Statistical Analysis of Performance Data – Linear Group

Thursday 16 January, 2020

Methods and Software

Statistical analysis was based on the Generalized Estimating Equations method (Liang and Zeger 1986) as this is implemented in the R (R Core Team 2019) package geepack (Højsgaard, Halekoh, and Yan 2005). In addition, the function `ComparisonStats` was developed to evaluate the statistical significance of the desired comparisons for the accuracy and time data.

```
> ComparisonStats <- function(FittedModel, Lmatrix, alpha = 0.05) {
+     Lmatrix <- matrix(Lmatrix, nrow = 1)
+     ModelBetas <- FittedModel$geese$beta
+     ModelVCov <- FittedModel$geese$vbeta
+     Estimate <- drop(Lmatrix %*% ModelBetas)
+     SdError <- sqrt(drop(Lmatrix %*% ModelVCov %*% t(Lmatrix)))
+     CBs <- Estimate + qnorm(c(alpha/2, 1 - alpha/2)) * SdError
+     pvalue <- 2 * pnorm(-abs(Estimate/SdError))
+     ans <- c(exp(c(Estimate, CBs)), round(pvalue, 4))
+     names(ans) <- c("Estimate", paste0((1 - alpha) * 100, "% LB"), paste0((1 - alpha) *
+         100, "% UB"), "p-value")
+     ans
+ }
```

Import Data

The performance data for the linear group were obtained as follows:

```
> library(readxl)
> performance_data <- read_excel("data/mainAllPerformance_GS.xlsx")
> names(performance_data) <- gsub("[.]", "_", make.names(names(performance_data), unique = TRUE))
> names(performance_data)[4] <- "Treatment"
> linear_performance_data <- performance_data[performance_data$Group == "Linear", ]
```

Accuracy Data

First, the following regression model was fitted to the accuracy data

$$\log\left[\frac{\Pr(Y_{ij}=1)}{1-\Pr(Y_{ij}=1)}\right] = \beta_0 + \beta_1 x_{ij1} + \beta_2 x_{ij2} + \beta_3 x_{ij3} + \beta_4 x_{ij4} + \beta_5 x_{ij1} x_{ij4} + \beta_6 x_{ij2} x_{ij4} + \beta_7 x_{ij3} x_{ij4}$$

where

- $\Pr(Y_{ij} = 1)$ is the probability for participant i to answer question j correctly,
- x_{ij1} is the indicator for the *NWMA* treatment,
- x_{ij2} is the indicator for the *WM* treatment,
- x_{ij3} is the indicator for the *WMA* treatment,
- x_{ij4} is the indicator for the *subset* task type,

for $i = 1, \ldots, 104$, corresponding to the individual participants and $j = 1, \ldots, 8$ corresponding to questions. This model assumes that the success probability depends on the treatment, on the task type and on their interaction.

```
> library(geepack)
> fitted_model <- geeglm(formula = Correct ~ Treatment * Question_Type, id = Study_Id,
+     data = linear_performance_data, family = binomial(link = "logit"))
```

By comparing this model to one with no interaction terms

```
> reduced_model <- geeglm(formula = Correct ~ Treatment + Question_Type, id = Study_Id,
+     data = linear_performance_data, family = binomial(link = "logit"))
> Waldts <- anova(fitted_model, reduced_model)
> Waldts
Analysis of 'Wald statistic' Table

Model 1 Correct ~ Treatment * Question_Type
Model 2 Correct ~ Treatment + Question_Type
  Df     X2 P(>|Chi|)
1  3 1.9062    0.5921
> fitted_model <- reduced_model
```

we can conclude that at least one of the interplay of *treatment* and *type* is not significant (p - value $= 0.5921$). This means that the effect of each treatment is constant across the two tasks and the effect of each task is constant across the four treatments.

