Rexample

Lingxiao Zhou

EX (Drug)

- Response variable Y
- Dosage level (d)
- Drug product with levels: A, B, C
  - reference level: A
  - Dummy variables: P_B, P_c

Create the dataset

```
Dose = rep(c(0.2, 0.4, 0.8, 1.6), each = 3)
Product = rep(c("A", "B", "C"), 4)
y = c(2, 1.8, 1.3, 4.3, 4.1, 2, 6.5, 4.9, 2.8, 8.9, 5.7, 3.4)
ds = data.frame(Dose = (Dose), Product = factor(Product), Response = y)
xtabs(Response ~ Dose + Product, data = ds)
```

```
##      Product
## Dose    A   B   C
##   0.2 2.0 1.8 1.3
##   0.4 4.3 4.1 2.0
##   0.8 6.5 4.9 2.8
##   1.6 8.9 5.7 3.4
```

```
ds
```

```
##    Dose Product Response
## 1   0.2       A      2.0
## 2   0.2       B      1.8
## 3   0.2       C      1.3
## 4   0.4       A      4.3
## 5   0.4       B      4.1
## 6   0.4       C      2.0
## 7   0.8       A      6.5
## 8   0.8       B      4.9
## 9   0.8       C      2.8
## 10  1.6       A      8.9
## 11  1.6       B      5.7
## 12  1.6       C      3.4
```

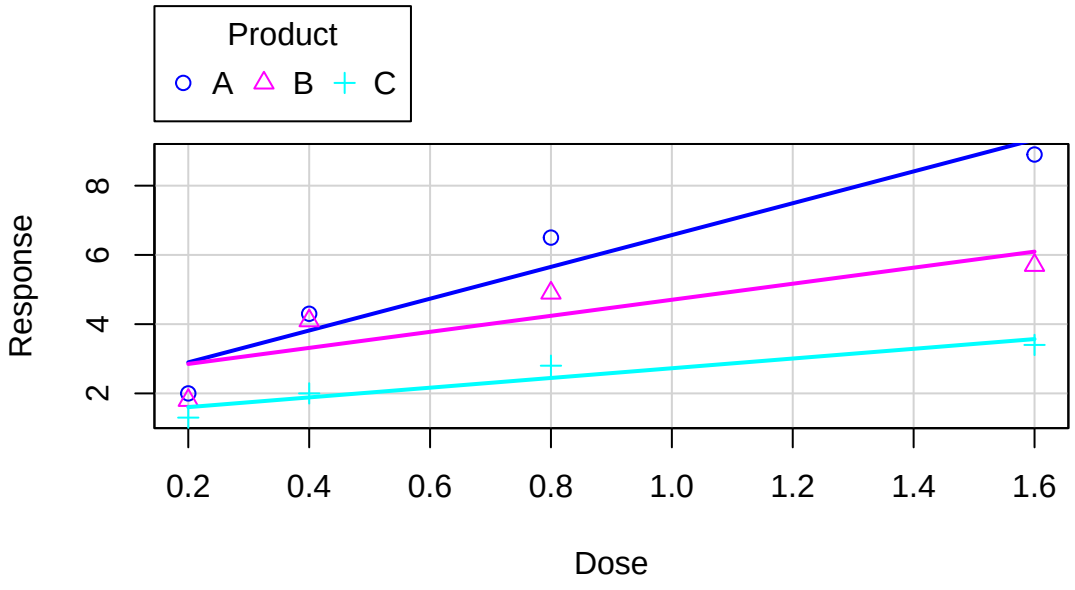
Fit multiple regression with interaction

$Y = \beta_0 + \beta_1 d + \beta_2 P_B + \beta_3 P_C + \beta_4 d P_B + \beta_5 d P_C + \epsilon$

```
modelfull = lm(Response ~ Dose * Product, data = ds)
summary(modelfull)
```

```
##
## Call:
## lm(formula = Response ~ Dose * Product, data = ds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.05043 -0.40391 -0.02609  0.52739  0.84522
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.9783     0.7454   2.654  0.03783 *
## Dose            4.5957     0.8085   5.684  0.00128 **
## ProductB        0.4087     1.0542   0.388  0.71163
## ProductC       -0.6565     1.0542  -0.623  0.55635
## Dose:ProductB  -2.2783     1.1434  -1.992  0.09339 .
## Dose:ProductC  -3.1913     1.1434  -2.791  0.03154 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.867 on 6 degrees of freedom
## Multiple R-squared:  0.9194, Adjusted R-squared:  0.8523
## F-statistic: 13.69 on 5 and 6 DF,  p-value: 0.003126
```

```
library(car)
scatterplot(Response ~ Dose | Product, smooth = FALSE, data = ds)
```

