Reanalyze SID trial by the LWO

Yongxi Long

# Aim

This script is to demonstrate how to fit a longitudinal probabilistic index model using the lwo() function and estimate win odds longitudinally.

rm(list = ls())  
suppressPackageStartupMessages({  
 library(ggplot2)  
 library(ggsci)  
 library(splines) # for the natural cubic splines function  
 library(dplyr)  
 library(knitr)  
 library(simstudy)  
 library(plyr)  
 library(geepack)  
 library(ggpubr)  
 library(pim)  
 library(fastDummies)  
})

Warning: package 'simstudy' was built under R version 4.3.3

Warning: package 'ggpubr' was built under R version 4.3.3

Warning: package 'pim' was built under R version 4.3.3

Warning: package 'fastDummies' was built under R version 4.3.3

# get user defined functions  
source("functions.r")  
# get the lwo function  
source("lwo.r")  
colors <- c("#AED0EE",  
 "#4d79a6",  
 "#489893",  
 "#A0BF52",  
 "#f1cd63",  
 "#F3993A",  
 "#C12C1F")

# Data

## The paper used SID trial patient data

load(file = "SID\_long\_cleaned.RData")  
load(file = "SID\_wide\_cleaned.RData")  
  
dat\_wide <- SID  
dat\_long <- SID\_long  
  
# there were no category 0 (no symptom) patients and no category 6 (death) in   
# before week 8, so combine category 0 with 1 and category 6 with 5   
#dat\_long$GBS\_DS <- ifelse(dat\_long$GBS\_DS==0,1,dat\_long$GBS\_DS)  
#dat\_long$GBS\_DS <- ifelse(dat\_long$GBS\_DS==6,5,dat\_long$GBS\_DS)

## For demonstration we use a simulated data set

We use the gen\_data function to simulate ordinal repeated measurement. The data generation is designed upon te SID trial (Walgaard, C., Second intravenous immunoglobulin dose in patients with Guillain-Barré syndrome with poor prognosis (SID-GBS): a double-blind, randomised, placebo-controlled trial. The Lancet Neurology, 20(4), 275-283.)

The simulated data set (wide/long format) has the following variables:

* Patient identifier: id
* Outcome: GBS-DS, a seven scale ordinal neuro-functionality score, but combined score 0 with 1 and score 6 with 5 due to small group sizes. Repeated measurements at all visits, no missing data
* Treatment assignment: trt, 1:1 equal randomization
* Covariate: age, continuous from a normal distribution with mean 60 and sd 10
* Covariate: pre\_diarrhea: binary, yes(1)/no(0), prevalence is 40%

## set simulation parameters

# sample size (two arms)  
N <- 100  
# baseline category probabilities  
baseprobs <- c(0.02,0.06,0.11,0.12,0.47,0.21,0.01)  
# number of visits, unit is week  
visits <- c(0,1,2,4,8,12,26) # baseline is visit = 0   
  
# simulate data from a proportional odds model  
# positive time trend towards higher score (self-healing)  
time\_effect <- -0.05   
# covariate effects estimated from SID trial  
covs\_effects\_baseline <- c("trt"=0,"age"=0.005,"pre\_diarrhea"= -0.23)  
#treatment accelerates healing process  
time\_trt\_interaction <- -0.05  
  
# correlation structure, the simstady package generates correlated values from   
# the logistic distribution using a standard normal copula-like approach   
# with supplied correlation matrix  
rho <- 0.6 ; corstr <- "ar1"  
corMatrix <- generate\_corMatrix(n\_visits = length(visits),rho=rho,corstr = corstr)

## get the data

set.seed(86)  
dat <- gen\_data(N,baseprobs,covs\_effects\_baseline,  
 time\_effect,time\_trt\_interaction,  
 visits,corMatrix)  
dat\_wide <- dat$wide\_format  
dat\_long <- dat$long\_format  
  
