Displays for Statistics 5401/8401

Lecture 11

September 30, 2005

Christopher Bingham, Instructor

612-625-1024, kb@umn.edu
372 Ford Hall

Class Web Page

http://www.stat.umn.edu/~kb/classes/5401

Unpooled two-sample T^2

Parameter vector is $\boldsymbol{\theta} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$

Estimate vector is $\hat{\boldsymbol{\theta}} = \overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2$

- *Unpooled* estimate of $V[\hat{\boldsymbol{\theta}}]$ is

  $\hat{V}[\hat{\boldsymbol{\theta}}] = \hat{V}[\overline{\mathbf{x}}_1] + \hat{V}[\overline{\mathbf{x}}_2] = (1/n_1)\mathbf{S}_1 + (1/n_2)\mathbf{S}_2$

  where $\mathbf{S}_1$ and $\mathbf{S}_2$ are (unbiased) sample variance matrices.

  $\hat{V}[\hat{\boldsymbol{\theta}}]$ is an unbiased estimate of $V[\hat{\boldsymbol{\theta}}]$

- $T^2 = T_{unpooled}^2 = (\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2)'\hat{V}[\hat{\boldsymbol{\theta}}]^{-1}(\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2)$

  $= (\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2)'\left(n_1^{-1}\mathbf{S}_1 + n_2^{-1}\mathbf{S}_2\right)^{-1}(\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2)$

  tests H_0: $\boldsymbol{\theta} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 = \mathbf{0}$

- With large n_1 and n_2, the null distribution of $T_{unpooled}^2 \tilde{=} \chi_p^2$. Thus the test of $\mu_1 = \mu_2$ is

  "reject when $T_{unpooled}^2 > \chi_p^2(\alpha)$"

  You don't need normality, although the further from multivariate normal, the larger the n_i must be for the χ_p^2 approximation to "work as advertised."
- Even with normal $\mathbf{x}_1$ and $\mathbf{x}_2$, and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$, when $n_1 \neq n_2$, $T_{unpooled}^2$ is *not* $((pf_e)/(f_e - p + 1))F_{p,f_e-p+1}$, although using $((pf_e)/(f_e-p+1))F_{p,f_e-p+1}(\alpha)$ to decide significance may "work" better than using $\chi_p^2(\alpha)$.
- Unpooled $T^2 \neq$ "classical" <u>pooled</u> two-sample T^2 except when $n_1 = n_2$.

Classical (pooled) Hotelling's 2 sample T^2

In the special case when $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$

$$V[\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2] = (1/n_1 + 1/n_2)\boldsymbol{\Sigma} = K\boldsymbol{\Sigma},$$

where $K = 1/n_1 + 1/n_2 = (n_1 + n_2)/(n_1 n_2)$.

Now you can estimate $\boldsymbol{\Sigma}$ by the *pooled variance matrix*

$$\hat{\boldsymbol{\Sigma}} = \mathbf{S}_{pooled} = \frac{(n_1 - 1)\mathbf{S}_1 + (n_2 - 1)\mathbf{S}_2}{(n_1 - 1) + (n_2 - 1)}$$

$$= \frac{(n_1 - 1)\mathbf{S}_1 + (n_2 - 1)\mathbf{S}_2}{n_1 + n_2 - 2} = \frac{f_{e_1}\mathbf{S}_1 + f_{e_2}\mathbf{S}_2}{f_e}$$

with $f_e = f_{e_1} + f_{e_2} = n_1 + n_2 - 2$.

