Advanced Dataanalysis and Statistical Modelling: Assignment 3

Spring 22, 02424
May 11, 2022

s170303, Andreas Engly
s174434, Mads Esben Hansen

Danmarks
Tekniske Universitet

Contents

Part 1: Strength of ready mixed concrete

Problem A

A.1: Plot of Concrete Strength

The file *concrete.csv* holds information about the strength of concrete in 5 different batches respectively 7- and 28-days after production. The strengths have been sampled at different air temperatures. At first, the 7-days samples are shown. As the data is *imbalanced* due to different number of samples from each batch, it cannot by initial visual inspection be ruled out that the variance is constant between batches.

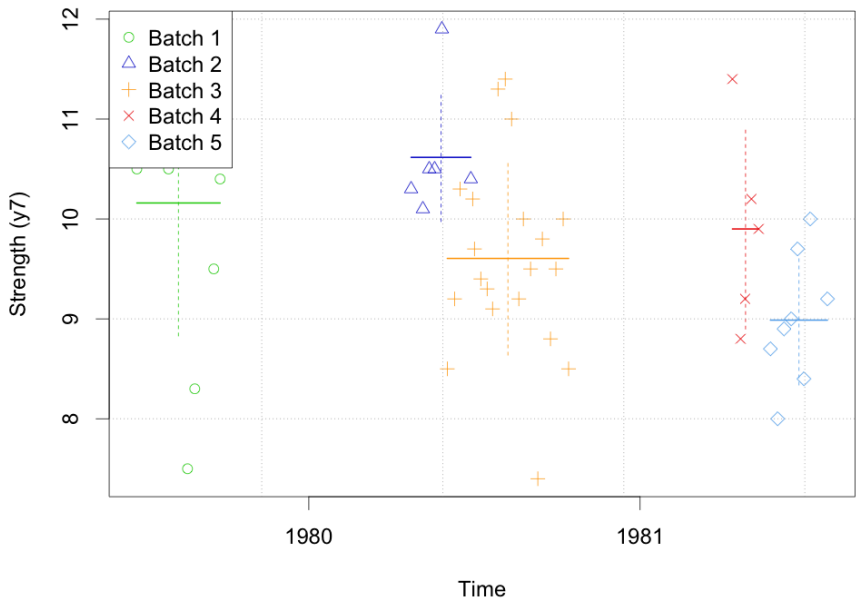


Figure 1 – Observations of 7-days strength.

Below the 28-days samples are shown.

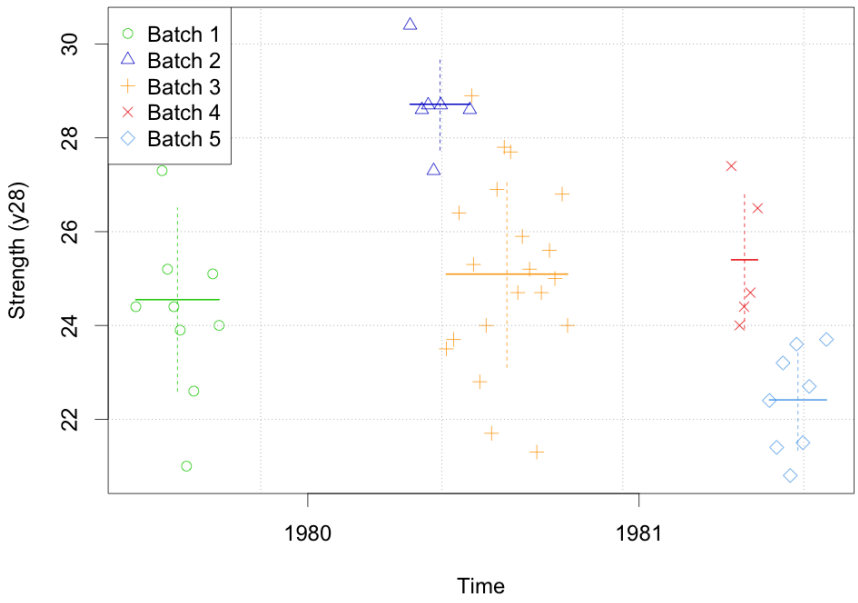


Figure 2 – Observations of 28-days strength.

A.2: Conditional Means

We can now estimate the conditional mean for each batch. Note that this value is represented by vertical lines in the plots above. The means for strength after 7 days are estimated as

$$\mathbb{E}[y_7|B=1] = 10.16000 \tag{1}$$
$$\mathbb{E}[y_7|B=2] = 10.61667 \tag{2}$$
$$\mathbb{E}[y_7|B=3] = 9.60500 \tag{3}$$
$$\mathbb{E}[y_7|B=4] = 9.90000 \tag{4}$$
$$\mathbb{E}[y_7|B=5] = 8.98750 \tag{5}$$

where B denotes the batch. Similarly, we can estimate the conditional means for the strength after 28 days

$$\mathbb{E}[y_{28}|B=1] = 24.55000 \tag{6}$$
$$\mathbb{E}[y_{28}|B=2] = 28.71667 \tag{7}$$
$$\mathbb{E}[y_{28}|B=3] = 25.09500 \tag{8}$$
$$\mathbb{E}[y_{28}|B=4] = 25.40000 \tag{9}$$
$$\mathbb{E}[y_{28}|B=5] = 22.41250 \tag{10}$$

A.3: Model Formulation

We can now model the strength after 28 days as a *one-way model with random effects* as given in definition 5.2 in [1].

$$Y_{ij} = \mu + U_i + \varepsilon_{ij} \tag{11}$$

where the indices i and j denote batches and samples respectively. The random variables $\varepsilon_{ij} \sim N(0, \sigma^2)$ and $U_i \sim N(0, \sigma_U^2)$ denote within-batch errors and between-batch errors. In addition, ε_{ij} are mutually independent, as are U_i, and ε_{ij} are independent of U_i. According to theorem 5.1 in [1], we have that the marginal distribution of Y_{ij} is normal with

$$\mathbb{E}[Y_{ij}] = \mu \tag{12}$$

$$Cov[Y_{ij}, Y_{hl}] = \begin{cases} \sigma_U^2 + \sigma^2 & for\ (i,j) = (h,l) \\ \sigma^2 & for\ i = h,\ j \neq l \\ 0 & otherwise \end{cases} \tag{13}$$

If we had been interested in the specific batches only, we could have formulated a systematic model with $U_i = 0$. However, as the batches in question only constitute part of the actual production capacity, describing the variance between batches $\mathbb{V}[\mu_i] = \sigma_U^2$ leads to a more rich description of the deliveries. It could e.g. be relevant when reporting risks as construction could require many batches of concrete.

A.4: Independence of Air Temperature

We have to test for the dependency on air temperature, we augment the model to include a design matrix $\boldsymbol{X}$. The generic form follows from definition 5.3 [1, p. 179].

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{U} + \boldsymbol{\epsilon} \tag{14}$$

where $\boldsymbol{\epsilon} \sim N(0, \boldsymbol{\Sigma})$ and $U \sim N(0, \sigma_U^2 \boldsymbol{I})$. Initially, $\boldsymbol{X} = [\mathbf{1}_N \text{ air.temp}]$ is assumed to be a matrix of N 1's in the first column and the air temperature in the second column. As a consequence, $\boldsymbol{\beta}$ corresponds to an intercept for the *systematic effect* and a common sensitivity to air temperature across the batches. The design matrix $\boldsymbol{Z}$ is as follows

$$\boldsymbol{Z} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

As the data is *imbalanced*, because a different number of samples is drawn from each batch, the number of 1's in each column are different. The models are fitted with the package *nlme*. Figure 3 shows the parameters for the one-way model with mixed effects and the mixed effects model with air temperature as a linear fixed input. To test whether it has a significant effect, we can perform a likelihood ratio test. Consider the hypotheses

$$\mathcal{H}_0 : \boldsymbol{\theta} \in \Omega_0 = \mathbb{R} \times \mathbb{R}_+^2$$
$$\mathcal{H}_1 : \boldsymbol{\theta} \in \Omega \setminus \Omega_0 = \mathbb{R}^2 \times \mathbb{R}_+^2 \setminus \mathbb{R} \times \mathbb{R}_+^2$$

where $\mathcal{H}_0$ is known as the *null hypothesis* and $\mathcal{H}_1$ is known as the *alternative hypothesis*. By *Wilk's Likelihood Ratio Test* [1, p. 26] it can be shown that

$$\lambda_{\mathcal{H}_0|\mathcal{H}_1}(\boldsymbol{Y}) = -2\log\frac{\sup_{\boldsymbol{\theta}\in\Omega_0} L(\boldsymbol{\theta};\boldsymbol{y})}{\sup_{\boldsymbol{\theta}\in\Omega} L(\boldsymbol{\theta};\boldsymbol{y})} \to \chi^2(k-m) \tag{16}$$

where $\to$ denotes convergence in probability. A general assumption of the theorem is that the parameters are in the interior of the parameter space. The *p-value* is found to be

$$\mathbb{P}(\lambda_{\mathcal{H}_0|\mathcal{H}_1}(\boldsymbol{y}) < \chi^2(1)) = 0.6672 \tag{17}$$

which means that there is not strong evidence against $\mathcal{H}_0$. Therefore, $\mathcal{H}_0$ cannot be rejected. We therefore conclude that the strength after 28 days is not dependent on air temperature, since it does not yield any significant improvements to the model. The model summaries below show the estimated parameters.

```
Linear mixed-effects model fit by maximum likelihood
  Data: conc.dat
       AIC      BIC    logLik
  211.2846 216.9601 -102.6423

Random effects:
 Formula: ~1 | batch
        (Intercept) Residual
StdDev:    1.898261 1.735903

Fixed effects:  y28 ~ 1
               Value Std.Error DF  t-value p-value
(Intercept) 25.20741 0.9018761 44 27.94997       0

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.18878056 -0.63339444  0.01324499  0.53170704  2.18934335

Number of Observations: 49
Number of Groups: 5
```

