Regression and its Applications

Workshop

Albert C. Yang, M.D., Ph.D.

Institutes of Brain Science/Digital Medicine Center

National Yang-Ming University

Mar 26, 2019

accyang@gmail.com

Objectives

- Read data into Matlab
- Simple linear regression
- Evaluate fitness of the model
- Training and testing data
- Evaluate trained model in the testing data

Dataset

https://www.kaggle.com/mirichoi0218/insurance

Medical Insurance Dataset

	A	B	C	D	E	F	G
1	age	sex	bmi	children	smoker	region	charges
2	19	female	27.9	0	yes	southwest	16884.92
3	18	male	33.77	1	no	southeast	1725.552
4	28	male	33	3	no	southeast	4449.462
5	33	male	22.705	0	no	northwest	21984.47
6	32	male	28.88	0	no	northwest	3866.855
7	31	female	25.74	0	no	southeast	3756.622
8	46	female	33.44	1	no	southeast	8240.59
9	37	female	27.74	3	no	northwest	7281.506
10	37	male	29.83	2	no	northeast	6406.411
11	60	female	25.84	0	no	northwest	28923.14
12	25	male	26.22	0	no	northeast	2721.321
13	62	female	26.29	0	yes	southeast	27808.73
14	23	male	34.4	0	no	southwest	1826.843
15	56	female	39.82	0	no	southeast	11090.72
16	27	male	42.13	0	yes	southeast	39611.76
17	19	male	24.6	1	no	southwest	1837.237
18	52	female	30.78	1	no	northeast	10797.34
19	23	male	23.845	0	no	northeast	2395.172
20	56	male	40.3	0	no	southwest	10602.39
21	30	male	35.3	0	yes	southwest	36837.47
22	60	female	36.005	0	no	northeast	13228.85

