

Randomized Block Designs

David M. Roche

February 6, 2024

Two Kinds of Issues

- One issue with the design and analysis of experimental data is at the design stage to defeat as many common sources of false results as possible. Methods that help with this include randomization, controlling treatment assignment, blocking, adequate sample size, and appropriate comparisons.
- The other kind of issue happens when many things are measured at once, as in proteomics, metabolomics, or transcriptomics.

- The design issues are always important, but the multiple readouts add additional complications both to the design stage and the analysis stage.
- Analyzing results from proteomics, metabolomics, etc. may need professional help, both to get the project funded (in the grant application) and to complete the design and analysis.
- We will look into these issues at the end of the quarter, but complete coverage is a course in itself.

Reference Distributions

- If we are comparing a set of results under one condition with a set of results under another, we need a reference distribution to see if the difference (say in means) that we observe would be unusual if only chance were playing a role. We often use a significance test that gives us a p-value, which is the probability that a result this large or larger could have happened by chance.
- One method might be a two-sample t-test that you would have learned about in your previous statistics class, but this is a derived idea, not the basic idea of comparison.

time	method	yield
1	A	89.7
2	A	81.4
3	A	84.5
4	A	84.8
5	A	87.3
6	A	79.7
7	A	85.1
8	A	81.7
9	A	83.7
10	A	84.5
11	B	84.7
12	B	86.1
13	B	83.2
14	B	91.9
15	B	86.3
16	B	79.3
17	B	82.6
18	B	89.1
19	B	83.7
20	B	88.5

- These are the data from Table 3.1 in BHH and show the yield from 10 runs with the standard method (A) and 10 with a possibly improved method (B).
- The “standard analysis” is based on the two-sample t-distribution and depends on “assumptions” that the observations are statistically independent and approximately normally distributed.

```
> yield1 <- read.table("tab0301.dat",header=T)
> t.test(yield~method,data=yield1)
```

Welch Two Sample t-test

```
data: yield by method
t = -0.88158, df = 17.129, p-value = 0.3902
alternative hypothesis: true difference in means between group A and group B
is not equal to 0
95 percent confidence interval:
 -4.409417  1.809417
sample estimates:
mean in group A mean in group B
      84.24      85.54
```

The yield from the new method is higher than the yield from the standard method by 1.30, but not by enough to be “significant”. One possibility is that this is because of serial correlation between successive yields (which turns out to be true).

- This could have been fixed in the design stage by not running all the standard ones first and then the new ones.
- Usually, this could not be fixed after collecting the data, but in this case they had 210 past observations of the standard method.
- Suppose we compute the difference of 10 vs. the next 10, of which the starting index is 1, 2, ..., 191.
- Then this set of 191 differences forms a new reference set that has the same correlation structure as the A/B data set.

File ‘RandBlocks.R’

```
-----  
yield1 <- read.table("tab0301.dat",header=T)  
yield2 <- read.table("tab03B1.dat",header=T)  
  
refdis <- function(){  
  ref1 <- rep(0,191)  
  for (i in 1:191){  
    j <- i+10  
    mn1 <- mean(yield2$obs[i:(i+9)])  
    mn2 <- mean(yield2$obs[j:(j+9)])  
    ref1[i] <- mn2-mn1  
  }  
  return(ref1)  
}  
ref1 <- refdis()  
print(t.test(yield~method,data=yield1)$estimate %*%c(-1,1))  
print(sum(ref1 > 1.30))  
print(sum(ref1 == 1.30))  
print(sum(ref1 < -1.30))  
print(sum(ref1 == -1.30))
```



```
yield1 <- read.table("tab0301.dat",header=T)
yield2 <- read.table("tab03B1.dat",header=T)
ref1 <- refdis()
print(t.test(yield~method,data=yield1)$estimate %*%c(-1,1))
print(sum(ref1 > 1.30))
print(sum(ref1 == 1.30))
print(sum(ref1 < -1.30))
print(sum(ref1 == -1.30))
```

```
-----
> source("RandBlocks.R")
```

```
      [,1]
[1,]  1.3
[1]  9
[1]  0
[1]  7
[1]  0
> 9/191
[1] 0.04712042
> 16/191
[1] 0.08376963
```

The difference observed could be significant, depending on whether the one-sided test was appropriate, but the p-value is in any case much smaller.