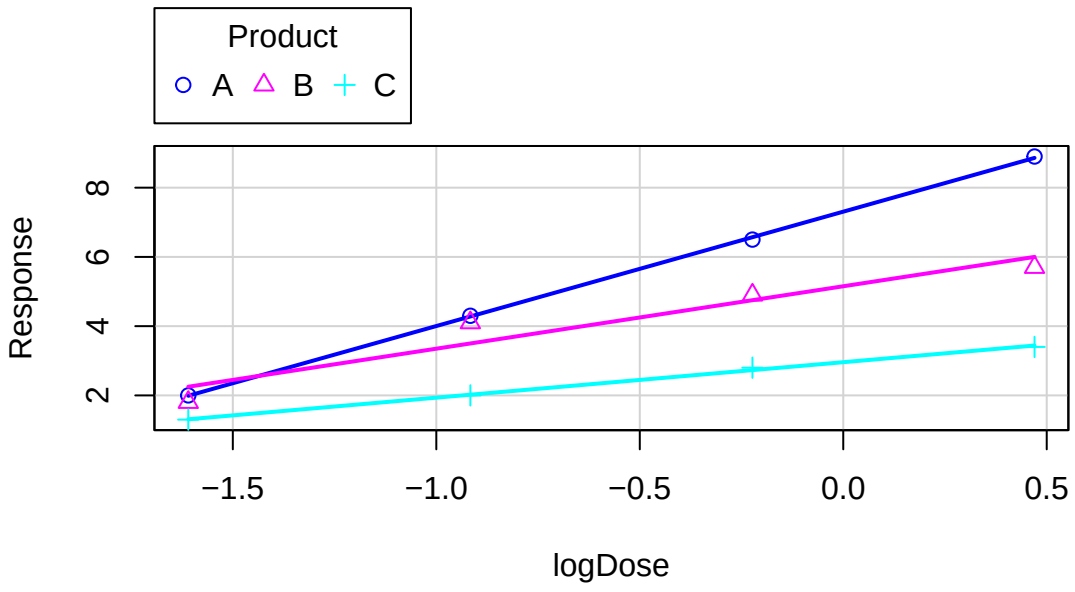


Notice that a log transformation on Dose may help fit a more linear relationship

```
ds = data.frame(logDose = log(Dose), Product = factor(Product),
    Response = y)
# Full model
modelfull = lm(Response ~ logDose * Product, data = ds)
summary(modelfull)
```

```
##
## Call:
## lm(formula = Response ~ logDose * Product, data = ds)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4500 -0.0475  0.0000  0.0475  0.6000
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         7.3072     0.2103  34.748 3.79e-08 ***
## logDose             3.3038     0.2186  15.111 5.30e-06 ***
## ProductB           -2.1548     0.2974  -7.245 0.000351 ***
## ProductC           -4.3486     0.2974 -14.622 6.42e-06 ***
## logDose:ProductB   -1.5004     0.3092  -4.853 0.002844 **
## logDose:ProductC   -2.2795     0.3092  -7.372 0.000319 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3389 on 6 degrees of freedom
## Multiple R-squared:  0.9877, Adjusted R-squared:  0.9774
## F-statistic:  96.3 on 5 and 6 DF,  p-value: 1.207e-05
```

```
scatterplot(Response ~ logDose | Product, smooth = FALSE, data = ds)
```



CI for coefficient

Create at least 90% simultaneous CI for β_4, β_5 and $\beta_4 - \beta_5$ using **Bonferroni method**

```
coefficients(modelfull)
```

```
##      (Intercept)          logDose         ProductB         ProductC
##         7.307215         3.303772        -2.154805        -4.348646
## logDose:ProductB logDose:ProductC
##        -1.500403        -2.279458
```

- $b_4 = -1.50$
- $b_5 = -2.28$
- $b_4 - b_5 = -3.78$
- Critical value $t_{1-0.1/(2\cdot 3),n-p} = t_{1-0.1/6,6}$

```
vmat = vcov(modelfull)
round(vmat, 3)
```

```
##                  (Intercept) logDose ProductB ProductC logDose:ProductB
## (Intercept)            0.044   0.027   -0.044   -0.044           -0.027
## logDose                0.027   0.048   -0.027   -0.027           -0.048
## ProductB              -0.044  -0.027    0.088    0.044            0.054
## ProductC              -0.044  -0.027    0.044    0.088            0.027
## logDose:ProductB      -0.027  -0.048    0.054    0.027            0.096
## logDose:ProductC      -0.027  -0.048    0.027    0.054            0.048
##                  logDose:ProductC
## (Intercept)              -0.027
## logDose                  -0.048
## ProductB                  0.027
## ProductC                  0.054
## logDose:ProductB          0.048
## logDose:ProductC          0.096
```

- $S_{b_4-b_5} = \sqrt{S^2_{b_4} + S^2_{b_5} - 2S_{b_4 b_5}}$
- $S^2_{b_4}$ is the (5,5)th element in the matrix
- $S^2_{b_5}$ is the (6,6)th element in the matrix
- $S_{b_4 b_5}$ is the (5,6)th element in the matrix

CI for beta4

```
coefficients(modelfull)[5] + c(-1, 1) * qt(1 - 0.1/6, 6) * sqrt(vmat[5,
    5])
```

```
## [1] -2.3504094 -0.6503963
```

CI for beta5

```r
coefficients(modelfull)[6] + c(-1, 1) * qt(1 - 0.1/6, 6) * sqrt(vmat[6,
    6])
```

```
## [1] -3.129465 -1.429452
```

CI for beta4-beta5

```r
coefficients(modelfull)[5] - coefficients(modelfull)[6] + c(-1,
    1) * qt(1 - 0.1/6, 6) * sqrt(vmat[5, 5] + vmat[6, 6] - 2 *
    vmat[5, 6])
```

```
## [1] -0.07095119  1.62906183
```