Practical 6
R – linear models, random factors.

Open the *R* and run the module *Rcmdr*.

1.

Let study the simple experiment: four different sorts are cultivated on different years (2003-2007) to study the differences in crop yield [‘saak’, ‘saagikus’ in Estonian].

Load the *R* dataset:

```
load("http://ph.emu.ee/~ktanel/DK_0007/saagikus.rda")
```

If this command is not working save the datset from internet adress

http://ph.emu.ee/~ktanel/DK_0007/saagikus.rda

and load into the *R Commander* (you may use function `load` or select the command from menus *Data -> Load data set …*).

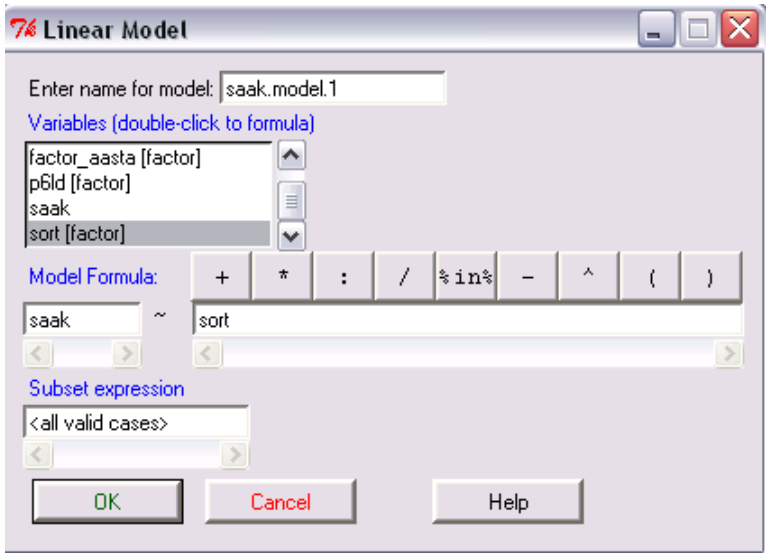
1.1. Estimate the sort effect on yield with simple linear model.

Type (or copy) the following commands into the *R Commander*’s script window:

```
saak.model.1 <- lm(saak ~ sort, data=saagikus)
summary(saak.model.1)
```

Or order the linear models analysis from *R Commander* menus:

Statistics -> Fit models -> Linear model …



Excerpt from results:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3330.3      135.5  24.584  < 2e-16 ***
sortsort2     -556.0      191.6  -2.902  0.00413 **
sortsort3     -471.5      191.6  -2.461  0.01472 *
sortsort4     -152.5      191.6  -0.796  0.42686
```

The average yield of sort 1 is estimated as 3330.3 with standard deviation 135.5. The yields of sorts 2, 3 and 4 are 556.0, 471.5 and 152.5 lower, respectively.

95% confidence interval for sort 1 yield is calculable with command

```
predict(saak.model.1, data.frame(sort="sort1"), interval="confidence")
```

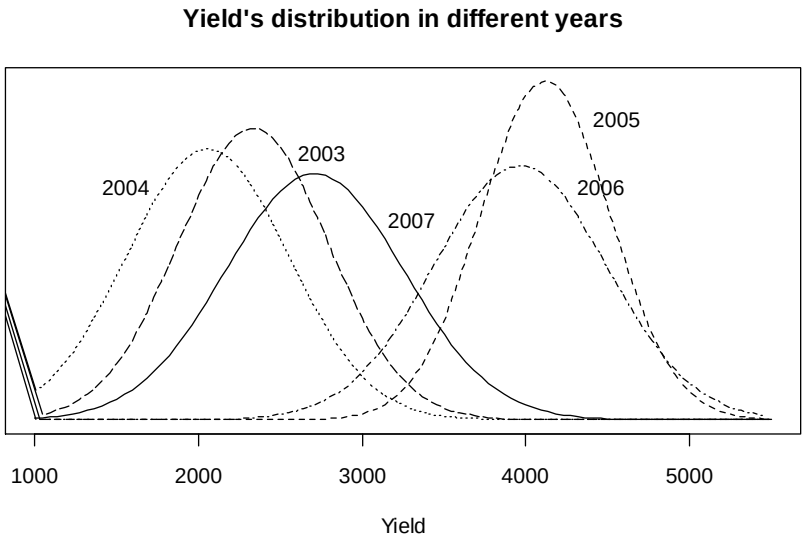
```
          fit      lwr      upr
[1,] 3330.275 3063.121 3597.429
```

95% confidence interval for average yield of sort 1 is 3063.1…3597.4.

1.2.

Question: when are made conclusions about sorts' differences correct?

Answer: if the yield doesn't depend on the year, then the made conclusions apply for all years; if the yield vary in years, then are made conclusions true only for years 2003-2007.



By the way, the diagram describing the yields' distributions in different years (assuming, that the yields are distributed normally) is made with following program.

