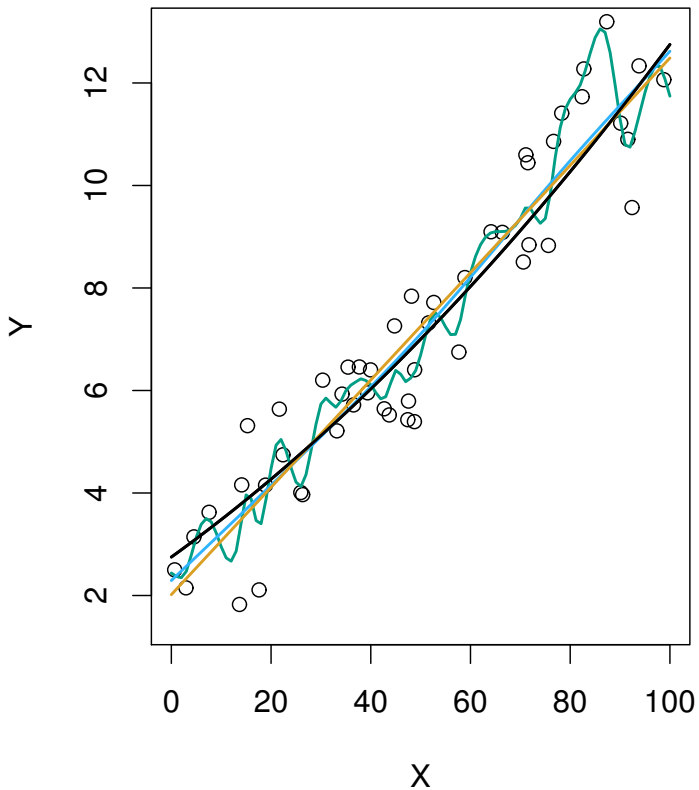
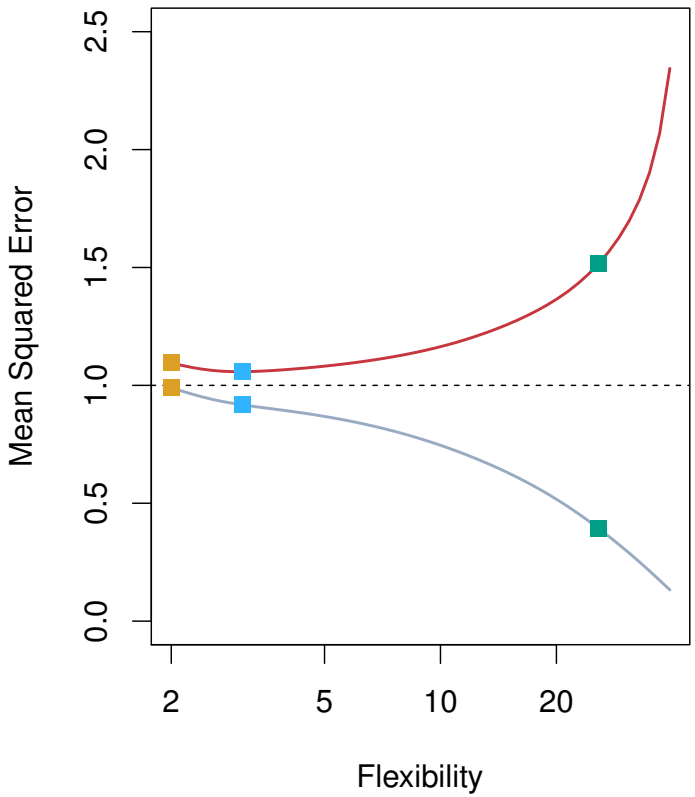
Cross-validation

