APPLIED REGRESSION ANALYSIS
Lecture 2

Ch2: Simple Linear Regression

Gengxin li

Department of Mathematics and Statistics
University of Michigan dearborn

Example 2.1 The Rocket Propellant Data

A rocket motor is manufactured by bonding an igniter propellant and a sustainer propellant together inside a metal housing. The shear strength of the bond between the two types of propellant is an important quality characteristic.

It is suspected that shear strength is related to the age in weeks of the batch of sustainer propellant.

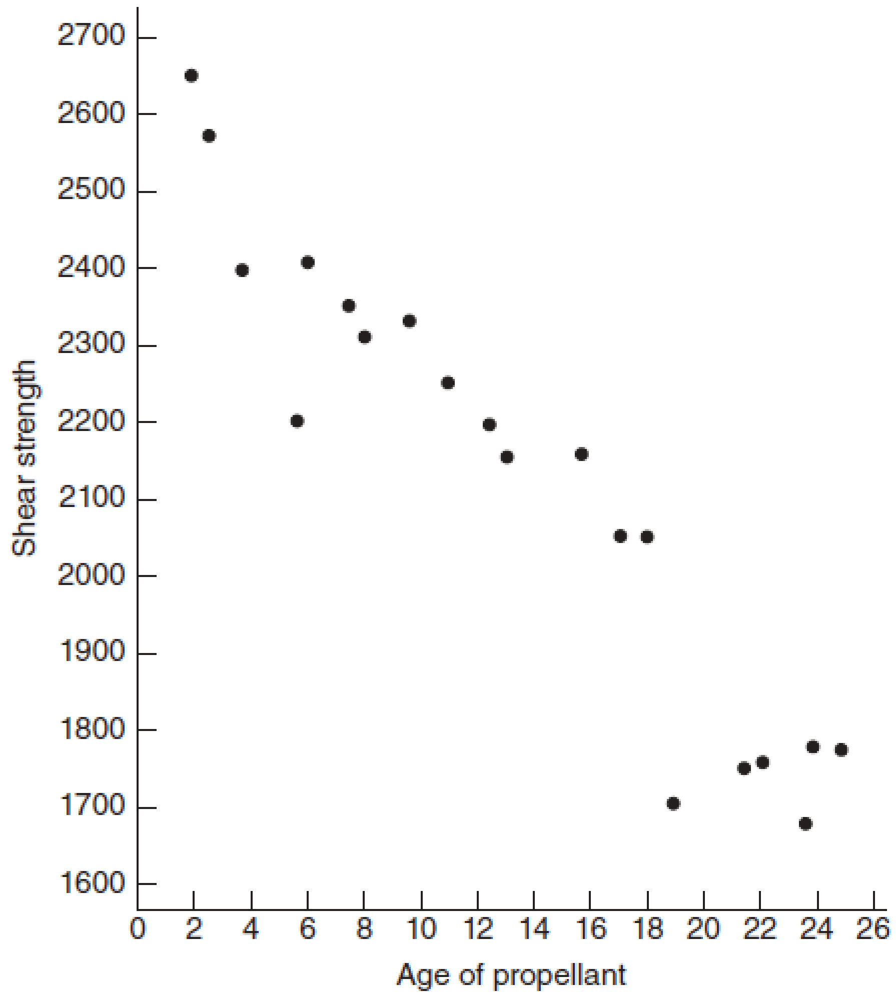
Twenty observations on shear strength and the age of the corresponding batch of propellant have been collected and are shown in Table 2.1.

