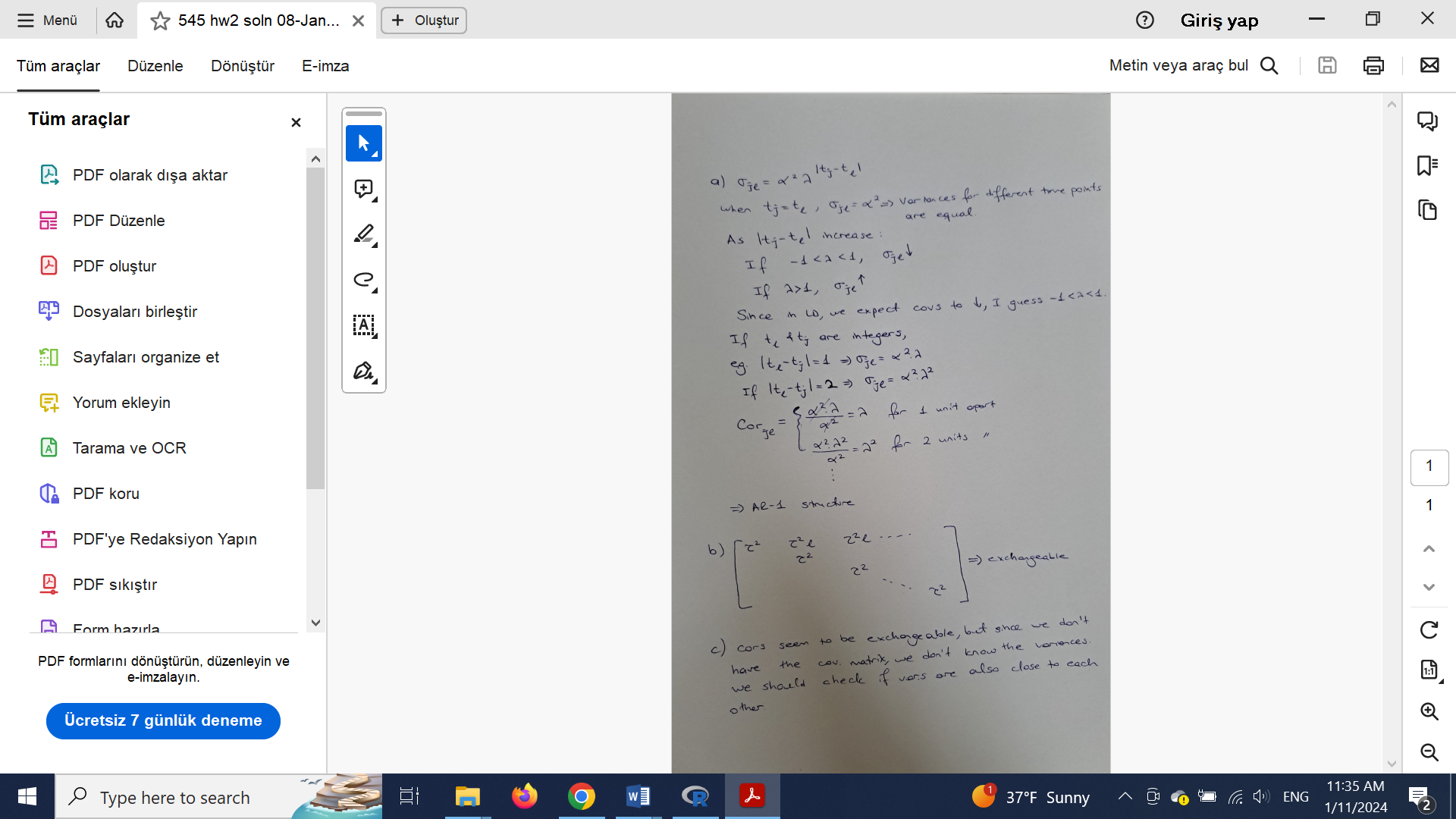
Dr. Özlem İLK DAĞ

Fall 2022-2023

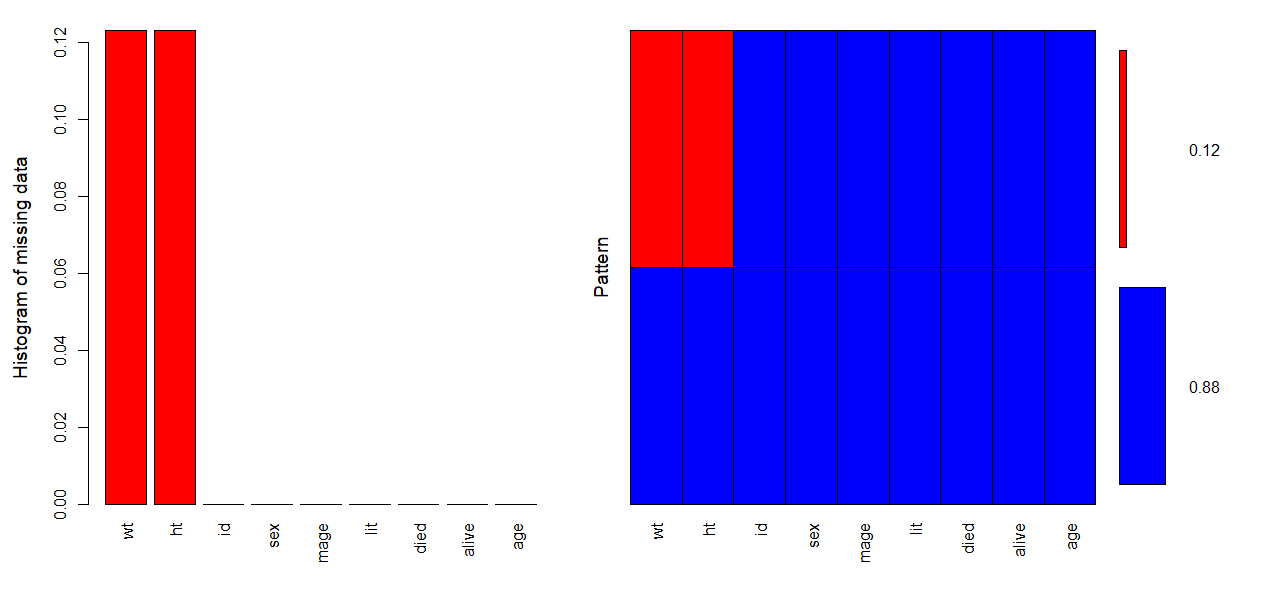
**STAT 561- PANEL DATA ANALYSIS**

**HOMEWORK 2 KEY**

1.



2. a) First, lets investigate the missingness pattern:



There are 123 (12%) missing observations in weight and height variables. As it can be seen from marginal plots (not given here) or tables below, the missingness seem to be non-ignorable. Table is only given for wt response, but we observe exactly the same for ht response, since both responses are missing for the same subjects.

Missing data analysis: **wt**  Not missing Missing p

sex 1 455 (85.0) 80 (15.0) 0.008

2 422 (90.8) 43 (9.2)

mage Mean (SD) 28.9 (6.2) 25.3 (5.4) <0.001

lit 0 848 (88.8) 107 (11.2) <0.001

29 (64.4) 16 (35.6)

died Mean (SD) 0.7 (1.0) 0.3 (0.7) <0.001

alive Mean (SD) 4.5 (2.5) 3.0 (1.8) <0.001

age Mean (SD) 38.0 (18.5) 35.3 (19.7) 0.133

b) I used ri (random indicator) since this method seems to work better for MNAR.

> ini <- mice(nepali, maxit = 0)

> pred <- ini$pred

> pred["wt", ] <- c(-2, 1, 0, 1, 1, 1, 1, 1, 1)

> pred["ht", ] <- c(-2, 1, 1, 0, 1, 1, 1, 1, 1)

> # 1 for fixed vrs, 2 for random variables, 0 for vr to impute, -2 for ID (ie class)

> meth <- ini$meth

> meth <- c("", "", "ri", "ri", "","","","","")

> # now lets run the updated mice with the appropriate $pred and $meth

> imp.upd <- mice(nepali, pred = pred, meth=meth, print = FALSE)

> nepali.imp=complete(imp.upd,2)

**Note that, your results might change slightly from now on, since we might be using slightly different imputed values.**

