Sampling and Experimental Design

Contents

1 Sampling (under Simple Ramdom Sampling without Replacement)

1.1 Estimate of Interest

The whole project will base on the method of simple random sampling without replacement (SRSWOR). And the estimate of interest of our sampling estimatation will be the mean of

the responses (y_i) along with its distribution.

Here are some useful notations and proofs:

- μ and σ^2 : the true population average and population variance.
- $\tilde{\mu}$ and $\tilde{\sigma}^2$: the estimator of the population average and population variance.

Under SRSWOR, we have:

$$E(\tilde{\mu}) = \mu$$
$$Var(\tilde{\mu}) = (1 - \frac{n}{N})\frac{\sigma^2}{n}$$

where n is the size of the sample (S) and N is the size of the population.

Proof:

Define indicator variable as:

$$I_i = I(i \in S) = \begin{cases} 1 & i \in S \\ 0 & i \notin S \end{cases}$$

for i = 1,. . . ,N. This change avoid us using sample S to be the random variable, and the random variable now is the I_i

So that the sample size $n = \sum_{i=1}^{N} I_i$ and then $\tilde{\mu} = \frac{1}{n}\sum_{i \in S} y_i = \frac{1}{n}\sum_{i=1}^{N} I_i y_i$

Knowing the $E(I_i) = P(I_i = 1) = \frac{n}{N}$.

Therefore,

$$E(\tilde{\mu}) = E(\frac{1}{n}\sum_{i=1}^{N} I_i y_i) = \frac{1}{n}\sum_{i=1}^{N}\frac{n}{N} y_i = \frac{1}{N}\sum_{i=1}^{N} y_i = \mu$$

$$Var(\tilde{\mu}) = Var(\frac{1}{n}\sum_{i=1}^{N} I_i y_i) = \frac{1}{n^2}[\sum_{i=1}^{N} y_i^2 Var(I_i) + \sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j Cov(I_i, I_j)]$$

For $Var(I_i)$, we have $Var(I_i) = E[I_i^2] - [E(I_i)]^2 = \frac{n}{N} - (\frac{n}{N})^2$

For $Cov(I_i, I_j)$, we know that $Cov(I_i, I_j) = E[I_i I_j] - E[I_i]E[I_j] = \frac{n(n-1)}{N(N-1)} - (\frac{n}{N})^2$

Put them together with some algebra, we get $Var(\tilde{\mu}) = (1 - \frac{n}{N})\frac{\sigma^2}{n}$

1.2 Sample Size Determination

Sometimes, question also rise from finding the smallest sample size that can achieve our goal. Suppose that we want to estimate the mean with a confidence interval of length 2L (which means that $\hat{\mu} \pm L$). We know that under the SRSWOR,

$$L = c \times se(\hat{\mu}) = c \times \sqrt{(1 - \frac{n}{N})\frac{\hat{\sigma}^2}{n}}$$

where c is the critical value.

From which we can find that the required sample size is

$$n = (\frac{L^2}{c^2\hat{\sigma}^2} + \frac{1}{N})^{-1}$$

This tells us that if we know the population size and the population variance, then the size of the sample will be settled.

1.3 Regression Estimation

1.3.1 Theory

With the assumption of relationship between X and Y is linear with constant variation of errors. We can fit a linear model as:

$$Y_i = \alpha + \beta(X_i - \bar{X}) + R_i$$

where R_i is the residuals or the noise.

Also recall that the $\hat{\mu}_x = \bar{x}$ and under least square estimation we estimate $\hat{\alpha}$ and $\hat{\beta}$ as $\alpha = \bar{y} = \hat{\mu}_y$ and $\hat{\beta} = \frac{\sum_{i \in S}(x_i - \bar{x})y_i}{\sum_{i \in S}(x_i - \bar{x})^2}$.

Combine everything together, we have:

$$\hat{\mu}_{reg} = \hat{\mu}_y + \hat{\beta}(\mu_x - \hat{\mu}_x)$$

As long as we know the true mean value of the response X who is associated with the estimate of interest Y, then we can use it to make a more precise estimation by fitting a linear regression between them.

1.3.2 Facebook Example

In this example, we are interested in the average number of Facebook friends of UWaterloo employees among the people aged 25 and above. Let's assume there must be some relationship

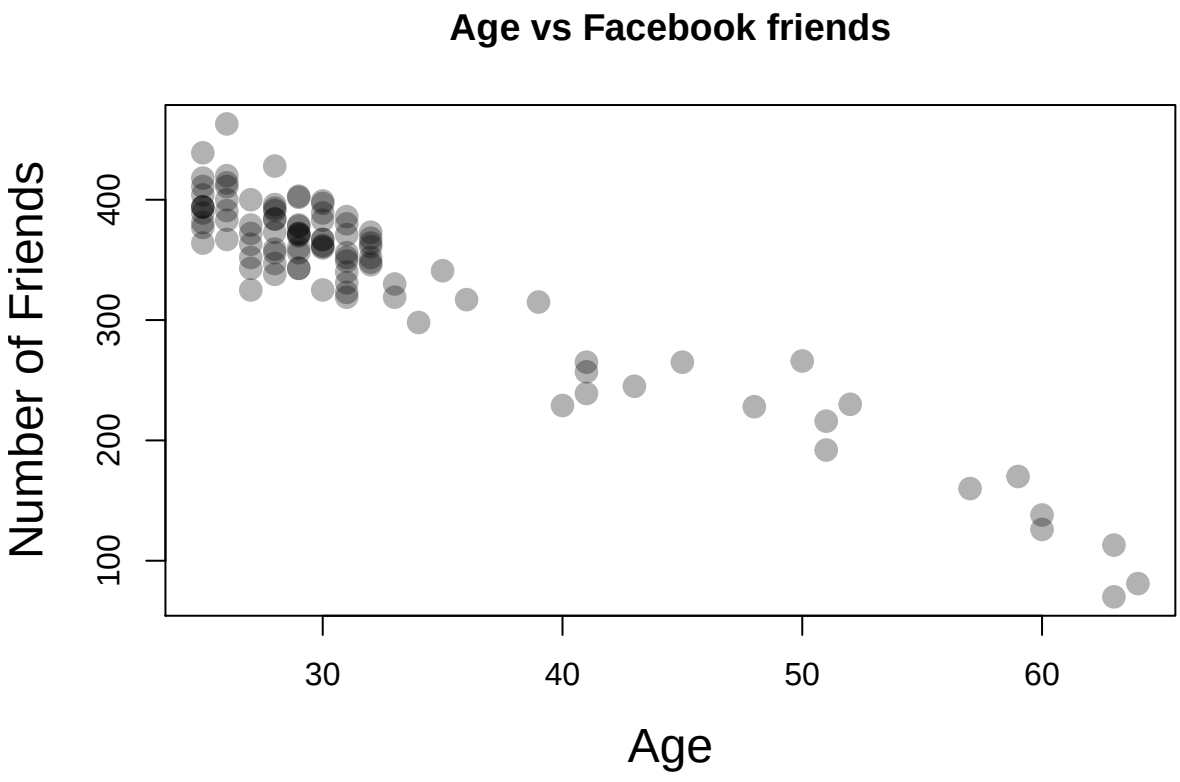
between a person's number of Facebook friends and her age. And using the data provided by Department of Statistics and Actuarial Science, this is a sample of size 100 getting using the SRSWOR from the whole population of UWaterloo employees.

