Data Management and Statistical Analysis Supplement to "The Identification of Pretreatment Trajectories of Alcohol Use and Their Relationship to Treatment Outcome in Men and Women with Alcohol Use Disorder" by Stasiewicz, Bradizza, et al.

Joseph F Lucke*
State University of New York at Buffalo

September 3, 2019

Abstract

This supplement documents the data management and statistical analyses of "The identification of pretreatment trajectories of alcohol use ...", by Stasiewicz, Bradizza, et al. I begin with a very brief introduction to reproducible research. I then follow with data acquisition, checking, summaries, and restructuring. An overview of the statistical models is presented, followed by the statistical procedures for estimating model parameters, and model checking. Finally, I present the statistical analyses, results, and graphs.

Contents

*I thank Melanie Ruszczyk for providing the data. I also thank Braden Linn and Junru Zhao for reviewing this supplement.

List of Tables

List of Figures

1 Approach

The statistical analysis of these data is complicated. Here I present the analysis is considerable detail so that the interested reader can follow, critique and indeed improve the analysis. The overall approach follows the recommended guidelines for reproducible research, (Barba, 2018; Gandrud, 2016; Gentleman & Lang, 2007; Peng, 2009). Donoho (2010) provides the motivation underlying this approach.

> [E]rror is ubiquitous in scientific computing [and statistical analysis], and one needs to work very diligently and energetically to eliminate it. One needs a very clear idea of what has been done in order to know where to look for likely sources of error. I often cannot really be sure what a [researcher] has done from his/her own presentation, and in fact often his/her description does not agree with my own understanding of what has been done, once I look carefully at the scripts. Actually, I find that researchers quite generally forget what they have done and misrepresent their computations (p. 385).

Recent controversies (Baggerly & Coombes, 2009; Herndon, Ash & Pollin, 2013) in which research results could not be reproduced because of data mismanagement and statistical errors starkly highlight the issues involved.

This document comprises both LaTeX text (Mittelbach & Goossens, 2004) and R code (R Core Team, 2019) woven into an Rweave (.rnw) file. The .rnw file was then executed by *Sweave* (Leisch, 2002; Leisch & R-Core, 2015), creating a pure LaTeX (.tex) file. The LaTeX file was then converted to a .pdf file by the program MiKTeX (Schenk, 2019). The .rnw and corresponding .R files are available from this author.

The presentation reflects changes in names, labels, etc. that have been made over the course of the study. I have tried to update the original names, labels, etc. to the revised names, etc., but I may have missed a few changes.

Finally, I make a distinction in usage between "we" and "I". All references to "we" mean that the enclosing statement refers to a mutual decision by all the authors. References to "I" mean that this author alone made the relevant decision or action.

2 Setup

All data management and analyses were conducted in R (Ihaka, 2010; Ihaka & Gentleman, 1996; R Core Team, 2019; Thieme, 2018; Venables, Smith & The R Development Core Team, 2019). The project was conducted within RStudio integrated development environment (IDE) (RStudio Team,, 2015). The current R session is given by Table 1 on the following page.

The required packages for this analysis are loaded here.

```
> library(flexmix)
> library(geepack)
> library(Hmisc)
> library(MASS)
> library(readxl)
> library(splines)
> library(tidyverse)
> library(mice)
```

Table 1: R Session Information

- R version 3.6.1 (2019-07-05), `x86_64-w64-mingw32`
- Locale: `LC_COLLATE=English_United States.1252`, `LC_CTYPE=English_United States.1252`, `LC_MONETARY=English_United States.1252`, `LC_NUMERIC=C`, `LC_TIME=English_United States.1252`
- Running under: `Windows 10 x64 (build 17763)`
- Matrix products: default
- Base packages: base, datasets, graphics, grDevices, methods, splines, stats, utils
- Other packages: dplyr 0.8.3, flexmix 2.3-15, forcats 0.4.0, Formula 1.2-3, geepack 1.2-1, ggplot2 3.2.0, Hmisc 4.2-0, lattice 0.20-38, MASS 7.3-51.4, mice 3.6.0, purrr 0.3.2, readr 1.3.1, readxl 1.3.1, stringr 1.4.0, survival 2.44-1.1, tibble 2.1.3, tidyr 0.8.3, tidyverse 1.2.1
- Loaded via a namespace (and not attached): acepack 1.4.1, assertthat 0.2.1, backports 1.1.4, base64enc 0.1-3, boot 1.3-22, broom 0.5.2, cellranger 1.1.0, checkmate 1.9.4, cli 1.1.0, cluster 2.1.0, colorspace 1.4-1, compiler 3.6.1, crayon 1.3.4, data.table 1.12.2, digest 0.6.20, fansi 0.4.0, foreign 0.8-71, generics 0.0.2, glue 1.3.1, grid 3.6.1, gridExtra 2.3, gtable 0.3.0, haven 2.1.1, hms 0.5.0, htmlTable 1.13.1, htmltools 0.3.6, htmlwidgets 1.3, httr 1.4.0, jomo 2.6-9, jsonlite 1.6, knitr 1.23, labeling 0.3, latticeExtra 0.6-28, lazyeval 0.2.2, lme4 1.1-21, lubridate 1.7.4, magrittr 1.5, Matrix 1.2-17, minqa 1.2.4, mitml 0.3-7, modelr 0.1.4, modeltools 0.2-22, munsell 0.5.0, nlme 3.1-140, nloptr 1.2.1, nnet 7.3-12, pan 1.6, parallel 3.6.1, pillar 1.4.2, pkgconfig 2.0.2, R6 2.4.0, RColorBrewer 1.1-2, Rcpp 1.0.1, rlang 0.4.0, rpart 4.1-15, rstudioapi 0.10, rvest 0.3.4, scales 1.0.0, stats4 3.6.1, stringi 1.4.3, tidyselect 0.2.5, tools 3.6.1, utf8 1.1.4, vctrs 0.2.0, withr 2.1.2, xfun 0.8, xml2 1.2.0, zeallot 0.1.0

```
> #library(xtable)
> #library(ggpubr)
```

3 The Data

Study participants were 205 men and women between the ages of 18 to 65 years who (1) met DSM-5 criteria for an alcohol use disorder, (2) lived within commuting distance of the treatment site, and (3) provided written informed consent.

The data for the number of days abstinent (NDA) is a time series of 20 weekly intervals in which each interval, except the first, consists of the NDA for a participant for that week. The first interval is the mean of a participant's NDA's for the previous 19 weeks, i.e., NDA history. The response variable NDA is an integer value ranging from 0 to 7.

The 20-week time series was segmented into three phases anchored at Week 0, the start of treatment. The first phase was the *Distal Pretreatment* phase from Weeks −8 to −4, also denoted as $\boldsymbol{H}0$ to $\boldsymbol{H}4$. The second was the *Proximal Pretreatment* phase from Weeks −4 to 0, also denoted as $\boldsymbol{P}1$, $\boldsymbol{P}2$, $\boldsymbol{B}1$, and $\boldsymbol{B}2$. The third phase was the *Treatment* phase from Weeks 0 to 19, also denoted as $\boldsymbol{S}01$ to $\boldsymbol{S}11$

A fourth *Follow-up* phase was appended at Months 3 and 6, also denoted as $\boldsymbol{F}3$ and $\boldsymbol{F}6$. This phase was analyzed separately.

4 Data Acquisition

All data managment was undertaken with the package **tidyverse** (Wickham, 2017; Wickham & Grolemund, 2017). Data was imported from Excel© with the package **readxl** (Wickham & Bryan, 2018).

The working data file was imported from the Excel file "../Working/PDAintervalsHxthruFU3-5-19.xlsx" and saved as the working data frame `NDA0`. Checks were made to ensure the working data frame was current. The data frame `NDA1` was created from `NDA0` The variable names were converted into simpler names.

```
> #options(warn=2)
> options(width=72)
> WorkFileName <- "Working/PDA intervals Hx thru FU 3-5-19.xlsx"
> if(!file.exists("NDA0.rds")){
    Status <- "Working data frame did not exist.  All analyses will be computed from original data."
    file.remove(list.files(pattern="*.rds"))
  } else if( file.mtime("NDA0.rds") < file.mtime(WorkFileName) ) {
    Status <- paste(
      "Working data frame had a younger date ",
      file.mtime("NDA0.rds"),
     "than original Excel file",
      file.mtime(WorkFileName),
      ".  All analyses will be computed from new data.")
    file.remove(list.files(pattern="*.rds"))
  } else {
    Status <- "Working data frame is current."
  }
> if(!file.exists("NDA0.rds")){
    NDA0 <- read_excel("Working/PDA intervals Hx thru FU 3-5-19.xlsx")
    saveRDS(NDA0, file="NDA0.rds")
  }
> NDA0 <- readRDS("NDA0.rds")
> NDA1 <- NDA0
```

```
> names(NDA1) <- c("ID",
       paste0("H", 0:4), "P1", "P2", "B1", "B2", paste0("S0", 1:9),
       paste0("S", 10:11), "F3", "F6")
```

Working data frame is current.

5 Data Checks

An intial check was made for duplicate cases

5.1 Duplicate Cases

I check for duplicate cases

```
>   if(anyDuplicated(NDA0$ID)){
     which(duplicated(NDA0$ID))
   }
```

There were 0 duplicate case IDs.

5.2 Missing Data

The first table counts the amount of missingness. Thus 188 (91.7%) had no missing data. The rest had 1 to 11 missing, from 1 (0.5%) with 1 missing point to 5 (2.4%) with 11 missing data points.

The second shows the case numbers for the 17 cases with missing data.

The missing data patterns were examined with the function *md.pattern* from R's **mice** (van Buuren & Groothuis-Oudshoorn, 2011). The first table shows the missing data patterns with 1 denoting present data and 0 denoting missing data. The leftmost column counts the number of cases for each pattern, the rightmost column counts the number missing data points in that pattern (row). The bottom row counts the number of cases with missing data at each time point. The rightmost bottom cell counts the number of missing data points.

There was no missing data for the initial 9 data points, those data crucial to determining the finite mixture model. The missing data pattern is monotone.

```
> N <- nrow(NDA1)
> table(apply(is.na(NDA1[,2:21]),1,sum))
  0   1   3   4   6   8   9  10  11
188   1   1   1   2   2   1   4   5
> round(table(apply(is.na(NDA1[,2:21]),1,sum))/N,3)
    0     1     3     4     6     8     9    10    11
0.917 0.005 0.005 0.005 0.010 0.010 0.005 0.020 0.024
> xmis <- which(apply(is.na(NDA1[,2:21]), 1,sum) >0)
> length(xmis)
[1] 17
> NDA1$ID[xmis]
 [1] 120 135 148 173 235 247 287 330 333 338 366 371 385 424 438 442 462
> md.pattern(NDA1[,2:21], plot=FALSE)
    H0 H1 H2 H3 H4 P1 P2 B1 B2 S01 S02 S03 S04 S05 S06 S07 S08 S09 S10
188  1  1  1  1  1  1  1  1  1   1   1   1   1   1   1   1   1   1   1
1    1  1  1  1  1  1  1  1  1   1   1   1   1   1   1   1   1   1   1
1    1  1  1  1  1  1  1  1  1   1   1   1   1   1   1   1   1   0   0
```

```
1    1  1  1  1  1  1  1  1  1   1   1   1   1   1   1   1   0   0   0
2    1  1  1  1  1  1  1  1  1   1   1   1   1   1   0   0   0   0   0
2    1  1  1  1  1  1  1  1  1   1   1   1   0   0   0   0   0   0   0
1    1  1  1  1  1  1  1  1  1   1   1   0   0   0   0   0   0   0   0
4    1  1  1  1  1  1  1  1  1   1   0   0   0   0   0   0   0   0   0
5    1  1  1  1  1  1  1  1  1   0   0   0   0   0   0   0   0   0   0
     0  0  0  0  0  0  0  0  0   5   9  10  12  12  14  14  15  16  16
    S11
188   1   0
1     0   1
1     0   3
1     0   4
2     0   6
2     0   8
1     0   9
4     0  10
5     0  11
     17 140
>
```

6 Data Summaries

We present two data summaries, each yielding slightly different information. The first data summary uses the R function *summary*.

```
> options(width=72)
> summary(NDA1)
       ID              H0               H1               H2       
 Min.   :103   Min.   :0.000   Min.   :0.000   Min.   :0.000  
 1st Qu.:194   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000  
 Median :290   Median :0.170   Median :0.140   Median :0.140  
 Mean   :289   Mean   :0.269   Mean   :0.266   Mean   :0.288  
 3rd Qu.:384   3rd Qu.:0.490   3rd Qu.:0.430   3rd Qu.:0.570  
 Max.   :478   Max.   :0.970   Max.   :1.000   Max.   :1.000  

       H3               H4               P1               P2        
 Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000  
 1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000  
 Median :0.140   Median :0.140   Median :0.140   Median :0.250  
 Mean   :0.272   Mean   :0.282   Mean   :0.298   Mean   :0.327  
 3rd Qu.:0.430   3rd Qu.:0.430   3rd Qu.:0.570   3rd Qu.:0.500  
 Max.   :1.000   Max.   :1.000   Max.   :1.000   Max.   :1.000  

       B1               B2              S01             S02      
 Min.   :0.000   Min.   :0.000   Min.   :0.00   Min.   :0.00  
 1st Qu.:0.110   1st Qu.:0.140   1st Qu.:0.29   1st Qu.:0.29  
 Median :0.370   Median :0.500   Median :0.62   Median :0.59  
 Mean   :0.435   Mean   :0.504   Mean   :0.57   Mean   :0.59  
 3rd Qu.:0.800   3rd Qu.:1.000   3rd Qu.:0.90   3rd Qu.:1.00  
 Max.   :1.000   Max.   :1.000   Max.   :1.00   Max.   :1.00  
```

```
                                   NA's   :5       NA's   :9
      S03            S04            S05            S06
 Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00
 1st Qu.:0.33   1st Qu.:0.43   1st Qu.:0.40   1st Qu.:0.43
 Median :0.69   Median :0.71   Median :0.71   Median :0.71
 Mean   :0.63   Mean   :0.64   Mean   :0.66   Mean   :0.66
 3rd Qu.:1.00   3rd Qu.:1.00   3rd Qu.:1.00   3rd Qu.:1.00
 Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00
 NA's   :10     NA's   :12     NA's   :12     NA's   :14
      S07            S08            S09            S10
 Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00
 1st Qu.:0.40   1st Qu.:0.43   1st Qu.:0.43   1st Qu.:0.43
 Median :0.71   Median :0.71   Median :0.78   Median :0.86
 Mean   :0.66   Mean   :0.66   Mean   :0.68   Mean   :0.68
 3rd Qu.:1.00   3rd Qu.:1.00   3rd Qu.:1.00   3rd Qu.:1.00
 Max.   :1.00   Max.   :1.00   Max.   :1.00   Max.   :1.00
 NA's   :14     NA's   :15     NA's   :16     NA's   :16
      S11             F3             F6
 Min.   :0.00   Min.   :0.00   Min.   :0.00
 1st Qu.:0.43   1st Qu.:0.44   1st Qu.:0.42
 Median :0.83   Median :0.83   Median :0.74
 Mean   :0.69   Mean   :0.67   Mean   :0.66
 3rd Qu.:1.00   3rd Qu.:0.99   3rd Qu.:1.00
 Max.   :1.00   Max.   :1.00   Max.   :1.00
 NA's   :17     NA's   :26     NA's   :31
```

