Featback analyses: SSL

Data preparation

Primaire uitkomstmaat

- Eetstoornispathologie (EDE-Q_TOT)

Secundaire uitkomstmaten

- Self-efficacy (GSES_TOT)
- Angst en depressie (PHQ_TOT)
- Sociale steun (SSL_TOT)

Conditie indicatoren

- 1 = Featback
- 2 = Featback + ondersteuning van een ervaringsdeskundige via chat of email
- 3 = Ondersteuning van een ervaringsdeskundige via chat of email
- 4 = Wachtlijst controle conditie (hen werd conditie 2 aangeboden na 12 maanden + 8 weken wachttijd)

Tijdsindicatoren

- 1 = baseline
- 2 = post-interventie (8 weken)
- 3 = 3 maanden follow-up (i.e., 3 maanden + 8 weken)
- 4 = 6 maanden FU
- 5 = 9 maanden FU
- 6 = 12 maanden FU

Mogelijke moderatoren

- Leeftijd (Age)
- Educatie (moet nog omgezet worden naar 3 levels; laag, middel, hoog) (T0_edu)
- Behandelgeschiedenis (T0_treatment)
- BMI bij Baseline (BMI)
- Eetstoornispathologie bij baseline (EDEQ_TOT)
- (Evt. duur eetstoornis, maar is zeer hoog gecorreleerd met leeftijd; T0_yrsED)
- Motivatie om te veranderen bij baseline (T0_Motiv_TOT)
- Zelfwaardering bij baseline (T0_RSES_TOT)
- Self-efficacy bij baseline (GSES_TOT)
- Angst en depressie bij baseline (PHQ_TOT)
- Type eetstoornis bij baseline (staat nog niet tussen de variabelen; nog even kijken hoe we dit moeten aanpakken . . .)

Toevoeging 06-08-2021:

- T0_EDEQ_ObjEet_Keer (= aantal (objectieve) eetbuien in de afgelopen 28 dagen)
- T0_SSL_TOT

```
## Read in data
library("foreign")
data <- read.spss("20210316_Featback_AllMerged_LongFormat_imputed.sav", to.data.frame = TRUE)
#names(data)
```

```r
## Set aprropriate variable classes
data$ID <- factor(data$ID)
data$T0_edu <- ordered(data$T0_edu)
## Abbreviate condition levels
levels(data$Condition)[levels(data$Condition)=="Waiting list"] <- "WL"
levels(data$Condition)[levels(data$Condition)=="Featback"] <- "Fb"
levels(data$Condition)[levels(data$Condition)=="Featback + expert-patient support"] <- "Fb+eps"
levels(data$Condition)[levels(data$Condition)=="expert-patient support"] <- "eps"
data$Condition <- factor(data$Condition, levels = c("WL", "eps", "Fb", "Fb+eps"))
## Check if every subject has a time 1
all(table(data$ID, data$Time == 1)[,2] == 1L)
```

```
## [1] FALSE
```

```r
## Select only pre- and post assessments
#data <- data[data$Time %in% 1:2, ]
#data$Time <- factor(data$Time)
## Construct T0 variables
for (i in unique(data$ID)) {
  data$T0_EDEQ_TOT[data$ID == i] <- data$EDEQ_TOT[data$ID == i & data$Time == 1]
  data$T0_BMI[data$ID == i] <- data$BMI[data$ID == i & data$Time == 1]
  data$T0_GSES_TOT[data$ID == i] <- data$GSES_TOT[data$ID == i & data$Time == 1]
  data$T0_PHQ_TOT[data$ID == i] <- data$PHQ_TOT[data$ID == i & data$Time == 1]
  data$T0_SSL_TOT[data$ID == i] <- data$SSL_TOT[data$ID == i & data$Time == 1]
  data$T0_eetbuien[data$ID == i] <- data$EDEQ_ObjEet_Keer[data$ID == i & data$Time == 1]
}
## Check for missings and remove
unlist(sapply(data[ , c("Age", "T0_edu", "T0_BMI", "T0_EDEQ_TOT", "T0_Motiv_TOT",
                        "T0_GSES_TOT", "T0_PHQ_TOT")], function(x) table(is.na(x))))
```

```
##          Age.FALSE       T0_edu.FALSE        T0_edu.TRUE       T0_BMI.FALSE
##             215130             215124                  6             215124
##        T0_BMI.TRUE  T0_EDEQ_TOT.FALSE   T0_EDEQ_TOT.TRUE T0_Motiv_TOT.FALSE
##                  6             215112                 18             215112
##  T0_Motiv_TOT.TRUE  T0_GSES_TOT.FALSE   T0_GSES_TOT.TRUE   T0_PHQ_TOT.FALSE
##                 18             215112                 18             215130
```

