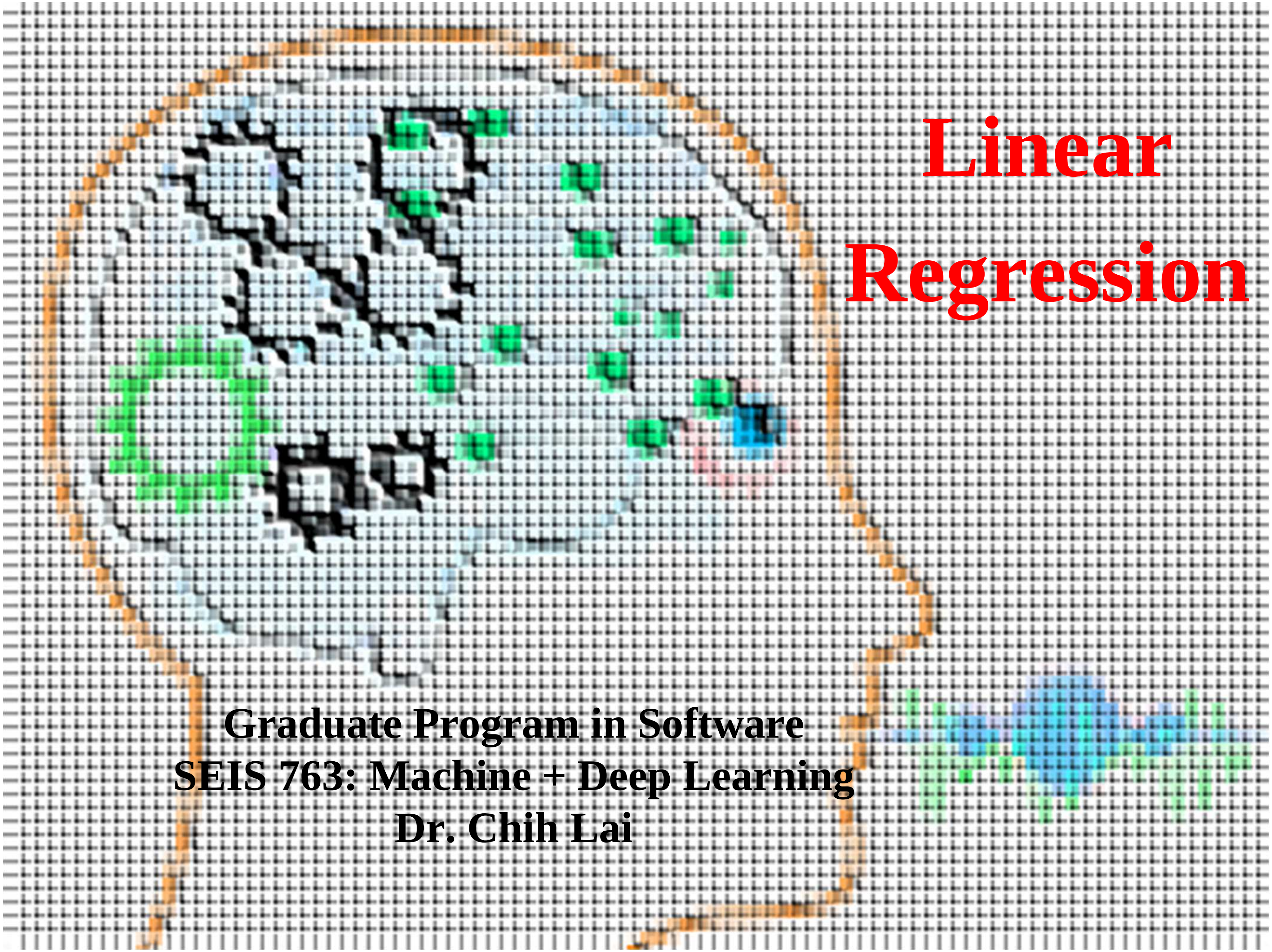


Linear Regression

Graduate Program in Software
SEIS 763: Machine + Deep Learning
Dr. Chih Lai

References to Matlab / Python LR-Related Functions

- Matlab
  - http://www.mathworks.com/help/stats/multiple-linear-regression.html
  - **LinearModel class** http://www.mathworks.com/help/stats/linearmodel-class.html

- Python
  - http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

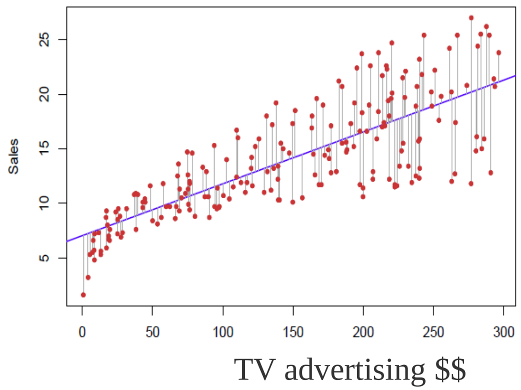
Linear Regression (LR) Concepts

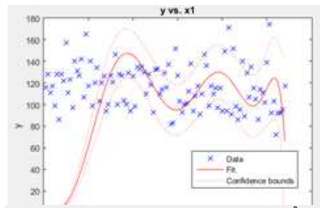
- LR models must have numerical ***responses*** (i.e. dependent vars, target features).
  - LR estimates ***linear parameters*** **(i.e. a line)** to fit data to minimize residue or error.
  - LR search/optimize ***linear parameters*** to fit data and minimize residue.

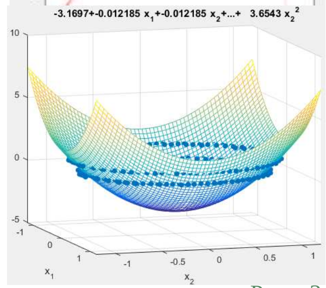
- Model relationships between a ***scalar dependent variable y*** & 1^+ variables in ***X***.
  - *Y* is our target (response, dependent), *X* is our **features** (predictors, independent vars).
  - ***Univariate*** *linear regression* has ONE independent variable.
  - ***Multivariate*** *linear regression* has 1+ independent variables

- Linear regression
  1) Build a regression model by computing coefficients θ as $\hat{y} = \theta^{\mathrm{T}}\boldsymbol{X}$
  2) Verify $\hat{y}$ against $\boldsymbol{Y}$

$$\hat{y} = \boldsymbol{b} + \boldsymbol{w}_1 \times \boldsymbol{x}_1 + \ldots + \boldsymbol{w}_n \times \boldsymbol{x}_n$$
$$\hat{y} = \boldsymbol{b} + \boldsymbol{w}\boldsymbol{X}$$
$$\boldsymbol{h}(\theta) = \hat{y} = \theta^{\mathrm{T}}\boldsymbol{X} = \theta_0 \times \mathbf{1} + \theta_1 \times \boldsymbol{x}_1 + \ldots + \theta_n \times \boldsymbol{x}_n$$







Multivariate Linear Regression

- Machine derives weights (importance) for each predictor to predict target.

X / Predictors / Features / Independent Variables

Weights = 0.08 -1.48 0.5 -0.01 9.7 ???

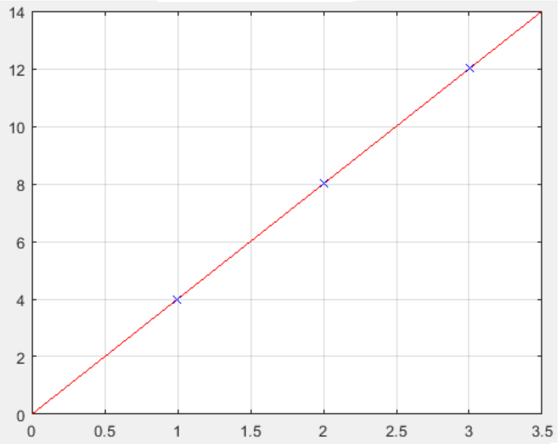
	Age	Gender	Height	Weight	Smoker	HealthStatus	Systolic
1 Smith	38	'Male'	71	176	1	'Excellent'	124
2 Johnson	43	'Male'	69	163	0	'Fair'	109
3 Williams	38	'Female'	64	131	0	'Good'	125
4 Jones	40	'Female'	67	133	0	'Fair'	117
5 Brown	49	'Female'	64	119	0	'Good'	122
6 Davis	46	'Female'	68	142	0	'Good'	121
7 Miller	33	'Female'	64	142	1	'Good'	130
8 Wilson	40	'Male'	68	180	0	'Good'	115
9 Moore	28	'Male'	68	183	0	'Excellent'	115
10 Taylor	31	'Female'	66	132	0	'Excellent'	118

Y
Target
Dependent
Response

- Initialize a random weight for each predictor, then…
- Try to reduce error (i.e. ***cost***)
  - You define "***error***".

X	Y
1	\$4
2	\$8
3	\$12

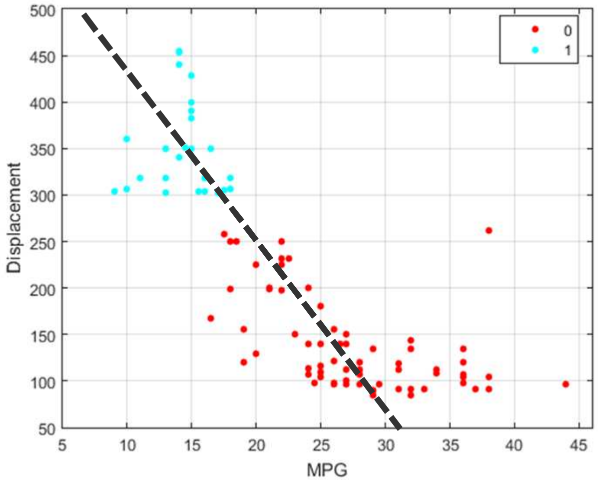
$W_1 = 4$

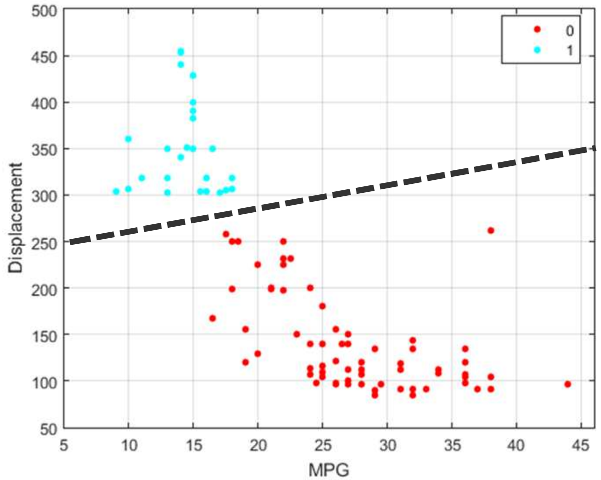


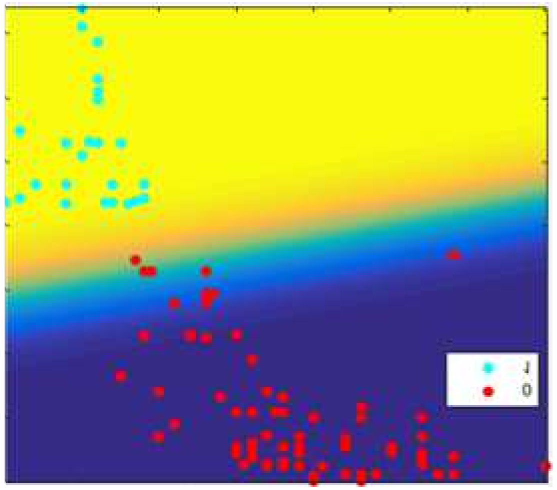
Linear Regression (LR) vs. Classification

- Linear regression
  1) Build a regression model by computing coefficients θ as $\hat{y}= \theta_0 + \theta^T \boldsymbol{X}$
  2) Verify $\hat{y}$ against $\boldsymbol{Y}$

- LR models must have **numerical** responses (i.e. dependent vars, target features).
  - If responses are categorical, use "**Classification**".
  - Classification– find a "boundary" to **separate** data points based on their classes

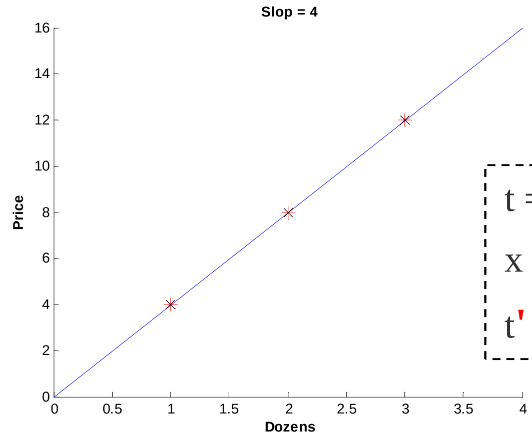






$$h(\theta) = \hat{y} = \theta^{\mathrm{T}}X = \theta_0 \times 1 + \theta_1 \times x_1 + \ldots + \theta_n \times x_n$$

- LR basic concept➔ find $h_\theta(x)$ hypothesis by estimating θ.
- Estimate the insurance cost (dependent var) from predictor (i.e. # workers)…
  - **Training** data➔ (x = 1 worker, y = $4).   (x = 2 workers, y = $8)…
  - Univariate LR: $\hat{y} = \theta_0 + \theta_1 \times x_1$
  - Assume we find slop = 4 = θ_1. (8 – 4) / (2 – 1)
  - Assume we find intercept = $\theta_0 = 0$.
  - Expected price $\hat{y} = 0 + 4X$.
  - $\theta = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$, $X = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix}$,
  - $\hat{y} = \theta^{\mathrm{T}}X = [0\ \ 4] \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} = [4\ \ 8\ \ 12]$.
  - $\theta_1 > 0$ means ↑var ➔ ↑$.
    - Add a dozen, add $\theta_1 = 4$ to $ (**positive** effect).
  - $\theta_1 < 0$ means ↑var ➔ ↓$.
  - $\theta_1 = 0$ means …

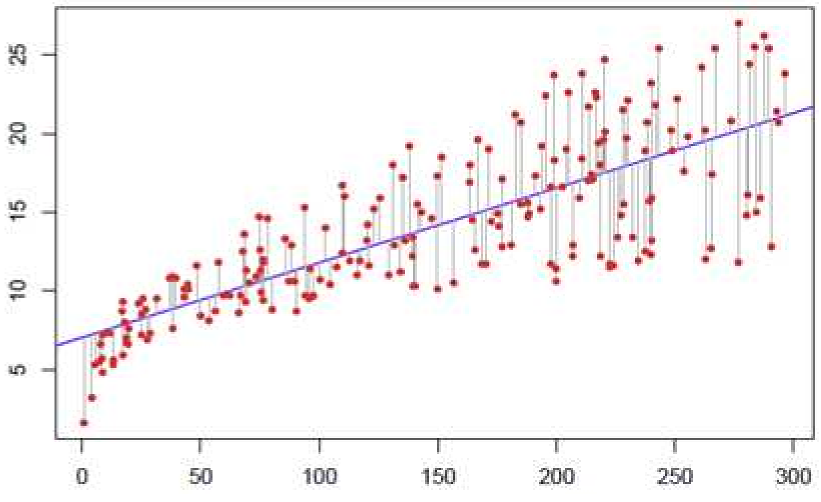



```
t = [0; 4];
x = [1 1 1; 1 2 3];
t' * x
```

Linear Regression, $\hat{y}= \theta^{T}\boldsymbol{X}$

- Linear regression
  1) Build a model by learning coefficients θ as $\hat{y} = \theta^{T}\boldsymbol{X}$ (column) or $\hat{y} = \theta\boldsymbol{X}^{T}$ (row)
  2) Verify $\hat{y}$ against $\boldsymbol{Y}$

$$\boldsymbol{h}(\theta) = \hat{y} = \theta^{T}\boldsymbol{X} = \theta_0 \times 1 + \theta_1 \times \boldsymbol{x}_1 + \ldots + \theta_n \times \boldsymbol{x}_n$$



$$\theta = \begin{array}{cl} \begin{bmatrix} 3 \\ 1 \\ 5 \end{bmatrix}, & \begin{matrix} \textbf{Base} \\ \textbf{H} \\ \textbf{W} \end{matrix} \end{array} \quad X = \begin{array}{cl} \begin{matrix} \textbf{R}_1 & \textbf{R}_2 \end{matrix} & \\ \begin{bmatrix} 2 & 6 \\ 4 & 3 \end{bmatrix} & \begin{matrix} \textbf{H} \\ \textbf{W} \end{matrix} \end{array} \quad \hat{y} = \theta^{T}\boldsymbol{X}$$

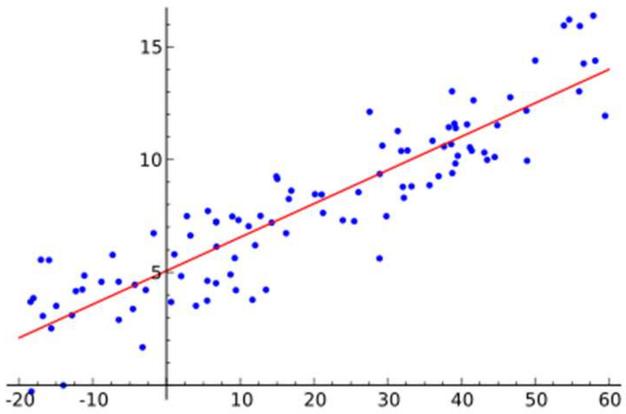
$$\theta = \begin{array}{cl} \begin{bmatrix} 3 \\ 1 \\ 5 \end{bmatrix}, & \begin{matrix} \textbf{Base} \\ \textbf{H} \\ \textbf{W} \end{matrix} \end{array} \quad X = \begin{array}{cl} \begin{matrix} \textbf{R}_1 & \textbf{R}_2 \end{matrix} & \\ \begin{bmatrix} \textbf{1} & \textbf{1} \\ 2 & 6 \\ 4 & 3 \end{bmatrix} & \begin{matrix} \textbf{Base} \\ \textbf{H} \\ \textbf{W} \end{matrix} \end{array} \quad \hat{y} = \theta^{T}\boldsymbol{X}$$

$$\theta = \begin{array}{c} \begin{matrix} \textbf{B,} & \textbf{H,} & \textbf{W} \end{matrix} \\ [3 \quad 1 \quad 5], \end{array} \quad X = \begin{array}{cl} \begin{matrix} \textbf{H,} & \textbf{W} \end{matrix} & \\ \begin{bmatrix} 2 & 4 \\ 6 & 3 \end{bmatrix} & \begin{matrix} \textbf{R}_1 \\ \textbf{R}_2 \end{matrix} \end{array} \quad \hat{y} = \theta\boldsymbol{X}^{T}$$

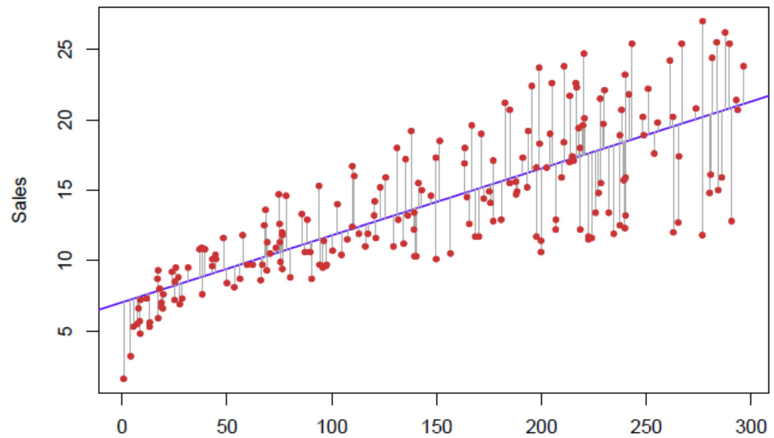
$$\theta = \begin{array}{c} \begin{matrix} \textbf{B,} & \textbf{H,} & \textbf{W} \end{matrix} \\ [3 \quad 1 \quad 5], \end{array} \quad X = \begin{array}{cl} \begin{matrix} \textbf{B,} & \textbf{H,} & \textbf{W} \end{matrix} & \\ \begin{bmatrix} \textbf{1} & 2 & 4 \\ \textbf{1} & 6 & 3 \end{bmatrix} & \begin{matrix} \textbf{R}_1 \\ \textbf{R}_2 \end{matrix} \end{array} \quad \hat{y} = \theta\boldsymbol{X}^{T}$$

Visualizing Regression

- Predicting response (indicator, dependent) variable $\boldsymbol{y}$,
  - By computing some function from known independent (attribute) variables $\boldsymbol{x_i}$.
  - $\hat{y}$ is the sum of $\boldsymbol{x_i}$ (all $\boldsymbol{p}$ attributes) w/ coefficients θ: $\hat{y} = \theta^T\boldsymbol{X} = \theta_0 \times 1 + \sum_{i=1}^{p} \theta_i x_i$
  - Understand the impact $\boldsymbol{x_i}$ on $\hat{y}$ (θ_i = magnitude and direction of $\boldsymbol{x_i}$) (**signs** of θ_i).
  - Find a line (or a plane) that minimizes the distance from points to the line.
  - Error = the sum of all those distances. Sum-of-squares of the errors = $\sum_{j=1}^{N} e_j^2$.
  - Predicting unobserved values of the response $y(x)$ for new unseen x.



