Ve406 Lecture 10

Jing Liu

UM-SJTU Joint Institute

June 13, 2018

- So far we have only considered continuous predictor variables, e.g.

`height`, `hardness`, `tensile` and etc

- But not all variables are continuous, there are variables take only a specified, but finite, set of possible values/levels, e.g.

`gender`

  we call such a variable a factor.

- Factors enter model formulae in the same way as continuous variables, but the interpretation of the output is slightly different.
- Instead of having only one coefficient for a continuous variable X_i, e.g. `Age`

```
Salary ~  Age + Gender + Ethnicity
```

  a factor variable, e.g. `Age` or `Ethnicity` gives rise to

$$L - 1$$

  coefficients, where L is the number of distinct levels of the factor.

- Suppose we have the followings

		Level 1	Level 2	Level 3
`Age`	X_1			
`Gender`	X_2	Female	Male	
`Ethnicity`	X_3	Chinese	Hispanic	Latino

then the output have $1+1+1+2=5$ coefficients and the model becomes

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 I_m + \hat{\beta}_3 I_h + \hat{\beta}_3 I_l$$

where

$$I_m = \begin{cases} 1 & \text{if } \texttt{Gender=male}. \\ 0 & \text{if } \texttt{Gender=female}; \end{cases}$$

and

$$I_h = \begin{cases} 1 & \text{if } \texttt{Ethnicity=Hispanic}; \\ 0 & \text{otherwise}; \end{cases} \quad ; \quad I_l = \begin{cases} 1 & \text{if } \texttt{Ethnicity=Latino}; \\ 0 & \text{otherwise0} \end{cases}$$

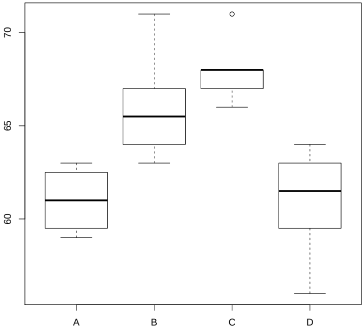
Q: What does that mean in terms of the regression line?

Q: Anyone remember one way analysis of variance?

Q: Can you see one way anova is just a special case of linear regression?

- Consider the following dataset

`coag`	Blood coagulation time
`diet`	Four types of diets



- Recall one way analysis of variance compares the means

```
> coag.LM = lm(coag ~ diet, data = coag.df)
> anova(coag.LM)
```

```
Analysis of Variance Table

Response: coag
          Df Sum Sq Mean Sq F value    Pr(>F)
diet       3    228    76.0  13.571 4.658e-05 ***
Residuals 20    112     5.6
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1
```

which is equivalent to

```
> summary(coag.LM)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>t)
(Intercept) 6.100e+01  1.183e+00  51.554  < 2e-16 ***
dietB       5.000e+00  1.528e+00   3.273 0.003803 **
dietC       7.000e+00  1.528e+00   4.583 0.000181 ***
dietD       2.991e-15  1.449e+00   0.000 1.000000
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.366 on 20 degrees of freedom
Multiple R-squared:  0.6706,    Adjusted R-squared:  0.6212
F-statistic: 13.57 on 3 and 20 DF,  p-value: 4.658e-05
```

Q: Can you remember two-way analysis of variance?

- Consider the following dataset

gain	Weight gain
source	Source of protein
amount	High or low

```
> diets.LM = lm(gain ~ source + level
+                  + source:level, data=diets.df)
> anova(diets.LM)
```

```
Analysis of Variance Table

Response: gain
             Df  Sum Sq Mean Sq F value    Pr(>F)
source        2   266.5   133.3  0.6211 0.5411319
level         1  3168.3  3168.3 14.7666 0.0003224 ***
source:level  2  1178.1   589.1  2.7455 0.0731879 .
Residuals    54 11586.0   214.6
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1
```

Q: How many coefficients do we have in our regression model?

