Applied Statistical Methods

Tests of Means of Numerical Data- Part IV(random effects ANOVA)

Xuemao Zhang
East Stroudsburg University

March 1, 2023

Outline

- One-Way ANOVA
- Two-Way ANOVA
- ANOVA for Nested Designs
- Mixed Effects Models

Introduction

- Random effects are another approach to designing experiments and modeling data. Random effects are appropriate when the treatments are random samples from a population of potential treatments.
- Example: A company has 50 machines that make cardboard cartons for canned goods, and they want to understand the variation in strength of the cartons. They choose **ten machines at random** from the 50 and make 40 cartons on each machine, assigning 400 lots of cardboard at random to the ten chosen machines.
  - Fixed-effects models are not appropriate.
  - We are trying to learn about and make inferences about the whole population of machines, not just these ten machines that we tested in the experiment.
  - We can learn all we want about these ten machines, but a replication of the experiment will give us an entirely different set of machines.

One-Way ANOVA

- Random effects model

$$Y_{ij} = \mu + \tau_i + \varepsilon_{ij}, j = 1, 2, \ldots, n_i, i = 1, 2, \ldots, k,$$

  where

  - $\varepsilon_{ij} \sim N(0, \sigma^2)$ are independent normal random errors with common variance σ^2 and we also assume that
  - τ_i's are independent normal with mean 0 and variance σ_τ^2, and
  - τ_i's and ε_{ij}'s are independent of each other.

- Testing the treatment effects

$$H_0 : \sigma_\tau^2 = 0$$
$$H_a : \sigma_\tau^2 > 0$$

One-Way ANOVA

- $var(Y_{ij}) = \sigma^2 + \sigma_\tau^2$.
- The terms σ^2 and σ_τ^2 are called components of variance or **variance components.**
- The covariance between Y_{ij} and Y_{kl} is

$$cov(Y_{ij}, Y_{kl}) = \begin{cases} 0, & i \neq k \\ \sigma^2 + \sigma_\tau^2, & i = k, j = l \\ \sigma_\tau^2 & i = k, j \neq l \end{cases}$$

and thus the correlation between Y_{ij} and Y_{kl} is

$$corr(Y_{ij}, Y_{kl}) = \begin{cases} 0, & i \neq k \\ 1, & i = k, j = l \\ \sigma_\tau^2/(\sigma^2 + \sigma_\tau^2) & i = k, j \neq l \end{cases}$$

One-Way ANOVA

- For example, if there are two measurements for each treatment, then the variance-covariance matrix is

$$\begin{bmatrix} \sigma_\tau^2 + \sigma^2 & \sigma_\tau^2 \\ \sigma_\tau^2 & \sigma_\tau^2 + \sigma^2 \end{bmatrix}$$

- Or correlation matrix

$$\begin{bmatrix} 1 & \frac{\sigma_\tau^2}{\sigma_\tau^2 + \sigma^2} \\ \frac{\sigma_\tau^2}{\sigma_\tau^2 + \sigma^2} & 1 \end{bmatrix}.$$

- We also call $\frac{\sigma_\tau^2}{\sigma_\tau^2 + \sigma^2}$ intraclass correlation (ICC).

One-Way ANOVA

- 1-way ANOVA Table for the Random Effects Model

Table 1: 1-way ANOVA Table

Source	df	SS	MS	F
Treatments	$k-1$	SS_T	$SS_T/(k-1)$	MS_T/MSE
Error	$N-k$	SSE	$SSE/(N-k)$	
Total	$N-1$	SS_{total}		

- $E(MSE) = \sigma^2, E(MS_T) = \sigma^2 + [(N - \sum_{i=1}^{k} n_i^2/N)/(k-1)]\sigma_\tau^2$, where $N = \sum_{i=1}^{k} n_i$.
- For balanced design, $E(MSE) = \sigma^2, E(MS_T) = \sigma^2 + n\sigma_\tau^2$.

One-Way ANOVA

- Example. Suppose we want to obtain a precise measurement of calcium concentration in turnip greens. A single calcium measurement nor even a single turnip leaf is considered sufficient to estimate the population mean calcium concentration, so 4 measurements from each of 4 leaves were obtained. The resulting data are given below.

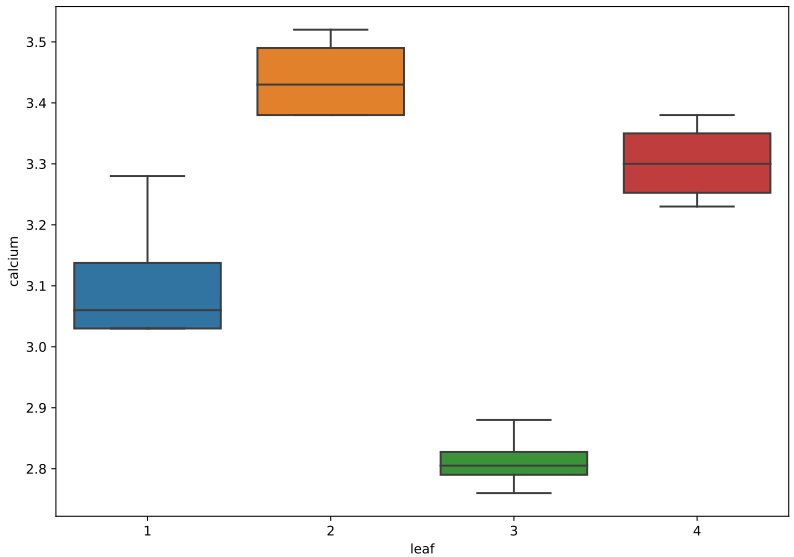
Leaf	Calcium Concentration			
1	3.28	3.09	3.03	3.03
2	3.52	3.48	3.38	3.38
3	2.88	2.80	2.81	2.76
4	3.34	3.38	3.23	3.26

- We are not interested in a population level mean for each leaf. Instead, we are interested in the population of all leaves, from which these four leaves may be thought of as a random (or at least representative) sample.

One-Way ANOVA

- Recall ANOVA with fixed effects

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
turnip=pd.DataFrame({"calcium":[3.28,3.09,3.03,3.03,3.52,3.48,3.38,
3.38,2.88,2.80,2.81, 2.76,3.34,3.38,3.23, 3.26],
'leaf':sum([["1"]*4, ["2"]*4, ["3"]*4,["4"]*4],[])} )
sns.boxplot(data=turnip, x="leaf", y='calcium')
plt.show()
```



One-Way ANOVA

- ANOVA table

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
fit1=ols('calcium ~ leaf', data=turnip).fit()
anova_table = sm.stats.anova_lm(fit1, typ=2)
anova_table
```

```
##               sum_sq    df          F        PR(>F)
## leaf        0.888369   3.0  44.85295  8.520464e-07
## Residual    0.079225  12.0       NaN           NaN
```

One-Way ANOVA

```
print(fit1.summary())
```

```
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                calcium   R-squared:                       0.918
## Model:                            OLS   Adj. R-squared:                  0.898
## Method:                 Least Squares   F-statistic:                     44.85
## Date:                Wed, 01 Mar 2023   Prob (F-statistic):           8.52e-07
## Time:                        18:22:00   Log-Likelihood:                 19.761
## No. Observations:                  16   AIC:                            -31.52
## Df Residuals:                      12   BIC:                            -28.43
## Df Model:                           3
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept      3.1075      0.041     76.489      0.000       3.019       3.196
## leaf[T.2]      0.3325      0.057      5.787      0.000       0.207       0.458
## leaf[T.3]     -0.2950      0.057     -5.134      0.000      -0.420      -0.170
## leaf[T.4]      0.1950      0.057      3.394      0.005       0.070       0.320
## ==============================================================================
## Omnibus:                        2.885   Durbin-Watson:                   1.697
## Prob(Omnibus):                  0.236   Jarque-Bera (JB):                1.798
## Skew:                           0.818   Prob(JB):                        0.407
## Kurtosis:                       2.866   Cond. No.                         4.79
## ==============================================================================
```

One-Way ANOVA

- Estimation of the mean parameters
  - https://www.statsmodels.org/dev/generated/statsmodels.regression.linear_model.OLSResults.conf_int.html

```
fit1.conf_int(alpha=0.05)
```

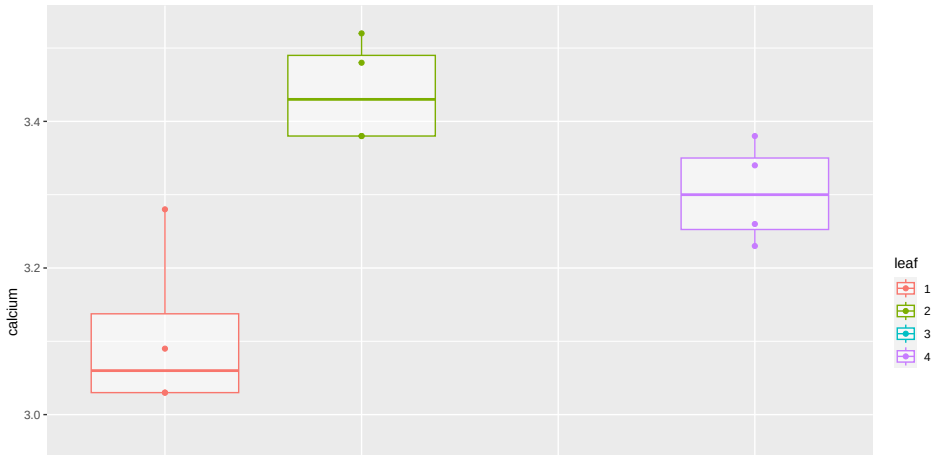
```
##                   0         1
## Intercept  3.018982  3.196018
## leaf[T.2]  0.207317  0.457683
## leaf[T.3] -0.420183 -0.169817
## leaf[T.4]  0.069817  0.320183
```

