Ve406 Lecture 9

Jing Liu

UM-SJTU Joint Institute

June 11, 2018

- As we have discussed, the relationship between the response

$$Y_i \qquad i = 1, 2, \ldots, n$$

and two or more predictors, k of them in general,

$$X_{i1}, \quad X_{i2}, \qquad \ldots \qquad X_{ij}, \qquad \ldots \qquad X_{ik} \qquad i = 1, 2, \ldots, n$$

can be well described by a hyperplane locally, that is,

1. The conditional mean of the response is given by

$$\begin{aligned} \mathbb{E}\left[Y_i \mid X_{i1}, X_{i2}, \ldots, X_{ik}\right] &= \mathbb{E}\left[Y_i \mid \mathbf{X}_i\right] \\ &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} \end{aligned}$$

- For a a general region, there is no reason to expect it is a flat hyperplane.

- Polynomials are the easiest way to add curvature to the hyperplane

$$\mathbb{E}\left[Y_i \mid X_{i1}, X_{i2}, \ldots, X_{ik}\right] = \beta_0 + P_1^{d_1}(x_{i1}) + P_2^{d_2}(x_{i2}) + \cdots + P_k^{d_k}(x_{ik})$$

where $P_j^{d_j}(x_{ij}) = \sum_{\ell=1}^{d_j} \beta_{j\ell} x_{ij}^{\ell}$ is a d_jth degree polynomial of jth predictor.

- This is often known as polynomial regression, which is still considered to be linear regression. Because the coefficients still have a linear relation with

$$\mathbb{E}\left[\mathbf{Y} \mid \mathbf{X}\right] = \mathbf{X}\boldsymbol{\beta}$$

- Except the interpretation of coefficients

$$\beta_{j\ell}$$

becomes more difficult, the rest of what we have done remain as they were.

- Consider the following tyre abrasion data

`abloss`	abrasion loss in gm/hr.
`hard`	the hardness in Shore units.
`tensile`	tensile strength in kg/sq meters

- All three variables seem to be correctly stored

```
> rubber.df = read.table("~/Desktop/rubber.txt",
+                          header = TRUE)
```

```
> str(rubber.df)
```

```
'data.frame':   30 obs. of  3 variables:
 $ hardness: int  45 61 71 81 53 64 79 56 75 88 ...
 $ tensile : int  162 232 231 224 203 210 196 200 188 119 ...
 $ abloss  : int  372 175 136 55 221 164 82 228 128 64 ...
```

- There seem to be no unusual value in the data according to summary

```
> summary(rubber.df)
```

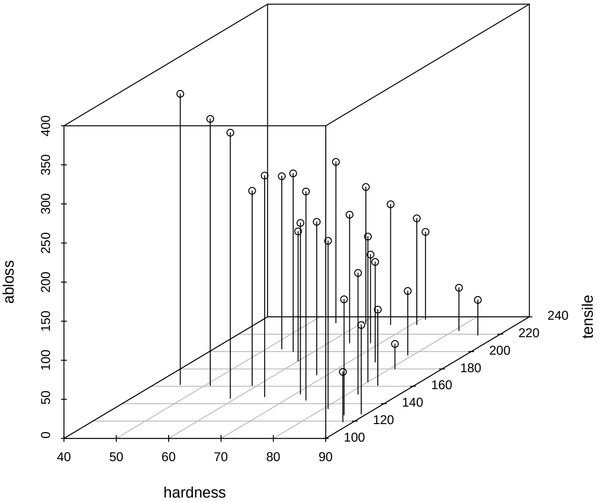
```
    hardness         tensile          abloss     
 Min.   :45.00   Min.   :119.0   Min.   : 32.0  
 1st Qu.:60.25   1st Qu.:151.0   1st Qu.:113.2  
 Median :71.00   Median :176.5   Median :165.0  
 Mean   :70.27   Mean   :180.5   Mean   :175.4  
 3rd Qu.:81.00   3rd Qu.:210.0   3rd Qu.:220.5  
 Max.   :89.00   Max.   :237.0   Max.   :372.0  
```