The second summary uses the function *describe* from **Hmisc** (Harrell, 2019).

```
> options(width=72)
> describe((NDA1))
(NDA1)

 23  Variables      205  Observations
------------------------------------------------------------------------
ID
       n  missing distinct     Info     Mean      Gmd      .05      .10
     205        0      205        1    288.7    125.5    129.4    143.4
     .25      .50      .75      .90      .95
   194.0    290.0    384.0    437.6    455.6

lowest : 103 106 108 110 111, highest: 472 474 476 477 478
------------------------------------------------------------------------
H0
       n  missing distinct     Info     Mean      Gmd      .05      .10
     205        0       35    0.973   0.2689   0.3031     0.00     0.00
     .25      .50      .75      .90      .95
    0.00     0.17     0.49     0.69     0.80

lowest : 0.00 0.03 0.06 0.09 0.11, highest: 0.83 0.86 0.89 0.91 0.97
------------------------------------------------------------------------
H1
       n  missing distinct     Info     Mean      Gmd
```

```
     205        0        8    0.905   0.2661   0.3368

Value       0.00  0.14  0.29  0.43  0.57  0.71  0.86  1.00
Frequency     92    25    20    21    14    13     7    13
Proportion 0.449 0.122 0.098 0.102 0.068 0.063 0.034 0.063
------------------------------------------------------------------------
H2
       n  missing distinct     Info     Mean      Gmd
     205        0        8    0.925    0.288   0.3408

Value       0.00  0.14  0.29  0.43  0.57  0.71  0.86  1.00
Frequency     84    20    26    22    22    11     8    12
Proportion 0.410 0.098 0.127 0.107 0.107 0.054 0.039 0.059
------------------------------------------------------------------------
H3
       n  missing distinct     Info     Mean      Gmd
     205        0        8    0.933   0.2719   0.3238

Value       0.00  0.14  0.29  0.43  0.57  0.71  0.86  1.00
Frequency     80    29    30    20    17    12     6    11
Proportion 0.390 0.141 0.146 0.098 0.083 0.059 0.029 0.054
------------------------------------------------------------------------
H4
       n  missing distinct     Info     Mean      Gmd
     205        0        8    0.921   0.2823   0.3426

Value       0.00  0.14  0.29  0.43  0.57  0.71  0.86  1.00
Frequency     86    22    26    20    19    11     7    14
Proportion 0.420 0.107 0.127 0.098 0.093 0.054 0.034 0.068
------------------------------------------------------------------------
P1
       n  missing distinct     Info     Mean      Gmd
     205        0        8    0.934   0.2982   0.3557

Value       0.00  0.14  0.29  0.43  0.57  0.71  0.86  1.00
Frequency     80    27    25    20    15    14     7    17
Proportion 0.390 0.132 0.122 0.098 0.073 0.068 0.034 0.083
------------------------------------------------------------------------
P2
       n  missing distinct     Info     Mean      Gmd
     205        0        9     0.95    0.327   0.3691

Value       0.00  0.13  0.25  0.38  0.50  0.63  0.75  0.88  1.00
Frequency     72    23    21    14    27    14     8     7    19
Proportion 0.351 0.112 0.102 0.068 0.132 0.068 0.039 0.034 0.093
------------------------------------------------------------------------
B1
       n  missing distinct     Info     Mean      Gmd      .05      .10
     205        0       51    0.988    0.435   0.4155     0.00     0.00
     .25      .50      .75      .90      .95
    0.11     0.37     0.80     1.00     1.00
```

```
lowest : 0.00 0.04 0.05 0.06 0.07, highest: 0.88 0.89 0.92 0.94 1.00
--------------------------------------------------------------------------
B2
       n  missing distinct     Info     Mean      Gmd      .05      .10
     205        0       49    0.977   0.5041   0.4416     0.00     0.00
     .25      .50      .75      .90      .95
    0.14     0.50     1.00     1.00     1.00

lowest : 0.00 0.05 0.06 0.08 0.10, highest: 0.86 0.87 0.93 0.95 1.00
--------------------------------------------------------------------------
S01
       n  missing distinct     Info     Mean      Gmd      .05      .10
     200        5       58    0.985   0.5662   0.4065   0.0000   0.0000
     .25      .50      .75      .90      .95
  0.2857   0.6225   0.9023   1.0000   1.0000

lowest : 0.0000 0.0500 0.0556 0.0667 0.0700, highest: 0.9000 0.9091 0.9100 0.9500 1.0000
--------------------------------------------------------------------------
S02
       n  missing distinct     Info     Mean      Gmd      .05      .10
     196        9       51    0.973   0.5902   0.4062   0.0000   0.0000
     .25      .50      .75      .90      .95
  0.2857   0.5900   1.0000   1.0000   1.0000

lowest : 0.0000 0.0357 0.0741 0.0909 0.1000, highest: 0.9000 0.9231 0.9300 0.9643 1.0000
--------------------------------------------------------------------------
S03
       n  missing distinct     Info     Mean      Gmd      .05      .10
     195       10       47    0.961   0.6297    0.392   0.0000   0.0000
     .25      .50      .75      .90      .95
  0.3333   0.6875   1.0000   1.0000   1.0000

lowest : 0.0000 0.0500 0.0714 0.1111 0.1400, highest: 0.8889 0.9000 0.9286 0.9400 1.0000
--------------------------------------------------------------------------
S04
       n  missing distinct     Info     Mean      Gmd      .05      .10
     193       12       44    0.961   0.6393    0.386   0.0000   0.0000
     .25      .50      .75      .90      .95
  0.4286   0.7143   1.0000   1.0000   1.0000

lowest : 0.0000 0.0370 0.0476 0.1429 0.1538, highest: 0.8571 0.8600 0.9286 0.9333 1.0000
--------------------------------------------------------------------------
S05
       n  missing distinct     Info     Mean      Gmd      .05      .10
     193       12       39    0.952   0.6566   0.3889   0.0000   0.0000
     .25      .50      .75      .90      .95
  0.4000   0.7143   1.0000   1.0000   1.0000

lowest : 0.0000 0.0714 0.0909 0.1429 0.2500, highest: 0.8929 0.9000 0.9091 0.9500 1.0000
--------------------------------------------------------------------------
```

```
S06
       n  missing distinct     Info     Mean      Gmd      .05      .10
     191       14       38    0.949   0.6608   0.3831   0.0000   0.0500
     .25      .50      .75      .90      .95
  0.4286   0.7143   1.0000   1.0000   1.0000

lowest : 0.0000 0.0500 0.0588 0.0625 0.0952, highest: 0.8571 0.8800 0.8889 0.9286 1.0000
--------------------------------------------------------------------------------
S07
       n  missing distinct     Info     Mean      Gmd      .05      .10
     191       14       34    0.946   0.6603   0.3849   0.0000   0.0000
     .25      .50      .75      .90      .95
  0.4043   0.7143   1.0000   1.0000   1.0000

lowest : 0.0000 0.0714 0.0909 0.1429 0.1667, highest: 0.8139 0.8182 0.8571 0.9286 1.0000
--------------------------------------------------------------------------------
S08
       n  missing distinct     Info     Mean      Gmd      .05      .10
     190       15       37    0.951   0.6573   0.3899   0.0000   0.0000
     .25      .50      .75      .90      .95
  0.4286   0.7143   1.0000   1.0000   1.0000

lowest : 0.0000 0.0714 0.1111 0.1429 0.1667, highest: 0.8667 0.9091 0.9286 0.9643 1.0000
--------------------------------------------------------------------------------
S09
       n  missing distinct     Info     Mean      Gmd      .05      .10
     189       16       28    0.927   0.6789   0.3824   0.0000   0.0000
     .25      .50      .75      .90      .95
  0.4286   0.7778   1.0000   1.0000   1.0000

lowest : 0.0000 0.0714 0.1429 0.2857 0.3333, highest: 0.8571 0.8611 0.8889 0.9000 1.0000
--------------------------------------------------------------------------------
S10
       n  missing distinct     Info     Mean      Gmd      .05      .10
     189       16       28    0.939   0.6784    0.387   0.0000   0.0000
     .25      .50      .75      .90      .95
  0.4286   0.8571   1.0000   1.0000   1.0000

lowest : 0.0000 0.0357 0.1429 0.2000 0.2692, highest: 0.8889 0.8929 0.9000 0.9048 1.0000
--------------------------------------------------------------------------------
S11
       n  missing distinct     Info     Mean      Gmd      .05      .10
     188       17       72    0.945   0.6925   0.3735  0.00000  0.05849
     .25      .50      .75      .90      .95
 0.43000  0.83333  1.00000  1.00000  1.00000

lowest : 0.0000 0.0248 0.0504 0.0619 0.1071, highest: 0.9557 0.9565 0.9853 0.9903 1.0000
--------------------------------------------------------------------------------
F3
       n  missing distinct     Info     Mean      Gmd      .05      .10
     179       26       57    0.991   0.6741   0.3808    0.000    0.008
```

```
     .25      .50      .75      .90      .95 
   0.445    0.830    0.990    1.000    1.000 

lowest : 0.00 0.01 0.03 0.04 0.08, highest: 0.96 0.97 0.98 0.99 1.00
------------------------------------------------------------------------
F6 
       n  missing distinct     Info     Mean      Gmd      .05      .10 
     174       31       65    0.983   0.6646   0.3849   0.0000   0.0130 
     .25      .50      .75      .90      .95 
  0.4225   0.7350   0.9975   1.0000   1.0000 

lowest : 0.00 0.01 0.02 0.04 0.05, highest: 0.96 0.97 0.98 0.99 1.00
------------------------------------------------------------------------
```

7 Data Restructuring

The data frame NDA1 was converted from wide to long format into NDA2. The new variable NDA, representing the number of days abstinent was created from the orginal proportion data by multiplying the latter by 7 and rounding to an integer. The week numbers 0 to 19 were appended.

```
> NDA2 <- gather(NDA1[,1:21], week, NDA, H0:S11)
> NDA2 <- arrange(NDA2, ID, factor(week, levels=c("H0", "H1", "H2", "H3", "H4",
                                                  "P1", "P2", "B1", "B2",
                                                  "S01", "S02", "S03", "S04", "S05", "S06",
                                                  "S07", "S08", "S09", "S10", "S11")))
> NDA2$NDA <- round(NDA2$NDA*7)
> NDA2$W <- 0:19
> as.data.frame(head(NDA2, n=20))
    ID week NDA  W
1  103   H0   3  0
2  103   H1   0  1
3  103   H2   6  2
4  103   H3   3  3
5  103   H4   3  4
6  103   P1   4  5
7  103   P2   4  6
8  103   B1   2  7
9  103   B2   6  8
10 103  S01   5  9
11 103  S02   5 10
12 103  S03   5 11
13 103  S04   5 12
14 103  S05   3 13
15 103  S06   5 14
16 103  S07   6 15
17 103  S08   5 16
18 103  S09   4 17
19 103  S10   6 18
20 103  S11   6 19
> as.data.frame(tail(NDA2, n=20))
```

```
    ID week NDA  W
1  478   H0   1  0
2  478   H1   1  1
3  478   H2   1  2
4  478   H3   2  3
5  478   H4   1  4
6  478   P1   2  5
7  478   P2   1  6
8  478   B1   1  7
9  478   B2   1  8
10 478  S01   3  9
11 478  S02   4 10
12 478  S03   4 11
13 478  S04   4 12
14 478  S05   4 13
15 478  S06   4 14
16 478  S07   4 15
17 478  S08   4 16
18 478  S09   4 17
19 478  S10   4 18
20 478  S11   4 19
```

The data frame NDA2 was then reduced to all non-missing values. The cases with incomplete data are listed.

```
> NDA2 <- na.omit(NDA2)
> UID <- unique(NDA2$ID)
> table(NDA2$ID)
103 106 108 110 111 113 114 118 120 128 129 131 132 133 134 135 137 138
 20  20  20  20  20  20  20  20  16  20  20  20  20  20  20  19  20  20
139 140 143 144 145 146 148 150 151 154 155 156 160 161 165 166 167 168
 20  20  20  20  20  20   9  20  20  20  20  20  20  20  20  20  20  20
170 171 173 175 176 178 179 181 182 186 187 188 189 191 193 194 195 196
 20  20  12  20  20  20  20  20  20  20  20  20  20  20  20  20  20  20
197 198 201 202 204 205 206 207 208 209 210 211 214 215 217 218 220 223
 20  20  20  20  20  20  20  20  20  20  20  20  20  20  20  20  20  20
228 231 233 235 236 237 240 244 247 249 251 252 253 256 257 258 259 263
 20  20  20  11  20  20  20  20   9  20  20  20  20  20  20  20  20  20
265 266 269 271 274 276 279 280 281 286 287 288 290 291 292 295 297 300
 20  20  20  20  20  20  20  20  20  20  10  20  20  20  20  20  20  20
301 302 306 307 308 309 311 314 315 317 319 322 323 324 326 327 328 330
 20  20  20  20  20  20  20  20  20  20  20  20  20  20  20  20  20   9
331 333 334 335 338 339 340 341 343 346 351 353 355 356 361 364 366 367
 20  10  20  20  17  20  20  20  20  20  20  20  20  20  20  20  14  20
370 371 373 374 376 377 378 379 383 384 385 386 387 388 389 390 391 395
 20   9  20  20  20  20  20  20  20  20  10  20  20  20  20  20  20  20
396 398 400 403 404 409 411 413 415 416 418 421 423 424 425 429 430 433
 20  20  20  20  20  20  20  20  20  20  20  20  20   9  20  20  20  20
434 435 436 437 438 440 441 442 444 446 447 450 451 454 456 457 462 463
 20  20  20  20  12  20  20  14  20  20  20  20  20  20  20  20  10  20
469 470 472 474 476 477 478
 20  20  20  20  20  20  20
```

```
> UID[which(table(NDA2$ID) < 20)]
 [1] 120 135 148 173 235 247 287 330 333 338 366 371 385 424 438 442 462
```

8 Overview of Statistical Models

The statistical model was a finite mixture model (McLachlan & Peel, 2000) for which each basic model was a longitudinal binomial model with serial correlation. We postulated that the responses could exhibit different trajectories in the Distal-Pretreatment, and Proximal-Pretreatment, and Treatment phases. To accommodate these possibly different trajectories the binomial model used logistic link for a cubic spline with predetermined knots at Week -4 and Week 0. The knot at Week -4 corresponded to a possible change in trajectory based on previous findings, and the knot at Week 0 corresponded to a possible change in trajectory when participants began treatment. The spline model essentially allowed the estimation of three piecewise cubic polynomials smoothed at the two knots. The serial correlation was assumed to be first-order autoregressive [AR(1)] such that the correlation between responses diminishes diminished exponentially over time. To determine the number of classes, a second, truncated basic model was used in which considered only Weeks -8 to 0 were used with a single knot at -4 and for which responses were assumed independent. Once the number of classes was determined, the full basic model was re-estimated within each class.

8.1 The Basic Model

Random variables are underlined (Hemelrijk, 1966). Let $\underline{y}_{iw} \in \{0, \ldots, 7\}$ denote the number of days abstinent for Participant i during Week w, where $i = 1, \ldots, N$, and $w = -8, \ldots, 11$. Let $\theta(w)$ be the unkown propensity of a participant to be abstinent at Week w. The *basic model* was assumed to be the binomial function with AR(1) serial correlation:

$$\Pr(\underline{y}_{iw} = y) = \binom{7}{y} [\theta(w)]^y [(1 - \theta(w)]^{7-y} \quad \text{with} \quad \operatorname{cor}(\underline{y}_{iu}, \underline{y}_{iv}) = \rho^{|v-u|}. \tag{1}$$

(The choice of AR(1) serial correlation was based on diagnostics given by Section 10.2 on page 18 and Figure 1 on page 20.) The mean of $\underline{y}_{iw}$ is $E(\underline{y}_{iw}) = 7\theta(w)$, but the variance, which depends on ρ, is unknown. This model assumes independence of abstinence among the days within a week.

8.2 The Full Basic Model

We postulated that the responses could exhibit different trajectories in the Distal-Pretreatment, the Proximal-Pretreatment, and the Treatment phases. To accommodate these possibly different trajectories in $\theta(w)$, the *full basic model* used a logistic link of Equation (1) to a cubic spline with predetermined knots at Week -4 and Week 0. (The choice of a *cubic* spline over other possible degrees was based on model diagnostics given in Section 10.3 on page 19.) The knot at Week -4 corresponded to a possible change in trajectory when participants begin Proximal Pretreatment, and the knot at Week 0 corresponded to a possible change in trajectory when participants began treatment. The spline model essentially allowed the estimation of three piecewise cubic polynomials smoothed at the two knots separating each of the three phases of the time series. The *naive* parameterization of the full basic model is

$$\operatorname{logit} \theta(w) = \beta_0 + \beta_1 w + \beta_2 w^2 + \beta_3 w^3 + \delta_1 I_{[-4:\infty[}(w)(w+4)^3 + \delta_2 I_{[0:\infty[}(w) w^3, \tag{2}$$

where I is the indicator function (Hastie, Tibshirani & Friedman, 2001, Chap. 5).