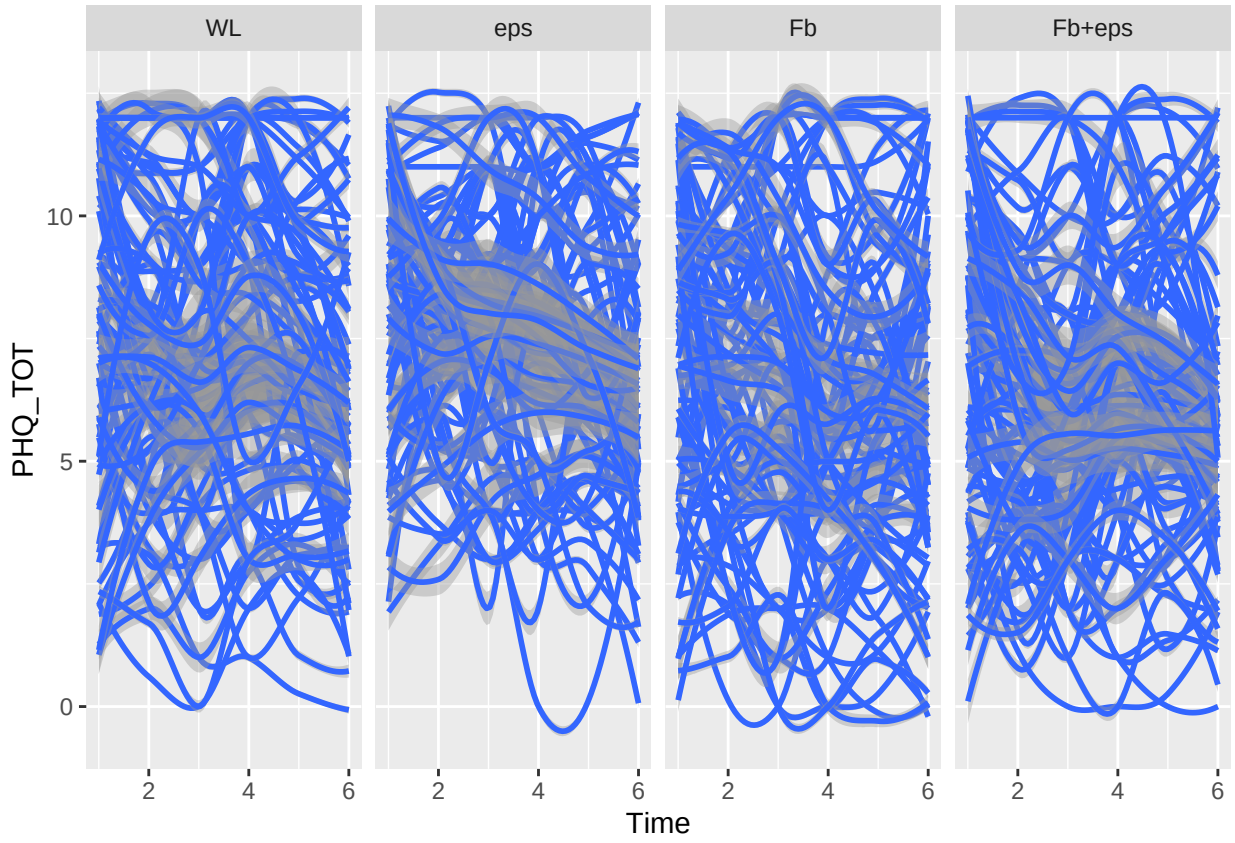
```r
data <- data[!is.na(data$T0_BMI), ]
data <- data[!is.na(data$T0_EDEQ_TOT), ]
data <- data[!is.na(data$T0_edu), ]
data <- data[!is.na(data$T0_yrsED), ]
unlist(sapply(data[ , c("Age", "T0_edu", "T0_BMI", "T0_EDEQ_TOT", "T0_Motiv_TOT",
                        "T0_GSES_TOT", "T0_PHQ_TOT")], function(x) table(is.na(x))))
```

```
##          Age.FALSE       T0_edu.FALSE       T0_BMI.FALSE  T0_EDEQ_TOT.FALSE
##             215058             215058             215058             215058
## T0_Motiv_TOT.FALSE  T0_Motiv_TOT.TRUE  T0_GSES_TOT.FALSE   T0_PHQ_TOT.FALSE
##             215043                 15             215058             215058
```

```r
## Exploratory plot
library("ggplot2")
ggplot(data = data, aes(x = Time, y = PHQ_TOT, group = ID)) +
  facet_grid(. ~ Condition) + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning: Removed 477 rows containing non-finite values (stat_smooth).
```