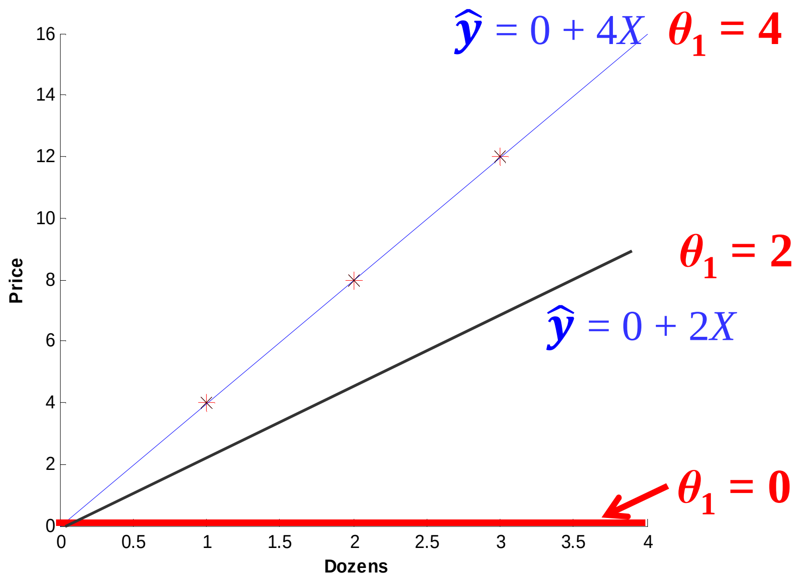
http://en.wikipedia.org/wiki/Linear_regression



TV advertising $$

Root Mean Square Errors (RMSE)

- All machine learning methods are based on measuring some kinds of ***error***.
- Measure ***square sum of error*** (SS_E) or **Residuals**.
  - $SS_E = \sum_i (y_i - \hat{y}_i)^2$ How good (or difference) LR model fits to training data?
    - **Purposes of squaring: To make error all positive.**
  - Mean Square of Errors (MSE) = $\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$.
  - Root Mean Square of Errors (***RMSE***) = $\sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$
    - $\sqrt{\quad}$ so RMSE has the same unit as predictors.



Matlab Linear Regression Function **fitlm()**

$\boldsymbol{h}(\theta) = \hat{y} = \theta^{\mathrm{T}}\boldsymbol{X}$

$\boldsymbol{h}(\theta) = \theta_0 \times \mathbf{1} + \theta_1 \times \mathbf{x}_1 + \ldots + \theta_n \times \mathbf{x}_n$

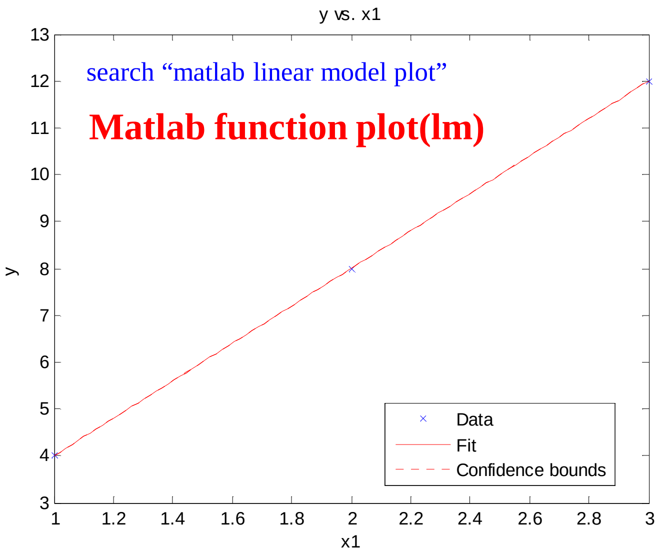
- Use Matlab function **fitlm()** to build a LR model.
  - To estimate θ for the LR model.
  - To measure ***square sum of error*** (***SS_E***) or **Residuals**.
    - $\boldsymbol{SS_E} = \sum_i (y_i - \hat{y}_i)^2$ How much difference between LR model and training data?
    - Mean Square of Errors (MSE) = $\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$.
    - Root Mean Square of Errors (***RMSE***) = $\sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$ = RMSD.

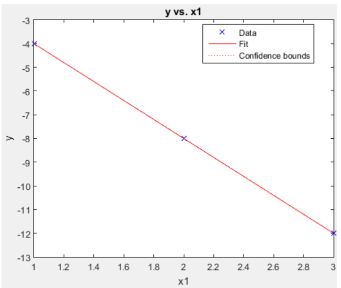
<u>**% inputs must be row vector**</u>

```
% rows = records, columns = vars
X = [1 2 3]'; Y=[4 8 12]';
lm = fitlm(X, Y)
plot(lm)
```

```
regr = linear_model.LinearRegression()
regr.fit(x, y)
```

R: fit <- lm(y ~ x1 + x2 + x3, data=mydata)



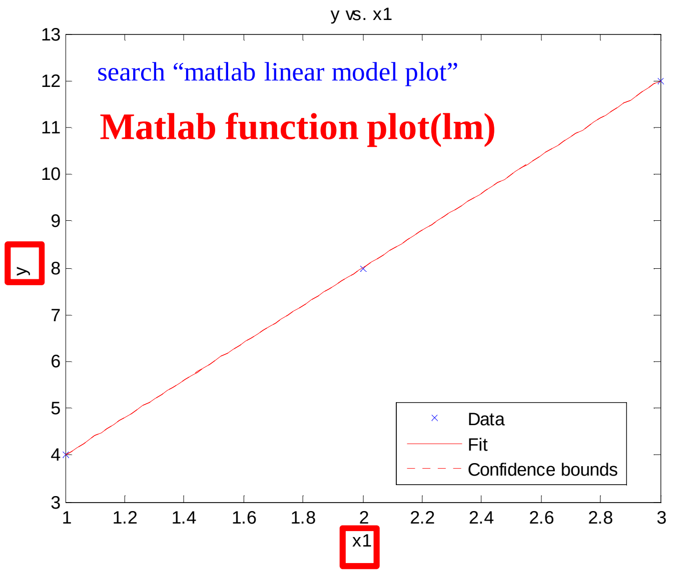



```
X = [1 2 3]';
Y=[-4 -8 -12]';
lm = fitlm(X, Y)
plot(lm)
```

Output from Matlab fitlm() Function

- A standard error column for each estimated coefficient.
- If ***p***Value derived from *t* test for $\boldsymbol{x}_1$ is very small➔ Good predictors to ***y***.
- If ***p***Value > 0.01➔ Not a good predictor to ***y***.
- R^2, adjusted R^2, and *F* statistics.
- ***RMSE*** $= \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$.

```
X = [1 2 3]'; Y=[4 8 12]';
lm = fitlm(X, Y)
plot(lm)
```



Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	-1.3567e-15	3.6419e-07	-3.7253e-09	1
x1	4	1.6859e-07	2.3727e+07	2.6832e-08

Number of observations: 3, Error degrees of freedom: 1

Root Mean Squared Error: 2.38e-07

R-squared: 1, Adjusted R-Squared 1

F-statistic vs. constant model: 5.63e+14, p-value = 2.68e-08

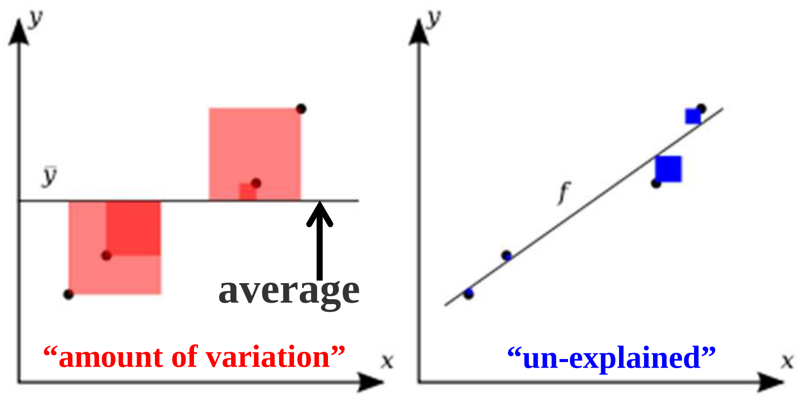
(see Appendix for details)

R-Square, R^2 (Coefficient of Determination)

- **$0 \le R^2 \le 1$** **(??)**
  - Blue squares (area) = squared residuals against ***LR*** = **SSE** = $\sum_i (y_i - \hat{y}_i)^2$
  - Red squares (area) = squared residuals against ***average*** = **SST** (total) = $\sum_i (y_i - \bar{y})^2$
  - $R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{unexplaned}{amount\ of\ variation}$
  - Smaller residual (LR fits better to data) comparing to average ➔ R^2 closer to 1.

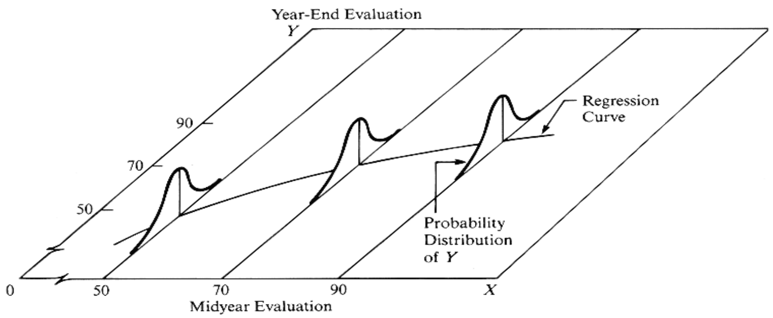
```
Estimated Coefficients:
                   Estimate        SE           tStat        pValue
                 ___________   __________   ___________   ___________

  (Intercept)    -1.3567e-15   3.6419e-07   -3.7253e-09             1
   x1                      4   1.6859e-07    2.3727e+07    2.6832e-08

Number of observations: 3, Error degrees of freedom: 1
Root Mean Squared Error: 2.38e-07
R-squared: 1,  Adjusted R-Squared 1
F-statistic vs. constant model: 5.63e+14, p-value = 2.68e-08
```

(see Appendix for details)



http://en.wikipedia.org/wiki/Coefficient_of_determination

http://www.unc.edu/~nielsen/soci709/m1/m1004.gif

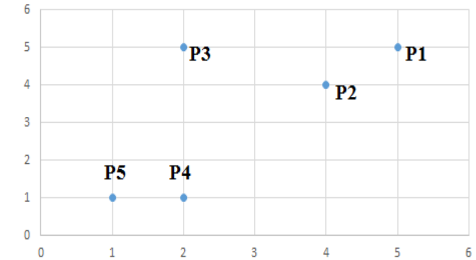


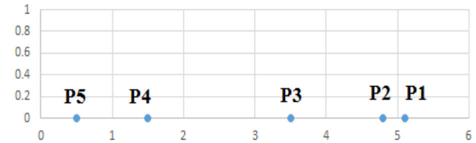
Adjusted R-Square

- $0 \le R^2 \le 1$ $\quad\quad R^2 = 1 - \frac{SSE}{SST}$
  - Smaller residual (LR fits better to data) comparing to average➔ R^2 closer to 1.
  - Tell us how many points fall within the line of estimated regression equation.

- Adjusted $\boldsymbol{R}^2$.
  - **Def 1** $= 1 - \frac{(1-R^2)(n-1)}{n-k-1}$. $\quad\quad$ **Def 2** $= 1 - \frac{n-1}{n-p} \times \frac{SSE}{SST}$
    - Adj_ $\boldsymbol{R}^2$↓ if adding $\boldsymbol{k}$ predictors in the model. $\quad\quad$ $\boldsymbol{k}$↑➔Def1↓ unless $\boldsymbol{R}^2$↑.
    - Adj_ $\boldsymbol{R}^2$ is normalized by $\boldsymbol{p}$ predictors for comparing models w/ different # of predictors.

- Why Adjusted R^2?
  - R^2 increases w/ every added predictor
    - more model wiggle room to reduce explained variations).
  - As R^2 always increases & never decreases.
  - Model appear better fit w/ more predictors. ➔ misleading.
  - A model w/ too many predictors ➔ **overfitting**.



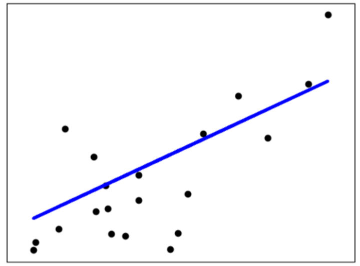


Linear Regression in scikit-learn

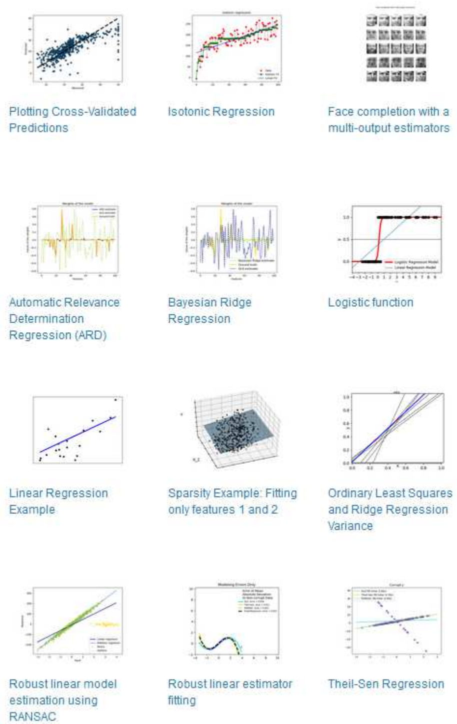
- http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score

# Load dataset……
# Create linear regression object
regr = linear_model.LinearRegression()

# Train the model using the training sets
regr.fit(X_train, y_train)
# Make predictions using the testing set
y_pred = regr.predict(X_test)
```

```
# The coefficients
print('Coefficients: \n', regr.coef_ )
# The mean squared error
print("MSE: %.2f"  %mean_squared_error(y_test, y_pred))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % r2_score(y_test, y_pred))

# Plot outputs
plt.scatter(X_test, y_test,  color='black')
plt.plot(X_test, y_pred, color='blue', linewidth=3)

plt.xticks(());         plt.yticks(());     plt.show()
```



http://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html

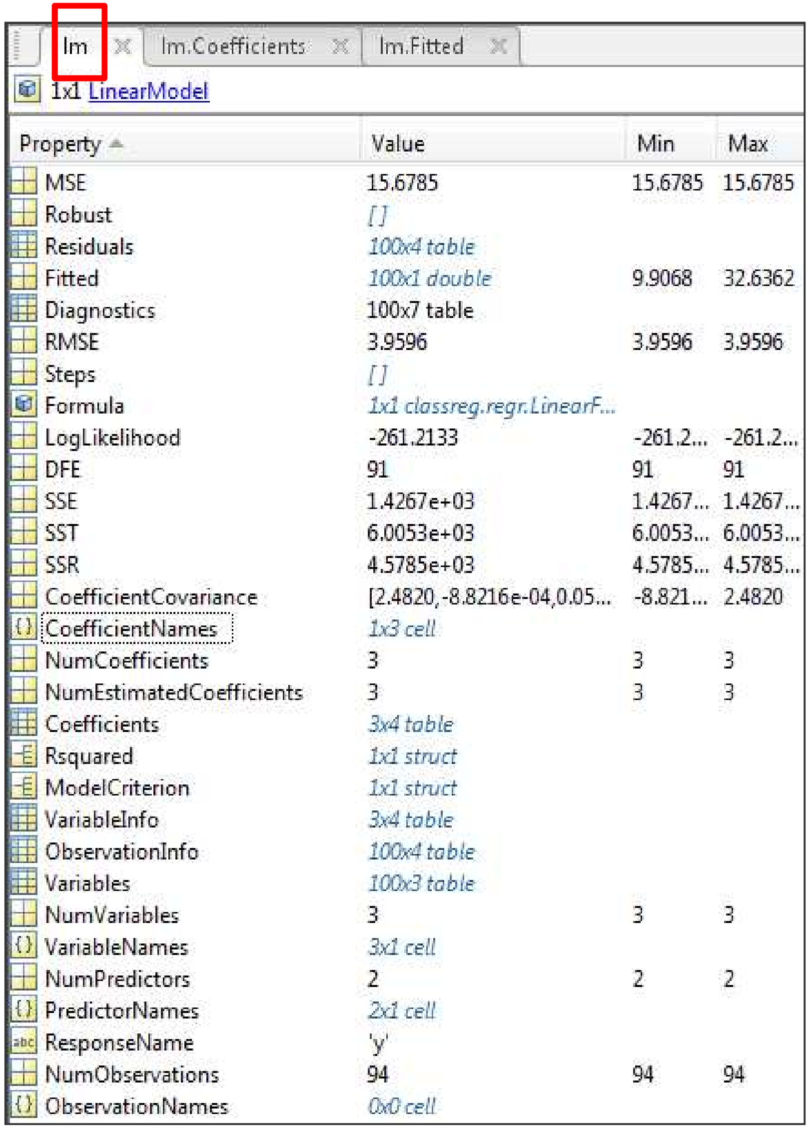


Detailed Information Returned from Matlab LR

- http://www.mathworks.com/help/stats/linearmodel-class.html

```
X = [1 2 3]'; Y=[4 8 12]';

lm = fitlm(X, Y)
```

- **MSE**
- **Residuals, .Raw**
- **Fitted**
- **SSE, SST, SSR**
- **Coefficients .Estimate**
- **Rsquared .Ordinary .Adjusted**
- **Diagnostics, .Leverage .CooksDistance**
- **LogLikelihood**



lm | lm.Coefficients | lm.Fitted

1x1 LinearModel

Property	Value	Min	Max
MSE	15.6785	15.6785	15.6785
Robust	[]		
Residuals	100x4 table		
Fitted	100x1 double	9.9068	32.6362
Diagnostics	100x7 table		
RMSE	3.9596	3.9596	3.9596
Steps	[]		
Formula	1x1 classreg.regr.LinearF...		
LogLikelihood	-261.2133	-261.2...	-261.2...
DFE	91	91	91
SSE	1.4267e+03	1.4267...	1.4267...
SST	6.0053e+03	6.0053...	6.0053...
SSR	4.5785e+03	4.5785...	4.5785...
CoefficientCovariance	[2.4820,-8.8216e-04,0.05...	-8.821...	2.4820
CoefficientNames	1x3 cell		
NumCoefficients	3	3	3
NumEstimatedCoefficients	3	3	3
Coefficients	3x4 table		
Rsquared	1x1 struct		
ModelCriterion	1x1 struct		
VariableInfo	3x4 table		
ObservationInfo	100x4 table		
Variables	100x3 table		
NumVariables	3	3	3
VariableNames	3x1 cell		
NumPredictors	2	2	2
PredictorNames	2x1 cell		
ResponseName	'y'		
NumObservations	94	94	94
ObservationNames	0x0 cell		

Prediction **AFTER** Building An LR Model

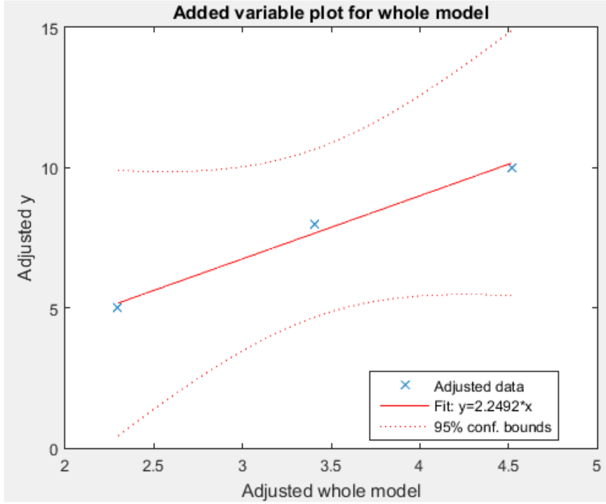
```
X = [1 2 3]';
Y = [4 8 12]';                        % Y=[5 8 10];

mdl = fitlm(X, Y),
testX = [2.5,  3]';          % testX must be raw vector. Raws are records, Cols are vars.