Q: Can you identify the mean for each categorical

```
> summary(diets.LM)
```

```
Call:
lm(formula = gain ~ source + level + source:level, data = diets.df)

Residuals:
   Min     1Q Median     3Q    Max
-29.90  -8.75   2.20  10.80  27.30

Coefficients:
                        Estimate Std. Error t value Pr(>t)
(Intercept)            1.000e+02  4.632e+00  21.589  < 2e-16 ***
sourceCereal          -1.410e+01  6.551e+00  -2.152  0.03585 *
sourcePork            -5.000e-01  6.551e+00  -0.076  0.93944
levelLow              -2.080e+01  6.551e+00  -3.175  0.00247 **
sourceCereal:levelLow  1.880e+01  9.264e+00   2.029  0.04736 *
sourcePork:levelLow    2.247e-15  9.264e+00   0.000  1.00000
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 14.65 on 54 degrees of freedom
Multiple R-squared:  0.2848,    Adjusted R-squared:  0.2185
F-statistic:   4.3 on 5 and 54 DF,  p-value: 0.002299
```

```
> summary(diets.LM)$coefficients
```

```
                         Estimate Std. Error       t value     Pr(>t)
(Intercept)           1.000000e+02   4.632014  2.158888e+01 3.143304e-28
sourceCereal         -1.410000e+01   6.550657 -2.152456e+00 3.584814e-02
sourcePork           -5.000000e-01   6.550657 -7.632822e-02 9.394401e-01
levelLow             -2.080000e+01   6.550657 -3.175254e+00 2.473734e-03
sourceCereal:levelLow  1.880000e+01   9.264028  2.029355e+00 4.736307e-02
sourcePork:levelLow   2.246933e-15   9.264028  2.425439e-16 1.000000e+00
```

- In this case, the mean for each categorial is given by

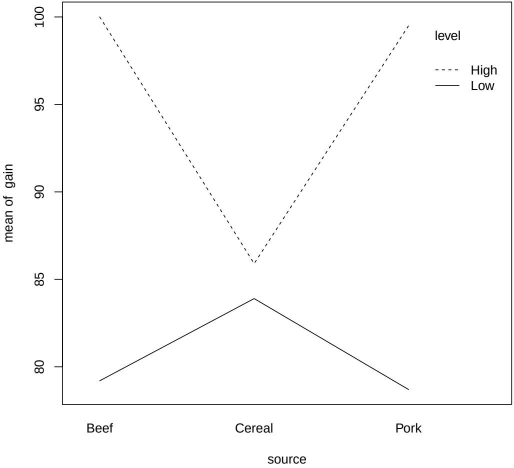
	Beef	Cereal	Pork
High	$\mu_{11} = 100$	$\mu_{12} = 85.9$	$\mu_{13} = 99.5$
Low	$\mu_{21} = 79.2$	$\mu_{22} = 83.9$	$\mu_{23} = 78.7$

- The interaction is defined to be the difference of column/row differences

$$\begin{aligned}(\mu_{ij} - \mu_{1j}) - (\mu_{i1} - \mu_{11}) &= (\mu_{ij} - \mu_{i1}) - (\mu_{1j} - \mu_{11}) \\ &= \mu_{ij} - \mu_{i1} - \mu_{1j} + \mu_{11}\end{aligned}$$

Q: Can you see the R output naturally decompose the mean μ_{ij} in general?

```
> # Interaction plot
> with(diets.df,
+       interaction.plot(source, level, gain))
```



```r
> head(cars.df)
```

```
                    Price    Country Reliability Mileage  Type Weight Disp.  HP
Eagle Summit 4       8895        USA           4      33 Small   2560    97 113
Ford Escort   4      7402        USA           2      33 Small   2345   114  90
Ford Festiva 4       6319      Korea           4      37 Small   1845    81  63
Honda Civic 4        6635  Japan/USA           5      32 Small   2260    91  92
Mazda Protege 4      6599      Japan           5      32 Small   2440   113 103
Mercury Tracer 4     8672     Mexico           4      26 Small   2285    97  82
```

`Mileage`	fuel consumption miles per US gallon
`Type`	a factor with 6 levels
`Disp.`	the engine capacity (displacement) in litres

```r
> summary(cars.df[,c("Mileage","Type","Disp.")])
```

```
    Mileage             Type         Disp.
 Min.   :18.00   Compact:15   Min.   : 73.0
 1st Qu.:21.00   Large  : 3   1st Qu.:113.8
 Median :23.00   Medium :13   Median :144.5
 Mean   :24.58   Small  :13   Mean   :152.1
 3rd Qu.:27.00   Sporty : 9   3rd Qu.:180.0
 Max.   :37.00   Van    : 7   Max.   :305.0
```

