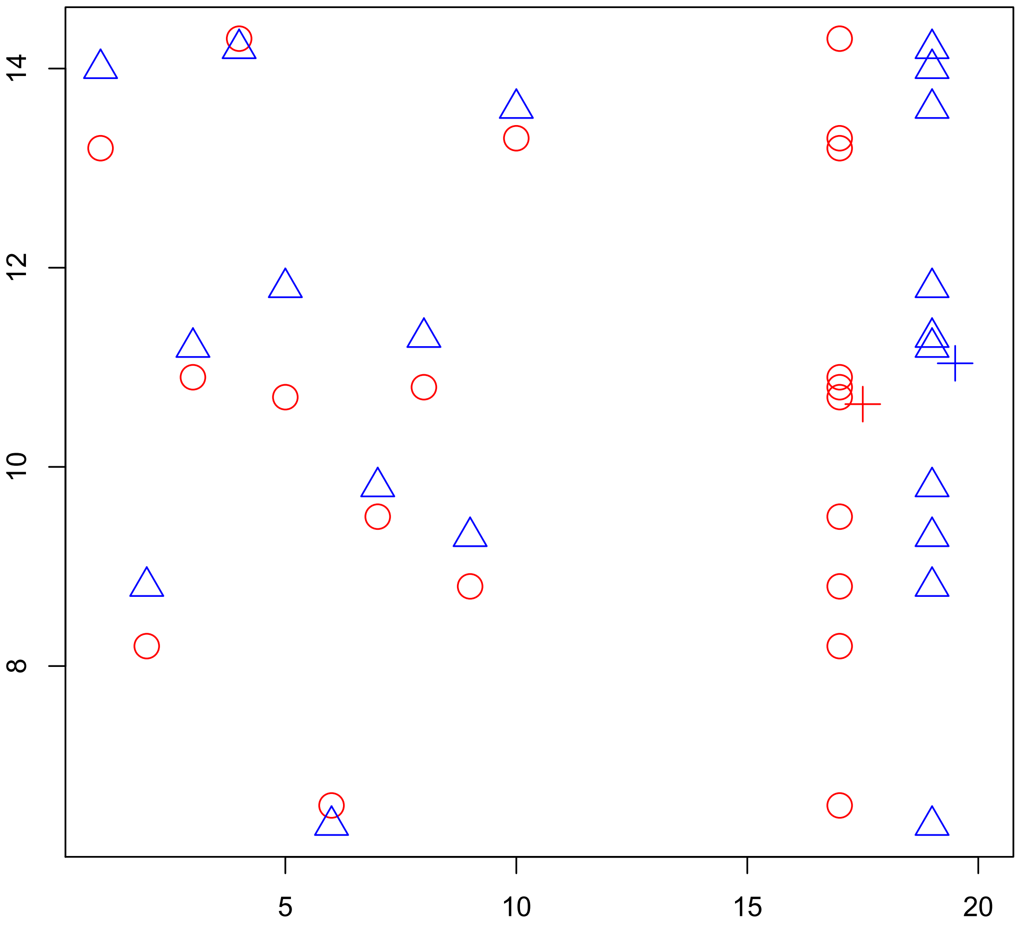
```
> library(MASS)
> shoes
$A
 [1] 13.2  8.2 10.9 14.3 10.7  6.6  9.5 10.8  8.8 13.3

$B
 [1] 14.0  8.8 11.2 14.2 11.8  6.4  9.8 11.3  9.3 13.6
```



```
> t.test(shoes$A, shoes$B, var.equal=TRUE

data:  shoes$A and shoes$B
t = -0.3689, df = 18, p-value = 0.7165

> t.test(shoes$A, shoes$B, paired=TRUE)

data:  shoes$A and shoes$B
t = -3.3489, df = 9, p-value = 0.008539

> diff = shoes$A - shoes$B
[1] -0.8 -0.6 -0.3  0.1 -1.1  0.2 -0.3 -0.5 -0.5 -0.3
> mean(diff)/sqrt(var(diff)/10)
[1] -3.3489
> mean(diff)/sqrt(var(shoes$A)/2 + var(shoes$B)/2)/
sqrt(1/10+1/10)
[1] -0.3689

> y=c(shoes$A, shoes$B);
> boy=as.factor(rep(1:10, 2))
> AB = as.factor(rep(1:2, each=10))
> anova(lm(y~AB+boy))
Analysis of Variance Table

Response: y
          Df  Sum Sq Mean Sq F value    Pr(>F)
AB         1   0.840  0.8405  11.215  0.008539 **
boy        9 110.491 12.2767 163.811 6.871e-09 ***
Residuals  9   0.675  0.0749

> myanova=anova(lm(y~AB+boy))

> sqrt(myanova[1,2]/(myanova[3,2]/9))
[1] 3.3489

> sqrt(myanova[1,2]/((myanova[3,2]+myanova[2,2])/18))
[1] 0.3689
```

```
########################################
# Admission Rate (admission.dat)

# In folklore, the full moon is often portrayed as
# something sinister, a kind of evil force possessing the
# power to control our behavior. Over the centuries, many
# prominent writer and philosophers have shared this belief.
# In scientific community, studies by reputable medical
# researchers have attempted to link the "lunar cycle
# effect" with higher suicide rates and madness. The
# following table shows the admission rates to the
# emergency room of a Virginia mental health clinic before,
# during, and after the 12 full moons from August 1971 to
# July 1972.