- Ordinarily, we cannot construct a reference distribution from a large collection of previous null tests.
- This means that we must anticipate the problem, and forestall it through design.
- Block what you can and randomize to forestall the influences that you cannot block on.
- To compare two things, we have the two-sample t-test and the paired t-test and associated confidence intervals for measured outcomes.
- For counted outcomes we have chi-square tests and tests and intervals for differences of proportions.

Fully Randomized Designs

- In the context of a two-sample tests, a completely randomized design is a way of generating the most important assumptions of the two-sample t-test.
- If I want to measure blood coagulation time in C57BL/6 mice and I want 4 mice in each group with two diets, I can take the labels of the mice and randomly assign each to a diet so that there are 4 with each diet.
- If we can perform 8 assays per day, then I can assign 8 samples from two treatments randomly to the 8 times. We can do this as well with more than two treatments.

Variance Assumptions

- There is a two-sample t-test method that treats the variance in each group as the same, but the method that does not make that assumption is better because it works over a larger range of circumstances.
- If the sample standard deviations of the two groups are s_1 and s_2 , then the usual two-sample t-test has denominator $\sqrt{s_1^2/n_1 + s_2^2/n_2}$ and the equal variance one has denominator $\sqrt{s^2(1/n_1 + 1/n_2)}$ where $s^2 = [(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2]/(n_1 + n_2 - 2)$.

Variance Assumptions

- When there are more than two groups, there is more than one comparison, and if we allowed separate estimates of variance for each group, things would get really complicated. The number of comparisons with k groups is $k(k - 1)/2$ or ∞ depending on assumptions.
- So for more than two groups, we estimate a common variance for the combined within group variability.

Blood Coagulation Time Example

- Table 4.1 shows blood coagulation time for 24 animals, six each on each of four diets. The animals were (conceptually) numbered 1 to 24 and were placed in a random order (called run in the data set) and assigned to diets A, B, C, and D.
- There are many hypotheses we could test. If we let μ_A , μ_B , μ_C , and μ_D be the true mean coagulation time for the four diets, meaning that if we tested 1000 animals each we would get that mean, then one hypothesis concerns if chance alone could explain the different means.

```

> coag <- read.table("tab0401.dat",header=T)
> coag
  run diets y
1   20    A 62
2    2    A 60
3   11    A 63
4   10    A 59
5    5    A 63
6   24    A 59
7   12    B 63
8    9    B 67
9   15    B 71
10  14    B 64
11   4    B 65
12   8    B 66
13  16    C 68
14   7    C 66
15   1    C 71
16  17    C 67
17  13    C 68
18  21    C 68
19  23    D 56
20   3    D 62
21   6    D 60
22  18    D 61
23  22    D 63
24  19    D 64

```

This hypothesis would be that all the means are equal

$$H_0 : \mu_A = \mu_B = \mu_C = \mu_D.$$

This means that not only is the difference of any two means equal to 0, but any linear combination of the means where the coefficients add to 0 (called a *contrast*) is then 0.

$$6\mu_A - 2\mu_B + \mu_C - 5\mu_D = 0$$

We estimate the variance of the values within each group together as a single quantity.

- In effect, we are comparing two models of the data. One is that $y \sim \text{diets}$, where the mean coagulation time depends on the diet and is the same within the diet, and one is $y \sim 1$, where all the animals have the same mean coagulation time, regardless of diet.
- In the first case, we are estimating four means $(\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4)$ and a variance, and in the second case, one mean $\bar{x} = \text{mean}(\bar{x}_i)$ and a variance.
- The random assignment of diets to animals means that any systematic other effect (cage location for example) is averaged out.


```
> coag.lm <- lm(y~diets,data=coag)
> summary(coag.lm)
```

Call:

```
lm(formula = y ~ diets, data = coag)
```

Residuals:

Min	1Q	Median	3Q	Max
-5.00	-1.25	0.00	1.25	5.00

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	6.100e+01	9.661e-01	63.141	< 2e-16	***
dietsB	5.000e+00	1.366e+00	3.660	0.00156	**
dietsC	7.000e+00	1.366e+00	5.123	5.18e-05	***
dietsD	-9.999e-15	1.366e+00	0.000	1.00000	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.366 on 20 degrees of freedom