Let's first get a sense of the data by looking the first a few rows and by plotting it if possible.

```
FB <- read.csv("Facebook.csv")
head(FB)
```

```
##   age friends
## 1  27     372
## 2  27     363
## 3  26     391
## 4  28     356
## 5  31     331
## 6  32     346
```

```
age <- FB$age
friends <- FB$friends
plot(age, friends, pch=19, col=adjustcolor("black",.3), cex.lab=1.5, cex=1.5, xlab="Age"
```



Knowing that the sample size n is 100, population size N = 2700, and the true average age μ_x is 44.464, which is easy to get.

```
n <- 100
N <- 2700
mu.age.true <- 44.464
```

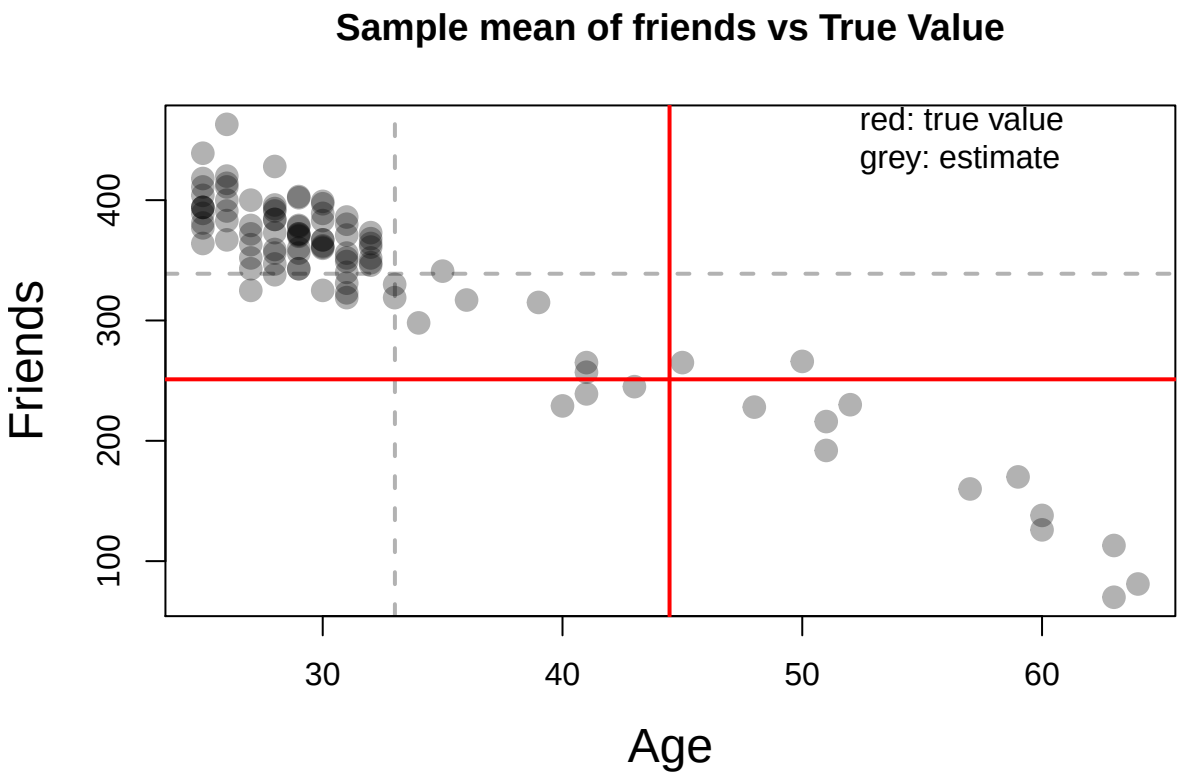
1.3.2.1 Performance of simple sample estimation

- It is provided that the true average number of friends is around 248.1 through a survey that we can treat as the census and now let's frist see the performance of sample estimates!

```
sample.friends <- mean(friends)
```

- The sample average for the number of friends is 338.9
- Our sample estimated mean was much too high, reflecting our sample biasedly picked younger people (with more Facebook friends), showing in the plot below:

```
plot(age, friends, pch=19, col=adjustcolor("black",.3), cex.lab=1.5, cex=1.5, xlab="Age'

abline(v= mean(age), lwd=2, col=adjustcolor("black",.3),lty=c(2))

abline( h=mean(friends), lwd=2, col=adjustcolor("black",.3),lty=2)

abline(v = c( mu.age.true),lwd = 2, col = 'red',lty =c(1))

abline(h = c(251.1),lwd = 2, col = 'red',lty =c(1))

legend(50,500,c('red: true value', 'grey: estimate'),bty='n')
```



where red lines are the true average number of friends and age, and the grey lines are the sample estimated average number of friends and age.