% method 1
[newY1 Conf] = predict(mdl, testX);     % http://www.mathworks.com/help/stats/linearmodel.predict.html


% method 2
newY2 = feval(mdl, testX)


% method 3
%% Compute prediction yourself, θ^T X
newY3 = (mdl.Coefficients.Estimate)' * [ones(1, 2);    testX' ]


% check yourself,       newY1 == newY2 == newY3 = [10,   12]
```

$$\theta = \begin{bmatrix} \approx 0 \\ 4 \end{bmatrix}, \quad X = \begin{bmatrix} \mathbf{1} & \mathbf{1} \\ 2.5 & 3 \end{bmatrix}, \ \hat{y} = \theta^{\mathbf{T}}\boldsymbol{X} = [0 \ \ 4] \times \begin{bmatrix} \mathbf{1} & \mathbf{1} \\ 2.5 & 3 \end{bmatrix}.$$

T = [0;4]; x = [1 1; 2.5 3];   t'*x

Visualizing Higher Dimension LR

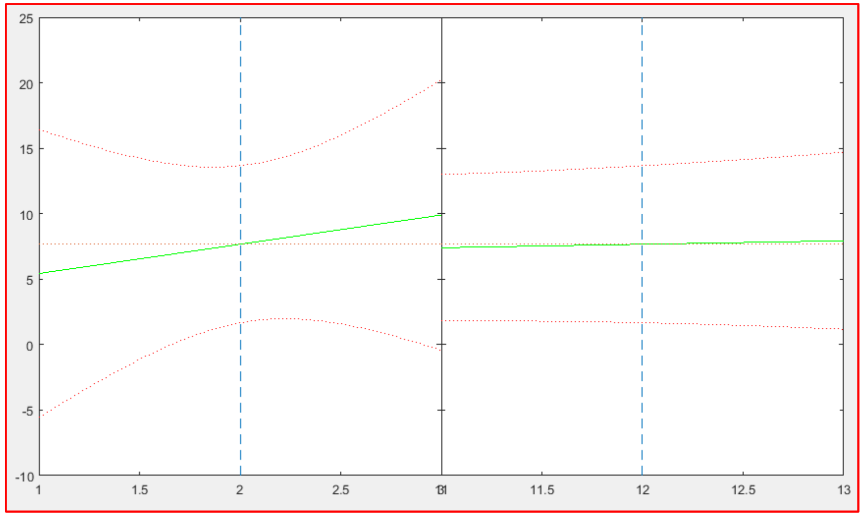
- Shows $\hat{y}$ against ***adjusted*** predictors. (Adjusted Response Plot)
  - Predictors are averaged out by the averaged $\hat{y}$.
  - Adjusted data points are computed by adding the residual to $\hat{y}$ for each observation.



- Which predictor is more significant? **plotSlice()**

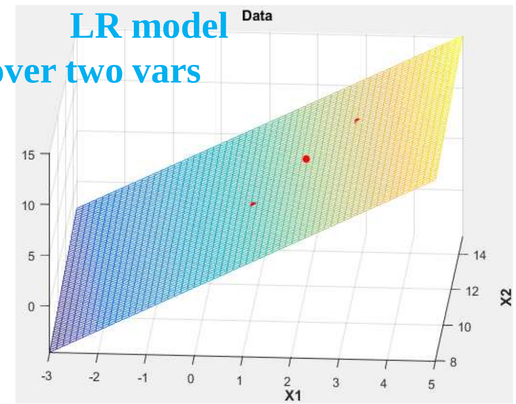
```
X2 = [1 2 3]'; Y2 = [5 8 10]';
X2=[X2,   X2+10]            % 2-D predictors!!!
lm2 = fitlm(X2, Y2)
figure,    plot(lm2),
plotSlice(lm2)
figure, subplot(1,2,1), plotAdjustedResponse(lm2, 1),
subplot(1,2,2), plotAdjustedResponse(lm2, 2)
```

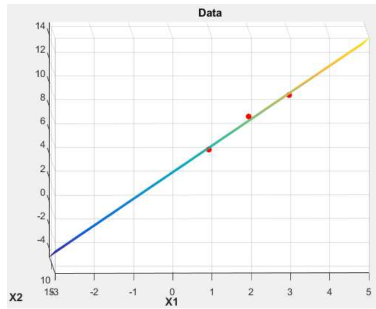
$$\begin{bmatrix} 1 & 11 \\ 2 & 12 \\ 3 & 13 \end{bmatrix}$$



```
b = num2str(lm2.Coefficients.Estimate);
figure,          syms x1 x2;                    hold on,
% 3-D plot
% 0+ 4.0745*x1+-1.5745*x2
ezmesh([b(1,:) '+' b(2,:) '*x1' '+' b(3,:) '*x2'], [-3 5 8 15])
scatter3(X2(:, 1), X2(:, 2), Y2, 'filled', 'r'),     hold off
title('\bf Data'),  xlabel('\bf X1'), ylabel('\bf X2')
```

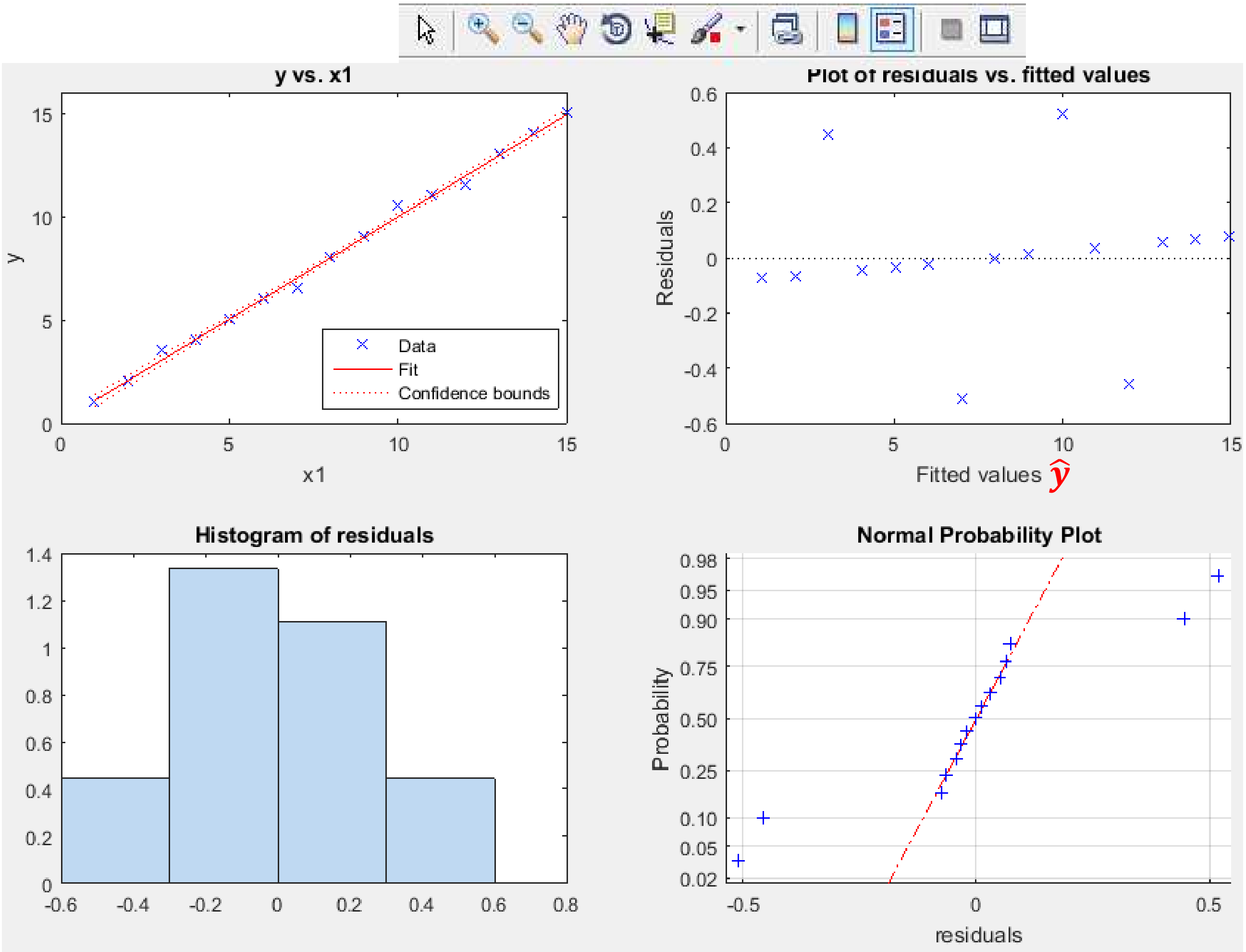
LR model over two vars





Normal Probability Plot of Residual Distribution

- Norm-Plot can be used to spot potential **outliers**.
  - A solid line connects the 25th and 75th percentiles in the data.

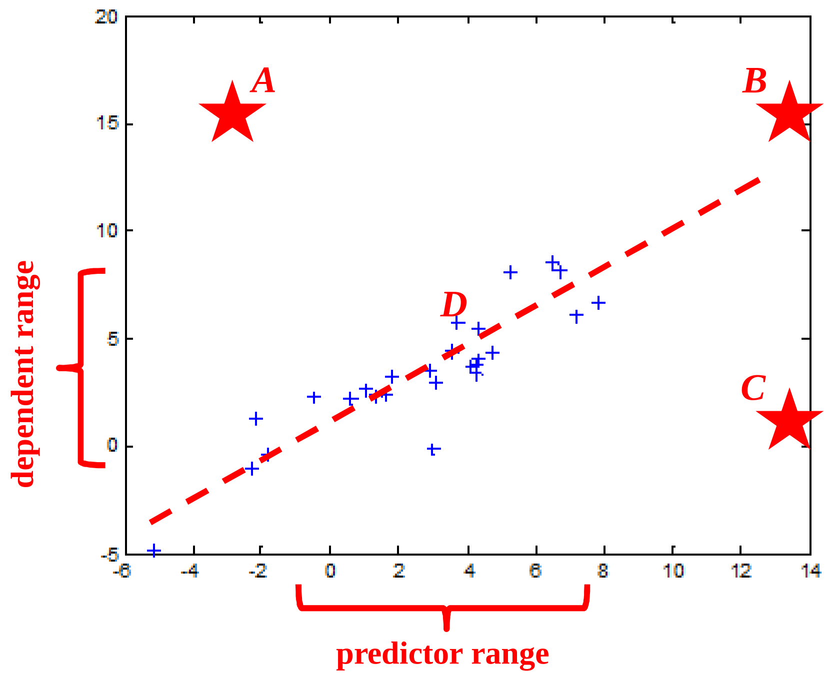



```
x=[1:15]; y=[1:15];
y(3)=3.5;      y(7)=6.5;
y(10)=10.5;    y(12)=11.5;
mdl = fitlm(x, y)
figure, subplot(2,2,1),
plot(mdl)
subplot(2,2,2),
plotResiduals(mdl, 'fitted')
subplot(2,2,3),
plotResiduals(mdl)
subplot(2,2,4),
normplot(mdl.Residuals.Raw)
xlabel('residuals')
%plotResiduals(mdl, 'probability')
```

plotResiduals() ➔ http://www.mathworks.com/help/stats/linearmodel.plotresiduals.html

Diagnostics

- Measure how each data $\boldsymbol{x_i}$ influence the LR model?
  - ***Leverage***,
  - ***Cooks Distance*** (***Influence***).
  - **See Appendix for details.**

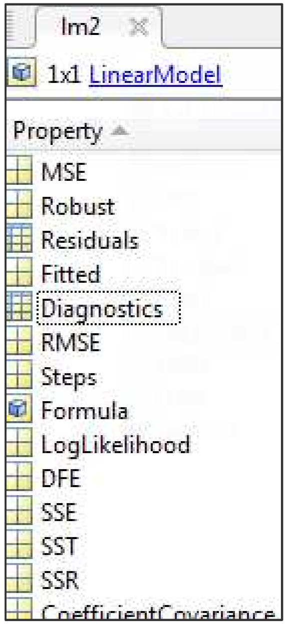


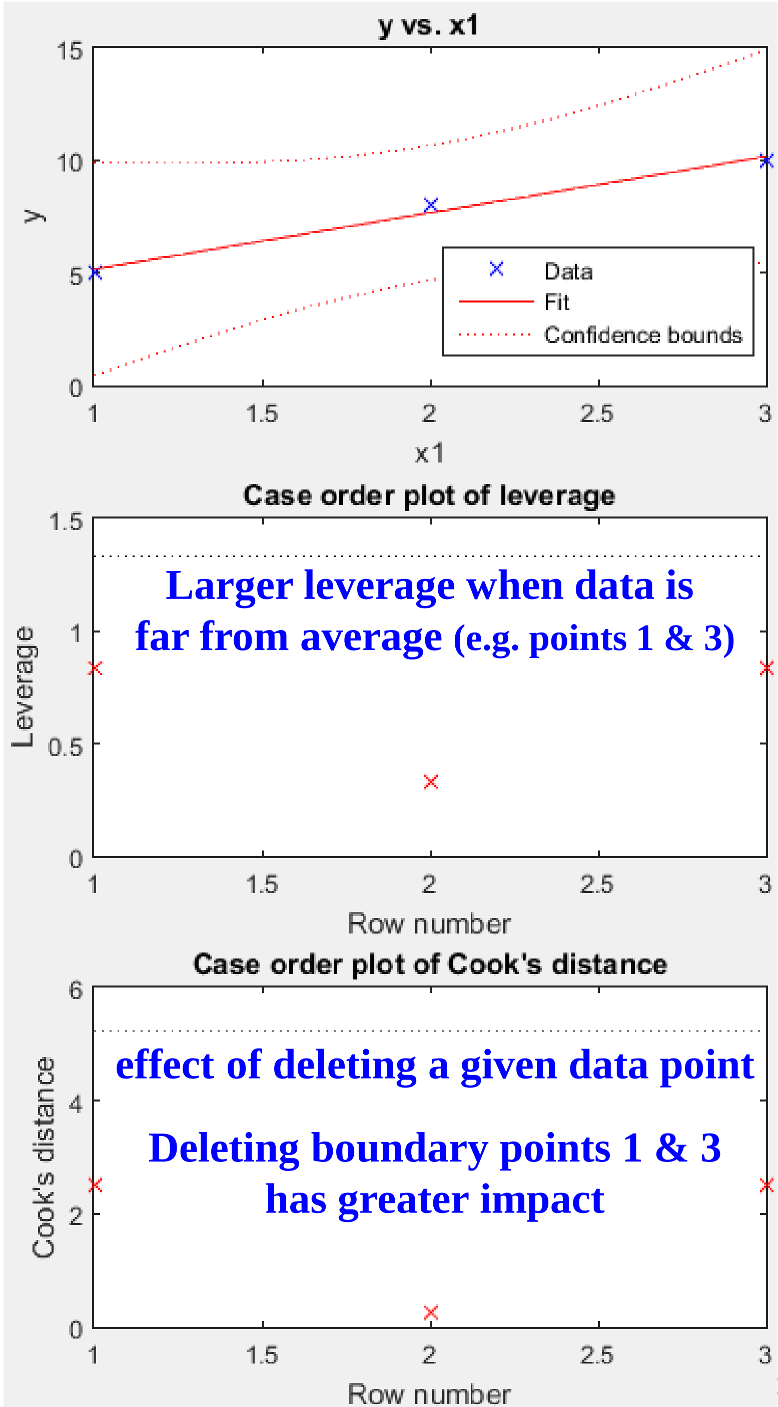
- Simple intuitive explanation…
  - Leverage **≈** $|x_i - \bar{x}|$ = how much predictor vars diff from mean of predictor vars.
    - ***D*** very close to the μ of predictor, almost no impact to LR.
    - ***A***, ***C***, ***B*** very far from the μ of predictor, **potential** impact to (the **slope** of) LR.
  - Cooks distance (influence) **≈** measures the effect of deleting a given observation.
    - ***A*** & ***C*** has large error (far from LR line). ***B*** & ***D*** has small error (close to LR line).
    - Shows the influence of each observation to the fitted response (predicted $\hat{y}$).
    - **A point likely be an outlier IF its Cook's distance > (3 × average Cook's distance).**

Leverage & Cooks Distance

- **mdl.Diagnostics.Leverage**
- **mdl.Diagnostics.CooksDistance**

```
X2 = [1 2 3]';    Y2=[5 8 10]';
mdl = fitlm(X2, Y2);
figure,
subplot(3,1,1),    plot(mdl)
subplot(3,1,2),
plotDiagnostics(mdl)
subplot(3,1,3),
plotDiagnostics(mdl,'cookd')
```





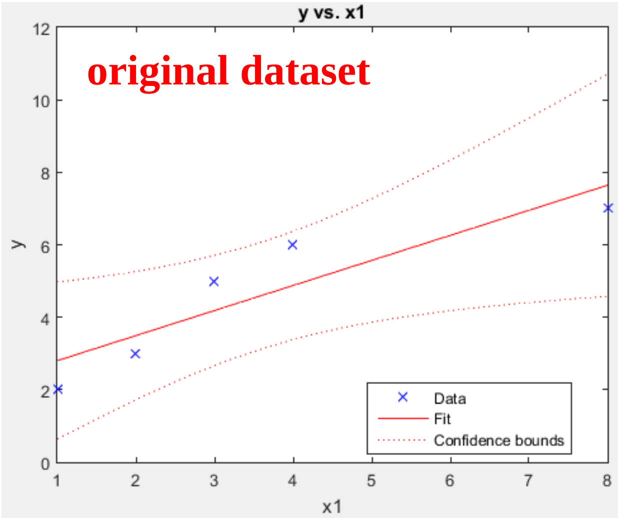
- statsmodels.stats.outliers_influence.OLSInfluence

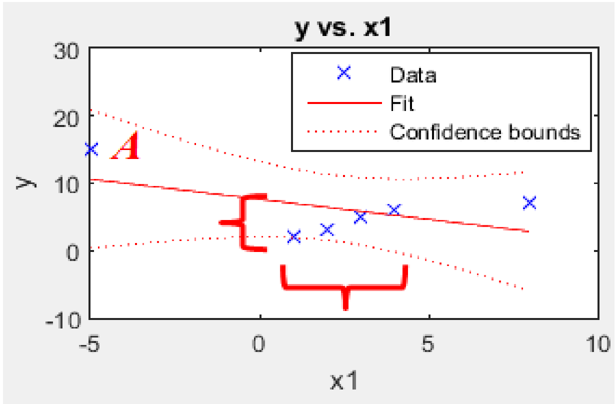
```
cooks_distance()
cov_ratio()
det_cov_params_not_obsi()
dfbetas()
dffits()
```

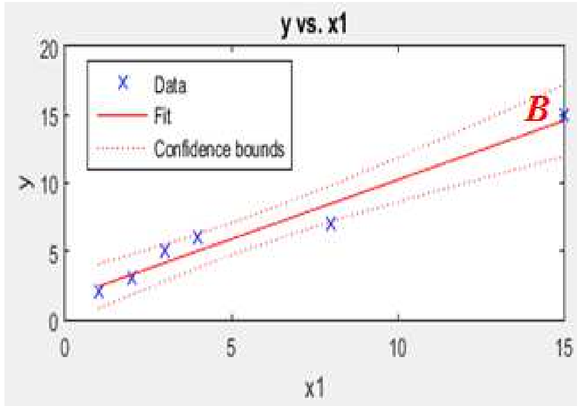
http://www.statsmodels.org/devel/generated/statsmodels.stats.outliers_influence.OLSInfluence.html

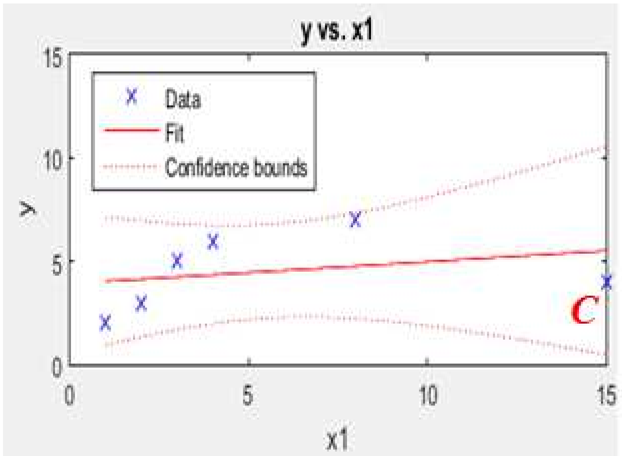
Impact of Outliers w.r.t. Leverage and Distance

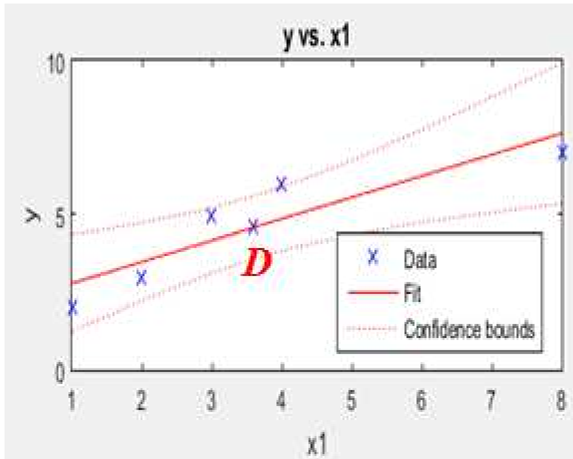
- Two influences of each data $\boldsymbol{x_i}$ on LR include:
  - Leverage $\approx |x_i - \bar{x}|$ = how much predictor vars diff from mean of predictor vars.
  - Distance $\approx |y_i - \hat{y}_i|$ = measures the effect of deleting a given observation.









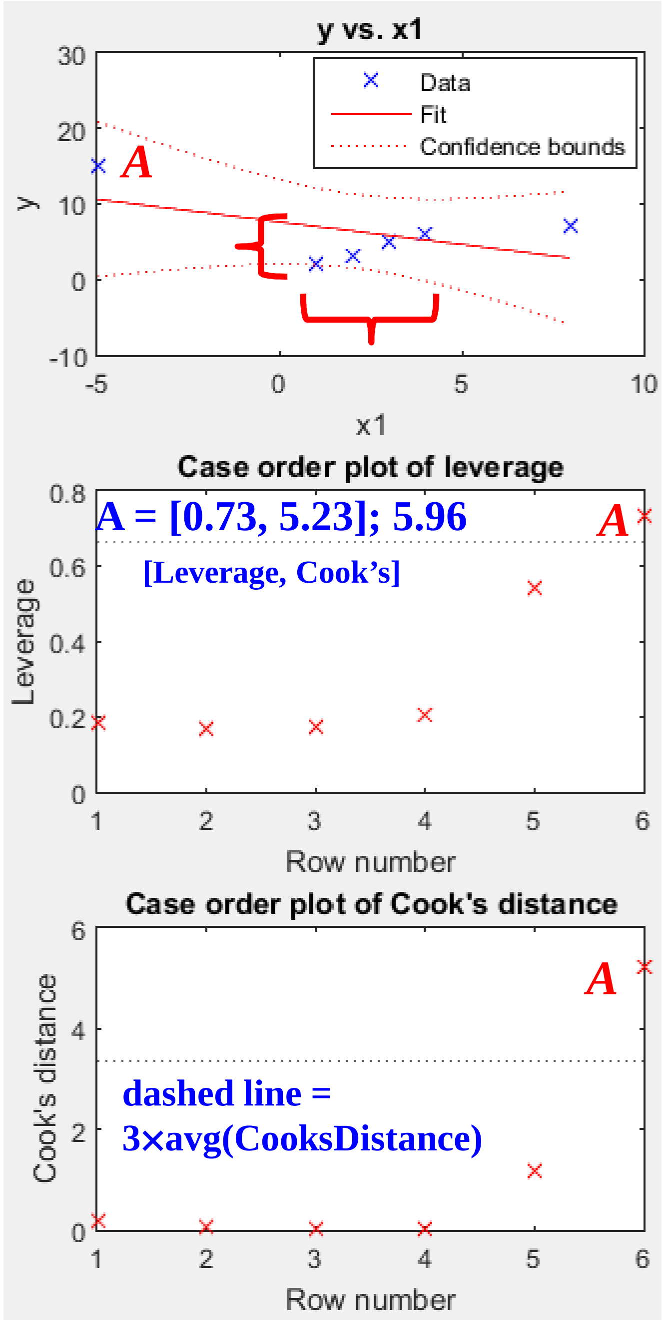


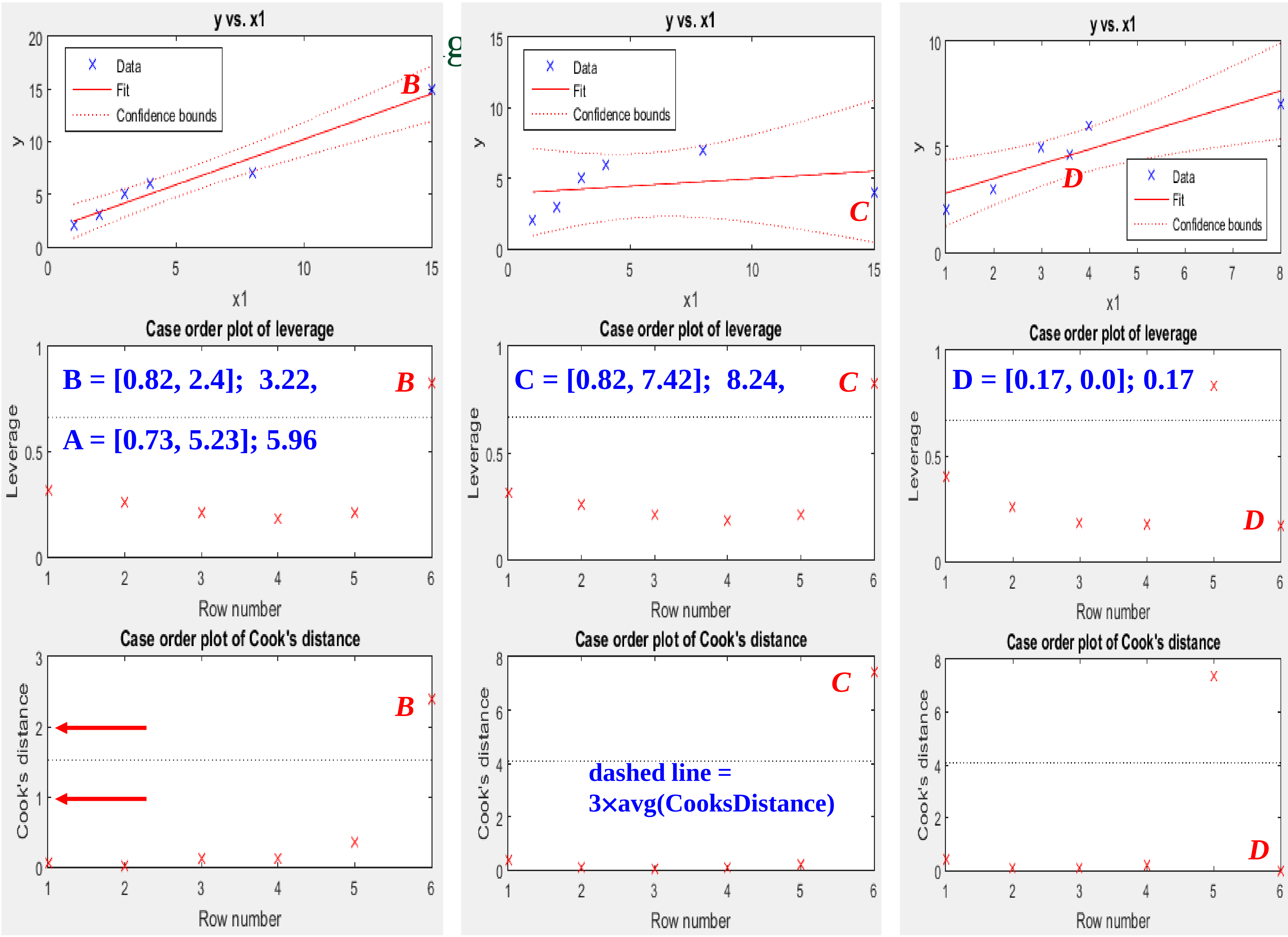
http://onlinestatbook.com/2/regression/influential.html

Diagnoses of Point A

X	Y
1	2
2	3
3	5
4	6
8	7