$\mathbf{S}_1$ and $\mathbf{S}_2$ are the unbiased sample covariance matrices from the two samples. Because $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$, $\hat{\boldsymbol{\Sigma}}$ is <u>unbiased</u>:,

$$E[\hat{\boldsymbol{\Sigma}}] = E[\mathbf{S}] = (f_{e_1}\boldsymbol{\Sigma} + f_{e_2}\boldsymbol{\Sigma})/(f_{e_1} + f_{e_2}) = \boldsymbol{\Sigma}$$

Recall we are dealing with two independent random samples $\{\mathbf{x}_{i1}\}_{1\le i\le n_1}$ and $\{\mathbf{x}_{i2}\}_{1\le i\le n_2}$. When all the $\mathbf{x}_{i,j}$'s are MVN,

- $f_e\mathbf{S}_{pooled} = W_p(f_e, \mathbf{\Sigma})$, $f_e = n_1 + n_2 - 2$
- $\mathbf{S}_{pooled}$ is *independent* of $\overline{\mathbf{x}}_1$ and $\overline{\mathbf{x}}_2$.

Then the standard (pooled) two sample T^2 statistic to test H_0: $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 = \mathbf{0}$ is

$$T^2 = T_{pooled}^2 = (\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2)'\{\hat{V}[\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2]\}^{-1}(\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2)$$

with

$$\hat{V}[\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2] = K\mathbf{S}_{pooled} = (1/n_1+1/n_2)\mathbf{S}_{pooled}$$

You can factor out the constant $K = (n_1+n_2)/(n_1n_2)$ to get the "special" formula

$$T_{pooled}^2 = (n_1n_2/(n_1+n_2))(\overline{\mathbf{x}}_1-\overline{\mathbf{x}}_2)'\mathbf{S}_{pooled}^{-1}(\overline{\mathbf{x}}_1-\overline{\mathbf{x}}_2)$$

- $T_{pooled}^2 = ((f_ep)/(f_e-p+1))F_{p,f_e-p+1}$

  $= (p(n_1+n_2-2)/(n_1+n_2-p-1))F_{p,n_1+n_2-p-1}$

The assumption that $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2$ is a <u>very strong assumption</u> because it requires

- $\sigma_{jj}^{(1)} = \sigma_{jj}^{(2)}$, $j = 1, \ldots, p$ (<u>equality of variances</u>)
- $\rho_{ij}^{(1)} = \rho_{ij}^{(2)}$, all $1 \le i < j \le p$ (<u>equality of correlations</u>).

You can seldom appeal to *a priori* evidence that two populations with possibly different means should have

- exactly the same variances $\sigma_{11}, \ldots, \sigma_{pp}$

and

- exactly the same $p(p - 1)$ correlations $\rho_{1,2}, \rho_{1,3}, \ldots, \rho_{p-1,p}$.

Instead, you need to use the data to check it.

The problem of testing H_0: $\mu_1 = \mu_2$ *without* assuming that $\Sigma_1 = \Sigma_2$ is the *multivariate* ***Behrens-Fisher*** *problem*.

When $\Sigma_1 \neq \Sigma_2$ and $n_1 \neq n_2$,

$$E[\hat{V}_{pooled}] = E[(1/n_1+1/n_2)\mathbf{S}_{pooled}] \neq V[\overline{\mathbf{x}}_1-\overline{\mathbf{x}}_2].$$

The pooled T^2 is not $(f_e p/(f_e-p+1))F_{p,f_e-p+1}$ and not χ_p^2, <u>even in large samples</u>.

But, when $n_1 = n_2 = n$,

- $\hat{V}_{unpooled} = (1/n_1)\mathbf{S}_1 + (1/n_2)\mathbf{S}_2$
  $= (2/n)\ \mathbf{S}_{pooled} = \hat{V}_{pooled}$
- $T_{unpooled}^2 = (\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2)'(n_1^{-1}\mathbf{S}_1+n_2^{-1}\mathbf{S}_2)^{-1}(\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2)$
  $= (\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2)'((2/n)\mathbf{S}_{pooled})^{-1}(\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2) = T_{pooled}^2$
  will be approximately χ_p^2, whether or not $\Sigma_1 = \Sigma_2$. This provides a reason to use equal sample sizes.