Read Data into Matlab

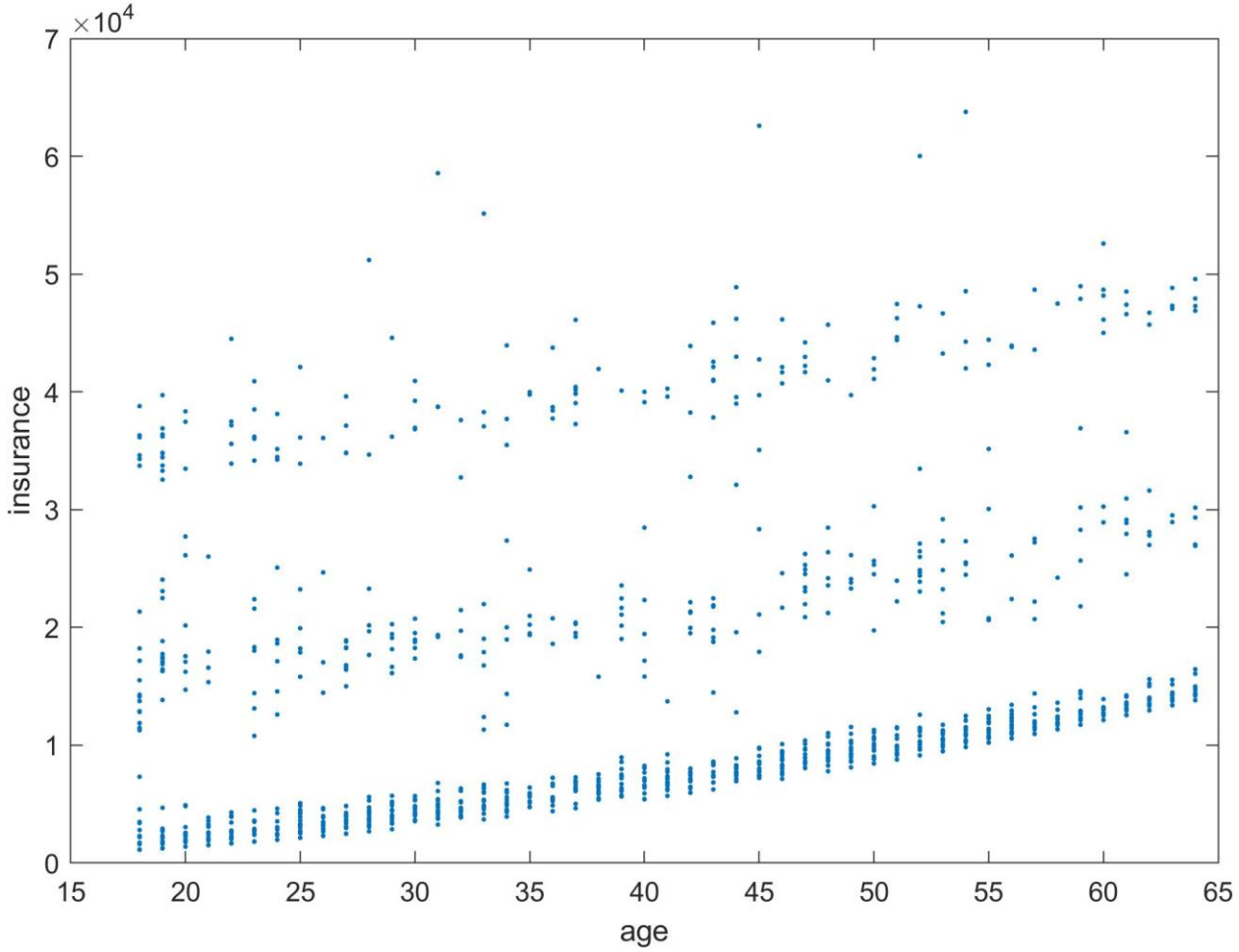
Read Data into Matlab

- [num,txt,raw] = xlsread('insurance.csv’);
- age=num(:,1);
- insurance=num(:,7);

Relationship between Age and Insurance Claims

- plot(age,insurance,'.');
- xlabel('age');
- ylabel('insurance');

Relationship between Age and Insurance Claims



Simple Linear Regression

Fitting Simple Linear Regression Model

- model1 = fitlm(age,insurance)

```
>> model1 = fitlm(age,insurance)

model1 =

Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE       tStat       pValue
                   ________    _______    _______    __________

    (Intercept)     3165.9      937.15     3.3782     0.0007506
    x1              257.72      22.502     11.453     4.8867e-29


Number of observations: 1338, Error degrees of freedom: 1336
Root Mean Squared Error: 1.16e+04
R-squared: 0.0894,  Adjusted R-Squared 0.0887
F-statistic vs. constant model: 131, p-value = 4.89e-29
```