```
X = [1; 2; 3; 4; 8];
Y = [2; 3; 5; 6; 7];
X(end + 1) = -5;     Y(end + 1) = 15;          % A
%X(end + 1) = 15;    Y(end + 1) = 15;          % B
%X(end + 1) = 15;    Y(end + 1) = 4;           % C
%X(end + 1) = 3.6;   Y(end + 1) = 4.6;         % D

mdl = fitlm(X, Y)
plot(mdl)
figure,
subplot(3,1,1),
plot(mdl)
subplot(3,1,2),
plotDiagnostics(mdl)
subplot(3,1,3),
plotDiagnostics(mdl,'cookd')

mdl.Diagnostics                     % print Diagnostics info.
```




y vs. x1
Data
Fit
Confidence bounds
B
C
D
x1
y
Case order plot of leverage
B = [0.82, 2.4]; 3.22,
A = [0.73, 5.23]; 5.96
C = [0.82, 7.42]; 8.24,
D = [0.17, 0.0]; 0.17
Leverage
Row number
Case order plot of Cook's distance
Cook's distance
dashed line =
3×avg(CooksDistance)

Matlab Summary of LR Diagnoses

Summary of Output and Diagnostic Statistics

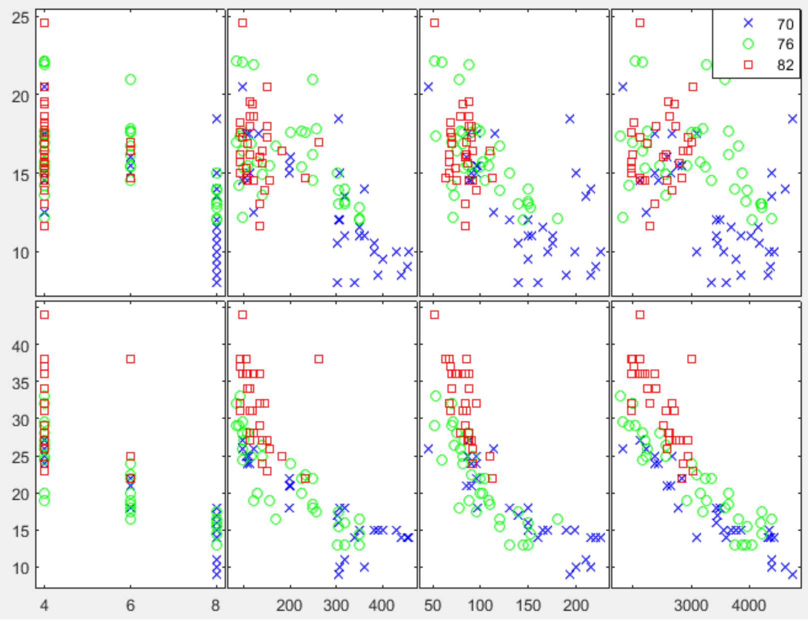
Name	LinearModel	regstats
"Cook's Distance" on page 9-70	`CooksDistance` and `cookd`	`cookd`
"Coefficient Confidence Intervals" on page 9-75	`coefCI`	N/A
"Coefficient Covariance and Standard Errors" on page 9-74	`CoefficientCovariance`	`covb`
"Coefficient of Determination (R-Squared)" on page 9-78	`Rsquared`: `Ordinary`, `Adjusted`	`rsquare`, `adjrsquare`
"Delete-1 Change in Covariance (covratio)" on page 9-81	`CovRatio`	`covratio`
"Delete-1 Scaled Difference in Coefficient Estimates (Dfbetas)" on page 9-84	`Dfbetas`	`dfbetas`
"Delete-1 Scaled Change in Fitted Values (Dffits)" on page 9-85	`Dffits`	`dffits`
"Delete-1 Variance (S2_i)" on page 9-88	`S2_i`	`s2_i`
"Durbin-Watson Test" on page 9-91	`dwtest`	`dwstat`
"F-statistic" on page 9-93	`Fstat`	`fstat`
"Hat Matrix" on page 9-99	`HatMatrix`	`hatmat`
"Leverage" on page 9-100	`Leverage`	`leverage`
"Residuals" on page 9-103	`Residuals`: `Raw`, `Pearson`, `Studentized`, `Standardized`	`r`, `studres`, `standres`
"t-statistic" on page 9-96	`tstats`	`tstat`

Predict MPG… First, Understand Data

- gplotmatrix()– a matrix of scatter plots.
  - Each axis contains a scatter plot of predictors ***X*** against response(s) ***Y***.
- More clear relationship between displacement, horsepower, weight, **to MPG**.
  - Newer cars tend to be lighter and have better MPG than older cars.

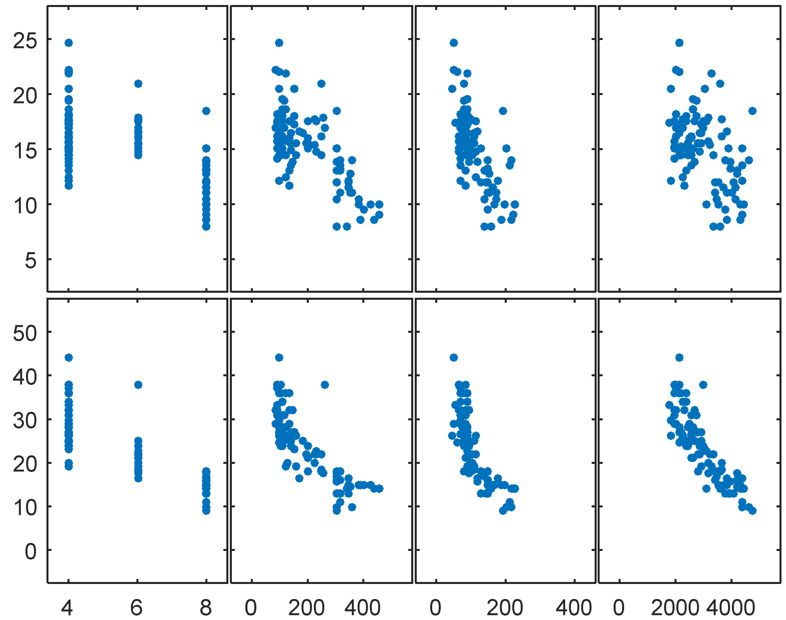
```
load carsmall
xvars = [Cylinders Displacement Horsepower Weight];
yvars = [Acceleration MPG];
figure, gplotmatrix(xvars,yvars,Model_Year,'','xos')
figure, plotmatrix(xvars, yvars)
```

Cylinders Displacement Horsepower Weight



Acceleration

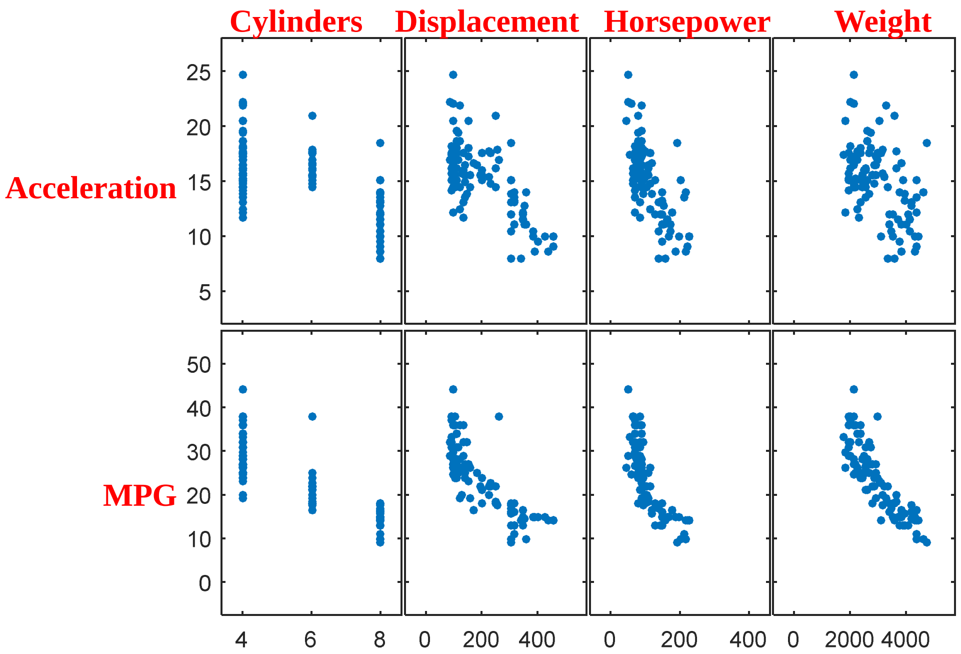
MPG



Forming Strategies

- So, we are going to build an LR model
  - To explain MPG.
  - Based on Displacement, Horsepower, and Weight.

```
load carsmall
X = [Displacement Horsepower Weight];
Y = [MPG];
mdl = fitlm(X, Y)
Plot_LR_Figures(mdl)
```

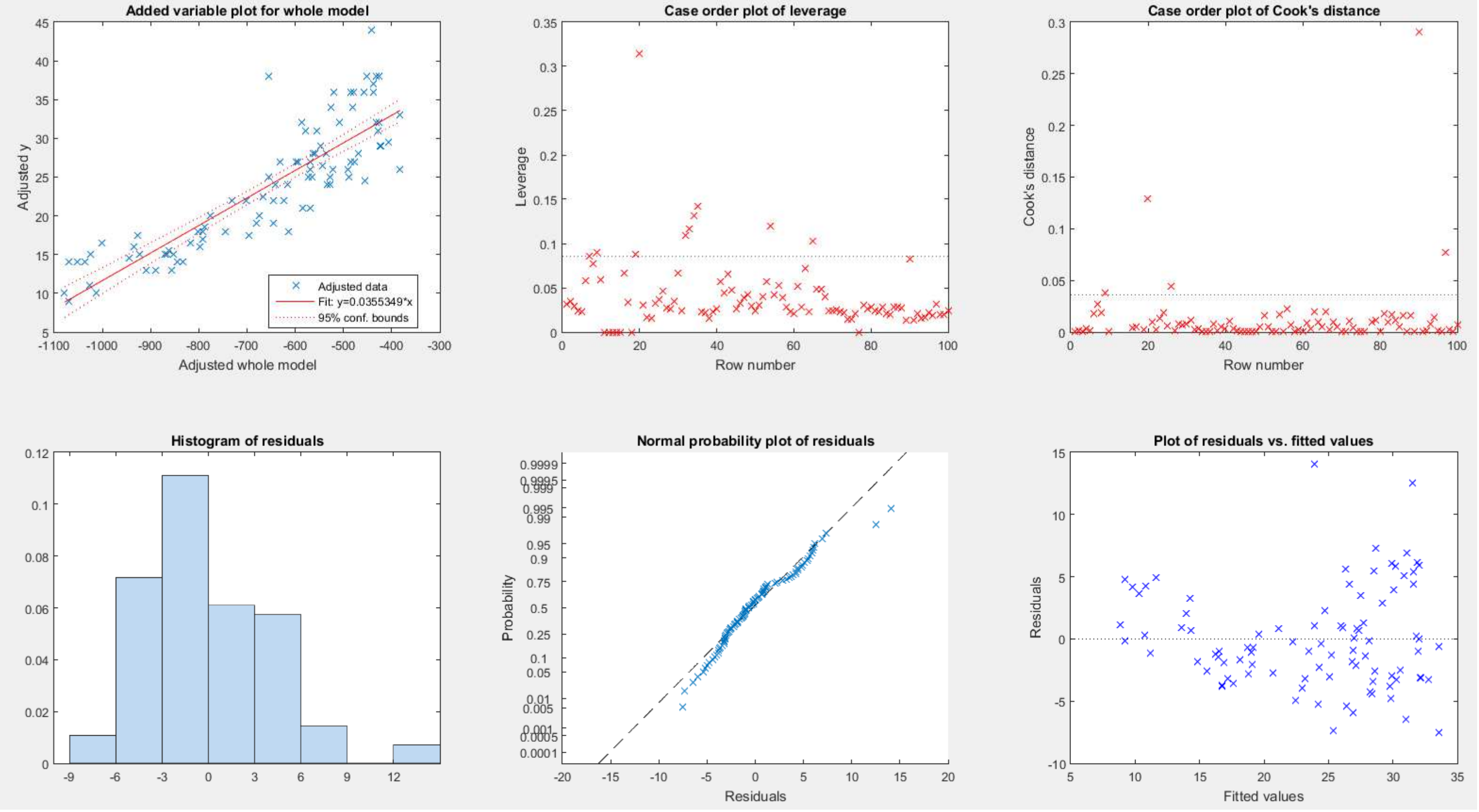



```
function Plot_LR_Figures(mdl)
figure,
subplot(2, 3, 1), plot(mdl),
subplot(2, 3, 2), plotDiagnostics(mdl),
subplot(2, 3, 3), plotDiagnostics(mdl,'cookd')
subplot(2, 3, 4), plotResiduals(mdl),
subplot(2, 3, 5), plotResiduals(mdl, 'probability')
% above probability plot = normplot(abs(mdl.Residuals.Raw))
subplot(2, 3, 6), plotResiduals(mdl, 'fitted')

figure,
for i = 1 : mdl.NumCoefficients,
  subplot(mdl.NumCoefficients, 1, i),
  plotAdded(mdl, mdl.CoefficientNames{i});
end
```

1^{st} Try, Result 1

- PRG in the carsmall.m program file.
- Based on Displacement, Horsepower, and Weight.
- Displacement has largest p-value.
- Two likely outliers from the histogram & probability plots.

```
Estimated Coefficients:
                       Estimate        SE         tStat       pValue
                       __________   _________   ________   __________

    (Intercept)            47.182      2.0973     22.497   1.7329e-38
    x1    disp.        -0.0053631    0.010574   -0.50719      0.61328
    x2    H.P.          -0.034562    0.023824    -1.4507      0.15037
    x3    W.           -0.0062775   0.0011978    -5.2409   1.0646e-06