(a) One-way model with mixed effects.

```
Linear mixed-effects model fit by maximum likelihood
  Data: data
       AIC     BIC    logLik
  213.0998 220.667 -102.5499

Random effects:
 Formula: ~1 | batch
        (Intercept) Residual
StdDev:    1.950391 1.727518

Fixed effects:  y28 ~ 1 + air.temp
                Value Std.Error DF   t-value p-value
(Intercept) 24.881256 1.2057038 43 20.636292  0.0000
air.temp     0.025848 0.0601759 43  0.429547  0.6697
 Correlation:
             (Intr)
air.temp -0.633

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.09389748 -0.57732517  0.01507754  0.51751861  2.12143605

Number of Observations: 49
Number of Groups: 5
```

(b) Mixed effects model with air temperature as fixed input.

Figure 3 – Summary of Mixed-Effect Model.

Finally, we would like to justify the model assumption. Figure 4 shows the QQ-plot of the within-batch residuals with a 95% confidence interval. As they appear to be normal distributed, we are happy with the model.

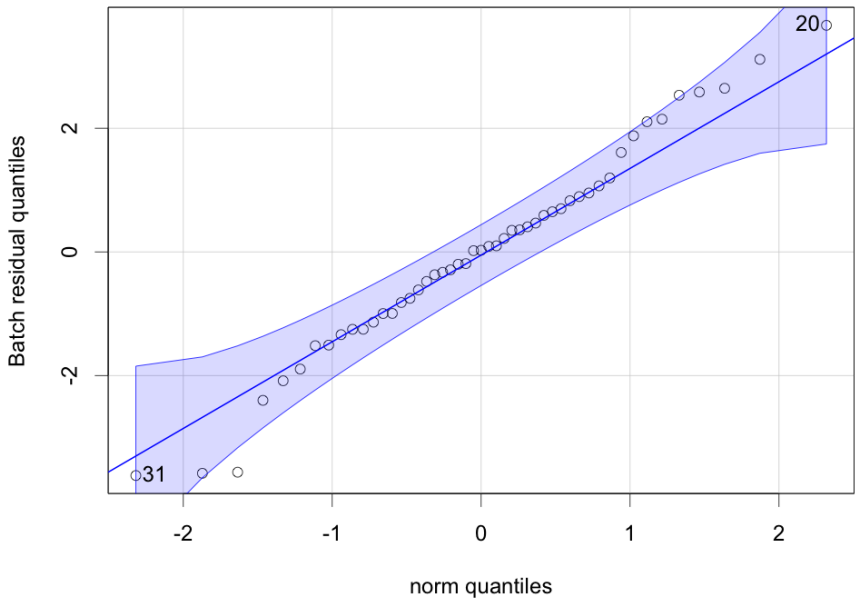


Figure 4 – QQ-plot of within-batch residuals.

Problem B

B.5: Multivariate Mixed Effect Model

We would like to establish a multivariate mixed effect model for the concrete measurements. In this case, the observations are 2-dimensional vectors denoted by $\boldsymbol{x} = (x_1, x_2)$. The model has the following structure.

$$\boldsymbol{X}_{ij} = \boldsymbol{\mu} + \boldsymbol{u}_i + \boldsymbol{\epsilon}_{ij}, \quad i = 1, 2, ..., k; \; j = 1, 2, ..., n_i \tag{18}$$

where $\boldsymbol{u}_i \sim N(\boldsymbol{0}, \boldsymbol{\Sigma_0})$, $\boldsymbol{\epsilon}_{ij} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$, and all $\boldsymbol{u}$ are independent on all $\boldsymbol{\epsilon}$.

B.6: Estimation of Model Parameters

The moments can be estimated as outlined in [1, p. 193]. First let

$$\bar{\boldsymbol{X}}_{++} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} \boldsymbol{X}_{ij}/N, \tag{19}$$

$$\boldsymbol{SSE} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (\boldsymbol{X}_{ij} - \bar{\boldsymbol{X}}_{i+})(\boldsymbol{X}_{ij} - \bar{\boldsymbol{X}}_{i+})^T \tag{20}$$

and

$$\boldsymbol{SSB} = \sum_{i=1}^{k} n_i (\bar{\boldsymbol{X}}_{i+} - \bar{\boldsymbol{X}}_{++})(\bar{\boldsymbol{X}}_{i+} - \bar{\boldsymbol{X}}_{++})^T. \tag{21}$$

The estimate for SSE is given by

$$\boldsymbol{SSE} = \begin{bmatrix} 43.07058 & 55.46008 \\ 55.46008 & 132.33158 \end{bmatrix}.$$

The SSB is given by

$$\boldsymbol{SSB} = \begin{bmatrix} 11.34942 & 33.26849 \\ 33.26849 & 139.42801 \end{bmatrix}.$$

From (5.29) in [1, p. 167] the weighted group size is

$$n_0 = \frac{N - \sum_i n_i^2/N}{k-1} = 9.061224.$$

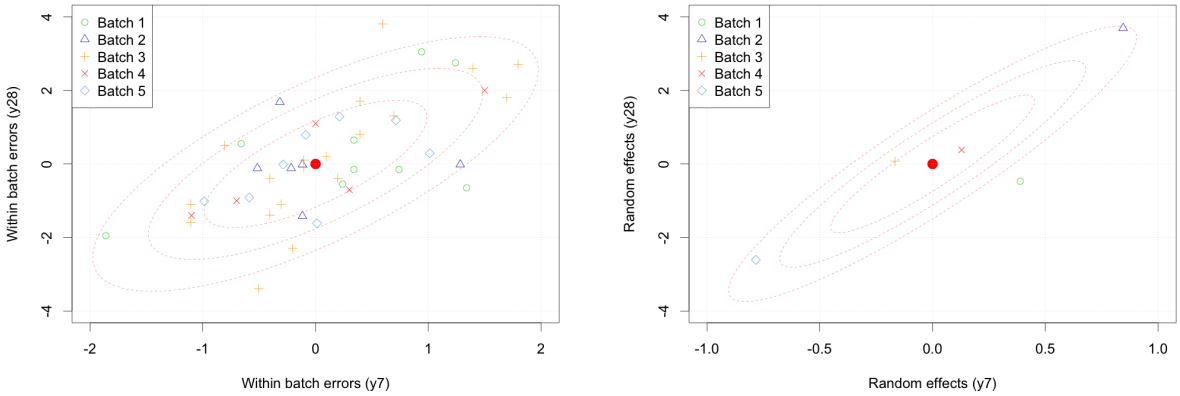
We then get the parameter estimates

$$\tilde{\boldsymbol{\mu}} = \begin{bmatrix} 9.771429 \\ 25.020408 \end{bmatrix}, \tag{22}$$

$$\tilde{\boldsymbol{\Sigma}} = \begin{bmatrix} 0.9788769 & 1.260456 \\ 1.2604564 & 3.007536 \end{bmatrix} \text{ and} \tag{23}$$

$$\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{0}} = \begin{bmatrix} 0.2051022 & 0.7787762 \\ 0.7787762 & 3.5149186 \end{bmatrix}. \tag{24}$$

Figure 5 shows the within-group error and the random effect. We can see that there is a positive correlation between the strength after 7 and 28 days based on the ellipses.



(a) Within group error, i.e., $\boldsymbol{\epsilon}_{ij}$.

(b) Random effect, i.e., $\boldsymbol{u}_i$.

Figure 5

B.7: Correlation

As just hinted, there is a significant positive correlation between the strength after 7 days and the strength after 28 days. We can estimate the correlation utilizing theorem 5.10 from [1]. The estimate of the correlation coefficients is found as

$$\hat{\rho}_{ij} = \frac{\tilde{\boldsymbol{\Sigma}}_{ij}}{\sqrt{\tilde{\boldsymbol{\Sigma}}_{11}\tilde{\boldsymbol{\Sigma}}_{22}}} \tag{25}$$

The estimate for correlation is found to be

$$\hat{\rho}_{12} = \hat{\rho}_{21} = 0.73 \tag{26}$$

As the samples after respectively 7 and 28 days follows a bivariate normal distribution, it can be shown that $\hat{\rho} \sim N(\operatorname{arctanh}(\rho), \frac{1}{\sqrt{N-3}})$. The 95% confidence interval is then computed and transformed back via *tanh*. This transformation is commonly known as a *Fischer transformation*.

$$\rho_{12} = \rho_{21} \in [0.57 \quad 0.84] \tag{27}$$

As the interval does not include 0, we conclude that the correlation is significant.

Part 2: Clothing insulation count data

We now return to the clothing insulation data used in assignment 2.