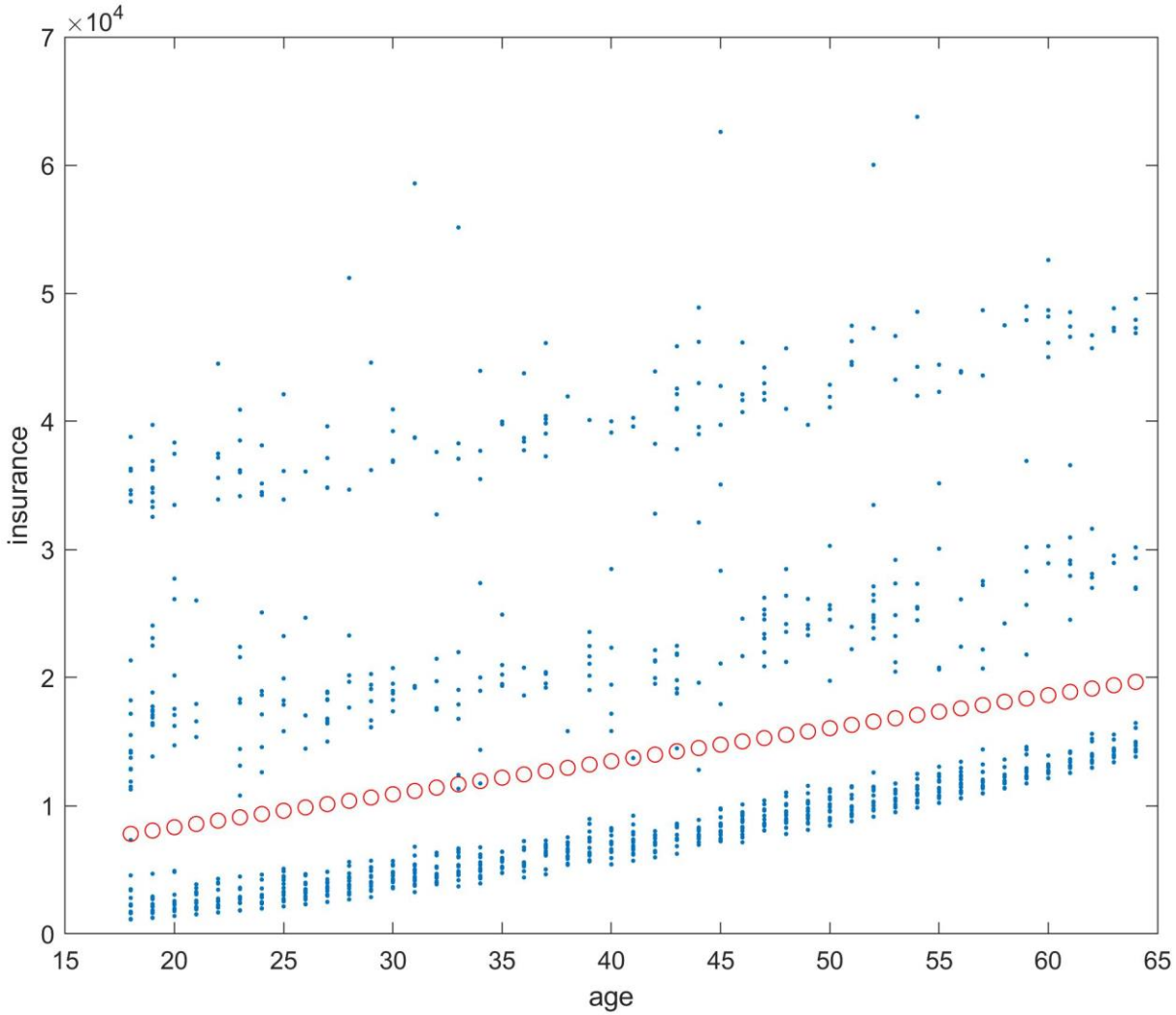
insurance = 3165.9 + 257.72*age

Visualization of Regression Results

- ypred = predict(model1,age);

- plot(age,insurance,'.');
- hold on
- plot(age,ypred,'ro');
- xlabel('age');
- ylabel('insurance');

Visualization of Regression Results



Evaluate Fitness of the Model

Model Evaluation

- Mean Squared Error(MSE)

$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \widehat{y}\right)}^{2}$$

The square of the difference between actual and predicted

- Root-Mean-Squared-Error(RMSE)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(Predicted_i - Actual_i\right)^2}{N}}$$

Model Evaluation

- R^2 or Coefficient of Determination

$$R^2 = 1 - \frac{\text{MSE(model)}}{\text{MSE(baseline)}}$$

- Adjusted R^2

$$R_a^2 = 1 - \left[\left(\frac{n-1}{n-k-1}\right) \times (1 - R^2)\right]$$

where:

n = number of observations

k = number of independent variables

R_a^2 = adjusted R^2

Model Evaluation

- model1.RMSE;

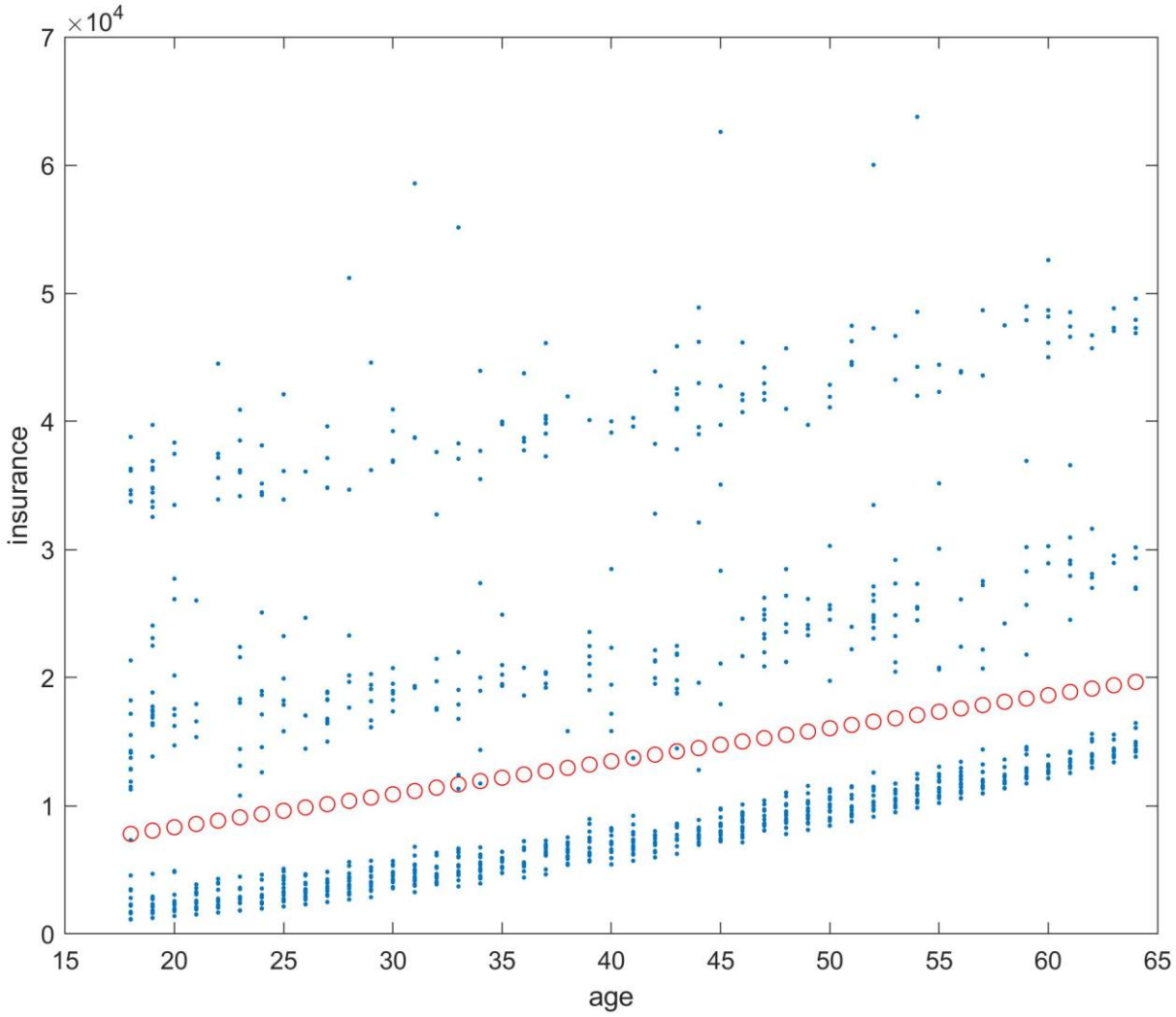
```
>> model1.RMSE

ans =

    1.1560e+04
```