Number of observations: 93, Error degrees of freedom: 89
Root Mean Squared Error: 4.08
R-squared: 0.753,  Adjusted R-Squared 0.744
F-statistic vs. constant model: 90.3, p-value = 6.51e-27
```



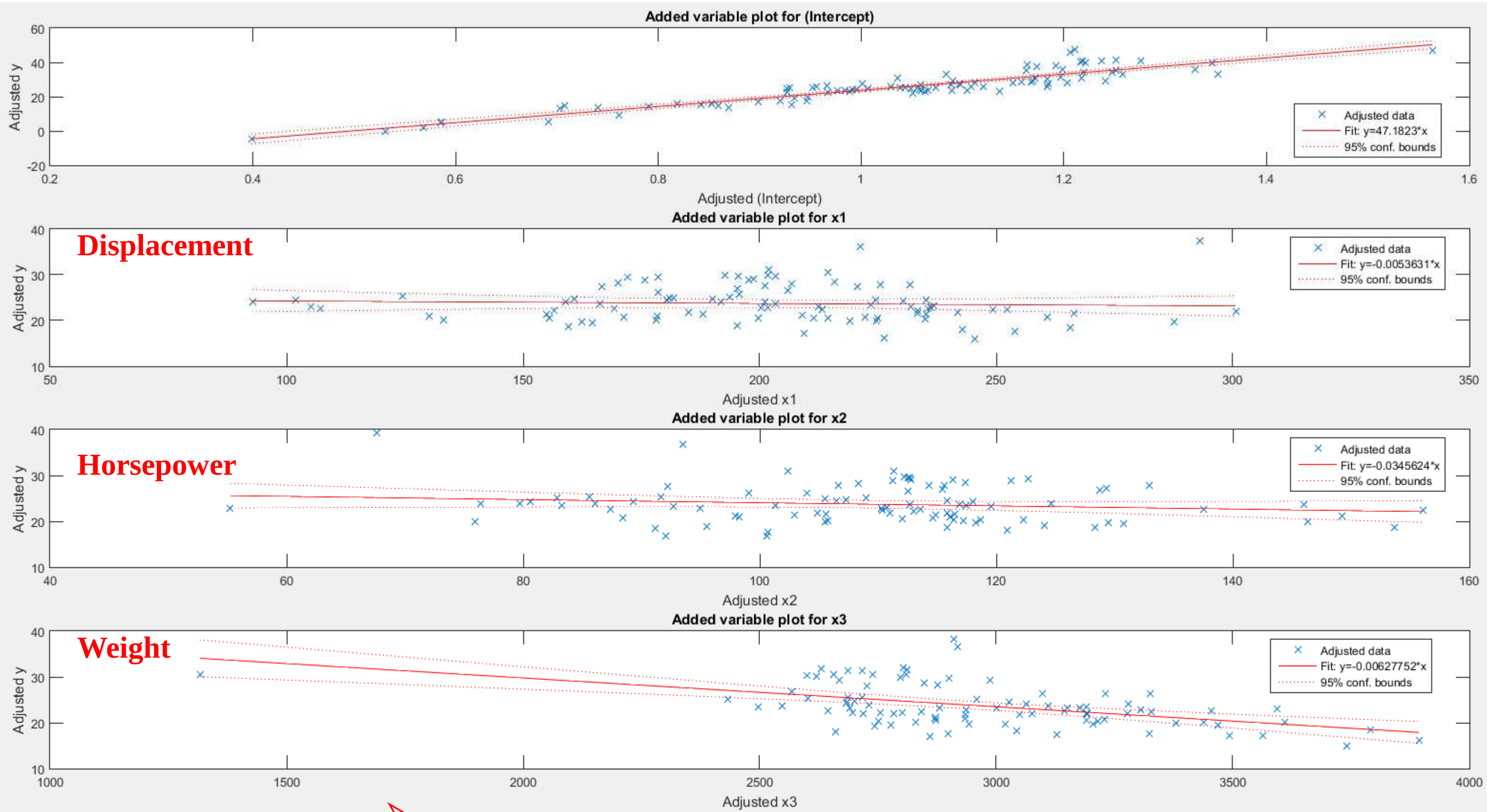
1^{st} Try, Result 2

- PRG in the carsmall.m program file.
- Based on Displacement, Horsepower, and Weight.
- **Displacement has largest p-value.**
- **Displacement has flat Added Plot.**
  - **See Appendix for details.**

Estimated Coefficients:

		Estimate	SE	tStat	pValue
(Intercept)		47.182	2.0973	22.497	1.7329e-38
x1	disp.	-0.0053631	0.010574	-0.50719	0.61328
x2	H.P.	-0.034562	0.023824	-1.4507	0.15037
x3	W.	-0.0062775	0.0011978	-5.2409	1.0646e-06

Number of observations: 93, Error degrees of freedom: 89
Root Mean Squared Error: 4.08
R-squared: 0.753, Adjusted R-Squared 0.744
F-statistic vs. constant model: 90.3, p-value = 6.51e-27



➢ **If a line i near horizontal, then variable x_i is less significant.**

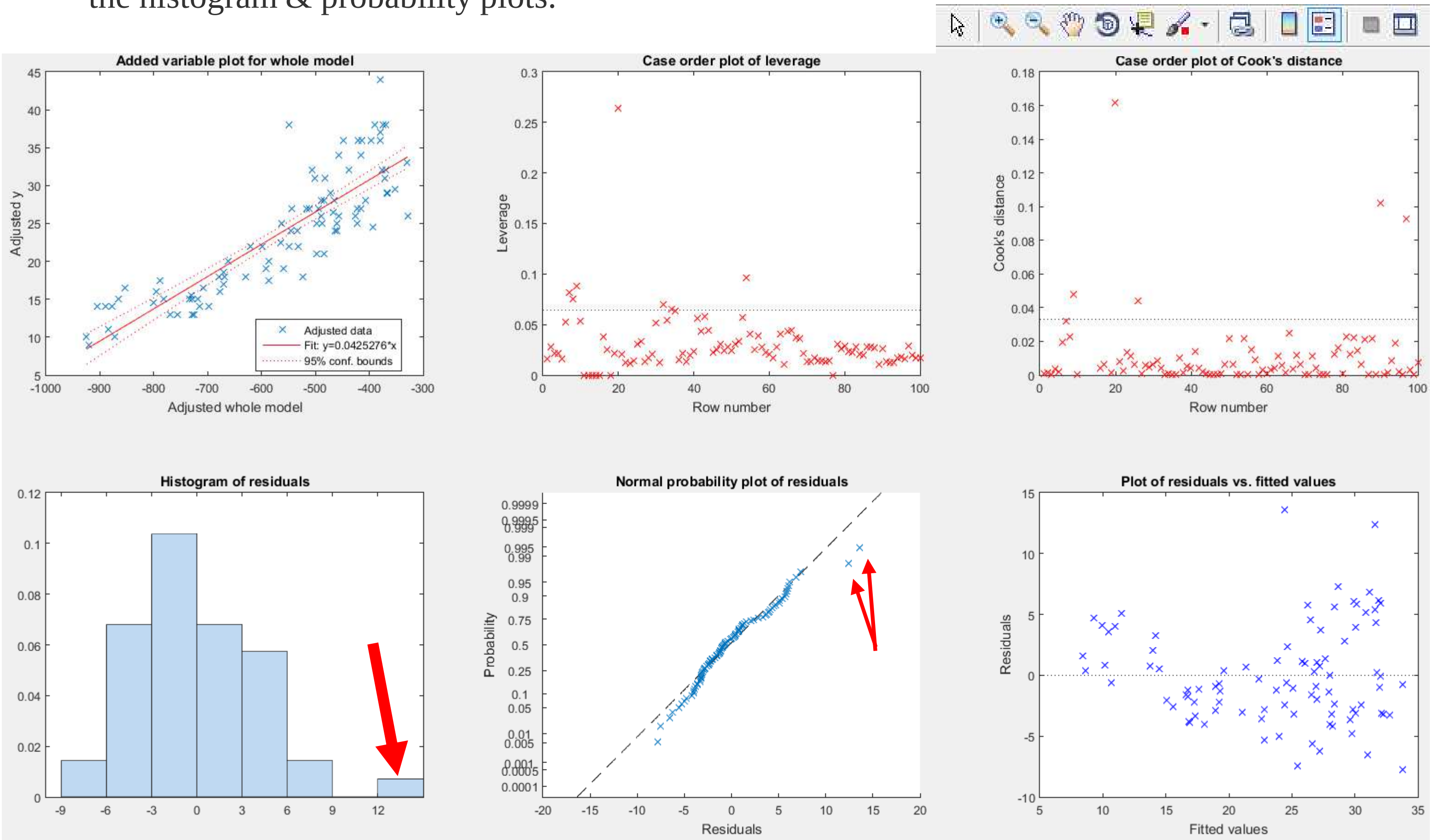
2^{nd} Try, Result 1

- Remove Displacement.
- Based on Horsepower, and Weight.
- **No significant impact on RMSE & R^2.**

- Next, we are going to remove 2 outliers.
- Two likely outliers from the histogram & probability plots.

		Estimate	SE	tStat	pValue
(Intercept)		47.769	1.7417	27.427	1.751e-45
x1	H.P.	-0.042018	0.018671	-2.2504	0.02686
x2	W.	-0.0065651	0.0010507	-6.2484	1.3519e-08

Number of observations: 93, Error degrees of freedom: 90
Root Mean Squared Error: 4.07
R-squared: 0.752, Adjusted R-Squared 0.747
F-statistic vs. constant model: 136, p-value = 5.57e-28

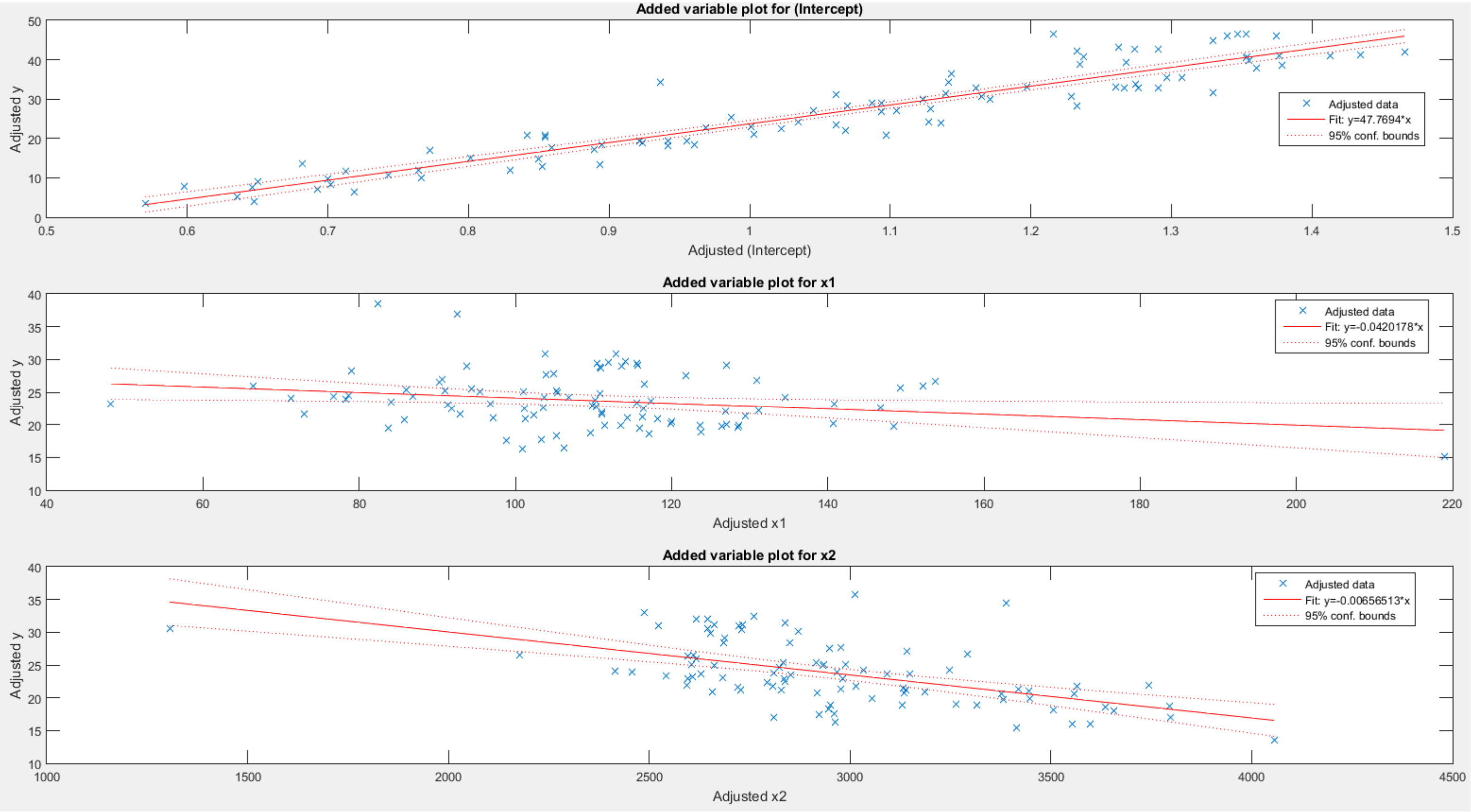


2nd Try, Result 2

- Next, we are going to remove 2 outliers.

		Estimate	SE	tStat	pValue
(Intercept)		47.769	1.7417	27.427	1.751e-45
x1	H.P.	-0.042018	0.018671	-2.2504	0.02686
x2	W.	-0.0065651	0.0010507	-6.2484	1.3519e-08

```
Number of observations: 93, Error degrees of freedom: 90
Root Mean Squared Error: 4.07
R-squared: 0.752,  Adjusted R-Squared 0.747
F-statistic vs. constant model: 136, p-value = 5.57e-28
```

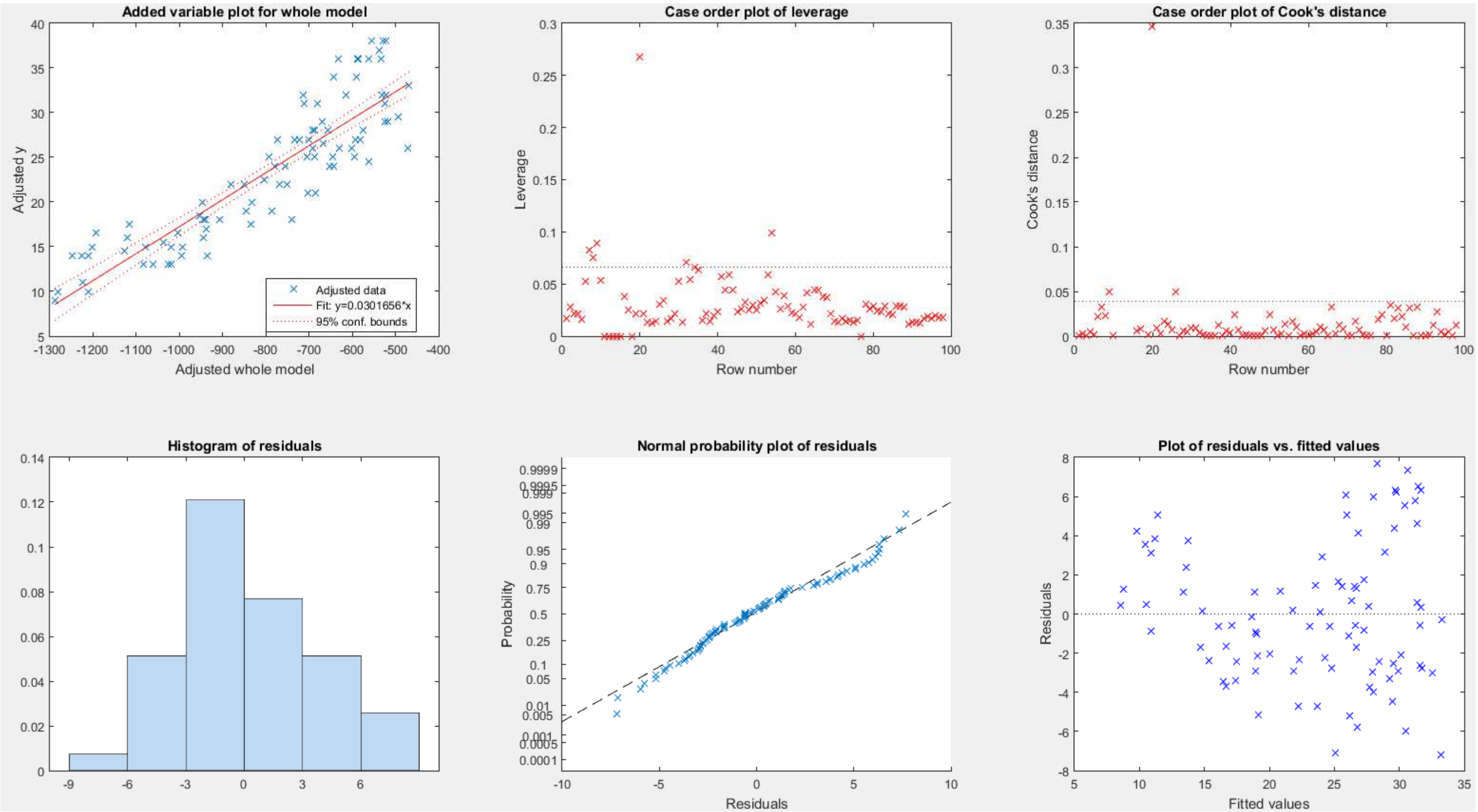


Third Try, Result 1

- **After removing outlier rec 90 & 97**.
- Reduce RMSE & increase R^2.

		Estimate	SE	tStat	pValue
(Intercept)		47.418	1.5443	30.706	8.3991e-49
x1	H.P.	-0.029339	0.016659	-1.7612	0.081686
x2	W.	-0.0070135	0.00093311	-7.5162	4.4572e-11

```
Number of observations: 91, Error degrees of freedom: 88
Root Mean Squared Error: 3.59
R-squared: 0.789,  Adjusted R-Squared 0.784
F-statistic vs. constant model: 165, p-value = 1.8e-30
```

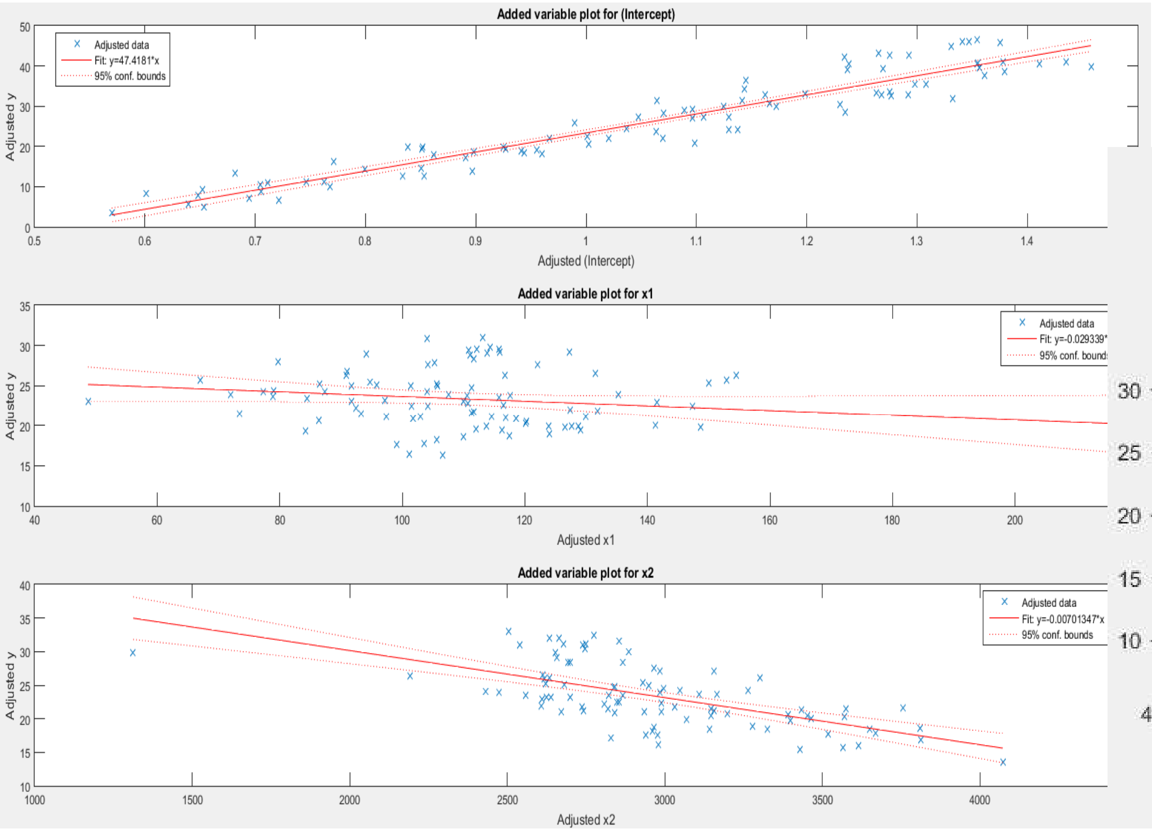


Third Try, Result 2

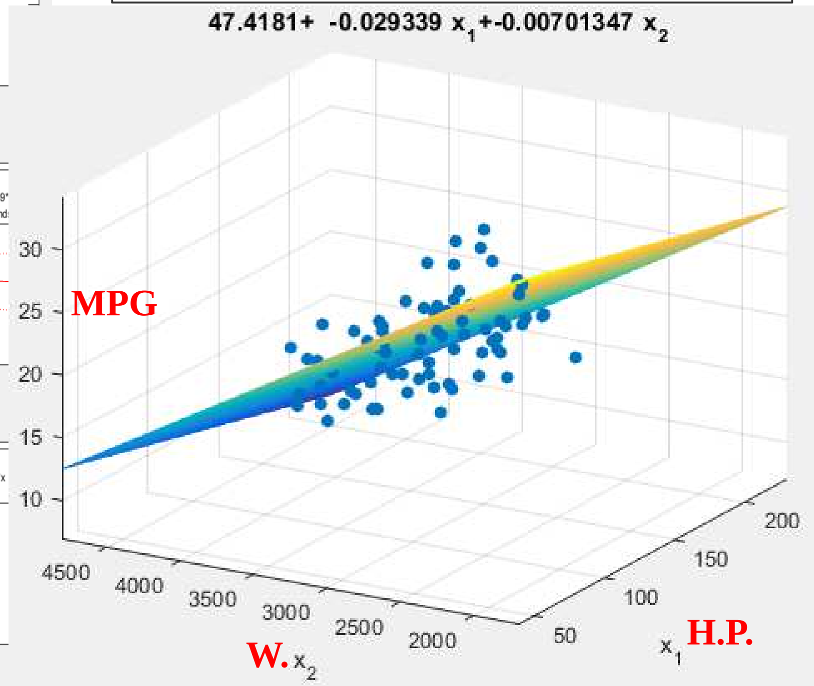
- **After removing outlier rec 90 & 97**.
- Reduce RMSE & increase R^2.

		Estimate	SE	tStat	pValue
(Intercept)		47.418	1.5443	30.706	8.3991e-49
x1	H.P.	-0.029339	0.016659	-1.7612	0.081686
x2	W.	-0.0070135	0.00093311	-7.5162	4.4572e-11

```
Number of observations: 91, Error degrees of freedom: 88
Root Mean Squared Error: 3.59
R-squared: 0.789,  Adjusted R-Squared 0.784
F-statistic vs. constant model: 165, p-value = 1.8e-30
```