One-Way ANOVA

- R code

```
calcium=c(3.28,3.09,3.03,3.03,  ## leaf 1
          3.52,3.48,3.38,3.38,  ## leaf 2
          2.88,2.80,2.81, 2.76,  ## leaf 3
          3.34,3.38,3.23, 3.26);  ## leaf 4
leaf=factor(rep(1:4, each = 4))
turnip= data.frame(calcium, leaf)

library(ggplot2)
ggplot(data=turnip, aes(x=leaf, y=calcium,color=leaf))+
geom_boxplot(alpha = 0.5)+ #set transparency
geom_point()
```



One-Way ANOVA

- R code

```
options(contrasts = c("contr.sum", "contr.poly"));
#In contr.sum, the coefficients for each categorical are constrained to add
# contr.poly: Polynomial contrasts
#ANOVA table:
fit.aov= aov(calcium~leaf, data=turnip)
summary(fit.aov)
# Estimation of the mean parameters:
confint(fit.aov)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## leaf          3 0.8884  0.2961   44.85 8.52e-07 ***
## Residuals    12 0.0792  0.0066
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##                   2.5 %      97.5 %
## (Intercept)  3.1213661  3.20988388
## leaf1       -0.1347836  0.01853363
## leaf2        0.1977164  0.35103363
## leaf3       -0.4297836 -0.27646637
```

One-Way ANOVA

```
anova_table['MS']=anova_table['sum_sq']/anova_table['df']
print(anova_table)
```

```
##                sum_sq    df          F        PR(>F)        MS
## leaf         0.888369   3.0  44.85295  8.520464e-07  0.296123
## Residual     0.079225  12.0        NaN           NaN  0.006602
```

- The above ANOVA table is for fixed-effect model
- From the ANOVA table, we get

$$\hat{\sigma^2} = 0.0066 > 0,$$

$$\hat{\sigma_\tau^2} = (MS_T - MSE)/n = (0.2961 - 0.0066)/4 = 0.0724.$$

- Do we reject $H_0 : \sigma_\tau^2 = 0$ and support $H_a : \sigma_\tau^2 > 0$?

One-Way ANOVA

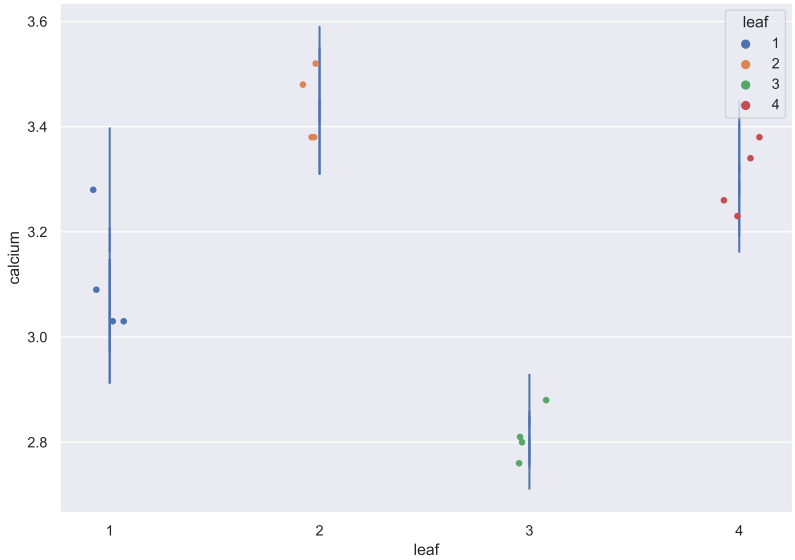
- Let's visualize the data first and check statistical significance later.
  - https://seaborn.pydata.org/generated/seaborn.stripplot.html

```
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
sns.set()
sns.stripplot(data=turnip, x='leaf',y='calcium', jitter=True, hue='leaf')
turnip['y_error']=turnip.groupby(['leaf'])['calcium'].transform(np.std)
plt.errorbar(x=turnip['leaf'],y=turnip['calcium'],
yerr = turnip['y_error'], fmt='none')
plt.show()
```

One-Way ANOVA

- scatter plot with error bars `matplotlib.pyplot.errorbar()`
  - https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.errorbar.html

```
## <ErrorbarContainer object of 3 artists>
```



- R code

```
#The R-package `VCA` is used to perform variance component analysis
library(VCA)
varPlot(form=calcium~leaf, Data=turnip)
```

One-Way ANOVA

- https://www.statsmodels.org/stable/generated/statsmodels.formula.api.mixedlm.html#statsmodels.formula.api.mixedlm
  - To tell the model that a variable is categorical, it needs to be wrapped in `C(independent variable)`.

```
import statsmodels.api as sm
import statsmodels.formula.api as smf
fit1= smf.mixedlm("calcium~leaf", groups="leaf", data=turnip).fit()
```

```
## C:\Users\xzhang2\AppData\Local\Programs\Python\PYTHON~2\lib\site-packages\statsmo
##   warnings.warn(msg, ConvergenceWarning)
## C:\Users\xzhang2\AppData\Local\Programs\Python\PYTHON~2\lib\site-packages\statsmo
##   warnings.warn(msg, ConvergenceWarning)
```

One-Way ANOVA

```
print(fit1.summary())
```

```
##          Mixed Linear Model Regression Results
## ============================================================
## Model:             MixedLM Dependent Variable: calcium
## No. Observations: 16      Method:             REML
## No. Groups:       4       Scale:              0.0066
## Min. group size:  4       Log-Likelihood:     10.3224
## Max. group size:  4       Converged:          Yes
## Mean group size:  4.0
## ------------------------------------------------------------
##             Coef.  Std.Err.    z    P>|z| [0.025 0.975]
## ------------------------------------------------------------
## Intercept    3.107    0.089 34.838 0.000  2.933  3.282
## leaf[T.2]    0.333    0.128  2.592 0.010  0.081  0.584
## leaf[T.3]   -0.295    0.119 -2.474 0.013 -0.529 -0.061
## leaf[T.4]    0.195    0.112  1.742 0.081 -0.024  0.414
## leaf Var     0.007
## ============================================================
```

One-Way ANOVA

- The `leaf Var` ($\hat{\sigma^2}$) is the estimate of the random effects.

```
#fit1.vcomp
#fit1.random_effects
fit1.conf_int()
```

```
##                    0         1
## Intercept   2.932673  3.282327
## leaf[T.2]   0.081040  0.583960
## leaf[T.3]  -0.528723 -0.061277
## leaf[T.4]  -0.024339  0.414339
## leaf Var         NaN       NaN
```

```
fit1.cov_params()
```

```
##                  Intercept     leaf[T.2]     leaf[T.3]     leaf[T.4]       l
## Intercept  7.956465e-03 -8.367770e-03 -7.429994e-03 -9.338449e-03  1.185
## leaf[T.2] -8.367770e-03  1.646042e-02  8.572508e-03  7.830331e-03  4.610
## leaf[T.3] -7.429994e-03  8.572508e-03  1.422018e-02  1.126884e-02 -3.293
## leaf[T.4] -9.338449e-03  7.830331e-03  1.126884e-02  1.252378e-02  4.346
## leaf Var   1.185506e+12  4.610300e+11 -3.293071e+12  4.346854e+12 -4.745
```

One-Way ANOVA

- The above analysis is not complete yet. I found no other Python packages for random-effects model.
- The second option is to directly access the `LMER` packages in R through the rpy2 interface.
  - https://rviews.rstudio.com/2022/05/25/calling-r-from-python-with-rpy2/
- **The remaining lecture slides about analysis with R only are left for your own reading**.