Problem A

A.1: Difference between Subjects

Once again, we use information about clothing insulation. This time we consider the variables given in Table 1.

Table 1 – Variables in clothing data.

Variable	Type	Description
clo	Continuous	Number of changes
tOut	Continuous	Outdoor temperature
tInOp	Continuous	Indoor operating temperature
sex	Factor	Sex of the subject
subjId	Factor	Identifier for subject
time	Continuous	Total measurement time
day	Factor	Day (within the subject)
nobs	Integer	Measurement number (within the day)

Figure 6 shows the number of changes each subject made to their clothing. Most subjects were monitored 3-4 days with 5 observations on each day. This means that each box in the box represent how many of the 5 observations each person changed across 3-4 days of observation. It seems that there is quite different behaviour between the people. Some people tend to change clothes much more frequently than others. The red bars are female subjects, and the green bars are male subjects.

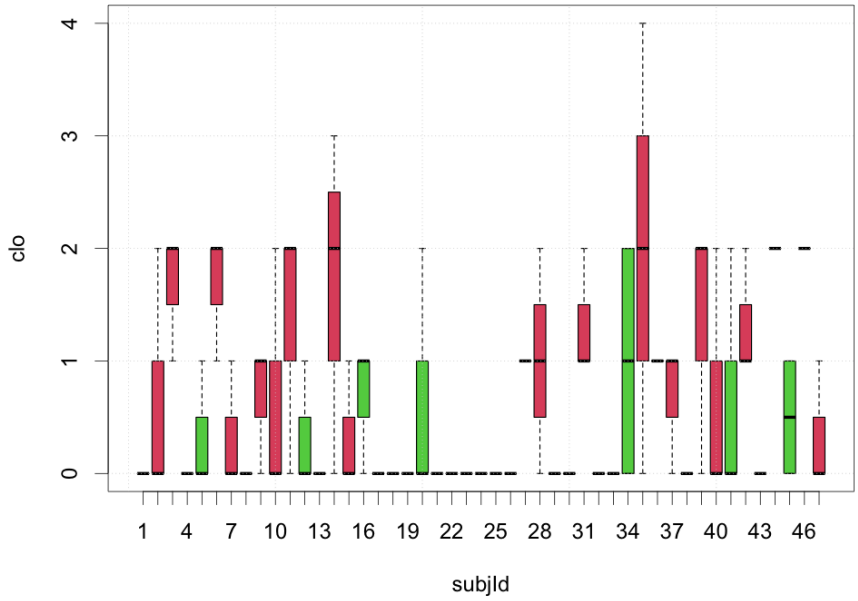


Figure 6 – Number of clothes changes per subject.

Figure 7 shows a box-plot of number of clothing changed for females and males. From the box-plot it seems that females change their clothes more frequently than males.

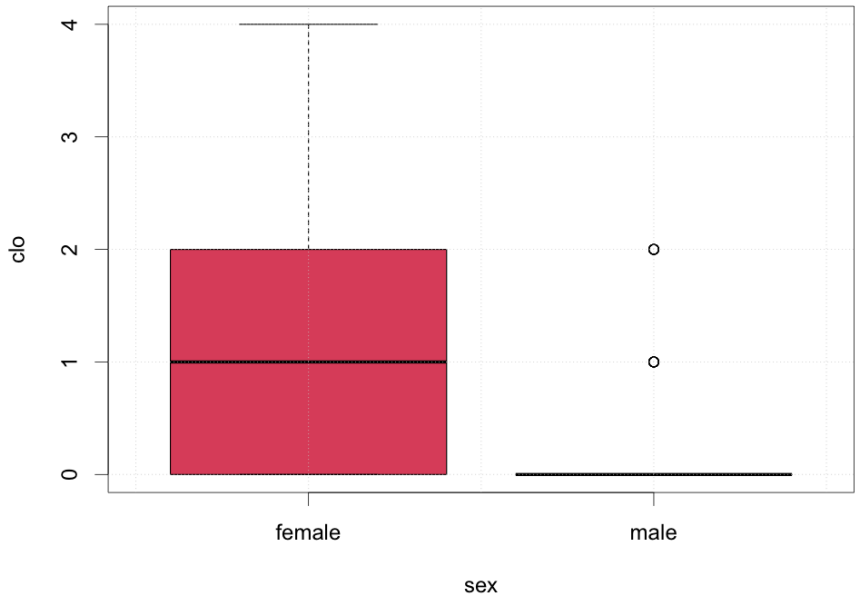


Figure 7 – Number of clothes changes per sex.

A.2: Generalized Linear Mixed Effect Model

In the following model, we relax the assumption of Gaussian distributed data. Instead we introduce a generalized mixed effects model where the first stage is a assumed to be distributed according to a distribution from the exponential family. In particular, we investigate the Binomial and the Poisson distributions.

We follow example 5.4 [1] and start by formulating the fixed model. As mentioned, each subject is measured 3 or 4 times. This puts a limitation on how complex a model we can build. Building a model with individual intercepts and dependency of indoor and outdoor temperature for each subject does not only seem extreme, it will also lead to a over-parameterization, i.e., more parameters than the data allows. We will therefore limit ourselves to only using first order terms. That is, no interaction between parameters. Additionally, only subjId will be conditioned on as random effect, and only the intercept will be subject to such random effect—we will not consider random effects on any slopes.

Using the binomial distribution and modelling the probability of changing clothes, this leads us to the full model on hierarchical form

$$\texttt{clo}_{id}|\texttt{id} \sim Binom(n_{id}, logit^{-1}(\alpha_i + \beta \cdot \texttt{tInOp} + \gamma \cdot \texttt{tOut} + U_{id})), \tag{28}$$

$$U_{id} \sim N(0, \sigma^2), \ and \tag{29}$$

$$id \in \{\texttt{subjId}\}, \ i \in \{male, female\}. \tag{30}$$

We now proceed to model reduction. At first we found that `tOut` was insignificant. Then the same was concluded for for `tInOp`, as the removal of the parameters increased the *Akeike Information Criterion* (AIC). Finally, we test if the random effect gives any meaningful improvements to the model—we found that it does. In turn, we end with the model given by

$$\texttt{clo}_{id}|\texttt{id} \sim Binom(n_{id}, logit^{-1}(\alpha_i + U_{id})), \tag{31}$$

$$U_{id} \sim N(0, \sigma^2), \ and \tag{32}$$

$$id \in \{\texttt{subjId}\}, \ i \in \{male, female\}. \tag{33}$$

Figure 8 shows the parameter estimates for the model given in Equation 33.

```
 Family: binomial  ( logit )
Formula:          cbind(clo, nobs - clo) ~ as.factor(sex) + (1 | subjId) - 1
Data: clo

     AIC      BIC   logLik deviance df.resid
   263.4    272.1   -128.7    257.4      133

Random effects:

Conditional model:
 Groups Name        Variance Std.Dev.
 subjId (Intercept) 0.8195   0.9053
Number of obs: 136, groups:  subjId, 47

Conditional model:
                    Estimate Std. Error z value Pr(>|z|)
as.factor(sex)female  -1.7859     0.2540  -7.031 2.06e-12 ***
as.factor(sex)male    -3.1402     0.3652  -8.598  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 8 – Parameter estimates of mixed effects model for probability of changing clothes.

Instead of using the binomial distribution, and modelling the change of clothes by means of the probability of change, we could model the rate of change instead. Under the assumption that the rate of change is constant, rather than the probability, it makes sense to use the Poisson distribution.

As before, and for many of the same reasons, we will limit ourselves to only using first order term, so no interaction between parameters. Once again, only subjId will be conditioned on as random effect, and only the intercept will be subject to such random effects; we will not consider random effects on any slopes. Using the Poisson distribution and modelling the rate of change, this leads us to the full model on hierarchical form

$$\texttt{clo}_{id}|\texttt{id} \sim Pois(\exp(\alpha_i + \beta \cdot \texttt{tInOp} + \gamma \cdot \texttt{tOut} + U_{id})), \tag{34}$$

$$U_{id} \sim N(0, \sigma^2), \ and \tag{35}$$

$$id \in \{\texttt{subjId}\}, \ i \in \{male, female\}. \tag{36}$$

Again, we found that first `tOut` and then `tInOp` yields no significant; the AIC is improved when the parameters are removed. We test if the random effect gives any meaningful improvements to the model—and once again, it does. In turn, we end with the model given by

$$\texttt{clo}_{id}|\texttt{id} \sim Pois(\exp(\alpha_i + U_{id})), \tag{37}$$

$$U_{id} \sim N(0, \sigma^2), \ and \tag{38}$$

$$id \in \{\texttt{subjId}\}, \ i \in \{male, female\}. \tag{39}$$

Figure 9 shows the parameter estimates for the model given in Equation 39.

```
 Family: poisson  ( log )
Formula:          clo ~ as.factor(sex) + (1 | subjId) - 1
Data: clo

     AIC      BIC   logLik deviance df.resid
   266.1    274.8   -130.0    260.1      133

Random effects:

Conditional model:
 Groups Name        Variance Std.Dev.
 subjId (Intercept) 0.4596   0.6779
Number of obs: 136, groups:  subjId, 47

Conditional model:
                    Estimate Std. Error z value Pr(>|z|)
as.factor(sex)female  -0.3399     0.2107  -1.613    0.107
as.factor(sex)male    -1.4962     0.3145  -4.757 1.96e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 9 – Parameter estimates of mixed effects model for rate of changing clothes.