Q: What are the differences between the next 6 models?

```
> cars.type.LM = lm(Mileage~Type, data = cars.df)
>
> summary(cars.type.LM)
```

```
Call:
lm(formula = Mileage ~ Type, data = cars.df)

Residuals:
    Min      1Q  Median      3Q     Max
-7.0000 -1.1333  0.1868  1.2308  7.0000

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept)  24.1333     0.7128  33.856  < 2e-16 ***
TypeLarge    -3.8000     1.7460  -2.176 0.033921 *
TypeMedium   -2.3641     1.0461  -2.260 0.027886 *
TypeSmall     6.8667     1.0461   6.564  2.1e-08 ***
TypeSporty    1.8667     1.1640   1.604 0.114628
TypeVan      -5.2762     1.2637  -4.175 0.000109 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.761 on 54 degrees of freedom
Multiple R-squared:  0.6962,    Adjusted R-squared:  0.668
F-statistic: 24.75 on 5 and 54 DF,  p-value: 7.213e-13
```

```
> cars.base.df = cars.df
> cars.base.df$Type = relevel(
+   cars.base.df$Type, ref = "Sporty")
>
> cars.base.LM = lm(Mileage~Type, data=cars.base.df)
> summary(cars.base.LM)
```

```
Call:
lm(formula = Mileage ~ Type, data = cars.relevel.df)

Residuals:
    Min      1Q  Median      3Q     Max
-7.0000 -1.1333  0.1868  1.2308  7.0000

Coefficients:
            Estimate Std. Error t value Pr(>t)
(Intercept)  26.0000     0.9202  28.254  < 2e-16 ***
TypeCompact  -1.8667     1.1640  -1.604 0.114628
TypeLarge    -5.6667     1.8405  -3.079 0.003263 **
TypeMedium   -4.2308     1.1971  -3.534 0.000848 ***
TypeSmall     5.0000     1.1971   4.177 0.000109 ***
TypeVan      -7.1429     1.3913  -5.134 3.98e-06 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.761 on 54 degrees of freedom
Multiple R-squared:  0.6962,    Adjusted R-squared:  0.668
F-statistic: 24.75 on 5 and 54 DF,  p-value: 7.213e-13
```

```r
> cars.disp.LM = lm(Mileage~Disp., data = cars.df)
>
> summary(cars.disp.LM)
```

```

Call:
lm(formula = Mileage ~ Disp., data = cars.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.6477 -2.2328 -0.8693  2.9120  8.0595 

Coefficients:
             Estimate Std. Error t value Pr(>t)    
(Intercept) 33.907958   1.350146  25.114  < 2e-16 ***
Disp.       -0.061326   0.008373  -7.325 8.35e-10 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 3.483 on 58 degrees of freedom
Multiple R-squared:  0.4805,    Adjusted R-squared:  0.4716 
F-statistic: 53.65 on 1 and 58 DF,  p-value: 8.348e-10
```

```r
> cars.both.LM = lm(Mileage~Type+Disp.,
+                    data = cars.df)
>
> summary(cars.both.LM)
```

```
Call:
lm(formula = Mileage ~ Type + Disp., data = cars.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.4782 -1.1965 -0.0137  1.4351  5.2723 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 30.398504   1.291698  23.534  < 2e-16 ***
TypeLarge    2.399722   1.818184   1.320 0.192559    
TypeMedium  -0.782363   0.895757  -0.873 0.386380    
TypeSmall    4.943728   0.918185   5.384 1.69e-06 ***
TypeSporty   2.924745   0.962353   3.039 0.003680 ** 
TypeVan     -4.203946   1.041983  -4.035 0.000177 ***
Disp.       -0.044624   0.008231  -5.421 1.48e-06 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.235 on 53 degrees of freedom
Multiple R-squared:  0.8046,    Adjusted R-squared:  0.7824 
F-statistic: 36.36 on 6 and 53 DF,  p-value: < 2.2e-16
```

```
> cars.slope.LM = lm(Mileage~Type:Disp.,
+                         data = cars.df)
>
> summary(cars.slope.LM)
```

```
Call:
lm(formula = Mileage ~ Type:Disp., data = cars.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.5605 -1.4213 -0.1679  1.4201  5.8699 

Coefficients:
                   Estimate Std. Error t value Pr(>t)
(Intercept)       32.778022   1.532308  21.391  < 2e-16 ***
TypeCompact:Disp. -0.061388   0.011575  -5.304 2.26e-06 ***
TypeLarge:Disp.   -0.044720   0.007309  -6.119 1.17e-07 ***
TypeMedium:Disp.  -0.061306   0.009347  -6.559 2.31e-08 ***
TypeSmall:Disp.   -0.020345   0.016943  -1.201    0.235
TypeSporty:Disp.  -0.042508   0.008852  -4.802 1.33e-05 ***
TypeVan:Disp.     -0.083629   0.010665  -7.841 2.01e-10 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.398 on 53 degrees of freedom
Multiple R-squared:  0.775,     Adjusted R-squared:  0.7495
F-statistic: 30.42 on 6 and 53 DF,  p-value: 1.64e-15
```

```r
> cars.LM = lm(Mileage~Type*Disp., data = cars.df)
> summary(cars.LM)
```