#
#  6.4                    5                      5.8
#  7.1                   13                      9.2
#  6.5                   14                      7.9
#  8.6                   12                      7.7
#  8.1                    6                     11
# 10.4                    9                      12.9
# 11.5                   13                      13.5
# 13.8                   16                      13.1
# 15.4                   25                      15.8
# 15.7                   13                      13.3
# 11.7                   14                      12.8
# 15.8                   20                      14.5

> rate=scan("~/… admission.dat")
> rate
 [1]  6.4  5.0  5.8  7.1 13.0  9.2  6.5 14.0  7.9  8.6 12.0
7.7  8.1  6.0
[15] 11.0 10.4  9.0 12.9 11.5 13.0 13.5 13.8 16.0 13.1 15.4
25.0 15.8 15.7
[29] 13.0 13.3 11.7 14.0 12.8 15.8 20.0 14.5

> data=data.frame(rate=rate, Month=rep(1:12,each=3),
time=rep(c("Before", "During" , "After"), rep=12))

> data$Month=as.factor(data$Month)

> data$time=as.factor(data$time)
> anova(lm(rate~time, data=data))

Analysis of Variance Table

Df Sum Sq Mean Sq F value Pr(>F)
time       2  38.60  19.299  1.0921 0.3473
Residuals 33 583.15  17.671

> anova(lm(rate~Month+time, data=data))

Analysis of Variance Table

Df Sum Sq Mean Sq F value    Pr(>F)
Month     11 451.08  41.008  6.8310 7.036e-05 ***
time       2  38.60  19.299  3.2147   0.05959 .
Residuals 22 132.07   6.003

> mean.after=mean(data$rate[data$time=='After']);

> mean.before=mean(data$rate[data$time=='Before']);

> mean.during=mean(data$rate[data$time=='During']);

> r1=(mean.after+mean.before)/2 - mean.during

> lmfit=lm(rate~Month+time, data=data)

> t1 = r1/(summary(lmfit)$sigma*sqrt(1/24+1/24+1/12))

> pt(t1, 22)
[1] 0.02161297
```