One-Way ANOVA

- R code

```
library(VCA)
fit1= anovaVCA(form=calcium~leaf, Data=turnip)
fit1
```

```
##
##
## Result Variance Component Analysis:
## -----------------------------------
##
##   Name  DF       SS       MS       VC       %Total    SD       CV[%]
## 1 total 3.410917                    0.078982 100       0.281038 8.877801
## 2 leaf  3        0.888369 0.296123 0.07238  91.641059 0.269036 8.49866
## 3 error 12       0.079225 0.006602 0.006602 8.358941  0.081253 2.566735
##
## Mean: 3.165625 (N = 16)
##
## Experimental Design: balanced  |  Method: ANOVA
```

One-Way ANOVA

- The variance of Y_{ij} is estimated by $0.072380 + 0.006602 = 0.078982$. Hence, about $0.072380/0.078982 = 91.64\%$ of the total variance of the calcium is due to leaf (this is the ICC (intraclass correlation)).

```
#Extract the covariance matrix of variance components which is $cov
vcovVC(fit1); #method=c("scm","gb");
```

```
##                  leaf         error
## leaf   3.654153e-03 -1.816146e-06
## error -1.816146e-06  7.264584e-06
## attr(,"method")
## [1] "scm"
```

One-Way ANOVA

- confidence intervals for VC's

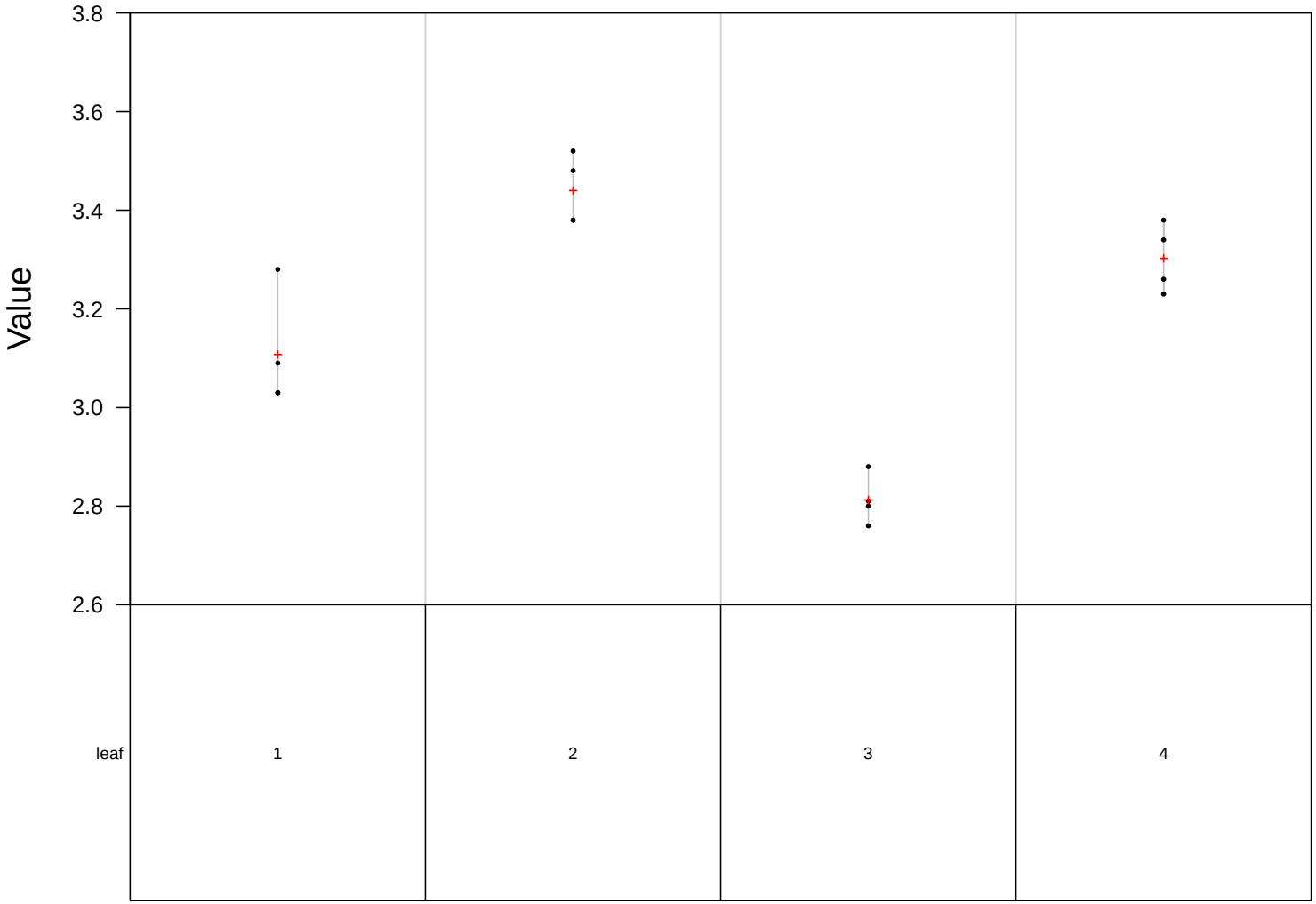
```
ci.fit1=VCAinference(fit1,alpha=0.05,VarVC=TRUE);
ci.fit1$ConfInt;
```

```
## $VC
## $VC$OneSided
##        Name          LCL         UCL
## total total 0.031635397 0.55167516
## leaf   leaf 0.000000000 0.17181094
## error error 0.003767941 0.01515969
##
## $VC$TwoSided
##        Name          LCL         UCL
## total total 0.026668388 0.85468768
## leaf   leaf 0.000000000 0.19085923
## error error 0.003394873 0.01799019
##
##
## $SD
## $SD$OneSided
##        Name         LCL        UCL
## total total 0.17786342 0.7427484
## leaf   leaf 0.00000000 0.4145008
## error error 0.06138356 0.1231247
##
```

One-Way ANOVA

- Visualize the fit:

```
plot(fit1)
```



One-Way ANOVA

- Now we use REML-estimation to estimate the variance components.

```
remlVCA(form=calcium~leaf, Data=turnip, VarVC=TRUE);
```

```
##
##
## Result Variance Component Analysis:
## -----------------------------------
##
##   Name  DF       VC       %Total    SD       CV[%]    Var(VC)
## 1 total 3.410917 0.078982 100       0.281038 8.877801 0.003658
## 2 leaf  2.867364 0.07238  91.641059 0.269036 8.49866  0.003654
## 3 error 12       0.006602 8.358941  0.081253 2.566735 7e-06
##
## Mean: 3.165625 (N = 16)
##
## Experimental Design: balanced  |  Method: REML
```

```
#the variance-covariance matrix of variance components
```

One-Way ANOVA

- Or we use REML-estimation with R-package `lme4`. For large data sets REML-estimation implemented in the VCA-package does not work, the problem size is too large.
- In the model formula `(1|leaf)`: **the random effect is specified after the vertical bar** `|`. All observations sharing the same level of `leaf` will get the same random effect τ_i; The 1 means that we want to have a random intercept per `leaf`.

```
library(lme4);
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'lme4'
```

```
## The following objects are masked from 'package:VCA':
##
##     fixef, getL, ranef
```

```
fit2 <- lmer(calcium~(1|leaf), data = turnip);
```

One-Way ANOVA

```
summary(fit2);
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: calcium ~ (1 | leaf)
##    Data: turnip
##
## REML criterion at convergence: -18.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.9698 -0.6831 -0.2410  0.6091  2.1070
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  leaf     (Intercept) 0.072380 0.26904
##  Residual             0.006602 0.08125
## Number of obs: 16, groups:  leaf, 4
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    3.166      0.136   23.27
```

One-Way ANOVA

- Under Fixed effects we find the estimate $\hat{\mu} = 3.166$. It is an estimate for the expected calcium concentration of a randomly selected leaf(randomly selected from the whole population of all leaves).
- we can use `aov()` function and `confint()` to get esimation of the mean parameters. See slide 11.

```
confint(fit.aov)
```

```
##                   2.5 %      97.5 %
## (Intercept)  3.1213661  3.20988388
## leaf1       -0.1347836  0.01853363
## leaf2        0.1977164  0.35103363
## leaf3       -0.4297836 -0.27646637
```

One-Way ANOVA

- Approximate confidence intervals of variance components can be obtained with the function `confint`.

```
#confint(fit2);
confint(fit2, oldNames = FALSE);
```

```
## Computing profile confidence intervals ...
```

```
##                             2.5 %    97.5 %
## sd_(Intercept)|leaf 0.12756685 0.5764242
## sigma               0.05707956 0.1288825
## (Intercept)         2.86639791 3.4648518
```

- Hence, an approximate confidence interval for σ_τ ($\hat{\sigma_\tau} \approx \sqrt{0.07238} = 0.269$) is given by [0.127566850.5764242]. We see that the estimate is therefore quite wide. The reason is that we only have four leaves to estimate the variance.

One-Way ANOVA

- The `lmerTest` package can be used to test `lmer` model fits

```
library(lmerTest);
ranova(fit2)
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## calcium ~ (1 | leaf)
##             npar  logLik      AIC   LRT Df Pr(>Chisq)
## <none>         3  9.2773 -12.5546
## (1 | leaf)     2 -2.1129   8.2258 22.78  1  1.816e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Two-Way ANOVA

- A factorial design becomes blocking design when there is no replicate in each cell.
- Random effects model:

$$Y_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ijk}, k = 1, 2, \ldots, n, j = 1, 2, \ldots, b, i = 1, 2, \ldots, a,$$

  - where $\varepsilon_{ijk} \sim N(0, \sigma^2)$ are independent normal random errors with common variance σ^2.
  - Both τ_i or β_j are random and are independent of the random errors:
  - τ_i *iid* $\sim N(0, \sigma_\tau^2)$
  - β_j *iid* $\sim N(0, \sigma_\beta^2)$
  - $(\tau\beta)_{ij}$ *iid* $\sim N(0, \sigma_{\tau\beta}^2)$

Two-Way ANOVA

- Testing the treatment effects for both factors

$$H_0 : \sigma_\tau^2 = 0$$
$$H_a : \sigma_\tau^2 > 0$$

and

$$H_0 : \sigma_\beta^2 = 0$$
$$H_a : \sigma_\beta^2 > 0$$

Two-Way ANOVA

- Example 7.1 in Kuehl (2000): A manufacturer was developing a new spectrophotometer for medical labs. A critical issue is consistency of measurements from day to day among different machines. 4 machines were randomly selected from the production process and tested on 4 randomly selected days. Per day 8 serum samples were randomly assigned to the 4 machines (2 samples per machine). Response is the triglyceride level [mg/dl] of a sample.

