

## Discussion 12

### Review

#### 0.1 Simple Linear Regression

The simple linear regression model for the data is

$$Y_i = \alpha + \beta X_i + \varepsilon_i$$

where

1.  $\varepsilon_i$  is the random vertical deviation between the line and the  $i$ th observed data point.
2. the deviations are assumed to be independent and normally distributed with standard deviation  $\sigma$ .

#### About some useful estimators

- 

$$\hat{\beta} = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sum (X_i - \bar{X})^2} = r \frac{s_y}{s_x}$$

- 

$$\hat{\alpha} = \bar{Y} - \hat{\beta} \bar{X}$$

- 

$$RSS = \sum (Y_i - \hat{\alpha} - \hat{\beta} X_i)^2,$$

where  $\hat{Y}_i = \hat{\alpha} + \hat{\beta} X_i$  is the predicted value and

- $\hat{\varepsilon}_i = Y_i - \hat{Y}_i$  is the residual

#### 0.2 Linear Regression with one factor

ANOVA is an example of a linear model.

If the first group is used as reference in a one-way ANOVA with  $k$  groups,

$$Y_i = \beta_0 + \beta_2 I_{\{group=2\}} + \beta_3 I_{\{group=3\}} + \cdots + \beta_k I_{\{group=k\}} + \varepsilon_i,$$

where  $\varepsilon_i \sim \text{i.i.d. } N(0, 1)$ .

$\beta_0$  is the expected mean of the first group.

$\beta_2$  is the expected mean difference between the second and the first group.

...

$\beta_k$  is the expected mean difference between the  $k$ th group and the first group.

#### 0.3 Correlation Coefficient

The correlation coefficient  $r$  is a measure of the strength of the linear relationship between two variables.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

Correlation is not affected by linear transformation of the data.

## Practice Problems

1. A researcher is studying the effectiveness of three methods of reducing smoking. He wants to determine whether the mean reduction in the number of cigarettes smoked daily differs from one method to another among men patients. Each smoked about 60 cigarettes per day before treatment. Four randomly chosen members of the group pursue method I; four pursue method II ; and so on. The reductions in the number of cigarettes smoked daily are as follows:

Method		
I	II	III
10	19	11
9	20	13
9	21	15
8	20	13

- (a) Using `lm()` in R to find the estimated difference between the second method and the first method?
- (b) What is the estimated difference between the second method and the third method?
- (c) Construct a 95% confidence interval for the mean difference between the first and second method.
2. Read the dataset from <http://lib.stat.cmu.edu/DASL/Datafiles/airpollutionfiltersdat.html>. It contains 36 observations with 4 variables:

- NOISE = Noise level reading (decibels)
- SIZE = Vehicle size: 1 small 2 medium 3 large
- TYPE = 1 standard silencer 2 Octel filter
- SIDE = 1 right side 2 left side of car

- (a) Plot the noise level versus the vehicle size, type of the filter and filter location, in order to get a visual estimate of which one(s) seem to be most affecting the noise level.
- (b) How many replicates are there in each experimental condition (combination of SIZE, TYPE and SIDE)? is it balanced?
- (c) Fit a linear model to predict noise level using both SIZE and TYPE as predictors, without interaction. Then write the model with the estimated coefficients in two different but equivalent ways:

$$\text{mean noise level} = \mu + \alpha_2 \mathbf{1}_{\text{SIZE}=2} + \alpha_3 \mathbf{1}_{\text{SIZE}=3} + \beta_2 \mathbf{1}_{\text{TYPE}=2}$$

$$\text{mean noise level} = \tilde{\mu} + \tilde{\alpha}_1 \mathbf{1}_{\text{SIZE}=1} + \tilde{\alpha}_3 \mathbf{1}_{\text{SIZE}=3} + \tilde{\beta}_1 \mathbf{1}_{\text{TYPE}=1}$$

- (d) Complete the following table with the mean noise level predicted by the model in 3.

	SIZE=1	SIZE=2	SIZE=3
TYPE=1			
TYPE=2			

Then compare the values in the table with the mean noise level observed in the data.

3. Suppose  $\text{cor}(X, Y) = 0.2$ , calculate the following:

- (a)  $\text{cor}(3X, 5Y)$
- (b)  $\text{cor}(2X + 9, 6Y + 5)$

## Solution

```

1. > method1=c(10,9,9,8)
  > method2=c(19,20,21,20)
  > method3=c(11,13,15,13)
  > alldata=c(method1,method2,method3)
  > trt=c(rep(1,4), rep(2,4), rep(3,4))
  > trt=factor(trt); trt
  [1] 1 1 1 1 2 2 2 2 3 3 3 3
Levels: 1 2 3
  > library(lattice)
  > dotplot(alldata~trt)
  > fit1=lm(alldata~trt)
  > summary(fit1)

Call:
lm(formula = alldata ~ trt)

Residuals:
      Min       1Q   Median       3Q      Max
-2.000e+00 -2.500e-01 -1.333e-16  2.500e-01  2.000e+00

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.0000     0.5774  15.588 8.07e-08 ***
trt2          11.0000     0.8165  13.472 2.86e-07 ***
trt3           4.0000     0.8165   4.899 0.000849 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

Residual standard error: 1.155 on 9 degrees of freedom
Multiple R-squared:  0.9538,    Adjusted R-squared:  0.9436
F-statistic:    93 on 2 and 9 DF,  p-value: 9.748e-07

2. > noise = read.table("noise.txt",header=T)
  > str(noise)
'data.frame':  36 obs. of  4 variables:
 $ NOISE: int  810 820 820 840 840 845 785 790 785 835 ...
 $ SIZE : int  1 1 1 2 2 2 3 3 3 1 ...
 $ TYPE : int  1 1 1 1 1 1 1 1 1 1 ...
 $ SIDE : int  1 1 1 1 1 1 1 1 1 2 ...
  > # transform those categorical variables into factors.
  > noise$SIZE = factor(noise$SIZE)
  > noise$TYPE = factor(noise$TYPE)
  > noise$SIDE = factor(noise$SIDE)
  > #1
  > plot(NOISE~SIZE,noise)
  > plot(NOISE~TYPE,noise)
  > plot(NOISE~SIDE,noise)
  > #2

