A. Supplement

Variable	Symbol	Values	Improvement
Clinical global impression of severity	CGIS	1-7	Lower
Leeds sleep evaluation-awakening from sleep	LSEAFS	0-100	Higher
Leeds sleep evaluation-quality of sleep	LSEQOS	0-100	Higher
LSE-Behaviour following wakefulness	LSEBFW	0-100	Higher
Multidimensional assessment of fatigue index	MAFGFI	1-50	Lower
Patient global impression of change	PGIC	1-7	Lower
Restorative sleep questionnaire weekly	RSQW	0-100	Higher
Sheehan disability scale	SDS	0-30	Lower
SF36-Vitality	SF36V	0-100	Higher
Daytime consequences of sleep questionnaire	DCSQTOT	0-100	Lower
Benzodiazepine withdrawal symptom questionnaire	BWSQTOT	0-22	Lower

Table A.1. Efficacy variables collected at weekly clinic visits.

	Dose				
	0	15	30	45	60
RANDOMIZED	143	134	134	124	137
COMPLETED STUDY	124	110	115	104	109
ADVERSE EVENT	3	4	1	5	13
SUBJECT DIED	0	0	0	0	1
PROTOCOL VIOLATION	4	5	2	3	3
LOST TO FOLLOW UP	1	1	3	5	4
OTHER	2	0	2	0	1
FAILED ENTRANCE CRITERIA	1	0	1	0	0
SUBJECT WITHDREW CONSENT	8	13	10	7	6
PREGNANCY	0	1	0	0	0

Table A.2. Reason for end of dosing as recorded on the case report form at the final visit. The number randomized is included in first row.

	15mg vs PBO			60mg vs PBO		
Endpoint	MLLM	LOCF	AC	MLLM	LOCF	AC
swaso	-2.93	-2.7	-1.59	-26.61	-24.42	-23.22
	(5.97)	(5.84)	(6.46)	(6)	(5.92)	(6.46)
slso	-5.79	-5.08	-5.24	-6.1	-5.44	-5.46
	(4.23)	(4.16)	(4.49)	(4.25)	(4.22)	(4.49)
tst	4.51	4.63	1.93	23.34	22.85	23.18
	(7.83)	(7.68)	(8.29)	(7.87)	(7.77)	(8.28)
squality	1.01	0.97	1.43	8.04	7.46	7.88
	(2.49)	(2.43)	(2.66)	(2.5)	(2.45)	(2.66)

Table A.3. Estimates (standard errors) for the Week 4 endpoints formed from weekly averages of the daily diaries. Results are presented for the low and high doses versus placebo. Estimation based on the saturated longitudinal model (MLLM), last observation carried forward (LOCF), and available cases at Week 4 (AC).