Two sample T^2 computation

```
Cmd> irisdata <- read("","t11_05",quiet:T) #read JWdata5.txt
Read from file "TP1:Stat5401:Data:JWData5.txt"
Cmd> varieties <- irisdata[,1]
Cmd> setosa <- irisdata[varieties == 1,-1]      # Group 1
Cmd> versicolor <- irisdata[varieties == 2,-1]  # Group 2
Cmd> xbar1 <- tabs(setosa,mean:T) # column vector
Cmd> xbar2 <- tabs(versicolor,mean:T) # column vector
Cmd> s1 <- tabs(setosa, covar:T)  # 4 by 4 matrix
Cmd> s2 <- tabs(versicolor, covar:T)
Cmd> n1 <- nrows(setosa) # n1 = 50
Cmd> n2 <- nrows(versicolor) # n2 = 50
Cmd> df1 <- n1 - 1; df2 <- n2 - 1# both 49
Cmd> fe <- df1 + df2 # 98 = n1 + n2 - 2
Cmd> s_pooled <- (df1*s1 + df2*s2)/fe # pooled variance matrix
Cmd> diff <- xbar1 - xbar2 # column vector
Cmd> vhat <- (1/n1 + 1/n2)*s_pooled # vhat[xbar1-xbar2]
Cmd> se <- sqrt(diag(vhat)) # std errors sqrt(vhat[i,i])
Cmd> print(diff, se)
diff:   differences of means
(1)       -0.93       0.658      -2.798       -1.08
se:     standard errors of differences
(1)    0.088395    0.069593    0.070849     0.03169
Cmd> tstats <- diff/se;print(tstats) #2-sample pooled t-stats
tt:
(1)     -10.521       9.455     -39.493      -34.08
Cmd> twotailt(tstats,fe) # two-tail P-values
(1)  8.9852e-18  1.8712e-15  5.4049e-62  3.8311e-56
```

The t-statistics here are classic pooled two-sample univariate t-statistics.

The groups differ very significantly on all 4 variables based on univariate t-tests.

Compute Hotelling's T^2 to test H_0: $\mu_1 = \mu_2$:

```
Cmd> t2 <- diff' %*% solve(vhat) %*% diff; t2
(1,1)      2580.8
Cmd> p <- ncols(setosa) # p = 4
Cmd> f_value <- (fe-p+1)*t2/(fe*p)
Cmd> cumF(f_value,p, fe-p+1,upper:T) # P-value
(1,1)  2.6649e-67
```

This is the "white box" approach. **hotell2val()** allows a "black box" approach:

```
Cmd> hotell2val(setosa,versicolor,pval:T)
component: hotelling
(1,1)      2580.8
component: pvalue
(1,1)           0
```

Bonferronized t-statistics

```
Cmd> t2val(setosa,versicolor,df:T) #pooled
component: t          Pooled 2-sample t and d.f.
(1)     -10.521       9.455     -39.493      -34.08
component: df
(1)          98          98          98          98
Cmd> stuff <- t2val(setosa,versicolor,pooled:F); stuff
component: t          Unpooled 2-sample t and d.f.
(1)     -10.521       9.455     -39.493      -34.08
component: df
(1)      86.538      94.698       62.14      74.755
Cmd> 4*twotailt(stuff$t,stuff$df) # Bonferronized P-values
(1)           0    9.77e-15           0           0
```


$\mathbf{S}_1$ and $\mathbf{S}_2$ are quite different so possibly $\mathbf{\Sigma}_1 \neq \mathbf{\Sigma}_2$:

```
Cmd> print(variances1:diag(s1),variances2:diag(s2))
variances1:    Setosa variances
(1)     0.12425     0.14369    0.030159    0.011106
variances2:    Versicolor variances
(1)     0.26643    0.098469     0.22082    0.039106
```

The variances appear to be different. You could formally test

H_0: $\sigma_{jj}^{(1)} = \sigma_{jj}^{(2)}$, j = 1, ... 4

by Bonferronized F-tests ($F_j = s_{jj}^{(1)}/s_{jj}^{(2)}$) or Levine tests (t-tests computed from $z_{ij} = |x_{ij} - \overline{x}_j|$, see for example, Ott and Longnecker, Ed 5, p. 368).

```
Cmd> z1 <- abs(setosa - xbar1')
Cmd> z2 <- abs(versicolor - xbar2')
Cmd> levinetstats <- t2val(z1,z2,pooled:F); levinetstats
component: t
(1)     -2.9043     0.76051     -5.9514     -3.9224
component: df
(1)      91.554      90.063      65.087      75.844
Cmd> 4*twotailt(levinetstats$t, levinetstats$df)
(1)    0.018455      1.7958  4.6761e-07  0.00076399
```

These are Bonferronized approximate P-values. Conclusion: the variances differ.