The interaction terms can be removed and the regression model fitted to the accuracy data can be simplified to

$$\log\left[\frac{\Pr(Y_{ij} = 1)}{1 - \Pr(Y_{ij} = 1)}\right] = \beta_0 + \beta_1 x_{ij1} + \beta_2 x_{ij2} + \beta_3 x_{ij3} + \beta_4 x_{ij4}.$$

Comparison of treatments

Ranking: WM $>$ WMA $>$ NWM $\approx$ NWMA

```
> ## NWMA vs. NWM
> ComparisonStats(fitted_model, c(0, 1, 0, 0, 0))
 Estimate    95% LB    95% UB   p-value
0.8696363 0.6658175 1.1358477 0.3053000
> ## WM vs. NWM
> ComparisonStats(fitted_model, c(0, 0, 1, 0, 0))
Estimate   95% LB   95% UB  p-value
4.740765 2.935125 7.657205 0.000000
> ## WMA vs. NWM
> ComparisonStats(fitted_model, c(0, 0, 0, 1, 0))
Estimate   95% LB   95% UB  p-value
1.575433 1.081090 2.295823 0.018000
```

```
> ## NWMA vs. WM
> ComparisonStats(fitted_model, c(0, 1, -1, 0, 0))
 Estimate    95% LB    95% UB   p-value
0.1834380 0.1136435 0.2960970 0.0000000
> ## NWMA vs. WMA
> ComparisonStats(fitted_model, c(0, 1, 0, -1, 0))
 Estimate    95% LB    95% UB   p-value
0.5519982 0.3842643 0.7929489 0.0013000
> ## WM vs. WMA
> ComparisonStats(fitted_model, c(0, 0, 1, -1, 0))
Estimate   95% LB   95% UB  p-value
3.009181 1.763933 5.133513 0.000100
```

Comparison of task type

Ranking: subset < disjoint

```
> ## subset vs. disjoint
> ComparisonStats(fitted_model, c(0, 0, 0, 0, 1))
 Estimate    95% LB    95% UB   p-value
0.4733299 0.3330892 0.6726162 0.0000000
```

Time Data

First, the following regression model was fitted to the time data

$$\log(Z_{ij}) = \gamma_0 + \gamma_1 x_{ij1} + \gamma_2 x_{ij2} + \gamma_3 x_{ij3} + \gamma_4 x_{ij4} + \gamma_5 x_{ij1} x_{ij4} + \gamma_6 x_{ij2} x_{ij4} + \gamma_7 x_{ij3} x_{ij4}$$

where Z_{ij} is the time required for participant i to answer question j correctly for $i = 1, \ldots, 104$, corresponding to the individual participants and $j = 1, \ldots, 8$ corresponding to questions. This model assumes that the time to answer a question correctly depends on the treatment, onthe question type and on their interaction.

```
> linear_time <- linear_performance_data[linear_performance_data$Correct == 1, ]
> fitted_model <- geeglm(formula = log(Time) ~ Treatment * Question_Type, id = Study_Id,
+     data = linear_time, family = gaussian(link = "identity"))
```

Testing for the interaction term,

```
> anova(fitted_model)
Analysis of 'Wald statistic' Table
Model: gaussian, link: identity
Response: log(Time)
Terms added sequentially (first to last)

                        Df     X2 P(>|Chi|)
Treatment                3 55.205 6.208e-12 ***
Question_Type            1  4.141   0.04185 *
Treatment:Question_Type  3  8.044   0.04512 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

implies that the interplay of *treatment* and *type* is significant (p - value $< 0.001$$).

Comparison of treatments

Subset

Ranking: WM > WMA ≈ NWM ≈ NWMA

