001074    0.001150    0.001135    0.001025    0.001346    0.001404    0.001343    0.001439
11.845686    0.001274    0.001416    0.001107    0.001355    0.001451    0.001433    0.001293    0.001698    0.001772    0.001694    0.001816
11.613102    0.001608    0.001787    0.001397    0.001710    0.001831    0.001808    0.001632    0.002143    0.002236    0.002138    0.002292
11.380518    0.002029    0.002255    0.001763    0.002158    0.002310    0.002281    0.002059    0.002704    0.002821    0.002698    0.002892
...
-9.784658    -290.823989    336.929968    37.322686    -59.748520    -26.507086    134.855915    -17.216195    -387.775826    89.573601    -12.273926    476.079273
-10.017243    -290.879272    337.113713    37.431373    -59.916820    -26.606957    134.900549    -17.108041    -388.458404    89.000707    -12.661459    477.031349
-10.249827    -290.923382    337.260446    37.518064    -60.051166    -26.686604    134.936136    -17.022194    -388.997470    88.537380    -12.971603    477.791860
-10.482412    -290.958537    337.377455    37.587122    -60.158256    -26.750044    134.964477    -16.954081    -389.423414    88.164178    -13.219329    478.398404
-10.714996    -290.986528    337.470648    37.642077    -60.243522    -26.800522    134.987027    -16.900054    -389.760135    87.864551    -13.416889    478.881540
```

100 rows × 19 columns


make a plot

```
path_fig, ax =subplots(figsize=(8,8))
soln_path.plot(ax=ax, legend=False)
ax.set_xlabel('$\log(\lambda)$', fontsize = 20)
ax.set_ylabel('Standardized coefficients', fontsize = 20)
ax.legend(loc='upper left', bbox_to_anchor=(1,1))
#
```

 <matplotlib.legend.Legend at 0x7d28b826d420>



```
# show an example on the 40th lambda
beta_hat = soln_path.loc[soln_path.index[60]]
lambdas[60], beta_hat
```

 (0.19316585312165907,
AtBat -22.689698
Hits 66.754391
HmRun -7.409850
Runs 29.868018
RBI 21.784924
Walks 49.392717
Years -16.590135
CAtBat 20.897615
CHits 56.013847
CHmRun 45.619633
CRuns 57.028589
CRBI 58.853412
CWalks -15.864036
League[N] 19.285863
Division[W] -55.514189
PutOuts 65.064927
Assists 12.282992
Errors -18.299548
NewLeague[N] -0.887209
Name: -1.6442061165014366, dtype: float64)

▼ Make a pipeline

```
ridge = skl.ElasticNet(alpha=lambdas[60], l1_ratio = 0)
scaler = StandardScaler(with_mean=True, with_std=True)
pipe = Pipeline([('scaler', scaler), ('ridge', ridge)])
pipe.fit(X,Y)
```



✓ Set up the k-fold cross validation

```

K = 5
kfold = skm.KFold(K,
                  random_state=0,
                  shuffle=True)

param_grid = {'ridge__alpha': lambdas}
grid = skm.GridSearchCV(pipe,
                        param_grid,
                        cv=kfold,
                        scoring='neg_mean_squared_error')

grid.fit(X, Y)
grid.best_params_['ridge__alpha']
grid.best_estimator_

```



```

rige_fig, ax = subplots(figsize=(8,8))
ax.errorbar(-np.log(lambdas),
            -grid.cv_results_['mean_test_score'],
            yerr=grid.cv_results_['std_test_score']/ np.sqrt(K))
ax.set_ylim([50000,250000])
ax.set_xlabel('$\log(\lambda)$', fontsize=20)
ax.set_ylabel('Cross-validated MSE', fontsize=20)

```

```
Text(0, 0.5, 'Cross-validated MSE')
```