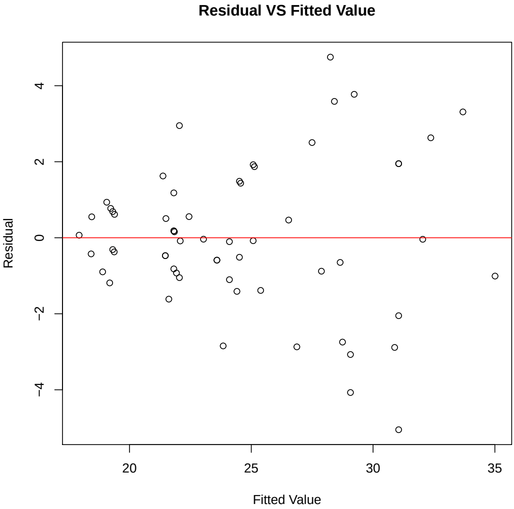
```
Call:
lm(formula = Mileage ~ Type * Disp., data = cars.df)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.0507 -0.9475 -0.0807  0.9965  4.7525 

Coefficients:
                  Estimate Std. Error t value Pr(>t)
(Intercept)      31.340651   4.064089   7.712 6.02e-10 ***
TypeLarge         4.623248  10.817859   0.427   0.6710
TypeMedium      -11.684691   6.017001  -1.942   0.0580 .
TypeSmall        15.704812   6.383658   2.460   0.0175 *
TypeSporty        2.547553   4.392143   0.580   0.5646
TypeVan          -8.435883   7.313291  -1.154   0.2544
Disp.            -0.051334   0.028685  -1.790   0.0798 .
TypeLarge:Disp.  -0.004623   0.045738  -0.101   0.9199
TypeMedium:Disp.  0.063352   0.038058   1.665   0.1025
TypeSmall:Disp.  -0.113560   0.057845  -1.963   0.0554 .
TypeSporty:Disp.  0.003268   0.030124   0.108   0.9141
TypeVan:Disp.     0.026718   0.046547   0.574   0.5686
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Residual standard error: 2.11 on 48 degrees of freedom
Multiple R-squared:  0.8422,    Adjusted R-squared:  0.8061
F-statistic: 23.29 on 11 and 48 DF,  p-value: 1.364e-15
```

- For the final model, heteroskedasticity seems to be the only problem.



