Part I

More than one independent variable

Fully factorial designs

In a fully factorial design you are interested in the effect of two or more factors on the dependent variable.
Every level of each factor is run with every level of the others. These combinations may or may not be replicated
As the numbers of factors increases, these experiments become unwieldy. When $f = 5$ running a fully factorial experiment is difficult to implement.

Interactions

Factorial designs allow the estimation of effects due to the main factors as well as *interactions* among those factors.

- Interactions occur when the response variable for a level in factor 'A' changes with changes of the level in factor 'B'. This is known as
  - A X B interaction
  - A *by* B interaction

Guinea pig tooth growth

The `ToothGrowth` dataframe is available in R base. It outlines the growth of guinea pig teeth under various vitamin C dosages administered in 2 ways. This is a balanced design with 10 replicates per combination of treatment levels.

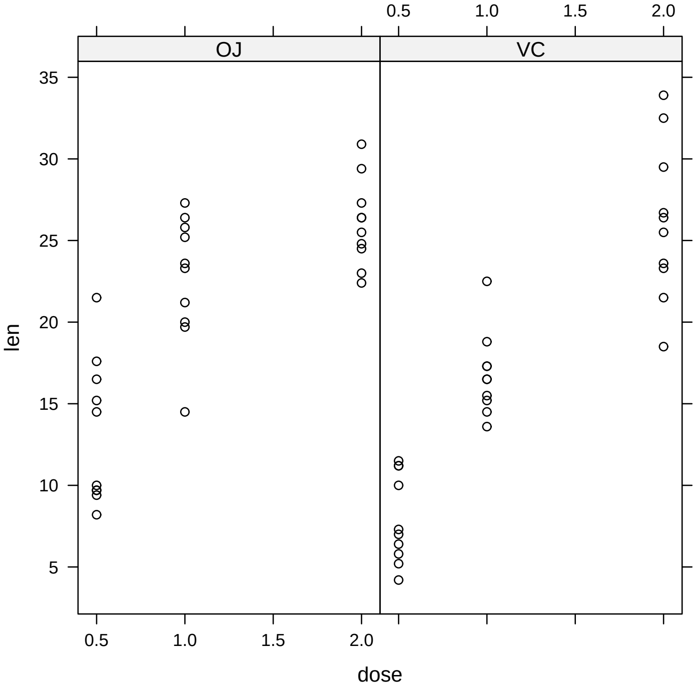
Teeth, cont.

```
> data(ToothGrowth)
> names(ToothGrowth)

[1] "len"  "supp" "dose"

> ToothGrowth[1:12, ]

    len supp dose
1   4.2   VC  0.5
2  11.5   VC  0.5
3   7.3   VC  0.5
```

```
4   5.8   VC  0.5
5   6.4   VC  0.5
6  10.0   VC  0.5
7  11.2   VC  0.5
8  11.2   VC  0.5
9   5.2   VC  0.5
10  7.0   VC  0.5
11 16.5   VC  1.0
12 16.5   VC  1.0
```

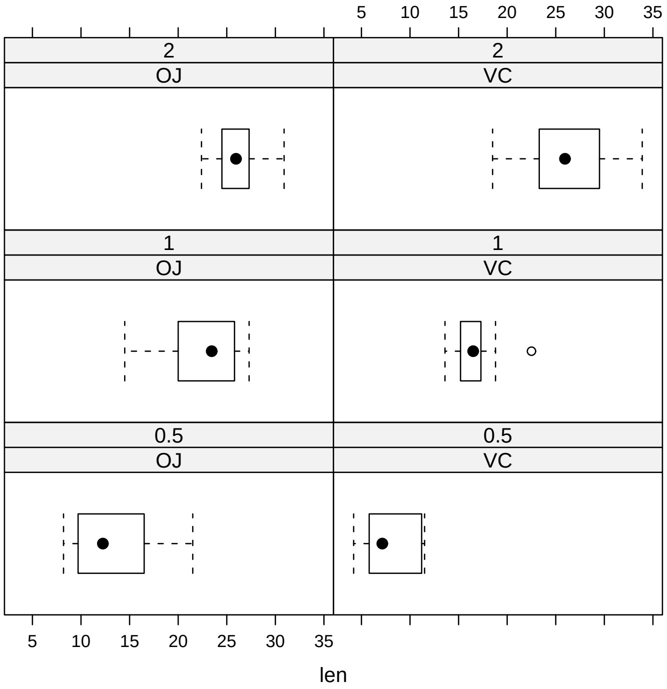
Teeth, cont.

```
> library(lattice)
> print(xyplot(len ~ dose | supp, data = ToothGrowth))
```