For computational purposes, the basic model is not parameterized as in Equation (2), but in a more general parametrization for splines (Hastie et al., 2001, Chap. 5). This reparameterization is accomplished with the R function *bs* from the package **splines**.

Marginal predicted values were obtained from

$$\hat{\underline{y}}_w = 7\hat{\underline{\theta}}(w), \tag{3}$$

with

$$\operatorname{logit}\hat{\underline{\theta}}(w) = \hat{\underline{\beta}}_0 + \hat{\underline{\beta}}_1 w + \hat{\underline{\beta}}_2 w^2 + \hat{\underline{\beta}} w^3 + \hat{\underline{\delta}}_1 I_{[-4:\infty[}(w)(w+4)^3 + \hat{\underline{\delta}}_2 I_{[0:\infty)[}(w) w^3 \tag{4}$$

Because the estimators $\hat{\underline{\beta}}$'s and $\hat{\underline{\delta}}$'s in Equations (4) are asymptotically normal, 95% pointwise probability intervals for $\hat{\underline{y}}_w = 7\hat{\underline{\theta}}(w)$ were obtained by (1) extracting the estimated mean vector and covariance matrix from the GEE statistical, (2) simulating the coefficients by large number of multivariate normal deviates using the means and covariances, (3) transforming the simulated coefficents into $\hat{\theta}(w)$ by the inverse link of Equation (4), and then (4) finding the lower and upper 2.5% quantiles from the simulated distribution.

8.3 The Truncated Basic Model

To determine the number of classes K in the finite mixture model, detailed below, we confined the analysis to the two pretreatment phases. Thus we used a second *truncated* basic model, being Equation (1) but with independence between responses, i.e., $\operatorname{cor}(\underline{y}_{iu}, \underline{y}_{iv}) = 0$ and with Equation (2) truncated to $-8 \le w \le 0$ with a single knot at -4 i.e.,

$$\operatorname{logit}\theta(w) = \beta_0 + \beta_1 w + \beta_2 w^2 + \beta_3 w^3 + \delta_1 I_{[-4:\infty[}(w)(w+4)^3. \tag{5}$$

Again, the predicted values were obtained using Equation (3) using the estimators of the parameters of Equation (5). Pointwise 95% probability intervals were likewise obtained by simulation as outlined above.

8.4 The Finite Mixture Model

We further postulated that the NDA would be better modeled as a finite mixture of the basic model, Equation (1). That is, we proposed that there were $K \ge 1$ classes, to be determined, such that

$$\Pr(\underline{y}_{iw} = y) = \sum_{k=1}^{K} \pi_k \binom{7}{y} \left[\theta_k(w)\right]^y \left[(1-\theta_k(w)\right]^{7-y}, \tag{6}$$

where K is the number of classes, π_k is the probability of belonging in class k, and $\theta_k(w)$ has been specialized to class k (McLachlan & Peel, 2000)

To determine the number of classes K, I employed the truncated basic model, Equation (5). The number of classes K was determined by a combination of the Akaike Information Criterion (AIC) and heuristic reasoning. Each participant was then *hard-assigned* to the class showing maximum likelihood of membership (Fraley & Raftery, 2002).

Having determined K, the final mixture model was obtained by re-estimation of Equation (6) with Equation (2) for each class. Predicted values for each class k were obtained from

$$\hat{\underline{y}}_{wk} = 7\left[\hat{\underline{\theta}}_k(w)\right]. \tag{7}$$

For each class, the predicted values were obtained using Equation (3) with 95% pointwise probability intervals obtained by simulation.

9 Statistical Procedures

The estimation of the finite mixture of the models proceeded in three steps. At all steps, missing responses were effectively assumed to be missing-completely-at-random (MCAR) (Little & Rubin, 2002). Attempts to analyze the data with monotone missing-at-random (MAR) using inverse probability of missing (Robins, Rotnitzky & Zhao, 1995) did not work. However, as seen in Sections 5.2 on page 7 and 6 on page 8, the amount of missing data was so small (8.3%) in 17 out of 205 subjects that including the missing data model with analytic model proved ineffective. Missing responses (not cases) were deleted from the data.

9.1 Full Model

The first step determined the adequacy of the cubic spline, binomial-logistic, AR(1) regression model. The cubic spline regression with knots at -4 and 0 weeks was fitted to the response variable for the entire 20 weeks of observations for all subjects without class structure by generalized estimating equations (GEE) (Diggle, Liang & Zeger, 1994; Hardin & Hilbe, 2013; Laird, 2004; Schafer, 2006). The GEE procedure was implemented in the R package **geepack** (Halekoh, Højsgaard & Yan, 2006; Yan, 2002; Yan & Fine, 2004). GEE yields consistent estimates of model parameters that are asymptotically normal. Predicted values were obtained from the estimated model along with pointwise 95% confidence intervals via simulation. GEE requires no assumptions regarding prior distributions of model parameters, allows specification of serial correlation, and is robust against model misspecification. The estimated parameters are marginal estimates, as in repeated-measures analysis of variance, rather than individual estimates as in multilevel models. The spline parameters do not have a simple interpretations. GEE also requires that missing data be missing-completely-at-random rather than the more popular missing-at-random (Laird, 2004; Little & Rubin, 2002). Fortunately, the amount of missing data was small and monotone, with only 17 out of 205 subjects (8.3%) having any missing data, and the amount ranging from 1 to 11 missing responses per these 17 subjects. Model adequacy was determined by comparisons with mean responses at each data point, confidence intervals, data visualization, and assessing the serial correlation structure.

9.2 Truncated Model for Determining the Number of Classes

In the second step, the number of components in the mixture model (McLachlan & Peel, 2000) was determined from the first 8 weeks of observations, i.e., the combined Distal- and Proximal-Pretreatment phases. Thus the model was a cubic spline regression truncated at Week 0 with the single knot a -4 weeks. The finite mixture procedure assumed independence (zero serial correlation) of responses within subject. As seen in Section 6 on page 8, there was no missing data in this subset of the data. The mixture model was estimated with **flexmix** (Grün & Leisch, 2007; Grün & Leisch, 2007; Grün & Leisch, 2008; Leisch, 2004). The mixture was fitted for 1 to 10 possible classes. The choice of number of classes was based on information statistics (AIC, BIC) and their scree plots, heuristics, together with theoretical knowledge. The adequacy of the class separation was assessed by clustering diagnostics.

9.3 Full Models for Each Class

Once the number of classes was determined, the full cubic spline, binomial-logistic, AR(1) regression model was again used for all 20 observations within each class. Separate GEE analyses were used to estimate the parameters for each within-class model. Each model-based estimate was the expected value of the model belonging to its hard-assigned class (Fraley & Raftery, 2002). Point-wise 95% probabilitiy intervals were obtained by simulation. As previously mentioned, missing data was assumed to be MCAR.

9.4 Follow-up Analyses

The means and 95% confidence intervals were estimated for the follow-up responses at 3 and 6 months. The follow-up responses were analyzed separately and not incorporated into the 20-week analysis.

9.5 Additional Analyses

Two additional, post-hoc analysis were conducted. The first analysis compared the trajectories of the responses during the Proximal Pretreatment phase among the three classes. Models consisting of logistic-binomial, AR(1), quadratic polynomials with class-by-polynomial interactions were fitted to these data. Splines were unnecessary as there were no knots. There was no missing data.

The second analyses compared the change in NDAs from Week 0 to the end of treatment at Week 19, and to the two follow-up sessions among the three classes.

10 Full Model

10.1 Observed Means

I first obtain the observed means for the 20 weeks of observations.

```
> M <- geeglm(cbind(NDA, 7-NDA) ~ 0 + factor(W), id=ID, data=NDA2,
              family="binomial", waves=W)
> NDA2$Obs <- as.numeric(7*fitted(M, type="response"))
```

10.2 Within-Subject Correlation

The correlation matrix displays a decay in correlations with respect to lags from 0 to 19.. Figure **??** on page ?? displays the observed autocorrelations as black lines and points together with the average observed autocorrelations as a blue line and points with respect to lags. The decay indicates that an AR(1) structure for the residuals should be included in the model.

```
> options(warn=-15)
> options(width=72)
> X <- NDA1[, 2:21]
> X <- round(7*X)
> R <- matrix(NA, nrow=205, ncol=20)
> for (i in 1:20){
   y <- X[, i]
   ok <- !is.na(y)
   m <- glm(cbind(y, 7-y) ~ 1, family=binomial)
   R[ok,i] <- residuals(m, type="pearson")
 }
> X <- cor(R, method="pearson", use="pairwise.complete.obs")
> AR <- NA*X
> for(i in 1:20){
   j <- 21-i
   AR[1:j,i] <- X[i:20,i]
 }
> AR
       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11]
 [1,] 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000
 [2,] 0.894 0.830 0.873 0.857 0.836 0.854 0.747 0.654 0.735 0.862 0.803
 [3,] 0.927 0.782 0.817 0.789 0.791 0.614 0.524 0.633 0.695 0.693 0.801
 [4,] 0.910 0.707 0.725 0.734 0.574 0.386 0.547 0.623 0.539 0.715 0.751
 [5,] 0.866 0.612 0.688 0.584 0.391 0.438 0.548 0.446 0.509 0.659 0.724
 [6,] 0.761 0.612 0.539 0.388 0.413 0.428 0.388 0.481 0.495 0.631 0.686
 [7,] 0.734 0.483 0.364 0.433 0.412 0.305 0.411 0.428 0.466 0.601 0.686
 [8,] 0.568 0.310 0.401 0.430 0.307 0.327 0.357 0.405 0.459 0.604 0.621
 [9,] 0.384 0.345 0.403 0.332 0.323 0.266 0.346 0.388 0.453 0.531 0.622
[10,] 0.413 0.377 0.330 0.358 0.272 0.250 0.322 0.428 0.389 0.528 0.683
[11,] 0.438 0.308 0.350 0.315 0.262 0.228 0.362 0.364 0.392 0.567    NA
[12,] 0.350 0.299 0.282 0.286 0.252 0.274 0.300 0.342 0.425    NA    NA
[13,] 0.363 0.235 0.281 0.261 0.291 0.209 0.271 0.410    NA    NA    NA
[14,] 0.300 0.272 0.240 0.312 0.256 0.198 0.346    NA    NA    NA    NA
[15,] 0.299 0.222 0.287 0.258 0.224 0.243    NA    NA    NA    NA    NA
[16,] 0.261 0.248 0.241 0.249 0.267    NA    NA    NA    NA    NA    NA
```

```
[17,] 0.313 0.216 0.218 0.296    NA    NA    NA    NA    NA    NA    NA
[18,] 0.265 0.191 0.279    NA    NA    NA    NA    NA    NA    NA    NA
[19,] 0.247 0.225    NA    NA    NA    NA    NA    NA    NA    NA    NA
[20,] 0.293    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
      [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]
 [1,] 1.000 1.000 1.000 1.000 1.000 1.000 1.000 1.000     1
 [2,] 0.918 0.891 0.894 0.919 0.903 0.901 0.880 0.865    NA
 [3,] 0.864 0.860 0.894 0.910 0.882 0.890 0.852    NA    NA
 [4,] 0.861 0.836 0.889 0.872 0.855 0.884    NA    NA    NA
 [5,] 0.835 0.841 0.855 0.855 0.815    NA    NA    NA    NA
 [6,] 0.810 0.818 0.810 0.852    NA    NA    NA    NA    NA
 [7,] 0.775 0.804 0.844    NA    NA    NA    NA    NA    NA
 [8,] 0.763 0.840    NA    NA    NA    NA    NA    NA    NA
 [9,] 0.795    NA    NA    NA    NA    NA    NA    NA    NA
[10,]    NA    NA    NA    NA    NA    NA    NA    NA    NA
[11,]    NA    NA    NA    NA    NA    NA    NA    NA    NA
[12,]    NA    NA    NA    NA    NA    NA    NA    NA    NA
[13,]    NA    NA    NA    NA    NA    NA    NA    NA    NA
[14,]    NA    NA    NA    NA    NA    NA    NA    NA    NA
[15,]    NA    NA    NA    NA    NA    NA    NA    NA    NA
[16,]    NA    NA    NA    NA    NA    NA    NA    NA    NA
[17,]    NA    NA    NA    NA    NA    NA    NA    NA    NA
[18,]    NA    NA    NA    NA    NA    NA    NA    NA    NA
[19,]    NA    NA    NA    NA    NA    NA    NA    NA    NA
[20,]    NA    NA    NA    NA    NA    NA    NA    NA    NA
> ARmean <- apply(AR,1,mean, na.rm=TRUE)
> ARmean <- data.frame(ARmean=ARmean, Lag=0:19)
> rm(X, R, y, ok, m)
> ARdata <- data.frame(AR)
> ARdata<-gather(ARdata, key=Week, value=ar, X1:X20)
> ARdata$Lag <- 0:19
> ARplot <- ggplot(data=ARdata) +
   xlab("Lag") + ylab("Autocorrelation") +
   geom_line(aes(x=Lag, y=ar, group=Week), alpha=.3) +
   geom_line(data=ARmean, aes(x=Lag,y=ARmean), col="blue") +
   geom_point(aes(x=Lag, y=ar, group=Week), size=1) +
   geom_point(data=ARmean, aes(x=Lag,y=ARmean), col="blue", size=2) +
   scale_x_continuous(limits=c(0,19), breaks = 0:19, minor_breaks = NULL)
> pdf("NDA/ARplot0.pdf")
> print(ARplot)
> dev.off()
RStudioGD
        2
```

10.3 Linear, Quadratic, and Quartic Models

The fitted valued from inear, quadratic, cubic, and quartic spline models with knots at $H4 \equiv -4$ and $B2 \equiv 0$ were compared to the observed means. The observed means and fitted means are presented in Figure 2 on page 22.