Q: Assuming all assumptions hold, how well can we estimate the coefficients?

$$y_i = \beta_0 + \beta_1 x_i + e_i$$

- Recall the standard error of $\hat{\beta}_1$ is given by

$$\mathrm{Var}\left[\hat{\beta}_1 \mid x_1, x_2, \ldots, x_n\right] = \frac{\sigma^2}{(n-1)s_x^2}$$

$$\implies \mathrm{SE}\left(\hat{\beta}_1\right) = \frac{\sigma}{\sqrt{(n-1)s_x^2}} = \frac{\sigma}{\left(\sum_{i=1}^{n} (x_i - \bar{x})^2\right)^{1/2}}$$

from which we see the accuracy is partly determined by the extent of scatter about the true regression line, measured by σ, but also by

"configuration"

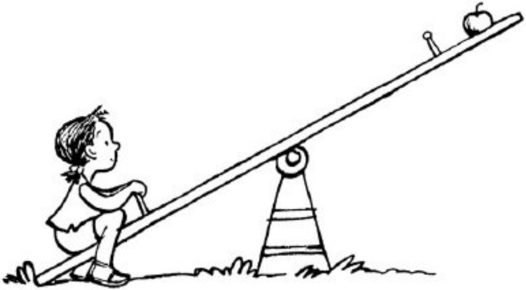
of observed x_i, that is, the spread of the observed x_i.

- If the x_i's are spread out, the estimated regression line is well

"supported"

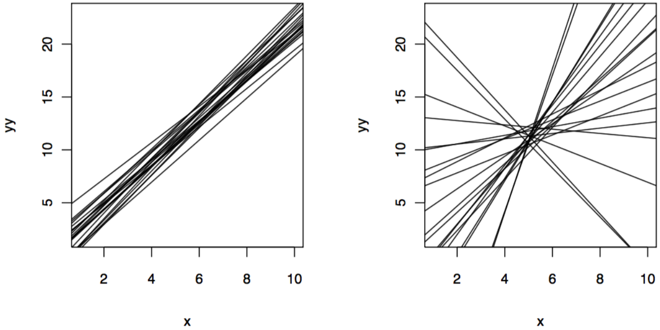
and it estimates the true line well.

- One the other hand, if the x_i's are bunched up, then it is not well supported,



very much like a seesaw.

- On the left, we have $\sum_{i=1}^{10} (x_i - \bar{x})^2 = 82.5$, 20 simulated sets of $\{y_i\}$,



while $\sum_{i=1}^{10} (x_i - \bar{x})^2 = 0.825$ on the right.

Q: What happens with two predictor variables x_{i1} and x_{i2}?

- Intuitively, the spread are still important in the stability of the regression,

$$\sum_{i=1}^{n} (x_{i1} - \bar{x}_1)^2 \qquad \text{and} \qquad \sum_{i=1}^{n} (x_{i2} - \bar{x}_2)^2$$

but covariance/correlation between x_{i1} and x_{i2} also matters.

- If there is strong linear relationship between X_{i1} and X_{i2}, we tends to have a

"knife edge"

support for the regression plane, which is also not stable.

- On the other hand, if the predictor variables are uncorrelated (or orthogonal), then they will be well spread out and support the fitted plane well.
- It can be shown the standard errors of the coefficients are proportional to

$$(1 - r^2)^{-1/2}$$

where r is the correlation coefficient between the two predictor variables.

- In terms of the matrix representation,

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1k} \\ 1 & x_{21} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nk} \end{bmatrix}$$

multicollinearity occurs when the columns of the matrix are almost

linearly dependent

- It can happen if

1. One or more of the predictor variables has/have very little variation.
2. One or more of the predictor variables has/have very large mean.
3. Two or more of the predictor variables have a linear relationship.

- We can think of multicollinearity caused by these two factors as inessential, since we can remove it by standardising the data

$$x_{ij}^* = \frac{x_{ij} - \bar{x}_j}{\left(\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2\right)^{1/2}}$$

or

$$x_{ij}^* = \frac{x_{ij} - \bar{x}_j}{\left(\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2\right)^{1/2}}$$

- The "Essential" multicollinearity is what remains after standardisation, and is caused by almost linear relationships between the predictor variables.

- Suppose we have standardised our data, and constructed the design matrix

$$\mathbf{X}$$

using the new data, and run the linear model on this matrix.

- Recall we derived the following

$$\operatorname{Var}\left[\hat{\boldsymbol{\beta}} \mid \mathbf{X}\right] = \sigma^2 \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$$

- We can calculate variance inflation factors (VIF), which are the diagonals of

$$\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$$

Q: How to determine which predictor variable is causing the multicollinearity?

$$\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{u} = \lambda\mathbf{u}$$