Training and Testing Data

Is Model Too Good to Be True?



Divide Data Into Training and Testing Subset

	A	B	C	D	E	F	G
1	age	sex	bmi	children	smoker	region	charges
2	19	female	27.9	0	yes	southwest	16884.92
3	18	male	33.77	1	no	southeast	1725.552
4	28	male	33	3	no	southeast	4449.462
5	33	male	22.705	0	no	northwest	21984.47
6	32	male	28.88	0	no	northwest	3866.855
7	31	female	25.74	0	no	southeast	3756.622
8	46	female	33.44	1	no	southeast	8240.59
9	37	female	27.74	3	no	northwest	7281.506
10	37	male	29.83	2	no	northeast	6406.411
11	60	female	25.84	0	no	northwest	28923.14
12	25	male	26.22	0	no	northeast	2721.321
13	62	female	26.29	0	yes	southeast	27808.73
14	23	male	34.4	0	no	southwest	1826.843
15	56	female	39.82	0	no	southeast	11090.72
16	27	male	42.13	0	yes	southeast	39611.76
17	19	male	24.6	1	no	southwest	1837.237
18	52	female	30.78	1	no	northeast	10797.34
19	23	male	23.845	0	no	northeast	2395.172
20	56	male	40.3	0	no	southwest	10602.39
21	30	male	35.3	0	yes	southwest	36837.47
22	60	female	36.005	0	no	northeast	13228.85

Training: Testing = 7:3

Divide Data Into Training and Testing Subset

randsample

Random sample

Syntax

```
y = randsample(n,k)
y = randsample(population,k)
y = randsample( ___ ,replacement)
y = randsample(n,k,true,w)
y = randsample(population,k,true,w)
y = randsample(s, ___ )
```

Description

`y = randsample(n,k)` returns `k` values sampled uniformly at random, without replacement, from the integers `1` to `n`.

Divide Data Into Training and Testing Subset

- test_index = zeros(length(insurance),1);
- test_sample = randsample(length(insurance),fix(length(insurance)*0.3));
- test_index(test_sample) = 1;
- train_index = ~test_index;

- train_data = data(train_index==1,:);
- test_data = data(test_index==1,:);

Evaluate Trained Model in Testing Data

Fit Training Data

- model2 = fitlm(train_data(:,1),train_data(:,2))

```
>> model2 = fitlm(train_data(:,1),train_data(:,2))

model2 =


Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate       SE       tStat       pValue
                   ________    ______    _______    __________

    (Intercept)     4675.5      1108.2    4.2189      2.694e-05
    x1              217.38      26.397    8.2353      5.9966e-16


Number of observations: 937, Error degrees of freedom: 935
Root Mean Squared Error: 1.14e+04
R-squared: 0.0676,  Adjusted R-Squared 0.0666
F-statistic vs. constant model: 67.8, p-value = 6e-16
```

Predict Response in Testing Data

- ypred = predict(model2,test_data(:,1));

- RMSE_test = sqrt(sum((ypred-test_data(:,2)).^2))

```
>> RMSE_test = sqrt(sum((ypred-test_data(:,2)).^2))

RMSE_test =

   2.3768e+05
```

Model work poorly in testing data

```
>> RMSE_train = model2.RMSE

RMSE_train =

   1.1446e+04
```

Summary of Machine Learning Workflow

- Divide data into training and testing subset
- Model training data
- Evaluate trained model in training data
- Use trained model to predict response in testing data
- Evaluate model performance in testing data