De trajecten lijken niet erg lineair.

Splits based only on treatment-time-subgroup interactions

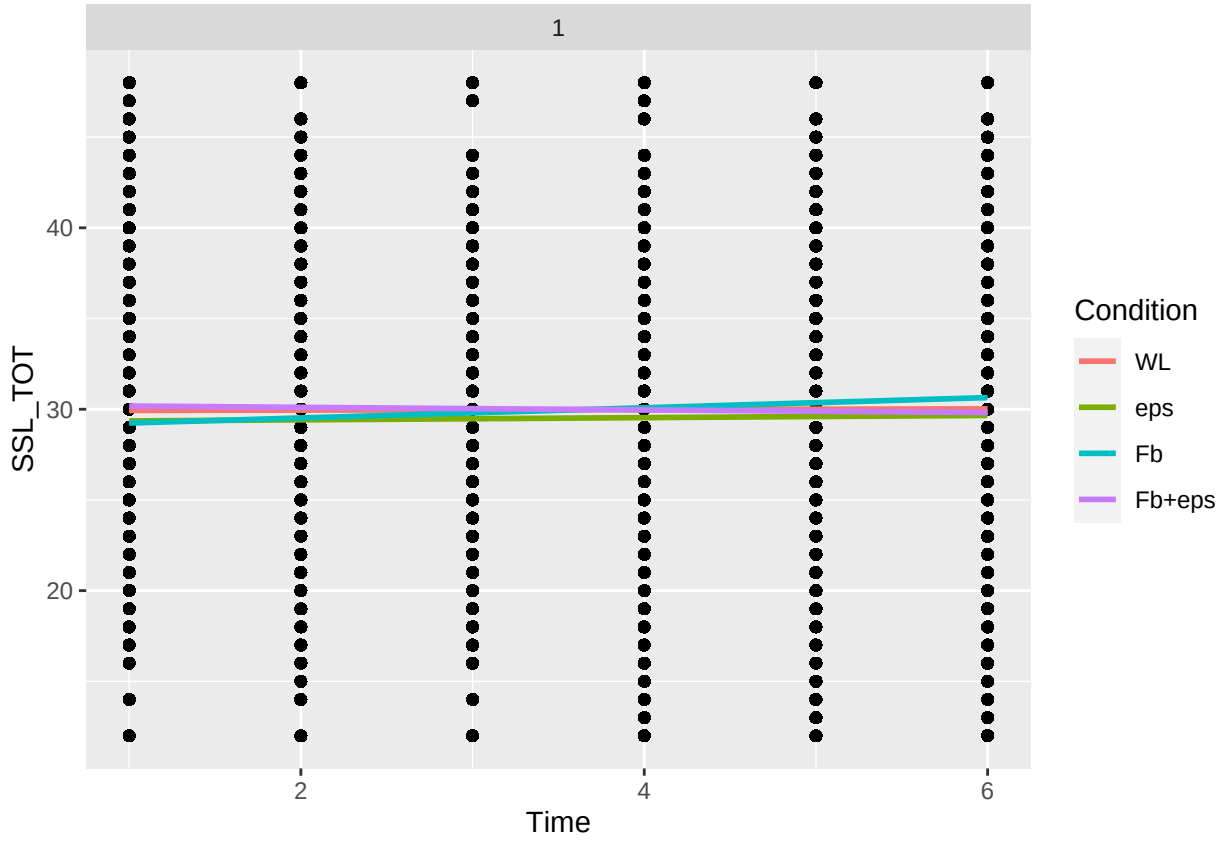
```
library("glmertree")
library("lmerTest")
library("strucchange")
dat1 <- data[!is.na(data$SSL_TOT),]
lt1 <- lmertree(SSL_TOT ~ Time*Condition | (1|ID) | Age + T0_edu + T0_BMI + T0_EDEQ_TOT +
                  T0_Motiv_TOT + T0_GSES_TOT + T0_PHQ_TOT + T0_treatment +  T0_yrsED +
                  T0_SSL_TOT + T0_eetbuien,
                data = dat1, cluster = ID, parm = 6:8, maxdepth = 4L)
if (length(lt1$tree) > 1L) {
  plot(lt1, which = "tree", type = "simple", fitted = "marginal", gp = gpar(cex = .5))
  #fixef(lt1)
  apply(fixef(lt1), 2, sd)
  VarCorr(lt1)
  tmp <- summary(lmer(attr(lt1$lmer, "call")$formula, data = lt1$data))
  round(tmp$coefficients[tmp$coefficients[,"Pr(>|t|)"] < 0.05, -3], digits = 4)
} else {
  sctest(lt1$tree)
}
```

```
##                 Age    T0_edu     T0_BMI T0_EDEQ_TOT T0_Motiv_TOT T0_GSES_TOT
## statistic 3.212595 14.75395 12.7361413    4.633832      2.17896   8.7160771
```