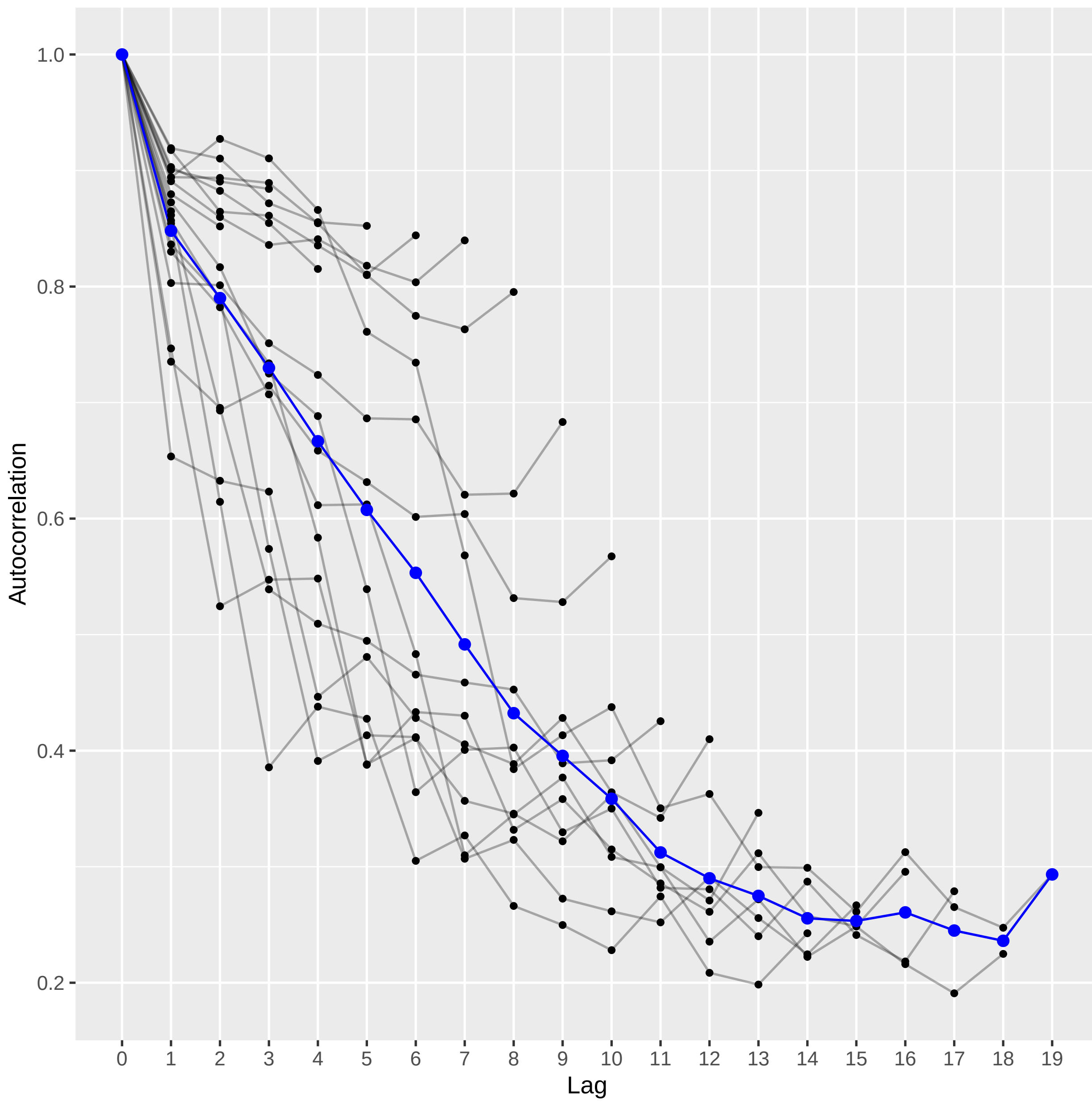

Figure 1: Observed autocorrelations (black lines and points) with observed average autocorrelation (blue line and points)

```
> NDA2$lin <- 7*fitted(geeglm(cbind(NDA, 7-NDA) ~ 1 + bs(W, knots=c(4,8), degree=1),
              id=ID, data=NDA2,
              family="binomial", waves=W, corstr = "ar1"))
> NDA2$quad <- 7*fitted(geeglm(cbind(NDA, 7-NDA) ~ 1 + bs(W, knots=c(4,8), degree=2),
               id=ID, data=NDA2,
               family="binomial", waves=W, corstr = "ar1"))
> NDA2$cub <- 7*fitted(geeglm(cbind(NDA, 7-NDA) ~ 1 + bs(W, knots=c(4,8), degree=3),
              id=ID, data=NDA2,
              family="binomial", waves=W, corstr = "ar1"))
> NDA2$quart <- 7*fitted(geeglm(cbind(NDA, 7-NDA) ~ 1 + bs(W, knots=c(4,8), degree=4),
                id=ID, data=NDA2,
                family="binomial", waves=W, corstr = "ar1"))
> ModelCompare <- ggplot(data=NDA2) +
    xlab("Weekly Interval") + ylab("Number of Days Abstinent per Week") +
    scale_x_continuous(limits=c(0,19), breaks = 0:19, minor_breaks = NULL,
    labels=c(paste0("H",0:4), "P1", "P2", "B1", "B2", paste0("S0", 1:9), "S10", "S11")) +
    scale_y_continuous(breaks=0:7, limits=c(0,7)) +
    geom_vline(xintercept = c(4,8), color="gray60") +
    geom_point(aes(x=W, y=Obs), size=3, alpha=.5) +
    geom_line(aes(x=W, y=lin), color="purple", size=1) +
    geom_line(aes(x=W, y=quad), color="blue", size=1) +
    geom_line(aes(x=W, y=cub), color="red", size=1) +
    geom_line(aes(x=W, y=quart), color="green", size=1) +
    theme_minimal()
> pdf("NDA/ModelCompare.pdf")
> print(ModelCompare)
> dev.off()
RStudioGD
        2
> NDA2 <- NDA2[, -c(6:9)]  #remove fit variables from data frame.
```

The linear model overestimated the means in the Proximal phase and underestimated them in the Treatment phase. The quadratic model both over- and underestimated the means in all phases. The cubic model estimated the means reasonably well. The quartic model did not improve estimation over the cubic.

10.4 Full Model Spline Fit

The model diagnostics conducted in Section 10.3 on page 19 step showed that the cubic spline was a reasonble model for the longitudinal, logistic-binomial regression and was thus chosen. The intial fit used the spline parameterization.

```
> G <- geeglm(cbind(NDA, 7-NDA) ~ 1 + bs(W, knots=c(4,8), degree=3), id=ID, data=NDA2,
             family="binomial", waves=W, corstr = "ar1")
```

I also obtain for confirmation purposes the naive model paramaterized as Equation (2).

```
> G_alt <- geeglm(cbind(NDA, 7-NDA) ~ 1 + W + I(W^2) + I(W^3) +
             I((W>4)*(W-4)^3) + I((W>8)*(W-8)^3),
             id=ID, data=NDA2, family="binomial", waves=W, corstr = "ar1")
```

The spline model's parameters are given in the Table 2 on page 23.
The naive model's for Equation (2) parameters are given in the Table 3 on page 24.
Here I store the fitted values and obtain the lower and upper confidence bounds by simulation.

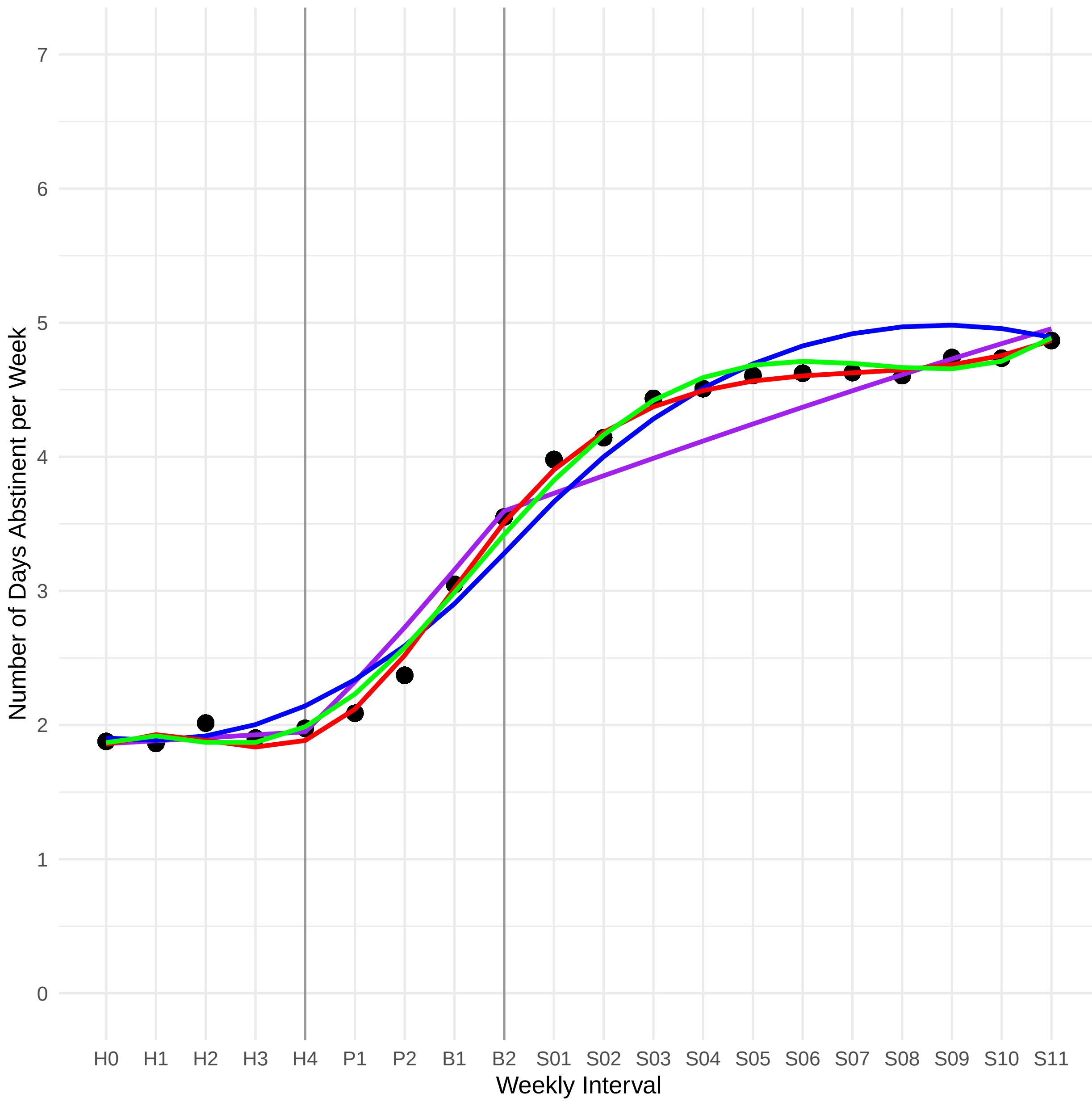

Figure 2: Fitted values from linear (purple line), quadratic (blue line), cubic (red line), and quartic (green line) binomial regression splines plotted against the observed mean (black points).

Table 2: Estimated Parameters the Full Cubic Spline Logistic-Binomial with AR(1) Correlation

```
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + bs(W, knots = c(4,
    8), degree = 3), family = "binomial", data = NDA2, id = ID,
    waves = W, corstr = "ar1")

 Coefficients:
                                    Estimate Std.err   Wald Pr(>|W|)
(Intercept)                           -1.019   0.100 103.25  < 2e-16
bs(W, knots = c(4, 8), degree = 3)1    0.160   0.080   3.99  0.04568
bs(W, knots = c(4, 8), degree = 3)2   -0.379   0.110  11.81  0.00059
bs(W, knots = c(4, 8), degree = 3)3    2.141   0.202 112.33  < 2e-16
bs(W, knots = c(4, 8), degree = 3)4    1.503   0.167  81.44  < 2e-16
bs(W, knots = c(4, 8), degree = 3)5    1.845   0.132 195.93  < 2e-16

(Intercept)                         ***
bs(W, knots = c(4, 8), degree = 3)1 *
bs(W, knots = c(4, 8), degree = 3)2 ***
bs(W, knots = c(4, 8), degree = 3)3 ***
bs(W, knots = c(4, 8), degree = 3)4 ***
bs(W, knots = c(4, 8), degree = 3)5 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.526  0.0216

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha    0.903  0.0112
Number of clusters:   205   Maximum cluster size: 20
```

Table 3: Estimated Parameters the Naive Full Cubic Logistic-Binomial with AR(1) Correlation

```
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + W + I(W^2) + I(W^3) +
    I((W > 4) * (W - 4)^3) + I((W > 8) * (W - 8)^3), family = "binomial",
    data = NDA2, id = ID, waves = W, corstr = "ar1")

 Coefficients:
                       Estimate  Std.err   Wald Pr(>|W|)
(Intercept)            -1.01930  0.10031 103.25  < 2e-16 ***
W                       0.11994  0.06002   3.99  0.04568 *
I(W^2)                 -0.08052  0.03095   6.77  0.00929 **
I(W^3)                  0.01296  0.00358  13.12  0.00029 ***
I((W > 4) * (W - 4)^3) -0.02213  0.00495  19.96  7.9e-06 ***
I((W > 8) * (W - 8)^3)  0.01084  0.00182  35.63  2.4e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.526  0.0216

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha    0.903  0.0112
Number of clusters:   205   Maximum cluster size: 20
```

```
> NDA2$Fit <- as.numeric(7*predict(G, type="response"))
> NDA2$UFit <- 0
> NDA2$LFit <- 0
> b <- coef(G)
> names(b) <- NULL
> Sigma <- G$geese$vbeta
> X <- cbind(1, G$model$`bs(W, knots = c(4, 8), degree = 3`)
> K <- 1000
> R <- mvrnorm(K, mu=b, Sigma=Sigma)
> U <- 7*plogis(X %*% (t(R)))
> u <- apply(U,1,quantile, prob=.025)
> v <- apply(U,1,quantile, prob=.975)
> NDA2$UFit <- u
> NDA2$LFit <- v
> head(as.data.frame(NDA2), n=20)
    ID week NDA  W  Obs  Fit UFit LFit
1  103   H0   3  0 1.88 1.86 1.60 2.12
2  103   H1   0  1 1.86 1.93 1.66 2.23
3  103   H2   6  2 2.01 1.89 1.61 2.19
4  103   H3   3  3 1.90 1.84 1.58 2.13
5  103   H4   3  4 1.98 1.88 1.62 2.17
6  103   P1   4  5 2.09 2.12 1.85 2.41
7  103   P2   4  6 2.37 2.52 2.23 2.82
8  103   B1   2  7 3.05 3.02 2.73 3.31
9  103   B2   6  8 3.55 3.51 3.20 3.81
10 103  S01   5  9 3.98 3.90 3.58 4.21
11 103  S02   5 10 4.14 4.18 3.86 4.49
12 103  S03   5 11 4.44 4.37 4.06 4.67
13 103  S04   5 12 4.51 4.49 4.19 4.79
14 103  S05   3 13 4.61 4.57 4.26 4.86
15 103  S06   5 14 4.62 4.60 4.29 4.90
16 103  S07   6 15 4.63 4.63 4.31 4.93
17 103  S08   5 16 4.61 4.65 4.31 4.96
18 103  S09   4 17 4.74 4.69 4.35 5.00
19 103  S10   6 18 4.74 4.76 4.43 5.06
20 103  S11   6 19 4.87 4.87 4.54 5.17
```

Here I generate the plot for the full model.

```
> NDAPlot0 <- ggplot(data=NDA2) +
    xlab("Weekly Interval") + ylab("Number of Days Abstinent per Week") +
    scale_x_continuous(limits=c(0,19), breaks = 0:19, minor_breaks = NULL,
    labels=c(paste0("H",0:4), "P1", "P2", "B1", "B2", paste0("S0", 1:9), "S10", "S11")) +
    scale_y_continuous(breaks=0:7, limits=c(0,7)) +
    geom_vline(xintercept = c(4,8), color="gray60") +
    geom_point(aes(x=W, y=Obs), size=1.5) +
    geom_line(aes(x=W, y=Fit), color="red", alpha=.5) +
    geom_ribbon(aes(x=W, ymin=LFit, ymax=UFit), fill="red", alpha=.2) +
    theme_minimal()
> pdf("NDA/NDAPlot0.pdf")
> print(NDAPlot0)
> dev.off()
```

```
null device
          1
```

The observed means and the fitted means from the full model are presented in Figure 3 on the following page.

```
> ARdata <- data.frame(AR)
> ARdata<-gather(ARdata, key=Week, value=ar, X1:X20)
> ARdata$Lag <- 0:19
> alpha <- G$geese$alpha
> alpha.se <- sqrt(G$geese$valpha[1,1])
> Loalpha <- alpha-2*alpha.se
> Upalpha <- alpha+2*alpha.se
> ARdata$alpha <- alpha^(ARdata$Lag)
> ARdata$alphaLo <- Loalpha^(ARdata$Lag)
> ARdata$alphaUp <- Upalpha^(ARdata$Lag)
> ARdata <- na.omit(ARdata)
> ARplot <- ggplot(data=ARdata) +
   xlab("Lag") + ylab("Autocorrelation") +
   geom_line(aes(x=Lag, y=ar, group=Week), alpha=.2) +
   geom_point(data=ARmean, aes(x=Lag,y=ARmean), col="blue", size=2) +
   geom_line(aes(x=Lag, y=alpha), color="red", size=1) +
   geom_ribbon(aes(x=Lag, ymin=alphaLo, ymax=alphaUp), fill="red", alpha=.1) +
   scale_x_continuous(limits=c(0,19), breaks = 0:19, minor_breaks = NULL)
> pdf("NDA/ARplot.pdf")
> print(ARplot)
> dev.off()
null device
          1
```

Diagnostics for the autocorrelation are presented in Figure 4 on page 28, which updates Figure 1 on page 20 with the estimated autocorrelation. The observed autcorrelations follow the same pattern as the estimated autocorrelation. All but two of the average observed autocorrelations fall within the 95% probability ribbon for the estimated autocorrelation. These results indicate that the assumption of AR(1) autocorrelation should be adequate.