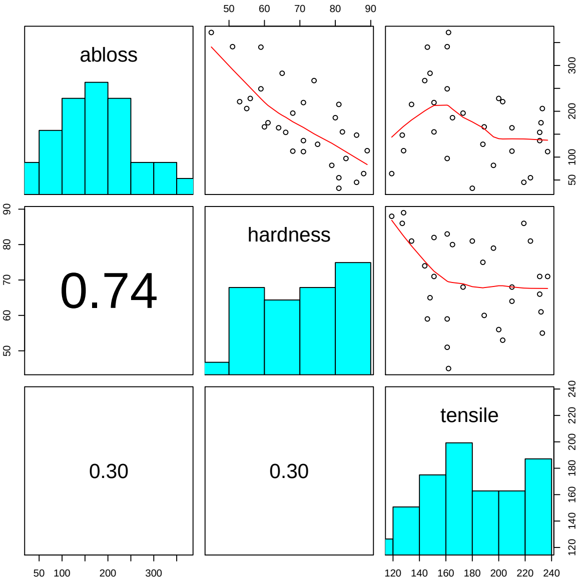
and a 3D scatter plot confirms that, and indicated a reasonably flat plane.

```
> library(scatterplot3d)
> with(rubber.df,
+      scatterplot3d(
+        hardness, tensile, abloss, type = "h"))
```

thus we move to paris plot for pairwise relationship

```
> pairs(rubber.df[, c(3, 1:2)],
+       diag.panel = panel.hist,
+       lower.panel = panel.cor,
+       upper.panel = panel.smooth)
```


abloss
hardness
tensile
0
50
100
150
200
250
300
350
400
40
50
60
70
80
90
100
120
140
160
180
200
220
240


abloss
hardness
tensile
0.74
0.30
0.30
50
60
70
80
90
300
200
100
50
90
80
70
60
50
240
220
200
180
160
140
120
50
100
200
300
120
140
160
180
200
220
240

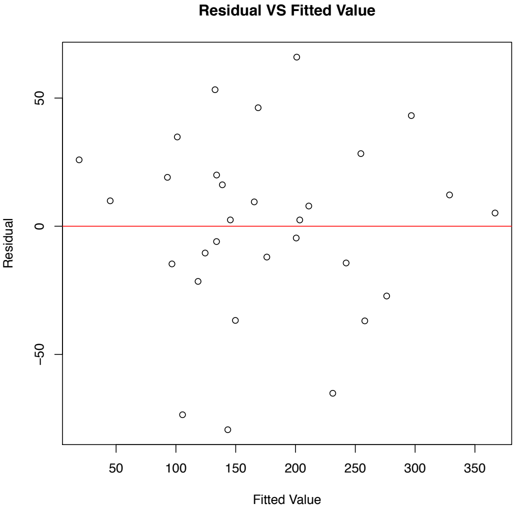
- Pairs plot indicates

1. There is a strong linear relationship between `abloss` and `hardness`.
2. The linear relationship between `abloss` and `tensile` is relatively weak.
3. The linear relationship between `hardness` and `tensile` is relatively weak.
4. There might be nonlinear relationship between `abloss` and `tensile`.