```
## p.value   1.000000  1.00000  0.6751708    1.000000      1.00000   0.9943708
##           T0_PHQ_TOT T0_treatment T0_yrsED T0_SSL_TOT T0_eetbuien
## statistic  6.3810384    2.9575271 3.818895   9.583726    3.379898
## p.value    0.9999964    0.9962519 1.000000   0.976356    1.000000
```

```
ggplot(data = lt1$data, aes(x = Time, y = SSL_TOT, group = ID)) + geom_point() +
  facet_grid(. ~ .tree) + geom_smooth(method = "lm", se = FALSE,
aes(x = Time, y = SSL_TOT, group = Condition, color = Condition))
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
dat2 <- dat1[dat1$Time %in% 1:2, ]
lt3 <- lmertree(SSL_TOT ~ Time*Condition | (1|ID) | Age + T0_edu + T0_BMI + T0_EDEQ_TOT +
                  T0_Motiv_TOT + T0_GSES_TOT + T0_PHQ_TOT + T0_treatment + T0_yrsED +
                  T0_SSL_TOT + T0_eetbuien,
                data = dat2, cluster = ID, parm = 6:8, maxdepth = 4L)
```

```
## Warning in lmertree(SSL_TOT ~ Time * Condition | (1 | ID) | Age + T0_edu + :
## 'data' contains missing values, note that listwise deletion will be employed.
```