```
b = num2str(mdl3.Coefficients.Estimate);
syms x1 x2;
figure,  ezmesh([b(1,:) '+' b(2,:) '*x1' '+' b(3,:) '*x2'], [40 230 1700 4800])
hold on, scatter3(X3(:, 1), X3(:, 2), Y3, 'filled'), hold off
```



Predicting Car Prices

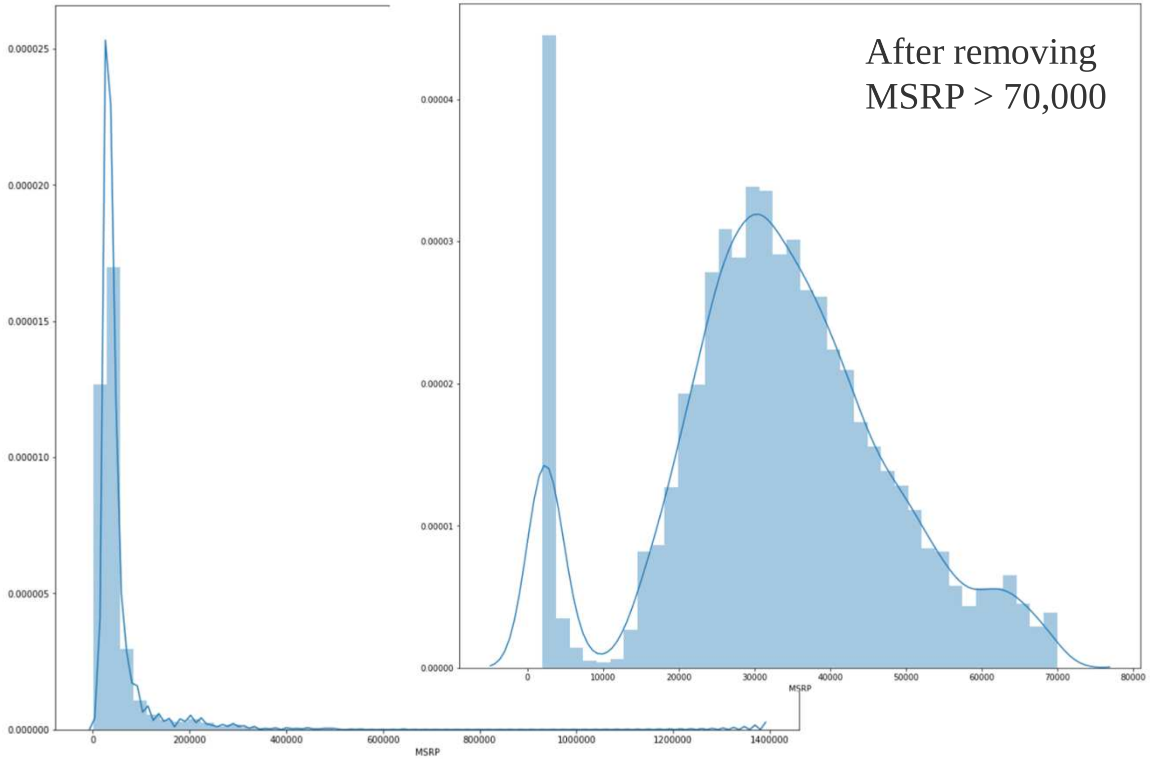
- Can we identify outliers before building any regression model?

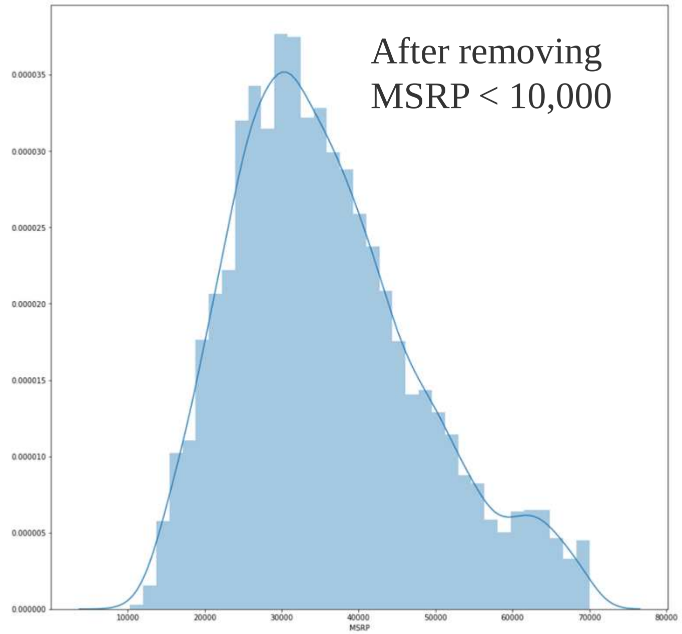
data	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Make	Model	Year	Engine Fu	Engine HP	Engine Cy	Transmiss	Driven_Wh	Number of	Market Ca	Vehicle Si	Vehicle St	highway M	city mpg	Popularity	MSRP
2	BMW	1 Series M	2011	premium u	335	6	MANUAL	rear wheel	2	Factory Tu	Compact	Coupe	26	19	3916	46135
3	BMW	1 Series	2011	premium u	300	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Convertible	28	19	3916	40650
4	BMW	1 Series	2011	premium u	300	6	MANUAL	rear wheel	2	Luxury,Hig	Compact	Coupe	28	20	3916	36350
5	BMW	1 Series	2011	premium u	230	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Coupe	28	18	3916	29450
6	BMW	1 Series	2011	premium u	230	6	MANUAL	rear wheel	2	Luxury	Compact	Convertible	28	18	3916	34500
7	BMW	1 Series	2012	premium u	230	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Coupe	28	18	3916	31200
8	BMW	1 Series	2012	premium u	300	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Convertible	26	17	3916	44100
9	BMW	1 Series	2012	premium u	300	6	MANUAL	rear wheel	2	Luxury,Hig	Compact	Coupe	28	20	3916	39300
10	BMW	1 Series	2012	premium u	230	6	MANUAL	rear wheel	2	Luxury	Compact	Convertible	28	18	3916	36900
11	BMW	1 Series	2013	premium u	230	6	MANUAL	rear wheel	2	Luxury	Compact	Convertible	27	18	3916	37200
12	BMW	1 Series	2013	premium u	300	6	MANUAL	rear wheel	2	Luxury,Hig	Compact	Coupe	28	20	3916	39600
13	BMW	1 Series	2013	premium u	230	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Coupe	28	19	3916	31500
14	BMW	1 Series	2013	premium u	300	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Convertible	28	19	3916	44400
15	BMW	1 Series	2013	premium u	230	6	MANUAL	rear wheel	2	Luxury	Compact	Convertible	28	19	3916	37200
16	BMW	1 Series	2013	premium u	230	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Coupe	28	19	3916	31500
17	BMW	1 Series	2013	premium u	320	6	MANUAL	rear wheel	2	Luxury,Hig	Compact	Convertible	25	18	3916	48250
18	BMW	1 Series	2013	premium u	320	6	MANUAL	rear wheel	2	Luxury,Hig	Compact	Coupe	28	20	3916	43550
19	Audi	100	1992	regular unl	172	6	MANUAL	front wheel	4	Luxury	Midsize	Sedan	24	17	3105	2000
20	Audi	100	1992	regular unl	172	6	MANUAL	front wheel	4	Luxury	Midsize	Sedan	24	17	3105	2000

Predicting Car Prices
MDL-01 2018 spring
Saleh Alkadayar
Rathana Sorn
Jose Rodriguez
Julie Flater
Gassan Zaid

Remove Outliers before Regression?

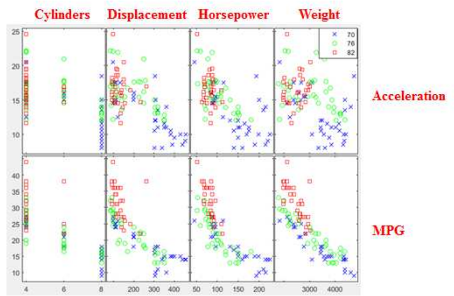
- So far we remove outliers based on regression results.
- Can we remove outliers before regression?
- Build different models for cars in different MSRP ranges…

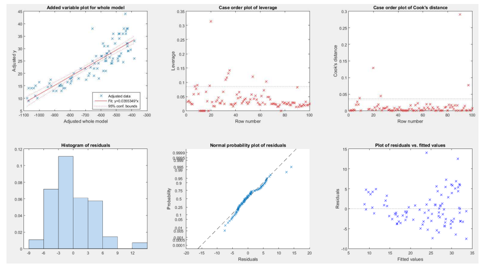


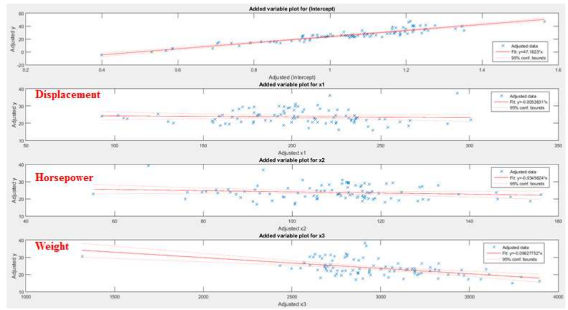


Predicting Car Prices
MDL-01 2018 spring
Saleh Alkadayar
Rathana Sorn
Jose Rodriguez
Julie Flater
Gassan Zaid

Summary of Steps







- Linear Regression Workflow
  - Step 1. Understand data.
  - Step 2. Create a fitted model & evaluate prediction quality.
  - Step 3. Simplify the model (by removing some predictors).
  - Step 3. Locate and remove outliers.
  - Step 5. Predict responses to new data.

- Linear Regression Summary
  - Measuring overall accuracy: RMSE↓, R^2↑.
  - Identifying outliers: leverage, Cook's distance, probability plot, residual histogram.
  - Identifying useful variables: p-value, Added-Variable plot, plotSlice. θ???

Hospital Example– **Which Predictor Has Greater Impact???**

- "hospital.xls" has patient names, gender, age, weight, blood pressure, & treatments.

	name	sex	age	wgt	smoke	sys	dia	trial1	trial2	trial3	trial4
YPL-320	'SMITH'	'm'	38	176	1	124	93	18	-99	-99	-99

- Predict systolic pressure as a function of sex, age, wgt, smoke.
- sex, age, & weight have high ***p***-values, indicating some of them unnecessary.

- plotResiduals(mdl) and improve RMSE & R^2 after removing outliers.

(Intercept)		118.28	9.1557e-28
x1	**gender**	0.88162	0.76549
x2	**age**	0.08602	0.20438
x3	**wgt**	-0.016685	0.76524
x4	**smoke**	9.884	1.9546e-15

RMSE = 4.81, R^2 = 0.508

(Intercept)		115	2.3258e-27
x1	**gender**	0.22181	0.93846
x2	**age**	0.10678	0.10721
x3	**wgt**	0.00036854	0.9946
x4	**smoke**	10.002	2.8087e-16

RMSE = 4.66, R^2 = 0.536

Output Interpretation

- Risk of high systolic pressure appears to be…   (**w/o considering p-values**)
  - **MUCH** higher among smokers (or non-smokers??).
  - Higher among males or females??
  - Increasing with age. But, how much???
  - Compare to weight?
  - “Gender” and “Smoke” are **categorical** var.

		θ	
(Intercept)		115	2.3258e-27
x1	**gender**	0.22181	0.93846
x2	**age**	0.10678	0.10721
x3	**wgt**	0.00036854	0.9946
x4	**smoke**	10.002	2.8087e-16

RMSE = 4.66, R^2 = 0.536

$$z_i = \frac{x_i - \mu}{SD}$$

		θ
(Intercept)		119.15
x1	**gender**	0.22181
x2	**age**	0.77047
x3	**wgt**	0.0097927
x4	**smoke**	10.002

RMSE = 4.66, R^2 = 0.536

SD
7.2154
26.5714

Trained θ May **Flip** **After** Standardization

load patients

patients = table(Systolic, Age, Gender, Height, Weight, Smoker, Location, SelfAssessedHealthStatus, 'RowNames',LastName);

Y = Systolic;

X = [**Age Weight**];

mdl = fitlm(X, Y)

		pValue
(Intercept)	112.84	2.3349e-125
x1	0.11248	0.22943
x2	0.036567	0.15086

mdl2 = fitlm(zscore(X) , Y)

% zscore(Y) ??? 🡸 🡸 🡸

		pValue
(Intercept)	122.78	2.3349e-125
x1	0.81161	0.22943
x2	0.97164	0.15086

mean(X), **% 38.28, 154.00**

std(X) **% 7.2154, 26.5714**

$$z_i = \frac{x_i - \mu}{SD}$$

Standardize New Data **After Building A Model, Before Prediction**

- If we build model with standardized dataset, do we need to standardize the **new data** before prediction?
- **Yes, but how???**

```
%% *** Assume you have zscore your training data:
[Z, mu, sigma] = zscore( X )
```

```
%% *** Before predicting ŷ for new data, do followings:
tmp = newX - mu;
newZ = tmp ./ sigma;                    % sigma = STD
```

$$z_i = \frac{x_i - \mu}{SD}$$

```
%% if you have multiple records of new data, do followings:
tmp = newX - (ones(length(newX), 1) * mu);
newZ = tmp * diag(1 ./ sigma);          % sigma = STD
```

After Removing Outliers, Standardization Again!!

$$z_i = \frac{x_i - \mu}{SD}$$

- After removing outliers, standardization **again** before building a new model!!!

Comparison of Standardization after Removing an Outlier

Thanks to Mr. Jerry High's test example
ML 2017 spring

Statistics for Original Data	Age	Height	Weight
Std_Dev	7.2154	2.8365	26.5714
Mean	38.2800	67.0700	154.0000

Statistics for Data with Outlier Removed	Age	Height	Weight
Std_Dev	7.2510	2.8431	26.4826
Mean	38.2929	67.0909	154.3434

	Original Data (100 rows)			Original Data (standardized)		
row	Age	Height	Weight	Age	Height	Weight
1	38	71	176	-0.0388	1.3855	0.8280
2	43	69	163	0.6542	0.6804	0.3387
3	38	64	131	-0.0388	-1.0823	-0.8656
4	40	67	133	0.2384	-0.0247	-0.7903
5	49	64	119	1.4857	-1.0823	-1.3172
6	46	68	142	1.0699	0.3279	-0.4516
7	33	64	142	-0.7318	-1.0823	-0.4516
8	40	68	180	0.2384	0.3279	0.9785
9	28	68	183	-1.4247	0.3279	1.0914
10	31	66	132	-1.0090	-0.3772	-0.8280
11	45	68	128	0.9313	0.3279	-0.9785
12	42	66	137	0.5156	-0.3772	-0.6398
13	25	71	174	-1.8405	1.3855	0.7527
14	39	72	202	0.0998	1.7381	1.8065
15	36	65	129	-0.3160	-0.7298	-0.9409
16	48	71	181	1.3471	1.3855	1.0161

	Data with Outlier Removed (99 rows - #93 removed)			Data with Outlier Removed (standardized)		
row	Age	Height	Weight	Age	Height	Weight
1	38	71	176	-0.0404	1.3749	0.8178
2	43	69	163	0.6492	0.6715	0.3269
3	38	64	131	-0.0404	-1.0871	-0.8815
4	40	67	133	0.2354	-0.0320	-0.8059
5	49	64	119	1.4766	-1.0871	-1.3346
6	46	68	142	1.0629	0.3197	-0.4661
7	33	64	142	-0.7300	-1.0871	-0.4661
8	40	68	180	0.2354	0.3197	0.9688
9	28	68	183	-1.4195	0.3197	1.0821
10	31	66	132	-1.0058	-0.3837	-0.8437
11	45	68	128	0.9250	0.3197	-0.9947
12	42	66	137	0.5113	-0.3837	-0.6549
13	25	71	174	-1.8333	1.3749	0.7422
14	39	72	202	0.0975	1.7266	1.7995
15	36	65	129	-0.3162	-0.7354	-0.9570
16	48	71	181	1.3387	1.3749	1.0066

Python Z-Score

```
# http://scikit-learn.org/stable/modules/preprocessing.html
from sklearn import preprocessing
import numpy as np
X_train = np.array([[ 1., -1.,  2.],
                    [ 2.,  0.,  0.],
                    [ 0.,  1., -1.]])
X_scaled = preprocessing.scale(X_train)
print(X_scaled, '\n')    # zscore of X
print(X_scaled.mean(axis=0), X_scaled.std(axis=0))
```

Categorical Variables

- Major variable types:
  - Continuous variables: age, income, salary.
  - Categorical variables: gender, rank, department, city, etc.

- How to represent categorical variables in LR?
  - Use 100, 200, 300 , etc. for cities? How about use Zip Code?
    - But, this assumes “order” and “magnitude”. city 300 > city 200 > city 100.
    - Zip code 90210 > 55105? 90210 more important than 55105?

- Use binary variables (or dummies).
  - Dummies?? ➔ not real variables, they are just to help representing something else.

Dummy Variables

- Use binary variables (or dummies).
  - Refer to as indicators (to indicate if a record **has** a particular value **or not**).
  - **0** represents the ***reference group***.
  - One categorical variable with ***c*** categories **usually** represented by ***c* – 1** indicators.
  - 3 categories??
    - Let a categorical variable with levels {Small, Medium, Large}.
    - Represent it using **two** dummy variables D_1 and D_2.
    - A record with Medium [D_1, D_2] = [1, 0]. Large [0, 1]. Small [0, 0].
    - The category represented by all **0**s is the ***reference group***.

	F	M	θ M
Male	0	1	1
Female	1	0	0
Female	1	0	0
Male	0	1	1
Female	1	0	0

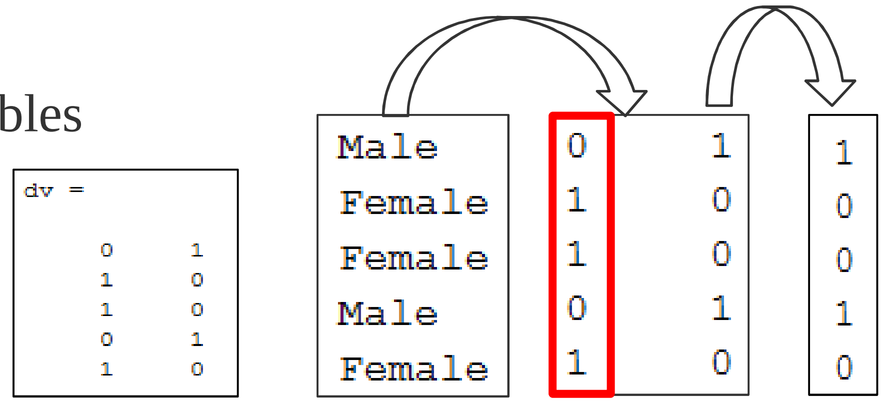
	S	M	L	M	L
Small	1	0	0	**0**	**0**
Medium	0	1	0	1	0
Large	0	0	1	0	1

			M	L
			D_1	D_2
M	**010**		**1**	**0**
M	**010**		**1**	**0**
S	**100**		**0**	**0**
L	**001**		**0**	**1**
S	**100**		**0**	**0**

Creating Dummy Variables in Matlab

- Data
  - gender = nominal({'Male'; 'Female'; 'Female'; 'Male'; 'Female'});

- Covert categorical data to dummy variables
  - dv = **dummyvar**(gender)



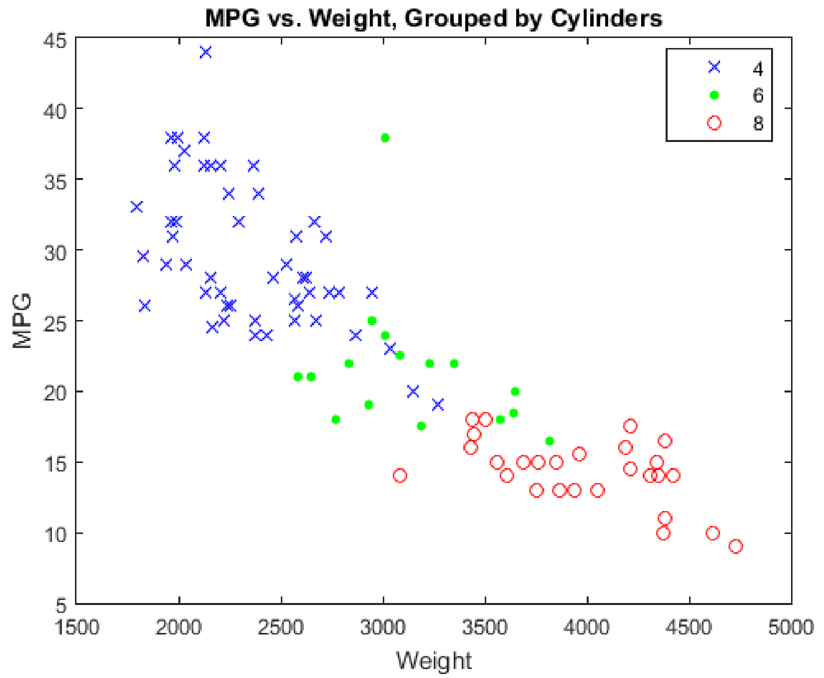
- How to use dummy variables in a regression model?
  - **Must** delete a column (to create a reference group), (**or** do not use θ_0 in an LR model).
  - Gender example→ use only one column of the dummy variable.
  - LR coefficients remain the same but **opposite signs**, if **complement** the dummy.

- References to Matlab Dummy variables.
  - http://www.mathworks.com/help/stats/dummyvar.html
  - http://www.mathworks.com/help/stats/dummy-indicator-variables.html

Car Example with Dummy Variables (Cyclinders)

- Similar way / example.
  - http://www.mathworks.com/help/stats/group-comparisons-using-categorical-arrays.html

```
clear all, close all hidden
load carsmall;
figure,   gscatter(Weight, MPG, Cylinders, 'bgr', 'x.o')
title('MPG vs. Weight, Grouped by Cylinders')

X = [Weight, Cylinders];
Y = MPG;
lm = fitlm(X, Y)

dv_CL = dummyvar(Cylinders);
DX_CL = [Weight, dv_CL(:, [6 8])];   % WHY 6, 8?
D_lm = fitlm(DX_CL, Y)
```




```
(Intercept)MPG     48.908
x1         W    -0.0056549
x2         CYL     -1.5302
```

can tell it was degrading.

```
Root Mean Squared Error: 3.96
R-squared: 0.762,  Adjusted R-Squared 0.757
```

```
(Intercept)MPG     42.929
x1         W    -0.0056819
x2         C6      -3.5825
x3         C8        -6.01
```

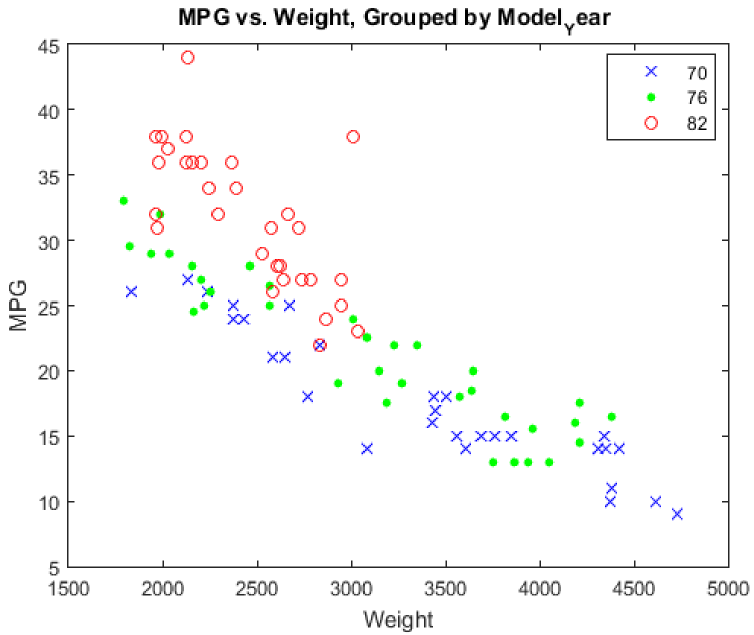
able to tell MPG worsening w/ more Cylinders vs 4.