Comparing the binomial model with the Poisson model, we see that the binomial has a slightly better AIC, with 263.4 and 266.1 respectively. This indicates that modelling the probability rather than the rate of change yields a slightly better model for the change of clothes.

Problem B

We consider a Poisson-Gamma model class for modelling the change of clothes.

$$\begin{aligned} Y_{ij}|U_i &\sim Pois(\mu_{ij}U_i) \\ U_i &\sim Gamma(\alpha, \beta) \\ \boldsymbol{\mu}_{ij} &= e^{\boldsymbol{x}_{\boldsymbol{ij}}^T\boldsymbol{\theta}} \end{aligned}$$

The two first conditional moments are given as

$$\mathbb{E}[Y_{ij}|U_i = u_i] = \mathbb{V}[Y_{ij}|U_i = u_i] = \mu_{ij}u_i \tag{40}$$

We are using a logarithmic link function, $\ln(\mu_{ij}) = x_{ij}^T\theta$, as it has the same range as the *parameter space* of $\lambda_{ij} = \mu_{ij}U_i \in (0, \infty)$. As the model is hierarchical in nature, the two stages can be represented by the following conditional probability distributions.

The probability mass function for the Poisson distribution is

$$f_{Y_{ij}|U_i}(y_{ij}; \lambda_{ij}) = \mathbb{P}(Y_{ij} = y_{ij}|U_i) = \frac{\lambda_{ij}^{y_{ij}}e^{-\lambda_{ij}}}{y_{ij}!} \tag{41}$$

and the probability density function for the Gamma distribution is

$$f_{U_i}(u_i; \alpha, \beta) = \frac{1}{\beta\Gamma(\alpha)}(\frac{u_i}{\beta})^{\alpha-1}e^{-u_i/\beta} \tag{42}$$

where α and β are assumed to be non-random, time-invariant population parameters. By Equation 40, the model can be written as

$$Y_{ij} = e^{\boldsymbol{x}_{\boldsymbol{ij}}^T\boldsymbol{\theta}}u_i \tag{43}$$

where $\boldsymbol{x_{ij}}$ are *observed*, and $\boldsymbol{u}_i$ are *unobserved*. We introduce one level of grouping, namely the *subjectId*. It seems reasonable to create the model in this way as the individuals can be considered a sample drawn from as population.

B.1: Implementation using Laplace Approximation

In order to obtain the marginal likelihood $L_M(\boldsymbol{\theta}; \boldsymbol{y})$, we have to integrate the random effects out. As the random effects are assumed to follow a gamma distribution, they belong to the space $\mathbb{R}_+$. The marginal likelihood is therefore given by

$$L_M(\boldsymbol{\theta}; \boldsymbol{y}) = \int_{\mathbb{R}_+^q} L(\boldsymbol{\theta}; \boldsymbol{u}, \boldsymbol{y})d\boldsymbol{u}$$

where q is the dimension of the vector $\boldsymbol{u}$. In general, this integral can be difficult to solve. To cope with this, we can approximate the integral with *Laplace's Method*. It can be shown that

$$\ell_M(\boldsymbol{\theta}; \boldsymbol{y}) \to \log f_{Y|u}(\boldsymbol{y}; \tilde{\boldsymbol{u}}, \boldsymbol{\theta}) + \log f_U(\tilde{\boldsymbol{u}}; \alpha, \beta) - \frac{1}{2}\log|\frac{\boldsymbol{H}(\tilde{\boldsymbol{u}})}{2\pi}| \tag{44}$$

where $\tilde{\boldsymbol{u}} = \text{argmax}_{\boldsymbol{u}} L(\boldsymbol{\theta}; \boldsymbol{u}, \boldsymbol{y})$. Hence, $\tilde{\boldsymbol{u}}$ is the configuration of the random effects that maximizes the likelihood function for a given set of parameters. For derivation see

appendix. We now have the necessary framework to estimate the marginal likelihood given a set of parameters, $\boldsymbol{\theta}$, the design matrix, $\boldsymbol{X}$, and the response, $\boldsymbol{y}$.

Formulating the model where the conditional mean is given by a seperate intercept for males and females, we define the design matrix by

$$\boldsymbol{X}_{i1} = \begin{cases} 1 & sex_i = male \\ 0 & sex_i = male \end{cases} \tag{45}$$

$$\boldsymbol{X}_{i2} = \begin{cases} 0 & sex_i = female \\ 1 & sex_i = female \end{cases} . \tag{46}$$

In turn we get the parameters

$$\boldsymbol{\theta} = [\theta_1,\ \theta_2] \tag{47}$$

where θ_1 is the log mean rate for males and θ_2 is the log mean rate for females. We can now optimize over the marginal likelihood function and thereby estimate the parameters. Table 2 shows the parameter estimates for the hierarchical model using the Laplacian method to approximate the likelihood function.

Table 2 – Parameters estimates for 2-step hierarchical model using Laplacian approximation.

Parameter	Estimate
$\hat{\theta}_1$	-1.23
$\hat{\theta}_2$	-0.15
$\hat{\alpha}$	3.59
$\hat{\beta}$	0.28

Below the R code used to solve the optimization problem and estimate the parameters is demonstrated.

```r
dat.count = read.csv('dat_count3.csv', sep = ";", stringsAsFactors = F)

X <- matrix(0,ncol=2,nrow=dim(dat.count)[1])
X[ ,1]<- dat.count$sex=="male"
X[ ,2]<-dat.count$sex=="female"

## Joint Likelihood
nll <- function(u,params,X){
  theta <- params[1:2] ## male, female
  alpha <- params[3]

  # Compute lambda for Poisson
  mu <- exp(X[,1]*theta[1] + X[,2]*theta[2])


  # Compute negative log-likelihood
  #first_stage <- -sum(dgamma(u[dat.count$subjId], shape = alpha, rate = beta, log = TRUE))
  first_stage <- -sum(dgamma(u, shape = alpha, scale = 1/alpha, log = TRUE))
```

```
  second_state <- -sum(dpois(dat.count$clo, mu*u[dat.count$subjId], log = TRUE))
  return(second_state + first_stage)
}

#####################################################
## use independence of u's in nlminb
nll.LA <- function(params,X){
  fun.tmp <- function(ui,u,params,X,i){
    u <- u*0+1
    u[i]<-ui
    nll(u,params,X)
  }
  u <- numeric(47)

  ## Use grouping structure
  for(i in 1:length(u)){
    u[i] <- nlminb(1,objective = fun.tmp, u=u,params=params,
                   X=X,i=i, lower = 0)$par
  }
  l.u <- nll(u,params,X)
  H <- numeric(length(u))
  for(i in 1:length(u)){
    H[i] <- hessian(func = fun.tmp, x = u[i], u=u,
                    params = params, X=X,i=i)}

  l.u + 0.5 * log(prod(H/(2*pi)))
}
system.time(fit <- nlminb(c(-1,0,2),nll.LA,X=X))
fit
```

B.2: Comparison with Previous Model

In the previous question, we modelled the change of clothing using a Poisson model with Gaussian random effects. In this model, we saw similar parameter estimates to the Poisson-Gamma model. However, for both males and females, the parameter estimates where slightly lower for the first model meaning a lower estimate of mean rate of change for both males and females.

We were confused as to why this is the case. After some investigations, we decided to calculate the mean of the estimated random effects for the two models. For the Poisson-Gaussian model we found a mean of $\bar{U} = 0.04$. Notice that we can write conditional model as

$$\texttt{clo}_{id}|\texttt{id} \sim Pois(\exp(\alpha_i)\exp(U_{id})). \tag{48}$$

If we compare with the Poisson-Gamma model, we see a strong connection between $\exp(U_{id})$ in the Poisson-Gaussian model and U_i in the Poisson-Gamma model. If we find the mean of this part, we get $\exp(\bar{U}_{id}) = 1.16$. For the Poisson-Gamma model we have $\bar{U}_i = 0.81$. This is a difference of 43%.

If we now return to the parameters estimates, and remember that the rate is given by $\exp(\theta)$, we see that in the Poisson-Gaussian model the rates, $\exp(\theta)$, are approximately 40% lower than the corresponding rates in the Poisson-Gamma model. In other words, due to the difference of shape of the likelihood functions for the random effects, one model *prefers* to have lower fixed parameters and generally increase these by the random effect, while the other is the opposite.