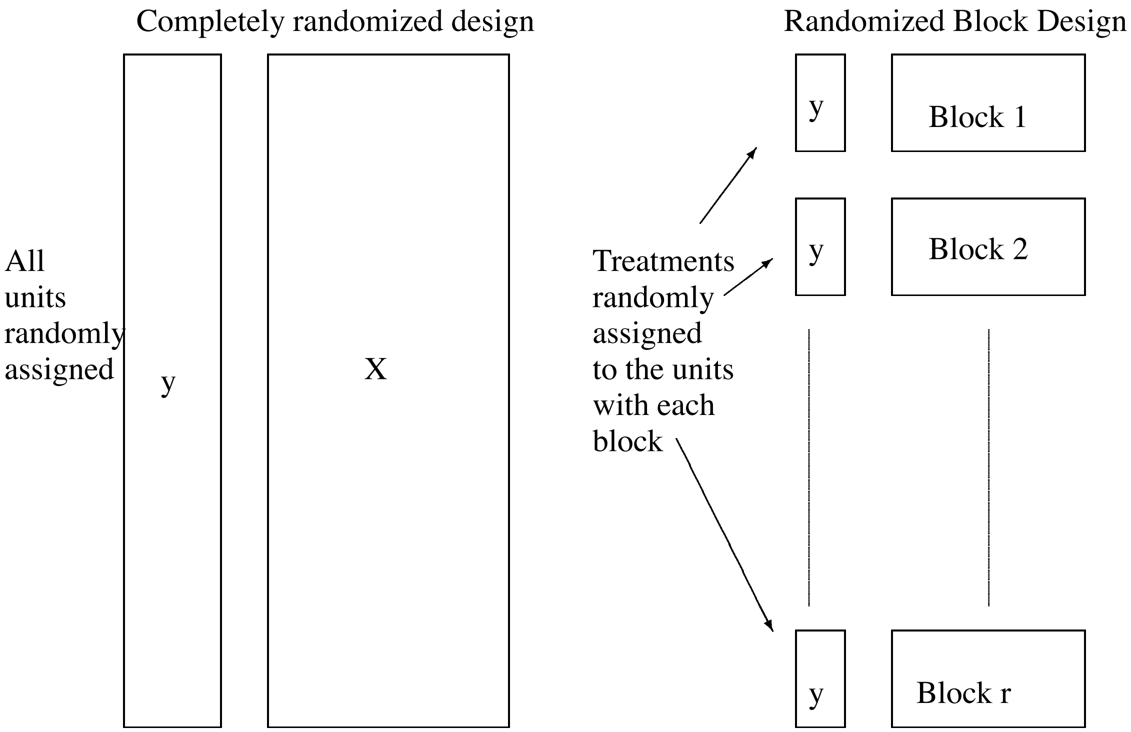

Completely randomized design
Randomized Block Design
All units randomly assigned
y
X
Treatments randomly assigned to the units with each block
y
Block 1
y
Block 2
y
Block r

Figure 16.6: Completely randomized design vs. Randomized Block design

the treatments. This means that we cannot regain the degrees of freedom devoted to blocking even if the blocking effect turns out not to be significant. The randomization test-based argument means that we must judge the magnitude of the treatment effect within the context of the restricted randomization that has been used.

We illustrate with an experiment to compare 4 processes, A,B,C,D for the production of penicillin. These are the treatments. The raw material, corn steep liquor, is quite variable and can only be made in blends sufficient for 4 runs. Thus a randomized complete block design is definitely suggested by the nature of the experimental units. The data is:

	A	B	C	D
Blend 1	89	88	97	94
Blend 2	84	77	92	79
Blend 3	81	87	87	85
Blend 4	87	92	89	84
Blend 5	79	81	80	88

We start with some graphical checks:

```
> data(penicillin)
> plot(yield ~ blend+treat,data=penicillin)
```

See the first two panels of Figure 16.3.1

Did you see any problems? Now check for interactions:

```
> interaction.plot(penicillin$treat,penicillin$blend,penicillin$yield)
> interaction.plot(penicillin$blend,penicillin$treat,penicillin$yield)
```

What do you think? It is hard to tell — interaction plots are only suggestive, not definitive.