> head(nepali.imp,10)

id sex wt ht mage lit died alive age

1 120011 1 12.80000 91.2000 35 0 2 5 41

2 120011 1 12.80000 93.9000 35 0 2 5 45

3 120011 1 13.10000 95.2000 35 0 2 5 49

4 120011 1 13.80000 96.9000 35 0 2 5 53

5 120011 1 15.96407 100.7022 35 0 2 5 57

6 120012 2 14.90000 103.9000 35 0 2 5 57

7 120012 2 15.10000 106.5000 35 0 2 5 61

8 120012 2 15.80000 107.9000 35 0 2 5 65

9 120012 2 16.20000 108.7000 35 0 2 5 69

10 120012 2 15.20933 107.5982 35 0 2 5 73

c) > library(reshape2)

> nepali.imp.wide=dcast(nepali.imp, id ~ Followup.No, value.var="wt")

> cov(nepali.imp.wide[,-1])

1 2 3 4 5

1 9.508176 8.013625 8.090572 7.931563 7.542761

2 8.013625 7.817211 7.219130 7.140415 6.865938

3 8.090572 7.219130 7.564674 7.160875 6.873817

4 7.931563 7.140415 7.160875 7.515896 6.887968

5 7.542761 6.865938 6.873817 6.887968 7.659447

> cor(nepali.imp.wide[,-1])

1 2 3 4 5

1 1.0000000 0.9295109 0.9539718 0.9382526 0.8838592

2 0.9295109 1.0000000 0.9387801 0.9315521 0.8873097

3 0.9539718 0.9387801 1.0000000 0.9496872 0.9030342

4 0.9382526 0.9315521 0.9496872 1.0000000 0.9078248

5 0.8838592 0.8873097 0.9030342 0.9078248 1.0000000

I will assume exchangeable structure because variances and correlations seem to be close. I first fit a model with all covariates except ht (model 1); then removed insignificant ones (though I kept mage variable in although it was only significant at 10%, model 2). I also fitted a model with mage removed as well (model 3). QIC values are 1041.108, 1021.565, and 1018.004, respectively. Models 2 or 3 could be chosen. Model 3 has fewer parameters and smaller QIC, so it might be slightly better. Though mage in model 2 is significant, and the difference in QICs is slight, so using model 2 is also reasonable.

> m3=gee(wt~sex+lit+age,id=id,data=nepali.imp,family="gaussian",corstr="exchangeable")

> summary(m3)

…

Coefficients:

Estimate Naive S.E. Naive z Robust S.E. Robust z

(Intercept) 6.6948527 0.301428571 22.210412 0.304887338 21.958448

sex -0.4316937 0.175640643 -2.457823 0.177676519 -2.429661

lit 1.1307160 0.422492795 2.676297 0.223987835 5.048114

age 0.1368196 0.003338093 40.987345 0.003832901 35.696096

fitted weight = 6.69 -0.43 sex + 1.13 lit + 0.14 age

On the average females (sex=2) have less weight compared to males (sex=1) as expected. Children whose mother are literate (lit=1) have higher average weights. As the age of the children increases we expect an increase in the weight, given all other covariates are fixed. All the coefficients have the expected sign.

> shapiro.test(m3$res)

Shapiro-Wilk normality test

data: m3$res

W = 0.99636, p-value = 0.01981

Residuals from model 3 are not from normal if I take alpha 5%. But, if I take it 1%, it may be safe to assume normality. For simplicity, I am assuming normality is not a big issue here and move on. But, if you would like you can try a transformation on wt to make things a little bit more ‘normal’. Alternatively, you can search if wt follows one of the distributions available in ‘family’, for example, gamma.

d) > nepali.imp.Yt=nepali.imp[nepali.imp$Fol>1,]

> nepali.imp.Ytm1=nepali.imp[nepali.imp$Fol<5,]

> nepali.imp.trans=cbind(nepali.imp.Yt,nepali.imp.Ytm1$wt)

I used the same covariates as in marginal model together with lag 1, and then dropped the insignificant ones.