TABLE 2.1 Data for Example 2.1

Observation, i	Shear Strength, y_i (psi)	Age of Propellant, x_i (weeks)
1	2158.70	15.50
2	1678.15	23.75
3	2316.00	8.00
4	2061.30	17.00
5	2207.50	5.50
6	1708.30	19.00
7	1784.70	24.00
8	2575.00	2.50
9	2357.90	7.50
10	2256.70	11.00
11	2165.20	13.00
12	2399.55	3.75
13	1779.80	25.00
14	2336.75	9.75
15	1765.30	22.00
16	2053.50	18.00
17	2414.40	6.00
18	2200.50	12.50
19	2654.20	2.00
20	1753.70	21.50

TABLE 2.1 Data for Example 2.1

Observation, i	Shear Strength, y_i (psi)	Age of Propellant, x_i (weeks)
1	2158.70	15.50
2	1678.15	23.75
3	2316.00	8.00
4	2061.30	17.00
5	2207.50	5.50
6	1708.30	19.00
7	1784.70	24.00
8	2575.00	2.50
9	2357.90	7.50
10	2256.70	11.00
11	2165.20	13.00
12	2399.55	3.75
13	1779.80	25.00
14	2336.75	9.75
15	1765.30	22.00
16	2053.50	18.00
17	2414.40	6.00
18	2200.50	12.50
19	2654.20	2.00
20	1753.70	21.50



$$b_1 = \frac{S_{xy}}{S_{xx}}\ ,\ b_0 = \bar{y} - b_1\bar{x}$$

The scatter diagram suggests that there is a strong statistical relationship between shear strength and propellant age.

Observation, i	Shear Strength, y_i (psi)	Age of Propellant, x_i (weeks)	y_i ^2	x_i ^2	xiyi
1	2158.7	15.5	4659986	240.25	33459.85
2	1678.15	23.75	2816187	564.0625	39856.06
3	2316	8	5363856	64	18528
4	2061.3	17	4248958	289	35042.1
5	2207.5	5.5	4873056	30.25	12141.25
6	1708.3	19	2918289	361	32457.7
7	1784.7	24	3185154	576	42832.8
8	2575	2.5	6630625	6.25	6437.5
9	2357.9	7.5	5559692	56.25	17684.25
10	2256.7	11	5092695	121	24823.7
11	2165.2	13	4688091	169	28147.6
12	2399.55	3.75	5757840	14.0625	8998.313
13	1779.8	25	3167688	625	44495
14	2336.75	9.75	5460401	95.0625	22783.31
15	1765.3	22	3116284	484	38836.6
16	2053.5	18	4216862	324	36963
17	2414.4	6	5829327	36	14486.4
18	2200.5	12.5	4842200	156.25	27506.25
19	2654.2	2	7044778	4	5308.4
20	1753.7	21.5	3075464	462.25	37704.55
total	42627.15	267.25	92547433	4677.688	528492.6
mean	2131.358	13.3625			

$$\boldsymbol{b_1 = \frac{S_{xy}}{S_{xx}}\ ,\ b_0 = \bar{y} - b_1\bar{x}}$$

$$S_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2 = \sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n} = 4677.9 - \frac{(267.25)^2}{20}$$
$$= 1106.56$$

$$S_{xy} = \sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}) = \sum_{i=1}^{n} x_i y_i - \frac{\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n} =$$
$$528492.64 - \frac{267.25 * 42627.15}{20} = -41112.65$$

$$b_1 = \frac{S_{xy}}{S_{xx}} = \frac{-41112.65}{1106.56} = -37.15$$ (**negative linear relationship**)