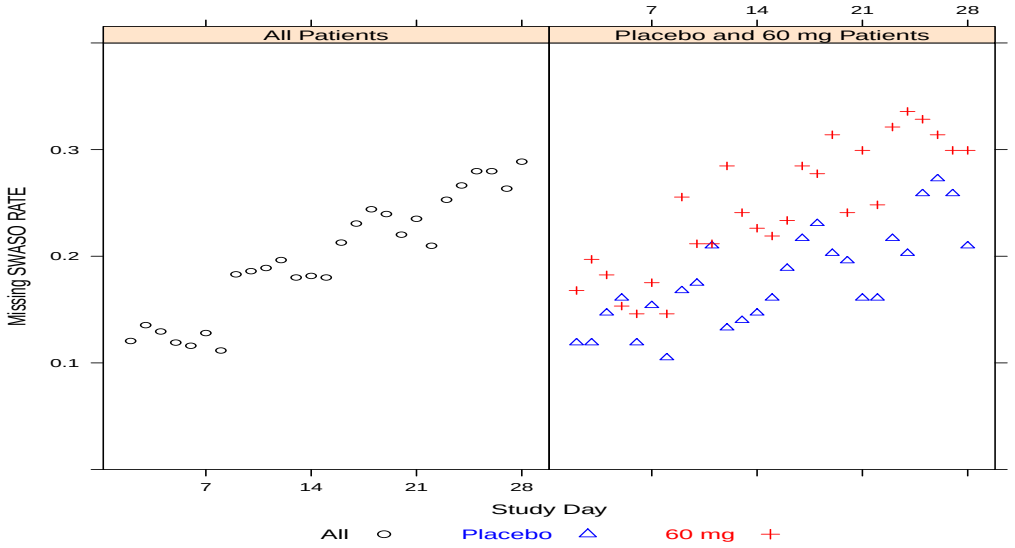

Figure A.1. The proportion of missing SWASO values for study days 1-28. The left panel displays the daily rates for all patients, and the right panel for placebo and 60 mg patients.

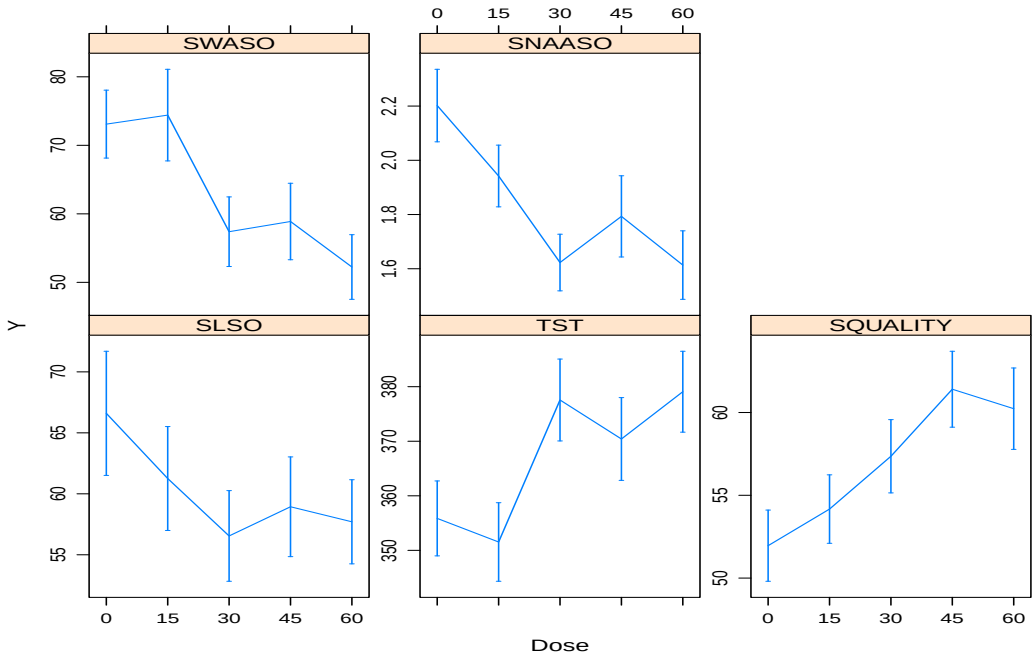

Figure A.2. Mean dose response at Week 4 for weekly endpoints computed from daily diary data using the pre-specified method for computing weekly averages. Dose group means are computed from available cases. The error bars are $\pm SE$. The variables are defined in Section 2.2.

