Latin Square

model:

$$Y_{ijl} = \mu + \alpha_i + r_j + c_l + \varepsilon_{ijl}$$

where

$i = 1, \ldots, k$	indexes treatment levels
$j = 1, \ldots, k$	indexes rows
$l = 1, \ldots, k$	indexes columns
$\varepsilon_{ijl} \sim \mathcal{N}(0, \sigma_\varepsilon^2)$	corresponds to plot error, e.g. residual variation

ANOVA table

Source	df	SS
Row	$k-1$	$k\sum_{j=1}^k(\bar{y}_{\cdot j \cdot} - \bar{y}_{\cdots})^2$
Column	$k-1$	$k\sum_{l=1}^k(\bar{y}_{\cdot\cdot l} - \bar{y}_{\cdots})^2$
Treatment	$k-1$	$k\sum_{i=1}^k(\bar{y}_{i\cdot\cdot} - \bar{y}_{\cdots})^2$
Error	$(k-1)(k-2)$	by subtraction
Total	k^2-1	$\sum_{i=1}^k\sum_{j=1}^k\sum_{l=1}^k(y_{ijl} - \bar{y}_{\cdots})^2$

- In a latin square design, there will not be values for every combination of i, j, and l: there will be k^2 values total, not k^3. The means include only the y_{ijl} terms that exist.
- As with block designs, the tests for row or column effects require some care in interpretation. Apart from that, all tests proceed as you expect: you test an item by looking at its MS divided by MSError and compare to F.
- Randomizing a latin square design is not trivial. See Oehlert for discussion of this.
  In R, we can use the `magic` library and its function `rlatin(k)` to generate a random latin square with `k` treatments, `k` rows and `k columns`. An earlier handout had an example with $k = 4$.