Comparison of correlations

```
Cmd> R1 <- cor(setosa); R2 <- cor(versicolor)

Cmd> print(R1, R2)
R1:     Setosa  Correlations
(1,1)          1     0.74255     0.26718      0.2781
(2,1)    0.74255           1      0.1777     0.23275
(3,1)    0.26718      0.1777           1     0.33163
(4,1)     0.2781     0.23275     0.33163           1
R2:     Versicolor  Correlations
(1,1)          1     0.52591     0.75405     0.54646
(2,1)    0.52591           1     0.56052       0.664
(3,1)    0.75405     0.56052           1     0.78667
(4,1)    0.54646       0.664     0.78667           1
```

Here is a graphical method to compare the correlations.

The first few lines extract the correlations below the diagonals into vectors of length 6,

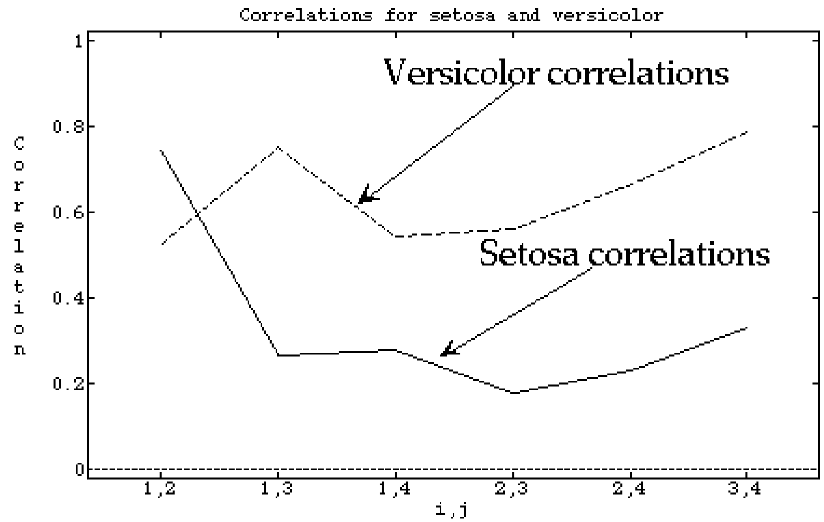
```
Cmd> J <- matrix(vector(1,2, 1,3, 1,4, 2,3, 2,4, 3,4),2)';J
(1,1)          1           2   Matrix of indices of
(2,1)          1           3   correlations below the
(3,1)          1           4   diagonal
(4,1)          2           3
(5,1)          2           4
(6,1)          3           4

Cmd> r1 <- R1[J]; r1 # uses "matrix" subscript
(1)     0.74255     0.26718      0.2781      0.1777     0.23275
(6)     0.33163    Below diagonal setosa correlations

Cmd> r2 <- R2[J]; r2 # see help on topic subscripts
(1)     0.52591     0.75405     0.54646     0.56052       0.664
(6)     0.78667    Below diagonal versicolor correlations
```


I plotted them with the correlations for each sample conected by lines:

```
Cmd> lineplot(1, hconcat(r1,r2), ymin:0, ymax:1,\
       min:.5, xmax:6.5,xticks:run(6),\
      xticklabs:vector("1,2","1,3","1,4","2,3","2,4","3,4"),\
      xlab:"i,j",ylab:"Correlation",\
      title:"Correlations for setosa and versicolor")
```



It looks like most setosa correlations are smaller than the corresponding versicolor correlations.

