AMS 597: Statistical Computing

Pei-Fen Kuan (c)

Applied Math and Stats, Stony Brook University

One way analysis of variance

- In this section, we consider comparisons among more than two groups parametrically, using analysis of variance.
- Let x_{ij} denote the $j = 1, ..., n_i$th observation in the ith group $(i = 1, ..., k)$, $\bar{x}_{i.}$ is the mean of the ith group, and $\bar{x}_{..}$ is the mean of all observations.
- We assume $x_{ij} \sim N(\mu_i, \sigma^2)$ and would like to test $H_0 : \mu_1 = \mu_2 = ... = \mu_k$
- We can decompose the observations as

$$x_{ij} = \bar{x}_{..} + (\bar{x}_{i.} - \bar{x}_{..}) + (x_{ij} - \bar{x}_{i.})$$

One way analysis of variance

- Informally, this also corresponds to the model

$$x_{ij} = \mu + \alpha_i + \epsilon_{ij}, \quad \epsilon_{ij} \sim N(0, \sigma^2)$$

in which the hypothesis is that all the group means are same. Now consider the following sums of squares, known as variation within groups

$$SSW = \sum_{ij} (x_{ij} - \bar{x}_{i.})^2$$

and variation between groups

$$SSB = \sum_{i} n_i (\bar{x}_{i.} - \bar{x}_{..})^2$$

One way analysis of variance

- Note that
$$SSW + SSB = SST = \sum_{ij}(x_{ij} - \bar{x}_{..})^2$$
- Let $MSW = SSW/(N-k)$, and $MSB = SSB/(k-1)$. We calculate the F-statistics $F = MSB/MSW$
- $F \sim F_{k-1,N-k}$ under $H_0 : \mu_1 = \mu_2 = ... = \mu_k$

One way analysis of variance

- ANOVA Table

Source of variation	df	SS	MS	F	p
Between groups	$k-1$	SSB	MSB	MSB/MSW	pvalue
Within groups	$N-k$	SSW	MSW		
Total	$N-1$				

One way analysis of variance

```
y1 <- c(18.2, 20.1, 17.6, 16.8, 18.8, 19.3, 19.1)
y2 <- c(17.4, 18.7, 19.1, 16.4, 15.2, 18.4)
y3 <- c(15.2, 18.8, 17.7, 16.5, 15.9, 17.1, 16.3)
y <- c(y1, y2, y3)
n <- c(7, 6, 7)
group <- c(rep(1, 7), rep(2, 6), rep(3, 7))
ydata <- data.frame(y = y, group = factor(group))
fit <- lm(y ~ group, data = ydata)
```

One way analysis of variance

```
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value  Pr(>F)
## group      2 11.063  5.5315  3.4396 0.05567 .
## Residuals 17 27.339  1.6082
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

```
summary(aov(y ~ group, data = ydata))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## group        2  11.06   5.531    3.44 0.0557 .
## Residuals   17  27.34   1.608
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

Two way ANOVA

- Consider the model which decomposes observations into a general level, a row effect, a column effect and a noise term.

$$x_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij}, \quad \epsilon_{ij} \sim N(0, \sigma^2)$$

in which $\sum_i \alpha_i$ and $\sum_j \beta_j$. Let x_{ij} denote the jth observation in the ith row, $\bar{x}_{i.}$ is the mean of the ith row, and $\bar{x}_{..}$ is the overall mean.

Two way ANOVA

```
heart.rate <- data.frame(hr = c(96, 110, 89, 95, 128,
    100, 72, 79, 100, 92, 106, 86, 78, 124, 98, 68,
    75, 106, 86, 108, 85, 78, 118, 100, 67, 74, 104,
    92, 114, 83, 83, 118, 94, 71, 74, 102), subj = gl(9,
    1, 36), time = gl(4, 9, 36, labels = c(0, 30, 60,
    120)))
str(heart.rate)
```

```
## 'data.frame':    36 obs. of  3 variables:
##  $ hr  : num  96 110 89 95 128 100 72 79 100 92 ...
##  $ subj: Factor w/ 9 levels "1","2","3","4",..: 1 2 3 4 5 6
##  $ time: Factor w/ 4 levels "0","30","60",..: 1 1 1 1 1 1 1
```