```
##  machine 1    machine 2   machine 3   machine 4
y=c(142.3,144.0,148.6,146.9,142.9,147.4,133.8,133.2, ## day 1
    134.9,146.3,145.2,146.3,125.9,127.6,108.9,107.5, ## day 2
    148.6,156.5,148.6,153.1,135.5,138.9,132.1,149.7, ## day 3
    152.0,151.4,149.7,152.0,142.9,142.3,141.7,141.2) ## day 4
trigly=data.frame(y=y, day=factor(rep(1:4, each=8)),
                  machine=factor(rep(rep(1:4, each=2), 4)));
str(trigly);
```

```
## 'data.frame':    32 obs. of  3 variables:
##  $ y       : num  142 144 149 147 143 ...
##  $ day     : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 2
##  $ machine: Factor w/ 4 levels "1","2","3","4": 1 1 2 2 3 3 4 4 1
```

Two-Way ANOVA

- verify number of observations

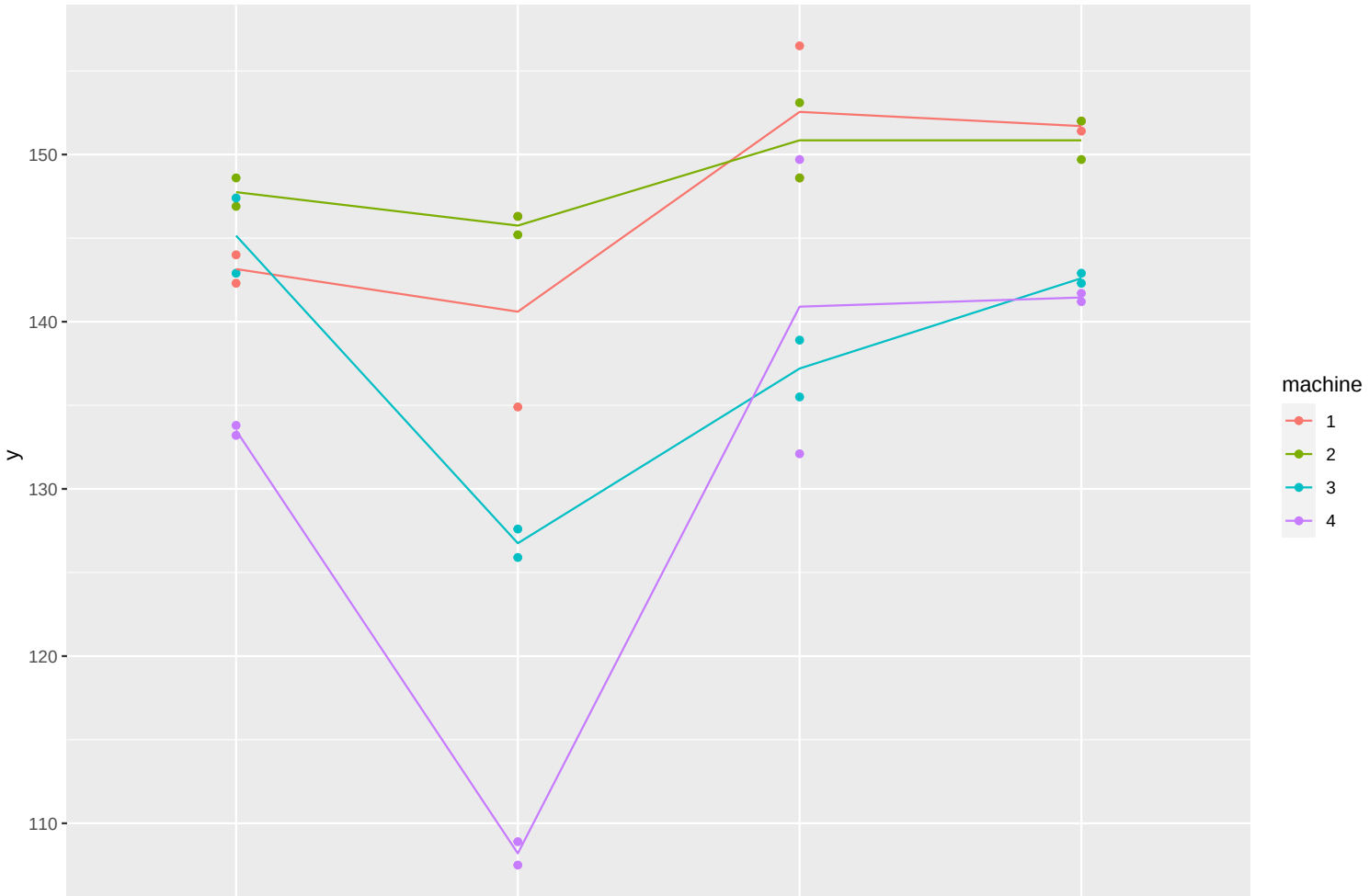
```
xtabs(~ day + machine, data = trigly)
```

```
##    machine
## day 1 2 3 4
##   1 2 2 2 2
##   2 2 2 2 2
##   3 2 2 2 2
##   4 2 2 2 2
```

Two-Way ANOVA

- Interaction plot

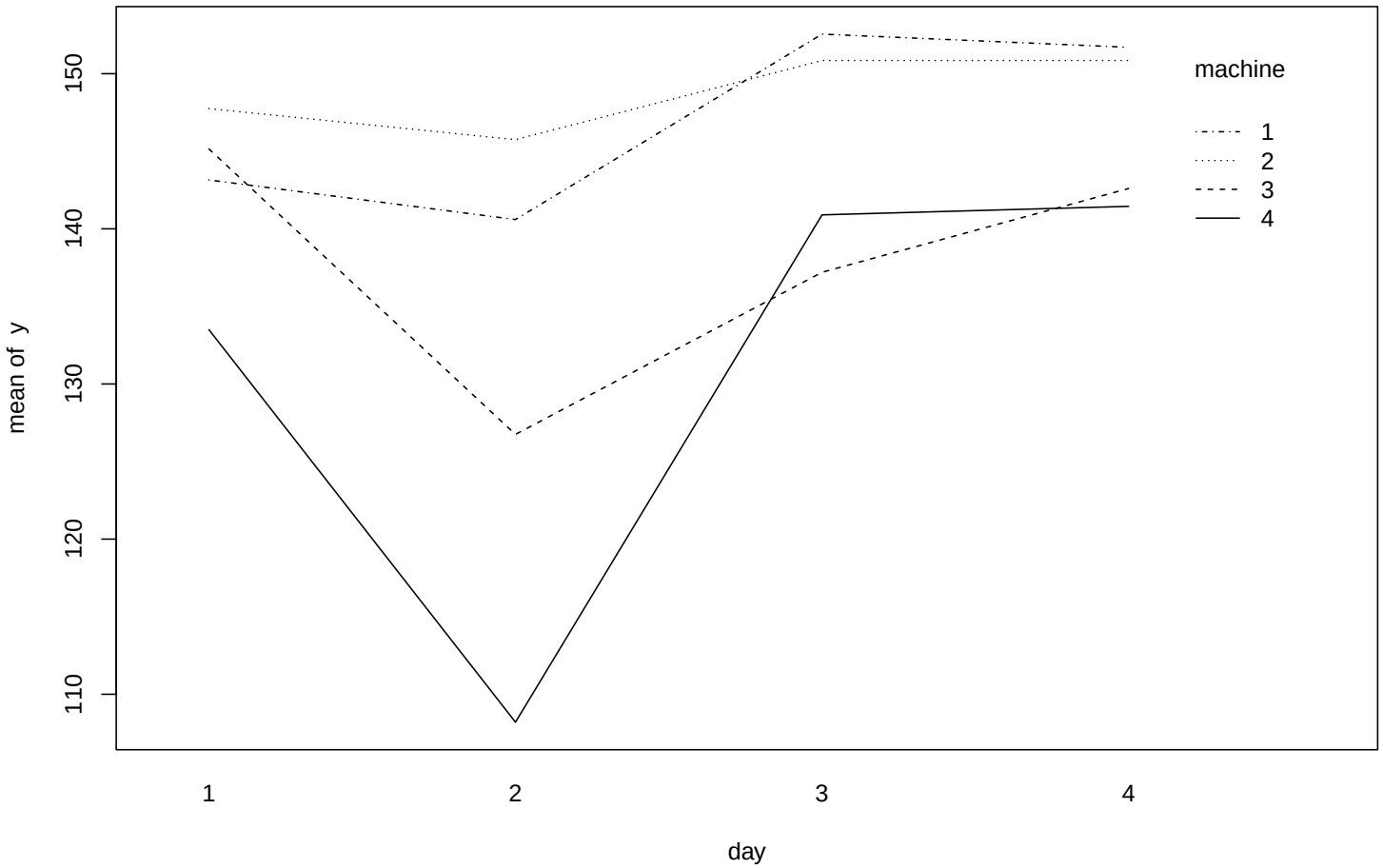
```
library(ggplot2);
ggplot(trigly, aes(x=day, y=y, group=machine, col=machine)) +
  geom_point()+stat_summary(fun= mean, geom = "line");
```



Two-Way ANOVA

- Interaction plot

```
with(trigly, interaction.plot(x.factor = day,
                    trace.factor = machine, response = y));
```



Two-Way ANOVA

- In this example, all effects are random:

$$\tau_i \ iid \sim N(0, \sigma_\tau^2), \quad \beta_j \ iid \sim N(0, \sigma_\beta^2), \quad (\tau\beta)_{ij} \ iid \sim N(0, \sigma_{\tau\beta}^2)$$

- We can use `anovaVCA` function in the `VCA` package to get ANOVA.