```
attach(saagikus)
.x <- seq(1000, 5500, length=100)
plot(.x, dnorm(.x, mean=mean(saak[aasta==2005]),sd=sd(saak[aasta==2005])), xlab="Yield",
ylab="", main="Yields' distribution in different years", lty=2, type="l", yaxt="n")
lines(.x, dnorm(.x, mean=mean(saak[aasta==2007]),sd=sd(saak[aasta==2007])),lty=1)
lines(.x, dnorm(.x, mean=mean(saak[aasta==2004]),sd=sd(saak[aasta==2004])),lty=3)
lines(.x, dnorm(.x, mean=mean(saak[aasta==2006]),sd=sd(saak[aasta==2006])),lty=4)
lines(.x, dnorm(.x, mean=mean(saak[aasta==2003]),sd=sd(saak[aasta==2003])),lty=5)
text(2900,0.0008, "2003", adj=c(1, 0.5))
text(1700,0.0007, "2004", adj=c(1, 0.5))
text(4700,0.0009, "2005", adj=c(1, 0.5))
text(4600,0.0007, "2006", adj=c(1, 0.5))
text(3450,0.0006, "2007", adj=c(1, 0.5))
remove(.x)
```

To study more precisely, how big is the difference between years, the complicated model with year effect can be fitted ('factor_aasta'; 'year' = 'aasta' in Estonian):

```
saak.model.2 <- lm(saak ~ sort + factor_aasta, data=saagikus)
summary(saak.model.2)
```

```
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       2627.61      86.55  30.361  < 2e-16 ***
sortsort2         -556.03      86.55  -6.425 1.02e-09 ***
sortsort3         -471.47      86.55  -5.448 1.55e-07 ***
sortsort4         -152.54      86.55  -1.762 0.079578 .
factor_aasta2004  -278.38      96.76  -2.877 0.004470 **
factor_aasta2005  1784.75      96.76  18.445  < 2e-16 ***
factor_aasta2006  1631.55      96.76  16.862  < 2e-16 ***
factor_aasta2007   375.39      96.76   3.880 0.000144 ***
```

The average yield of sort 1 on 2003 is 2627.6. Also we can find, for example, that the average yield of sort 4 on 2007 is 2627.6 – 152.5 + 375.4 = 2850.5.

So we can estimate the average yield for all sorts and years represented in database.

Even though the statistical significance of years' differences implies from the summary output (function `summary`), the additional tests about the statistical significance of overall effects (so called omnibus tests) can be ordered with function `Anova` or selected from *R Commander* menus: *Models -> Hypothesis tests -> ANOVA table…*) :

```
Anova(saak.model.2)
```

```
> Anova(saak.model.2)
Anova Table (Type II tests)