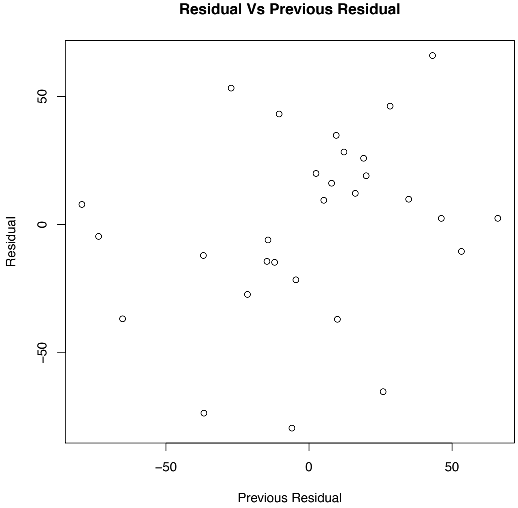
- Thus we might need to fit a polynomial terms to `tensile`.

```
> # Multiple Linear is a good place to start
> rubber.LM = lm(abloss~., data = rubber.df)
>
> res = rubber.LM$residuals
> fit = rubber.LM$fitted.values
>
> # Wrapper for residual plot, res vs previous
> # QQ normal plot and partial residual plots,
> diagnostic_plot_func(res, fit)
```

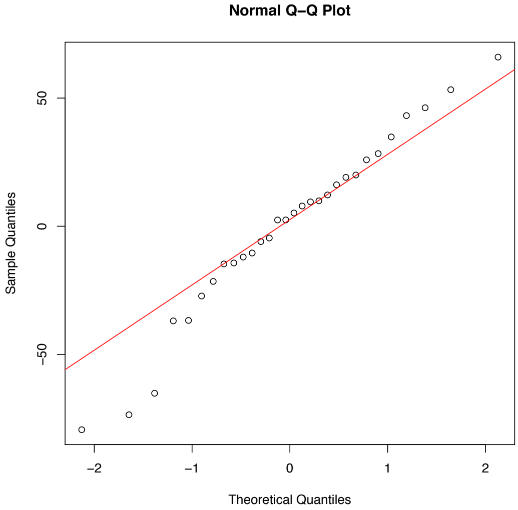
- The linear assumption seems to be reasonable,



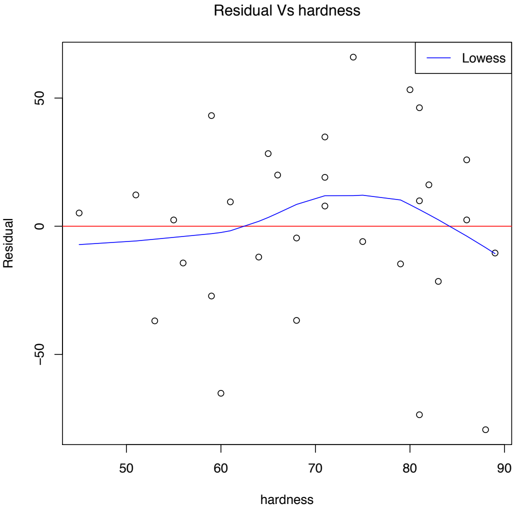
- There is no clear indication of autocorrelation,



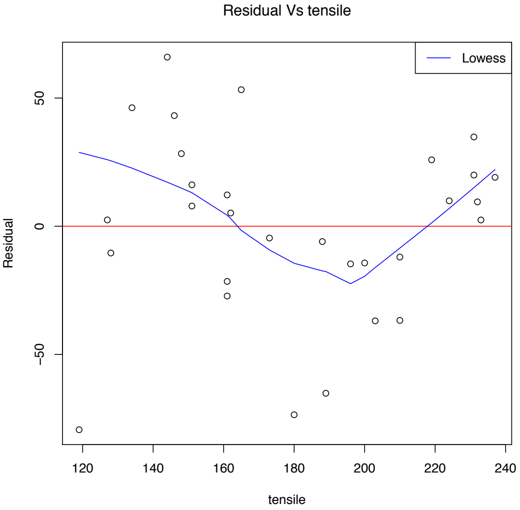
- There might be a problem with normality.



- The partial residual plot for `hardness` is reasonably straight.



- But the partial residual plot for `tensile` seems to have more curvature in it.