1.3.2.2 Performance of regression estimation

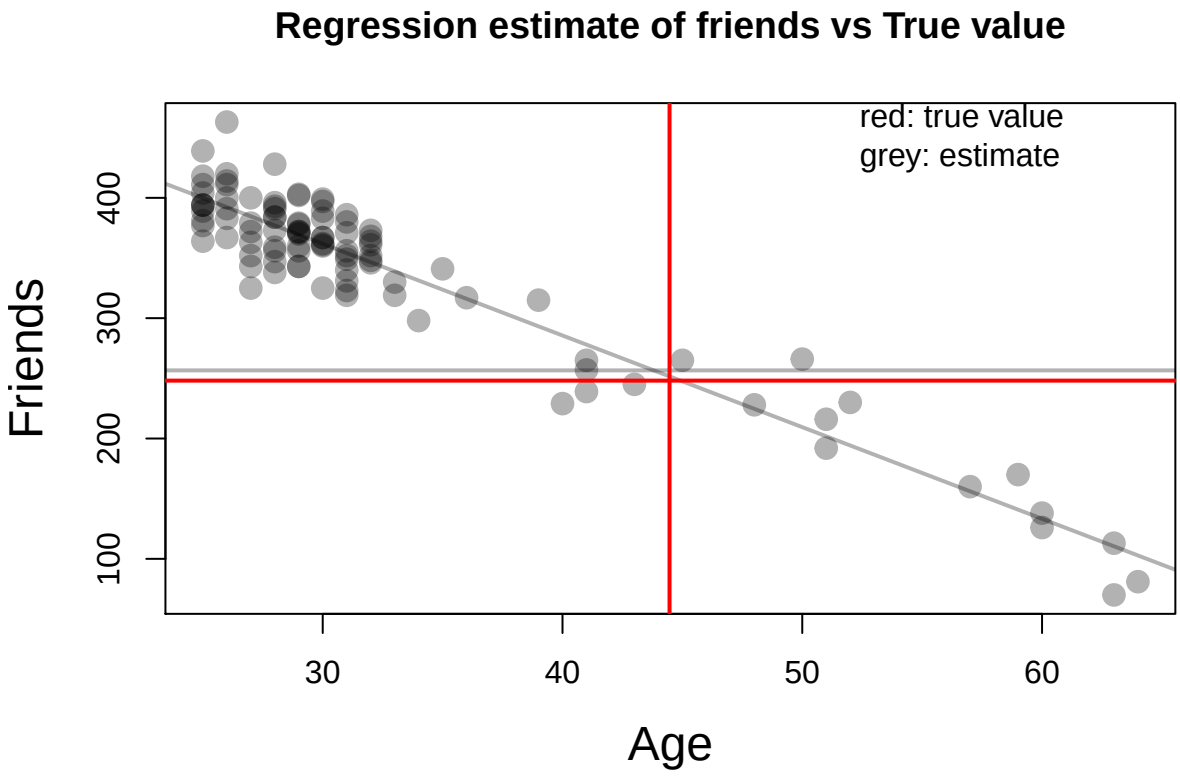
Now, let's see the performance of the regression estimation. First we compute the point estimate and 95% confidence interval:

```
mod <- lm(friends~age)

beta.hat <- coef(mod)[2]

mu.reg <- mean(friends) + beta.hat*(mu.age.true - mean(age))


#95% confidence interval:
residual = friends - mean(friends) - beta.hat*(age - mean(age) )
sigma2.hat.reg <- sum( residual^2 )/(n-1)

alpha = 0.05
Zval = qnorm(0.975)
```

```
friends.CI <- mu.reg + c(-1,+1)*Zval*sqrt( (1-n/N)*sigma2.hat.reg/n )
```

- The regression estimated average for the number of friends is 251.6, which is incredibly close to the true value.
- A 95% confidence interval for the population average using regression estimation is [247, 256.3], which is very narrow.

```
plot(age, friends, pch=19, col=adjustcolor("black",.3), cex.lab=1.5, cex=1.5, xlab="Age"

abline(lm(friends~age), lwd=2, col=adjustcolor("black",.3),lty=1)

abline(h=mu.reg+5, lwd=2, col=adjustcolor("black",.3),lty=1)


abline(v = c( mu.age.true),lwd = 2, col = 'red',lty =c(1))

abline(h = c(248.1),lwd = 2, col = 'red',lty =c(1))

legend(50,500,c('red: true value', 'grey: estimate'),bty='n')
```



where red lines are the true average number of friends and age, and the grey lines are the regression estimated average number of friends and age.

From above we know that if we have additional information (μ_x) related to the estimate of interest, then the estimation will improved in an amazing way.

1.4 Strata Sampling

The idea of strata sampling is to divide the population into the sub-populations which we call strata. And this can group the similar units together and lead to a more precise estimate.

1.4.1 Some useful notations:

- We divide the population U into H strata: $U_1, \ldots, U_H$.
- Size of each strata N_h, h = 1,..,H.
- Stratum weight, $W_h = \frac{N_h}{N}$
- The true average of strata h, μ_h.
- The variance of strata h, σ_h^2.
- $\hat{\mu}_s = \sum_{h=1}^{H} W_h \hat{\mu}_h$
- $E(\tilde{\mu}_s) = \mu$
- $Var(\tilde{\mu}_s) = \sum_{h=1}^{H} W_h^2 (1 - \frac{n_h}{N_h}) \frac{\sigma_h^2}{n_h}$.

Knowing that same formular also apply for post strata sampling, but the only difference between strata and post strata sampling is that we will only know what stratum someone belongs to after they've been picked.

1.4.2 UWaterloo Students Communication Survey, 2015

In this section, we will use an example of UWaterloo student communications survey, 2015.

This is an open survey to all graduates and undergraduates (31,631 students in total). One of the question was asking if the students use the LinkedIn at least once a week. We broke the data by faculty that the students belong to.

We take the students who finish the survey as units in our sample, and get the information about students faculty, whether they use LinkedIn at least once a week. Then calculate the proportion of students using LinkedIn at least once a week. The result is summarized as follows:

```
faculty <- c("AHS","Arts","Eng","Env","Math","Sci")
population <- c(2434,6661,7998,2503,6661,5374)
sample <- c(149,341,202,119,165,223)
prop <- c(0.13,0.26,0.34,0.32,0.29,0.18)
```

```
temp = data.frame(faculty, population, sample, prop)
temp
```

```
##   faculty population sample prop
## 1     AHS       2434    149 0.13
## 2    Arts       6661    341 0.26
## 3     Eng       7998    202 0.34
## 4     Env       2503    119 0.32
## 5    Math       6661    165 0.29
## 6     Sci       5374    223 0.18
```

We then reformat the data to a format that would be easier to use:

```
Nh <- population
Wh <- population/sum(population)
fh <- sample/population
nh <- sample
n <- sum(sample)
wh <- sample/n
pih <- prop


temp2 = data.frame(faculty, Nh, Wh, wh, pih)
names(temp2)[4] = "nh/n"
format(temp2,digits=2)
```

```
##   faculty   Nh    Wh  nh/n  pih
## 1     AHS 2434 0.077 0.124 0.13
## 2    Arts 6661 0.211 0.284 0.26
## 3     Eng 7998 0.253 0.168 0.34
## 4     Env 2503 0.079 0.099 0.32
## 5    Math 6661 0.211 0.138 0.29
## 6     Sci 5374 0.170 0.186 0.18
```

From there we can calculate the estimate of the average which is a proportion π and its 95% confidence interval:

```
pi.strata.hat = sum(Wh *pih)


var.pi.strata.hat = sum(Wh^2 * (1-fh)/nh * pih * (1 - pih))
```

Knowing that the $\sigma^2 = \frac{n}{n-1}\pi(1-\pi) \approx \pi(1-\pi)$ if sample is large.

$\hat{pi} = 0.268$

$\widetilde{Var}(\tilde{\pi}) = \sum_{h=1}^{H} W_h^2(1 - \frac{n_h}{N_h})\frac{\sigma_h^2}{n_h} = 0.00018$

```
Zval  = qnorm(0.975)
strata.ci = pi.strata.hat + c(-1,+1) * Zval * sqrt(var.pi.strata.hat)
```

95% confidence interval is $\hat{\pi} \pm Z_{0.975} \times \sqrt{\widetilde{Var}(\tilde{\pi})} = (0.241, 0.294)$

2 Experimental Design

2.1 Completely Randomized Design

In this section, we will first look at a completely randomized design by randomly assigning the units to different treatments. And then we will analysis the different treatment effects.