Response: saak
               Sum Sq  Df F value    Pr(>F)
sort         10329848   3  18.388 1.576e-10 ***
factor_aasta 143881439   4 192.091 < 2.2e-16 ***
Residuals    35953332 192
```

Both the sort and year effects are statistically significant.

But if we are interested in predicting the yield for year 2011, then we are in trouble – we can predict the potential yield (for example, as the average yield of studied years), but we haven't the information to estimate the accuracy of got prediction.

1.3.

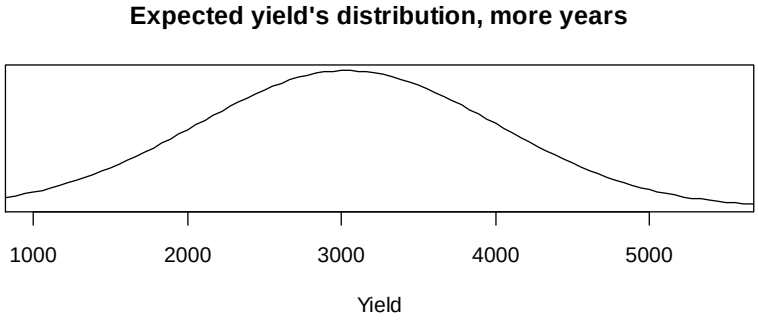
Solution: consider the studied years as a random sample from all possible years = consider the factor year as random factor.

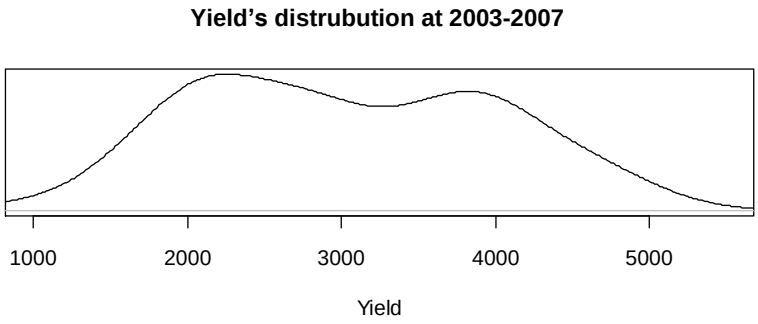
If the year is the random factor, then the year effects A_j are assumed to follow the normal distribution, $A_j \sim N(0, \sigma^2_{aasta})$, this means that

– average year effect is 0 (there is equal number of years better and worse than the average);
– the chance of the next year to be good or bad is random;
– σ_{aasta} is the standard deviation of year ('year' = 'aasta' in Estonian) effects;
as in case of normal distribution ~95% of values are in interval $\pm 2\sigma$, then we can note that the prediction of next year's average yield can vary $\pm 2\sigma_{aasta}$.

The program drawing the digram:

```
par(mfrow=c(2,1))
.y <- seq(500, 6000, length=100)
plot(.y, dnorm(.y,
mean=mean(saak), sd=sd(saak)),
xlim=c(1000,5500), xlab="Yield",
ylab="", main="Expected yield's
distribution, more years",
type="l", yaxt="n")
remove(.y)
plot(density(saagikus$saak),
xlim=c(1000,5500), xlab="Yield",
ylab="", main="Yield's
distrubution at 2003-2007",
yaxt="n")
par(mfrow=c(1,1))
```





To fit such model there are not commands in *R Commander* :(

But there are several different modules with different functions in *R* to fit models with random effects.

For example

```
library(nlme)
saak.model.3 <- lme(saak ~ sort, random=~1|factor_aasta, data=saagikus)
summary(saak.model.3)
```

or

```
library(lme4)
saak.model.4 <- lmer(saak ~ sort + (1|factor_aasta), data=saagikus)
summary(saak.model.4)
```

- The first of them, module `nlme` with function `lme`, is *R*'s classical tool used with mixed models;
- the second module `lme4` with command `lmer`, is new and developing module, which allows to fit mixed models also for non-normal variables and allows easily incorporate more than one random effect.

Both these modules will be installed with *R Commander*.

Still it is necessary to activate these modeles with command `library` to apply them.

- The random factor presentation of the form '`1|factor_aasta`' means, that for each year the different random intercept is estimated (from distribution $A_j \sim N(0, \sigma^2_{aasta})$).

The results are the same in both cases:

```
> library(nlme)
> saak.model.3 <- lme(saak ~ sort, random=~1|factor_aasta, data=saagikus)
> summary(saak.model.3)
Linear mixed-effects model fit by REML
 Data: saagikus
       AIC      BIC    logLik
  2984.390 3004.059 -1486.195

Random effects:
 Formula: ~1 | factor_aasta
        (Intercept) Residual
StdDev:    945.8211 432.7319

Fixed effects: saak ~ sort
                Value Std.Error  DF   t-value p-value
(Intercept) 3330.275  427.3882 192  7.792156  0.0000
sortsort2   -556.026   86.5464 192 -6.424605  0.0000
sortsort3   -471.470   86.5464 192 -5.447595  0.0000
sortsort4   -152.537   86.5464 192 -1.762490  0.0796
```

and

```
> library(lme4)
> saak.model.4 <- lmer(saak ~ sort + (1|factor_aasta), data=saagikus)
> summary(saak.model.4)
Linear mixed-effects model fit by REML
Formula: saak ~ sort + (1 | factor_aasta)
   Data: saagikus
  AIC  BIC logLik MLdeviance REMLdeviance
 2982 2999  -1486       3018         2972
Random effects:
 Groups       Name        Variance Std.Dev.
 factor_aasta (Intercept) 894578   945.82
 Residual                 187257   432.73
number of obs: 200, groups: factor_aasta, 5