```
fit3=anovaVCA(y~day+machine+day*machine, trigly);
#summary( aov(y~day+machine+day*machine, trigly) );
```

Two-Way ANOVA

```
fit3;
```

```
##
##
## Result Variance Component Analysis:
## --------------------------------
##
##   Name        DF       SS           MS          VC          %Total    SD
## 1 total       9.038323                          155.021215 100       12.45
## 2 day         3        1334.463437 444.821146  44.685486  28.8254   6.684
## 3 machine     3        1647.278438 549.092813  57.719444  37.233255 7.597
## 4 day:machine 9        786.035312  87.337257   34.720972  22.397562 5.892
## 5 error       16       286.325     17.895313   17.895313  11.543783 4.230
##   CV[%]
## 1 8.818789
## 2 4.734745
## 3 5.381142
## 4 4.173585
## 5 2.996284
##
## Mean: 141.1844 (N = 32)
##
```

Two-Way ANOVA

- confidence intervals for VC's

```
ci.fit3=VCAinference(fit3,alpha=0.05,VarVC=TRUE);
ci.fit3$ConfInt;
```

```
## $VC
## $VC$OneSided
##                       Name        LCL       UCL
## total                total 82.5526675 418.46133
## day                    day  0.0000000 119.83906
## machine            machine  0.0000000 150.28746
## day:machine day:machine  0.4632065  68.97874
## error                error 10.8884439  35.96304
##
## $VC$TwoSided
##                       Name       LCL       UCL
## total                total 73.43760 514.95438
## day                    day  0.00000 134.23650
## machine            machine  0.00000 168.02104
## day:machine day:machine  0.00000  75.54162
## error                error  9.92621  41.45033
##
##
## $SD
## $SD$OneSided
##                       Name        LCL       UCL
```

Two-Way ANOVA

- Now we use REML-estimation to estimate the variance components.

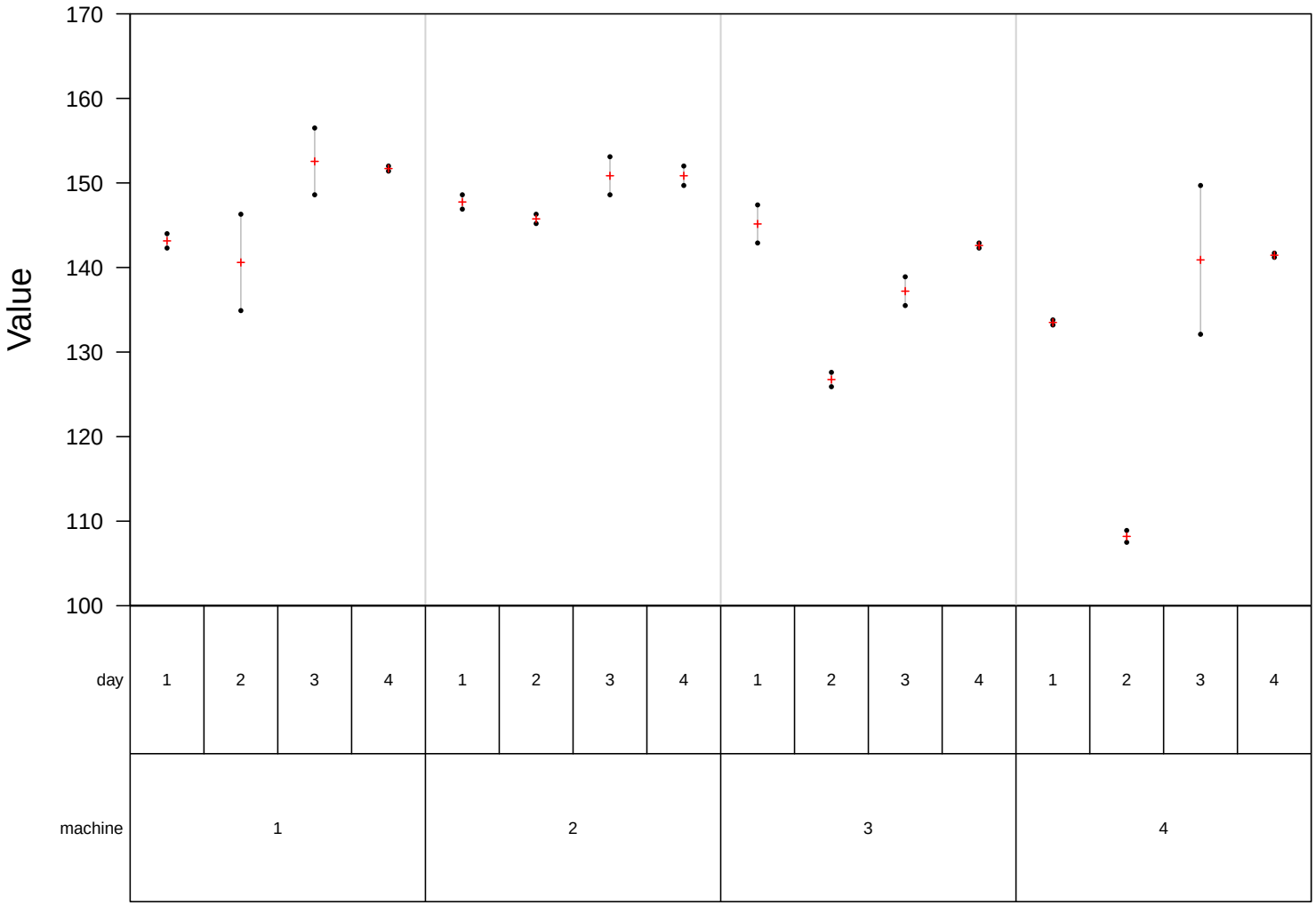
```
remlVCA(form=y~day+machine+day*machine,trigly,VarVC=TRUE);
```

```
##
##
## Result Variance Component Analysis:
## ----------------------------------
##
##   Name         DF       VC          %Total    SD         CV[%]    Var(VC)
## 1 total        9.038323 155.021214  100       12.450752 8.818789 5317.7068
## 2 day          1.913014 44.685495   28.825406 6.684721  4.734746 2087.5886
## 3 machine      2.103812 57.719434   37.233249 7.597331  5.381141 3167.1398
## 4 day:machine  5.558405 34.720972   22.397562 5.89245   4.173585 433.77402
## 5 error        16       17.895313   11.543783 4.230285  2.996284 40.030277
##
## Mean: 141.1844 (N = 32)
##
## Experimental Design: balanced  |  Method: REML
```

Two-Way ANOVA

- Visualize the data

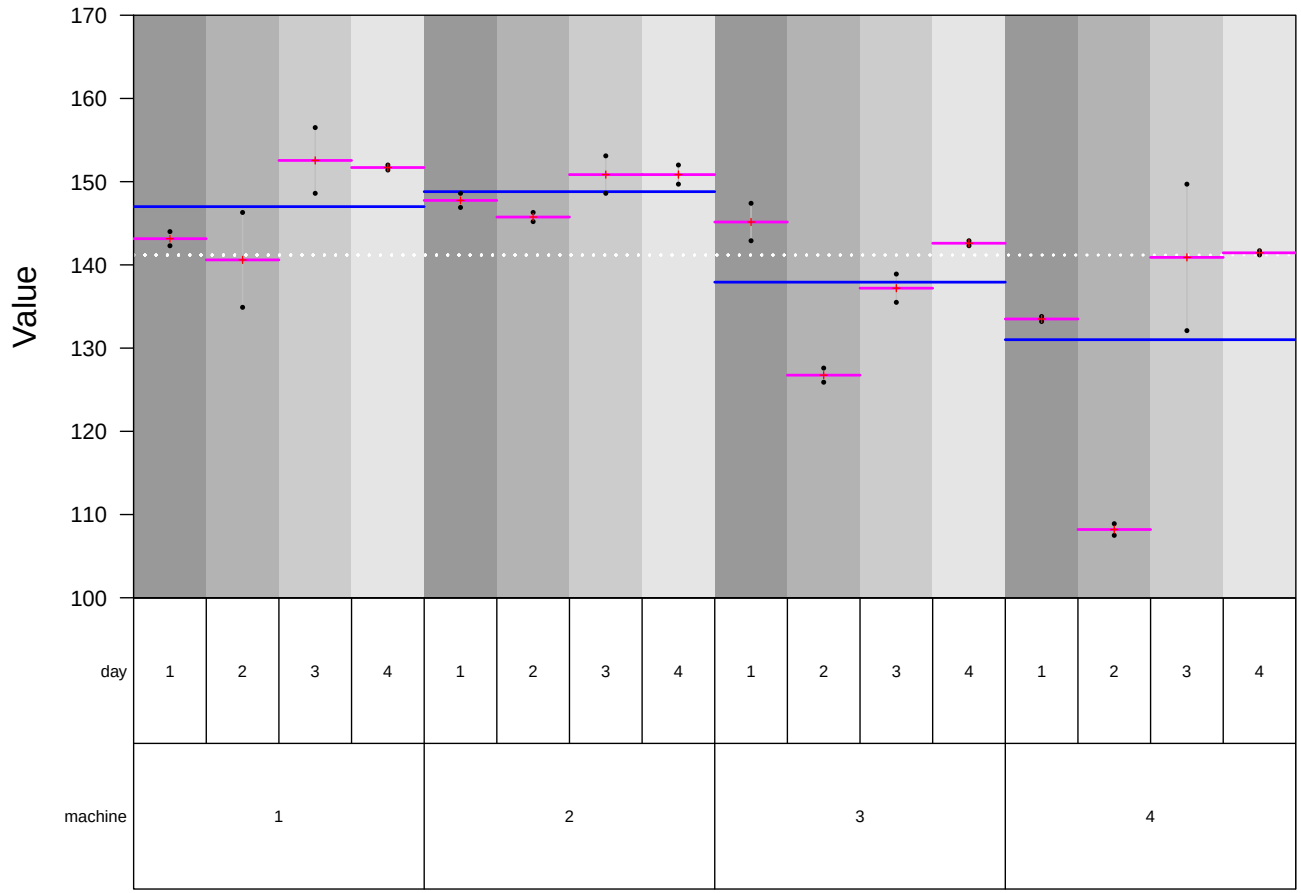
```
varPlot(form=y~machine+day, trigly);
```



Two-Way ANOVA

- Visualize the data