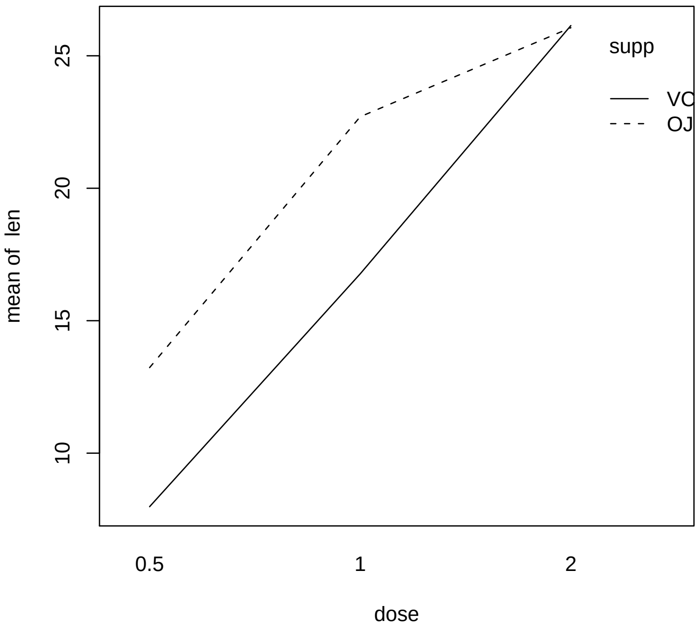
Teeth, cont.

```
> print(bwplot(~len | as.factor(supp) + as.factor(dose), data = ToothGrowth))
```



Teeth, “official” interaction plot

```
> attach(ToothGrowth)
> interaction.plot(dose, supp, len)
> detach(ToothGrowth)
```



Linear Model 2-way

$$Y_{ilj} = \mu + \tau_i + \beta_l + \gamma_{li} + \epsilon_{lij} \tag{1}$$

- The τ_i term represents the deviance of the first factor
- The β_l term represents the second factor deviance
- The γ_{li} term represents the interaction deviance

2-way anova (main and int)

```
> ToothGrowth$supp <- as.factor(ToothGrowth$supp)
> ToothGrowth$dose <- as.factor(ToothGrowth$dose)
> tooth.main <- aov(len ~ supp + dose, data = ToothGrowth)
> summary(tooth.main)
```

```
            Df  Sum Sq Mean Sq F value    Pr(>F)
supp         1  205.35  205.35  14.017 0.0004293 ***
dose         2 2426.43 1213.22  82.811 < 2.2e-16 ***
Residuals   56  820.43   14.65
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> tooth.int <- aov(len ~ supp + dose + supp * dose, data = ToothGrowth)
> summary(tooth.int)

            Df  Sum Sq Mean Sq F value    Pr(>F)    
supp         1  205.35  205.35  15.572 0.0002312 ***
dose         2 2426.43 1213.22  92.000 < 2.2e-16 ***
supp:dose    2  108.32   54.16   4.107 0.0218603 *  
Residuals   54  712.11   13.19                      
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

Some things to notice

- d.f.
  - $SS_{\text{total}} = N - 1$
  - $SS_{\text{supp}} = k_{\text{supp}} - 1 = 1$
  - $SS_{\text{dose}} = k_{\text{dose}} - 1 = 2$
  - $SS_{\text{supp:dose}} = (k_{\text{supp}} - 1) \times (k_{\text{dose}} - 1) = 2$
  - $SS_{\text{resid}} = k_{\text{supp}} \times k_{\text{dose}} \times (n - 1)$ (only true for balanced designs)

Additivity

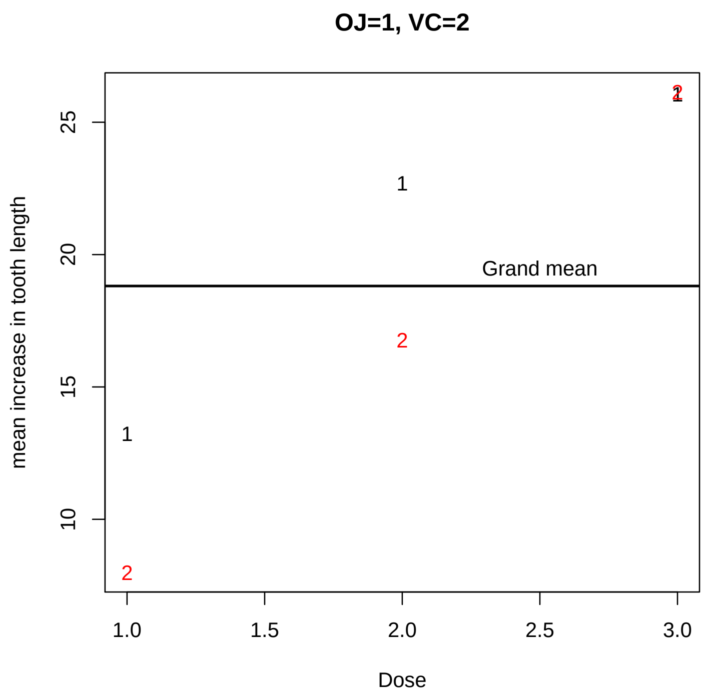
- Additivity for SS

$$SS_{\text{total}} = SS_{\text{supp}} + SS_{\text{dose}} + SS_{\text{supp:dose}} + SS_{\text{resid}}$$

More detail

- means:

  ```
  > tapply(ToothGrowth$len, list(ToothGrowth$dose, ToothGrowth$supp),
  +     mean)

         OJ    VC
  0.5 13.23  7.98
  1   22.70 16.77
  2   26.06 26.14
  ```



more More detail

- Testing components of the model

```
> lm(len ~ supp * as.factor(dose), data = ToothGrowth)

Call:
lm(formula = len ~ supp * as.factor(dose), data = ToothGrowth)

Coefficients:
             (Intercept)                    suppVC        as.factor(dose)1
                   13.23                     -5.25                    9.47
        as.factor(dose)2  suppVC:as.factor(dose)1  suppVC:as.factor(dose)2
                   12.83                     -0.68                    5.33
```

more more More detail

```
> summary(lm(len ~ supp * as.factor(dose), data = ToothGrowth))

Call:
lm(formula = len ~ supp * as.factor(dose), data = ToothGrowth)

Residuals:
   Min     1Q Median     3Q    Max
 -8.20  -2.72  -0.27   2.65   8.27
```

```
Coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept)               13.230      1.148  11.521 3.60e-16 ***
suppVC                    -5.250      1.624  -3.233  0.00209 ** 
as.factor(dose)1           9.470      1.624   5.831 3.18e-07 ***
as.factor(dose)2          12.830      1.624   7.900 1.43e-10 ***
suppVC:as.factor(dose)1   -0.680      2.297  -0.296  0.76831    
suppVC:as.factor(dose)2    5.330      2.297   2.321  0.02411 *  
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1 

Residual standard error: 3.631 on 54 degrees of freedom
Multiple R-Squared: 0.7937,	Adjusted R-squared: 0.7746 
F-statistic: 41.56 on 5 and 54 DF,  p-value: < 2.2e-16 
```

Shrimp data

A multiway shrimp culture experiment was conducted. The design was full factorial with three factors. Three replicate aquaria were set up for each combination of factor levels. The factors were:

watch for pseudoreplication

- Temperature (25C, 35C)
- Density (80 or 160 per 40 liter tank)
- Salinity (10, 25, or 40 ppt)

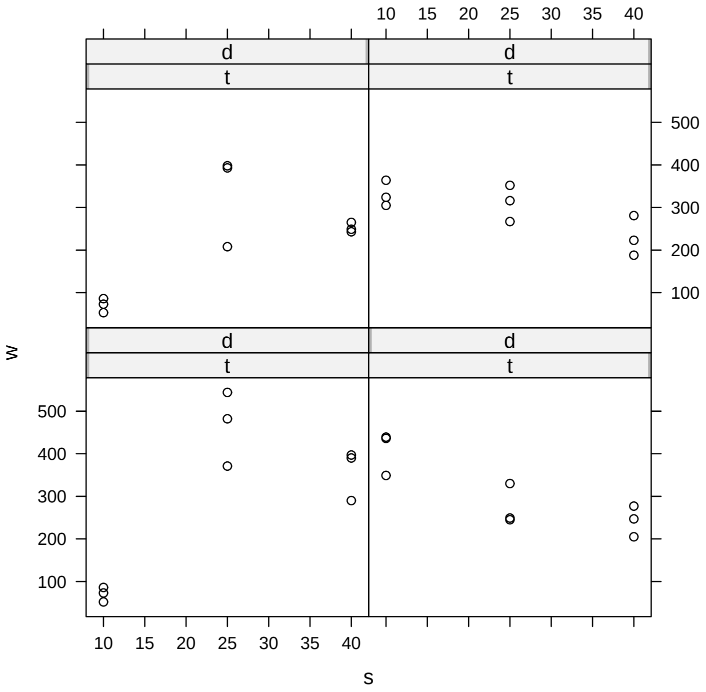
These data are in 'shrimp.csv'

Shrimp data cont