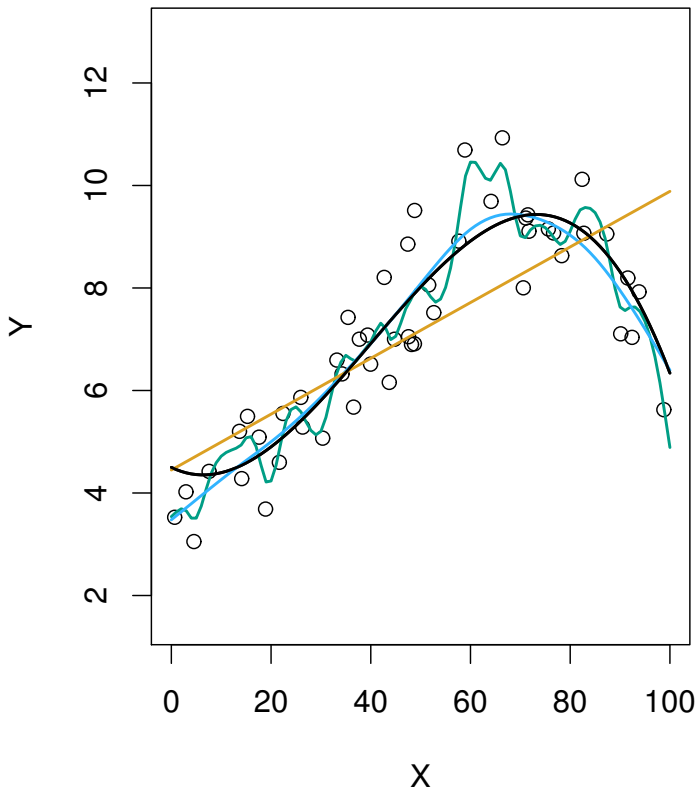
Vahid Partovi Nia

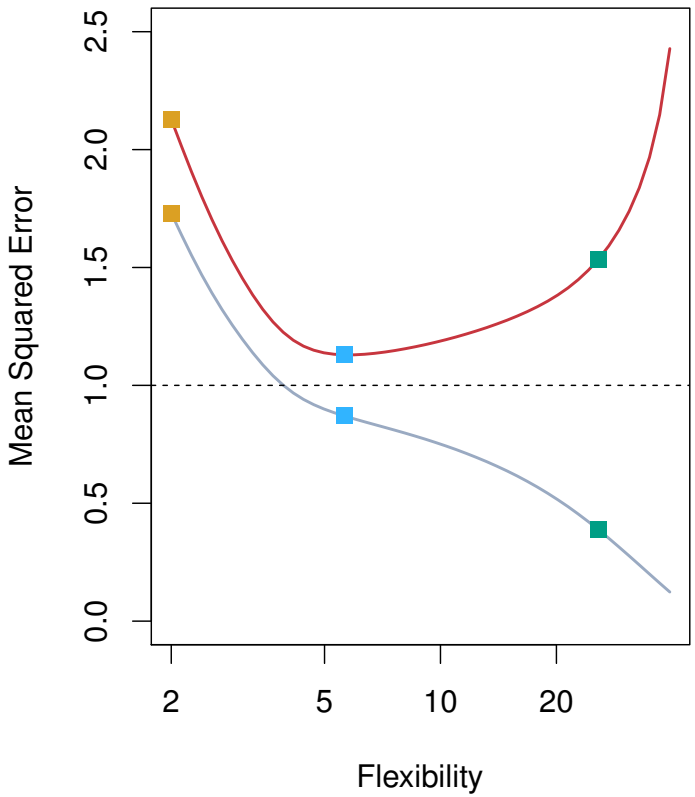
Lecture 04

Outline









Auto Dataset

	mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name
0	18.0	8	307.0	130.0	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	1	ford torino

```
import pandas as pd
path='data/'
filename = path+'Auto.csv'
auto = pd.read_csv(filename, na_values=['?'], na_filter=True)
auto = auto.dropna()
```