You can use Fisher's z-transform of the sample correlations to carry out a formal test of

$$H_0: \rho_{ij}^{(1)} = \rho_{ij}^{(2)} = \rho_{ij}, \text{ all } i < j$$

```
Cmd> z1<- atanh(r1); z2<- atanh(r2) # Fisheer z-transforms

Cmd> z <- (z1 - z2)/sqrt(1/(n1-3) + 1/(n2-3)); z
(1)      1.8017     -3.4344     -1.5886     -2.2008     -2.7284
(6)     -3.4805
```

Under H_0 (and approximate multivariate normality), each $z_{ij} = \tanh^{-1} r_{ij}$ is approximately $N(\tanh^{-1}(\rho_{ij}),1/(n_i-3))$.

However, since you are testing them all simultaneously, you need to Bonferronize by K = 6:

```
Cmd> 6*2*cumnor(abs(z),upper:T) # Bonferronized P-values
(1)     0.42958   0.0035637     0.67291     0.16651     0.03818
(6)    0.003003
```

Three differ significantly at the 5% level so you reject H_0.

Note: `2*cumnor(abs(z),upper:T)` computes the non-Bonferronized two-tail P-values.

I did a simulation to evaluate the actual α of this test and the power = $1 - \beta$ when $\mathbf{\Sigma}_1 \neq \mathbf{\Sigma}_2$.

I used M = 10,000 independent pairs of random samples with $n_1 = n_2 = 50$ and $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{S}_{pooled} = (49\,\mathbf{S}_1 + 49\,\mathbf{S}_2)/98$ (H_0 true) and 10,000 pairs of samples with $\mathbf{\Sigma}_1 = \mathbf{S}_1$, $\mathbf{\Sigma}_2 = \mathbf{S}_2$ (H_0 false) ($\mathbf{S}_i$ were the sample variance matrices for *Iris setosa* and *Iris versicolor* data). Here are the results

α	.10	.05	.01
$\hat{\alpha}$	0.0868	0.0452	0.0107
$1 - \hat{\beta}$	0.9936	0.9803	0.8995

The $\hat{\alpha}$ comes from the H_0 true simulation; power = $1 - \hat{\beta}$ (power) line comes from the H_0 false simulation

I did another simulation to see how much $\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$ might affect the distribution of T^2. I generated M = 5000 pairs of samples with $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}_i = \mathbf{S}_i$, i = 1,2 and computed M values of T^2 with $\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$.

Here are the proportions exceeding the small sample critical values for various α's when $n_1 = n_2 = 50$ (equal n).

α	.10	.05	.01
$\hat{\alpha}$	.1094*	.056	.0122

* ⇒ significantly different from .10.

The observed proportions $\hat{\alpha}$ of T^2 exceeding the small sample critical values are close to "advertised" α even though $\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$.

This is mainly because, when $n_1 = n_2$,

$$E[\hat{V}_{pooled}[\overline{\mathbf{x}}_1-\overline{\mathbf{x}}_2]] = E[\hat{V}_{unpooled}[\overline{\mathbf{x}}_1-\overline{\mathbf{x}}_2]] = V[\overline{\mathbf{x}}_1-\overline{\mathbf{x}}_2]$$

I ran a similar simulation with $n_1 = 50$ and $n_2 = 150$ ($n_2 = 3\times n_1$).

Now the two ways to compute T^2, with $\hat{\mathbf{V}}_{pooled} = (1/n_1+1/n_2)\mathbf{S}_{pooled}$ and with $\hat{\mathbf{V}}_{unpooled} = \mathbf{S}_1/n_1+\mathbf{S}_2/n_2$ give different results.

Here are the estimated actual α's.

α	.10	.05	.025	.01
Unpooled $\hat{\alpha}$	.1102*	.0546	.0268	.0112
Pooled $\hat{\alpha}$	.0846†	.0440	.0224	.0100

* $P < .05$, † $P < .01$, H_0: $E[\hat{\alpha}] = \alpha$

Note that, except for $\alpha = .01$, the estimated $\hat{\alpha}$'s when using the biased $\hat{V}_{pooled}$ in computing T^2 are further from intended α than is $\hat{\alpha}$ when using the unbiased $\hat{V}_{unpooled}$.