Fixed effects:
            Estimate Std. Error t value
(Intercept)  3330.28     427.39   7.792
sortsort2    -556.03      86.55  -6.425
sortsort3    -471.47      86.55  -5.448
sortsort4    -152.54      86.55  -1.762
```

- The average yield of sort 1 is 3330.3, which is similar to this found previously; but the standard error is much bigger, 427.4 (compare with the summary of model `saak.model.1` presented in the end of the 1st page).
The reason is, that now we are trying to model more general situation (instead of 5 fixed years the yield of any years). The 3330.3 is the estimated yield of sort 1 for any years, not only for years 2002-2007.

- Differently from the estimates of fixed year effects the predicted values of random year effects are not printed out by default. The reason is, that considering the year effects as random variables the effects of concrete years representing in the dataset are not interesting and the primarily purpose is to estimate the overall variability of the year effects.

In the circumstances the standard deviation of the random year effects is 945.8 (look at the function `lmer` output in previous page).
Therefore is the crop yield on worser (better) years roughly by 2×945.8 = 1891.6 smaller (higher) than on average years (according to the properties of the normal distribution the ~2,5% of worser years should have by $2\times\sigma_{aasta}$ smaller crop yield compared to the average years, the same applies for better years).
Such a big variability follows from the big differences of observed years.

- Still it is possible to order the predicted values of observed years applying the function

`ranef(saak.model.4)`

(`ranef` = „random effect“) on the model fitted by function `lmer`:

	(Intercept)
2003	-699.0053
2004	-975.9346
2005	1076.4546
2006	924.0540
2007	-325.5687

Comparing the got parameters with the estimates of fixed year effects from the model `saak.model.2` (look at the results of the command `summary` in the beginning of page 3), it follows that the difference between years are slightly decreased.
Considering the year as fixed factor is the difference between years 2005 and 2004
1784.75 – (–278.38) = 2063.13;
but if the year is random factor is the same difference 1076.46 – (–975.93) = 2052.39.
The reason is once again the more general nature of random effects; less attention is assigned to the comparison of concrete years, rather are these differences considered as random – wherefore are the estimated differences between years' effects realised in dataset smaller.

- Applying the function `predict` to the model fitted with function `lme` it is possible to predict the crop yield of sort 1 for optional years (option `level=0`) and for the concrete years appearing in the dataset (`level=1`):

```
predict(saak.model.3,data.frame(sort=c("sort1","sort1"),factor_aasta=2004:2005),level=0:1)
```

	factor_aasta	predict.fixed	predict.factor_aasta
1	2004	3330.275	2354.341
2	2005	3330.275	4406.730

The average crop yield of sort 1 for optional years is 3330.275, the prediction of the year 2004 yield is 3330.275 – 975.9346 = 2354.341 (–975,9346 is the predicted value of the year 2004 effect, look at the output of function `ranef`).

What is the result of the next commands?

```
predict(saak.model.3,
 data.frame(sort=c("sort2","sort2","sort2"),factor_aasta=2004:2006),level=0:1)
predict(saak.model.3,
 data.frame(sort=c("sort1","sort2","sort3"),factor_aasta=2004:2006),level=0:1)
```

1.4.

But field? The experiment was performed on 10 fields (trait 'p6ld'; 'field' = 'põld' in Estonian).
It would be nice to make conclusions not only for these 10 concrete fields …
Then also the field effect must be treated as random:

```
saak.model.5 <- lmer(saak ~ sort + (1|factor_aasta) + (1|p6ld), data=saagikus)
summary(saak.model.5)
```

```
> saak.model.5 <- lmer(saak ~ sort + (1|factor_aasta) + (1|p6ld), data=saagikus)
> summary(saak.model.5)
Linear mixed-effects model fit by REML
Formula: saak ~ sort + (1 | factor_aasta) + (1 | p6ld)
   Data: saagikus
  AIC  BIC logLik MLdeviance REMLdeviance
 2899 2919  -1443       2931         2887
Random effects:
 Groups       Name        Variance Std.Dev.
 p6ld         (Intercept)  87962   296.58
 factor_aasta (Intercept) 896639   946.91
 Residual                 104793   323.72
number of obs: 200, groups: p6ld, 10; factor_aasta, 5