For a completely randomized design, we observe our response y_{ij} from the jth unit receiving treatment i. So we model this using:

$$Y_{ij} = \mu + \tau_i + R_{ij}$$

where i=1,...,t. j =1,...,r_i. $R_{ij} \sim N(0, \sigma^2)$ r_i is the number of replicates under each treatment.

- Useful notations:
- μ: the overall mean response across all treatments.
- τ_i: the mean treatment effect for treatment i.
- $\mu + \tau_i$: the mean response for treatment i.
- $\sum_i r_i \tau_i = 0$: the constraint of the balanced design.

$+SS_{total} = SS_{treatment} + SS_{residual}$

2.1.1 Estimation using OLS

We will be using ordinary least square estimation to estimate the parameters. First, construct the Lagrangion:

$$L(\mu, \tau_1, ..., \tau_t, \lambda) = \sum_{i=1}^{t} \sum_{j=1}^{r_i} (y_{ij} - \mu - \tau_i)^2 + \lambda \sum_{i=1}^{t} \tau_i$$

Take parcial derivatives with respect to μ,τ_i and λ and combine the result with some algebra, then we will get the following:

- $\hat{\mu} = \bar{y}_{ij}$: the overall average.
- $\hat{\tau}_i = \bar{y}_{i.} - \bar{y}_{..}$: the average of treatment i minus the overall average.
- $\hat{\tau}_i - \hat{\tau}_j$: the mean difference between the treatment i and treatment j. This is also the thing that we are interested in quite often.

2.1.2 Milk Protein Example

2.1.2.1 Taste of data

We will study the impact on milk protein level from 3 different diets: barley, lupins, barley and lupins. And see if the different diets have any differences among their milk protein levels.

```
milk <- read.csv("Milk.csv")

dim(milk)
```

```
## [1] 1337    4
```

```
head(milk)
```

```
##   protein Time Cow   Diet
## 1    3.63    1 B01 barley
## 2    3.57    2 B01 barley
## 3    3.47    3 B01 barley
## 4    3.65    4 B01 barley
## 5    3.89    5 B01 barley
## 6    3.73    6 B01 barley
```

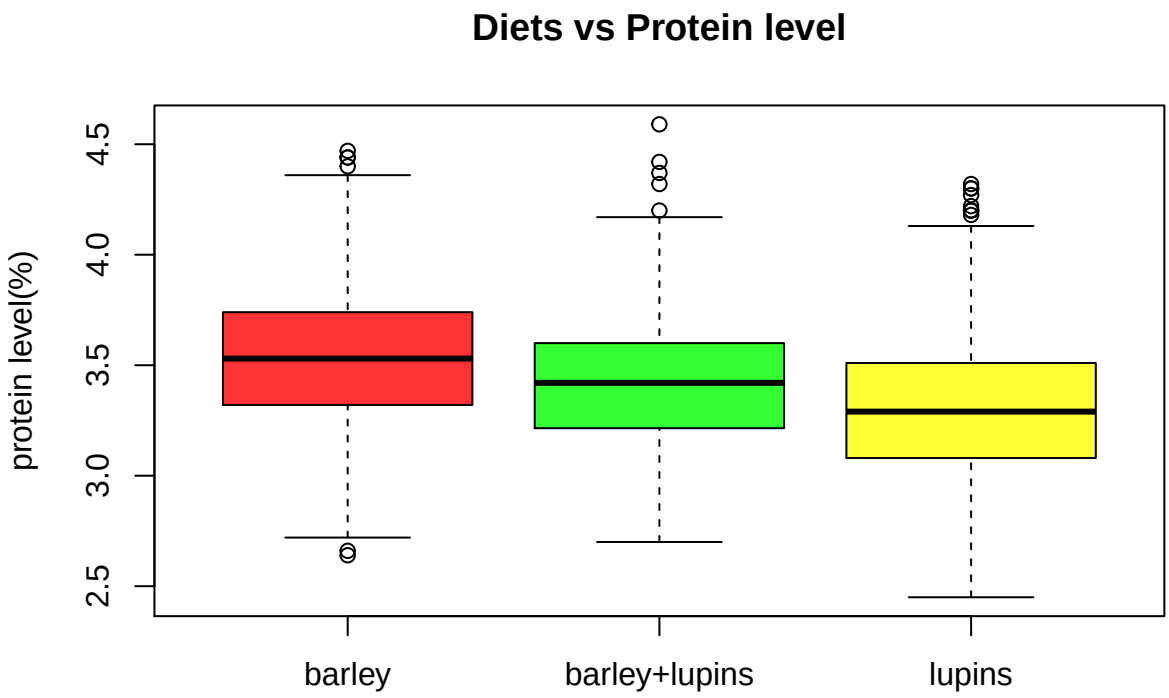
By calling the dimension and head function, we got a sense of how the data looks like: It has 1337 observations with 4 columns. Also, we saw the first 6 row of the data in above table.

```
summary(milk$Diet)
```

```
##        barley barley+lupins        lupins
##           425           459           453
```

Also, we can know that there are 425 milk using diet barley, 459 milk using diet barley and lupins, 453 milk using diet lupins only.

```
boxcol = c(adjustcolor("red", 0.8), adjustcolor("green", 0.8), adjustcolor("yellow", 0.
boxplot(protein~Diet, data=milk, col=boxcol, ylab = 'protein level(%)', main='Diets vs P
```



From the box plot, we find the data are kind of following normal distribution and variation are kind of similar. But we can test the assumption of equal variances by doing Levene's test:

```
library(car)
```