```
> shrimp <- read.csv("shrimp.csv")
> shrimp[1:10, ]

    t   d  s   w
1  25  80 10  86
2  25  80 25 544
3  25  80 40 390
4  25 160 10  53
5  25 160 25 393
6  25 160 40 249
7  35  80 10 439
8  35  80 25 249
9  35  80 40 247
10 35 160 10 324
```

Shrimp data cont

```
> library(lattice)
> print(xyplot(w ~ s | t + d, data = shrimp))
```



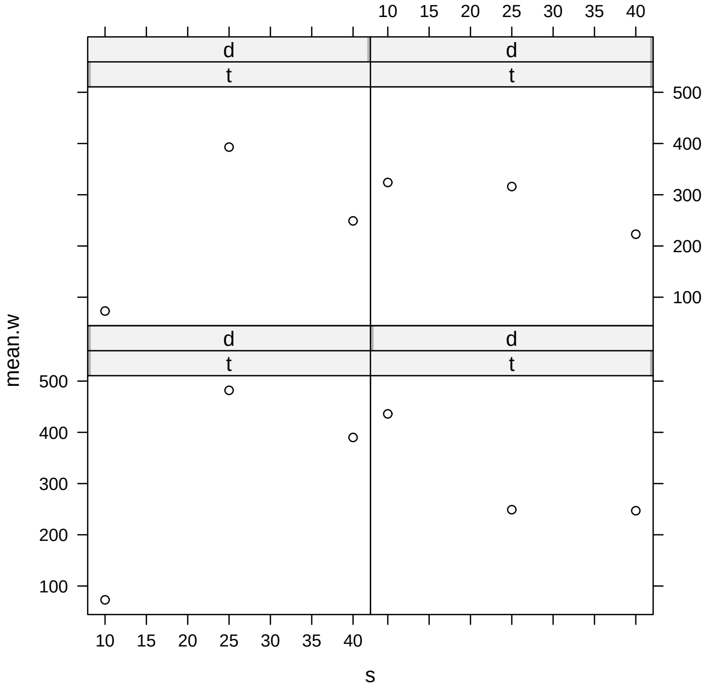
Shrimp data cont