- Let us begin with considering a quadratic term for tensile

```
> # Create a quadratic term as a new variable
> rubber.df$sqtensile = rubber.df$tensile^2
>
> head(rubber.df, 10)
```

```
   hardness tensile abloss sqtensile
1        45     162    372     26244
2        61     232    175     53824
3        71     231    136     53361
4        81     224     55     50176
5        53     203    221     41209
6        64     210    164     44100
7        79     196     82     38416
8        56     200    228     40000
9        75     188    128     35344
10       88     119     64     14161
```

- Treating the quadratic term as a new variable

```
> rubber.q.LM = lm(abloss~., data = rubber.df)
>
> summary(rubber.q.LM)
```

```
Call:
lm(formula = abloss ~ ., data = rubber.df)

Residuals:
    Min      1Q  Median      3Q     Max
-92.223 -13.725   1.978  18.280  65.253

Coefficients:
              Estimate Std. Error t value Pr(>t)
(Intercept)  1.082e+03  2.323e+02   4.659 8.27e-05 ***
hardness    -6.760e+00  6.239e-01 -10.836 3.89e-11 ***
tensile     -3.461e+00  2.377e+00  -1.456    0.157
sqtensile    5.690e-03  6.457e-03   0.881    0.386
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 36.64 on 26 degrees of freedom
Multiple R-squared:  0.8449,    Adjusted R-squared:  0.827
F-statistic:  47.2 on 3 and 26 DF,  p-value: 1.167e-10
```

- The following is equivalent to what we have done

```
> rubber.q.LM = lm(abloss~hardness+tensile
+                  +I(tensile^2), data = rubber.df)
>
> summary(rubber.q.LM)
```

```
Call:
lm(formula = abloss ~ hardness + tensile + I(tensile^2), data = rubber.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-92.223 -13.725   1.978  18.280  65.253 

Coefficients:
               Estimate Std. Error t value Pr(>t)    
(Intercept)   1.082e+03  2.323e+02   4.659 8.27e-05 ***
hardness     -6.760e+00  6.239e-01 -10.836 3.89e-11 ***
tensile      -3.461e+00  2.377e+00  -1.456    0.157    
I(tensile^2)  5.690e-03  6.457e-03   0.881    0.386    
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 36.64 on 26 degrees of freedom
Multiple R-squared:  0.8449,    Adjusted R-squared:  0.827 
F-statistic:  47.2 on 3 and 26 DF,  p-value: 1.167e-10
```

- The following is another approach without creating a new column

```
> rubber.q.LM = lm(abloss~hardness+
+                     poly(tensile, 2, raw = TRUE),
+                   data = rubber.df)
>
> summary(rubber.q.LM)
```

```
Call:
lm(formula = abloss ~ hardness + poly(tensile, 2, raw = TRUE),
    data = rubber.df)

Residuals:
    Min      1Q  Median      3Q     Max
-92.223 -13.725   1.978  18.280  65.253

Coefficients:
                               Estimate Std. Error t value Pr(>t)
(Intercept)                   1.082e+03  2.323e+02   4.659 8.27e-05 ***
hardness                     -6.760e+00  6.239e-01 -10.836 3.89e-11 ***
poly(tensile, 2, raw = TRUE)1 -3.461e+00  2.377e+00  -1.456    0.157
poly(tensile, 2, raw = TRUE)2  5.690e-03  6.457e-03   0.881    0.386
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 36.64 on 26 degrees of freedom
Multiple R-squared:  0.8449,    Adjusted R-squared:  0.827
F-statistic:  47.2 on 3 and 26 DF,  p-value: 1.167e-10
```

- To summarise the following three are the same

```
> # Not recommended
> rubber.df$sqtensile = rubber.df$tensile^2
> rubber.q.LM = lm(abloss~., data = rubber.df)
>
> # Not usually used
> rubber.q.LM = lm(abloss~hardness+
+                      poly(tensile, 2, raw = TRUE),
+                    data = rubber.df)
>
> # ^ has to be enclosed by I()
> rubber.q.LM = lm(abloss~hardness+tensile
                  +I(tensile^2), data = rubber.df)
```