Two way ANOVA

- The `gl` (generate levels) function is specially designed for generating patterned factors for balanced experimental designs.
- It has three arguments: the number of levels, the block length (how many times each level should repeat), and the total length of the result. The two patterns in the data frame are thus

```
gl(9, 1, 36)
```

```
##  [1] 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9
## Levels: 1 2 3 4 5 6 7 8 9
```

```
gl(4, 9, 36, labels = c(0, 30, 60, 120))
```

```
##  [1] 0   0   0   0   0   0   0   0   0   30  30  30  30  30
## [20] 60  60  60  60  60  60  60  60  120 120 120 120 120 12
## Levels: 0 30 60 120
```

Two way ANOVA

```
anova(lm(hr ~ subj + time, data = heart.rate))
```

```
## Analysis of Variance Table
##
## Response: hr
##           Df Sum Sq Mean Sq F value    Pr(>F)
## subj       8 8966.6 1120.82 90.6391 4.863e-16 ***
## time       3  151.0   50.32  4.0696   0.01802 *
## Residuals 24  296.8   12.37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

Two way ANOVA

- Exercise: Extend the heart.rate dataset to make it unbalanced (call it heart.rate2) using the following code:

```
heart.rate2 <- rbind(heart.rate, c(80, 9, 120))
```

- Compare the results
  - anova(lm(hr~subj+time,data=heart.rate2))
  - anova(lm(hr~time+subj,data=heart.rate2))

Two way ANOVA

- Discussion on type I, II and III sum of squares (SS)
- Suppose Y=A+B+A*B
- Type I SS (also known as sequential SS):
  - A: SS(A)
  - B: SS(B|A)
  - AB: SS(AB|A,B)

Two way ANOVA

- Type II SS:
  - A: SS(A|B)
  - B: SS(B|A)
  - AB: SS(AB|A,B)
- Type III SS:
  - A: SS(A|B,AB)
  - B: SS(B|A,AB)
  - AB: SS(AB|A,B)

ANOVA table in regression analysis

- The variation between and within groups for a one-way analysis of variance generalizes to model variation

$$SS_{model} = \sum_i (\hat{y}_i - \bar{y}.)^2$$

and residual variation

$$SS_{resid} = \sum_i (y_i - \hat{y}_i)^2$$

which partition the total variation

$$SS_{total} = \sum_i (y_i - \bar{y}.)^2$$

- This applies only when the model contains an intercept. The role of the group means in the one-way classification is taken over by the fitted values in the more general linear model.

ANOVA table in regression analysis

- Data url: "http://www.ams.sunysb.edu/~pfkuan/Teaching/AMS597/Data/d_logret_6stocks.txt", also on Brightspace.

```
logret <- read.table(paste0(dataPath, "d_logret_6stocks.txt"),
    header = T)
fit1 <- lm(Pfizer ~ Intel, data = logret)
anova(fit1)
```

```
## Analysis of Variance Table
##
## Response: Pfizer
##            Df   Sum Sq    Mean Sq F value Pr(>F)
## Intel       1 0.000154 0.00015441  0.2865 0.5944
## Residuals 62 0.033409 0.00053885
```

ANOVA table in regression analysis

```
fit2 <- lm(Pfizer ~ Intel + AmerExp, data = logret)
anova(fit2)
```

```
## Analysis of Variance Table
##
## Response: Pfizer
##            Df   Sum Sq    Mean Sq F value Pr(>F)
## Intel       1 0.000154 0.00015441  0.2942 0.5895
## AmerExp     1 0.001396 0.00139571  2.6595 0.1081
## Residuals 61 0.032013 0.00052480
```

ANOVA table in regression analysis

```
SST <- sum((logret$Pfizer - mean(logret$Pfizer))^2)
SSM <- sum((fit1$fitted - mean(logret$Pfizer))^2)
SSR <- sum((logret$Pfizer - fit1$fitted)^2)
SSM + SSR
```