Paired Hotelling's T^2

In the two-sample situation there is *no* meaningful correspondence between any observation in sample 1 and any observation in sample 2. In the paired case there is a complete correspondence.

Example: Administer a battery of p tests to n subjects *before* a treatment and *after* a treatment. Suppose the outcome is represented by a vector **x** of scores. Data are of the form

$$\mathbf{x}_{11},\mathbf{x}_{21},\ldots,\mathbf{x}_{n1} \text{ and } \mathbf{x}_{12},\mathbf{x}_{22},\ldots,\mathbf{x}_{n2}$$

pre-treatment post-treatment

The first subscript has the same meaning in both samples -- it identifies the subject. That is, there is a *pairing* of observations $\mathbf{x}_{i1} \rightleftarrows \mathbf{x}_{i2}$, all i. The arrows above link paired vectors.

In a paired situation, you should *always* assume that $\mathbf{x}_{i1}$ and $\mathbf{x}_{i2}$ are *not* independent. A two sample test is *not* OK.

That is, you *must not ignore* pairing.

Put $\mathbf{d}_i = \mathbf{x}_{i1} - \mathbf{x}_{i2}$, i = 1,...,n. That is, the $\mathbf{d}_i$'s are the Pre-Post differences.

$$E[\mathbf{d}_i] = \boldsymbol{\mu}_d = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$$

The usual null hypothesis is

$$H_0: \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 = \mathbf{0},$$

that is, H_0: $\boldsymbol{\mu}_d = \mathbf{0}$.

This is a now *single* sample (of $\mathbf{d}_i$'s) problem. ***Hotelling's paired*** T^2 is

$$T^2 = \overline{\mathbf{d}}'(\hat{V}[\overline{\mathbf{d}}])^{-1}\overline{\mathbf{d}} = \overline{\mathbf{d}}'((1/n)\mathbf{S}_d)^{-1}\overline{\mathbf{d}},$$

the 1-sample T^2 based on $\{\mathbf{d}_i\}$. Here,

$$\mathbf{S}_d = (1/(n-1))\sum_{1\le i\le n}(\mathbf{d}_i - \overline{\mathbf{d}})(\mathbf{d}_i - \overline{\mathbf{d}})'.$$

MacAnova: `hotellval(x1 - x2,pval:T)`.

For **small n**, assuming normality of the d_i's, T^2 is distributed (under H_0) as

$$T^2 = (pf_e/(f_e - p + 1))F_{p,f_e-p+1}$$
$$= (p(n - 1)/(n - p))F_{p,n-p},$$

since f_e = n-1 and $f_e - p + 1 = n - p$.

Reversing this, as usual, you get

$$((f_e-p+1)/(pf_e))T^2 = ((n-p)/(p(n-1))T^2 = F_{p,n-p}$$

For both the large- and small-sample distributions, $\{\mathbf{d}_i\}_{1\leq i\leq n}$ must be a *random sample*, that is

- The $\mathbf{d}_i$'s must be mutually independent
- All $\mathbf{d}_i$'s have the same distribution.

When the $\mathbf{x}_1$ and $\mathbf{x}_2$ consist of measurements or observations on individuals randomly selected from a population of individuals, $\{\mathbf{d}_i\}$ is a random sample.

An alternative formulation for paired T^2

Define the combined 2p × 1 vector

$$\mathbf{y} = \begin{bmatrix} \mathbf{x}_1 \\ \\ \mathbf{x}_2 \end{bmatrix}, \text{ with sample } \mathbf{y}_1, \ldots, \mathbf{y}_n$$

- The first p elements $y_1, y_2, \ldots, y_p$ of $\mathbf{y}$ are the "before" scores
- The last p elements $y_{p+1}, y_{p+2}, \ldots, y_{2p}$ are the "after" scores.