```
> ## NWMA vs. NWM
> ComparisonStats(fitted_model, c(0, 1, 0, 0, 0, 1, 0, 0))
 Estimate    95% LB    95% UB   p-value
1.1975405 0.9702002 1.4781519 0.0933000
> ## WM vs. NWM
> ComparisonStats(fitted_model, c(0, 0, 1, 0, 0, 0, 1, 0))
 Estimate    95% LB    95% UB   p-value
0.7035691 0.5857572 0.8450762 0.0002000
> ## WMA vs. NWM
> ComparisonStats(fitted_model, c(0, 0, 0, 1, 0, 0, 0, 1))
 Estimate    95% LB    95% UB   p-value
1.0025311 0.8168709 1.2303886 0.9807000
> ## NWMA vs. WM
> ComparisonStats(fitted_model, c(0, 1, -1, 0, 0, 1, -1, 0))
Estimate   95% LB   95% UB  p-value
1.702094 1.400201 2.069076 0.000000
> ## NWMA vs. WMA
> ComparisonStats(fitted_model, c(0, 1, 0, -1, 0, 1, 0, -1))
 Estimate    95% LB    95% UB   p-value
1.1945170 0.9841666 1.4498266 0.0721000
> ## WM vs. WMA
> ComparisonStats(fitted_model, c(0, 0, 1, -1, 0, 0, 1, -1))
 Estimate    95% LB    95% UB   p-value
0.7017928 0.6194295 0.7951076 0.0000000
```

Disjoint

Ranking: WM > WMA ≈ NWM ≈ NWMA

```
> ## NWMA vs. NWM
> ComparisonStats(fitted_model, c(0, 1, 0, 0, 0, 0, 0, 0))
Estimate   95% LB   95% UB  p-value
1.039395 0.887711 1.216996 0.631200
> ## WM vs. NWM
> ComparisonStats(fitted_model, c(0, 0, 1, 0, 0, 0, 0, 0))
 Estimate    95% LB    95% UB   p-value
0.7782493 0.6827298 0.8871327 0.0002000
> ## WMA vs. NWM
> ComparisonStats(fitted_model, c(0, 0, 0, 1, 0, 0, 0, 0))
 Estimate    95% LB    95% UB   p-value
0.9062678 0.7812065 1.0513497 0.1939000
> ## NWMA vs. WM
> ComparisonStats(fitted_model, c(0, 1, -1, 0, 0, 0, 0, 0))
Estimate   95% LB   95% UB  p-value
1.335555 1.142294 1.561513 0.000300
> ## NWMA vs. WMA
> ComparisonStats(fitted_model, c(0, 1, 0, -1, 0, 0, 0, 0))
```

```
 Estimate    95% LB    95% UB   p-value
1.1468957 0.9488166 1.3863267 0.1565000
> ## WM vs. WMA
> ComparisonStats(fitted_model, c(0, 0, 1, -1, 0, 0, 0, 0))
 Estimate    95% LB    95% UB   p-value
0.8587410 0.7474421 0.9866129 0.0315000
```

Comparison of question type

NWMA, WM and NWM: subset $\approx$ disjoint

WMA: subset $<$ disjoint

```
> ## (subset & NWMA) vs. (disjoint & NWMA)
> ComparisonStats(fitted_model, c(0, 0, 0, 0, 1, 1, 0, 0))
 Estimate    95% LB    95% UB   p-value
1.2311051 0.9881335 1.5338209 0.0638000
> ## (subset & WM) vs. (disjoint & WM)
> ComparisonStats(fitted_model, c(0, 0, 0, 0, 1, 0, 1, 0))
 Estimate    95% LB    95% UB   p-value
0.9659918 0.8837656 1.0558683 0.4459000
> ## (subset & WMA) vs. (disjoint & WMA)
> ComparisonStats(fitted_model, c(0, 0, 0, 0, 1, 0, 0, 1))
Estimate   95% LB   95% UB  p-value
1.182025 1.005555 1.389465 0.042700
> ## (subset & NWM) vs. (disjoint & NWM)
> ComparisonStats(fitted_model, c(0, 0, 0, 0, 1, 0, 0, 0))
 Estimate    95% LB    95% UB   p-value
1.0685267 0.8753336 1.3043591 0.5148000
```

References

Højsgaard, S., U. Halekoh, and J. Yan. 2005. "The R Package geepack for Generalized Estimating Equations." *Journal of Statistical Software* 15: 1–11.

Liang, K. Y., and S. L. Zeger. 1986. "Longitudinal Data Analysis Using Generalized Linear Models." *Biometrika* 73 (1): 13–22.

R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org/.