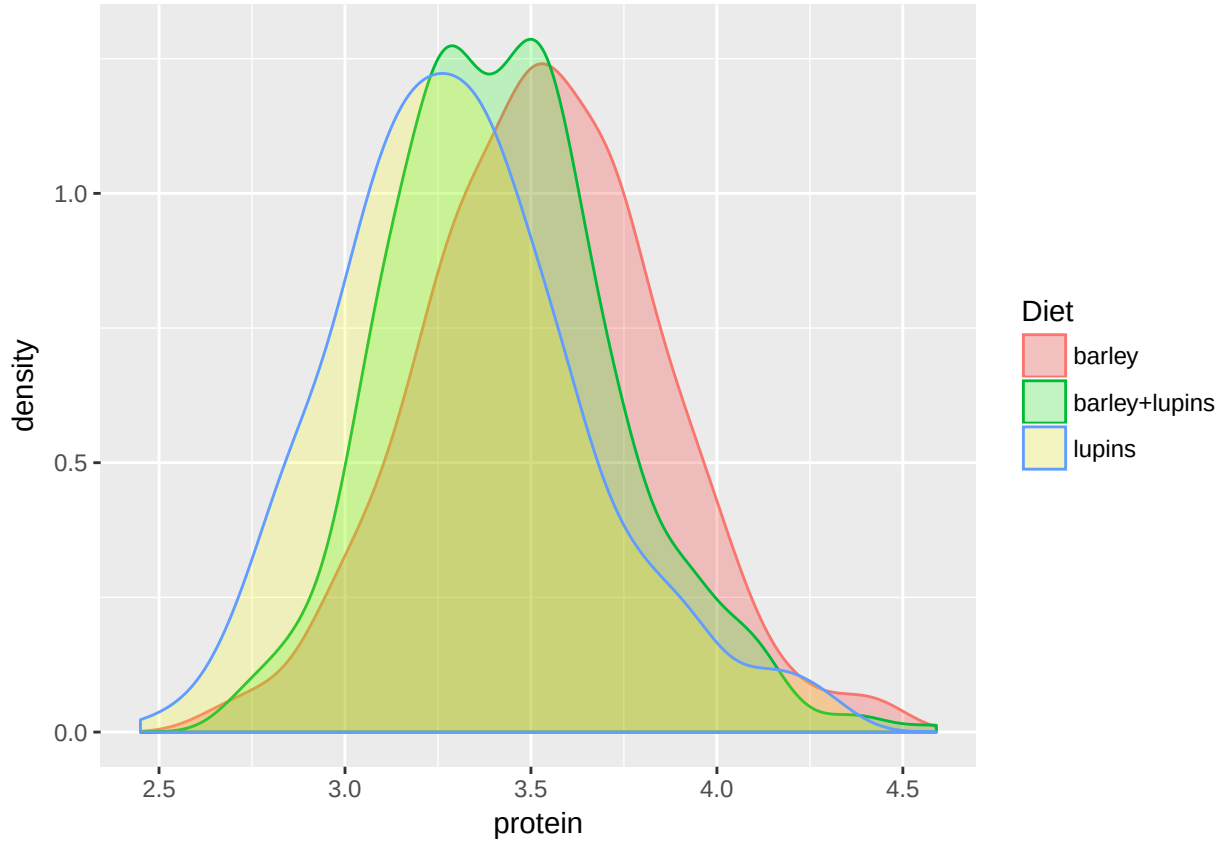
```
## Loading required package: carData
```

```
leveneTest(protein~Diet, data=milk, center='mean')
```

```
## Levene's Test for Homogeneity of Variance (center = "mean")
##         Df F value Pr(>F)
## group    2  1.8377 0.1596
##       1334
```

The p-value of equal variance is $0.1596 > 0.05$, we do not reject equal variance a 95% confidence level.

```
library(ggplot2)
ggplot(milk, aes(x=protein, color=Diet, fill=Diet),main='Density of protein levels among
geom_density(alpha=.2) + scale_fill_manual(values=boxcol)
```




```
round(tapply(milk$protein,milk$Diet,mean),2)
```

```
##        barley barley+lupins        lupins
##          3.53          3.43          3.31
```

We then use a ggplot to show the distribution of the three in one graph.

And taking a look at the average percentage protein level in these tree different kinds of milk, it shows their percentage protein level are 3.53%, 3.43%, 3.31%, which are similar and barley milk contains most protein.

But let's do a diagnostic test on that to see if the diets are indifferent or not.

2.1.2.2 ANOVA F-test

We wanna check that if all the treatment effects are the same, and this is equivalent to $H_0 : \tau_1 = ... = \tau_t$.

We can not use t-test this time, since we are testing multi-parameters simultaneously. So we use ANOVA test which based on two variances:

- $\sigma_t^2 = \frac{\sum_i r_i(\bar{y}_{i.}-\bar{y}_{..})^2}{t-1}$: mean square of the treatments
- $\sigma_r^2 = \frac{\sum_{ij}(y_{ij}-y_{..})^2}{\sum_i(r_i-1)}$: mean square of the residuals

- $\frac{df_t \tilde{\sigma}_t^2}{\sigma^2} \sim \chi^2_{df_t}$—I
- $\frac{df_r \tilde{\sigma}_r^2}{\sigma^2} \sim \chi^2_{df_r}$—II
- $F = \frac{I/df_t}{II/df_r} = \frac{\sigma_t^2}{\sigma_r^2} \sim F_{t-1,\sum(r_i-1)}$

If the hypothesis is true, we expect the ratio of the two variance close to 1.

```
summary(aov(protein~Diet,data=milk))
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)
## Diet           2  10.61   5.303   51.85 <2e-16 ***
## Residuals   1334 136.43   0.102
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The summary of the anova told us that the p-value is less that 2×10^{-16}, we reject that all the treatment effects (impact of different diets) are the same.

2.1.2.3 Contrast and Hypothesis test

If so, question becomes "Is there any two kinds of diets have same protein level?". For this question, we will construct a contrast to check it.

A contrast θ is a linear combination of the treatment effects where the sum of the coefficients is zero. ie, $\theta = \sum_i^t a_i \tau_i$ where $\sum_i a_i = 0$.

Knowing that we can get the distribution of θ by calculating the $E(\theta)$ and $Var(\theta)$. And due to the assumption of the residuals follow a normal distribution, and treatment effect is just an another linear expression of the residuals, $\tilde{\theta} \sim N(\sum a_i \tau_i, \sigma^2 \sum_{i=1}^t \frac{a_i^2}{r_i})$

We here will see an example by checking the difference between barley and lupins, $\theta = \tau_1 - \tau_2 = y_{1.} - y_{3.}$, by finding both confidence interval and p-value:

```
a = c(1,0,-1)
tauavg = tapply(milk$protein, milk$Diet, mean)
r = table(milk$Diet)
theta1 = sum(a*tauavg)

mod1 = aov(protein~Diet, data=milk)

sigma.theta1 = sum(mod1$residuals^2)/mod1$df.residual
se.theta1 = sqrt(sigma.theta1*sum(a^2/r))