```
Root Mean Squared Error: 3.98
R-squared: 0.763,  Adjusted R-Squared 0.755
```

Car Example with Dummy Variables (Model Years)

- Similar way / example.
  - http://www.mathworks.com/help/stats/group-comparisons-using-categorical-arrays.html

```
clear all, close all hidden
load carsmall;
figure,   gscatter(Weight, MPG, Model_Year, 'bgr', 'x.o')
title('MPG vs. Weight, Grouped by Model_Year ')

X = [Weight, Model_Year];
Y = MPG;
lm = fitlm(X, Y)

dv_MY = dummyvar(Model_Year);
DX_MY = [Weight, dv_MY(:, [76 82])];    % WHY 76, 82
D_lm = fitlm(DX_MY, Y)
```



E(MPG) $= \theta_0 + \theta_1$***Weight*** $+ \theta_2$***D*****[1976]** $+ \theta_3$***D*****[1982].**

```
(Intercept) MPG   -5.7045
x1          W   -0.0068023
x2          YR     0.65127
Root Mean Squared Error: 3.06
R-squared: 0.858,  Adjusted R-Squared 0.855
```

can tell general improvement.

```
(Intercept) MPG      40.11
x1          W   -0.0066475
x2          76      1.9291
x3          82      7.9093
Root Mean Squared Error: 2.92
R-squared: 0.873,  Adjusted R-Squared 0.868
```

able to tell how much MPG improves from '70s.

Dummy Variables using **pandas** or **sklearn**

```
import pandas as pd
import numpy as np
s = ['a', 'b', 'c', 'a']
dmy = pd.get_dummies(s)
print(dmy, '\n')
#x = np.zeros(4, 3)
x = np.array(dmy)
print(x, '\n')
print(x[:, 1:])  # get c – 1 columns
```

```
   a  b  c
0  1  0  0
1  0  1  0
2  0  0  1
3  1  0  0

[[1 0 0]
 [0 1 0]
 [0 0 1]
 [1 0 0]]

[[0 0]
 [1 0]
 [0 1]
 [0 0]]
```

```
from sklearn.preprocessing import OneHotEncoder
s = [[1], [2], [3], [1]]
enc = OneHotEncoder()
enc.fit(s)
x = enc.transform(s).toarray()
print(x)
```

```
[[ 1.  0.  0.]
 [ 0.  1.  0.]
 [ 0.  0.  1.]
 [ 1.  0.  0.]]
```

```
from sklearn.preprocessing import OneHotEncoder
s = [[0, 0, 3], [1, 1, 0], [0, 2, 1], [1, 0, 2]]
print(s, '\n')
enc = OneHotEncoder()
enc.fit(s)
print(enc.transform(s).toarray())
```

```
[[0, 0, 3], [1, 1, 0], [0, 2, 1], [1, 0, 2]]

[[ 1.  0.  1.  0.  0.  0.  0.  0.  1.]
 [ 0.  1.  0.  1.  0.  1.  0.  0.  0.]
 [ 1.  0.  0.  0.  1.  0.  1.  0.  0.]
 [ 0.  1.  1.  0.  0.  0.  0.  1.  0.]]
```

Predicting Car Prices

- Any preprocessing?

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Make	Model	Year	Engine Fu	Engine HP	Engine Cy	Transmiss	Driven_Wh	Number of	Market Ca	Vehicle Siz	Vehicle Sty	highway M	city mpg	Popularity	MSRP
2	BMW	1 Series M	2011	premium u	335	6	MANUAL	rear wheel	2	Factory Tu	Compact	Coupe	26	19	3916	46135
3	BMW	1 Series	2011	premium u	300	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Convertible	28	19	3916	40650
4	BMW	1 Series	2011	premium u	300	6	MANUAL	rear wheel	2	Luxury,Hig	Compact	Coupe	28	20	3916	36350
5	BMW	1 Series	2011	premium u	230	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Coupe	28	18	3916	29450
6	BMW	1 Series	2011	premium u	230	6	MANUAL	rear wheel	2	Luxury	Compact	Convertible	28	18	3916	34500
7	BMW	1 Series	2012	premium u	230	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Coupe	28	18	3916	31200
8	BMW	1 Series	2012	premium u	300	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Convertible	26	17	3916	44100
9	BMW	1 Series	2012	premium u	300	6	MANUAL	rear wheel	2	Luxury,Hig	Compact	Coupe	28	20	3916	39300
10	BMW	1 Series	2012	premium u	230	6	MANUAL	rear wheel	2	Luxury	Compact	Convertible	28	18	3916	36900
11	BMW	1 Series	2013	premium u	230	6	MANUAL	rear wheel	2	Luxury	Compact	Convertible	27	18	3916	37200
12	BMW	1 Series	2013	premium u	300	6	MANUAL	rear wheel	2	Luxury,Hig	Compact	Coupe	28	20	3916	39600
13	BMW	1 Series	2013	premium u	230	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Coupe	28	19	3916	31500
14	BMW	1 Series	2013	premium u	300	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Convertible	28	19	3916	44400
15	BMW	1 Series	2013	premium u	230	6	MANUAL	rear wheel	2	Luxury	Compact	Convertible	28	19	3916	37200
16	BMW	1 Series	2013	premium u	230	6	MANUAL	rear wheel	2	Luxury,Per	Compact	Coupe	28	19	3916	31500
17	BMW	1 Series	2013	premium u	320	6	MANUAL	rear wheel	2	Luxury,Hig	Compact	Convertible	25	18	3916	48250
18	BMW	1 Series	2013	premium u	320	6	MANUAL	rear wheel	2	Luxury,Hig	Compact	Coupe	28	20	3916	43550
19	Audi	100	1992	regular unl	172	6	MANUAL	front wheel	4	Luxury	Midsize	Sedan	24	17	3105	2000
20	Audi	100	1992	regular unl	172	6	MANUAL	front wheel	4	Luxury	Midsize	Sedan	24	17	3105	2000

Predicting Car Prices
MDL-01 2018 spring
Saleh Alkadayar
Rathana Sorn
Jose Rodriguez
Julie Flater
Gassan Zaid

Removing Less Useful Predictors? Which One(s)?

- "hospital.xls" has patient names, sex, age, weight, blood pressure, & treatments.

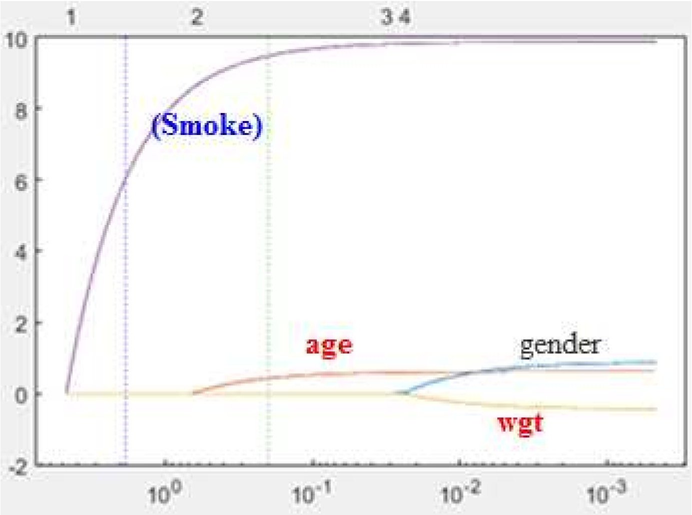
	name	sex	age	wgt	smoke	sys	dia	trial1	trial2	trial3	trial4
	______	___	___	___	_____	___	___	______	______	______	______
YPL-320	'SMITH'	'm'	38	176	1	124	93	18	-99	-99	-99

  - sex, age, & weight have high p-values, indicating some of them unnecessary.

- Later, we will use "**regularization**" to…
  - Identify useful predictors to simplify model, & maintain ***similar*** prediction quality.

(Intercept)		115	2.3258e-27
x1	**gender**	0.22181	0.93846
x2	**age**	0.10678	0.10721
x3	**wgt**	0.00036854	0.9946
x4	**smoke**	10.002	2.8087e-16

RMSE = 4.66, R^2 = 0.536

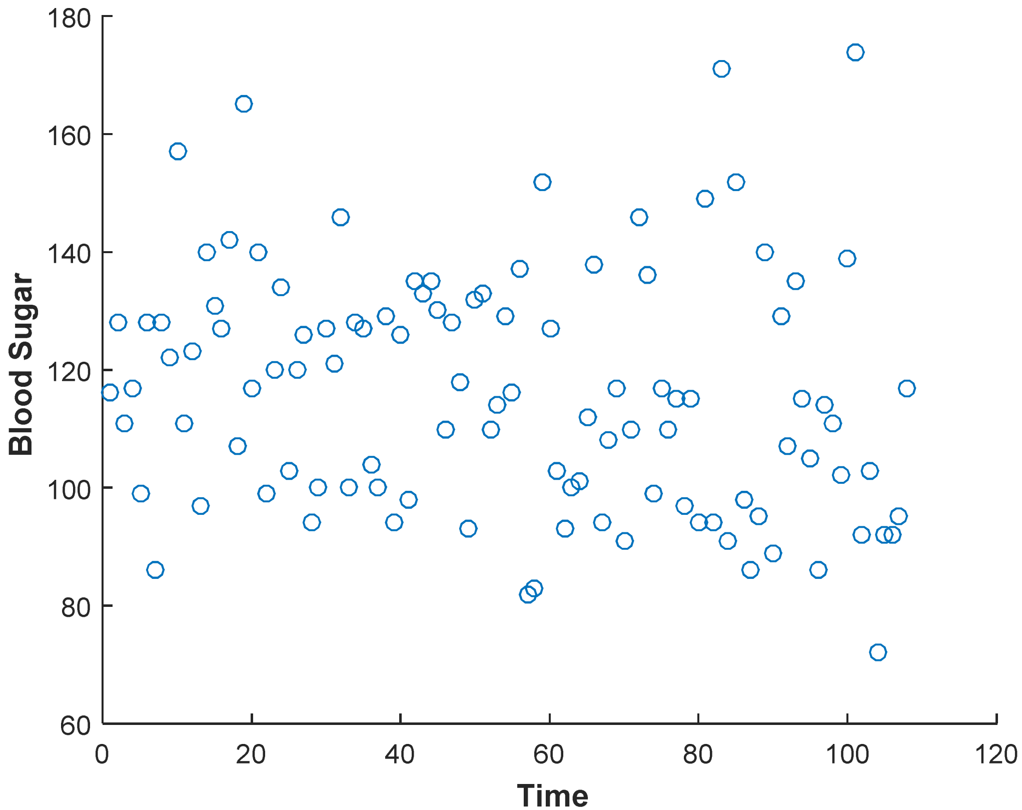


More Irregular Data– Blood Sugar Over Time

- Possible to fit this data into a **linear** model?

```
Y = [ 116   128   111   117    99   128    86   128   122   157   111   123    97   140   131   127   142   107   165   117   140    99   120 ...
134   103   120   126  94   100   127   121   146   100   128   127   104   100   129    94   126    98   135   133   135   130   110   128  ...
118    93   132   133   110   114   129 116   137    82    83   152   127   103    93   100   101   112   138    94   108   117    91   110  ...
146   136    99   117   110   115    97   115    94   149  94   171    91   152    98    86    95   140    89   129   107   135   115   105 ...
86   114   111   102   139   174    92   103    72    92    92    95   117];
```

X =[1 : length(Y)];



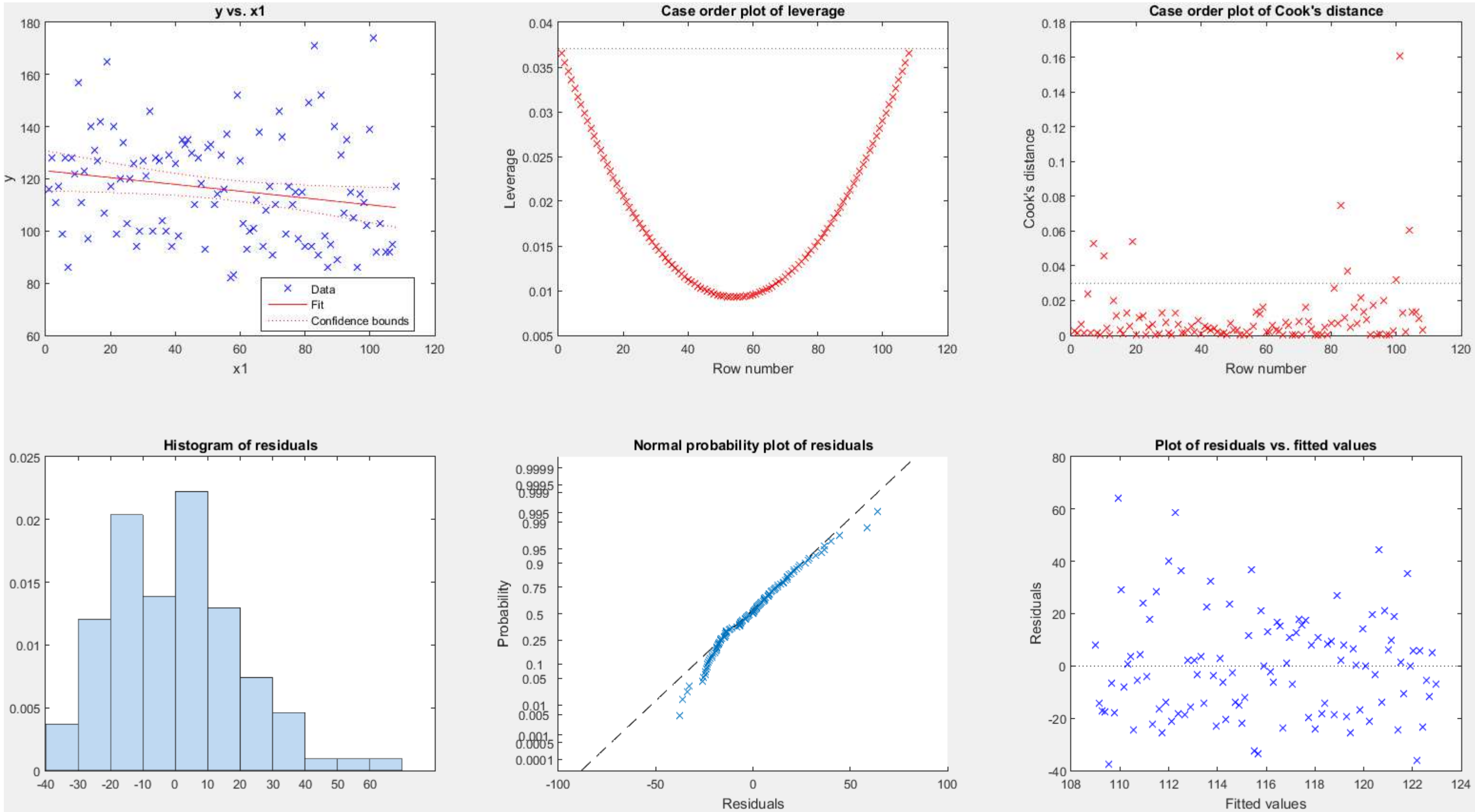
Fit A Linear Model

- Overall, not a good fit.
- But, easy to interpret.

mdl = fitlm(X, Y)

PlotFigures(mdl)

```
                  Estimate      SE        tStat       pValue
                  ________   ________    _______    __________

    (Intercept)     123.09     3.9042     31.527    1.1421e-55
    x1            -0.13043   0.062182    -2.0976      0.038321