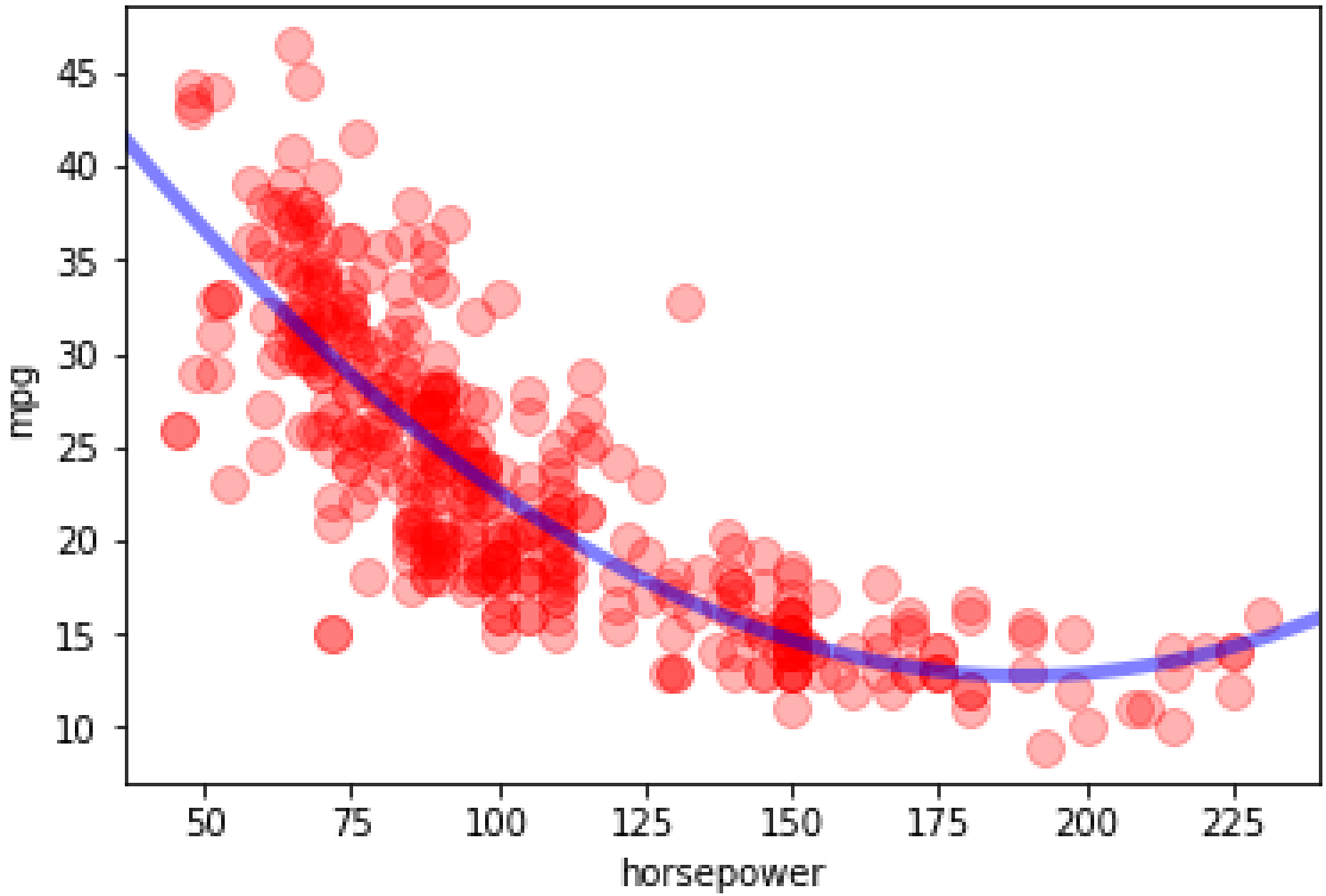
```
import pandas as pd
path='data/'
filename = path+'Auto.csv'
auto = pd.read_csv(filename, na_values=['?'], na_filter=True)
auto = auto.dropna()
```

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.plot(auto['horsepower'], auto['mpg'], 'or', mfc='none');
```