11 Determining the Number of Classes

We had reasons to believe that the data comprised different mixtures according to Equation (6) with Equation (5). The NDA3 was created for the pretreatment data by truncating NDA2 to the first 8 weeks. Recall that this data set had no missing data.

11.1 Create Pretreatment Dataframe

```
> NDA3 <- NDA2[NDA2$W<=8,]
> head(NDA3, n=9)
# A tibble: 9 x 8
     ID week    NDA     W   Obs   Fit  UFit  LFit
  <dbl> <chr> <dbl> <int> <dbl> <dbl> <dbl> <dbl>
1   103 H0        3     0  1.88  1.86  1.60  2.12
2   103 H1        0     1  1.86  1.93  1.66  2.23
```

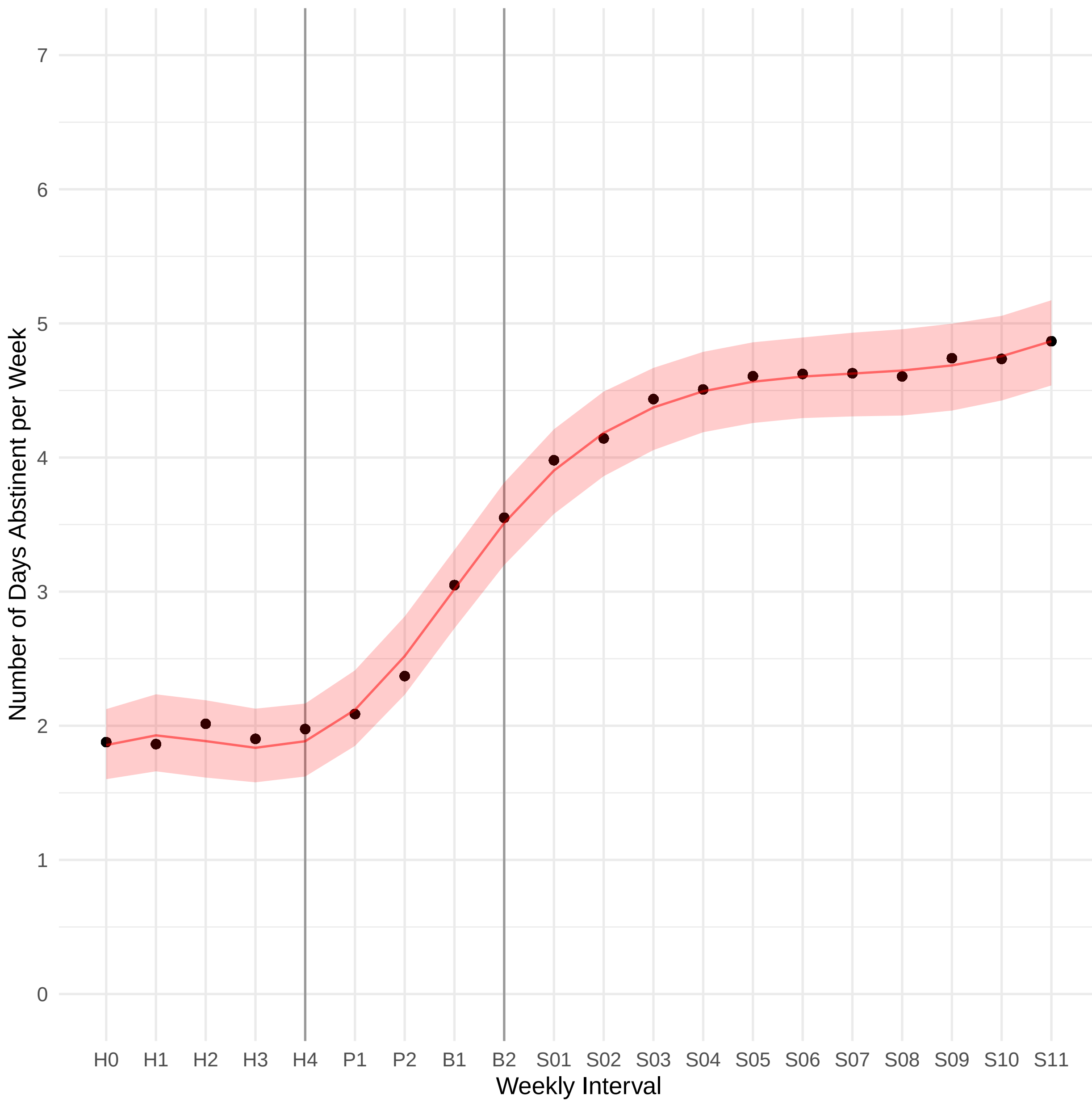

Figure 3: Observed (black circles) means and full model fitted values (red line) together with the 95% pointwise probability ribbon.

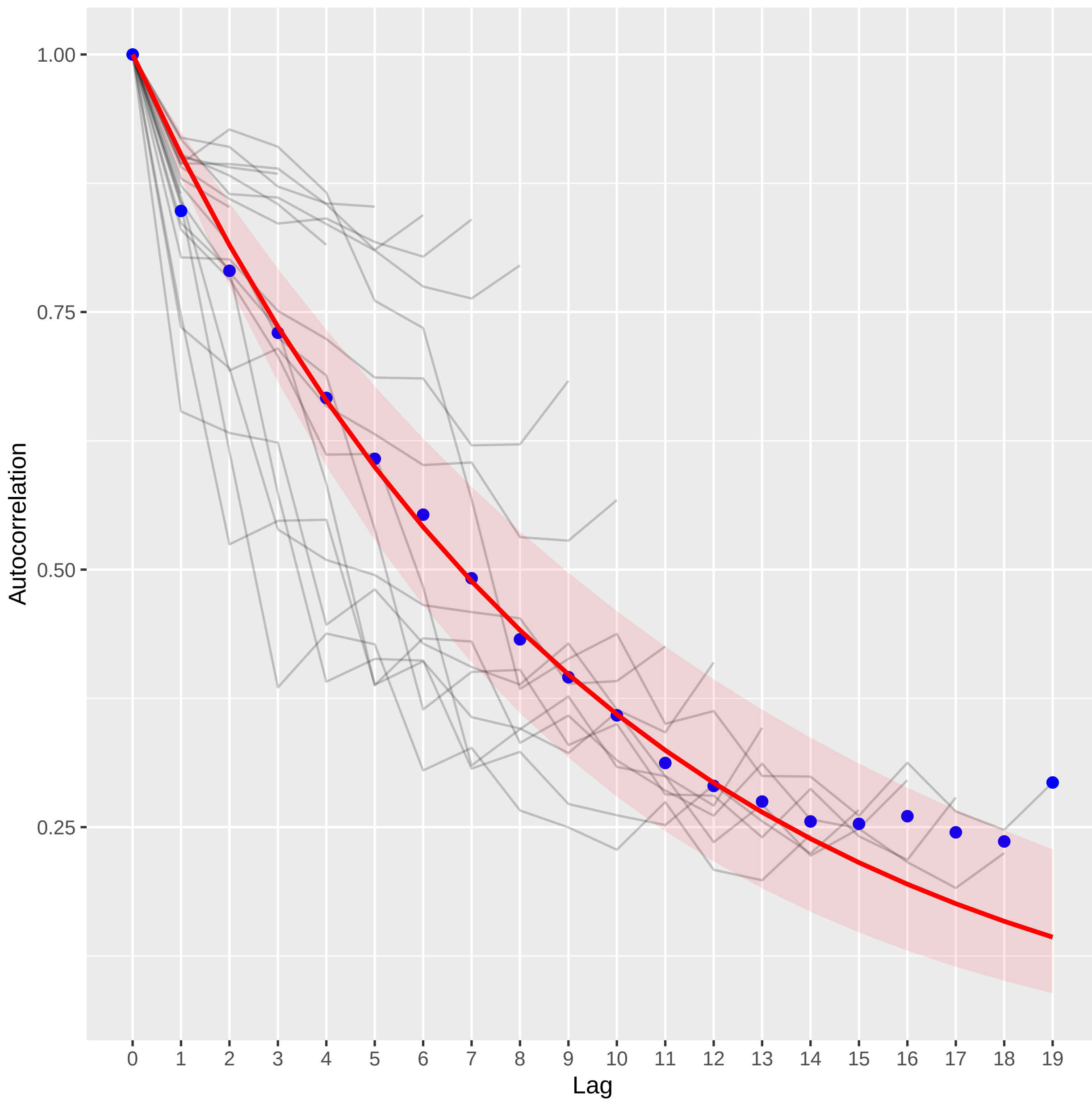

Figure 4: Observed average autocorrelations (blue points) with estimated autocorrelation (red line) with 2 standard error ribbon. Observed autocorrelations (gray lines) are presented in background.

```
3   103 H2        6     2  2.01  1.89  1.61  2.19
4   103 H3        3     3  1.90  1.84  1.58  2.13
5   103 H4        3     4  1.98  1.88  1.62  2.17
6   103 P1        4     5  2.09  2.12  1.85  2.41
7   103 P2        4     6  2.37  2.52  2.23  2.82
8   103 B1        2     7  3.05  3.02  2.73  3.31
9   103 B2        6     8  3.55  3.51  3.20  3.81
> tail(NDA3, n=9)
# A tibble: 9 x 8
     ID week    NDA     W   Obs   Fit  UFit  LFit
  <dbl> <chr> <dbl> <int> <dbl> <dbl> <dbl> <dbl>
1   478 H0        1     0  1.88  1.86  1.60  2.12
2   478 H1        1     1  1.86  1.93  1.66  2.23
3   478 H2        1     2  2.01  1.89  1.61  2.19
4   478 H3        2     3  1.90  1.84  1.58  2.13
5   478 H4        1     4  1.98  1.88  1.62  2.17
6   478 P1        2     5  2.09  2.12  1.85  2.41
7   478 P2        1     6  2.37  2.52  2.23  2.82
8   478 B1        1     7  3.05  3.02  2.73  3.31
9   478 B2        1     8  3.55  3.51  3.20  3.81
```

11.2 Finite Mixture Models

```
> if (!file.exists("SFM.rds")) {
   SFM <- stepFlexmix(cbind(NDA,7-NDA)~1 + bs(W, knots=4) |ID, data=NDA3,
     model = FLXMRglm(family = "binomial"), k=1:10, nrep=10)
   SFM.rds <- saveRDS(SFM, file="SFM.rds")
 }
> SFM <- readRDS("SFM.rds")
> SFM
Call:
stepFlexmix(cbind(NDA, 7 - NDA) ~ 1 + bs(W, knots = 4) |
    ID, data = NDA3, model = FLXMRglm(family = "binomial"),
    k = 1:10, nrep = 10)

   iter converged k k0 logLik   AIC   BIC   ICL
1     2      TRUE 1  1  -5099 10208 10236 10236
2     8      TRUE 2  2  -3450  6921  6982  6983
3    13      TRUE 3  3  -3113  6261  6355  6359
4    14      TRUE 4  4  -2925  5896  6023  6030
5    24      TRUE 5  5  -2779  5616  5776  5783
6    36      TRUE 6  6  -2697  5464  5657  5668
7    21      TRUE 7  7  -2645  5373  5599  5611
8    19      TRUE 7  8  -2648  5378  5604  5617
9    51      TRUE 8  9  -2608  5309  5569  5584
10   22      TRUE 8 10  -2604  5302  5561  5572
> getModel(SFM, "AIC")
Call:
stepFlexmix(cbind(NDA, 7 - NDA) ~ 1 + bs(W, knots = 4) |
    ID, data = NDA3, model = FLXMRglm(family = "binomial"),
```

```
    k = 10, nrep = 10)

Cluster sizes:
  1   2   3   4   5   6   7   8 
387 360 126 117 369 126 189 171 

convergence after 22 iterations
> getModel(SFM, "BIC")
Call:
stepFlexmix(cbind(NDA, 7 - NDA) ~ 1 + bs(W, knots = 4) | 
    ID, data = NDA3, model = FLXMRglm(family = "binomial"), 
    k = 10, nrep = 10)

Cluster sizes:
  1   2   3   4   5   6   7   8 
387 360 126 117 369 126 189 171 

convergence after 22 iterations
> AIC(SFM)
    1     2     3     4     5     6     7     8     9    10 
10208  6921  6261  5896  5616  5464  5373  5378  5309  5302 
> BIC(SFM)
    1     2     3     4     5     6     7     8     9    10 
10236  6982  6355  6023  5776  5657  5599  5604  5569  5561 


> if (!file.exists("AICB.rds")) {
  B <- 1000
  AICB <- BICB <- matrix(0, nrow=B, ncol=10)
  for (b in 1:B){
    XB1 <- NDA1[ ,2:10]
    XB1 <- XB1[sample(N, replace=TRUE), ]
    XB1$ID <- 1:205                         #create unique IDs for gather
    XB2 <- gather(XB1, week, NDA, H0:B2)
    XB2 <- arrange(XB2, ID, factor(week, levels=c("H0", "H1", "H2", "H3", "H4",
                                                  "P1", "P2", "B1", "B2")))
    XB2$NDA <- round(XB2$NDA*7)
    XB2$W <- 0:8
   #head(XB2, n=18)
   #tail(XB2, n=18)
    XFM <- stepFlexmix(cbind(NDA,7-NDA)~1 + bs(W, knots=4) |ID, data=XB2,
      model = FLXMRglm(family = "binomial"), k=1:10, nrep=10)
    AICB[b, ] <- AIC(XFM)
    BICB[b, ] <- BIC(XFM)
  }
 saveRDS(AICB, file="AICB.rds")
 saveRDS(BICB, file="BICB.rds")
 }
> AICB <- readRDS("AICB.rds")
> BICB <- readRDS("BICB.rds")
> AICQ <- apply(AICB, 2, quantile, prob=c(.025,.25, .5, .75, .975))
```

```
> AICQ <- t(AICQ)
> colnames(AICQ) <- c("Q025", "Q250", "Q500", "Q750", "Q975")
> AICQ <- data.frame(AICQ)
> AICQ$Classes <- 1:10
>
> #BICQ <- apply(BICB, 2, quantile, prob=c(.025,.25, .5, .75, .975))
> #BICQ <- t(BICQ)
> #colnames(BICQ) <- c("Q025", "Q250", "Q500", "Q750", "Q975")
> #BICQ <- data.frame(BICQ)
> #BICQ$Classes <- 1:10
>
> #ggplot(data=BICQ) +
> #  xlab("Estimated Number of Classes") + ylab("BIC") +
> #  scale_x_continuous(limits=c(1,10), breaks = 1:10, minor_breaks = NULL) +
> #  geom_pointrange(aes(x=Classes, y=Q500, ymin=Q025, ymax=Q975))
>
```

The scree plot for the AICs is given in Figure 5 on the next page.

```
> ScreePlot <- ggplot(data=AICQ) +
    xlab("Estimated Number of Classes") + ylab("AIC") +
    scale_x_continuous(limits=c(1,10), breaks = 1:10, minor_breaks = NULL) +
    geom_linerange(aes(x=Classes, ymin=Q025, ymax=Q975), size=1, color="red")+
    geom_linerange(aes(x=Classes, ymin=Q250, ymax=Q750), size=2, color="red", alpha=.6) +
    geom_point(aes(x=Classes, y=Q500), size=3, color="red") +
    geom_vline(xintercept = 3, color="blue", linetype="dashed")
> pdf(file="NDA/ScreePlot.pdf")
> print(ScreePlot)
> dev.off()
null device
          1
```

The decision was made to use 3 classes. Participants were classified into their respective classes by *hard assignment* to the class with maximum posterior probability (Fraley & Raftery, 2002).