Number of observations: 108, Error degrees of freedom: 106
Root Mean Squared Error: 20.1
R-squared: 0.0399,  Adjusted R-Squared 0.0308
F-statistic vs. constant model: 4.4, p-value = 0.0383
```



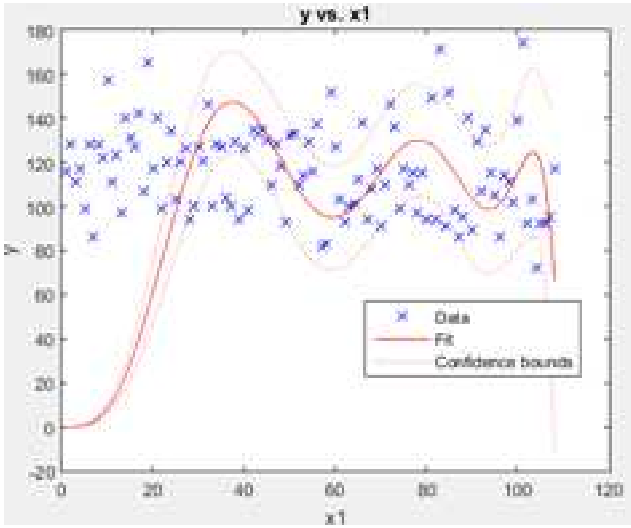
Fit Polynomial Degree 9

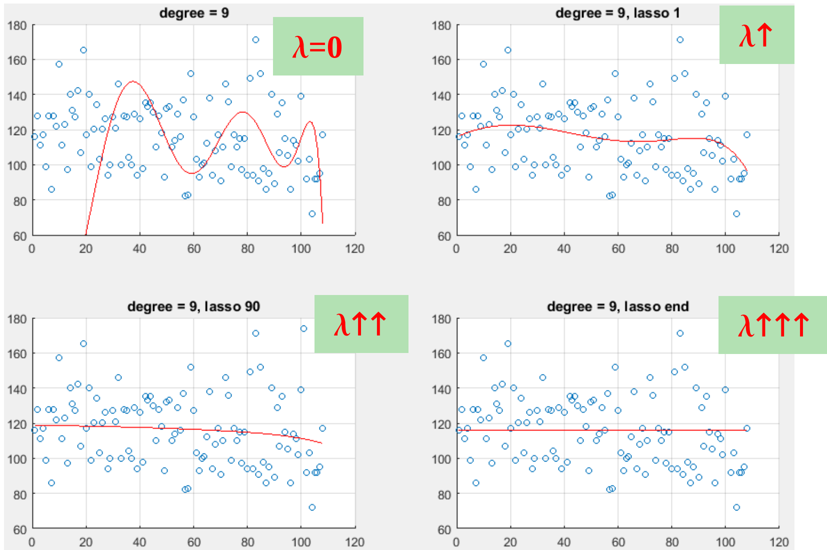
- Much better R^2 and Adjusted R^2.
  - **Interpretation???**
  - Should we use higher degree?!

mdl9 = fitlm(X, Y, **'poly9'**)

	Estimate	SE	tStat	pValue
(Intercept)	0	0	NaN	NaN
x1	0	0	NaN	NaN
x1^2	0	0	NaN	NaN
x1^3	0	0	NaN	NaN
x1^4	0.0015597	0.00022262	7.0063	3.1191e-10
x1^5	-9.5687e-05	1.5474e-05	-6.1837	1.4453e-08
x1^6	2.3463e-06	4.2002e-07	5.5863	2.0807e-07
x1^7	-2.8574e-08	5.5723e-09	-5.1278	1.4776e-06
x1^8	1.724e-10	3.6199e-11	4.7625	6.6187e-06
x1^9	-4.1193e-13	9.229e-14	-4.4634	2.1579e-05

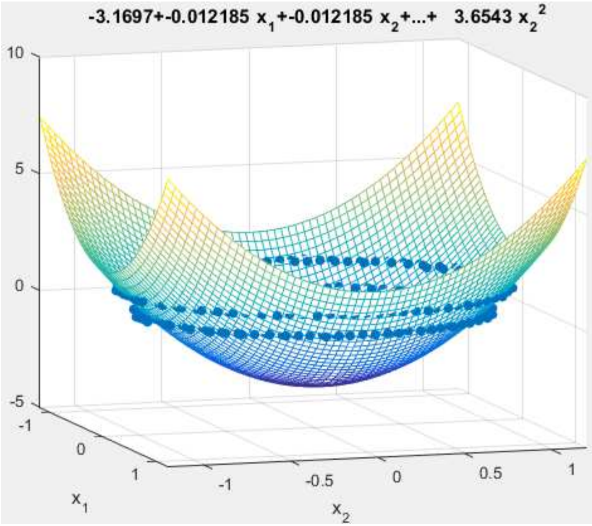
Number of observations: 108, Error degrees of freedom: 102
Root Mean Squared Error: 53.4
R-squared: 0.44, Adjusted R-Squared 0.413
F-statistic vs. constant model: 16.1, p-value = 1.18e-11

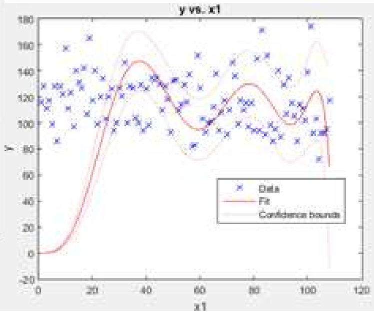




Fit Non-Linear Model in Matlab

- Specify non-linear model in the Matlab linear regression function➔
  - mdl = fitlm(tbl, **modelspec**)
    - fitlm(X, Y, **'poly9'**)
    - fitlm(X, Y, **'y ~ x1^6 + x1^2'**)
  - **Multivariant non-linear model:**



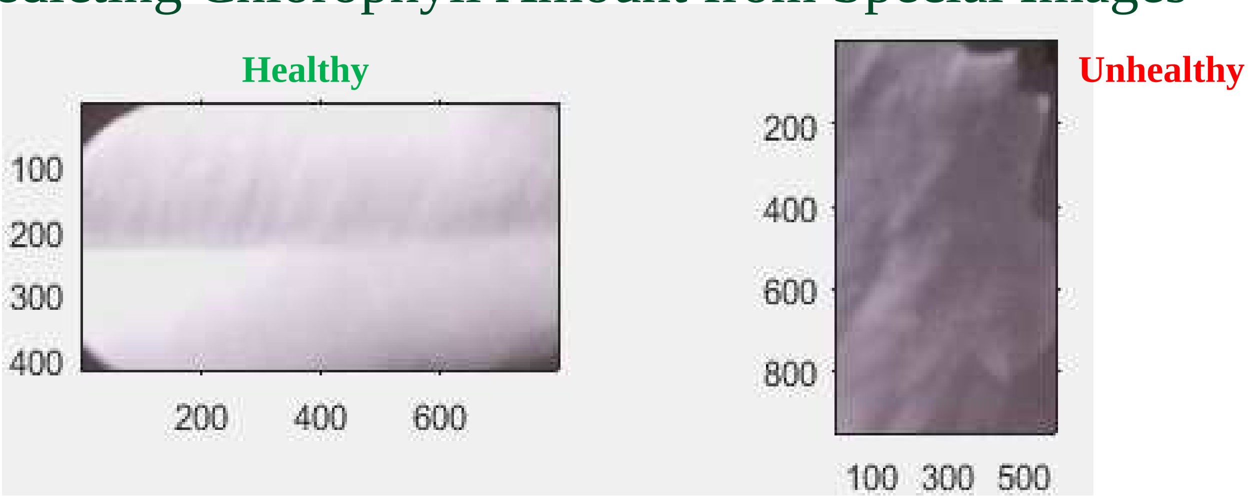


possible “modelspect”:

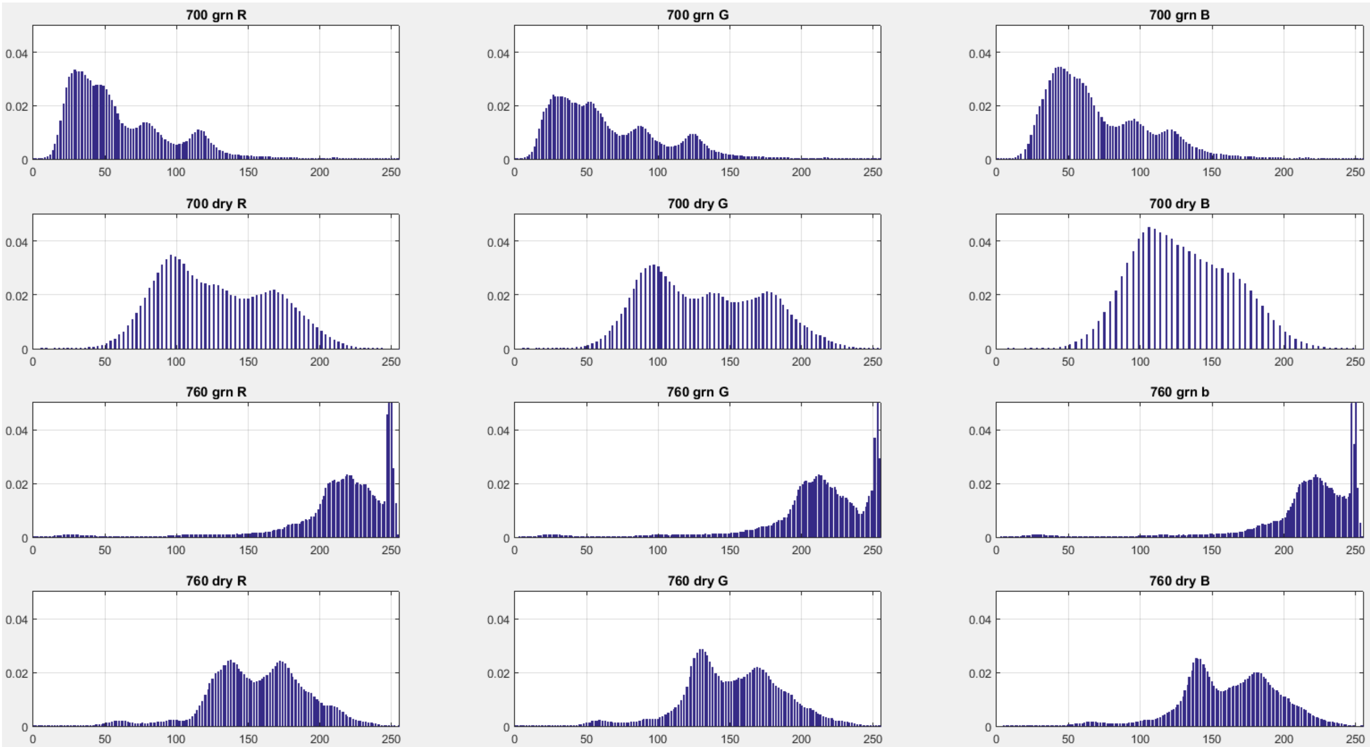
String	Model Type http://www.mathworks.com/help/stats/fitlm.html#inputarg_modelspec
'constant'	Model contains only a constant (intercept) term.
'linear'	Model contains an intercept and linear terms for each predictor.
'interactions'	Model contains an intercept, linear terms, and all products of pairs of distinct predictors (no squared terms).
'purequadratic'	Model contains an intercept, linear terms, and squared terms.
'quadratic'	Model contains an intercept, linear terms, interactions, and squared terms.
'poly*ijk*'	Model is a polynomial with all terms up to degree *i* in the first predictor, degree *j* in the second predictor, etc. Use numerals 0 through 9. For example, 'poly2111' has a constant plus all linear and product terms, and also contains terms with predictor 1 squared.

Appendix

Another LR Example
Predicting Chlorophyll Amount from Special Images



256 × 3 × 4



Dimensionality and R^2

Dimensionality = 3072 / 128 = 24

Root Mean Squared Error: 7.04

R-squared: 0.92,

Adjusted R-Squared 0.892

F-statistic vs. constant model: 32,

p-value = 7.35e-16

Dimensionality = 3072 / 64 = 48

Root Mean Squared Error: 7.2

R-squared: 0.968,

Adjusted R-Squared 0.887

F-statistic vs. constant model: 12,

p-value = 6.79e-06

Dimensionality = 3072

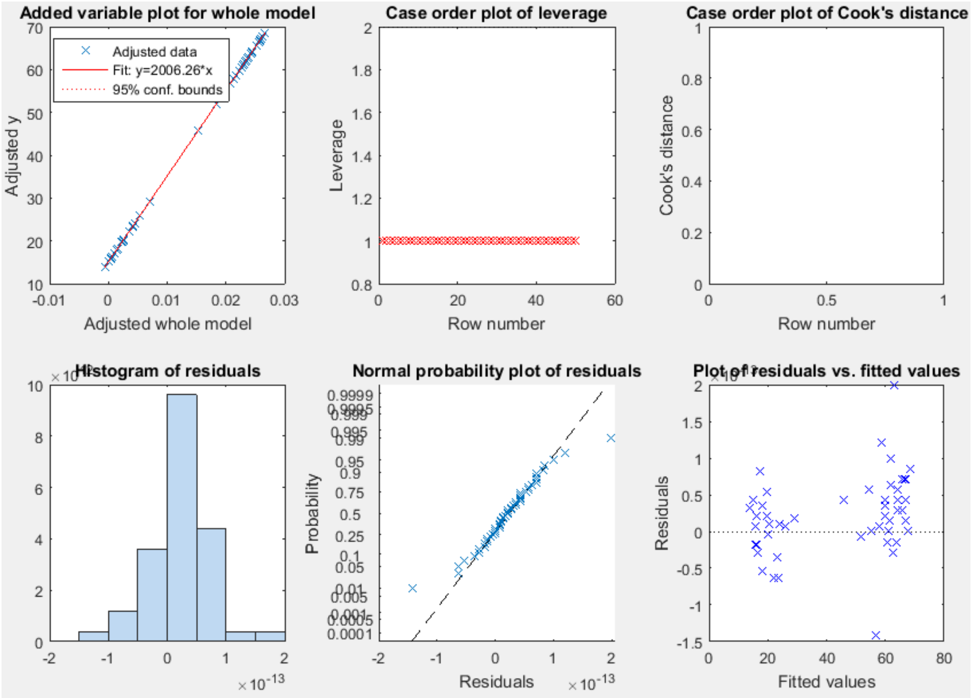
Root Mean Squared Error: 0

R-squared: 1,

Adjusted R-Squared NaN

F-statistic vs. constant model: NaN,

p-value = NaN



***F*-Statistics**

$$F = \frac{\text{explained variance}}{\text{unexplained variance}}$$

- ***F*-Statistics.**
  - ***F*** statistic is the distance from black dashed line to the ***y***-axis.
  - The ***p*** value is the dark blue area under the curve from ***F*** to infinity.
  - Higher ***F*** values correspond to lower ***p*** values (better).
  - http://stats.stackexchange.com/questions/12398/how-to-interpret-f-and-p-value-in-anova
  - In linear regression, the ***F***-statistic is the test statistic for the analysis of variance (ANOVA) approach to test the significance of the terms or components in the model.

Estimated Coefficients:

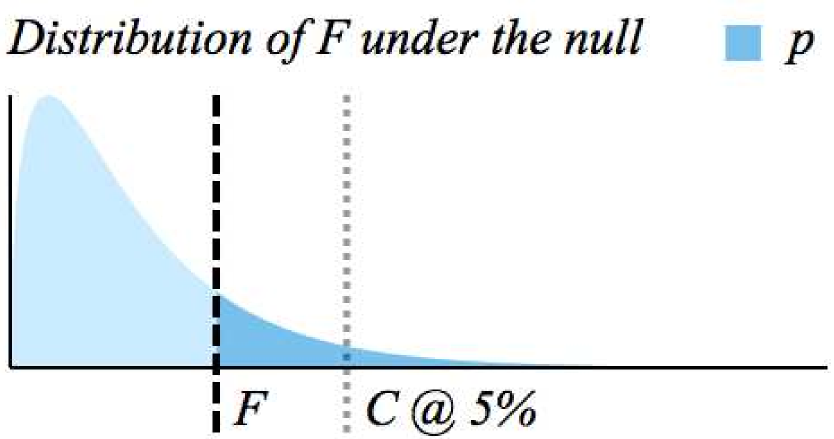
	Estimate	SE	tStat	pValue
(Intercept)	-1.3567e-15	3.6419e-07	-3.7253e-09	1
x1	4	1.6859e-07	2.3727e+07	2.6832e-08

Number of observations: 3, Error degrees of freedom: 1

Root Mean Squared Error: 2.38e-07

R-squared: 1, Adjusted R-Squared 1

F-statistic vs. constant model: 5.63e+14, p-value = 2.68e-08

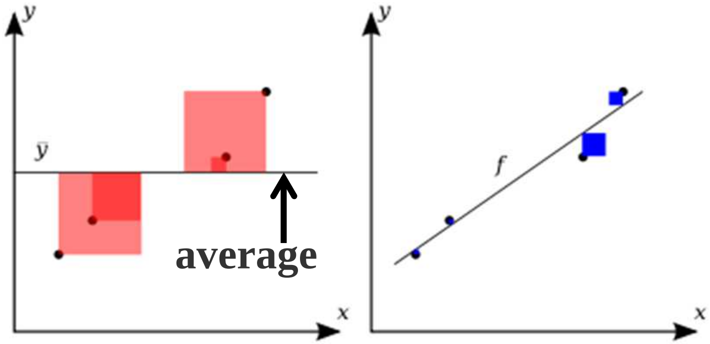


http://stats.stackexchange.com/questions/12398/how-to-interpret-f-and-p-value-in-anova

Explained / Unexplained Variance

https://en.wikipedia.org/wiki/Coefficient_of_determination

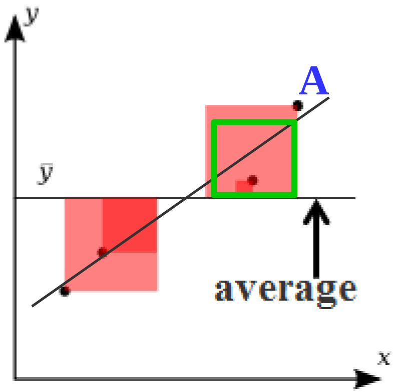
- **R^2**
  - **SST** (total) = $\sum_i (y_i - \bar{y})^2$ **SSE** = $\sum_i (y_i - \hat{y}_i)^2$ $R^2 = 1 - \frac{SSE}{SST}$
  - **Explained sum of square = $\sum_i (\hat{y}_i - \bar{y})^2$**

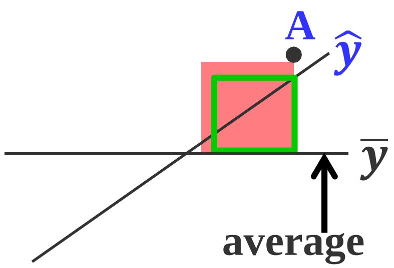


http://en.wikipedia.org/wiki/Coefficient_of_determination

Your text here

- Consider point **A**…
  - **R^2** is close to 1.
  - Explained sum of square (**green area**) is large.
  - ***F*** is VERY large.





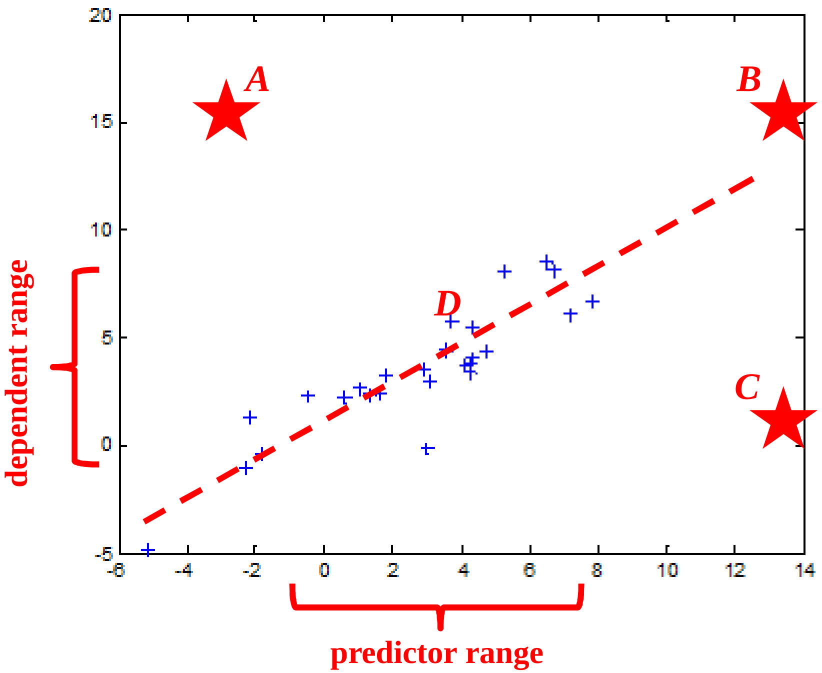
$$F = \frac{explained\ variance}{unexplained\ variance}$$

$$= \frac{green\ area}{red - green}$$

Leverage

- A measure of how a point affects the regression predictions due to its position.
- Generally, more leverage a point may have if it is farther from the center of input.
- A point i is a potential outlier if its leverage >> the mean leverage value, L/n, where L is the sum of the leverage values.

- **Leverage =** the diagonal of the **influence matrix** $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$
  - $\boldsymbol{h_{ii}} = \dfrac{\partial \hat{y}_i}{\partial y_i} \approx$ how much influence to $\hat{y}_i$ by changing of $\boldsymbol{y_i}$.



More Details on Influence Matrix

- **Influence matrix** $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$
- $\boldsymbol{h}_{ii} = \frac{1}{n} + \frac{(xi - \bar{x})}{SX}$
- $\boldsymbol{SX} = \sum_{i=1}^{n} x_i{}^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2$

- http://en.wikipedia.org/wiki/Leverage_%28statistics%29
- http://en.wikipedia.org/wiki/Hat_matrix

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \Rightarrow X^TX = \begin{bmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix}$$

$$\Rightarrow (X^TX)^{-1} = \frac{1}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \begin{bmatrix} \sum_{i=1}^{n} x_i^2 & -\sum_{i=1}^{n} x_i \\ -\sum_{i=1}^{n} x_i & n \end{bmatrix}$$

Note the definition of

$$SXX = \sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2$$

Hence you may rewrite

$$(X^TX)^{-1} = \frac{1}{SXX} \begin{bmatrix} \frac{1}{n}\sum_{i=1}^{n} x_i^2 & -\bar{x} \\ -\bar{x} & 1 \end{bmatrix}$$

$$\Rightarrow X(X^TX)^{-1} = \frac{1}{SXX} \begin{bmatrix} \frac{1}{n}\sum_{i=1}^{n} x_i^2 - x_1\bar{x} & -\bar{x} + x_1 \\ \frac{1}{n}\sum_{i=1}^{n} x_i^2 - x_2\bar{x} & -\bar{x} + x_2 \\ \vdots & \vdots \\ \frac{1}{n}\sum_{i=1}^{n} x_i^2 - x_n\bar{x} & -\bar{x} + x_n \end{bmatrix}$$

Hence,

$$h_{ii} = \frac{1}{SXX}\left[1 \times \left(\frac{1}{n}\sum_{j=1}^{n} x_j^2 - x_i\bar{x}\right) + x_i \times (-\bar{x} + x_i)\right]$$

$$= \frac{1}{SXX}\left[\frac{1}{n}\sum_{j=1}^{n} x_j^2 - (\bar{x})^2 + (\bar{x})^2 - 2x_i\bar{x} + x_i^2\right]$$

$$= \frac{1}{nSXX}\left[\sum_{j=1}^{n} x_j^2 - \frac{1}{n}\left(\sum_{j=1}^{n} x_j\right)^2\right] + \frac{(x_i - \bar{x})^2}{SXX}$$

$$= \frac{1}{n} + \frac{(x_i - \bar{x})^2}{SXX}$$

http://www.talkstats.com/showthread.php/15280-How-to-prove-the-formula-for-leverage-diagonal-of-H-matrix

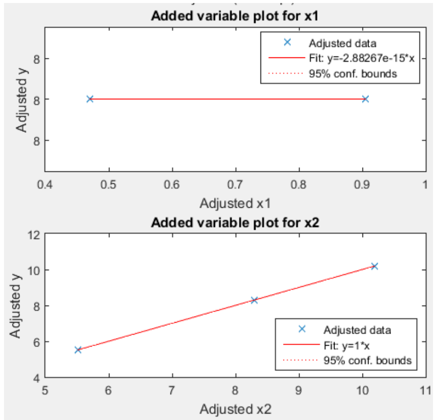
Cook’s Distance

- Cook's distance measures the effect of deleting a given observation.
  - Shows the influence of each observation to the fitted response (predicted $\hat{y}$).
  - **A point likely be an outlier IF Cook’s distance > (3 × average Cook’s distance).**
  - Points w/ ↑residuals (outliers) and/or ↑leverage may distort LR and its accuracy.
  - Points w/ ↑Cook's distance are considered to merit closer check in the analysis.

- Cook’s Distance = $\boxed{D_i = \frac{\sum_{j=1}^{n}(\hat{Y}_j - \hat{Y}_{j(i)})^2}{p \text{ MSE}}}$
  - $\hat{y}_j$ is the prediction from the full regression model for observation j;
  - $\hat{y}_{i_{(j)}}$ is the prediction for observation j from a refitted regression model in which observation i has been **deleted**;
  - $\boldsymbol{p}$ is the number of fitted parameters in the model;
  - MSE is the mean square error of the regression model.
  - http://en.wikipedia.org/wiki/Cook%27s_distance

Added-Variable Plot ≈ plotSlice()

- Construction of an Added-Variable Plot.
  - Let the original LR be $\hat{y} = \theta^{\mathrm{T}}\boldsymbol{X} = \theta_0 \times \mathbf{1} + \theta_1 \times \boldsymbol{x}_1 + \theta_2 \times \boldsymbol{x}_2 + \theta_3 \times \boldsymbol{x}_3$.
  - Build a model with variables $\boldsymbol{x}_2$ **&** $\boldsymbol{x}_3$ **against** $\boldsymbol{y}$ as $\widehat{y_{i1}} = \theta^{\mathrm{T}}\boldsymbol{X} = \theta_0 + \theta_2 \times \boldsymbol{x}_2 + \theta_3 \times \boldsymbol{x}_3$.
  - Compute residual $\boldsymbol{R}_{yi} = y_i - \widehat{y_{i1}}$.
  - Build a model with variables $\boldsymbol{x}_2$ & $\boldsymbol{x}_3$ **against** $\boldsymbol{x}_1$ as $\widehat{x_{i1}} = \theta^{\mathrm{T}}\boldsymbol{X} = \theta_0 + \theta_2 \times \boldsymbol{x}_2 + \theta_3 \times \boldsymbol{x}_3$.
  - Compute residual $\boldsymbol{R}_{xi} = x_{i1} - \hat{x}_{i1}$.
  - Build a regression with $\boldsymbol{R}_{yi}$ against $\boldsymbol{R}_{xi}$.
    - i.e. find a slope to fit $\boldsymbol{R}_{xi}$ to $\boldsymbol{R}_{yi}$.



- **Purposes and interpretation:**
  - **If a line i near horizontal, then variable x_i is insignificant.**
    - $\boldsymbol{R}_{yi} = y_i - \widehat{y_{i1}}$ closes to be constant (low fit error) while $\boldsymbol{R}_{xi}$ keeps changing (or growing).
  - To evaluate the marginal role of each variable in the multiple regression models.
  - To evaluate if a variable is significantly associated with $\boldsymbol{Y}$ to be included in LR.
  - Also refer to as ***partial regression plots***.

Identify Insignificant Variables

- ***Ry*** closes to constant while $\boldsymbol{R_1}$ changing

```
rng(5),                Y = [5; 8; 10];
X = [rand(3, 1) Y];      % X1   X2, where X2 = Y

mdl = fitlm(X, Y)
figure, plot(mdl),    % whole add-var plot unable to tell important vars
figure,
for i = 1 : mdl.NumCoefficients,
  subplot(mdl.NumCoefficients, 1, i),
  plotAdded(mdl, mdl.CoefficientNames{i});
end
plotSlice(mdl)
```