Multiple R-squared: 0.6706, Adjusted R-squared: 0.6212

F-statistic: 13.57 on 3 and 20 DF, p-value: 4.658e-05

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	6.100e+01	9.661e-01	63.141	< 2e-16	***
dietsB	5.000e+00	1.366e+00	3.660	0.00156	**
dietsC	7.000e+00	1.366e+00	5.123	5.18e-05	***
dietsD	-9.999e-15	1.366e+00	0.000	1.00000	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
> predict(coag.lm,data.frame(diets=c("A","B","C","D")))
 1  2  3  4
61 66 68 61
> with(coag,tapply(y,diets,mean)) #predictions = means in one-way ANOVA
 A  B  C  D
61 66 68 61

61 Intercept = default level A
66 Intercept + dietsB = 61 + 5 #dietsB = mean(B) - mean(A)
68 Intercept + dietsC = 61 + 7 #dietsC = mean(C) - mean(A)
61 Intercept + dietsD = 61 + 0 #dietsD = mean(D) - mean(A)
```

Diets B and C differ in coagulation from diet A. Diet D does not differ from Diet A. Do diets B and C differ?

```
> coag.lm <- lm(y~diets,data=coag)
```

Residual standard error: 2.366 on 20 degrees of freedom

Multiple R-squared: 0.6706, Adjusted R-squared: 0.6212

F-statistic: 13.57 on 3 and 20 DF, p-value: 4.658e-05

```
> anova(coag.lm)
```

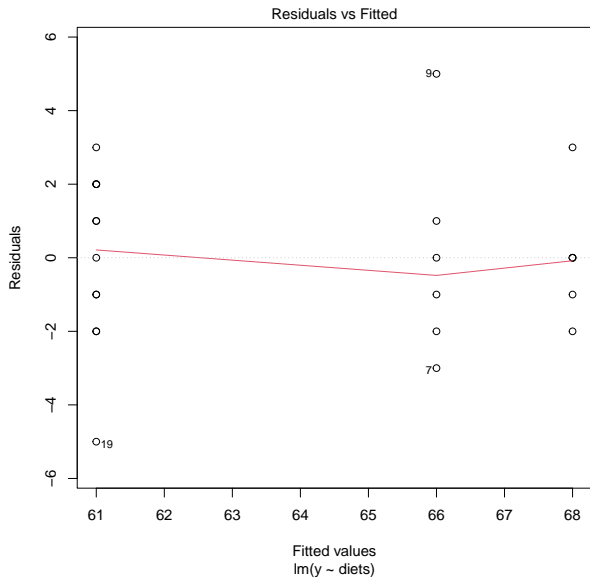
Analysis of Variance Table

Response: y

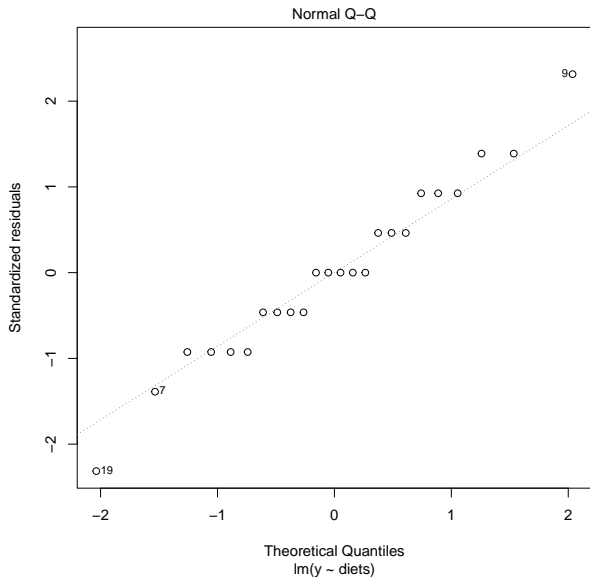
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
diets	3	228	76.0	13.571	4.658e-05 ***
Residuals	20	112	5.6		

Residual standard error is the square root of the combined within-diet variance ($\sqrt{5.6} = 2.366$).

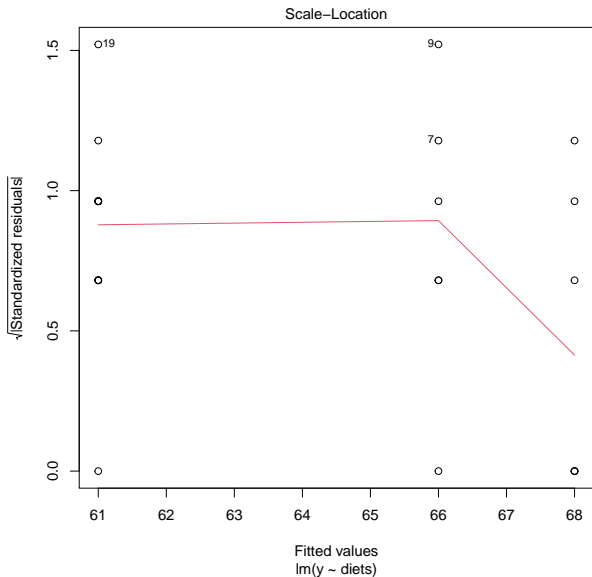
The F-statistic (MS-diets/MS-error) is a test of the hypothesis that all means are equal. The variance of the model without diet is $(128 + 112)/23 = 14.78$ which is much bigger than the within-diets variance of 5.6.



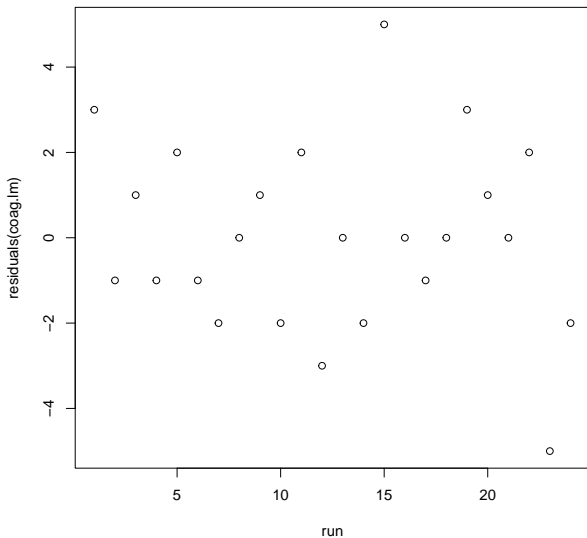
The next 3 slides show what results from `plot(coag.lm)`. From this plot, no outliers, and approximately equal variance.



If the errors were normally distributed, then the points would be near the line. The rows of points result from the book rounding to make things whole numbers.



The residuals from diet B are a little smaller in absolute value, but nothing shocking.



The plot of residuals vs. run order. Nothing problematic.