Regardless, we now fit the model:

```
> g <- lm(yield ~ treat+blend,penicillin)
> anova(g)
Analysis of Variance Table

Response: yield
          Df Sum Sq Mean Sq F value Pr(>F)
treat      3   70.0    23.3    1.24  0.339
blend      4  264.0    66.0    3.50  0.041
Residuals 12  226.0    18.8
```

We see no significant treatment effect but the block effect is, as suspected, significant. The analysis of variance table corresponds to a sequential testing of models, here corresponding to the sequence

```
y ~ 1
y ~ treat
y ~ treat + blend
```

So here the p-value 0.339 corresponds to a comparison of the first two models in this list, while the p-value of 0.041 corresponds to the test comparing the second two. One small point to note is that the denominator in both F-test is the mean square from the full model, here 18.8

Notice that if we change the order of the terms in the ANOVA, it makes no difference because of the orthogonal design:

```
> anova(lm(yield ~ blend+treat,penicillin))
Analysis of Variance Table

Response: yield
          Df Sum Sq Mean Sq F value Pr(>F)
blend      4  264.0    66.0    3.50  0.041
treat      3   70.0    23.3    1.24  0.339
Residuals 12  226.0    18.8
```

By way of comparison, see what happens if we omit the first observation in the dataset — this might happen in practice if this run was lost:

```
> anova(lm(yield ~ blend+treat,penicillin[-1,]))
Analysis of Variance Table

Response: yield
          Df Sum Sq Mean Sq F value Pr(>F)
blend      4  266.5    66.6    3.27  0.054
treat      3   59.7    19.9    0.98  0.439
Residuals 11  224.3    20.4
```

```
> anova(lm(yield ~ treat+blend,penicillin[-1,]))
Analysis of Variance Table

Response: yield
          Df Sum Sq Mean Sq F value Pr(>F)
treat      3   91.8    30.6    1.50  0.269
blend      4  234.4    58.6    2.87  0.075
Residuals 11  224.3    20.4
```

Notice that now the order does matter. If we want to test for a treatment effect, we would prefer the first table since in that version the blocking factor `blend` is already included when we test the treatment factor. Since the blocking factor is an unalterable feature of the chosen design, this is as it should be.

Check the diagnostics:

```
> plot(g$fitted,g$res,xlab="Fitted",ylab="Residuals")
> qqnorm(g$res)
```

And that might be the end of the story except for that worrying interaction effect possibility. We execute the Tukey non-additivity test:

```
> summary(g)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    90.00       2.74   32.79  4.1e-13
treatB          1.00       2.74    0.36   0.7219
treatC          5.00       2.74    1.82   0.0935
treatD          2.00       2.74    0.73   0.4802
blendBlend2    -9.00       3.07   -2.93   0.0125
blendBlend3    -7.00       3.07   -2.28   0.0416
blendBlend4    -4.00       3.07   -1.30   0.2169
blendBlend5   -10.00       3.07   -3.26   0.0068

Residual standard error: 4.34 on 12 degrees of freedom
Multiple R-Squared: 0.596,      Adjusted R-squared: 0.361
F-statistic: 2.53 on 7 and 12 degrees of freedom,       p-value: 0.0754

> alpha <- c(0,g$coef[2:4])
> alpha
       treatB treatC treatD
     0      1      5      2
> beta <- c(0,g$coef[5:8])
> beta
            blendBlend2 blendBlend3 blendBlend4 blendBlend5
          0          -9          -7          -4         -10
> ab <- rep(alpha,5)*rep(beta,rep(4,5))
> h <- update(g,.~.+ab)
> anova(h)
Analysis of Variance Table
```

16.4 Latin Squares

These are useful when there are two blocking variables. For example, in a field used for agricultural experiments, the level of moisture may vary across the field in one direction and the fertility in another. In an industrial experiment, suppose we wish to compare 4 production methods (the treatment) — A, B, C, and D. We have available 4 machines 1, 2, 3, and 4, and 4 operators, I, II, III, IV. A Latin square design is

	1	2	3	4
I	A	B	C	D
II	B	D	A	C
III	C	A	D	B
IV	D	C	B	A

Table 16.2: Latin Square

- Each treatment is assigned to each block once and only once.
- The design and assignment of treatments and blocks should be random.