```

```

> xtabs(~SIZE+TYPE+SIDE, noise)
, , SIDE = 1

    TYPE
SIZE 1 2
  1 3 3
  2 3 3
  3 3 3

, , SIDE = 2

    TYPE
SIZE 1 2
  1 3 3
  2 3 3
  3 3 3

> ftable(xtabs(~SIZE+TYPE+SIDE, noise))
      SIDE 1 2
SIZE TYPE
  1   1     3 3
     2     3 3
  2   1     3 3
     2     3 3
  3   1     3 3
     2     3 3
> #3
> fit1 = lm(NOISE~SIZE+TYPE, noise)
> summary(fit1)

Call:
lm(formula = NOISE ~ SIZE + TYPE, data = noise)

Residuals:
    Min       1Q   Median       3Q      Max
-19.583  -7.292   1.250   6.250  15.833

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  829.583     3.099  267.657 < 2e-16 ***
SIZE2         9.583      3.796   2.525 0.01674 *
SIZE3        -51.667     3.796 -13.611 7.4e-15 ***
TYPE2        -10.833     3.099  -3.495 0.00141 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.298 on 32 degrees of freedom
Multiple R-squared:  0.9074,    Adjusted R-squared:  0.8987
F-statistic: 104.5 on 3 and 32 DF,  p-value: < 2.2e-16

```

```
> # change the reference levels to obtain another formula for the same model.
> noise_m = noise
> noise_m$SIZE = relevel(noise_m$SIZE,"2")
> noise_m$TYPE = relevel(noise_m$TYPE,"2")
> fit1_m = lm(NOISE~SIZE+TYPE, noise_m)
> summary(fit1_m)
```

Call:

```
lm(formula = NOISE ~ SIZE + TYPE, data = noise_m)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-19.583  -7.292   1.250   6.250  15.833
```

Coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  828.333      3.099 267.253 < 2e-16 ***
SIZE1        -9.583      3.796  -2.525 0.01674 *
SIZE3       -61.250      3.796 -16.135 < 2e-16 ***
TYPE1        10.833      3.099   3.495 0.00141 **
```

---

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 9.298 on 32 degrees of freedom

Multiple R-squared: 0.9074, Adjusted R-squared: 0.8987

F-statistic: 104.5 on 3 and 32 DF, p-value: < 2.2e-16

```
> newC = data.frame(
+   SIZE = rep( c("1","2","3"), 2),
+   TYPE = rep( c("1", "2"), each=3)
+ )
```

```
> newC
  SIZE TYPE
```

```
1   1   1
2   2   1
3   3   1
4   1   2
5   2   2
6   3   2
```

```
> predict(fit1, newC)
```

```
      1      2      3      4      5      6
829.5833 839.1667 777.9167 818.7500 828.3333 767.0833
```

> # verify that different formulation doesnt change the predicted values.

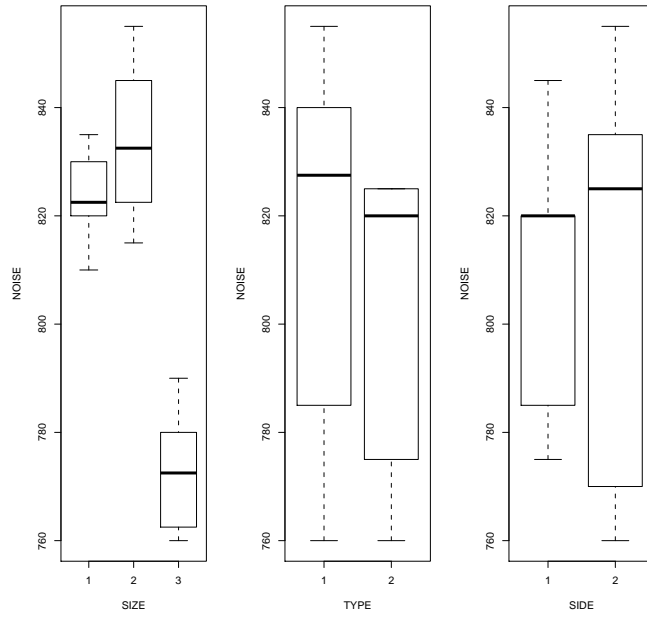
```
> predict(fit1_m, newC)
```

```
      1      2      3      4      5      6
829.5833 839.1667 777.9167 818.7500 828.3333 767.0833
```

> 5# compute the group means of the noise level.

```
> with(noise, tapply(NOISE, list(SIZE,TYPE), mean))
```

```
      1      2
1 825.8333 822.5000
2 845.8333 821.6667
```



3 775.0000 770.0000