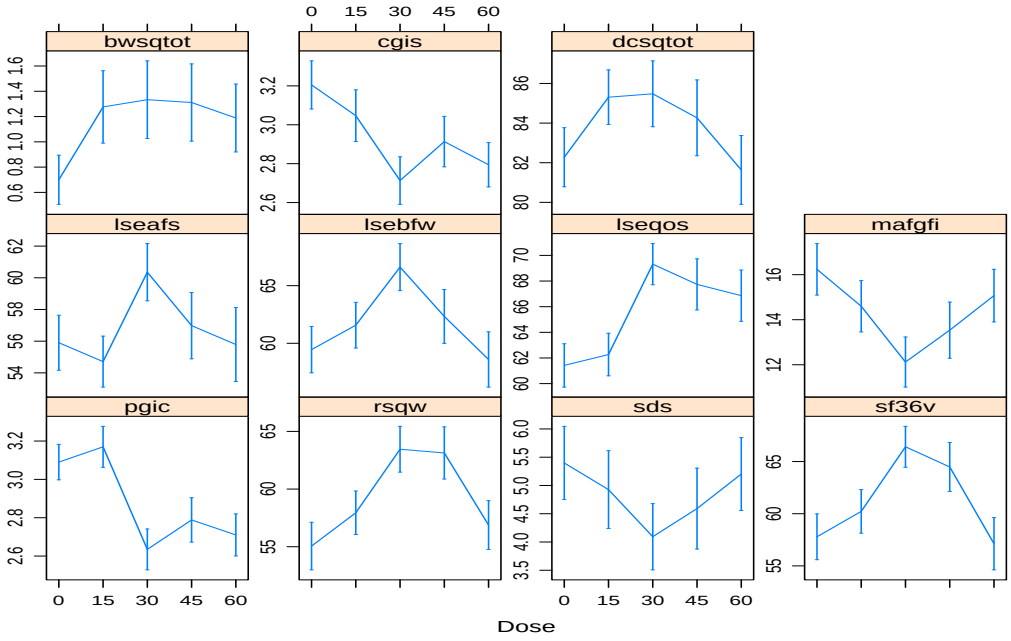

Figure A.3. Mean dose response at Week 4 for the secondary endpoints collected at the weekly visits. Dose group means are computed from available cases. The error bars are $\pm SE$. The variables are defined in Table A.1.

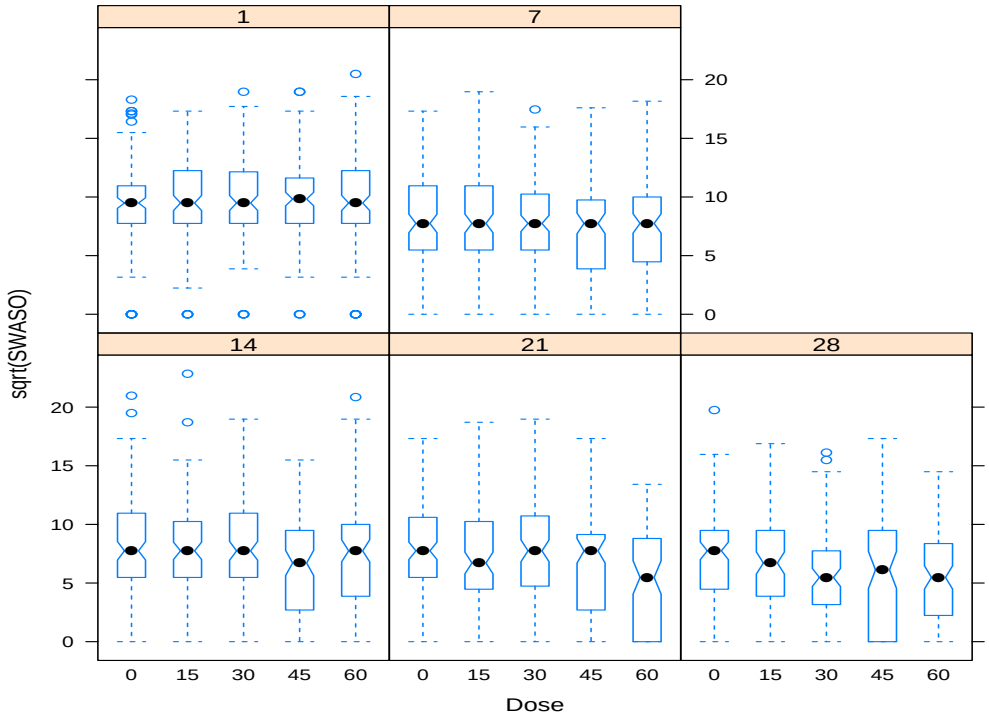

Figure A.4. Box plots of Square root transformed daily SWASO values at selected study days $(1, 7, 14, 21, 28)$.