theta1.ci = theta1 + c(-1,1)*qnorm(0.975)*se.theta1

theta.obs = theta1/se.theta1
pv = 2*pt(abs(theta.obs), df = mod1$df.residual,lower.tail = F)
```

H_0: $\theta = 0$ ie, $\tau_1 = \tau_2$

We can see that the estimated theta is 0.2195453, with its 95% confidence interval (0.1772169, 0.2618737). Noticing that zero is not included in it, meaning the protein level of barley vs lupins are not the same at 95% confidence level.

Also the result using p-value is p-value $= 1.96 \times 10^{-23}$, which is far smaller than 0.05, which also comfirmed that we should reject the null hypothesis.

2.2 Ramdomized Block Design

For a randomized block design without replication, we add block effect into the analysis. So we extend our model to be:

$$Y_{ij} = \mu + \tau_i + \beta_j + R_{ij}$$

where i=1,...,t. j =1,...,b. $R_{ij} \sim N(0, \sigma^2)$ t is the number of treatments, and b is number of blocks in this design.

- Useful notations:
- μ: the overall mean response across all treatments.
- $\hat{\tau}_i = \bar{y}_{i.} - \bar{y}_{..}$: the estimated mean treatment effect for treatment i.
- $\hat{\beta}_j = \bar{y}_{.j} - \bar{y}_{..}$: the estimated mean block effect for block j.
- $\sum \tau_i = 0$ and $\sum \beta_j = 0$: the constraints of the model.
- $SS_{total} = SS_{treatment} + SS_{block} + SS_{residuals}$

2.2.1 Company Software Choices Example

There is an example about a company wishing to replace some softwares. There are 4 different softwares (A-D) they are using and there are 6 different tasks that used to test the usefulness of the software. Now the company assigns 24 similar employees to the 6 tasks in a group of 4 randomly. Each people in the group is assigned to use a software randomly.

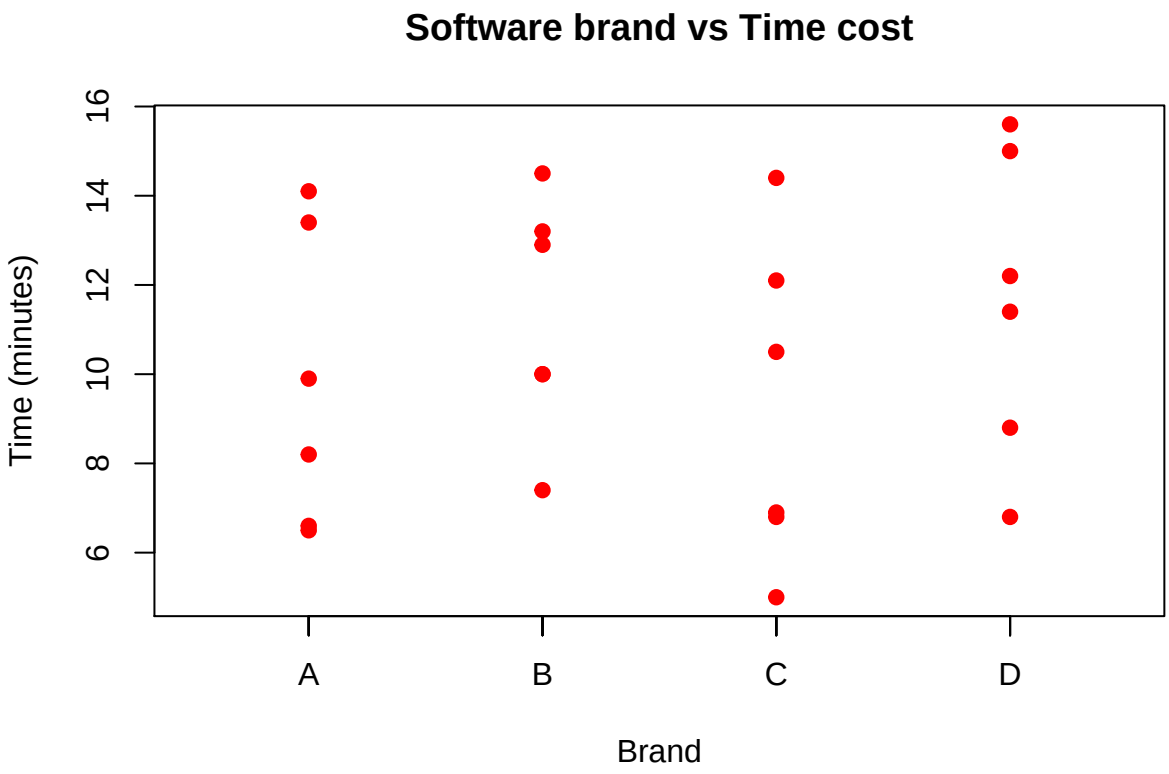
Let's summarize the information into the table:

```
RBD <- read.csv('RBDProducts.csv')

library(kableExtra)
kable(RBD,align='c')
```

task	brand	time
1	A	6.5
1	B	10.0
1	C	5.0
1	D	6.8
2	A	14.1
2	B	14.5
2	C	14.4
2	D	15.0
3	A	9.9
3	B	13.2
3	C	10.5
3	D	12.2
4	A	13.4
4	B	12.9
4	C	12.1
4	D	15.6
5	A	6.6
5	B	10.0
5	C	6.8
5	D	8.8
6	A	8.2
6	B	7.4
6	C	6.9
6	D	11.4

2.2.1.1 ANOVA Test

```
plot(as.numeric(RBD$brand), RBD$time, xlab = 'Brand', ylab = 'Time (minutes)',xaxt='n',
axis(side=1, as.numeric(RBD$brand), RBD$brand )
```



We plot the data above to get a sense of it, and we find that the data variates a lot. That's because we didn't eliminate the block effect. We couldn't tell much information from that. So let's do an ANOVA test with seperated treatment effect and block effect:

```
mod2 <- aov(time ~ brand + as.factor(task), data = RBD)
summary(mod2)
```

```
##                 Df Sum Sq Mean Sq F value   Pr(>F)
## brand            3  23.83    7.94   5.003   0.0133 *
## as.factor(task)  5 190.94   38.19  24.048 1.15e-06 ***
## Residuals       15  23.82    1.59
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

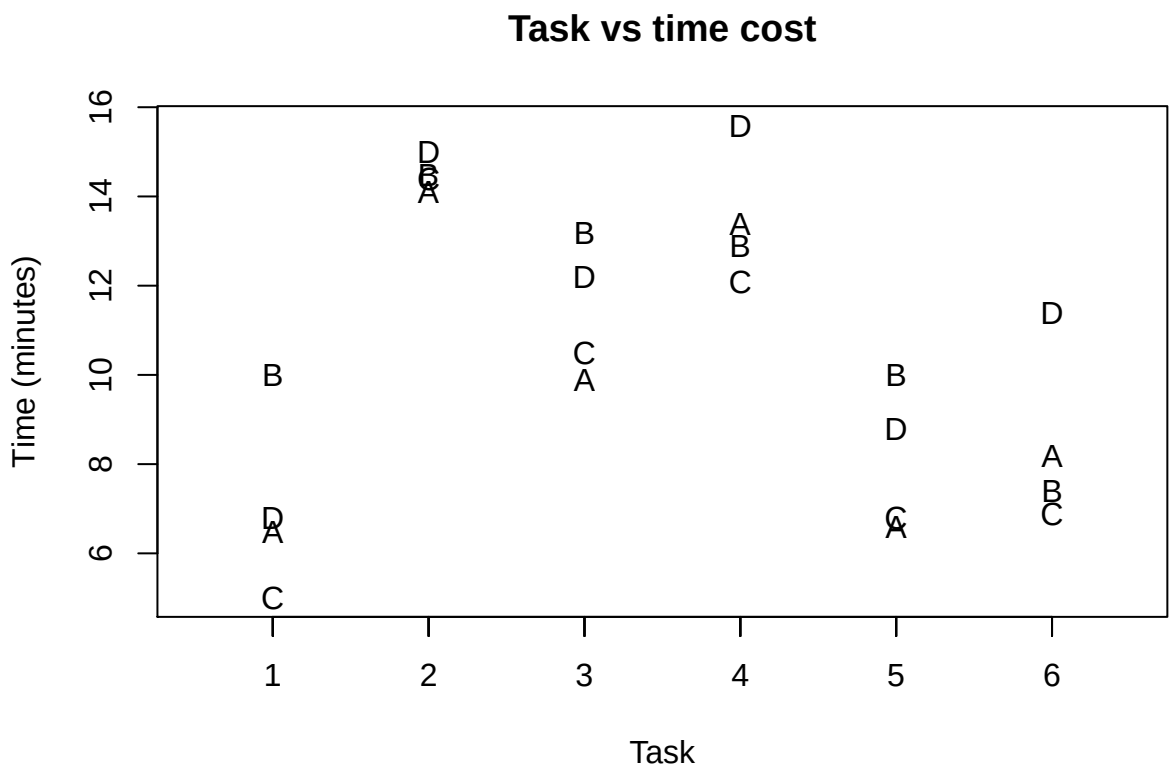
Now, we can see that the p-value of the block effect which is the effect of different tasks on the time cost is $1.15 \times 10^{-6} < 0.05$, saying that we reject that there is no block effect. For same reason we reject treatment effects are equal as well.