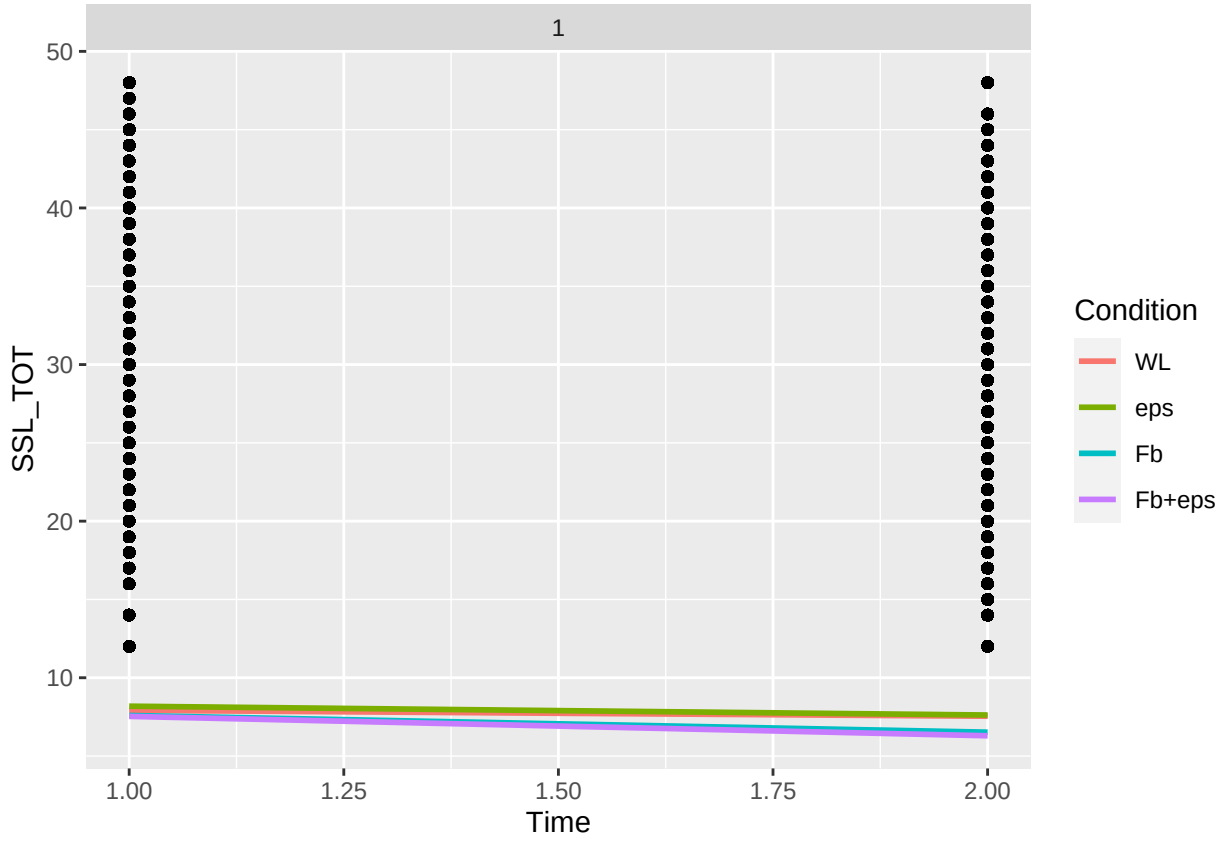
```
if (length(lt3$tree) > 1L) {
  plot(lt3, which = "tree", fitted = "marginal", gp = gpar(cex = .7))
  #fixef(lt3)
  apply(fixef(lt3), 2, sd)
  VarCorr(lt3)
  tmp <- summary(lmer(attr(lt3$lmer, "call")$formula, data = lt3$data))
  round(tmp$coefficients[tmp$coefficients[,"Pr(>|t|)"] < 0.05, -3], digits = 4)
} else {
```

```
  sctest(lt3$tree)
}
```

```
##                 Age     T0_edu     T0_BMI T0_EDEQ_TOT T0_Motiv_TOT T0_GSES_TOT
## statistic 7.2848711 19.2704570 14.5821287  11.2719323     2.089027   7.3084053
## p.value   0.9998582  0.9999996  0.4176365   0.8617574     1.000000   0.9998466
##           T0_PHQ_TOT T0_treatment   T0_yrsED T0_SSL_TOT T0_eetbuien
## statistic   3.712911    2.6065450 11.0870537  6.8909168   9.9394895
## p.value     1.000000    0.9987738  0.8803574  0.9999661   0.9622441
```

```
ggplot(data = lt3$data, aes(x = Time, y = SSL_TOT, group = ID)) + geom_point() +
  facet_grid(. ~ .tree) +
  geom_smooth(method = "lm", se = FALSE,
              aes(x = Time, y = PHQ_TOT, group = Condition, color = Condition))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Waitlist versus all treatment groups