```
varPlot(form=y~machine+day, trigly,
        MeanLine=list(var=c("int", "machine", "day"),
        col=c("white", "blue", "magenta"), lwd=c(2,2,2)),
        BG=list(var="day", col=paste0("gray", c(60,70,80,90))));
```



Two-Way ANOVA

- Or we use the R package `lme4`

```
fit.trigly=lmer(y~(1|day)+(1|machine)+(1|machine:day),
                data = trigly);
summary(fit.trigly);
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: y ~ (1 | day) + (1 | machine) + (1 | machine:day)
##    Data: trigly
##
## REML criterion at convergence: 215
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.84283 -0.35581  0.03485  0.20700  2.31766
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  machine:day (Intercept) 34.72    5.892
##  machine     (Intercept) 57.72    7.597
##  day         (Intercept) 44.69    6.685
```

Two-Way ANOVA

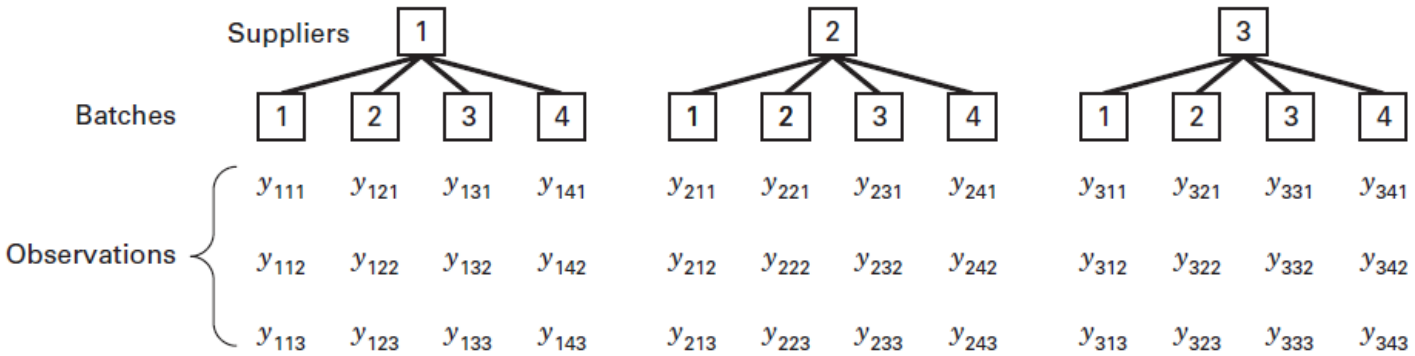
- Hypothesis testing

```
library(lmerTest);
ranova(fit.trigly);
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## y ~ (1 | day) + (1 | machine) + (1 | machine:day)
##                     npar  logLik    AIC    LRT Df Pr(>Chisq)
## <none>                 5 -107.52 225.04
## (1 | day)              4 -109.31 226.61 3.5730  1   0.058727 .
## (1 | machine)          4 -109.81 227.63 4.5924  1   0.032113 *
## (1 | machine:day)      4 -111.31 230.63 7.5879  1   0.005876 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

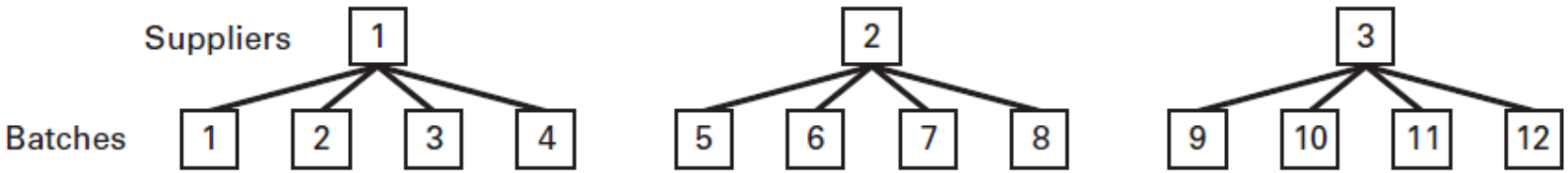
ANOVA for Nested Designs

- In certain multifactor experiments, the levels of one factor (e.g., factor B) are similar but not identical for different levels of another factor (e.g., A). Such an arrangement is called a **nested, or hierarchical**, design, with the levels of factor B nested under the levels of factor A.
- For example, consider a company that purchases its raw material from three different suppliers. The company wishes to determine whether the purity of the raw material is the same from each supplier.
  - There are four batches of raw material available from each supplier, and three determinations of purity are to be taken from each batch.



ANOVA for Nested Designs

- This is a **two-stage nested design**, with batches nested under suppliers.
  - The number of stages can be more than two. See `VCAdata1` in the `VCA` package of a three-stage nested design
- Batch 1 from supplier 1 has no connection with batch 1 from any other supplier, batch 2 from supplier 1 has no connection with batch 2 from any other supplier, and so forth.
- To emphasize the fact that the batches from each supplier are different batches, we may renumber the batches as 1, 2, 3, and 4 from supplier 1; 5, 6, 7, and 8 from supplier 2; and 9, 10, 11, and 12 from supplier 3,



ANOVA for Nested Designs

		Supplier 1				Supplier 2				Supplier 3			
	Batches	1	2	3	4	1	2	3	4	1	2	3	4
		1	−2	−2	1	1	0	−1	0	2	−2	1	3
		−1	−3	0	4	−2	4	0	3	4	0	−1	2
		0	−4	1	0	−3	2	−2	2	0	2	2	1
Batch totals	$y_{ij.}$	0	−9	−1	5	−4	6	−3	5	6	0	2	6
Supplier totals	$y_{i..}$		−5				4				14		

```
##  supplier 1  supplier 2   supplier 3
y=c(1, -1, 0,    1, -2, -3,   2, 4, 0, ## batch 1
    -2, -3, -4,  0, 4, 2,     -2, 0, 2,## batch 2
    -2, 0, 1,    -1, 0, -2,   1, -1, 2,## batch 3
    1, 4, 0,     0, 3, 2,     3, 2, 1);## batch 4
purity=data.frame(y=y, batch=factor(rep(1:4, each=9)),
                  supplier=factor(rep(rep(1:3, each=3), 4)));
str(purity);
```

```
## 'data.frame':    36 obs. of  3 variables:
##  $ y       : num  1 -1 0 1 -2 -3 2 4 0 -2 ...
##  $ batch   : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 2 ...
##  $ supplier: Factor w/ 3 levels "1","2","3": 1 1 1 2 2 2 3 3 3 1 ...
```

ANOVA for Nested Designs

- Random effects model:

$$Y_{ijk} = \mu + \tau_i + \beta_j(i) + \varepsilon_{(ij)k}, k = 1, 2, \ldots, n, j = 1, 2, \ldots, b, i = 1, 2, \ldots, a,$$

  - where The $j(i)$ indicates that the factor corresponding to j (factor B) is nested in the factor corresponding to i (factor A),
  - The random errors ε_{ijk} $iid \sim N(0, \sigma^2)$ are nested in factor A-B combinations.
  - τ_i $iid \sim N(0, \sigma_\tau^2)$
  - $\beta_j(i)$ $iid \sim N(0, \sigma_\beta^2)$