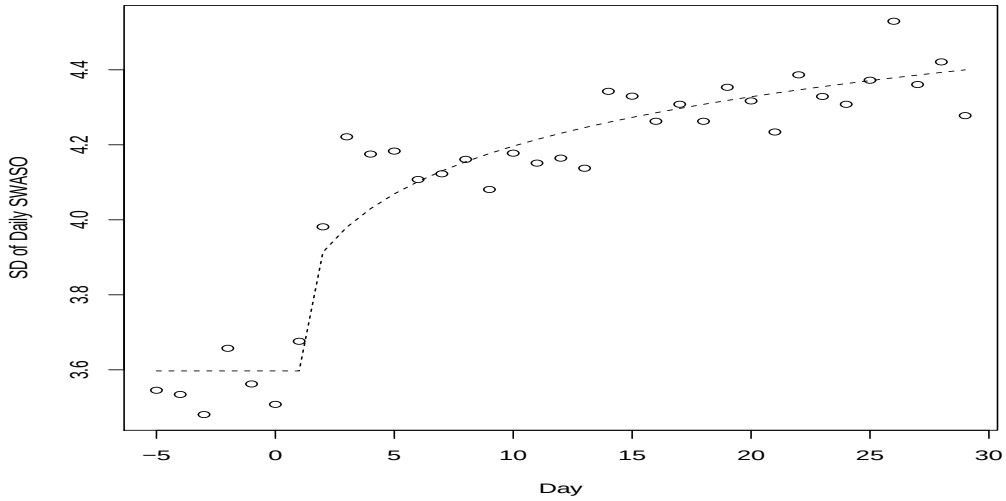

Figure A.5. Trend in SDs of daily $\sqrt{\text{SWASO}}$ after removing dose group mean differences. The dashed curve is $f(day) = 3.72 + 0.20 \log\{0.5 + (day)I(day > 1)\}$.

	15mg vs PBO			60mg vs PBO		
Endpoint	MLLM	LOCF	AC	MLLM	LOCF	AC
bwsqtot	0.6	0.64	0.66	0.4	0.4	0.39
	(0.33)	(0.3)	(0.34)	(0.33)	(0.3)	(0.35)
cgis	-0.06	-0.02	-0.15	-0.37	-0.29	-0.42
	(0.15)	(0.14)	(0.15)	(0.15)	(0.14)	(0.15)
dcsqtot	-0.98	-1.6	-0.26	-2.58	-2.66	-1.68
	(1.78)	(1.74)	(1.89)	(1.78)	(1.73)	(1.89)
lseafs	-0.39	-0.11	0.21	-1.46	-2.57	0.04
	(2.53)	(2.31)	(2.68)	(2.53)	(2.29)	(2.68)
lsebfw	1.22	1.12	2.05	-0.95	-1.35	-1.11
	(2.57)	(2.43)	(2.72)	(2.58)	(2.41)	(2.72)
lseqos	1.4	0.99	2.49	6.07	4.69	6.59
	(2.28)	(2.16)	(2.42)	(2.29)	(2.15)	(2.42)
mafgfi	0.12	0.19	-0.32	-0.31	-0.1	-0.66
	(1.26)	(1.22)	(1.36)	(1.27)	(1.22)	(1.36)
pgic	0.1	0.12	0.1	-0.34	-0.31	-0.35
	(0.14)	(0.14)	(0.15)	(0.14)	(0.14)	(0.15)
rsqw	1.09	0.99	1.49	2.76	2.14	2.97
	(2.33)	(2.22)	(2.46)	(2.34)	(2.21)	(2.47)
sds	0.16	0.36	-0.14	-0.07	0.11	-0.7
	(0.76)	(0.75)	(0.82)	(0.77)	(0.74)	(0.84)
sf36v	0.27	-0.17	0.76	1.34	1.26	1.92
	(2.41)	(2.27)	(2.54)	(2.44)	(2.28)	(2.56)

Table A.4. Estimates (standard errors) for the Week 4 secondary endpoints. Results are presented for the low and high doses versus placebo. Estimation based on the saturated longitudinal model (MLLM), last observation carried forward (LOCF), and available cases at Week 4 (AC).