# change variable names to matche the SID trial data set  
colnames(dat\_wide)[which(colnames(dat\_wide)=="id")] <- "patient\_ID"  
colnames(dat\_wide)[which(colnames(dat\_wide)=="trt")] <- "treat\_ITT"  
colnames(dat\_wide)[6:12] <- paste0("GBS\_DS\_week",c(0,1,2,4,8,12,26))  
# change GBS-DS range from 1-7 to 0-6 scale  
dat\_wide[,paste0("GBS\_DS\_week",c(0,1,2,4,8,12,26))] <- dat\_wide[,paste0("GBS\_DS\_week",c(0,1,2,4,8,12,26))] - 1  
  
colnames(dat\_long)[which(colnames(dat\_long)=="id")] <- "patient\_ID"  
colnames(dat\_long)[which(colnames(dat\_long)=="trt")] <- "treat\_ITT"  
colnames(dat\_long)[which(colnames(dat\_long)=="time")] <- "week"  
colnames(dat\_long)[which(colnames(dat\_long)=="y")] <- "GBS\_DS"  
dat\_long$GBS\_DS <- dat\_long$GBS\_DS - 1

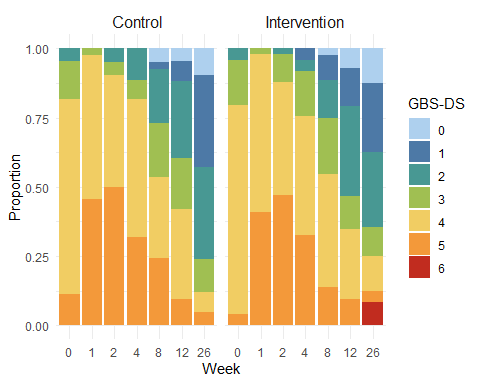
# Exploratory analysis

## Proportion of categories over time

# see the frequency distribution of ordinal categories over time  
tab <- as.data.frame.matrix(table("week"=dat\_long$week,"GBS-DS"=dat\_long$GBS\_DS))  
colnames(tab) <- paste0("GBS-DS-", sort(unique(dat\_long$GBS\_DS)))  
rownames(tab) <- paste0("week",c(0,1,2,4,8,12,26))  
kable(tab)

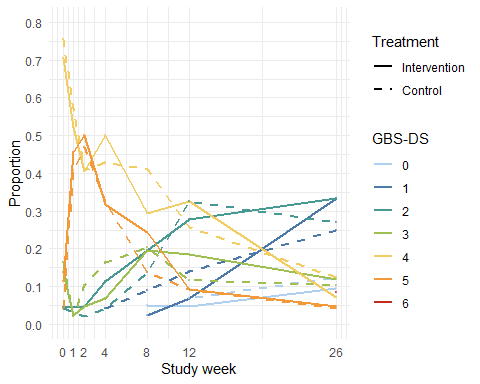
|  | GBS-DS-0 | GBS-DS-1 | GBS-DS-2 | GBS-DS-3 | GBS-DS-4 | GBS-DS-5 | GBS-DS-6 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| week0 | 0 | 0 | 4 | 14 | 68 | 7 | 0 |
| week1 | 0 | 0 | 0 | 2 | 51 | 40 | 0 |
| week2 | 0 | 0 | 3 | 7 | 37 | 44 | 0 |
| week4 | 0 | 2 | 7 | 11 | 43 | 30 | 0 |
| week8 | 3 | 5 | 14 | 17 | 30 | 16 | 0 |
| week12 | 5 | 9 | 26 | 13 | 25 | 8 | 0 |
| week26 | 10 | 26 | 27 | 10 | 9 | 4 | 4 |

# visualize observed proportions of each category over time by treatment group  
cat\_prop <- plyr::count(dat\_long,c("treat\_ITT","week","GBS\_DS"))  
cat\_prop$treat\_ITT <- factor(cat\_prop$treat\_ITT,levels = c(0,1),  
 labels = c("Control","Intervention"))  
ggplot(data = cat\_prop,aes(x=factor(week),y=freq,fill=factor(GBS\_DS)))+  
 facet\_wrap(~treat\_ITT)+  
 geom\_bar(position = "fill",stat = "identity")+  
 labs(  
 #title = "Stacked Bar Chart of Observed Proportion of Each GBS-DS Score over Time",  
 x="Week",  
 y="Proportion",  
 fill="GBS-DS")+  
 theme\_minimal()+  
 scale\_fill\_manual(values = colors)+  
 theme(  
 strip.text.x = element\_text(size=12)  
 )