```
plot(residuals(coag.lm) ~ run,  
data=coag)
```

Means and Differences

- The concept of the mean coagulation time for a diet is less important than that of the differences between diets.
- There are four diets and therefore six differences to consider. Three of these are the coefficients of diet and confidence intervals are attached, so we have data on A vs. B, A vs. C, and A vs. D.
- We can retrieve confidence intervals for the other ones by means of contrasts.

Means and Differences

- There are four levels in the factor diets. The coefficient in the model called the Intercept is the prediction for the default level of diet; i.e., diet A.
- The coefficient labeled `dietsB` represents the difference between diet B and diet A.
- The coefficient labeled `dietsC` represents the difference between diet C and diet A.
- So the difference `dietsB - dietsC` is $(B - A) - (C - A) = B - C$
- The estimated difference between diet C and diet B is the difference of the coefficients.

```

> coef(coag.lm)
  (Intercept)      dietsB      dietsC      dietsD
6.100000e+01  5.000000e+00  7.000000e+00 -9.999406e-15
> vcov(coag.lm)
      (Intercept)      dietsB      dietsC      dietsD
(Intercept)  0.9333333 -0.9333333 -0.9333333 -0.9333333
dietsB       -0.9333333  1.8666667  0.9333333  0.9333333
dietsC       -0.9333333  0.9333333  1.8666667  0.9333333
dietsD       -0.9333333  0.9333333  0.9333333  1.8666667

```

Estimated difference between diet C and diet B is $7 - 5 = 2$. The variance of a difference $x - y$ is $var(x) + var(y) - 2cov(x, y)$.

If the coefficient vector is $b = \text{coef}(\text{coag.lm})$ and the covariance matrix is $V = \text{vcov}(\text{coag.lm})$ and if $c = (0, -1, 1, 0)$ then the difference between diet C and diet B is $c^\top b$ and the variance is $c^\top Vc$.