>m5=gee(wt~age+nepali.imp.Ytm1$wt,id=id,data=nepali.imp.trans,family="gaussian",corstr="independence")

> summary(m5)

Coefficients:

Estimate Naive S.E. Naive z Robust S.E. Robust z

(Intercept) 2.18530185 0.158420695 13.794295 0.275559565 7.930416

age 0.03009045 0.003869511 7.776294 0.006257944 4.808361

nepali.imp.Ytm1$wt 0.74249493 0.024393156 30.438658 0.045502530 16.317662

fitted weight = 2.19 + 0.03 \* age + 0.74 \* weight in the previous follow-up

As expected, there is a positive relation between weight and age, as well as between weights at two different follow-ups. Note that, the time spaces between different follow-up times are not always equal (sometimes 3 months, sometimes 4 or 5 months…). So, transition model may not be measuring the correlation between weight and lagged weight correctly, but I hope it was a good practice of the model for you.

e) > library(lme4)

> nepali\_lmer1 <- lmer(wt~sex+lit+age+ (1 | id),data=nepali.imp)

> summary(nepali\_lmer1)

REML criterion at convergence: 2876.9

Random effects:

Groups Name Variance Std.Dev.

id (Intercept) 1.4128 1.1886

Residual 0.6227 0.7891

Number of obs: 1000, groups: id, 200

Fixed effects:

Estimate Std. Error t value

(Intercept) 6.253077 0.175104 35.711

sex -0.419649 0.176516 -2.377

lit 1.095789 0.424602 2.581

age 0.136621 0.003312 41.246

> ranef(nepali\_lmer1)

$id

(Intercept)

120011 0.662829562

120012 0.659372231

120021 0.680634583

120022 1.129390583

…

fitted weight for subject 1120011 = 6.25-0.42 sex + 1.1 lit + 0.14 age + 0.66

Fixed effects have similar interpretations with marginal model. In addition to these, now we have random coefficients. We see that (from ranef function) random coefficients are close to each other and not too far away from 0. Also, the variance of random coefficients (1.4128) is not too big. So, there is not too much variability from one child to another in the data. Hence using marginal model might be enough for this dataset.

I tried to fit another model with a random slope for age as well, but got a warning message saying it didn’t converge.

> nepali\_lmer2 <- lmer(wt~sex+lit+age+ (age | id),data=nepali.imp)

**Warning message:**

In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :

Model failed to converge with max|grad| = 0.0170336 (tol = 0.002, component 1)

If you insist, it will give you the results. Be careful! These results are not trustworthy and you shouldn’t use this model, since convergence is not satisfied.

> summary(nepali\_lmer2)

REML criterion at convergence: 2842.7

Random effects:

Groups Name Variance Std.Dev. Corr

id (Intercept) 0.950193 0.97478

age 0.001008 0.03174 -0.50

Residual 0.581612 0.76264

Number of obs: 1000, groups: id, 200

Fixed effects:

Estimate Std. Error t value

(Intercept) 6.207335 0.153191 40.520

sex2 -0.408583 0.157295 -2.598

lit1 1.124180 0.370678 3.033

age 0.140231 0.003831 36.609

**optimizer (nloptwrap) convergence code: 0 (OK)**

**Model failed to converge with max|grad| = 0.0170336 (tol = 0.002, component 1)**

f) I splitted the data into 2: train and test. You could divide into 3: train, validation and test. This would be better, since we can use validation set for tuning some hyperparameters. But, I don’t have many options to change in REEMtree, so I go with simpler option for a hw. For an article or thesis, using 3 sets might save some headaches.

> library(groupdata2)

> library(xgboost)

> library(dplyr)

> nepali.imp$id = factor(nepali.imp$id) # partition requires id\_col to have factor

> test.train.d=partition(nepali.imp,p = 0.2, id\_col = "id")

> test.d=test.train.d %>%

+ .[1] # See only the test set

> dim(test.d[[1]])

[1] 200 10

> train.d=test.train.d %>%

+ .[2] # See only the train set

> dim(train.d[[1]])

[1] 800 10

> library(LongituRF)

> X.fixed.effects <- as.data.frame(train.d[[1]][,c(5,6,7,8,9)])

> y=as.matrix(train.d[[1]]$wt)

> z=cbind(rep(1,dim(train.d[[1]])[1]))

> ml1 <- LongituRF::REEMtree(Y=y,X=X.fixed.effects,Z=z,id=train.d[[1]]$id,time=train.d[[1]]$Followup.No,sto="BM")