Fixed effects:
            Estimate Std. Error t value
(Intercept)  3330.28     436.14   7.636
sortsort2    -556.03      64.74  -8.588
sortsort3    -471.47      64.74  -7.282
sortsort4    -152.54      64.74  -2.356
```

The ratio of the variance component into the total variance shows the relative importance of the corresponding factor.

The total variance is estimated as

$\sigma^2_{saak} = \sigma^2_{p6ld} + \sigma^2_{aasta} + \sigma^2_{residual} = 87962 + 896639 + 104793 = 1089397$.

The relative importance of the year effect is

$$\frac{\sigma^2_{aasta}}{\sigma^2_{saak}} = \frac{896639}{1089397} = 0.823$$

and the relative importance of field effect is

$$\frac{\sigma^2_{põld}}{\sigma^2_{saak}} = \frac{87962}{1089397} = 0.081.$$

So the year effect is approximately 10 times bigger than the field effect.

If you applied (for example to make a figure in page 2) the command `attach(saagikus)` then now to finish the work with current dataset the command `detach(saagikus)` must be used.

In addition

Actually was the analysed dataset generetad by computer following the given scheme. If you wish you can generate the new dataset applying the following script (the dataset name can be changed, for example to 'saagikus2'). After that you can apply the used models again and try to understand the got results (as the data are generated randomly are the new results little different from the old).

Program generating the data:

```
p6ld <- rep(c("p1","p2","p3","p4","p5","p6","p7","p8","p9","p10"),c(20,20,20,20,20,20,20,20,20,20))
aasta <- rep(c(rep(2003,4),rep(2004,4),rep(2005,4),rep(2006,4),rep(2007,4)),10)
sort <- rep(c("sort1","sort2","sort3","sort4"),50)

p6lluefekt <- c(50+300*rnorm(20),-50+300*rnorm(20),0+300*rnorm(20),100+300*rnorm(20),
 -100+300*rnorm(20),300+300*rnorm(20),-300+300*rnorm(20),500+300*rnorm(20),
 -500+300*rnorm(20),70+300*rnorm(20))
saagikus <- data.frame(p6ld,aasta,sort,p6lluefekt)

saagikus$saak <- 3250+saagikus$p6lluefekt
saagikus$aastaefekt <- 0
saagikus$aastaefekt[aasta==2003] <- -700+600*rnorm(1)
saagikus$aastaefekt[aasta==2004] <- 50+400*rnorm(1)
saagikus$aastaefekt[aasta==2005] <- 500+310*rnorm(1)
saagikus$aastaefekt[aasta==2006] <- 1300+450*rnorm(1)
saagikus$aastaefekt[aasta==2007] <- -300+170*rnorm(1)

saagikus$sordiefekt <- 0
saagikus$sordiefekt[sort=="sort2"] <- -500
saagikus$sordiefekt[sort=="sort3"] <- -350
saagikus$sordiefekt[sort=="sort4"] <- -75


saagikus$saak <- saagikus$saak + saagikus$aastaefekt + saagikus$sordiefekt

saagikus$factor_aasta <- as.factor(saagikus$aasta)
```

2.

Save and import then into the *R Commander* (or import straight into the *R Commander*) the dataset of calves' weights ('calf' = 'vasikas' in Estonian)

http://ph.eau.ee/~ktanel/DK_0007/vasikas.xls

This dataset contains weights of 55 calves measured at ages 0 to 857 days with average interval 43 days. The task is to estimate the average and the calf-specific growth curves and predict the weights for age 700 days.

If all calves would be weighted exactly after every 100 days (at age 0 days, 100 days, …) than it would be possible to calculate the average weights for all these time moments and join these points with line to get the growth trajectory. Also is then known the weight at 700 days for all cows.

But what to do when

1. the calves are not weighted at the same age;
2. the intervals between weightings are different for different cows;
3. we wish to interpolate the growth curves for some ages for calves without weights from this period;
4. exists some questionable values which may give to the growth curve of single calf nonsensical shape?

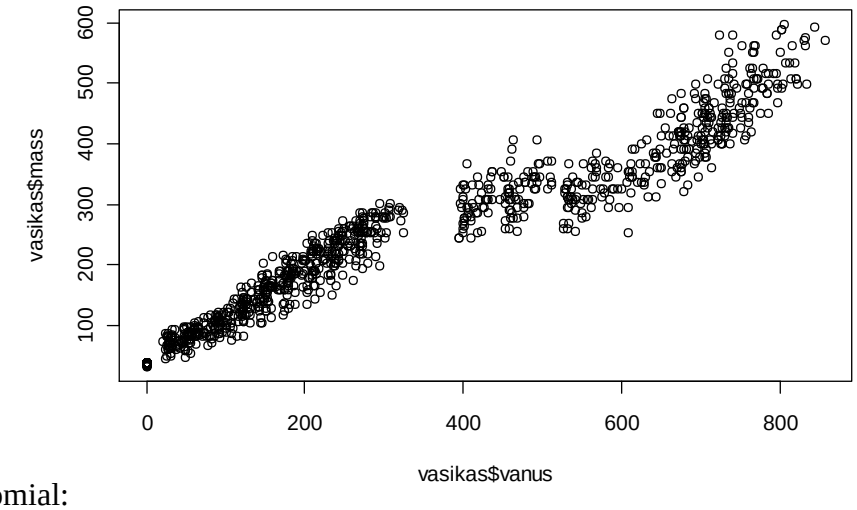
The solution is to estimate the average growth curve as the continuous function over all measurements and then additionally the calf-specific random parameters measuring the calf-specific deviance from the average curve. Such kind models are named as '***random regression model***' or '*random coefficient model*' or ….

2.1.

To get the first overview of the data the scatter plot can be used. For example applying the command