$$b_0 = \bar{y} - b_1\bar{x} = 2131.36 - (-37.15) * 13.36 = 2627.82$$

The fitted line will be

$$\hat{y} = b_0 + b_1 x = 2627.82 - 37.15x$$

Use R to analyze Rocket Propellant data

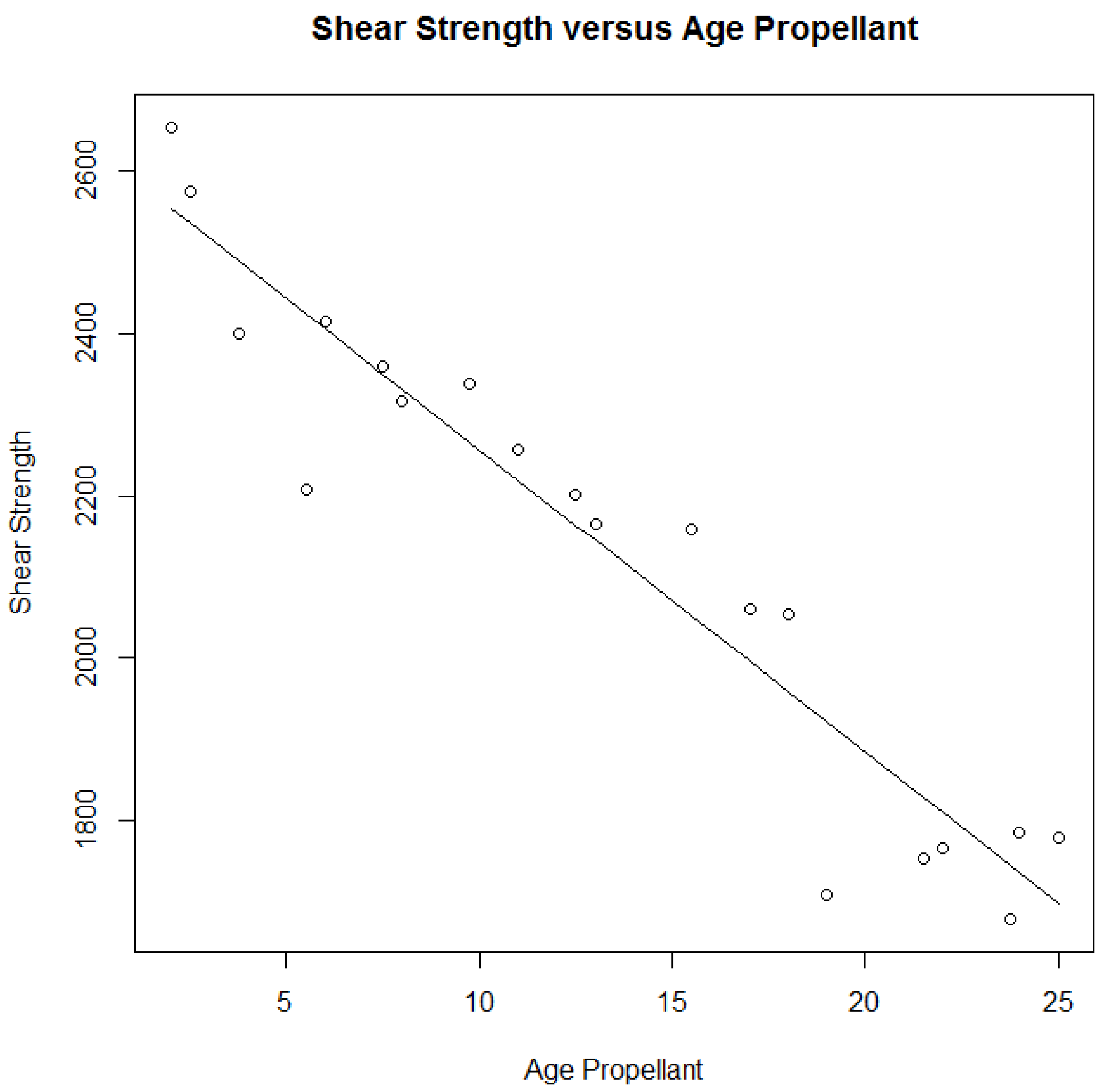
```
> #The Rocket Propellant Data
>
> setwd('F:\\UMD\\teaching\\STT530\\Lab')
> propellant <- read.csv("data-ex-2-1.csv", header=T)
> attach(propellant)
> dim(propellant)
[1] 20  3
> propellant
   id       y     x
1   1 2158.70 15.50
2   2 1678.15 23.75
3   3 2316.00  8.00
4   4 2061.30 17.00
5   5 2207.50  5.50
6   6 1708.30 19.00
7   7 1784.70 24.00
8   8 2575.00  2.50
9   9 2357.90  7.50
10 10 2256.70 11.00
11 11 2165.20 13.00
12 12 2399.55  3.75
13 13 1779.80 25.00
14 14 2336.75  9.75
15 15 1765.30 22.00
16 16 2053.50 18.00
17 17 2414.40  6.00
18 18 2200.50 12.50
19 19 2654.20  2.00
20 20 1753.70 21.50
```

```
> model <- lm(y ~ x)
> summary(model)

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-215.98  -50.68   28.74   66.61  106.76

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 2627.822     44.184   59.48  < 2e-16 ***
x            -37.154      2.889  -12.86 1.64e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 96.11 on 18 degrees of freedom
Multiple R-squared:  0.9018,    Adjusted R-squared:  0.8964
F-statistic: 165.4 on 1 and 18 DF,  p-value: 1.643e-10
```