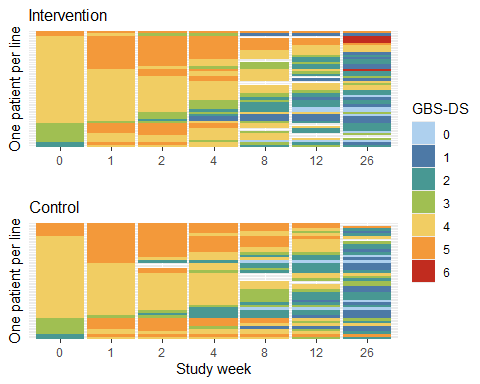
## empirical cumulative probability of each category

# get empirical CDF for each category over time  
dat\_long |>  
 mutate(GBS\_DS = as.factor(GBS\_DS),  
 treat = as.factor(treat\_ITT)) |>  
 group\_by(week,treat,GBS\_DS) |>  
 dplyr::count() |>  
 group\_by(week,treat) |>  
 dplyr::mutate(count=sum(n)) |>  
 dplyr::mutate(cumprob=n/count) |>  
 ggplot() +  
 aes(x = week, y = cumprob, color=GBS\_DS, linetype=treat) +  
 geom\_line(lwd=0.8) +  
 geom\_point(alpha=0.6, size=0.5) +  
 scale\_x\_continuous(breaks = c(0,1,2,4,8,12,26)) +  
 scale\_y\_continuous(breaks = seq(0, 1, by=0.1)) +  
 coord\_cartesian(ylim = c(0,0.8)) +  
 labs(x = "Study week",  
 # title = "Line Chart of Observed Proportion of Each GBS-DS Score over Time",  
 y = "Proportion",  
 color = "GBS-DS",  
 linetype = "Treatment") +  
 scale\_color\_manual(values = colors)+  
 theme\_minimal()+  
 scale\_linetype\_manual(values = c(1,2),  
 labels=c("Intervention","Control"))



## Individual horizontal line plot

# order patients for better illustration  
ID\_order0 <- dat\_wide |>  
 filter(treat\_ITT==0) |>  
 # arrange(visit1,visit2,visit3,visit4,visit5,visit6,visit7) |>  
 arrange(GBS\_DS\_week0,GBS\_DS\_week1,GBS\_DS\_week2,GBS\_DS\_week4,  
 GBS\_DS\_week8,GBS\_DS\_week12,GBS\_DS\_week26) |>  
 select(patient\_ID) |>  
 unlist() |>  
 as.vector()  
ID\_order1 <- dat\_wide |>  
 filter(treat\_ITT==1) |>  
 # arrange(visit1,visit2,visit3,visit4,visit5,visit6,visit7) |>  
 arrange(GBS\_DS\_week0,GBS\_DS\_week1,GBS\_DS\_week2,GBS\_DS\_week4,  
 GBS\_DS\_week8,GBS\_DS\_week12,GBS\_DS\_week26) |>  
 select(patient\_ID) |>  
 unlist() |>  
 as.vector()  
dat\_long\_ordered <- bind\_rows(sapply(c(ID\_order0,ID\_order1), function(id) {  
 df <- dat\_long[which(dat\_long$patient\_ID==id),]  
 if(id %in% ID\_order0)  
 {  
 df$order <- rep(which(ID\_order0==id),nrow(df))  
 } else  
 {  
 df$order <- rep(which(ID\_order1==id),nrow(df))  
 }  
 return(df)  
},simplify = FALSE))  
  
dat\_long\_ordered <- dat\_long\_ordered |>  
 mutate(id = as.factor(order),treat\_ITT=factor(treat\_ITT,  
 levels=c(0,1),  
 labels=c("Control","Intervention")))   
   