> ml1$forest

n= 800

node), split, n, deviance, yval

\* denotes terminal node

1) root 800 3820.231000 11.270850

2) age< 34.5 346 427.913200 9.096929

4) age< 14.5 94 52.928430 7.711527 \*

5) age>=14.5 252 121.700900 9.619481

10) age< 30.5 210 58.318780 9.411689 \*

11) age>=30.5 42 4.388648 10.702400 \*

3) age>=34.5 454 510.959300 12.927620

6) age< 48.5 183 76.274250 11.910010

12) age< 39.5 69 7.322800 11.241460 \*

13) age>=39.5 114 20.265600 12.301670 \*

7) age>=48.5 271 117.214900 13.614800

14) age< 62.5 179 37.182500 13.266180 \*

15) age>=62.5 92 18.899730 14.274980 \*

> ml1$forest$variable.importance

age mage alive died

3620.924242 540.535106 409.788438 8.083527

Age of the child is a dominating effect in predicting the weight of the child. So, when this variable is available and added to the analysis, we don’t need any other variable.

MSEs:

> p.train = predict(ml1$forest, X=X.fixed.effects,Z=z,id=train.d[[1]]$id, time=train.d[[1]]$Followup.No)

> p.test=predict(ml1$forest, test.d[[1]],X=as.data.frame(test.d[[1]][,c(5,6,7,8,9)]),Z=cbind(rep(1,dim(test.d[[1]])[1])),id=test.d[[1]]$id, time=test.d[[1]]$Followup.No)

> sum((train.d[[1]]$wt-p.train)\*\*2)/(dim(train.d[[1]])[1]-dim(X.fixed.effects)[2])

[1] 2.497518

> sum((test.d[[1]]$wt-p.test)\*\*2)/(dim(test.d[[1]])[1]-dim(X.fixed.effects)[2])

[1] 2.065153

MSE for train data is 2.5, while it is 2.07 for test data. These numbers seem small, but may not suggest much alone. We can use them to compare with other models.

I also tried a tree without age. Both of these models use only random intercept, no random slope.

> X.fixed.effects <- as.data.frame(train.d[[1]][,c(5,6,7,8)])

> ml2 <- LongituRF::REEMtree(Y=y,X=X.fixed.effects,Z=z,id=train.d[[1]]$id,time=train.d[[1]]$Followup.No,sto="BM")

[1] "stopped after 10 iterations."