We use the model

$$y_{ijk} = \mu + \tau_i + \beta_j + \gamma_k + \varepsilon_{ijk} \qquad i, j, k = 1, \ldots, t$$

To test for a treatment effect simply fit a model without the treatment effect and compare using the F-test. The Tukey pairwise CI's are

$$\hat{\tau}_l - \hat{\tau}_m \pm q_{t,(t-1)(t-2)} \hat{\sigma} \sqrt{1/t}$$

- The Latin square can be even more efficient than the RCBD provided that the blocking effects are sizable.
- We need to have both block sizes to be equal to the number of treatments. This may be difficult to achieve. Latin rectangle designs are possible by adjoining latin squares.
- The Latin square can be used for comparing 3 treatment factors. Only t^2 runs are required compared to the t^3 required if all combinations were run. (The downside is that you can't estimate the interactions if they exist). This is an example of a *fractional factorial*.
- The Latin square can be replicated if more runs are available.
- When there are 3 blocking variables, a Graeco-Latin square may be used but these rarely arise in practice.

An engineer wants to compare the qualities of raw materials from four suppliers, A, B, C, D. The raw material is used to produce a component whose breaking strength is measured. It takes an operator a whole day to make one component and there are 4 operators and 4 days on which the experiment will take place. A Latin square design is appropriate here where the operator and the day are the blocking effects.

```
> data(breaking)
> breaking
      y operator  day supplier
```

```
 1  810      op1 day1        B
 2 1080      op1 day2        C
...etc...
15 1025      op4 day3        D
16  900      op4 day4        C
```

We can check the Latin square structure:

```
> matrix(breaking$supplier,4,4)
     [,1] [,2] [,3] [,4]
[1,] "B"  "C"  "D"  "A"
[2,] "C"  "D"  "A"  "B"
[3,] "A"  "B"  "C"  "D"
[4,] "D"  "A"  "B"  "C"
```

Plot the data:

```
> plot(y ~ operator + day + supplier, breaking)
```

Examine the boxplots in Figure 16.4. There appear to be differences in suppliers but not in the two blocking variables. No outlier, skewness or unequal variance is apparent.

Now fit the Latin squares model:

```
> g <- lm(y ~ operator + day + supplier, breaking)
> anova(g)
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value Pr(>F)
operator   3   7662    2554    0.41 0.7510
day        3  17600    5867    0.94 0.4759
supplier   3 371138  123712   19.93 0.0016
Residuals  6  37250    6208
```

Does it make a difference if we change the order of fitting? Let's see:

```
> anova(lm(y ~ day + supplier + operator, breaking))
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value Pr(>F)
day        3  17600    5867    0.94 0.4759
supplier   3 371137  123712   19.93 0.0016
operator   3   7662    2554    0.41 0.7510
Residuals  6  37250    6208
```

They are the same because of the balanced design. We see that there is clear supplier effect but no evidence of an effect due to day or operator.

Now check the diagnostics

```
> plot(g$fit,g$res,xlab="Fitted",ylab="Residuals")
> qqnorm(g$res,ylab="Residuals")
```

I show only the residual-fitted plot which is fine as was the Q-Q plot. Now look at the estimates of the effects:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    667.5       62.3   10.72  3.9e-05
operatorop2    -35.0       55.7   -0.63  0.55302
operatorop3    -58.7       55.7   -1.05  0.33227
operatorop4    -46.2       55.7   -0.83  0.43825
dayday2        -40.0       55.7   -0.72  0.49978
dayday3         40.0       55.7    0.72  0.49978
dayday4        -40.0       55.7   -0.72  0.49978
supplierB      167.5       55.7    3.01  0.02381
supplierC      411.2       55.7    7.38  0.00032
supplierD      291.2       55.7    5.23  0.00196