Then

$\mathbf{d} = \mathbf{x}_1 - \mathbf{x}_2 = [\mathbf{I}_p, -\mathbf{I}_p]\,\mathbf{y} = \mathbf{C}\,\mathbf{y}$, where

$$\mathbf{C} = [\mathbf{I}_p, -\mathbf{I}_p] = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & -1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 & -1 & 0 & \ldots & 0 \\ \ldots\ldots\ldots & & & \ldots & & \ldots\ldots\ldots\ldots\ldots\ldots & & & & \\ 0 & 0 & 0 & \ldots & 1 & 0 & 0 & 0 & \ldots & -1 \end{bmatrix}$$

- $\mathbf{C} = [\mathbf{I}_p, -\mathbf{I}_p]$ is $p \times 2p$
- Rows of $\mathbf{C}$ define p linear combinations $d_i = y_i - y_{i+p} = x_{i1} - x_{i2}$, $i = 1, \ldots, p$ of $y_1, y_2, \ldots, y_{2p}$, the variables in $\mathbf{y}$.

$\mathbf{d}$ is p by 1 because $\mathbf{C}$ is p by $2p$.

You know a lot about sets of linear combinations:

- $\overline{\mathbf{d}} = \mathbf{C}\overline{\mathbf{y}} = [\mathbf{I}_p, -\mathbf{I}_p]\,\overline{\mathbf{y}} = \overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2$,
- $\mathbf{S}_d = \mathbf{C}\mathbf{S}_y\mathbf{C}' = [\mathbf{I}_p \ \ -\mathbf{I}_p]\ \mathbf{S}_y \begin{bmatrix} \mathbf{I}_p \\ \\ -\mathbf{I}_p \end{bmatrix}$
- The estimated variance of $\overline{\mathbf{d}}$ is

  $\hat{V}[\overline{\mathbf{d}}] = \hat{V}[\mathbf{C}\overline{\mathbf{y}}] = \mathbf{C}\hat{V}[\overline{\mathbf{y}}]\mathbf{C}' = (1/n)\mathbf{C}\mathbf{S}_y\mathbf{C}'$.

  This is exactly $(1/n)\mathbf{S}_d$ but comuted from $\mathbf{S}_y$.

$\mathbf{d}$ is an ***intra**-subject* or *within-subject* comparison where different variables measured on a case are compared.

It is a linear combination of the variables.

This is quite different from an ***inter**-subject* comparison where comparisons are made between different cases or individuals. This idea is fundamental to the analysis of repeated measures data.

A short example with *Iris setosa* data:

```
Cmd> getlabels(setosa,2) # labels for second dimentions
(1) "SepLen"
(2) "SepWid"
(3) "PetLen"
(4) "PetWid"

Cmd> x1 <- setosa[,vector(1,3)] # lengths

Cmd> x2 <- setosa[,vector(2,4)] # widths

Cmd> hotellval(x1 - x2, pval:T)
component: hotelling
(1,1)       4012.1
component: pvalue
(1,1)            0
```

`x1 - x2` is the matrix of differences.

```
Cmd> c <- matrix(vector(1,-1,0,0,  0,0, 1, -1),4)';c
(1,1)            1           -1            0            0
(2,1)            0            0            1           -1
```

This is a different form of **C** because of the way the variables are ordered. It compares sepal lengths with sepal widths, and petal lengths with petal widths. The null hypothesis says something about the shape of the flowers.

```
Cmd> hotellval(setosa %*% c',pval:T) # note the transpose on c
component: hotelling
(1,1)       4012.1      Black box computed T^2
component: pvalue
(1,1)            0

Cmd> s_x <- tabs(setosa,covar:T); xbar <- tabs(setosa,mean:T)

Cmd> vhat_xbar <- s_x/n

Cmd> (c %*% xbar)' %*% solve(c %*% vhat_xbar %*% c') %*% \
        (c %*% xbar)
(1,1)       4012.1      White box computed T^2 is the same
```