Problem C

C.1: Sex as Explanatory Variable

We now consider the model with only sex as explanatory variable. On hierarchical form, this is given by

$$\begin{aligned} Y_{ij}|U_i &\sim Pois(\mu_{ij}U_i) \\ U_i &\sim Gamma(\alpha, \beta) \\ \boldsymbol{\mu}_{ij} &= e^{\boldsymbol{x}_{ij}^T\boldsymbol{\theta}}. \end{aligned}$$

Where the probability density functions (PDF) of the two stages are given by

$$f_{Y_{ij}|U_i}(y_{ij}, \mu_{ij}U_i) = \frac{(\mu_{ij}U_i)^{y_{ij}}e^{-\mu_{ij}U_i}}{y_{ij}!} \tag{49}$$

$$f_{U_i}(u_i, \alpha, \beta) = \frac{1}{\beta\Gamma(\alpha)}(\frac{u_i}{\beta})^{\alpha-1}e^{-u_i/\beta}. \tag{50}$$

C.2: Marginal Distribution

We now want to write down the marginal PDF for Y_{ij}. Using the law of total probability we get

$$f_{Y_{ij}}(y_{ij}, \mu_{ij}U_i) = \int_0^\infty f_{Y_{ij}|U_i}(y_{ij}, \mu_{ij}u_i) \cdot f_{U_i}(u_i, \alpha, \beta)du_i \tag{51}$$

$$= \int_0^\infty \frac{(\mu_{ij}u_i)^{y_{ij}}e^{-\mu_{ij}u_i}}{y_{ij}!} \cdot \frac{1}{\beta\Gamma(\alpha)}(\frac{u_i}{\beta})^{\alpha-1}e^{-u_i/\beta}du_i \tag{52}$$

$$= \frac{1}{\beta^\alpha\Gamma(\alpha)y_{ij}!}\int_0^\infty (\mu_{ij}u_i)^{y_{ij}}e^{-\mu_{ij}u_i} \cdot (u_i)^{\alpha-1}e^{-u_i/\beta}du_i \tag{53}$$

$$= \frac{\mu_{ij}^{y_{ij}}}{\beta^\alpha\Gamma(\alpha)y_{ij}!}\int_0^\infty u_i^{y_{ij}+\alpha-1}e^{-u_i(\mu_{ij}+1/\beta)}du_i \tag{54}$$

We now recognize part of the integral as the kernel of a Gamma distribution with shape parameter $y_{ij} + \alpha$ and scale parameter $1/(\mu_{ij} + 1/\beta)$. Therefore we multiply and divide by the corresponding normalization term (from the PDF of the gamma distribution)

$$f_{Y_{ij}}(y_{ij}, \mu_{ij}U_i) = \frac{\mu_{ij}^{y_{ij}}}{\beta^\alpha\Gamma(\alpha)y_{ij}!}\int_0^\infty u_i^{y_{ij}+\alpha-1}e^{-u_i(\mu_{ij}+1/\beta)}du_i \tag{55}$$

$$= \frac{\mu_{ij}^{y_{ij}} \cdot \Gamma(y_{ij}+\alpha)\left(\frac{1}{\mu_{ij}+1/\beta}\right)^{y_{ij}+\alpha}}{\beta^\alpha\Gamma(\alpha)y_{ij}!}\int_0^\infty \frac{u_i^{y_{ij}+\alpha-1}e^{-u_i(\mu_{ij}+1/\beta)}}{\Gamma(y_{ij}+\alpha)\left(\frac{1}{\mu_{ij}+1/\beta}\right)^{y_{ij}+\alpha}}du_i. \tag{56}$$

We can now utilize the fact, that any PDF integrates to 1 over its domain, i.e.,

$$\int_0^\infty \frac{u_i^{y_{ij}+\alpha-1}e^{-u_i(\mu_{ij}+1/\beta)}}{\Gamma(y_{ij}+\alpha)\left(\frac{1}{\mu_{ij}+1/\beta}\right)^{y_{ij}+\alpha}}du_i = 1 \quad \Rightarrow \tag{57}$$

$$f_{Y_{ij}}(y_{ij}, \mu_{ij}U_i) = \frac{\mu_{ij}^{y_{ij}} \cdot \Gamma(y_{ij}+\alpha)\left(\frac{1}{\mu_{ij}+1/\beta}\right)^{y_{ij}+\alpha}}{\beta^\alpha\Gamma(\alpha)y_{ij}!} \tag{58}$$

We are given that $\mathbb{E}[U_i] = 1$, since we know that $\mathbb{E}[U_i] = \alpha\beta$ we have that $\beta = 1/\alpha$. Insertion into the expression, we get

$$f_{Y_{ij}}(y_{ij}, \mu_{ij}U_i) = \frac{\mu_{ij}^{y_{ij}} \cdot \Gamma(y_{ij}+\alpha)\left(\frac{1}{\mu_{ij}+\alpha}\right)^{y_{ij}+\alpha}\alpha^{\alpha}}{\Gamma(\alpha)y_{ij}!} \tag{59}$$

$$= \frac{\Gamma(\alpha+y_{ij})}{\Gamma(\alpha)y_{ij}!}\mu_{ij}^{y_{ij}} \cdot \left(\frac{1}{\mu_{ij}+\alpha}\right)^{y_{ij}+\alpha}\alpha^{\alpha} \tag{60}$$

$$= \frac{\Gamma(\alpha+y_{ij})}{\Gamma(\alpha)y_{ij}!}\left(\frac{\alpha}{\mu_{ij}+\alpha}\right)^{\alpha}\left(\frac{\mu_{ij}}{\mu_{ij}+\alpha}\right)^{y_{ij}} \tag{61}$$

$$= \frac{\Gamma(\alpha+y_{ij})}{\Gamma(\alpha)y_{ij}!}\left(\frac{\alpha}{\mu_{ij}+\alpha}\right)^{\alpha}\left(1-\frac{\alpha}{\mu_{ij}+\alpha}\right)^{y_{ij}}. \tag{62}$$

Note that we took the liberty of re-writing the expression quite extensively, for reasons that will be very clear shortly.

C.3: Parameters for Negative Binomial

The reason we did the re-writing of the PDF was because we want to write the PDF in terms of a negative binomial distribution. Cf page 265 in Madsen [1], the PDF of a negative binomial can be written as

$$g(z, \alpha, p) = \frac{\Gamma(\alpha+z)}{\Gamma(\alpha)z!}p^{\alpha}(1-p)^{z} \tag{63}$$

where $\alpha > 0$ and $0 \leq p \leq 1$. We can now immediately see that we in fact have that

$$f_{Y_{ij}}(y_{ij}, \mu_{ij}U_i) = g(y_{ij}, \alpha, \frac{\alpha}{\mu_{ij}+\alpha}). \tag{64}$$

That is, the marginal distribution of our model actually follows a negative binomial distribution with parameters α and $\frac{\alpha}{\mu_{ij}+\alpha}$.

C.4: Marginal Likelihood

To reiterate, we found the marginal distribution of our model to be

$$f_{Y_{ij}}(y_{ij}, \mu_{ij}U_i) = \frac{\Gamma(\alpha+y_{ij})}{\Gamma(\alpha)y_{ij}!}\left(\frac{\alpha}{\mu_{ij}+\alpha}\right)^{\alpha}\left(1-\frac{\alpha}{\mu_{ij}+\alpha}\right)^{y_{ij}}. \tag{65}$$

Hence, the likelihood function is given by

$$L(\boldsymbol{\theta}; \boldsymbol{Y}) = \prod_{i\in\{subjId\}}\prod_{j\in\{1,2,\ldots,n_i\}}\frac{\Gamma(\alpha+y_{ij})}{\Gamma(\alpha)y_{ij}!}\left(\frac{\alpha}{\mu_{ij}+\alpha}\right)^{\alpha}\left(1-\frac{\alpha}{\mu_{ij}+\alpha}\right)^{y_{ij}}. \tag{66}$$

C.5: Parameter Estimates

As the negative binomial distribution is part of *exponential dispersion family*, we can estimate the model parameters with a *generalized linear model*. Unfortunately, the distribution is not available in the package GLM. Therefore, we use the function *glm.nb* from the package MASS [2].

	Male	Female
$\hat{\theta}_i$	-1.2452	-0.1542
$\hat{SE}$	0.2376	0.1424
$\exp(\hat{\theta}_i)$	0.2879	0.8571
$\hat{\alpha}$	3.9660	3.9660

Alternatively, we can optimize over the likelihood function as stated in (62). The log-likelihood function is then

$$\ell(\boldsymbol{\theta}; \boldsymbol{Y}) = \sum_{i \in \{subjId\}} \sum_{j \in \{1,2,\ldots,n_i\}} \log\left(\Gamma(\alpha + y_{ij})\right) - \log\left(\Gamma(\alpha) y_{ij}!\right) + \alpha \log\left(\frac{\alpha}{\mu_{ij} + \alpha}\right) + y_{ij} \log\left(1 - \frac{\alpha}{\mu_{ij} + \alpha}\right).$$

The *maximum likelihood estimates* $\hat{\boldsymbol{\theta}}$ are then

$$\ell(\hat{\boldsymbol{\theta}}; \boldsymbol{y}) = \min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} -\ell(\boldsymbol{\theta}; \boldsymbol{y}) \tag{67}$$

which in our case is carried out with the function *nlminb*. The code is shown below.

```
# Read data
clo = read.csv('dat_count3.csv', sep = ";", stringsAsFactors = F)
X = clo$clo

# Define log-likelihood function
l.fun <- function(param) {
  alpha = param[1]
  mu.male = param[2]
  mu.female = param[3]
  -sum(dnbinom(clo$clo,alpha,alpha/(mu.male*(clo$sex=="male")+
  mu.female*(clo$sex=="female")+alpha), log = T))
}

# Find infimum with nlminb
sol = nlminb(c(2,0.5,0.5), l.fun)

# Alpha
sol$par[1]

# Beta = 1/alpha s.t. E[U] = 1
1/sol$par[1]

# Male fixed effect
log(sol$par[2])

# Female fixed effect
log(sol$par[3])
```