ANOVA for Nested Designs

- ANOVA Table

Analysis of Variance Table for the Two-Stage Nested Design

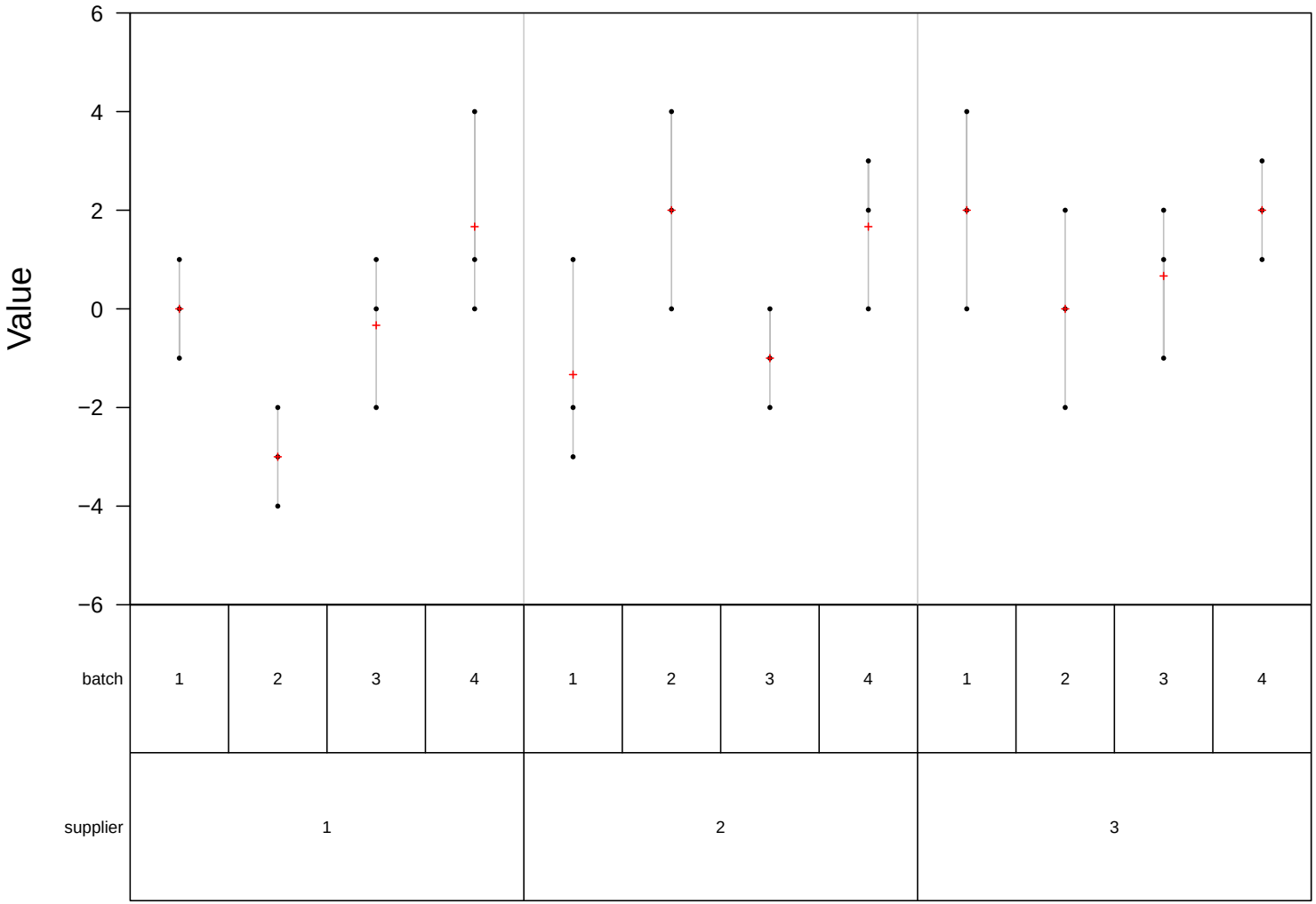
Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square
A	$bn\sum(\bar{y}_{i..}-\bar{y}_{...})^2$	$a-1$	MS_A
B within A	$n\sum\sum(\bar{y}_{ij.}-\bar{y}_{i..})^2$	$a(b-1)$	$MS_{B(A)}$
Error	$\sum\sum\sum(y_{ijk}-\bar{y}_{ij.})^2$	$ab(n-1)$	MS_E
Total	$\sum\sum\sum(y_{ijk}-\bar{y}_{...})^2$	$abn-1$	

- $E(MS_A)=\sigma^2+n\sigma_\beta^2+bn\sigma_\tau^2$
- $E(MS_{B(A)})=\sigma^2+n\sigma_\beta^2$
- $E(MS_E)=\sigma^2$

ANOVA for Nested Designs

- Visualize the data

```
varPlot(form=y~supplier/batch, Data=purity);#batch is nested in supp
```



ANOVA for Nested Designs

```
fit4=anovaVCA(y~supplier/batch, Data=purity);
fit4;
```

```
##
##
## Result Variance Component Analysis:
## -----------------------------------
##
##   Name            DF        SS        MS       VC       %Total    SD
## 1 total           24.962639                    4.348765 100       2.08536
## 2 supplier        2         15.055556 7.527778 0*       0*        0*
## 3 supplier:batch  9         69.916667 7.768519 1.709877 39.318666 1.30762
## 4 error           24        63.333333 2.638889 2.638889 60.681334 1.62446
##   CV[%]
## 1 577.486904
## 2 0*
## 3 362.11084
## 4 449.852047
##
## Mean: 0.361111 (N = 36)
##
## Experimental Design: balanced  |  Method: ANOVA | * VC set to 0 | adapte
```

ANOVA for Nested Designs

- Confidence intervals of VC's

```
ci.fit4=VCAinference(fit4,alpha=0.05,VarVC=TRUE);
ci.fit4$ConfInt;
```

```
## $VC
## $VC$OneSided
##                            Name      LCL      UCL
## total                     total 2.886659 7.444150
## supplier                supplier       NA       NA
## supplier:batch supplier:batch 0.000000 3.760739
## error                     error 1.739209 4.573324
##
## $VC$TwoSided
##                            Name      LCL      UCL
## total                     total 2.673898 8.291344
## supplier                supplier       NA       NA
## supplier:batch supplier:batch 0.000000 4.153630
## error                     error 1.608912 5.107053
##
##
## $SD
## $SD$OneSided
##                            Name      LCL      UCL
## total                     total 1.699017 2.728397
## supplier                supplier       NA       NA
```

ANOVA for Nested Designs

- REML-estimation of the variance components.

```
remlVCA(form=y~supplier/batch, Data=purity,VarVC=TRUE);
```

```
## boundary (singular) fit: see help('isSingular')

##
##
## Result Variance Component Analysis:
## -----------------------------------
##
##   Name            DF        VC       %Total    SD        CV[%]       Var(VC
## 1 total           25.673095 4.334175 100       2.081868 576.517341 1.4634
## 2 supplier        NaN       0        0         0        0           0
## 3 supplier:batch  4.52609   1.695286 39.114392 1.302032 360.56259  1.2699
## 4 error           24        2.638889 60.885608 1.624466 449.852046 0.5803
##
## Mean: 0.361111 (N = 36)
##
## Experimental Design: balanced  |  Method: REML
```

ANOVA for Nested Designs

- REML-estimation using the `lme4` package

```
lmerfit4=lmer(y~(1|supplier/batch), data=purity);
summary(lmerfit4);
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's metho
## lmerModLmerTest]
## Formula: y ~ (1 | supplier/batch)
##    Data: purity
##
## REML criterion at convergence: 148.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.38226 -0.75533 -0.07592  0.57348  1.71092
##
## Random effects:
##  Groups         Name        Variance  Std.Dev.
##  batch:supplier (Intercept) 1.696e+00 1.302285
##  supplier       (Intercept) 9.197e-07 0.000959
##  Residual                   2.639e+00 1.624383
```