I refit the 3-class model to get model parameters.

```
> FM3 <- flexmix(cbind(NDA,7-NDA)~1 + bs(W, knot=c(4)) |ID, data=NDA3,
                 model = FLXMRglm(family = "binomial"), k=3)
> saveRDS(FM3, file="FM3.rds")
> FM3r <- flexmix::refit(FM3)
> saveRDS(FM3r, file="FM3r.rds")
> #FM <- readRDS("FM.rds")
> #FMr <- readRDS("FMr.rds")
```

The 3-class classification results are given in Table 4 on page 37.

A visual comparison of the regression coefficient estimates is given in Figure 6 on page 33.

The assignment diagnostics are given in Figure 7 to 9 on pages 34–36. The overlapping class assignments are given in orange. Note the assignments show very little overlap.

12 Restructure Data for Plotting

In what follows I create the appropriate data frames for plotting the data. The creation of these data frames is complicated by the fact that the 3 classes created from the analysis are not invariant with respect to names. Thus the same results found in Class 1 in one run could be found in Class 3 in another.

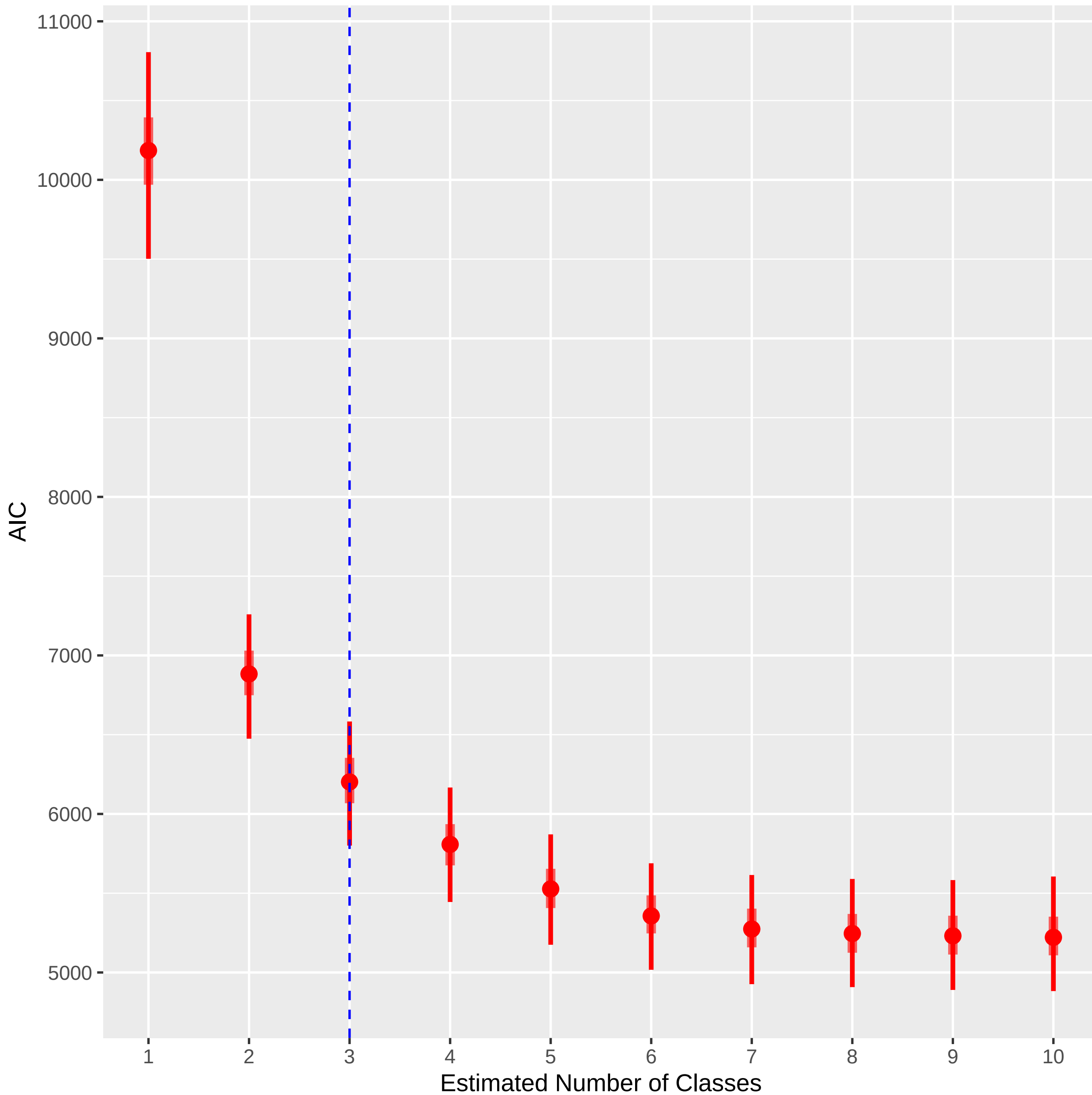

Figure 5: Scree plot for the estimated number of classes based on 1000 boostrapped AICs. The point is the median, the thick red line is the 50% probability interval, and the the thin line is the 95% probability interval. The dashed blue line denotes the chosen nuumber of classes.

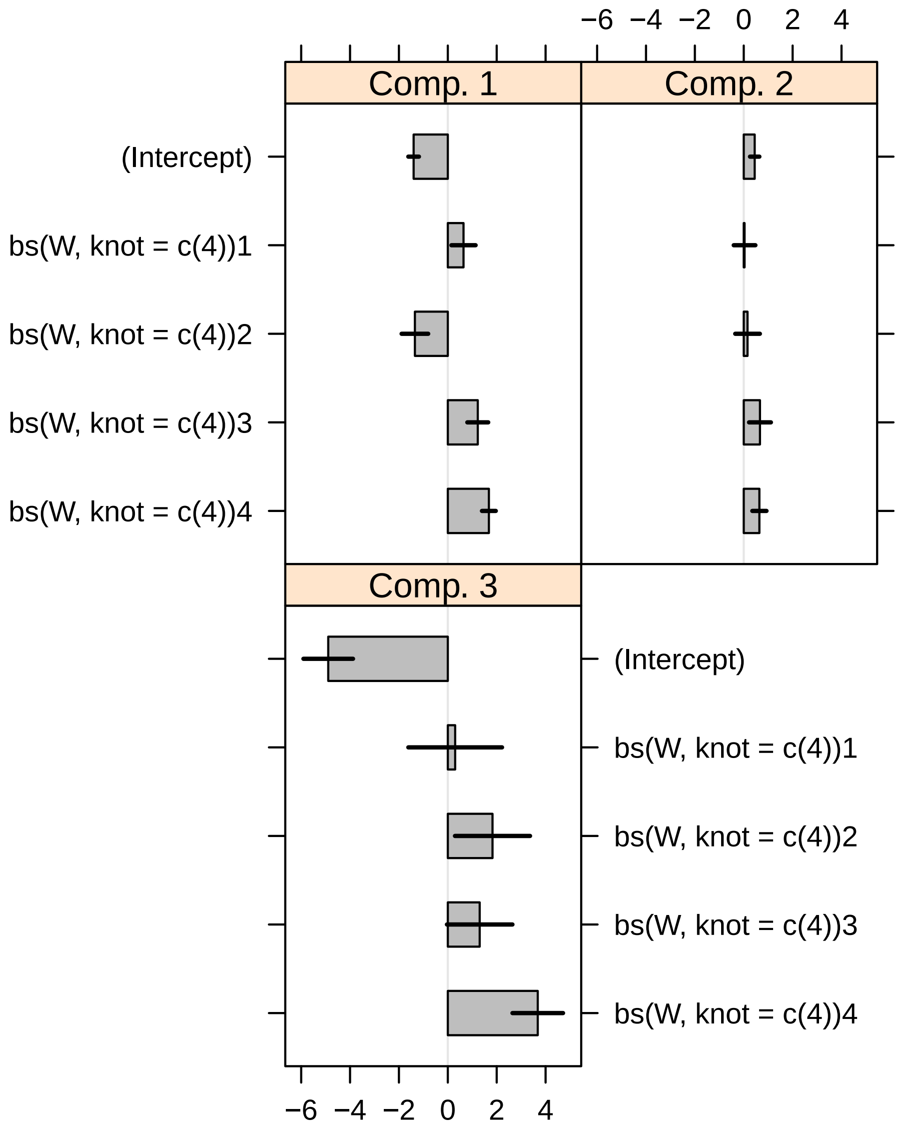

Figure 6: Visual comparison of regression coefficient estimates for the three classes