C.6: Conditional Moments

The conditional mean and variance of the random effects can be found by *Bayes Theorem* [1, p. 229].

$$
\begin{aligned}
f_{U_i|Y_{ij}=y_{ij}}(u_i) &= \frac{f_{U_i,Y_{ij}}(u_i, y_{ij})}{f_{Y_{ij}}(y_{ij})} \\
&= \frac{f_{Y_{ij}|U_i=u_i}(y_{ij}) f_{U_i}(u_i)}{f_{Y_{ij}}(y_{ij})} \\
&= \frac{1}{\frac{\Gamma(\alpha+y_{ij})}{\Gamma(\alpha)y_{ij}!}\left(\frac{\alpha}{\mu_{ij}+\alpha}\right)^{\alpha}\left(1-\frac{\alpha}{\mu_{ij}+\alpha}\right)^{y_{ij}}} \cdot \frac{(\mu_{ij}u_i)^{y_{ij}}e^{-\mu_{ij}u_i}}{y_{ij}!} \cdot \frac{1}{\beta\Gamma(\alpha)}(\frac{u_i}{\beta})^{\alpha-1}e^{-u_i/\beta} \\
&\propto u_i^{y_{ij}+\alpha-1}\exp\left(-u_i(\mu_{ij}+\frac{1}{\beta})\right)
\end{aligned}
$$

We identify the last term as the *kernel* of a Gamma distribution with parameters $(y_{ij}+\alpha)$ and $\frac{1}{\mu_{ij}+\frac{1}{\beta}}$. Therefore, the moments are respectively

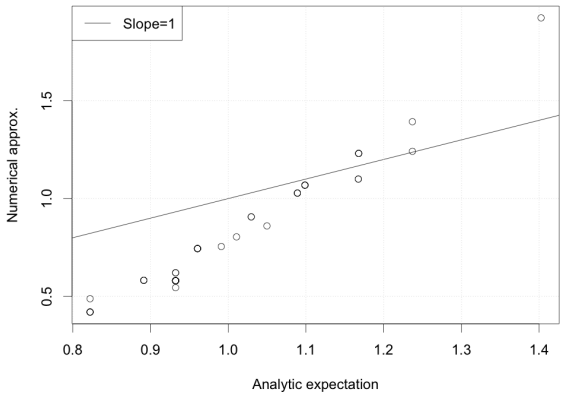
$$\mathbb{E}[U_i|Y_{ij}=y_{ij}] = \frac{y_{ij}+\alpha}{\mu_{ij}+\frac{1}{\beta}} \tag{68}$$

$$\mathbb{V}[U_i|Y_{ij}=y_{ij}] = \frac{y_{ij}+\alpha}{(\mu_{ij}+\frac{1}{\beta})^2} \tag{69}$$

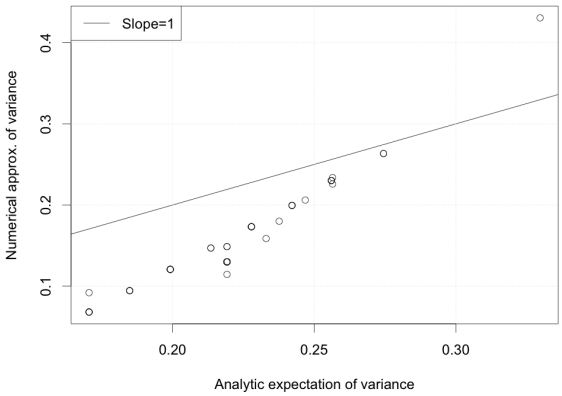
In the previous question, we used a Laplacian approximation to estimate parameters of the model. From the same optimization, we have estimates of the random effects for each subject. This allows us to compare the expectation of the random effect for the individual subject by means of the analytical expression above with the random effect that optimizes the Laplacian approximation of the likelihood function. Figure 10a shows a comparison between the analytical expectation of- and the numerical approximation of the random effects for each subject.

Similarly, we can use the Laplacian approximation of the likelihood function specifically the hessian at optimum, and use this as a estimation of the variance of the individual random effects. This allows us the compare the analytic expectation with the numerical approximation of the variance of the random effects. Figure 10b shows a comparison between the analytical expectation of- and the numerical approximation of the variance random effects for each subject.

Notice that there is a difference between the analytical expectations and the results obtained using numerical approximations. The main reason for this difference is that we in the numerical approximation describe the likelihood function by approximating 2. order Taylor expasions, i.e., we use a Laplacian approximation. For non-quadratic (so not Gaussian) likelihood functions this results in differences, simply because we might describe something asymmetrical with something symmetrical. With that being said, there is a obvious connection between the analytical and the numerical results, just not quite a 1:1 connection. Specifically notice that the numerical variance estimates are lower than the analytically obtained ones. This is because we use the maximum likelihood approximations of the variance in the numerical approach, which tend to underestimate the variance for low sample sizes.



(a) Comparison between analytic expectation and numerical approximation of the expectation of the random effects.



(b) Comparison between analytic expectation and numerical approximation of the variance of the random effects.

Figure 10

Problem D: Conclusion

In question A.2 we compare a Binomial-Gaussian model with a Poisson-Gaussian model. The main difference between the two is that the first models the probability of change upon inspection and the latter the constant rate of change. The parameters in the Binomial-Gaussian model corresponds to a probability of having changed clothes upon inspection. Assuming a inspection rate of 4/day, we can compare the rates obtained from the Poisson-Gaussian model with the Binomial-Gaussian model. Generally, we see that the Binomial-Gaussian model expects both men in women to change clothes less than the Poisson-Gaussian model.

In question B.2 we discussed the differences between using a Poisson-Gaussian and a Poisson-Gamma model. Specifically, we found that the parameter estimates in the Poisson-Gaussian model were relatively low—however, the estimated random effects generally increased the effective rate of the model.

By contrast, the parameter estimates for the Poisson-Gamma model using a Laplacian approximation of the likelihood function were high, relatively speaking. Here, the random effects generally decreased the effective rate. Whether one or the other is better than the other is difficult to determine. However, this demonstrates nicely the implication of choosing a suitable distribution for the random effect and how it can impact the model. Specifically, let us say a new subject arrived, then the ML estimate of their clothing rate would simply follow with $U = 0$ in the Poisson-Gaussian case and $U = 1$ in the Poisson-Gamma case. Consequently, the rate estimate will be larger for the Poisson-Gamma than the Poisson-Gaussian.

Finally, we derived an analytical expression for the Poisson-Gamma model by means of a negative binomial distribution. This allowed us to either use a library that could estimate the parameters for us, but also for us to fairly easily implement the likelihood function which we can use to estimate the parameters manually. Comparing the parameters obtained with the analytical derivation and the numerical approximation, we see that the parameters lie very close—indicating that the approximation works well. In question C.6 we took it a step further, and assessed the conditional mean and variance of the random effects. Here, we saw some difference between the analytical and numerical approach caused by the Laplacian approximation.

Table 3 shows an overview of the parameter estimates for the different models and approaches. Overall we saw in this report that both the choice of the first and second stage distribution in a hierarchical model plays an important role in dynamics of the model and hence must be chosen very carefully.

Table 3 – Comparison of Model Estimates.

Model	$(\alpha_{\text{female}}, \alpha_{\text{male}})$	σ_u^2	θ_i	(α, β)
Binomial-Gaussian	(-1.79, -3.14)	0.82	-	-
Poisson-Gaussian	(-0.34, -1.50)	0.46	-	-
Poisson-Gamma	-	-	(-1.23, -0.15)	(3.59, 0.28)
Negative Binomial	-	-	(-1.25, -0.15)	(3.97, 0.25)

References

[1] H. Madsen and P. Thyregod, *Introduction to General and Generalized Linear Models*, 2010.

[2] RDocumentation. Function 'glm.nb'. [Online]. Available: https://www.rdocumentation.org/packages/MASS/versions/7.3-57/topics/glm.nb

Appendix

Laplace Approximation

General Theory

The Laplace Approximation is also known as *Laplace's Method*. It is used to approximate complicated integrals by assuming that the integrand of $f(x)$ looks like a normal distribution.

$$\int_a^b e^{Mf(x)}dx \tag{70}$$

Then we can make a Taylor expansion around the global minimum x_0 of the function $f(x)$.

$$f(x) = \sum_{i=1}^{\infty} \frac{f^{(i)}(x_0)}{i!}(x - x_0)^i$$

Assuming $x_0 \in (a, b)$ and $M >> 0$ it can be justified that most of the contribution to the integral are in the proximity of x_0. Therefore, we can neglect higher order terms such that the function is well described by

$$\begin{aligned} f(x) &\approx P_2(x) \\ &= f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 \\ &= f(x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 \\ &= f(x_0) - \frac{1}{2}|f''(x_0)|(x - x_0)^2 \end{aligned}$$

Notice that $f'(x_0) = 0$ as x_0 is chosen to be the global minimum. Furthermore, as $f''(x_0) < 0$ the absolute values imply an opposite sign. Hence, (70) is approximately

$$\begin{aligned} \int_a^b e^{Mf(x)}dx &\approx \int_a^b e^{M(f(x_0) - \frac{1}{2}|f''(x_0)|(x-x_0)^2)}dx \\ &\approx \int_a^b e^{Mf(x_0) - M\frac{1}{2}|f''(x_0)|(x-x_0)^2}dx \\ &\approx e^{Mf(x_0)} \int_a^b e^{-M\frac{1}{2}|f''(x_0)|(x-x_0)^2}dx \end{aligned}$$