R Graphics Functions

```
library(ggplot2)

###  Fig 4
impPlot<-function(yin,id,impin,visits,impcol=1:ncol(impin),ylab='Y',xlab='Time',
                                    cbbPalette=c( "#E69F00", "#56B4E9", "#009E73", "#D55E00", "#0072B2"),
                                    obsColor="black",legend=TRUE)
        {
        ###     yin outcome data from first to last visit.  any missing data must
        ###             be NA so there exactly nvis values per patient.  outcomes from multiple
        ###             multiple patients are stacked
        ###
        ###     id  patient identifiers corresponding to yin
        ###
        ###     impin   matrix of multiply imputed values.  rows correspond to yin
        ###                     columns are imputations.  if a value in yin is observed, imputations
        ###                     are repeated observed values
        ###     visits  Increasing visit numbers corresponding to outcomes in yin
        ###
        ###     impcol   columns of impin to use. typically 1:(number imputations in impin)
        ###
        ###     cbbPalette      R cookbook color scheme used for up to 5 imputed values
        ###                             It will need to be redefined if more than 5 imputations plotted
        ###                             otherwise the colors will be repeated
        ###
        ###     obsColor        Color used to represent observed values
        ###
        ###     legend      FALSE to suppress print of legend
        ###
        ###                                             output
        ###     scatter plot of outcome vs time.  observed data plotted with larger black lines and dots.
        ###     the imputed values are assigned different colors with thin lines connecting them
        ###


        uid<-sort(unique(id))
        nimp<-length(impcol)
        nvis<-length(visits)

        if(length(yin)!=nvis*length(uid))stop('yin,␣id,␣and␣visits␣have␣inconsistent␣lengths')

        if(max(impcol)>ncol(impin))stop('impcol␣must␣be␣a␣subset␣of␣the␣columns␣of␣impin')

        rcol<-ceiling(nimp/length(cbbPalette))
        cbbPalette<-rep(cbbPalette,rcol)  ## repeat colors if needed
        cbbPalette<-c(cbbPalette[1:nimp],obsColor)

        obsdat<-NULL
        segdat<-NULL
        visdat<-NULL
        for(curid in uid){
                ### set up observed values for printing
                y<-yin[id==curid]
                misind<-is.na(y)
                nobs<-sum(!misind)
                nmis<-sum(misind)
                yobs<-y[!misind]
                visobs<-visits[!misind]
                idobs<-rep(curid,sum(!misind))
                impid<-rep('OBS',sum(!misind))
                obsdat<-rbind(obsdat,data.frame(id=idobs,impid,visobs,yobs))

                ### set up line segments between observed values
                xstart<-NULL
                xend<-NULL
                ystart<-NULL
                yend<-NULL
                for(i in 1:(nvis-1)){
                        if(!misind[i] & !misind[i+1]){
                                xstart<-c(xstart,visits[i])
                                xend<-c(xend,visits[i+1])
                                ystart<-c(ystart,y[i])
                                yend<-c(yend,y[i+1])
                        }
                }
                idseg<-rep(curid,length(xstart))
                impid<-rep('OBS',length(xstart))
```



```
                segdat<-rbind(segdat,data.frame(id=idseg,impid,xstart,xend,ystart,yend))

                ### create stacked imputed values vectors
                yimp<-impin[id==curid,impcol]
                yimp<-as.vector(yimp[,])
                idmis<-rep(curid,nvis*nimp)
                vismis<-rep(visits,nimp)