```
> c1 <- c(0,-1,1,0)
> t(c1) %*% coef(coag.lm)
      [,1]
[1,]      2
> t(c1) %*% vcov(coag.lm) %*% c1
      [,1]
[1,] 1.866667
> sqrt(t(c1) %*% vcov(coag.lm) %*% c1)
      [,1]
[1,] 1.36626
```

```
> summary(coag.lm)
```

Call:

```
lm(formula = y ~ diets, data = coag)
```

Residuals:

Min	1Q	Median	3Q	Max
-5.00	-1.25	0.00	1.25	5.00

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	6.100e+01	9.661e-01	63.141	< 2e-16	***
dietsB	5.000e+00	1.366e+00	3.660	0.00156	**
dietsC	7.000e+00	1.366e+00	5.123	5.18e-05	***
dietsD	-9.999e-15	1.366e+00	0.000	1.00000	

Estimated difference between diet C and diet B is $7 - 5 = 2$. The standard deviation of the difference is 1.36626 from the previous slide. Due to the balance of the design, this is the same as the other three differences.

Randomized Block Designs

- Here we have a factor/variable that is of main concern as well as a second factor (“block”) that is thought to possibly influence the results in a way that might not be reproducible or controllable.
- In the tomato yield experiment from Lecture 1, the tomato plants were in pairs in the same part of the greenhouse, which might influence yield, and then within the pair one plant was assigned at random to each fertilizer. Here the fertilizer is the factor of interest, and the pair of plants/location in the greenhouse are the blocks.

Penicillin Data

- The penicillin data from Table 4.4 propose the investigation of four variants of the process, called A, B, C, and D. In addition, one raw material (corn steep liquor) was found to be variable and also influential of the yield of penicillin. This kind of effect is often found in batches of reagent, particularly biologicals such as antibodies.
- In this instance, a batch of the ingredient, called “blends” here but functioning as “blocks” could accommodate four runs, so each treatment could be run once in each block.
- The run order of A, B, C, and D within the blend (block) was randomized and is given as a separate variable in the data set.

```
> penicillin <- read.table("tab0404.dat",header=T)
```

```
> penicillin
  run blend treatment  y
1    1     1         A 89
2    4     2         A 84
3    2     3         A 81
4    1     4         A 87
5    3     5         A 79
6    3     1         B 88
7    2     2         B 77
8    1     3         B 87
9    3     4         B 92
10   4     5         B 81
11   2     1         C 97
12   3     2         C 92
13   4     3         C 87
14   2     4         C 89
15   1     5         C 80
16   4     1         D 94
17   1     2         D 79
18   3     3         D 85
19   4     4         D 84
20   2     5         D 88
```

Note that the run order is 1–4 within a block, but is not otherwise given. There are five blends and four treatments and each treatment occurs once with each blend. Statistically, there are two factors here which do not differ in calculation but do differ in application. The blend effect is not reproducible blend-to-blend, but the treatment effect, being characteristics of the process is reproducible. Blocks often, but not always, have this character of importance but not reproducibility.

```
> penicillin$blend <- factor(penicillin$blend)
> pen.lm <- lm(y~blend+treatment,data=penicillin)
> summary(pen.lm)
```

Residuals:

Min	1Q	Median	3Q	Max
-5.00	-2.25	-0.50	2.25	6.00

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	90.000	2.745	32.791	4.1e-13	***
blend2	-9.000	3.069	-2.933	0.01254	*
blend3	-7.000	3.069	-2.281	0.04159	*
blend4	-4.000	3.069	-1.304	0.21686	
blend5	-10.000	3.069	-3.259	0.00684	**
treatmentB	1.000	2.745	0.364	0.72194	
treatmentC	5.000	2.745	1.822	0.09351	.
treatmentD	2.000	2.745	0.729	0.48018	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.34 on 12 degrees of freedom

Multiple R-squared: 0.5964, Adjusted R-squared: 0.361

F-statistic: 2.534 on 7 and 12 DF, p-value: 0.07535


```
> drop1(pen.lm,test="F")
```

```
Single term deletions
```

```
Model:
```

```
y ~ blend + treatment
```

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
<none>			226	64.496		
blend	4	264	490	71.973	3.5044	0.04075 *
treatment	3	70	296	63.893	1.2389	0.33866

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> plot(pen.lm,col=penicillin$blend,lwd=2)
```