```
> shrimp.mean <- aggregate(shrimp$w, list(t = shrimp$t, d = shrimp$d,
+     s = shrimp$s), median)
> names(shrimp.mean)[4] <- "mean.w"
> shrimp.mean[1:10, ]
```

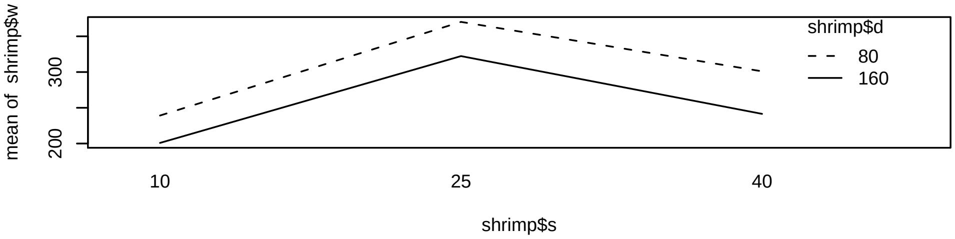
```
    t   d  s mean.w
1  25  80 10     73
2  35  80 10    436
3  25 160 10     73
4  35 160 10    324
5  25  80 25    482
6  35  80 25    249
7  25 160 25    393
8  35 160 25    316
9  25  80 40    390
10 35  80 40    247
```

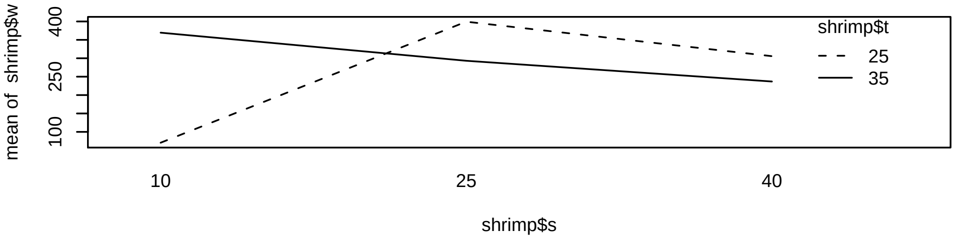
Shrimp data cont

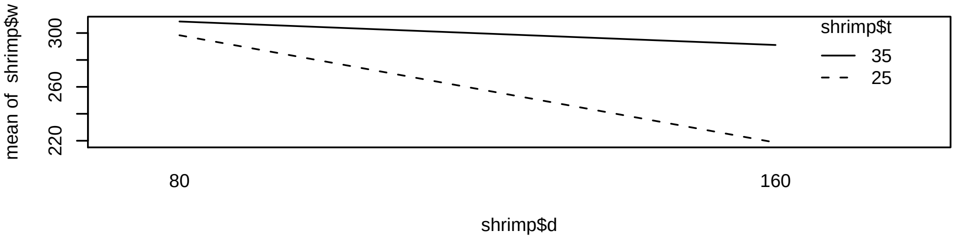
```
> shrimp.mean$s <- as.numeric(as.character(shrimp.mean$s))
> print(xyplot(mean.w ~ s | t + d, data = shrimp.mean))
```



Shrimp data interaction plot 1







Linear Model 3-way

$$Y_{lhij} = \mu + \tau_i + \beta_l + \alpha_h + \gamma_{li} + \lambda_{ih} + \beta_{lh} + \eta_{ilh} + \epsilon_{lhij} \tag{2}$$

- The τ_i term represents the effect of the first factor
- The β_l term represents the effect of the second factor
- The α_l term represents the effect of the third factor
- The $\gamma_{li}, \beta_{lh}, \lambda_{ih}$ terms represent the interaction deviances among pairs of terms.
- The η_{ilh} term represents the three-way interaction

Shrimp Anova

```
> shrimp$d <- as.factor(shrimp$d)
> shrimp$t <- as.factor(shrimp$t)
> shrimp$s <- as.factor(shrimp$s)
> shrimp2way <- aov(w ~ t + d + s + t:s + t:d + s:d, data = shrimp)
> summary(shrimp2way)

            Df Sum Sq Mean Sq F value    Pr(>F)
t            1  15376   15376  4.2652   0.04900 *
d            1  21219   21219  5.8860   0.02250 *
s            2  96763   48381 13.4207 9.908e-05 ***
t:s          2 300855  150428 41.7279 7.666e-09 ***
t:d          1   8711    8711  2.4164   0.13216
d:s          2    674     337  0.0935   0.91101
Residuals   26  93729    3605
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

Shrimp Anova-threeway interaction

```
> shrimp3way <- aov(w ~ t + d + s + t:s + t:d + s:d + s:d:t, data = shrimp)
> summary(shrimp3way)

            Df Sum Sq Mean Sq F value    Pr(>F)
t            1  15376   15376  5.2952   0.03038 *
d            1  21219   21219  7.3073   0.01242 *
s            2  96763   48381 16.6615 2.901e-05 ***
t:s          2 300855  150428 51.8041 1.959e-09 ***
t:d          1   8711    8711  2.9999   0.09610 .
d:s          2    674     337  0.1161   0.89086
t:d:s        2  24038   12019  4.1392   0.02855 *
Residuals   24  69691    2904
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

Shrimp Anova another parameterization

```
> shrimp3way <- aov(w ~ d * s * t, data = shrimp)
> summary(shrimp3way)

            Df Sum Sq Mean Sq F value    Pr(>F)
d            1  21219   21219  7.3073   0.01242 *
s            2  96763   48381 16.6615 2.901e-05 ***
```

```
t            1  15376   15376  5.2952   0.03038 *
d:s          2    674     337  0.1161   0.89086
d:t          1   8711    8711  2.9999   0.09610 .
s:t          2 300855  150428 51.8041 1.959e-09 ***
d:s:t        2  24038   12019  4.1392   0.02855 *
Residuals   24  69691    2904
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```