  
p0 <- ggplot(subset(dat\_long\_ordered,treat\_ITT=="Control")) +  
 #facet\_grid(rows = vars(treat\_ITT),switch = "y")+  
 aes(y=id, x=factor(week)) +  
 geom\_tile(mapping = aes(fill = as.factor(GBS\_DS)),  
 width=0.95, height=1) +  
 scale\_fill\_manual(values = colors) +  
 #scale\_x\_discrete(limits=as.character(c(0,1,2,4,8,12,26))) +  
 expand\_limits(fill = c(as.character(0:6))) +  
 labs(x = "Study week",  
 y = "One patient per line",  
 fill = "GBS-DS",  
 title = "Control")+  
 theme(  
 axis.text.y = element\_blank(),  
 axis.ticks.y = element\_blank(),  
 plot.title = element\_text(size=12)  
 )  
p1 <- ggplot(subset(dat\_long\_ordered,treat\_ITT=="Intervention")) +  
 #facet\_grid(rows = vars(treat\_ITT),switch = "y")+  
 aes(y=id, x=factor(week)) +  
 geom\_tile(mapping = aes(fill = as.factor(GBS\_DS)),  
 width=0.95, height=1) +  
 scale\_fill\_manual(values = colors) +  
 #scale\_x\_discrete(limits=as.character(c(0,1,2,4,8,12,26))) +  
 expand\_limits(fill = c(as.character(0:6))) +  
 labs( x="",  
 y = "One patient per line",  
 fill = "GBS-DS",  
 title = "Intervention")+  
 theme(  
 axis.text.y = element\_blank(),  
 axis.ticks.y = element\_blank(),  
 plot.title = element\_text(size=12)  
 )  
  
ggarrange(p1,p0,ncol=1,nrow=2,common.legend = TRUE,  
 legend = "right")



# Longitudinal win odds analysis

* Assume successful randomization, we don’t model treatment effect at baseline. We adjust for baseline GBS-DS
* The function internally convert individual-level data to pair-level data

## Analysis 1: model treatment effect change through spline

* From exploratory data analysis, we see a curved shape of individual patient trajectories over time, and also we see the distribution of the GBS-DS kind of shifted up first (week 0 to week 2) and then went down. So we model the treatment effect trajectory in the natural cubic spline form with one knot at week 2 (degree of freedom = 2, allows the curve to “bend” once).

dat\_long$GBS\_DS\_baseline <- rep(dat\_wide$GBS\_DS\_week0,as.vector(table(dat\_long$patient\_ID)))  
# delete baseline as an outcome  
dat\_long <- dat\_long[dat\_long$week>0,]  
# model time as a natural cubic spline with 3 knots  
# use auto regressive of order 1 as the working correlation structure  
# for the temporal correlation  
mod.lwo <- lwo(  
 GBS\_DS ~ treat\_ITT + age + pre\_diarrhea + GBS\_DS\_baseline+  
 treat\_ITT:ns(week,knots = 2),  
 data = dat\_long,  
 id = "patient\_ID",  
 visit = "time",  
 time.varname = "week",  
 corstr = "ar1",  
 larger = FALSE  
)

Warning in eval(family$initialize): non-integer #successes in a binomial glm!  
Warning in eval(family$initialize): non-integer #successes in a binomial glm!

summary(mod.lwo)

Call:  
lwo(formula = GBS\_DS ~ treat\_ITT + age + pre\_diarrhea + GBS\_DS\_baseline +   
 treat\_ITT:ns(week, knots = 2), data = dat\_long, id = "patient\_ID",   
 visit = "time", time.varname = "week", larger = FALSE, corstr = "ar1")  
  
 Coefficients:  
 Estimate Std.err Wald Pr(>|W|)  
treat\_ITT 0.091425 0.215299 0.180 0.671  
age -0.002913 0.004900 0.353 0.552  
pre\_diarrhea -0.037957 0.189349 0.040 0.841  
GBS\_DS\_baseline -0.003689 0.148385 0.001 0.980  
treat\_ITT:ns(week, knots = 2)1 0.209110 0.460510 0.206 0.650  
treat\_ITT:ns(week, knots = 2)2 -0.327576 0.204576 2.564 0.109  
  
Temporal Correlation Structure = ar1   
  
Estimated Correlation Parameters:  
 Estimate Std.err  
alpha 0.7522 0.006505  
Median Number of Visits Per Pseudo-pair 6

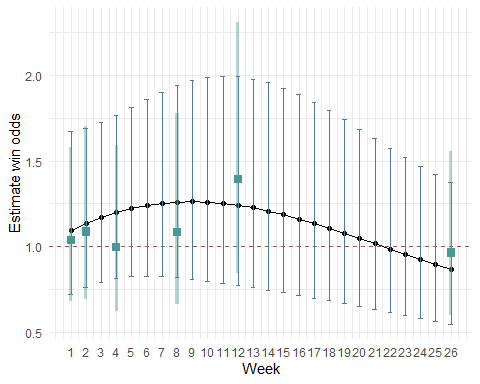
* To get win odds for a sequence of time points (week 1 till week 26), construct a new dataframe and use the predict() function.