```
import pandas as pd
path='data/'
filename = path+'Auto.csv'
auto = pd.read_csv(filename, na_values=['?'], na_filter=True)
auto = auto.dropna()
```

```
import matplotlib.pyplot as plt
%matplotlib inline
plt.plot(auto['horsepower'], auto['mpg'], 'or', mfc='none');
```

```
import seaborn as sns
sns.regplot(x="horsepower", y="mpg", data=auto, ci=False,
    scatter_kws={"color":"r", "alpha":0.3, "s":100},
    line_kws={"color":"b", "alpha":0.5, "lw":4}, marker="o", order=2)
```


mpg
45
40
35
30
25
20
15
10
50
75
100
125
150
175
200
225
horsepower

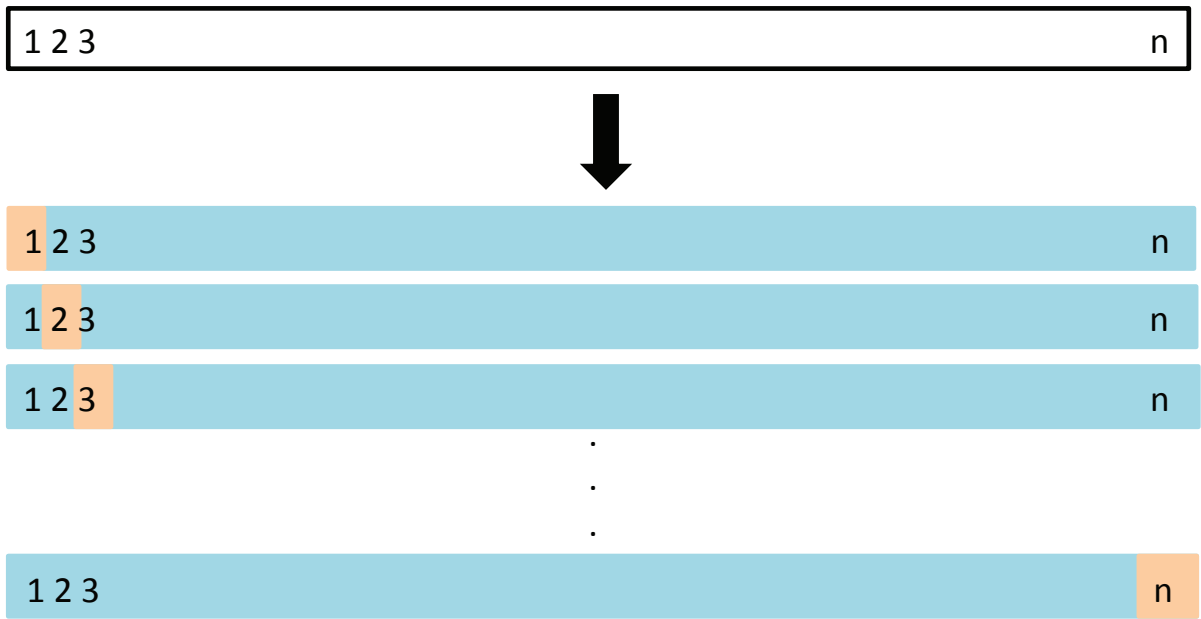
```
import numpy as np
import statsmodels.formula.api as smf
model = smf.ols(formula='mpg ~ horsepower', data=auto)

lr1 = model.fit()
lr1.summary2()

lr1.aic
```

```
model = smf.ols(formula='mpg ~ horsepower +
                    np.power(horsepower,2)', data = auto)
lr2 = model.fit()
lr2.aic
```

```
model = smf.ols(formula='mpg ~ horsepower +
                np.power(horsepower,2)', data = auto)
lr2 = model.fit()
lr2.aic
```