2.2.1.2 Contrast Test

But now the question becomes which treatment (brand of the software) is the most efficient to use, ie cost shortest time to finish the task. For this question, we can still perform a contrst

test. But let's first have a generate sense about which brand of software is most likley to be the best:

```
plot( as.numeric(RBD$task), RBD$time , xlab = 'Task', ylab = 'Time (minutes)', xaxt='n',
axis(side=1, as.numeric(RBD$task), RBD$task )
```



It looks like that, even for different tasks, using software C may be the most efficient one to finish the task, then A is the second efficient one.

With some useful results:

- $\tilde{\theta} = \sum_{i=1}^{t} a_i \bar{Y}_{i\cdot} \sim N(\theta, \sigma^2 \frac{\sum a_i^2}{b})$, where b is the number of blocks.

```
a = c(-1,0,1,0)
b = 6
y = aggregate(time ~ brand, RBD, mean)

tau.c = y[3,2] - mean(y$time)
tau.a = y[1,2] - mean(y$time)

theta2 = tau.c - tau.a

var.theta2 = sum(mod2$residuals^2)/mod2$df.residual
se.theta2 = sqrt(var.theta2*sum(a^2)/b)
```

```
theta2.obs = theta2/se.theta2
pv2 = 2*pt(abs(theta2.obs), df = mod2$df.residual,lower.tail = F)
```

H_0: the software of brand C is as efficient as software of brand A across different tasks, ie $\tau_c - \tau_a = 0$.

We can see that, the estimated difference between τ_c and τ_a is -0.5. But rom the hypothesis test, the associated p-value of the hypothesis is 0.5024226.

WOW! What can we learn from that? May be we do need to buy the services of software A but just use C now.

For any more similar question, we still can use contrast to perform hypothesis test on that.

2.3 Factorial Design

So far, we have seen how to exam the individual treatment effect and/or using blocks to help isolating the uninterested influencial factors. Now what if we are interested in the factors with interaction among them, or even with block effects?

- Some useful notations:
- To include the interaction in the model, we could write our model in a way like this:

$$Y_{ijk} = \mu + \alpha_i + \lambda_j + \gamma_{ij} + \beta_k + R_{ijk}$$

  where $R_{ijk} \sim N(0, \sigma^2)$, and γ_{ij} is the interaction term among between ith level of factor A and jth level of factor B.

- $\sum \alpha_i = 0$
- $\sum \lambda_j = 0$
- $\sum \beta_k = 0$
- $\sum_i \gamma_{ij} = 0$ for $\forall j$
- $\sum_j \gamma_{ij} = 0$ for $\forall i$

2.3.1 Ketchup vs Gravy Example

Let's see an simple example for handling interaction using a ketchup vs gravy example:

When people order french fries, they also can have four different choices of the sauce on it. Those are ketchup, gravy, both ketchup and gravy, and no sauce on it.

People will score the tastes of deliciousness based on a 0 to 10 scale. We found 32 volunteers and randomly assigned them to the different combination of the fries sauces.