```
plot(vasikas$vanus, vasikas$mass)
```

('age' = 'vanus' in Estonian)



Based on the figure the calves growth should be modelled for example with the 3rd order polynomial:

$$\text{mass}_i = \underbrace{b0 + b1\times\text{vanus}_i + b2\times(\text{vanus}_i)^2 + b3\times(\text{vanus}_i)^3}_{\text{fixed curve over all calves}} + \underbrace{b0_i + b1_i\times\text{vanus}_i + b2_i\times(\text{vanus}_i)^2 + b3_i\times(\text{vanus}_i)^3}_{\text{specific curve for calf } i}$$

Calf-specific random regression coefficients are assumed to follow the normal distribution: $b0_i \sim N(0,\sigma^2_{b0}), b1_i \sim N(0,\sigma^2_{b1}), b2_i \sim N(0,\sigma^2_{b2})$ and $b3_i \sim N(0,\sigma^2_{b3})$.

Such models with random regression coefficients can be fitted with function `lme` or with function `lmer`.
As usual for *R* is the syntax of these commands slightly different, but for both functions writing the factor name 'loom' ('animal') behind the vertical stroke (`|loom`) asks to consider the corresponding effect random. For factors before the vertical stroke are estimated different values for all levels of random factor behind the vertical stroke (`1|loom` means that for each animal the random level is estimated). The following functions will estimate the fixed regression coefficients and random calf-specific regression coefficients (calf-specific growth curves) as described in last page).

```
vasikas.model.1a <- lme(mass ~ vanus+I(vanus^2)+I(vanus^3),
 random=~1+vanus+I(vanus^2)+I(vanus^3)|loom, data=vasikas)
summary(vasikas.model.1a)
coef(vasikas.model.1a)
```

or

```
vasikas.model.1b <- lmer(mass ~ 1+vanus+I(vanus^2)+I(vanus^3) +
 (1+vanus+I(vanus^2)+I(vanus^3)|loom), data=vasikas)
summary(vasikas.model.1b)
```

Command `summary` outputs the estimates of model parameters, additionally allows the function `coef` print out the calf-specific regression coefficients.

- Results of the function `lme`:

Estimated standard deviations of random coefficients.

For example $\hat{\sigma}^2_{b3} = (0.0000000638)^2$ means that $b3_i \sim N[0;(0.0000000638)^2]$.

Estimates of the fixed coefficients

```
Random effects:
 Formula: ~1 + vanus + I(vanus^2) + I(vanus^3) | loom
 Structure: General positive-definite, Log-Cholesky pa
             StdDev       Corr
(Intercept) 4.304048e+00 (Intr) vanus  I(v^2)
vanus       8.989510e-02  0.853
I(vanus^2)  9.714711e-05 -0.800 -0.860
I(vanus^3)  6.388967e-08 -0.087 -0.282 -0.032
Residual    2.091722e+01

Fixed effects: mass ~ vanus + I(vanus^2) + I(vanus^3)
               Value Std.Error  DF   t-value p-value
(Intercept) 22.093326 2.0785240 930  10.62933       0
vanus        1.246283 0.0267173 930  46.64697       0
I(vanus^2)  -0.002217 0.0000756 930 -29.31386       0
I(vanus^3)   0.000002 0.0000001 930  27.55055       0
```

- The function `lmer` outputs only the error message:

```
Messages
[55] ERROR:  Downdated X'X is not positive definite, 4.
```

Or outputs some parameters estimates but writes also into the messages window that the estimation process did not converged and therefore the estimates of model parametrs should not fit the data best.