The fitted line will be

$$\hat{y} = b_0 + b_1 x = 2627.82 - 37.15x$$

```
> confint(model)
                 2.5 %    97.5 %
(Intercept) 2534.99540 2720.6493
x            -43.22338  -31.0838
>
> plot(x, y,
+      xlab="Age Propellant", ylab="Shear Strength",
+      main="Shear Strength versus Age Propellant",
+      panel.last = lines(sort(x), fitted(model)[order(x)]))
```



Use Python to analyze Rocket Propellant data

Environment: python 3.9 + PyCharm
I: Import necessary packages

```
# import packages

import ax
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.pyplot import subplots
import statsmodels.api as sm

from statsmodels.stats.outliers_influence \
    import variance_inflation_factor as VIF
from statsmodels.stats.anova import anova_lm

from ISLP import load_data
from ISLP.models import (ModelSpec as MS, summarize ,poly)
```

II: Load propellant data set

Code

```
# load chapter 2 exercise data set
propellant = pd.read_csv('data-ex-2-1.csv')
print(propellant.shape)
print(propellant)
```

Output

```
    id        y      x
0    1  2158.70  15.50
1    2  1678.15  23.75
2    3  2316.00   8.00
3    4  2061.30  17.00
4    5  2207.50   5.50
5    6  1708.30  19.00
6    7  1784.70  24.00
7    8  2575.00   2.50
8    9  2357.90   7.50
9   10  2256.70  11.00
10  11  2165.20  13.00
11  12  2399.55   3.75
12  13  1779.80  25.00
13  14  2336.75   9.75
14  15  1765.30  22.00
15  16  2053.50  18.00
16  17  2414.40   6.00
17  18  2200.50  12.50
18  19  2654.20   2.00
19  20  1753.70  21.50
```

II: Fit a simple linear regression model

Code

```python
# fit a simple linear regression model
X = MS(['x']).fit_transform(propellant)
y = propellant['y']
model1 = sm.OLS(y, X)
results1 = model1.fit()
print(summarize(results1))
```

Output

```
                coef  std err        t  P>|t|
intercept  2627.8224   44.184   59.475    0.0
x           -37.1536    2.889  -12.860    0.0
```

The fitted line will be

$$\hat{y} = b_0 + b_1 x = 2627.82 - 37.15x$$

III: Confidence Interval and Draw a fitted line

Code

```
# CI
CI = results1.conf_int(alpha=0.05)
print(CI)

# plot fitted line

def abline(ax, b, m):  1 usage  new *
    "Add a line with slope m and intercept b to ax"
    xlim = ax.get_xlim()
    ylim = [m * xlim[0] + b, m * xlim[1] + b]
    ax.plot(xlim , ylim)
```

Output

	0	1
intercept	2534.995405	2720.649313
x	-43.223379	-31.083803