As most of the contribution comes in the proximity of x_0, we can reasonably make the following approximation.

$$\int_a^b e^{-M\frac{1}{2}|f''(x_0)|(x-x_0)^2}dx \approx \int_{-\infty}^{\infty} e^{-M\frac{1}{2}|f''(x_0)|(x-x_0)^2}dx \tag{71}$$

Now we use that the integral (71) is given by the following

$$\int_{-\infty}^{\infty} e^{-a(x+b)^2} = \sqrt{\frac{\pi}{a}} \tag{72}$$

which implies that the approximation is given by

$$\int_a^b e^{M(f(x_0)+\frac{1}{2!}f''(x_0)(x-x_0)^2)}dx \approx e^{Mf(x_0)}\sqrt{\frac{2\pi}{M|f''(x_0)|}} \tag{73}$$

Marginal Joint Likelihood

For a given set of parameters the joint log-likelihood $\ell(\boldsymbol{\beta}, \boldsymbol{\Psi}; \boldsymbol{u}, \boldsymbol{y}) = \log(L(\boldsymbol{\beta}, \boldsymbol{\Psi}; \boldsymbol{u}, \boldsymbol{y}))$. Let $f(x) = \ell(\boldsymbol{\beta}, \boldsymbol{\Psi}; \boldsymbol{u}, \boldsymbol{y})$. Then we can use *Laplace's Method* to show that

$$\begin{aligned}
\int_{\mathbb{R}^q} e^{Mf(\boldsymbol{\beta},\boldsymbol{\Psi};\boldsymbol{u},\boldsymbol{y})}d\boldsymbol{u} &= \int_{\mathbb{R}^q} (e^{\ell(\boldsymbol{\beta},\boldsymbol{\Psi};\boldsymbol{u},\boldsymbol{y})})^M d\boldsymbol{u} \\
&= \int_{\mathbb{R}^q} L(\boldsymbol{\beta}, \boldsymbol{\Psi}; \boldsymbol{u}, \boldsymbol{y})^M d\boldsymbol{u} \\
&= \int_{\mathbb{R}^q} \prod_{i=1}^{M} L(\boldsymbol{\beta}, \boldsymbol{\Psi}; \boldsymbol{u}, \boldsymbol{y}) d\boldsymbol{u} \\
&= \int_{\mathbb{R}^q} \prod_{i=1}^{M} f_{Y|U_i}(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{\beta}) f_{U_i}(\boldsymbol{u}_i; \alpha, \beta) d\boldsymbol{u}_i \\
&= \prod_{i=1}^{M} \int_{\mathbb{R}^q} f_{Y|U_i}(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{\beta}) f_{U_i}(\boldsymbol{u}_i; \alpha, \beta) d\boldsymbol{u}_i \\
&\approx e^{Mf(x_0)}\sqrt{\frac{2\pi}{M|f''(x_0)|}} \\
&\approx L_M(\boldsymbol{\beta}, \boldsymbol{\Psi}; \boldsymbol{y})
\end{aligned}$$

