1.) Import the data from CCLE into a new Google Colab file

```
import pandas as pd
from google.colab import drive
import matplotlib.pyplot as plt
```

```
drive.mount('/content/gdrive/', force_remount = True)
```

```
Mounted at /content/gdrive/
```

```
df = pd.read_csv("/content/gdrive/MyDrive/Econ441B/insurance.csv")
df
```

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

```
# Convert categorical variables to dummy variables
df = pd.get_dummies(df)
df
```

	age	bmi	children	charges	sex_0	sex_1	smoker_no	smoker_yes	regi
0	19	27.900	0	16884.92400	0	1	0	1	
1	18	33.770	1	1725.55230	1	0	1	0	
2	28	33.000	3	4449.46200	1	0	1	0	
3	33	22.705	0	21984.47061	1	0	1	0	
4	32	28.880	0	3866.85520	1	0	1	0	
...	...	...	...	...	...	...	...	...	
1333	50	30.970	3	10600.54830	1	0	1	0	
1334	18	31.920	0	2205.98080	0	1	1	0	
1335	18	36.850	0	1629.83350	0	1	1	0	

```
# Drop the redundant dunmmy variables
df = df.drop(['sex_0','smoker_no','region_northeast'], axis=1)
df
```

	age	bmi	children	charges	sex_1	smoker_yes	region_northwest	regi
0	19	27.900	0	16884.92400	1	1	0	
1	18	33.770	1	1725.55230	0	0	0	
2	28	33.000	3	4449.46200	0	0	0	
3	33	22.705	0	21984.47061	0	0	1	
4	32	28.880	0	3866.85520	0	0	1	
...	...	...	...	...	...	...	...	
1333	50	30.970	3	10600.54830	0	0	1	
1334	18	31.920	0	2205.98080	1	0	0	
1335	18	36.850	0	1629.83350	1	0	0	
1336	21	25.800	0	2007.94500	1	0	0	
1337	61	29.070	0	29141.36030	1	1	1	

1338 rows × 9 columns

```
x = df.drop('charges', axis=1)
y = df['charges']
x
```

	age	bmi	children	sex_1	smoker_yes	region_northwest	region_southeast
0	19	27.900	0	1	1	0	0
1	18	33.770	1	0	0	0	1
2	28	33.000	3	0	0	0	1
3	33	22.705	0	0	0	1	0
4	32	28.880	0	0	0	1	0
...	...	...	...	...	...	...	...
1333	50	30.970	3	0	0	1	0
1334	18	31.920	0	1	0	0	0
1335	18	36.850	0	1	0	0	1
1336	21	25.800	0	1	0	0	0

```
y
```

```
0       16884.92400
1        1725.55230
2        4449.46200
3       21984.47061
4        3866.85520
           ...
1333    10600.54830
1334     2205.98080
1335     1629.83350
1336     2007.94500
1337    29141.36030
Name: charges, Length: 1338, dtype: float64
```

2.) Split the data into 80/20, in/out sample

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, train_size=0.8, random_state=1)
x_train
```

	age	bmi	children	sex_1	smoker_yes	region_northwest	region_southeast
216	53	26.600	0	1	0	1	0
731	53	21.400	1	0	0	0	0
866	18	37.290	0	0	0	0	1
202	60	24.035	0	1	0	1	0
820	45	33.700	1	0	0	0	0

3.) Normalize the Data

```
from sklearn import preprocessing
```

235	40	22.220	2	1	1	0	1

```
scaler = preprocessing.StandardScaler().fit(x_train)
x_train_a = scaler.transform(x_train)
x_test_a = scaler.transform(x_test)
x_train_a
```

```
array([[ 1.00228629, -0.66474472, -0.90705771, ...,  1.76954066,
        -0.59822071, -0.57519194],
       [ 1.00228629, -1.51402369, -0.07894188, ..., -0.56511841,
        -0.59822071,  1.73855008],
       [-1.50426607,  1.08117685, -0.90705771, ..., -0.56511841,
         1.67162383, -0.57519194],
       ...,
       [ 0.85905473,  0.70063454,  0.74917395, ..., -0.56511841,
        -0.59822071, -0.57519194],
       [ 0.07128113, -1.38009893,  0.74917395, ..., -0.56511841,
         1.67162383, -0.57519194],
       [ 1.28874942, -0.44589206, -0.07894188, ..., -0.56511841,
         1.67162383, -0.57519194]])
```

4.) Get lambda from Lasso cross validation

```
from sklearn.linear_model import LassoCV
```

```
modCV = LassoCV().fit(x_train_a,y_train)
```

```
#Optimized Lambda
f = modCV.alpha_
f
```

```
9.516307336182564
```

5.) Run a lambda regression with that Lambda

```
from sklearn.linear_model import Lasso


mod1 = Lasso(alpha = f).fit(x_train_a,y_train)


mod1.predict(x_train_a)
```

```
array([10528.79875767,  8543.36645006,  4095.27741985, ...,
       37566.95159623, 29713.00361202, 11528.21393581])
```

6.) Visualize the coefficients

```
mod1.coef_
```

```
array([3587.20696111, 1954.75656039,  481.64741897,  109.71211623,
       9602.45080015, -139.34597839, -423.25085492, -358.73545931])
```

```
mod1.intercept_
```

```
13230.161574933647
```

7.) Interpret the coefficients

1 unit increament in age will increase charges by 3587.2

1 unit increament in bmi will increase charges by 1954.8

The remaining variables will increase or decrease charges if it is set to 1

8.) Compare in and out of sample MSE's

```
from sklearn.metrics import mean_squared_error
# In sample MSE
ytr = mod1.predict(x_train_a)
MSE_train = mean_squared_error(y_train, ytr)
MSE_train
```

```
36789211.69656349
```

```
# Out of Sample MSE
yte = mod1.predict(x_test_a)
```

```
MSE_test = mean_squared_error(y_test, yte)
MSE_test
```

```
35406816.000325325
```

Since the results for both in and out of sample are very close, thus, the model is expected to perfrom well out of sample

Colab paid products - Cancel contracts here

0s completed at 18:37

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.