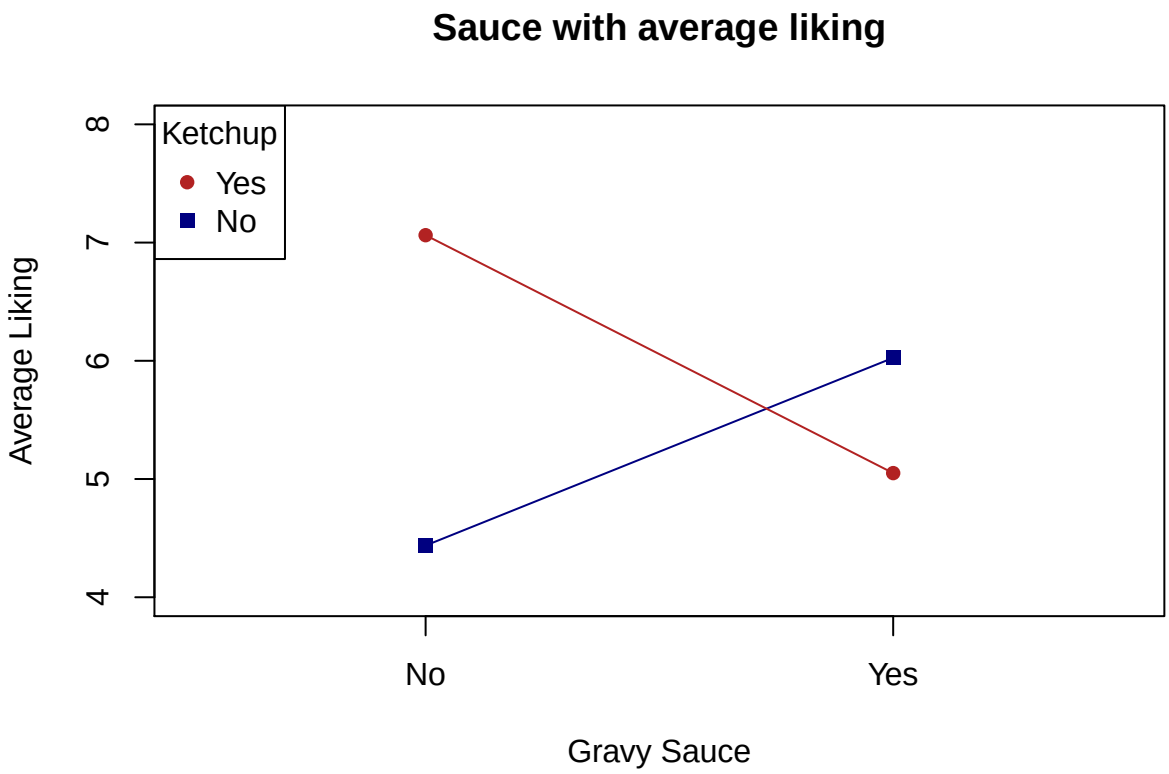
2.3.1.1 Interaction Estimation

- We can test the treatment effect as well as the interaction using plot:

```
Fries <- read.csv('FriesData.csv')
head(Fries)
```

```
##   treatment    ketchup    gravy response
## 1         1 ketchup.no gravy.no      6.3
## 2         1 ketchup.no gravy.no      4.4
## 3         1 ketchup.no gravy.no      5.4
## 4         1 ketchup.no gravy.no      7.0
## 5         1 ketchup.no gravy.no      2.7
## 6         1 ketchup.no gravy.no      3.3
```

```
mean_of_trt <- tapply(Fries$response,list(Fries$ketchup,Fries$gravy),mean)

plot(1, 1, xlim = c(0,4), ylim = c(4,8),xaxt = 'n', xlab = 'Gravy Sauce', ylab='Average

axis(side = 1, at = c(1,3), labels = c('No', 'Yes'))

points(c(1,3), mean_of_trt[1,], type='o', col='navy', pch=15)
points(c(1,3), mean_of_trt[2,], type='o', col='firebrick', pch=16)


legend('topleft', col=c('firebrick', 'navy'), pch = c(16,15), legend=c('Yes', 'No'), tit
```



From the plot we can see that, there is a great improvement having only ketchup sauce when there is no grave on fries. The same is true when the choice of sauce are opposite. But when both ketchup and gravy are added into the fries, we see the people don't like it.

That illustrate that there is interaction between the two treatments.

- Or we can exam the interaction using a contrast:

Under the hypothesis that there is no interaction between the two treatments, we should observe the effects on deliciousness of fries when adding the gravy sauce should be the same whether or not there is ketchup.

- τ_4: the treatment effect of adding both gravy and ketchup.
- τ_3: the treatment effect of adding only ketchup while no gravy.
- τ_2: the treatment effect of adding only gravy while no ketchup.
- τ_1: the treatment effect of adding no sauce.
- $\theta = (\tau_4 - \tau_3) - (\tau_2 - \tau_1)$: the contrast we are testing should be close to zero, if there is no interaction (null hypothesis).
- $Var(\tilde{\theta}) = \sigma^2 \frac{\sum a_i^2}{r}$

```r
a = c(1,-1,-1,1)
r = 8

theta3 = (mean_of_trt[2,2] - mean_of_trt[2,1]) - (mean_of_trt[1,2] - mean_of_trt[1,1])

mod.3 = aov(response~ ketchup*gravy, data= Fries)

summary(mod.3)
```

```
##                Df Sum Sq Mean Sq F value  Pr(>F)
## ketchup         1   5.45   5.445   2.803 0.10523
## gravy           1   0.36   0.361   0.186 0.66960
## ketchup:gravy   1  25.92  25.920  13.343 0.00106 **
## Residuals      28  54.39   1.943
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
sigma3 = (1/mod.3$df.residual)*sum(mod.3$residuals^2)

var.theta3 = sigma3*sum(a^2)/r

obs.3 = theta3/sqrt(var.theta3)

pv3 = 2*pt(abs(obs.3),df = mod.3$df.residual,lower.tail = F)
```

H_0: there is no interaction between ketchup and gravy. ie, $\theta = 0$

From the hypothesis test, we find that the estimated θ is -3.6, and p-value is 0.00106 less than 0.05. We reject that there is no interaction between the two.

2.3.2 Chemical Bath Example

Next we will see a more complex exam which involves interaction and blocks in the experiment at the same time:

We want to test the effect of four chemical baths on strenghth of cloth. These chemical baths were made using combinations of chemicals X and Y. In order to eliminate the effects of variety of clothes, we treat the 5 different types of clothes as blocks and record the strenghth of the clothes as follows:

			Cloth				
Bath	chemical X	chemical Y	1	2	3	4	5
1	low	low	73	68	74	71	67
2	low	high	73	67	75	72	70
3	high	low	75	68	78	73	68
4	high	high	73	71	75	75	69

```r
cloth <- data.frame(
  bath = c(rep(1,5),rep(2,5),rep(3,5),rep(4,5)),
  block = c(1,2,3,4,5,1,2,3,4,5,1,2,3,4,5,1,2,3,4,5),
  X = c(rep('low',10),rep('high',10)),
  Y = c(rep('low',5),rep('high',5),rep('low',5),rep('high',5)),
  response = c(73,68,74,71,67,73,67,75,72,70,75,68,78,73,68,73,71,75,75,69)
)


library(kableExtra)
kable(cloth,align='c')
```

bath	block	X	Y	response
1	1	low	low	73
1	2	low	low	68
1	3	low	low	74
1	4	low	low	71
1	5	low	low	67
2	1	low	high	73
2	2	low	high	67
2	3	low	high	75
2	4	low	high	72
2	5	low	high	70
3	1	high	low	75
3	2	high	low	68
3	3	high	low	78
3	4	high	low	73
3	5	high	low	68
4	1	high	high	73
4	2	high	high	71
4	3	high	high	75
4	4	high	high	75
4	5	high	high	69

In this case, we model the experiment using the model:

$$Y_{ijk} = \mu + \alpha_i + \lambda_j + \gamma_{ij} + \beta_k + R_{ijk}$$

- constraints:
- $\sum \alpha_i = 0$
- $\sum \lambda_j = 0$
- $\sum \beta_k = 0$
- $\sum_i \gamma_{ij} = 0$ for $\forall j$

- $\sum_j \gamma_{ij} = 0$ for $\forall i$

2.3.2.1 ANOVA Test

```
summary(aov(response ~ X+Y+X*Y + as.factor(block), data = cloth))
```

```
##                   Df Sum Sq Mean Sq F value   Pr(>F)
## X                  1  11.25   11.25   6.193   0.0285 *
## Y                  1   1.25    1.25   0.688   0.4230
## as.factor(block)   4 157.00   39.25  21.606 2.06e-05 ***
## X:Y                1   0.45    0.45   0.248   0.6277
## Residuals         12  21.80    1.82
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Using ANOVA test, we can see that from the F-test that there is no interaction effect of the Chemical X and Y ! Since p-value is 0.6277, we have no evidence to reject that there is no interaction at a 95% confidence level.

And using only Chemical X can make a greater positive impact on the strenghth of the cloth than using only Chemical Y.

So in all, we may just suggest using Chemical X for the cloth makers.