> ml2$forest

n= 800

node), split, n, deviance, yval

\* denotes terminal node

1) root 800 2391.0880000 10.549120

2) mage< 23.5 170 287.7903000 8.841804

4) alive< 2.5 115 29.6769900 8.199797 \*

5) alive>=2.5 55 117.0560000 10.158970

10) died>=0.5 20 0.6523506 8.237797 \*

11) died< 0.5 35 1.1457370 11.250540 \*

3) mage>=23.5 630 1474.0460000 11.009820

6) died< 3.5 620 1180.9000000 10.923280

12) mage< 38.5 575 1000.6860000 10.774100

24) alive>=1.5 560 881.3643000 10.699950

48) died>=1.5 110 333.9896000 10.119160

96) mage< 31 60 110.7373000 9.097628

192) alive>=7 10 0.7429474 6.618362 \*

193) alive< 7 50 37.1925100 9.590247

386) alive< 4.5 20 8.3501240 8.708457 \*

387) alive>=4.5 30 3.3646430 10.163130 \*

97) mage>=31 50 85.5064600 11.345000

194) alive>=6.5 40 1.6445070 10.695050 \*

195) alive< 6.5 10 0.6778990 13.920740 \*

49) died< 1.5 450 501.1996000 10.841920

98) died< 0.5 315 301.3634000 10.564390

196) mage< 26.5 105 91.5314800 9.799782

392) alive>=2.5 60 6.5266060 9.024395 \*

393) alive< 2.5 45 2.3673040 10.814510 \*

197) mage>=26.5 210 117.7531000 10.946700

394) mage>=33.5 60 3.1451850 10.436460 \*

395) mage< 33.5 150 93.1711100 11.148770

790) alive< 4.5 115 25.5667000 11.068140 \*

791) alive>=4.5 35 49.5661200 11.777350

1582) mage< 30.5 25 1.2030490 10.486860 \*

1583) mage>=30.5 10 0.3555231 13.620710 \*

99) died>=0.5 135 118.9611000 11.489500

198) mage>=27 105 10.5757500 11.004150 \*

199) mage< 27 30 0.8626058 13.155460 \*

25) alive< 1.5 15 1.2926150 13.537150 \*

13) mage>=38.5 45 3.9191400 12.823520 \*

7) died>=3.5 10 0.5985037 16.373030 \*

> ml2$forest$variable.importance

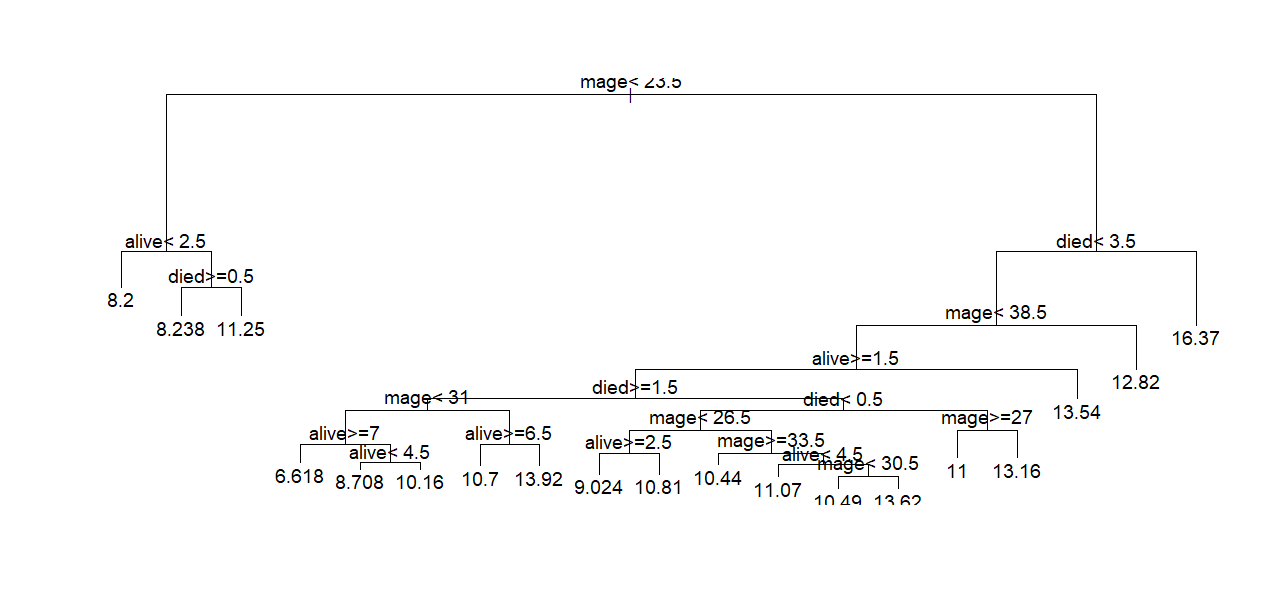
mage alive died

1303.1408 1038.0073 598.8882

Now, mother’s age, the number of children the mother has ever had born alive, and the number of children the mother has had that died are found important in predicting child’s weight.

> plot(ml2$forest)

> text(ml2$forest, use.n=F, all=F, cex=1.2)



When the mother’s age is less than 23.5 and the mother has less than 3 kids alive with no dead kids, we expect the average weight of child to be 8.238 kg. Under similar conditions, but if the mother has at least 1 dead kid, then the average weight is predicted as 11.25 kg. If mother is older than 23.5 years and has more than 4 dead kids, then average weight of kid is expected to be 16.37 kg. There are only 10 subjects with such property, though (not shown in plot to avoid even more crowded look).

> p.train = predict(ml2$forest, X=X.fixed.effects,Z=z,id=train.d[[1]]$id, time=train.d[[1]]$Followup.No)

> p.test=predict(ml2$forest, test.d[[1]],X=as.data.frame(test.d[[1]][,c(5,6,7,8)]),Z=cbind(rep(1,dim(test.d[[1]])[1])),id=test.d[[1]]$id, time=test.d[[1]]$Followup.No)

> sum((train.d[[1]]$wt-p.train)\*\*2)/(dim(train.d[[1]])[1]-dim(X.fixed.effects)[2])

[1] 6.989759

> sum((test.d[[1]]$wt-p.test)\*\*2)/(dim(test.d[[1]])[1]-dim(X.fixed.effects)[2])

[1] 10.08849

When age is removed in a hope to see interesting patterns, MSEs for train and test data have increased.