```
model = smf.ols(formula='mpg ~ horsepower +
                np.power(horsepower,2)+ np.power(horsepower,3)',
                data = auto)
lr3 = model.fit()
lr3.aic
```

Leave-one-out



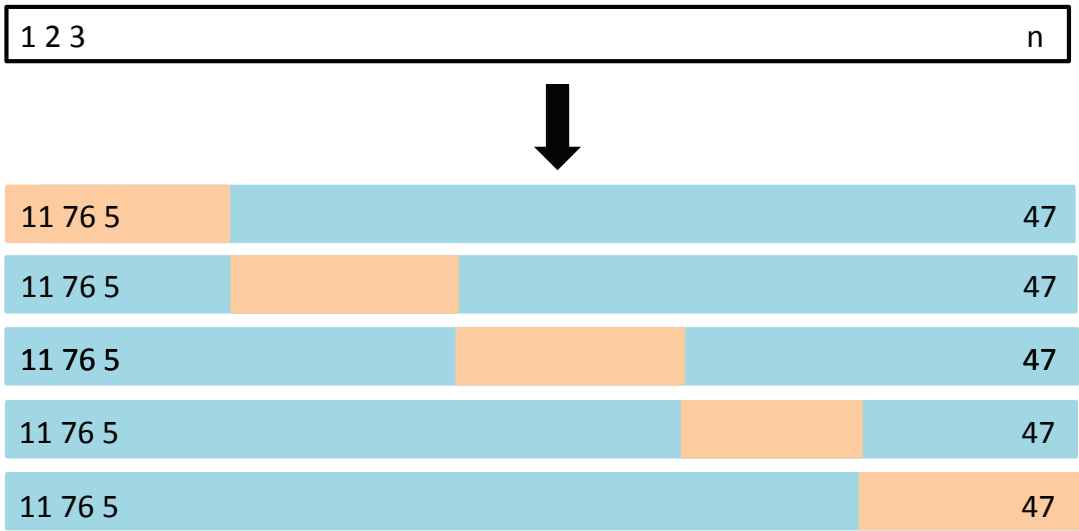
$$\text{RSS} = n\hat{\sigma}^2 = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

```
from sklearn.model_selection import LeaveOneOut
from sklearn.linear_model import LinearRegression
loo = LeaveOneOut()
loo.get_n_splits(auto)
```

```
X = auto[['horsepower']].values
y = auto['mpg'].values

rss = np.zeros(auto.shape[0])
i = 0
for train_i, test_i in loo.split(auto):
    lr = LinearRegression()
    lr = lr.fit(X[train_i], y[train_i])
    rss[i]=(lr.predict(X[test_i]) - y[test_i])**2
    i += 1
np.sum(rss)
```

```
X = auto[['horsepower', 'displacement']].values
rss = np.zeros(auto.shape[0])
i = 0
for train_i, test_i in loo.split(auto):
    lr = LinearRegression()
    lr = lr.fit(X[train_i], y[train_i])
    rss[i]=(lr.predict(X[test_i]) - y[test_i])**2
    i += 1
np.sum(rss)
```

5-fold



$$\text{RSS} = \{\text{RSS}_1 + \cdots + \text{RSS}_5\}$$

$$\begin{aligned}
\text{RSS}_1 &= \sum_{i=1}^{n/5} (y_i - \hat{y}_i)^2 \\
\text{RSS}_2 &= \sum_{i=1}^{n/5} (y_i - \hat{y}_i)^2 \\
&\vdots \\
\text{RSS}_5 &= \sum_{i=1}^{n/5} (y_i - \hat{y}_i)^2
\end{aligned}$$

5-fold

```
from sklearn.model_selection import KFold
X = auto[['horsepower', 'displacement']].values
k = 5
rss = np.zeros(k)
kf = KFold(n_splits=k, shuffle=True)
i = 0
for train_i, test_i in kf.split(auto):
    lr = LinearRegression()
    lr = lr.fit(X[train_i], y[train_i])
    rss[i]=np.sum((lr.predict(X[test_i]) - y[test_i])**2)
    i+=1
rss
```

Implement 10-fold