R-code

Part 1

```
setwd('/Users/mads/Google␣Drev/Skole/Uni/10_semester/02424/Assignment␣3')

#### Part 1 ----
conc.dat = read.csv('concrete.csv', sep = "")
conc.dat$date = as.Date(conc.dat$date)

## A.1 init plot
png('figures/y7-raw.png', width = 1000, height = 800, pointsize = 24)
cols = c('green3', 'blue3', 'orange', 'red2', 'steelblue2')
plot(conc.dat$date[conc.dat$batch == 1], conc.dat$y7[conc.dat$batch ==
    ↪ 1], col = 'green3',
     xlim = range(conc.dat$date), ylim = range(conc.dat$y7),
     xlab = "Time", ylab = "Strength␣(y7)")
lapply(2:5, function(i) points(conc.dat$date[conc.dat$batch == i], conc.
    ↪ dat$y7[conc.dat$batch == i], col = cols[i], pch = i))
grid(col = "darkgrey")
legend("topleft", paste('Batch',1:5), col = cols, pch = 1:5)
dev.off()
```

```
png('figures/y28-raw.png', width = 1000, height = 800, pointsize = 24)
plot(conc.dat$date[conc.dat$batch == 1], conc.dat$y28[conc.dat$batch ==
    ↪ 1], col = 'green3',
     xlim = range(conc.dat$date), ylim = range(conc.dat$y28),
     xlab = "Time", ylab = "Strength␣(y28)")
lapply(2:5, function(i) points(conc.dat$date[conc.dat$batch == i], conc.
    ↪ dat$y28[conc.dat$batch == i], col = cols[i], pch = i))
grid(col = "darkgrey")
legend("topleft", paste('Batch',1:5), col = cols, pch = 1:5)
dev.off()

## A.2
m7 = do.call(c,lapply(1:5, function(i) mean(conc.dat$y7[conc.dat$batch ==
    ↪  i])))
m28 = do.call(c,lapply(1:5, function(i) mean(conc.dat$y28[conc.dat$batch
    ↪ == i])))

png('figures/y7-means.png', width = 1000, height = 800, pointsize = 24)
plot(conc.dat$date[conc.dat$batch == 1], conc.dat$y7[conc.dat$batch ==
    ↪ 1], col = 'green3',
     xlim = range(conc.dat$date), ylim = range(conc.dat$y7),
     xlab = "Time", ylab = "Strength␣(y7)")
lapply(2:5, function(i) points(conc.dat$date[conc.dat$batch == i], conc.
    ↪ dat$y7[conc.dat$batch == i], col = cols[i], pch = i))
lapply(1:5, function(i) {
  lines(range(conc.dat$date[conc.dat$batch == i]), rep(m7[i],2), col =
      ↪ cols[i], lwd = 2)
  lines(rep(mean(range(conc.dat$date[conc.dat$batch == i])),2),m7[i]+sd(
      ↪ conc.dat$y7[conc.dat$batch == i])*c(-1, 1), lty = 2, col = cols[i
      ↪ ])})
grid(col = "darkgrey")
legend("topleft", paste('Batch',1:5), col = cols, pch = 1:5)
dev.off()

png('figures/y28-means.png', width = 1000, height = 800, pointsize = 24)
plot(conc.dat$date[conc.dat$batch == 1], conc.dat$y28[conc.dat$batch ==
    ↪ 1], col = 'green3',
     xlim = range(conc.dat$date), ylim = range(conc.dat$y28),
     xlab = "Time", ylab = "Strength␣(y28)")
lapply(2:5, function(i) points(conc.dat$date[conc.dat$batch == i], conc.
    ↪ dat$y28[conc.dat$batch == i], col = cols[i], pch = i))
lapply(1:5, function(i) {
  lines(range(conc.dat$date[conc.dat$batch == i]), rep(m28[i],2), col =
      ↪ cols[i], lwd = 2)
  lines(rep(mean(range(conc.dat$date[conc.dat$batch == i])),2),m28[i]+sd(
      ↪ conc.dat$y28[conc.dat$batch == i])*c(-1, 1), lty = 2, col = cols[
      ↪ i])})
grid(col = "darkgrey")
```

```
legend("topleft", paste('Batch',1:5), col = cols, pch = 1:5)
dev.off()

## A.3
## One-way model with random effects (def 5.2)
library(nlme)
fit0 = lme(y28 ~ 1, random = ~1|batch, data = conc.dat, method = 'ML');
    ↪ summary(fit0)


## A.4
conc.dat$air.aux = conc.dat$air.temp - mean(conc.dat$air.temp)
fit1 = lme(y28 ~ air.aux + 1, random = ~ (air.aux+1)|batch, data = conc.
    ↪ dat, method = 'ML'); summary(fit1)
#fit1 = lme(y28 ~ air.temp + 1, random = ~ (air.temp+1)|batch, data =
    ↪ conc.dat, method = 'ML'); summary(fit1)
anova(fit0,fit1)

library(car)
png('figures/batch-qqplot.png', width = 1000, height = 800, pointsize =
    ↪ 24)
qqPlot(fit1$residuals[,2], ylab = "Batch␣residual␣quantiles")
dev.off()

#### B
## B.5 (see page 195)
X = t(cbind(conc.dat$y7, conc.dat$y28))
xbar_ip = do.call(rbind, lapply(1:5, function(i) return(apply(X[,conc.dat
    ↪ $batch == i],1,mean))))
xbar_pp = apply(X,1,mean)

SSE = Reduce("+",lapply(1:5, function(i) (X[,conc.dat$batch == i] - xbar_
    ↪ ip[i,]) %*% t(X[,conc.dat$batch == i] - xbar_ip[i,])))
SSB = Reduce("+",lapply(1:5, function(i) sum(conc.dat$batch == i) * (xbar
    ↪ _ip[i,] - xbar_pp) %*% t(xbar_ip[i,] - xbar_pp)))
SST = SSE + SSB

(n0 = (nrow(conc.dat) - do.call(sum,lapply(1:5,function(i) sum(conc.dat$
    ↪ batch==i)^2/nrow(conc.dat))))/(5-1))
(mu_tilde = xbar_pp)
(Sigma_tilde = 1/(nrow(conc.dat)-5) * SSE)
(Sigma_tilde0 = 1/n0 * (SSB/(5-1) - Sigma_tilde))

Gamma = Sigma_tilde0 * solve(Sigma_tilde)
Gamma/(1+Gamma)

cov2cor(Sigma_tilde + Sigma_tilde0)
```

```
eps = do.call(cbind,lapply(1:5, function(i) X[, conc.dat$batch == i] -
    ↪ xbar_ip[i,]))
eps_group = t(xbar_ip) - xbar_pp
library(car)

png('figures/eps-within.png', width = 1000, height = 800, pointsize = 24)
plot(eps[1,conc.dat$batch==1], eps[2,conc.dat$batch==1], xlab = "Within␣
    ↪ batch␣errors␣(y7)", ylab = "Within␣batch␣errors␣(y28)",
     pch = 1, col = cols[1], xlim = c(-2,2), ylim=c(-4,4))
lapply(2:5, function(i) points(eps[1,conc.dat$batch==i], eps[2,conc.dat$
    ↪ batch==i], pch = i, col = cols[i]))
ellipse(c(0, 0), shape=Sigma_tilde, radius=1, col="red", lty=2, lwd = .5)
ellipse(c(0, 0), shape=Sigma_tilde, radius=1.5, col="red", lty=2, lwd =
    ↪ .5)
ellipse(c(0, 0), shape=Sigma_tilde, radius=2, col="red", lty=2, lwd = .5)
grid()
legend("topleft", paste('Batch',1:5), col = cols, pch = 1:5)
dev.off()

png('figures/rand-eff.png', width = 1000, height = 800, pointsize = 24)
plot(eps_group[1,1], eps_group[2,1], xlab = "Random␣effects␣(y7)", ylab =
    ↪  "Random␣effects␣(y28)",
     pch = 1, col = cols[1], xlim = c(-1,1), ylim=c(-4,4))
lapply(2:5, function(i) points(eps_group[1,i], eps_group[2,i], pch = i,
    ↪ col = cols[i]))
ellipse(c(0, 0), shape=Sigma_tilde0, radius=1, col="red", lty=2, lwd =
    ↪ .5)
ellipse(c(0, 0), shape=Sigma_tilde0, radius=1.5, col="red", lty=2, lwd =
    ↪ .5)
ellipse(c(0, 0), shape=Sigma_tilde0, radius=2, col="red", lty=2, lwd =
    ↪ .5)
grid()
legend("topleft", paste('Batch',1:5), col = cols, pch = 1:5)
dev.off()



#### Part 2
## clo data
clo = read.csv('dat_count3.csv', sep = ";", stringsAsFactors = F)

#### A ####
## Present data
pairs(clo[,-6])

png("subjId_box.png", width = 1000, height = 800, pointsize = 24)
plot(as.numeric(clo$clo) ~ as.factor(clo$subjId), col = 2+(clo$sex == "
    ↪ male"), lwd = 1.5,
```

```
    xlab = "subjId", ylab = "clo")
grid()
dev.off()

png("sex_box.png", width = 1000, height = 800, pointsize = 24)
plot(as.numeric(clo$clo) ~ as.factor(clo$sex), col = 2:3, lwd = 1.5,
    xlab = "sex", ylab = "clo")
grid()
dev.off()



plot(clo$clo[clo$subjId == ids[i]], clo$tOut[clo$subjId == ids[i]], col =
   ↪  i, pch = i, xlim = c(0,5), ylim = c(10,35))
lapply(min(ids):max(ids), function(i) points(clo$clo[clo$subjId == ids[i
   ↪ ]], clo$tOut[clo$subjId == ids[i]], col = i, pch = i))
grid()
clo$clo[clo$subjId == ids[i]]



## fit generalized mixed model
library(glmmTMB)

## binom
fit1TMB <- glmmTMB(cbind(clo,nobs-clo) ~ tInOp+tOut+as.factor(sex) + (1|
   ↪ subjId) - 1,
                   family="binomial",data=clo)
drop1(fit1TMB)
fit1TMB = update(fit1TMB,.~.-tOut)
drop1(fit1TMB)
fit1TMB = update(fit1TMB,.~.-tInOp)
drop1(fit1TMB)
summary(fit1TMB)

fit1.1TMB <- glmmTMB(cbind(clo,nobs-clo) ~ as.factor(sex) - 1,
                   family="binomial",data=clo)
AIC(fit1.1TMB)

## pois
fit2TMB <- glmmTMB(clo ~ tInOp+tOut+as.factor(sex) + (1|subjId) - 1,
                   family="poisson",data=clo)
drop1(fit2TMB)
fit2TMB = update(fit2TMB,.~.-tOut)
drop1(fit2TMB)
fit2TMB = update(fit2TMB,.~.-tInOp)
drop1(fit2TMB)
summary(fit2TMB)
```

```
fit2.1TMB <- glmmTMB(clo ~ as.factor(sex),
                  family="poisson",data=clo)
AIC(fit2.1TMB)


#### B ####
dat.count = read.csv('dat_count3.csv', sep = ";", stringsAsFactors = F)

X <- matrix(0,ncol=2,nrow=dim(dat.count)[1])
X[ ,1]<- dat.count$sex=="male"
X[ ,2]<-dat.count$sex=="female"

## Joint Likelihood
nll <- function(u,params,X){
  theta <- params[1:2] ## male, female
  alpha <- params[3]

  # Compute lambda for Poisson
  mu <- exp(X[,1]*theta[1] + X[,2]*theta[2])


  # Compute negative log-likelihood
  #first_stage <- -sum(dgamma(u[dat.count$subjId], shape = alpha, rate =
     ↪ beta, log = TRUE))
  first_stage <- -sum(dgamma(u, shape = alpha, scale = 1/alpha, log =
     ↪ TRUE))
  second_state <- -sum(dpois(dat.count$clo, mu*u[dat.count$subjId], log =
     ↪  TRUE))
  return(second_state + first_stage)
}

#####################################################
## use independence of u's in nlminb
nll.LA <- function(params,X){
  fun.tmp <- function(ui,u,params,X,i){
    u <- u*0+1
    u[i]<-ui
    nll(u,params,X)
  }
  u <- numeric(47)

  ## Use grouping structure
  for(i in 1:length(u)){
    u[i] <- nlminb(1,objective = fun.tmp, u=u,params=params,
                  X=X,i=i, lower = 0)$par
  }
  l.u <- nll(u,params,X)
```

```
  H <- numeric(length(u))
  for(i in 1:length(u)){
    H[i] <- hessian(func = fun.tmp, x = u[i], u=u,
                    params = params, X=X,i=i)}

  l.u + 0.5 * log(prod(H/(2*pi)))
}
system.time(fit <- nlminb(c(-1,0,2),nll.LA,X=X))
fit


#### C ####
## neg binom likelihood fun
l.fun <- function(param) {
  alpha = param[1]
  mu.male = param[2]
  mu.female = param[3]
  -sum(dnbinom(clo$clo,alpha,alpha/(mu.male*(clo$sex=="male")+mu.female*(
     ↪ clo$sex=="female")+alpha), log = T))
}

sol = nlminb(c(2,0.5,0.5), l.fun)
sol

(alpha = sol$par[1])
(beta = 1/alpha)
mu = sol$par[2:3] # male, female
(theta = log(mu))

sqrt(1/diag(hessian(l.fun, sol$par)))
sol$par


dat.count = read.csv('dat_count3.csv', sep = ";", stringsAsFactors = F)
uranus = sapply(1:47, function(i){
  dat.count = dat.count[dat.count$subjId == i,]
  X <- matrix(0,ncol=2,nrow=dim(dat.count)[1])
  X[ ,1]<- dat.count$sex=="male"
  X[ ,2]<-dat.count$sex=="female"

  mu_ij = exp(X[,1]*theta[1] + X[,2]*theta[2])
  ((mean(dat.count$clo) + alpha)/(mean(mu_ij) + alpha))
})

png("figures/analysnum_ex.png", width = 1000, height = 800, pointsize =
   ↪ 24)
plot(uranus, rand.eff$par, xlab = "Analytic␣expectation", ylab = "
   ↪ Numerical␣approx.")
```

```
abline(a=0,b=1)
grid()
legend("topleft",c("Slope=1"),lty =1)
dev.off()


uranus = sapply(1:47, function(i){
  dat.count = dat.count[dat.count$subjId == i,]
  X <- matrix(0,ncol=2,nrow=dim(dat.count)[1])
  X[ ,1]<- dat.count$sex=="male"
  X[ ,2]<-dat.count$sex=="female"

  mu_ij = exp(X[,1]*theta[1] + X[,2]*theta[2])
  ((mean(dat.count$clo) + alpha)/(mean(mu_ij) + alpha)^2)
})

png("figures/analvsnum_var.png", width = 1000, height = 800, pointsize =
    ↪ 24)
plot(uranus, var.numest, xlab = "Analytic␣expectation␣of␣variance", ylab
    ↪ = "Numerical␣approx.␣of␣variance")
abline(a=0,b=1)
grid()
legend("topleft",c("Slope=1"),lty =1)
dev.off()
```