                impid<-rep(impcol,rep(nvis,nimp))
                visdat<-rbind(visdat,data.frame(id=idmis,vismis,impid,yimp))
        }

        ### create factor variables
        obsdat[,'id']<-factor(obsdat[,'id'])
        segdat[,'id']<-factor(segdat[,'id'])
        visdat[,'id']<-factor(visdat[,'id'])
        obsdat[,'impid']<-factor(obsdat[,'impid'],levels=c('OBS',impcol))
        segdat[,'impid']<-factor(segdat[,'impid'],levels=c('OBS',impcol))
        visdat[,'impid']<-factor(visdat[,'impid'],levels=c('OBS',impcol))

        plotimp<-ggplot(data=visdat,aes(x=vismis,y=yimp,color=impid))+geom_line()+geom_point()
        plotimp<-plotimp+facet_grid(id~.)
        plotimp<-plotimp+geom_point(data=obsdat,aes(x=visobs,y=yobs),size=3)
        plotimp<-plotimp+geom_segment(data=segdat,aes(x=xstart,xend=xend,y=ystart,yend=yend),size=1)
        plotimp<-plotimp+xlab(xlab)+ylab(ylab)
        plotimp<-plotimp+scale_colour_manual(values=cbbPalette,name="Imputation:")
        plotimp<-plotimp+theme_bw()
        if(legend){ plotimp<-plotimp+theme(legend.position='top')
        }else plotimp<-plotimp+theme(legend.position='none')
        print(plotimp)
        return(invisible(plotimp))
}

##############################################################################################
#### fig 5

impBox<-function(yin,id,impin,visits,bwidth=1,ylab='Y',xlab='Time')
        {
        ###     yin outcome data from first to last visit.  any missing data must
        ###             be NA so there exactly nvis values per patient.  outcomes from multiple
        ###             multiple patients are stacked
        ###
        ###     id  patient identifiers corresponding to yin
        ###
        ###     impin   matrix of multiply imputed values.  rows correspond to yin
        ###                     columns are imputations.  if a value in yin is observed, imputations
        ###                     are repeated observed values
        ###     visits  Increasing visit numbers corresponding to outcomes in yin
        ###
        ###     bwidth  The user can specify the (uniform) widths of the boxplots summarizing
        ###                     the imputed values
        ###
        ###                                                     output
        ###     Observed outcomes are represented as a scatter plot vs time.  Imputations at times with
        ###     missing outcomes are represented by boxplots
        ###

        uid<-sort(unique(id))
        nimp<-ncol(impin)
        nvis<-length(visits)

        if(length(yin)!=nvis*length(uid))stop('yin, id, and visits have inconsistent lengths')


        obsdat<-NULL
        visdat<-NULL
        for(curid in uid){
                ### set up observed values for printing
                y<-yin[id==curid]
                misind<-is.na(y)
                nobs<-sum(!misind)
                nmis<-sum(misind)
                yobs<-y[!misind]
                visobs<-visits[!misind]
                idobs<-rep(curid,sum(!misind))
                obsdat<-rbind(obsdat,data.frame(id=idobs,visobs,yobs))

                ### create stacked imputed values vectors
```

```
                yimp<-impin[id==curid,]
                yimp<-as.vector(yimp[misind,])
                idmis<-rep(curid,nmis*nimp)
                vismis<-rep(visits[misind],nimp)
                visdat<-rbind(visdat,data.frame(id=idmis,vismis,yimp))
        }

        ### create factor variables
        obsdat[,'id']<-factor(obsdat[,'id'])
        visdat[,'id']<-factor(visdat[,'id'])

        boxp<-ggplot(data=visdat)+geom_boxplot(aes(x=vismis,y=yimp,group=vismis),outlier.colour='grey',
                                                                                width=bwidth,outlier.shape=8)