Residual standard error: 78.8 on 6 degrees of freedom
Multiple R-Squared: 0.914,      Adjusted R-squared: 0.785
F-statistic: 7.09 on 9 and 6 degrees of freedom,        p-value: 0.0135
```

We see that Supplier C looks best followed by D. Is the difference significant though? Which suppliers in general are significantly better than others? We need the Tukey pairwise intervals to help determine this. The width of the bands calculated in the usual manner:

```
> qtukey(0.95,4,6)*55.7/sqrt(2)
[1] 193
```

The width of the interval is 193 - what can we say about differences between suppliers? We can make a handy table of the supplier differences:

```
> scoefs <- c(0,g$coef[8:10])
> outer(scoefs,scoefs,"-")
                  supplierB supplierC supplierD
             0.00   -167.50   -411.25   -291.25
supplierB 167.50       0.00   -243.75   -123.75
supplierC 411.25     243.75      0.00    120.00
supplierD 291.25     123.75   -120.00      0.00
```

We see that the (A,B), (B,D) and (D,C) differences are not significant at the 5% level. Notice that it would not be reasonable to include that A is no different from C by chaining these comparisons together because each comparison is made using a statistical test where doubt exists about the conclusion and not a logical and definite assertion of equality.

If maximizing breaking strength is our aim, we would pick supplier C but if supplier D offered a better price we might have some cause to consider switching to D. The decision would need to be made with cost-quality trade-offs in mind.

How much more (or less) efficient is the Latin square compared to other designs? First compare to the completely randomized design:

```
> gr <- lm(y ~ supplier,breaking)
> (summary(gr)$sig/summary(g)$sig)^2
[1] 0.8391
```

We see that the LS is 16% less efficient than the CRD. Now compare to the blocked designs:

```
> gr <- lm(y ~ supplier+operator,breaking)
> (summary(gr)$sig/summary(g)$sig)^2
[1] 0.98166
> gr <- lm(y ~ supplier+day,breaking)
> (summary(gr)$sig/summary(g)$sig)^2
[1] 0.8038
```

We see that the Latin square turned out to be a bad choice of design because there was very little if any difference between the operators and days but we did not know that until after the experiment! Next time we will know better.

16.5 Balanced Incomplete Block design

For a complete block design, the block size is equal to the number of treatments. When the block size is less than the number of treatments, an incomplete block design must be used. For example, in the penicillin example, suppose 6 production processes were to be compared but each batch of material was only sufficient for four runs.

In an incomplete block design, the treatments and blocks are *not* orthogonal. Some treatment contrasts will not be identifiable from certain block contrasts - this is an example of *confounding*. This means that those treatment contrasts effectively cannot be examined. In a *balanced incomplete block design*, all the pairwise differences are identifiable and have the same standard error. Pairwise differences are more likely to be interesting than other contrasts. Here is an example:

We have 4 treatments (t=4) A,B,C,D and the block size, $k = 3$ and there are $b = 4$ blocks. Therefore, each treatment appears $r = 3$ times in the design. One possible BIB design is

1	A	B	C
2	A	B	D
3	A	C	D
4	B	C	D

Table 16.3: BIB design

Each pair of treatments appears in the same block $\lambda = 2$ times — this feature enables simple pairwise comparison. For a BIB design, we require

$$\begin{aligned} b &\geq t > k \\ rt &= bk = n \\ \lambda(t-1) &= r(k-1) \end{aligned}$$

This last relation holds because the number of pairs in a block is $k(k-1)/2$ so the total number of pairs must be $bk(k-1)/2$. On the other hand the number of treatment pairs is $t(t-1)/2$. The ratio of these two quantities must be λ.