```
## [1] 0.03356306
```

```
SST
```

```
## [1] 0.03356306
```

ANOVA table in regression analysis

- Exercise: Can you fit regression through the origin to the above dataset and compare the sum of squares?

Kruskal-Wallis

- Nonparametric inference for more than two samples
- $$Y_{ij} = \mu_i + \epsilon_{ij}$$

  for $i = 1, \ldots, K; j = 1, \ldots, n_i$
- ϵ_{ij} are independent and identically distributed with mean zero, but not necessarily normal
- $H_0 : \mu_1 = \cdots = \mu_K$ vs H_a : these μ_i's are not all equal
- Pool all N observations and rank from smallest to largest
- Let R_{ij} be the rank of the j^{th} obs in the i^{th} group
- Let $\bar{R}_i = \sum_{j=1}^{n_i} R_{ij}/n_i$ equal the average rank in the i^{th} group
- Let $\bar{R}$ denote the overall average rank

Kruskal Wallis

- $$T_{KW} = \frac{12 \sum_{i=1}^{K} n_i (\bar{R}_i - \bar{R})^2}{N(N+1)}$$
- Equivalently

$$T_{KW} = \frac{12 \sum_{i=1}^{K} (\sum_{j=1}^{n_i} R_{ij})^2 / n_i}{N(N+1)} - 3(N+1)$$

- Reject H_0 for large values of T_{KW}

Kruskal Wallis

- There are
$$\binom{N}{n_1 n_2 \cdots n_K} = \frac{N!}{n_1! n_2! n_3! \cdots n_K!}$$
possible ways to assign n_1 ranks to group 1, n_2 ranks to group 2, ...
- Under H_0 each occurs with equal probability, and one can then obtain the exact p-value (similar to the idea of Wilcoxon Rank Sum test).
- If the n_i are moderately large (rule of thumb: $n_i \geq 5$), then
$$T_{KW} \sim \chi^2_{K-1}$$

Kruskal Wallis

```
y1 <- c(18.2, 20.1, 17.6, 16.8, 18.8, 19.3, 19.1)
y2 <- c(17.4, 18.7, 19.1, 16.4, 15.2, 18.4)
y3 <- c(15.2, 18.8, 17.7, 16.5, 15.9, 17.1, 16.3)
y <- c(y1, y2, y3)
n <- c(7, 6, 7)
group <- c(rep(1, 7), rep(2, 6), rep(3, 7))
ydata <- data.frame(y = y, group = factor(group))
kruskal.test(y ~ group, data = ydata)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  y by group
## Kruskal-Wallis chi-squared = 5.5922, df = 2, p-value = 0.06
```

Single proportions

- Tests of single proportions are generally based on the binomial distribution with size parameter N and probability parameter p.
- For large sample sizes, this can be well approximated by a normal distribution with mean Np and variance $Np(1-p)$.
- As a rule of thumb, the approximation is satisfactory when $Np(1-p) \geq 10$.

Single proportions

- Denoting the observed number of "successes" by x, the test for the hypothesis $H_0 : p = p_0$ that can be based on

$$u = \frac{x - Np_0}{\sqrt{Np_0(1 - p_0)}}$$

which has an approximate normal distribution with mean zero and standard deviation 1.

Single proportions

- We consider an example (Altman, 1991, p. 230) where 39 of 215 randomly chosen patients are observed to have asthma and one wants to test the hypothesis that the probability of a "random patient" having asthma is 0.15 (i.e., $H_0 : p = 0.15$). This can be done using `prop.test`:

```
prop.test(39, 215, 0.15)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  39 out of 215, null probability 0.15
## X-squared = 1.425, df = 1, p-value = 0.2326
## alternative hypothesis: true p is not equal to 0.15
## 95 percent confidence interval:
##  0.1335937 0.2408799
## sample estimates:
##         p
```

Single proportions

- Exact test for small sample scenario

```
binom.test(39, 215, 0.15)
```

```
##
##  Exact binomial test
##
## data:  39 and 215
## number of successes = 39, number of trials = 215, p-value =
## alternative hypothesis: true probability of success is not
## 95 percent confidence interval:
##  0.1322842 0.2395223
## sample estimates:
## probability of success
##              0.1813953
```

Two independent proportions

- The function `prop.test` can also be used to compare two or more proportions. For that purpose, the arguments should be given as two vectors, where the first contains the number of positive outcomes and the second the total number for each group.
- The theory is similar to that for a single proportion. Consider the difference in the two proportions $d = \frac{x_1}{N_1} - \frac{x_2}{N_2}$, which will be approximately normally distributed with mean zero and variance $V(p) = (\frac{1}{N_1} + \frac{1}{N_2})p(1-p)$ if the counts are binomially distributed with the same p parameter.

Two independent proportions

- So to test the hypothesis that $H_0 : p_1 = p_2$, plug the common estimate $\hat{p} = \frac{x_1+x_2}{n_1+n_2}$ into the variance formula and the test statistic is $\frac{d}{\sqrt{V(\hat{p})}}$, which follows a standard normal distribution approximately.
- On the other hand, exact test can be conducted based on Fisher's exact test `fisher.test`.

Two independent proportions