ANOVA for Nested Designs

- Hypothesis testing

```
library(lmerTest);
ranova(lmerfit4);
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## y ~ (1 | batch:supplier) + (1 | supplier)
##                         npar  logLik    AIC    LRT Df Pr(>Chisq)
## <none>                     4 -74.343 156.69
## (1 | batch:supplier)       3 -76.503 159.00 4.3186  1     0.0377 *
## (1 | supplier)             3 -74.343 154.69 0.0000  1     1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Mixed Effects Models

- In practice we often encounter models which contain both random and fixed effects. We call them **mixed effects models**.
- Consider the data set `Machines` from the package `nlme`: Data on an experiment to compare three brands of machines used in an industrial process are presented in Milliken and Johnson (p. 285, 1992). Six workers were chosen randomly among the employees of a factory to operate each machine three times. The response is an overall productivity score taking into account the number and quality of components produced."
- We assume that there is a population machine effect (fixed effect, think of an average "profile"), but each worker is allowed to have its own (random) deviation.

Mixed Effects Models

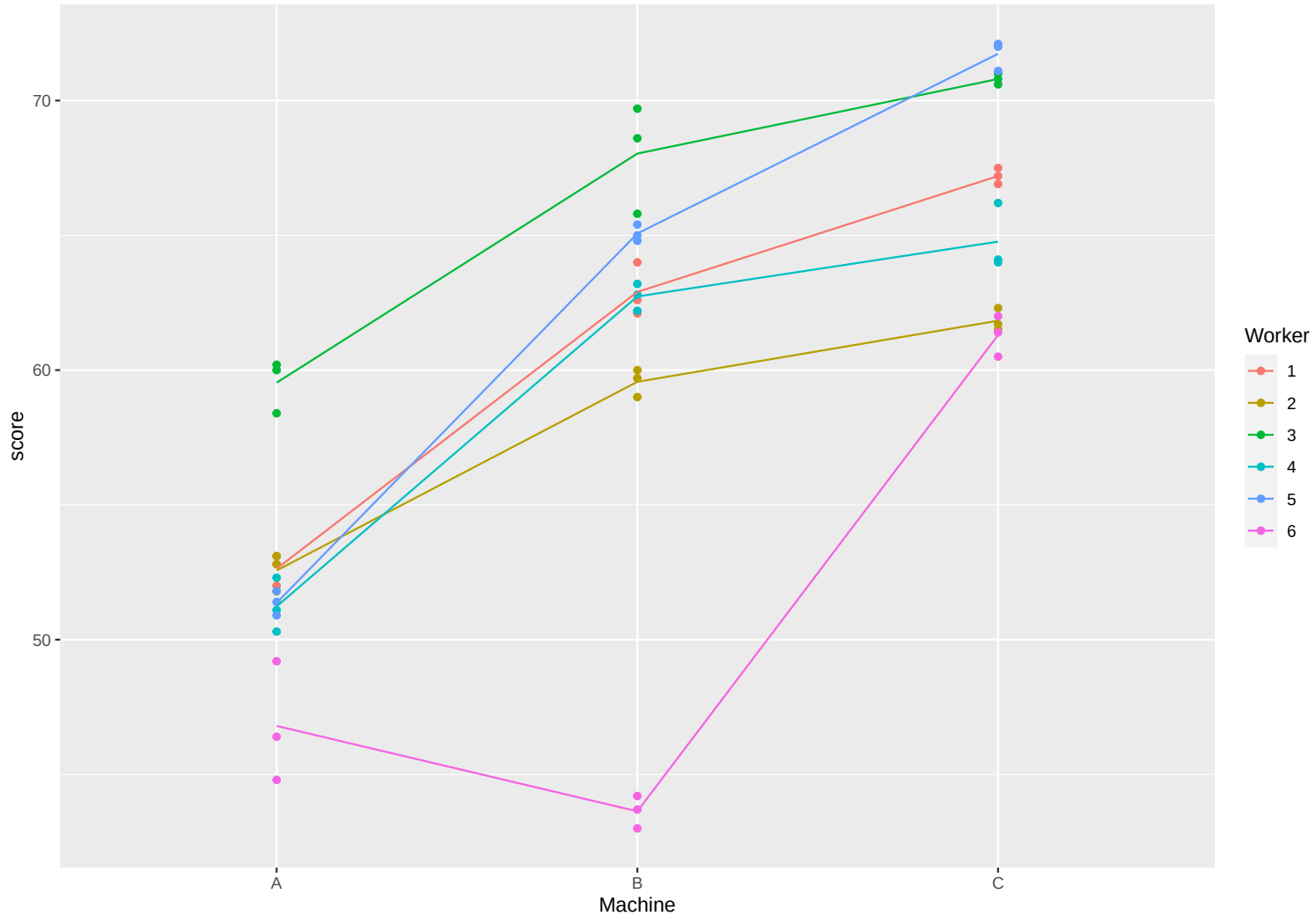
```
data("Machines", package = "nlme");
## technical detail for nicer output:
Machines[, "Worker"] <- factor(Machines[, "Worker"],
levels = 1:6, ordered = FALSE);
str(Machines, give.attr = FALSE); ## give.attr in order to shorten
```

```
## Classes 'nffGroupedData', 'nfGroupedData', 'groupedData' and 'dat
##  $ Worker : Factor w/ 6 levels "1","2","3","4",..: 1 1 1 2 2 2 3
##  $ Machine: Factor w/ 3 levels "A","B","C": 1 1 1 1 1 1 1 1 1 1 .
##  $ score  : num  52 52.8 53.1 51.8 52.8 53.1 60 60.2 58.4 51.1 ..
```

Mixed Effects Models

- Visualize the data

```
ggplot(Machines,aes(x=Machine,y=score,group=Worker,col=Worker))+
  geom_point()+stat_summary(fun=mean, geom="line");
```



Mixed Effects Models

- Consider this Mixed Effects Model:

$$Y_{ijk} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ijk}, k = 1, 2, \ldots, n, j = 1, 2, \ldots, b, i = 1, 2, \ldots, a,$$

  - where $\varepsilon_{ijk} \sim N(0, \sigma^2)$ are independent normal random errors with common variance σ^2.
  - τ_i (for machine) is fixed: $\sum_{i=1}^{a} \tau_i = 0$.
  - β_j *iid* $\sim N(0, \sigma_\beta^2)$ (for worker)
  - $(\tau\beta)_{ij}$ *iid* $\sim N(0, \sigma_{\tau\beta}^2)$

Mixed Effects Models

- In the formula of the anovaMM function, all random terms need to be enclosed by round brackets. Any variable not being bracketed will be considered as fixed.

```
Machines=as.data.frame(Machines)
fit5=anovaMM(score~Machine+(Worker)+(Machine*Worker),  Machines)
fit5;
```

```
##
##
## ANOVA-Type Estimation of Mixed Model:
## ------------------------------------
##
##  [Fixed Effects]
##
##       int  MachineA  MachineB  MachineC
##  66.27222 -13.91667  -5.95000   0.00000
##
##
##  [Variance Components]
##
##   Name           DF       SS        MS       VC         %Total    SD
## 1 total          8.806838                    37.692531 100       6.139424
```

Mixed Effects Models

- variance-covariance matrix of variance components (VC)

```
vcovVC(fit5);
```

```
##                          Worker Machine:Worker          error
## Worker             3.091445e+02   -13.47613636 -6.591486e-18
## Machine:Worker -1.347614e+01    40.43368650 -1.583222e-02
## error              -6.591486e-18    -0.01583222  4.749666e-02
## attr(,"method")
## [1] "scm"
```

Mixed Effects Models

```
fit5.inf=VCAinference(fit5, alpha=0.05, VarVC=TRUE);
fit5.inf$ConfInt;
```

```
## $VC
## $VC$OneSided
##                                 Name        LCL       UCL
## total                          total 19.9391317 103.45121
## Worker                        Worker  0.0000000  51.77909
## Machine:Worker Machine:Worker  3.4502457  24.36867
## error                          error  0.6526994   1.43054
##
## $VC$TwoSided
##                                 Name        LCL        UCL
## total                          total 17.7158035 127.788601
## Worker                        Worker  0.0000000  57.319524
## Machine:Worker Machine:Worker  1.4465381  26.372375
## error                          error  0.6114681   1.560126
##
##
## $SD
```

Mixed Effects Models

- The `fixef.VCA` function extracts fixed effects from a `VCA` Object. Or use the `coef()` function.

```
fixef.VCA(fit5, type = "complex");
```

```
## Note: 'ddfm' not specified, option "satterthwaite" was used!

##              Estimate     Pr > |t|        DF       SE   t Value
## int          66.27222 1.652292e-09  8.521699 2.485830 26.659995
## MachineA    -13.91667 7.906483e-05 10.000000 2.176975 -6.392661
## MachineB     -5.95000 2.107914e-02 10.000000 2.176975 -2.733150
## MachineC      0.00000           NA        NA       NA        NA
```

```
#coef(fit5);
```

Mixed Effects Models

- We could use the `lme4` package to fit the mixed model.
  - a random effect β_j per worker: (1|Worker)
  - a random effect $(\alpha\beta)_{ij}$ per combination of worker and machine: (1|Worker:Machine)

```
library(lme4);
fit6<-lmer(score~Machine+(1|Worker)+(1| Worker:Machine),
           data = Machines);
```

Mixed Effects Models

```
summary(fit6)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's metho
## lmerModLmerTest]
## Formula: score ~ Machine + (1 | Worker) + (1 | Worker:Machine)
##    Data: Machines
##
## REML criterion at convergence: 217.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.26959 -0.54847 -0.01071  0.43937  2.54006
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  Worker:Machine (Intercept) 13.9095  3.7295
##  Worker         (Intercept) 22.8584  4.7811
##  Residual                    0.9246  0.9616
## Number of obs: 54, groups:  Worker:Machine, 18; Worker, 6
##
```

Mixed Effects Models

- Hypothesis testing

```
library(lmerTest);
ranova(fit6);
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## score ~ Machine + (1 | Worker) + (1 | Worker:Machine)
##                        npar  logLik    AIC    LRT Df Pr(>Chisq)
## <none>                    6 -108.94 229.88
## (1 | Worker)              5 -111.73 233.45  5.568  1    0.01829 *
## (1 | Worker:Machine)      5 -144.54 299.07 71.191  1    < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Mixed Effects Models

- See more examples for nested designs: click R-Package VCA for Variance Component Analysis.

License