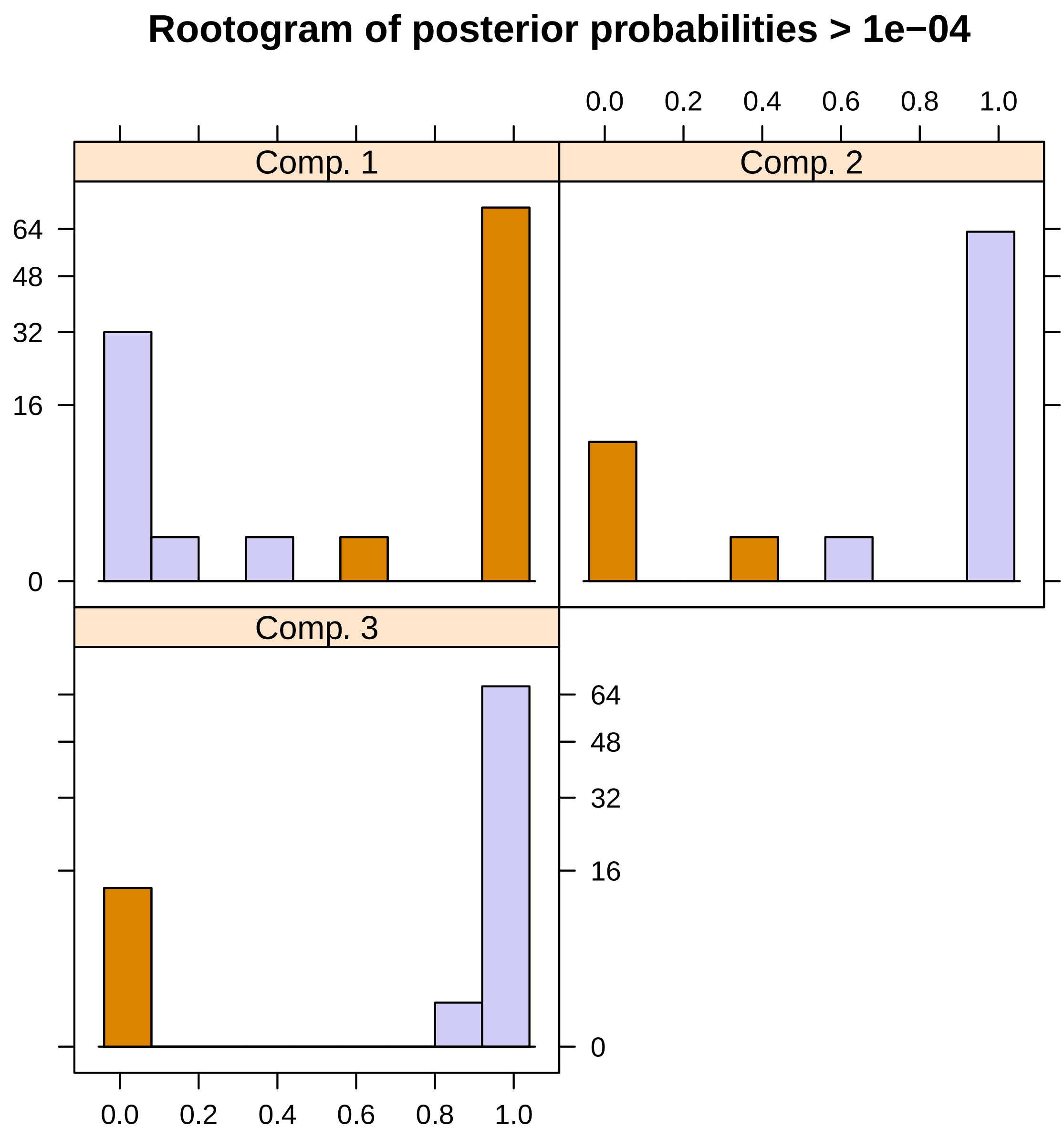

Figure 7: The assignment proportions to the three classes: Class 1

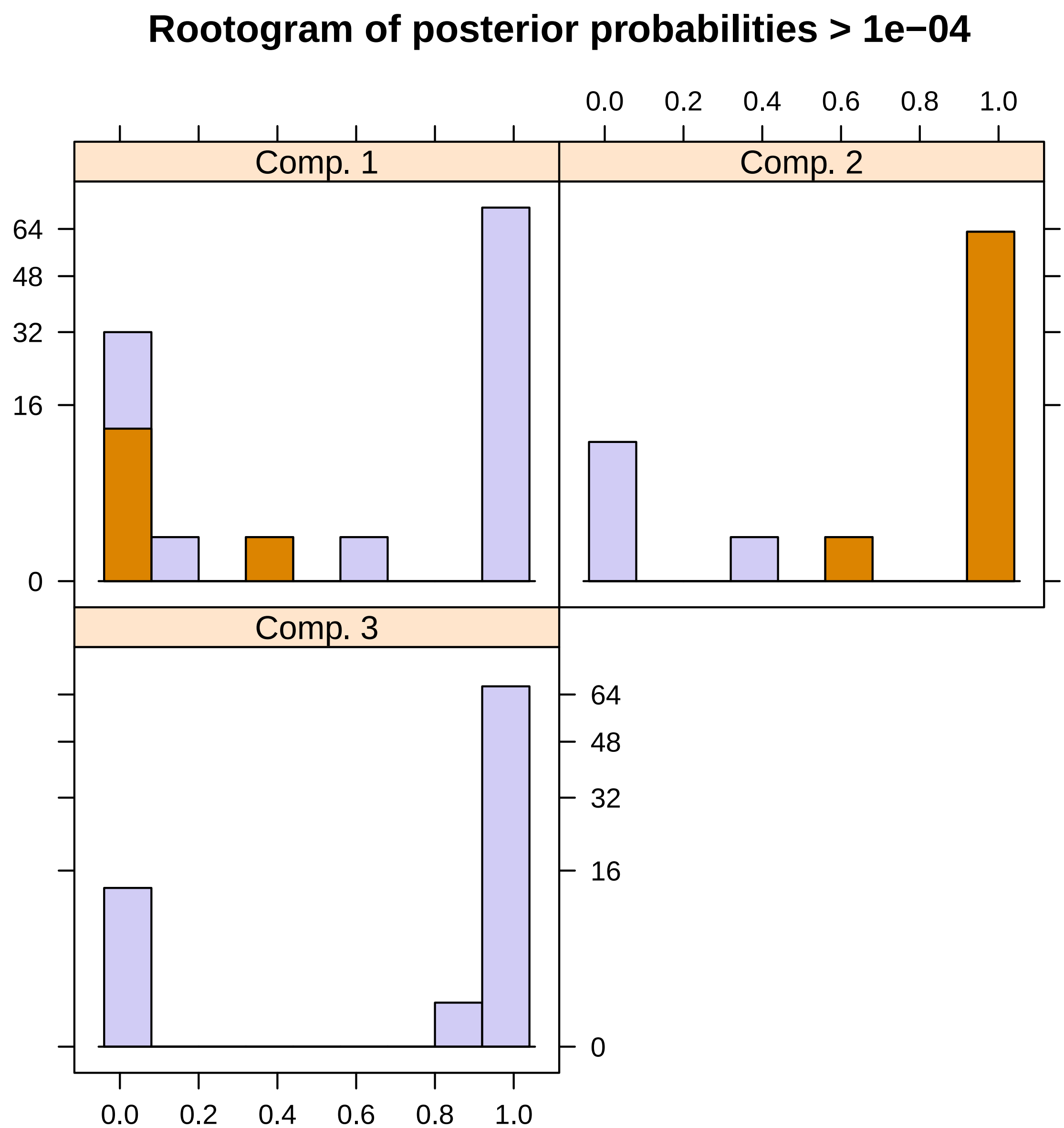

Figure 8: The assignment proportions to the three classes: Class 2

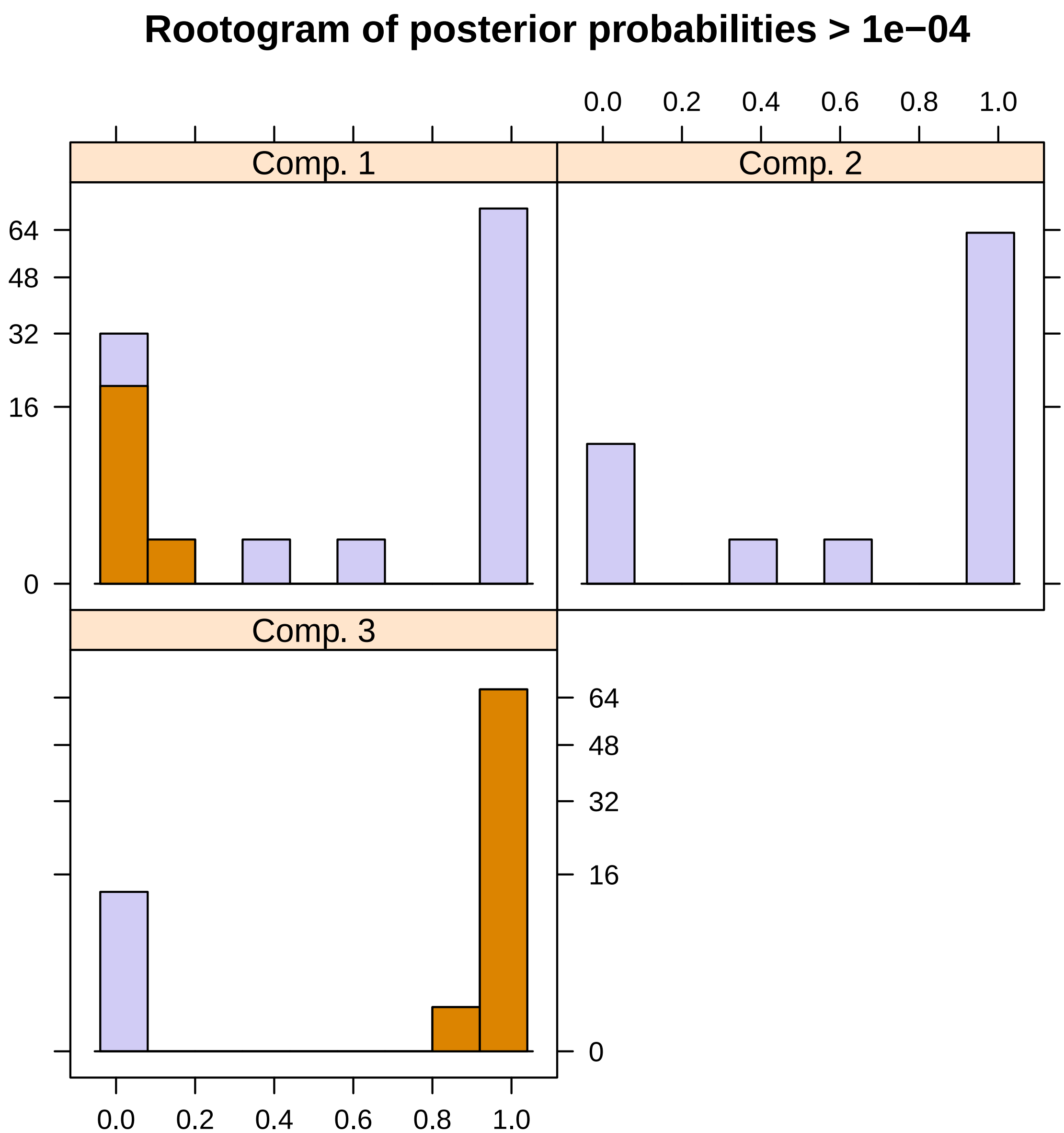

Figure 9: The assignment proportions to the three classes: Class 3

Table 4: Classification Results for Three-Class Binomial-Logistic Model

```
Call:
flexmix(formula = cbind(NDA, 7 - NDA) ~ 1 + bs(W, knot = c(4)) |
    ID, data = NDA3, k = 3, model = FLXMRglm(family = "binomial"))

       prior size post>0 ratio
Comp.1 0.357  657    963 0.682
Comp.2 0.313  576    675 0.853
Comp.3 0.331  612    729 0.840

'log Lik.' -3113 (df=17)
AIC: 6261   BIC: 6355
```

12.1 Append observed and Fitted Means to NDA Data Frame

Here I create the observed and fitted means for the quadratic binomial model. I order the classes from low to high in terms of estimated intercepts. In general, this may not the best way to identify the classes, but it works here.

```
> n <- nrow(NDA3)
> m <- g <- numeric(n)
> k <- FM3@cluster
> M1 <- geeglm(cbind(NDA, 7-NDA)~ 0 + factor(W), id=ID, data=NDA3, family="binomial",
              waves=W, subset=k==1)
> M2 <- geeglm(cbind(NDA, 7-NDA)~ 0 + factor(W), id=ID, data=NDA3, family="binomial",
              waves=W, subset=k==2)
> M3 <- geeglm(cbind(NDA, 7-NDA)~ 0 + factor(W), id=ID, data=NDA3, family="binomial",
              waves=W, subset=k==3)
> m[k==1] <- fitted(M1)
> m[k==2] <- fitted(M2)
> m[k==3] <- fitted(M3)
> G1 <- geeglm(cbind(NDA, 7-NDA)~1+ bs(W, knot=c(4)), id=ID, data=NDA3, family="binomial",
                          waves=W, corstr = "ar1", subset=k==1)
> G2 <- geeglm(cbind(NDA, 7-NDA)~1+ bs(W, knot=c(4)), id=ID, data=NDA3, family="binomial",
                          waves=W, corstr = "ar1", subset=k==2)
> G3 <- geeglm(cbind(NDA, 7-NDA)~1+ bs(W, knot=c(4)), id=ID, data=NDA3, family="binomial",
                          waves=W, corstr = "ar1", subset=k==3)
> g[k==1] <- fitted(G1)
> g[k==2] <- fitted(G2)
> g[k==3] <- fitted(G3)
> m <- m*7
> g <- g*7
> #ord <- order(c(max(m[k==1]),max(m[k==2]),max(m[k==3])))
> ord <- order( c(coef(G1)[1],coef(G2)[1],coef(G3)[1]) )
> NDA3$clus <- factor(FM3@cluster, levels=rev(ord), labels=paste0("Class", 1:3))
> NDA3$NDAobs <- m
> NDA3$NDAfit <- g
> head(NDA3[k==1,], n=9)
```

```
# A tibble: 9 x 11
     ID week    NDA     W   Obs   Fit  UFit  LFit clus   NDAobs NDAfit
  <dbl> <chr> <dbl> <int> <dbl> <dbl> <dbl> <dbl> <fct>   <dbl>  <dbl>
1   108 H0        0     0  1.88  1.86  1.60  2.12 Class2   1.45   1.43
2   108 H1        0     1  1.86  1.93  1.66  2.23 Class2   1.52   1.66
3   108 H2        0     2  2.01  1.89  1.61  2.19 Class2   1.59   1.47
4   108 H3        0     3  1.90  1.84  1.58  2.13 Class2   1.30   1.21
5   108 H4        0     4  1.98  1.88  1.62  2.17 Class2   1.19   1.16
6   108 P1        0     5  2.09  2.12  1.85  2.41 Class2   1.41   1.49
7   108 P2        4     6  2.37  2.52  2.23  2.82 Class2   2.05   2.24
8   108 B1        3     7  3.05  3.02  2.73  3.31 Class2   3.37   3.19
9   108 B2        3     8  3.55  3.51  3.20  3.81 Class2   3.89   3.91
> head(NDA3[k==2,], n=9)
# A tibble: 9 x 11
     ID week    NDA     W   Obs   Fit  UFit  LFit clus   NDAobs NDAfit
  <dbl> <chr> <dbl> <int> <dbl> <dbl> <dbl> <dbl> <fct>   <dbl>  <dbl>
1   103 H0        3     0  1.88  1.86  1.60  2.12 Class1   4.30   4.29
2   103 H1        0     1  1.86  1.93  1.66  2.23 Class1   4.17   4.28
3   103 H2        6     2  2.01  1.89  1.61  2.19 Class1   4.53   4.36
4   103 H3        3     3  1.90  1.84  1.58  2.13 Class1   4.45   4.50
5   103 H4        3     4  1.98  1.88  1.62  2.17 Class1   4.72   4.69
6   103 P1        4     5  2.09  2.12  1.85  2.41 Class1   4.84   4.91
7   103 P2        4     6  2.37  2.52  2.23  2.82 Class1   5.08   5.10
8   103 B1        2     7  3.05  3.02  2.73  3.31 Class1   5.27   5.23
9   103 B2        6     8  3.55  3.51  3.20  3.81 Class1   5.27   5.27
> head(NDA3[k==3,], n=9)
# A tibble: 9 x 11
     ID week    NDA     W   Obs   Fit  UFit  LFit clus   NDAobs NDAfit
  <dbl> <chr> <dbl> <int> <dbl> <dbl> <dbl> <dbl> <fct>   <dbl>  <dbl>
1   106 H0        0     0  1.88  1.86  1.60  2.12 Class3 0.0588 0.0542
2   106 H1        0     1  1.86  1.93  1.66  2.23 Class3 0.0588 0.0719
3   106 H2        0     2  2.01  1.89  1.61  2.19 Class3 0.103  0.103
4   106 H3        0     3  1.90  1.84  1.58  2.13 Class3 0.147  0.148
5   106 H4        0     4  1.98  1.88  1.62  2.17 Class3 0.235  0.192
6   106 P1        0     5  2.09  2.12  1.85  2.41 Class3 0.221  0.220
7   106 P2        0     6  2.37  2.52  2.23  2.82 Class3 0.162  0.279
8   106 B1        0     7  3.05  3.02  2.73  3.31 Class3 0.618  0.510
9   106 B2        5     8  3.55  3.51  3.20  3.81 Class3 1.57   1.59
```

Parameter estimates for the truncated logistic-binomial model within each class are given in Tables 5 to 7 on pages 39–41.

12.2 Get Tables

Assign class variable to original data frame and create tables.

```
> Cluster <- NDA3[NDA3$W==0,"clus"]
> table(Cluster)
Cluster
Class1 Class2 Class3
    64     73     68
```

Table 5: Parameter Estimates for Truncated Logistic-Binomial Model in Class 1

```
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + bs(W, knot = c(4)),
    family = "binomial", data = NDA3, subset = k == 1, id = ID,
    waves = W, corstr = "ar1")

 Coefficients:
                    Estimate Std.err  Wald Pr(>|W|)
(Intercept)           -1.357   0.107 161.2  < 2e-16 ***
bs(W, knot = c(4))1    0.589   0.163  13.1    3e-04 ***
bs(W, knot = c(4))2   -1.438   0.321  20.1  7.5e-06 ***
bs(W, knot = c(4))3    1.249   0.270  21.5  3.6e-06 ***
bs(W, knot = c(4))4    1.592   0.244  42.4  7.4e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.266  0.0273

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha    0.428  0.0375
Number of clusters:   73   Maximum cluster size: 9
```

Table 6: Parameter Estimates for Truncated Logistic-Binomial Model in Class 2

```
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + bs(W, knot = c(4)),
    family = "binomial", data = NDA3, subset = k == 2, id = ID,
    waves = W, corstr = "ar1")

 Coefficients:
                    Estimate Std.err  Wald Pr(>|W|)
(Intercept)           0.4619  0.0957 23.28  1.4e-06 ***
bs(W, knot = c(4))1  -0.0495  0.1996  0.06   0.8043
bs(W, knot = c(4))2   0.1904  0.3302  0.33   0.5642
bs(W, knot = c(4))3   0.6656  0.2804  5.63   0.0176 *
bs(W, knot = c(4))4   0.6490  0.2063  9.89   0.0017 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.279  0.0339

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha    0.627  0.0447
Number of clusters:   64   Maximum cluster size: 9
```

Table 7: Parameter Estimates for Truncated Logistic-Binomial Model in Class 3

```
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + bs(W, knot = c(4)),
    family = "binomial", data = NDA3, subset = k == 3, id = ID,
    waves = W, corstr = "ar1")

 Coefficients:
                    Estimate Std.err  Wald Pr(>|W|)
(Intercept)           -4.853   0.502 93.31  < 2e-16 ***
bs(W, knot = c(4))1    0.282   1.124  0.06    0.802
bs(W, knot = c(4))2    1.781   0.855  4.34    0.037 *
bs(W, knot = c(4))3    1.286   0.885  2.11    0.146
bs(W, knot = c(4))4    3.631   0.555 42.85  5.9e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.241   0.231

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha   0.0974  0.0873
Number of clusters:   68   Maximum cluster size: 9
```

```
> NDA1$Cluster <- Cluster$clus
> table(NDA1$Cluster)
Class1 Class2 Class3
    64     73     68
> round(prop.table(table(NDA1$Cluster)),3)
Class1 Class2 Class3
 0.312  0.356  0.332
> saveRDS(NDA1, file="NDA/NDA1.rds")
> #write.csv(NDA1, file="NDA1.csv")
```

Here I present the observed and fitted means and proportions.

12.3 Plot Results

```
> NDAPlot1 <-ggplot(data=NDA3) + xlab("Weekly Interval") +
  ylab("Number of Days Abstinent per Week") +
  geom_vline(xintercept=4, color="grey60")+
  geom_line(aes(x=W, y=NDAfit, group=clus, color=clus)) +
  geom_point(aes(x=W, y=NDAobs, group=clus, color=clus)) +
  geom_line(aes(x=W, y=Fit), color="grey80") +
  geom_point(aes(x=W, y=Obs), color="grey80") +
  scale_color_manual(values=c("red","blue", "green")) +
  scale_x_continuous(limits=c(0,8), breaks = 0:8, minor_breaks = NULL,
     labels=c(paste0("H",0:4), "P1", "P2", "B1", "B2")) +
  scale_y_continuous(breaks=0:7, limits=c(0,7)) +
  theme_minimal()
> pdf("NDA/NDAPlot1.pdf")
> print(NDAPlot1)
> dev.off()
null device
          1

> NDAPlot2 <- ggplot(data=NDA3, aes(x=W, y=NDAfit, group=clus, color=clus)) +
   xlab("Weekly Interval") + ylab("Number of Days Abstinent per Week") +
   geom_vline(xintercept = 4, color="gray60") +
   geom_line(size=1) +
   facet_wrap(vars(clus), ncol=1) +
   geom_point(aes(x=W, y=NDAobs, group=clus, color=clus)) +
   geom_line(data=NDA3, aes(x=W, y=NDA, group=ID, color=clus), alpha=.1) +
  scale_color_manual(values=c("red","blue", "green")) +
  scale_x_continuous(limits=c(0,8), breaks = 0:8, minor_breaks = NULL,
     labels=c(paste0("H",0:4), "P1", "P2", "B1", "B2")) +
  scale_y_continuous(breaks=0:7, limits=c(0,7)) +
  theme_gray()
> pdf("NDA/NDAPlot2.pdf")
> print(NDAPlot2)
> dev.off()
null device
          1
```

The fitted and observed means for each cluster are displayed in Figure 10 on the following page. The fitted and observe data within each class are displayed in Figure 11 on page 44.