```
Messages
[19] WARNING: Warning in mer_finalize(ans) : false convergence (8)
```

Obviously is the model for present dataset and function `lmer` too complicated.

2.2.

The estimated standard deviation of the calf-specific regression coefficients of cubed age ($\hat{\sigma}_{b3} = 0.0000000638$) is almost zero. This implies that the corresponding calf-specific cefficients are almost zero ($b3_i \approx 0$, $\forall i$; there is no differencies from the average fixed curve).

Therefore the model without random cubic effect should be fitted:

```
vasikas.model.2a <- lme(mass ~ vanus+I(vanus^2)+I(vanus^3),
 random=~1+vanus+I(vanus^2)|loom, data=vasikas)
summary(vasikas.model.2a)
```

This time are the parameters estimable with both functions `lme` and `lmer`.

```
Random effects:
 Formula: ~1 + vanus + I(vanus^2) | loom
 Structure: General positive-definite, Log-Cholesky pa
             StdDev       Corr
(Intercept) 4.047655e+00 (Intr) vanus
vanus       9.137866e-02  0.844
I(vanus^2)  1.105467e-04 -0.737 -0.900
Residual    2.092328e+01

Fixed effects: mass ~ vanus + I(vanus^2) + I(vanus^3)
                Value Std.Error  DF   t-value p-value
(Intercept) 22.084254 2.0686171 930  10.67585       0
vanus        1.246706 0.0267685 930  46.57361       0
I(vanus^2)  -0.002219 0.0000757 930 -29.32490       0
I(vanus^3)   0.000002 0.0000001 930  28.00254       0
```

Write the command for function `lmer` yourself. Result:

```
> vasikas.model.2b <- lmer(mass ~ 1+vanus+I(vanus^2)+I(vanus^3) + (1+vanus+I(vanus^2)|loom), data=vasikas)

> summary(vasikas.model.2b)
Linear mixed model fit by REML
Formula: mass ~ 1 + vanus + I(vanus^2) + I(vanus^3) + (1 + vanus + I(vanus^2) |      loom)
   Data: vasikas
  AIC  BIC logLik deviance REMLdev
 9117 9171  -4547     9038    9095
Random effects:
 Groups   Name        Variance   Std.Dev.   Corr
 loom     (Intercept) 1.4631e+01 3.8250e+00
          vanus       8.0842e-03 8.9912e-02  1.000
          I(vanus^2)  1.1587e-08 1.0764e-04 -0.901 -0.901
 Residual             4.3926e+02 2.0959e+01
Number of obs: 988, groups: loom, 55

Fixed effects:
             Estimate Std. Error t value
(Intercept)  2.209e+01  2.064e+00   10.70
vanus        1.247e+00  2.670e-02   46.69
I(vanus^2)  -2.219e-03  7.562e-05  -29.34
I(vanus^3)   1.798e-06  6.419e-08   28.01
```

The estimates of the variability of random parameters are slightly different – for example the function `lme` estimates the variance of calf-specific intercepts as $\hat{\sigma}^2_{b0,\text{lme}} = (4.048)^2$ and function `lmer` as $\hat{\sigma}^2_{b0,\text{lmer}} = (3.825)^2$.
It is quite usual for more complex models that different functions and/or computer programs will produce slightly different parameters' estimates, because the different estimation algoritms are used and it is not possible discover exactly which combination of parameters estimates fits the data best (for *R* the estimates got with function `lmer` are considered slightly more accurate, in the same time the algorithm used by functions `lmer` can not converge always).

- Just in case the additional test to compare the last model without the random cubic effect with the initial model should be performed:

```
anova(vasikas.model.1a,vasikas.model.2a)
```

```
> anova(vasikas.model.1a,vasikas.model.2a)
                  Model df      AIC      BIC    logLik   Test   L.Ratio p-value
vasikas.model.1a      1 15 9124.785 9198.160 -4547.393
vasikas.model.2a      2 11 9117.097 9170.905 -4547.549 1 vs 2 0.3117441   0.989
```

Conclusion: the initial and more complicated model does not fit the data better than the second model ($p = 0.989 > 0.05$). Thus it is enough to estimate the growth curves only with fixed cubic effect common for all calves.

The output of models comparison contains also the values of the *AIC* (Akaike information criteria) and *BIC* (Bayesian information criteria). These coefficients are applicable in models' comparison also in very complicated cases when the tests of statistical significance performed by *R* are not correct. These coefficients does not test the statistical significance of models' difference but just measure and describe the relative goodness (compromise between models' complexity and fitness). The model with smaller *AIC* and *BIC* fits the data better.

At the present situation both *AIC* and *BIC* are smaller for model 2 (model without random cubic term) and this is additional proof of advantage of model 2.

2.3.

But makes it sense to estimate random squared term for each calf?
In other words – is the variability of calf-specific squared terms different from zero?

Lets study this. Fit the model without random squared term:

```
vasikas.model.3a <- lme(mass ~ vanus+I(vanus^2)+I(vanus^3),
 random=~1+vanus|loom, data=vasikas)
summary(vasikas.model.3a)
```

```
Random effects:
 Formula: ~1 + vanus | loom
 Structure: General positive-definite, Log-Cholesky pa
             StdDev      Corr
(Intercept) 11.60696171 (Intr)
vanus        0.03766962 0.228
Residual    21.71359004

Fixed effects: mass ~ vanus + I(vanus^2) + I(vanus^3)
                Value Std.Error  DF   t-value p-value
(Intercept) 21.649331 2.5935096 930   8.34750       0
vanus        1.255950 0.0249687 930  50.30091       0
I(vanus^2)  -0.002256 0.0000756 930 -29.85199       0
I(vanus^3)   0.000002 0.0000001 930  28.33958       0
```

Comparing the results of models 2 and 3 it follows that omitting the random calf-specific squared term caused the increasing of the variability of random calf-specific intercepts almost 3 times (4.05 *vs* 11.61).

Therefore, considering also the squared term common for all calves the model tries to model the different growth of calves by incresing the difference of starting points of growth curves.

- Is this action successful enough? Lets test.

```
> anova(vasikas.model.3a,vasikas.model.2a)
                  Model df      AIC      BIC    logLik   Test  L.Ratio p-value
vasikas.model.3a      1  8 9150.988 9190.121 -4567.494
vasikas.model.2a      2 11 9117.097 9170.905 -4547.549 1 vs 2 39.89047  <.0001
```

Answer is no, the more complicated model (model 2) models the calves's growth statistically significantly better than more simple model (model 3), $p < 0.0001$. Also the values of *AIC* and *BIC* are smalle for model 2.

- Thus the growth curves can be modelled with cubic polynomial which contains the calf *i* specific intercept, linear and squared terms:

$$\text{mass}_i = \underbrace{(b0+b0_i)} + \underbrace{(b1+b1_i)}\times\text{vanus}_i + \underbrace{(b2+b2_i)}\times(\text{vanus}_i)^2 + b3\times(\text{vanus}_i)^3,$$

$$b0_i \sim N(0,\sigma^2_{b0}),\ b1_i \sim N(0,\sigma^2_{b1}),\ b2_i \sim N(0,\sigma^2_{b2})\ .$$

```
> coef(vasikas.model.2)
     (Intercept)    vanus I(vanus^2)   I(vanus^3)
2684    19.22594 1.188462 -0.002115887 1.797621e-06
2685    17.18089 1.150321 -0.002152235 1.797621e-06
2686    22.67008 1.277953 -0.002287691 1.797621e-06
2687    24.80621 1.313947 -0.002217329 1.797621e-06
2688    21.22402 1.238332 -0.002287055 1.797621e-06
2689    23.64002 1.279134 -0.002217506 1.797621e-06
2690    25.88169 1.334806 -0.002291997 1.797621e-06
2691    24.04751 1.288894 -0.002279265 1.797621e-06
2693    19.93963 1.194271 -0.002166115 1.797621e-06
2695    17.62261 1.140073 -0.002120251 1.797621e-06
2696    26.56294 1.362341 -0.002379284 1.797621e-06
2697    14.78917 1.065995 -0.002068288 1.797621e-06
```

2.4.

But how about the growth curves?

```
x=rep(seq(0,800,2), length(unique(vasikas$loom)))
y=predict(vasikas.model.2, data.frame(vanus=x, loom=unique(vasikas$loom)), level=1)
plot(x, y, cex=0.1, xlab="Vanus", ylab="Mass, kg")
lines(rep(seq(0,800,1)),
 predict(vasikas.model.2, data.frame(vanus=rep(seq(0,800,1))),level=0),lwd=2,col="red")
```

this command estimates the average weights for ages 0-800 days