Blends seem to have a detectable effect, but the evidence that treatments differ is not convincing. The best of them looks to be C, but more testing or other variants of the process may be needed. What is it about blend 1 that led to the high yields? Blend 4 - Blend 5?

```

> coef(pen.lm)
(Intercept)    blend2      blend3      blend4      blend5  treatmentB  treatmentC  treatmentD
          90         -9         -7         -4         -10           1           5           2

> print(round(vcov(pen.lm),3))
      (Intercept) blend2 blend3 blend4 blend5 treatmentB treatmentC treatmentD
(Intercept)    7.533 -4.708 -4.708 -4.708 -4.708    -3.767    -3.767    -3.767
blend2         -4.708  9.417  4.708  4.708  4.708     0.000     0.000     0.000
blend3         -4.708  4.708  9.417  4.708  4.708     0.000     0.000     0.000
blend4         -4.708  4.708  4.708  9.417  4.708     0.000     0.000     0.000
blend5         -4.708  4.708  4.708  4.708  9.417     0.000     0.000     0.000
treatmentB     -3.767  0.000  0.000  0.000  0.000     7.533     3.767     3.767
treatmentC     -3.767  0.000  0.000  0.000  0.000     3.767     7.533     3.767
treatmentD     -3.767  0.000  0.000  0.000  0.000     3.767     3.767     7.533

> c1 <- c(0,0,0,1,-1,0,0,0)    #Contrast for blend 4 - blend 5
> pen.coef <- coef(pen.lm)
> pen.vcov <- vcov(pen.lm)
> t(c1) %*% pen.coef
      6                                #Difference in predicted values (coefficients)
> t(c1) %*% pen.vcov %*% c1
  9.416667
> diff45 <- t(c1) %*% pen.coef
> se45 <- sqrt(t(c1) %*% pen.vcov %*% c1)
  3.068659                                #Standard error of the difference, same as others
> t45 <- diff45/se45
  1.955252                                #t-statistics (12df)
> pt(t45,12,lower=F)
  0.03712245                                #One sided p-value

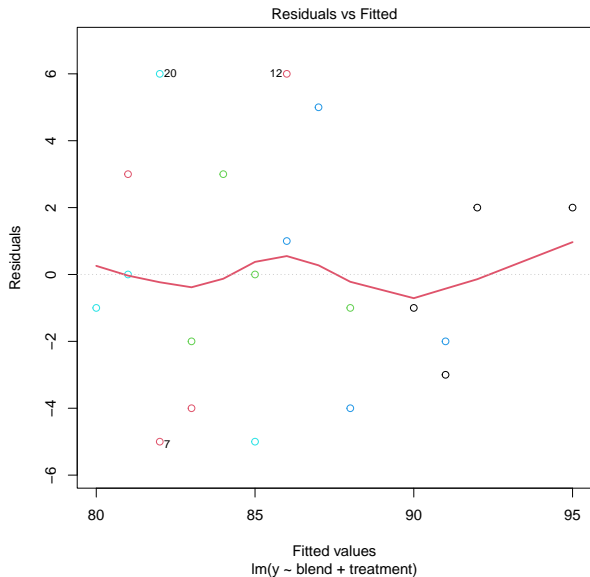
```

Coefficients:

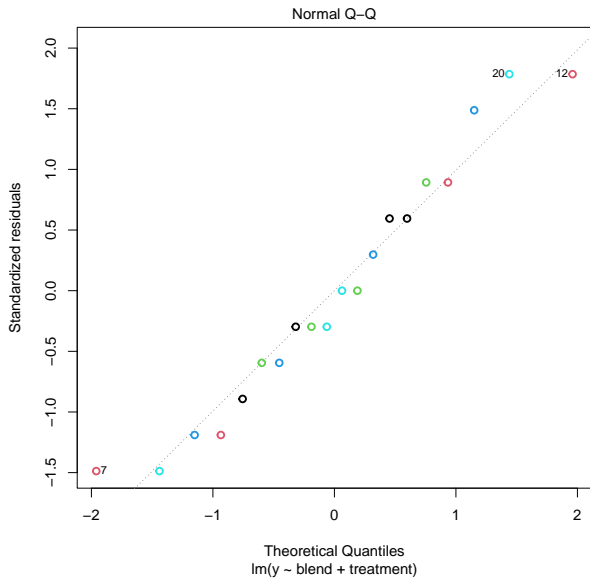
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	90.000	2.745	32.791	4.1e-13	***
blend2	-9.000	3.069	-2.933	0.01254	*
blend3	-7.000	3.069	-2.281	0.04159	*
blend4	-4.000	3.069	-1.304	0.21686	
blend5	-10.000	3.069	-3.259	0.00684	**
treatmentB	1.000	2.745	0.364	0.72194	
treatmentC	5.000	2.745	1.822	0.09351	.
treatmentD	2.000	2.745	0.729	0.48018	

```
> with(penicillin,tapply(y,blend,mean))
  1  2  3  4  5
92 83 85 88 82
> with(penicillin,tapply(y,treatment,mean))
  A  B  C  D
84 85 89 86
```