        boxp<-boxp+geom_point(data=obsdat,aes(x=visobs,y=yobs),size=3,col='black')
        boxp<-boxp+facet_grid(id~.)
        boxp<-boxp+xlab(xlab)+ylab(ylab)
        boxp<-boxp+theme_bw()
        print(boxp)
        return(invisible(boxp))
}

#############################################################################################
#### fig 6

patComp<-function(gmat,visits=0:(ncol(gmat)-1),vstart=2,bwidth=1,xlab='Time',ylab='Y'){
        ###     compare completer response to responses
        ###             for patient who dropout for each observed
        ###     dropout pattern using side-by-side boxplots
        ###     gmat rectangular data matrix with one row per
        ###             patient with visits sequentially ordered
        ###             and missing values assigned NA
        ###
        ###     visits  visit numbers corresponding to columns in gmat
        ###
        ###     vstart  Starting column for dropout patterns, i.e., patients
        ###                     are present at vstart and dropout
        ###                     at (vstart+1)
        ###
        ###     bwidth  The user can specify the (uniform) widths of the boxplots
        ###
        ###                                     output
        ###
        ###     side-by-side boxplots.  for visit i, find all patients who are present
        ###             at visit i, but dropout at visit i+1. form boxplots at each visit up
        ###             to visit i for these patients.  also form boxplots for these visits
        ###             based on the patients who complete the study. the dropout/completer
        ###             boxplots are side by side.  one panel for each dropout time.
        ###             The number of completers and dropouts contributing to each panel
        ###             are also output. patients with intermitent dropouts are excluded
        ###             from all patterns

        nvis<-length(visits)

        maxw<-ncol(gmat)
        gmis<-is.na(gmat)

        comp<-apply(!gmis,1,prod)
        compvec<-as.vector(gmat[comp==1,])
        compvis<-rep(1:maxw,rep(sum(comp),maxw))

        yvec<-NULL
        vvec<-NULL
        tvec<-NULL
        dvec<-NULL
        for(i in vstart:(maxw-1)){
                ycomp<-compvec[compvis<=i]
                leny<-length(ycomp)
                yvec<-c(yvec,ycomp)
                vvec<-c(vvec,visits[compvis[compvis<=i]])
                tvec<-c(tvec,rep('Completers',leny))
                dvec<-c(dvec,rep(visits[i],leny))

                monpatL<-apply(!gmis[,1:i,drop=F],1,prod)
                monpatH<-apply(gmis[,(i+1):maxw,drop=F],1,prod)
                monpat<-monpatL*monpatH
                ydrop<-as.vector(gmat[monpat==1,1:i,drop=F])
                lenyd<-length(ydrop)
```



```
                yvec<-c(yvec,ydrop)
                vvec<-c(vvec,rep(visits[1:i],rep(sum(monpat),i)))
                tvec<-c(tvec,rep('Dropouts',lenyd))
                dvec<-c(dvec,rep(visits[i],lenyd))
        }

        dveclev<-sort(unique(dvec))
        dveclev<-paste('Dropout-',dveclev,sep='')

        dvec<-paste('Dropout-',dvec,sep='')
        dvec<-factor(dvec,levels=dveclev)
        tvec<-factor(tvec)
        vvec<-factor(vvec)

        cbbPalette<-c( "#E69F00", "#56B4E9")   ### fill colors
        bplot<-ggplot(data=data.frame(yvec,dvec,tvec,vvec))+
                        geom_boxplot(aes(y=yvec,x=vvec,fill=tvec),width=bwidth,notch=TRUE)
        bplot<-bplot+facet_grid(dvec~.)
        bplot<-bplot+xlab(xlab)+ylab(ylab)
        bplot<-bplot+scale_fill_manual(values=cbbPalette,name="Status")
        bplot<-bplot+theme_bw()
        print(bplot)

    return(invisible(list(counts=table(dvec[vvec==visits[1]],tvec[vvec==visits[1]]),bplot=bplot)))
}
```