```
dat1$Condition2 <- factor(dat1$Condition == "WL")
lt2 <- lmertree(SSL_TOT ~ Time*Condition2 | (1|ID) | Age + T0_edu + T0_BMI + T0_EDEQ_TOT +
                  T0_Motiv_TOT + T0_GSES_TOT + T0_PHQ_TOT + T0_treatment +  T0_yrsED +
                  T0_SSL_TOT + T0_eetbuien,
                data = dat1, cluster = ID, parm = 4, verbose = TRUE, maxdepth = 4)
```

```
## 'log Lik.' -627714.1 (df=6)
```

```
## 'log Lik.' -627714.1 (df=6)
```

```
if (length(lt2$tree) > 1L) {
  plot(lt2, which = "tree", fitted = "marginal", gp = gpar(cex = .5))
  #fixef(lt2)
  apply(fixef(lt2), 2, sd)
  VarCorr(lt2)
  tmp <- summary(lmer(attr(lt2$lmer, "call")$formula, data = lt1$data))
  round(tmp$coefficients[tmp$coefficients[,"Pr(>|t|)"] < 0.05, -3], digits = 4)
} else {
  sctest(lt2$tree)
}
```

```
##                Age     T0_edu   T0_BMI T0_EDEQ_TOT T0_Motiv_TOT T0_GSES_TOT
## statistic 1.990454 11.5672224 3.655754     1.51911     1.099828    1.159244
## p.value   1.000000  0.8738927 0.999462     1.00000     1.000000    1.000000
##           T0_PHQ_TOT T0_treatment  T0_yrsED T0_SSL_TOT T0_eetbuien
## statistic   4.726468    2.5428417 2.5691632  4.3731580     1.65689
## p.value     0.987215    0.7251981 0.9999992  0.9944749     1.00000
```

```
dat2 <- dat1[dat1$Time %in% 1:2, ]
lt4 <- lmertree(SSL_TOT ~ Time*Condition2 | (1|ID) | Age + T0_edu + T0_BMI + T0_EDEQ_TOT +
                  T0_Motiv_TOT + T0_GSES_TOT + T0_PHQ_TOT + T0_treatment + T0_yrsED +
                  T0_SSL_TOT + T0_eetbuien,
                data = dat2, cluster = ID, parm = 4, maxdepth = 4)
if (length(lt4$tree) > 1L) {
  plot(lt4, which = "tree", fitted = "marginal", gp = gpar(cex = .7))
  #fixef(lt4)
  apply(fixef(lt4), 2, sd)
  VarCorr(lt4)
  tmp <- summary(lmer(attr(lt4$lmer, "call")$formula, data = lt3$data))
  round(tmp$coefficients[tmp$coefficients[,"Pr(>|t|)"] < 0.05, -3], digits = 4)
} else {
  sctest(lt4$tree)
}
```

```
##                 Age    T0_edu     T0_BMI T0_EDEQ_TOT T0_Motiv_TOT T0_GSES_TOT
## statistic 8.6386290 2.992161 11.7645670    1.914564     1.629332    1.582679
## p.value   0.4939073 1.000000  0.1441475    1.000000     1.000000    1.000000
##           T0_PHQ_TOT T0_treatment   T0_yrsED T0_SSL_TOT T0_eetbuien
## statistic   1.626673    2.7031307 12.3223994   4.880923   7.5871392
## p.value     1.000000    0.6867696  0.1124665   0.982407   0.6721118
```