# we want to get win odds from week 1 till week 26, conditional on that  
# the pairs have the same age and same preceding diarrhea status and same baseline GBS-DS  
# so set age and pre\_diarrhea and GBS\_DS\_baseline to 0  
nvisits <- 26  
newdata <- data.frame(  
 id = rep(c(1,2),each=nvisits),  
 week = rep(1:nvisits,2),  
 treat\_ITT = rep(c(0,1),each=nvisits),  
 age = rep(60,nvisits\*2),  
 pre\_diarrhea = rep(0,2\*nvisits),  
 GBS\_DS\_baseline = rep(4,2\*nvisits)  
)  
pred.lp <- predict(mod.lwo,newdata,  
 id="id",  
 visit="week",  
 time.varname = "week",  
 type="link",conf.int = TRUE)  
pred.wo <- exp(pred.lp)

* For comparison, we calculate the win odds at all visits of GBS-DS (primary endpoint) from single-time probabilistic index model

# adjusted win odds for week 1,2,4,8,12,26  
covs <- paste0("GBS\_DS\_week",c(1,2,4,8,12,26))  
wo.adj <- sapply(covs, function(cov)  
 {  
 data <- dat\_wide[!is.na(dat\_wide[,cov]),]  
 # by default PIM thinks larger outcome is a win,  
 # to accomodate that smaller GBS-DS is better  
 # we reverse the outcome  
 data[,cov] <- 6 - data[,cov]  
 formula <- as.formula(paste0(cov,"~treat\_ITT+age+pre\_diarrhea+GBS\_DS\_week0"))  
 mod.adj <- pim(formula = formula,data = data)  
 wo.adj <- c(exp(mod.adj@coef["treat\_ITT"]),exp(confint(mod.adj)["treat\_ITT",]))  
 names(wo.adj) <- c("estimate","lower.CI","upper.CI")  
 return(wo.adj)  
}) |>  
 t() |>  
 as.data.frame()

* Visualize estimated win odds for each week, add single-time point win odds analysis for comparison.



## Analysis 2: Response profile modelling (time as a categorical covariate)

* In this analysis, we use an unstructured model for the probabilistic index. Namely, we don’t assume a parametric form for the treatment effect trajectory, instead, we model time categorically.

# model time categorically  
dat\_long2 <- dummy\_cols(dat\_long,  
 select\_columns = "week",  
 remove\_first\_dummy = TRUE)  
mod.lwo <- lwo(  
 GBS\_DS ~ treat\_ITT + age + pre\_diarrhea + GBS\_DS\_baseline+  
 treat\_ITT:week\_2+  
 treat\_ITT:week\_4+  
 treat\_ITT:week\_8+  
 treat\_ITT:week\_12+  
 treat\_ITT:week\_26,  
 data = dat\_long2,  
 id = "patient\_ID",  
 visit = "time",  
 time.varname = paste0("week\_",c(2,4,8,12,26)),  
 corstr = "ar1",  
 larger = FALSE  
)

Warning in eval(family$initialize): non-integer #successes in a binomial glm!  
Warning in eval(family$initialize): non-integer #successes in a binomial glm!

newdata <- dat\_long2[dat\_long2$patient\_ID==1,]  
nvisits <- nrow(newdata)  
newdata <- rbind(newdata,newdata)  
newdata$patient\_ID[(nvisits+1):(2\*nvisits)] <- 2  
newdata$treat\_ITT <- rep(c(0,1),each=nvisits)  
pred.lp <- predict(mod.lwo,newdata,  
 id="patient\_ID",  
 visit="time",  
 time.varname = paste0("week\_",c(2,4,8,12,26)),  
 type="link",conf.int = TRUE)  
pred.wo <- exp(pred.lp)