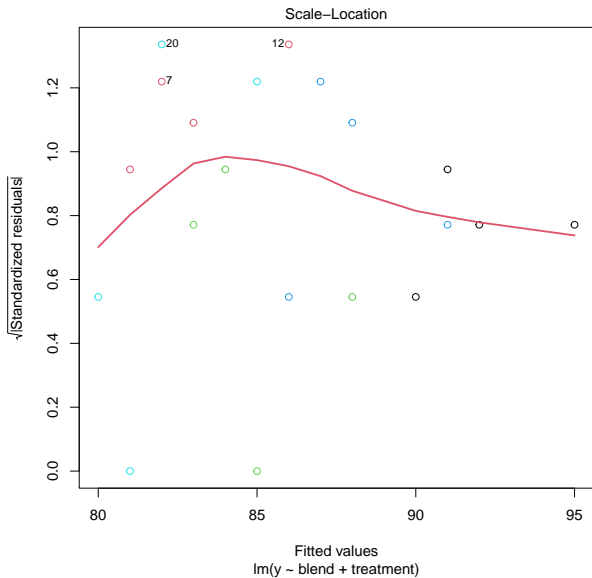
The spacing of the means of blends across treatments and of the treatments across blends is the same as the spacing of the coefficients (because data are balanced). Coefficients work even for unbalanced data.



Blend 1 (black) has four of the five highest predicted yields, and the residuals are closer to the 0 line.



All the large residuals are from Blends 2–5.



Again, blend 1 is less variable.

Interactions

- One possible issue is the presence of interactions
- This would mean that the differences in the treatments might themselves differ among the blends.
- If A is better than B on blend 1, but B is better than A on blend 2, then this complicates the analysis.
- We could ordinarily test for this by fitting a model that contains interaction terms, but this requires replicated data in the cells.
- In the presence of important interaction, we should see some signs of trouble in the plots.

Latin Squares

- Sometimes there is more than one blocking variable that needs to be considered.
- The auto emissions example in Table 4.8 supposes that we are investigating four gasoline additives (A, B, C, D) that might reduce tailpipe emissions.
- In road tests, the driver of the car might matter, as might the physical car itself.
- With four drivers, four cars, and four additives, all possible combinations would require $4 \times 4 \times 4 = 64$ experiments

Latin Squares

- 64 runs might be more expensive than desired, at least for a preliminary trial.
- Sometimes a blocking factor, such as batches of reagent, may have a limited number of runs that it can support.
- For either of both reasons, a smaller set of experiments may be the way to start.

California Reformulated Gasoline

- California regulations on gasoline are stricter than the EPA national regulations.
- These include limits on vapor pressure (for evaporative emissions), sulfur, benzene, total aromatics, olefins, and distillation temperature T50 and T90, among others.
- Refiners could adopt alternative limits only with extensive testing, so CARB developed the predictive model for emission.

California Predictive Model

- The database for the development of the predictive model had tests on over 1000 vehicles, each with multiple fuels from among over 200 different formulations, with the output of the tests (similar to the “smog check”) in emissions of O₃, NO_x (NO and NO₂), CO, SO₂, PM₁₀, and PM_{2.5}.
- Predicted emissions derived from these data (first in the 1990’s) were able to be used by refiners to qualify additional blends as acceptable, and the additional flexibility reduced the extra cost of the tighter gasoline regulations.

```
> emit <- read.table("tab0408.dat",header=T)
```

```
> emit
```

	driver	cars	additive	y
1	1	1	A	19
2	2	1	D	23
3	3	1	B	15
4	4	1	C	19
5	1	2	B	24
6	2	2	C	24
7	3	2	D	14
8	4	2	A	18
9	1	3	D	23
10	2	3	A	19
11	3	3	C	15
12	4	3	B	19
13	1	4	C	26
14	2	4	B	30
15	3	4	A	16
16	4	4	D	16

- Each driver drove each car once for a total of 16 runs.
- Each car was driven with each additive.
- Each driver drove with each additive. (1 with A, B, D, C), (2 with D, C, A, B), etc.
- But a given additive was not tested with all combinations of car and driver (A with D1C1, D4C2, D2C3, and D3C4).
- There are enough data (16) to fit intercept (1 df), drivers (3 df), cars (3 df), and additives (3 df), with 6 df left to estimate the error.