- The following has the wrong syntax,

```
> rubber.q.LM = lm(abloss~hardness+tensile
+                    +tensile^2, data = rubber.df)
```

- The following does orthogonalisation to $\mathbf{X}$, so don't use it for interpretation

```
> rubber.q.LM =
+   lm(abloss~hardness+
+        poly(tensile, 2), data = rubber.df)
>
> summary(rubber.q.LM)
```

```
Call:
lm(formula = abloss ~ hardness + poly(tensile, 2), data = rubber.df)

Residuals:
    Min      1Q  Median      3Q     Max
-92.223 -13.725   1.978  18.280  65.253

Coefficients:
                   Estimate Std. Error t value Pr(>t)
(Intercept)        650.4687    44.3476  14.668 4.35e-14 ***
hardness            -6.7605     0.6239 -10.836 3.89e-11 ***
poly(tensile, 2)1 -274.1968    38.6324  -7.098 1.55e-07 ***
poly(tensile, 2)2   34.3979    39.0373   0.881    0.386
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 36.64 on 26 degrees of freedom
Multiple R-squared:  0.8449,    Adjusted R-squared:  0.827
F-statistic:  47.2 on 3 and 26 DF,  p-value: 1.167e-10
```

- Using `poly(tensile, 2)` isolates the effect of having a quadratic term

```
> summary(rubber.q.LM)$coefficients
```

```
                     Estimate Std. Error    t value     Pr(>t)
(Intercept)        650.468713 44.3475663  14.667518 4.349525e-14
hardness            -6.760466  0.6239102 -10.835638 3.885262e-11
poly(tensile, 2)1 -274.196819 38.6323890  -7.097589 1.545917e-07
poly(tensile, 2)2   34.397950 39.0372966   0.881156 3.863067e-01
```

from which we know there is a highly significant linear relationship between `abrasion` and `tensile`, but quadratic term does not seem to add much.

- The adjusted coefficient of determination agrees with the t-test

```
> summary(rubber.LM)$adj.r.squared
```

```
[1] 0.8283967
```

```
> summary(rubber.q.LM)$adj.r.squared
```

```
[1] 0.826964
```

- We can do the same to hardness,

```
> rubber.q.LM =
+   lm(abloss~poly(hardness,2)
+      +tensile, data = rubber.df)
>
> summary(rubber.q.LM)
```

```
Call:
lm(formula = abloss ~ poly(hardness, 2) + tensile, data = rubber.df)

Residuals:
    Min      1Q  Median      3Q     Max
-72.724 -16.444   8.033  18.789  55.824

Coefficients:
                    Estimate Std. Error t value Pr(>t)
(Intercept)         440.3045    38.4275  11.458 1.16e-11 ***
poly(hardness, 2)1 -436.3713    38.3275 -11.385 1.33e-11 ***
poly(hardness, 2)2  -45.0596    39.3883  -1.144    0.263
tensile              -1.4677     0.2097  -6.998 1.98e-07 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 36.28 on 26 degrees of freedom
Multiple R-squared:  0.8479,    Adjusted R-squared:  0.8303
F-statistic: 48.31 on 3 and 26 DF,  p-value: 9.051e-11
```

- Again t-test suggests we don't really need a quadratic term,

```
> summary(rubber.LM)$adj.r.squared
```

```
[1] 0.8283967
```

```
> summary(rubber.q.LM)$adj.r.squared
```

```
[1] 0.8303365
```

and the adjusted R^2 only increases by a tiny amount.

- I would leave out the quadratic terms in favour of a simpler model.
- Normality is the only other thing that might be problematic since $n = 30$.
- However, Shapiro-Wilk indicates that we could see the QQ plot 52% of time.

```
> shapiro.test(rubber.LM$residuals)
```

```
        Shapiro-Wilk normality test

data:  rubber.LM$residuals
W = 0.96918, p-value = 0.5171
```