```
lewitt.machin.success <- c(9, 4)
lewitt.machin.total <- c(12, 13)
prop.test(lewitt.machin.success, lewitt.machin.total)
```

```
##
##  2-sample test for equality of proportions with continuity
##
## data:  lewitt.machin.success out of lewitt.machin.total
## X-squared = 3.2793, df = 1, p-value = 0.07016
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.01151032 0.87310506
## sample estimates:
##    prop 1    prop 2
## 0.7500000 0.3076923
```

r × c tables

- For the analysis of tables with more than two classes on both sides, you can use `chisq.test` or `fisher.test`, although you should note that the latter can be very computationally demanding if the cell counts are large and there are more than two rows or columns.

r × c tables

- An r × c table looks like this:

		j		
i	1	2	$\cdots$	c
1	n_{11}	n_{12}	$\cdots$	n_{1c}
2	n_{21}	n_{22}	$\cdots$	n_{2c}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
r	n_{r1}	n_{r2}	$\cdots$	n_{rc}

- Notation:

$$n_{i\cdot} = \sum_{j=1}^{c} n_{ij} \qquad n_{\cdot j} = \sum_{i=1}^{r} n_{ij}$$

r × c tables

- One common scenario such a table can arise is counting the frequencies over two variables.
- You would be interested in testing the hypothesis of statistical independence of the two variables, that the probability of an individual falling into the ijth cell is the product $p_i p_j$ of the marginal probabilities. (Test of independence)

r × c tables

- If there is no relation between rows and columns, then you would expect to have the following cell values:

$$E_{ij} = \frac{n_{i.} n_{.j}}{N}$$

- This can be interpreted as distributing each row total according to the proportions in each column (or vice versa) or as distributing the grand total according to the products of the row and column proportions.

r × c tables

- The test statistic

$$X^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(|O_{ij} - E_{ij}| - 0.5)^2}{E_{ij}}$$

  has an approximate chi-squared distribution with $(r-1) \times (c-1)$ degrees of freedom. O_{ij} denotes the observed values and E_{ij} the expected values as described above. The value 0.5 is Yates's correction for continuity.
- Note that in R `chisq.test()`, Yates correction is only implemented for 2×2 table.
- For larger tables

$$X^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

r × c tables

```
caff.marital <- matrix(c(652, 1537, 598, 242, 36, 46,
    38, 21, 218, 327, 106, 67), nrow = 3, byrow = T)
colnames(caff.marital) <- c("0", "1-150", "151-300",
    ">300")
rownames(caff.marital) <- c("Married", "Prev.married",
    "Single")
caff.marital
```

```
##                   0 1-150 151-300 >300
## Married         652  1537      598  242
## Prev.married     36    46       38   21
## Single          218   327      106   67
```

r × c tables

```
chisq.test(caff.marital)
```

```
##
##  Pearson's Chi-squared test
##
## data:  caff.marital
## X-squared = 51.656, df = 6, p-value = 2.187e-09
```

r × c tables

- The test is highly significant, so we can safely conclude that the data contradict the hypothesis of independence. However, you would generally also like to know the nature of the deviations. To that end, you can look at some extra components of the return value of `chisq.test()`

r × c tables

```
chisq.test(caff.marital)$expected
```

```
##                      0      1-150   151-300      >300
## Married      705.83179 1488.01183 578.06533 257.09105
## Prev.married  32.85648   69.26698  26.90895  11.96759
## Single       167.31173  352.72119 137.02572  60.94136
```

```
chisq.test(caff.marital)$observed
```

```
##                    0 1-150 151-300 >300
## Married          652  1537     598  242
## Prev.married      36    46      38   21
## Single           218   327     106   67
```

r × c tables

```
E <- chisq.test(caff.marital)$expected
O <- chisq.test(caff.marital)$observed
(O - E)^2/E
```

```
##                      0    1-150   151-300      >300
## Married      4.1055981 1.612783 0.6874502 0.8858331
## Prev.married 0.3007537 7.815444 4.5713926 6.8171090
## Single      15.3563704 1.875645 7.0249243 0.6023355
```

r $\times$ c tables

- Exercise: Write your own function which performs the chi-squared test for $r \times c$ table