Fuel Additive Assignments for Car/Driver Pairs

	Car1	Car2	Car3	Car4
Driver1	A	B	C	D
Driver2	D	C	A	B
Driver3	B	D	C	A
Driver4	C	A	B	D

Latin Squares

- There are $n \times n$ Latin squares for every value of n .
- The number of distinct Latin squares grows rapidly with n ; for example, there are 576 4×4 Latin squares, 161,280 5×5 Latin squares, and 812,851,200 6×6 Latin squares.
- The function `latin` in the R package `magic` can construct a random Latin square of any size.
- Any Latin square can be used with two blocking factors and one treatment factor, but blocks and treatments must be randomly assigned to the labels for treatments and blocks.

Analysis of Latin Square Designs

- When the Latin Square is set up as a data frame, one can analyze it as a three-factor anova using the command `lm()`.
- If any of the blocks/treatment factors are likely to have an interaction, then this is not a suitable design.
- Use `drop1()` to check significance unless $n = 2$.
- The coefficient values can be used to obtain predictions for each level of the treatment.

```
> emit <- read.table("tab0408.dat",header=T)
> emit$driver <- factor(emit$driver)
> emit$cars <- factor(emit$cars)
> emit
```

	driver	cars	additive	y
1	1	1	A	19
2	2	1	D	23
3	3	1	B	15
4	4	1	C	19
5	1	2	B	24
6	2	2	C	24
7	3	2	D	14
8	4	2	A	18
9	1	3	D	23
10	2	3	A	19
11	3	3	C	15
12	4	3	B	19
13	1	4	C	26
14	2	4	B	30
15	3	4	A	16
16	4	4	D	16

- The driver and cars predictors have to be changed to a factor because they are not numerical.
- The predictor additive is a character variable, but those are automatically changed to factors when used in `lm()`.
- Our model will use three predictors, the two blocking factors driver and cars and the treatment additive.


```
> summary(emit.lm)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-3	-1	0	1	2

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	2.000e+01	1.826e+00	10.954	3.44e-05	***
driver2	1.000e+00	1.633e+00	0.612	0.56276	
driver3	-8.000e+00	1.633e+00	-4.899	0.00271	**
driver4	-5.000e+00	1.633e+00	-3.062	0.02217	*
cars2	1.000e+00	1.633e+00	0.612	0.56276	
cars3	-1.904e-15	1.633e+00	0.000	1.00000	
cars4	3.000e+00	1.633e+00	1.837	0.11584	
additiveB	4.000e+00	1.633e+00	2.449	0.04983	*
additiveC	3.000e+00	1.633e+00	1.837	0.11584	
additiveD	1.000e+00	1.633e+00	0.612	0.56276	

```
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 2.309 on 6 degrees of freedom
```

```
Multiple R-squared:  0.8974,    Adjusted R-squared:  0.7436
```

```
F-statistic: 5.833 on 9 and 6 DF,  p-value: 0.0219
```

Additive predicted values are 20, 24, 23, and 21 respectively. The largest difference is A to B, which is statistically significant. (Intercept = 20, which is the predicted emissions from additive A with driver 1 and car 1).

Fuel Additive Predicted Values for Car/Driver Pairs

	Car1	Car2	Car3	Car4
Driver1	A 20	B 25	C 21	D 26
Driver2	D 22	C 25	A 21	B 28
Driver3	B 16	D 14	C 15	A 15
Driver4	C 18	A 16	B 19	D 19

```

> drop1(emit.lm, test="F")
Single term deletions

Model:
y ~ driver + cars + additive
      Df Sum of Sq RSS      AIC F value    Pr(>F)
<none>          32 31.090
driver    3      216 248 57.853     13.5 0.004466 **
cars      3       24  56 34.044      1.5 0.307174
additive  3       40  72 38.065      2.5 0.156490
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> plot(emit.lm)

```

There is no strong evidence that the additives differ in emissions, but the strong effect of drivers has been eliminated by use of the Latin square design. Some will choose a model based on the AIC, which is $-2 \ln(L) + 2k$ where L is the likelihood (basically the residual SSE) and k is the number of parameters.

```
> drop1(emit.lm, test="F")
```

```
Single term deletions
```

```
Model:
```

```
y ~ driver + cars + additive
```

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
<none>			32	31.090		
driver	3	216	248	57.853	13.5	0.004466 **
cars	3	24	56	34.044	1.5	0.307174
additive	3	40	72	38.065	2.5	0.156490

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> plot(emit.lm)
```

The model with the smallest (best) AIC is the full model!