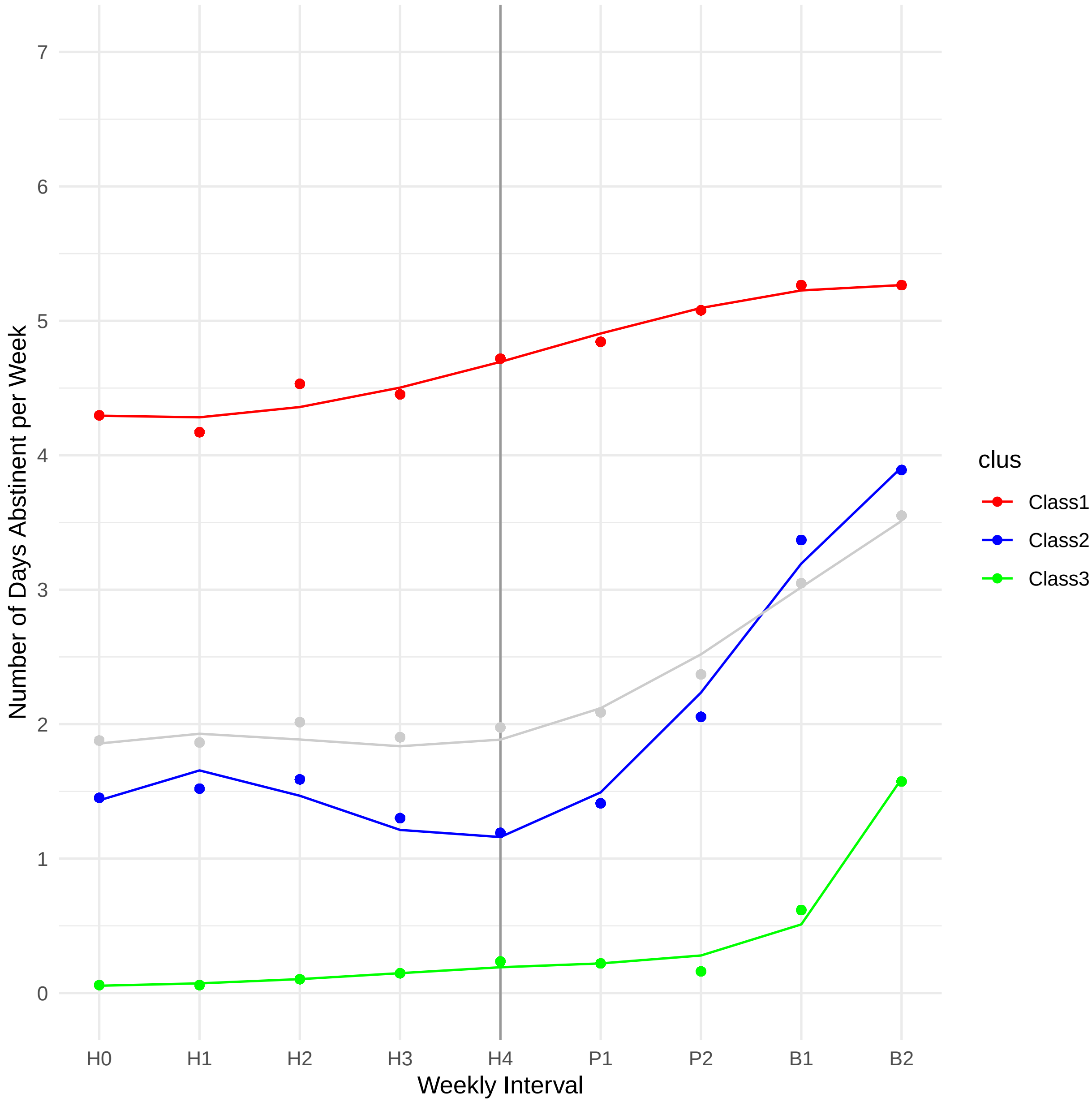

Figure 10: Observed and fitted pretreatment means for each class. The gray points and line are the observed and fitted means for the entire sample.

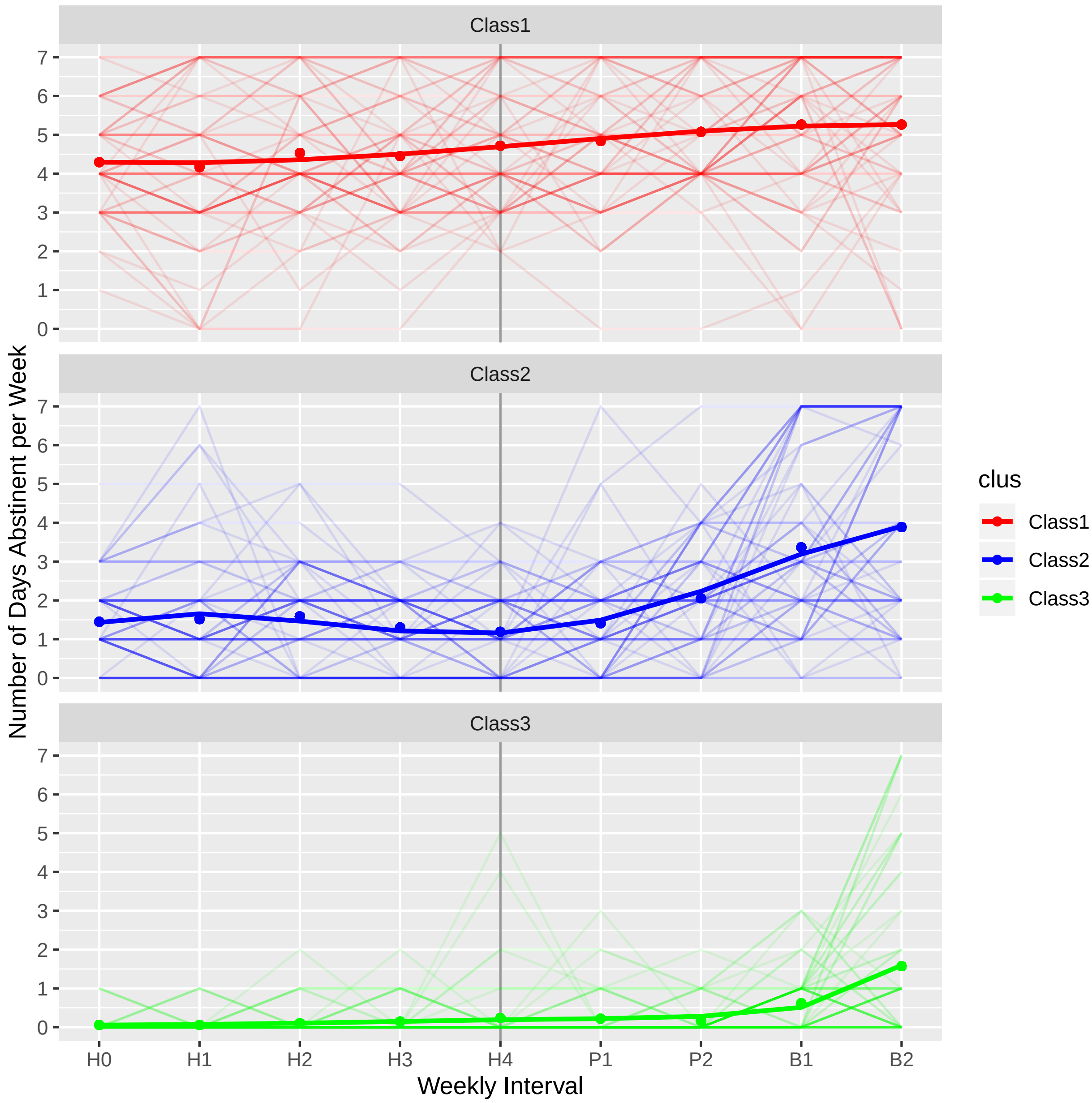

Figure 11: Observed and fitted pretreatment means within each class. The lines are the actual data.

13 Analysis of Treatment Effects

13.1 Dataframe with Pretreatment Class

Create a dataframe with pretreatment classification hard assignment.
Assign class variable to original data frame and create tables.

```
> NDA2$K <- 0
> UID <- unique(NDA2$ID)
> N <- length(UID)
> KK <- table(NDA2$ID)
> for (i in 1:N){
   NDA2$K[which(NDA2$ID==UID[i])] <- rep(tapply(k,NDA3$ID, max)[i], KK[i])
 }
> NDA2$Cluster <- factor(NDA2$K, levels=rev(ord), labels=paste0("Class", 1:3))
> n <- nrow(NDA2)
> k <- NDA2$K
> MM1 <- geeglm(cbind(NDA, 7-NDA)~0 + factor(W), data=NDA2, subset=(Cluster=="Class1"),
                family="binomial", id=ID, waves=W)
> MM2 <- geeglm(cbind(NDA, 7-NDA)~0 + factor(W), data=NDA2, subset=(Cluster=="Class2"),
                family="binomial", id=ID, waves=W)
> MM3 <- geeglm(cbind(NDA, 7-NDA)~0 + factor(W), data=NDA2, subset=(Cluster=="Class3"),
                family="binomial", id=ID, waves=W)
> HH1 <- geeglm(cbind(NDA, 7-NDA)~1+ bs(W, knots=c(4, 8)), data=NDA2,
                 subset=(Cluster=="Class1"), family="binomial", id=ID,
                waves=W, corstr = "ar1")
> #summary(HH1)
> HH2 <-  geeglm(cbind(NDA, 7-NDA)~1+ bs(W, knots=c(4, 8)), data=NDA2,
                 subset=(Cluster=="Class2"), family="binomial", id=ID,
                 waves=W, corstr = "ar1")
> #summary(HH2)
> HH3 <- geeglm(cbind(NDA, 7-NDA)~1+ bs(W, knots=c(4, 8)), data=NDA2,
                subset=(Cluster=="Class3"), family="binomial", id=ID,
                waves=W, corstr = "ar1")
> #summary(HH3)
>
> #Alternative representations of above spline
> HH1_alt <- geeglm(cbind(NDA, 7-NDA) ~ 1 + W + I(W^2) + I(W^3) + I((W>4)*(W-4)^3) +
              I((W>8)*(W-8)^3), id=ID, data=NDA2,
              family="binomial", waves=W, corstr = "ar1", subset=(Cluster=="Class1"))
> HH2_alt <- geeglm(cbind(NDA, 7-NDA) ~ 1 + W + I(W^2) + I(W^3) + I((W>4)*(W-4)^3) +
              I((W>8)*(W-8)^3), id=ID, data=NDA2,
              family="binomial", waves=W, corstr = "ar1", subset=(Cluster=="Class2"))
> HH3_alt <- geeglm(cbind(NDA, 7-NDA) ~ 1 + W + I(W^2) + I(W^3) + I((W>4)*(W-4)^3) +
              I((W>8)*(W-8)^3), id=ID, data=NDA2,
              family="binomial", waves=W, corstr = "ar1", subset=(Cluster=="Class3"))
>
> #summary(HH1_alt)
> #summary(HH2_alt)
> #summary(HH3_alt)
>
```

Parameter estimates for the full logistic-binomial model within each class are given in Tables 8 to 10 on

Table 8: Parameter Estimates for Full Logistic-Binomial Model in Class 1

```
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + bs(W, knots = c(4,
    8)), family = "binomial", data = NDA2, subset = (Cluster ==
    "Class1"), id = ID, waves = W, corstr = "ar1")

 Coefficients:
                        Estimate Std.err  Wald Pr(>|W|)
(Intercept)               0.4640  0.0958 23.48  1.3e-06 ***
bs(W, knots = c(4, 8))1  -0.0611  0.1755  0.12    0.728
bs(W, knots = c(4, 8))2   0.2087  0.2268  0.85    0.357
bs(W, knots = c(4, 8))3   1.2148  0.3698 10.79    0.001 **
bs(W, knots = c(4, 8))4   0.7136  0.3279  4.74    0.030 *
bs(W, knots = c(4, 8))5   1.1197  0.2326 23.17  1.5e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.375  0.0538

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha    0.856  0.0271
Number of clusters:   64   Maximum cluster size: 20
```

pages 46–48.

Parameter estimates for the naive full logistic-binomial model within each class are given in Tables 11 to 13 on pages 49–51.

```
> head(NDA2[NDA2$Cluster=="Class1",], n=20)
# A tibble: 20 x 10
      ID week    NDA     W   Obs   Fit  UFit  LFit     K Cluster
   <dbl> <chr> <dbl> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <fct>
 1   103 H0        3     0  1.88  1.86  1.60  2.12     2 Class1
 2   103 H1        0     1  1.86  1.93  1.66  2.23     2 Class1
 3   103 H2        6     2  2.01  1.89  1.61  2.19     2 Class1
 4   103 H3        3     3  1.90  1.84  1.58  2.13     2 Class1
 5   103 H4        3     4  1.98  1.88  1.62  2.17     2 Class1
 6   103 P1        4     5  2.09  2.12  1.85  2.41     2 Class1
 7   103 P2        4     6  2.37  2.52  2.23  2.82     2 Class1
 8   103 B1        2     7  3.05  3.02  2.73  3.31     2 Class1
 9   103 B2        6     8  3.55  3.51  3.20  3.81     2 Class1
10   103 S01       5     9  3.98  3.90  3.58  4.21     2 Class1
11   103 S02       5    10  4.14  4.18  3.86  4.49     2 Class1
```

Table 9: Parameter Estimates for Full Logistic-Binomial Model in Class 2

```
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + bs(W, knots = c(4,
    8)), family = "binomial", data = NDA2, subset = (Cluster ==
    "Class2"), id = ID, waves = W, corstr = "ar1")

 Coefficients:
                        Estimate Std.err   Wald Pr(>|W|)
(Intercept)               -1.333   0.102 172.37  < 2e-16 ***
bs(W, knots = c(4, 8))1    0.413   0.137   9.05   0.0026 **
bs(W, knots = c(4, 8))2   -0.993   0.218  20.75  5.2e-06 ***
bs(W, knots = c(4, 8))3    3.731   0.413  81.44  < 2e-16 ***
bs(W, knots = c(4, 8))4    1.414   0.244  33.45  7.3e-09 ***
bs(W, knots = c(4, 8))5    2.366   0.240  96.81  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.389  0.0366

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha     0.81  0.0217
Number of clusters:   73   Maximum cluster size: 20
```

Table 10: Parameter Estimates for Full Logistic-Binomial Model in Class 3

```
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + bs(W, knots = c(4,
    8)), family = "binomial", data = NDA2, subset = (Cluster ==
    "Class3"), id = ID, waves = W, corstr = "ar1")

 Coefficients:
                        Estimate Std.err   Wald Pr(>|W|)
(Intercept)               -4.925   0.450 119.98   <2e-16 ***
bs(W, knots = c(4, 8))1    0.728   0.891   0.67     0.41
bs(W, knots = c(4, 8))2    0.177   0.485   0.13     0.72
bs(W, knots = c(4, 8))3    5.960   0.612  94.97   <2e-16 ***
bs(W, knots = c(4, 8))4    4.719   0.563  70.17   <2e-16 ***
bs(W, knots = c(4, 8))5    5.021   0.487 106.31   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.414   0.152

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha    0.781  0.0798
Number of clusters:   68   Maximum cluster size: 20
```

Table 11: Parameter Estimates for Naive Full Logistic-Binomial Model in Class 1

```
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + W + I(W^2) + I(W^3) +
    I((W > 4) * (W - 4)^3) + I((W > 8) * (W - 8)^3), family = "binomial",
    data = NDA2, subset = (Cluster == "Class1"), id = ID, waves = W,
    corstr = "ar1")

 Coefficients:
                        Estimate   Std.err  Wald Pr(>|W|)
(Intercept)             0.464009  0.095759 23.48  1.3e-06 ***
W                      -0.045821  0.131602  0.12     0.73
I(W^2)                  0.036748  0.066774  0.30     0.58
I(W^3)                 -0.002462  0.007581  0.11     0.75
I((W > 4) * (W - 4)^3)  0.000421  0.010275  0.00     0.97
I((W > 8) * (W - 8)^3)  0.003145  0.003624  0.75     0.39
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.375  0.0538

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha    0.856  0.0271
Number of clusters:   64   Maximum cluster size: 20
```

Table 12: Parameter Estimates for Naive Full Logistic-Binomial Model in Class 2

```
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + W + I(W^2) + I(W^3) +
    I((W > 4) * (W - 4)^3) + I((W > 8) * (W - 8)^3), family = "binomial",
    data = NDA2, subset = (Cluster == "Class2"), id = ID, waves = W,
    corstr = "ar1")

 Coefficients:
                       Estimate  Std.err   Wald Pr(>|W|)
(Intercept)            -1.33268  0.10151 172.37  < 2e-16 ***
W                       0.30976  0.10299   9.05  0.00263 **
I(W^2)                 -0.20923  0.05588  14.02  0.00018 ***
I(W^3)                  0.03070  0.00653  22.10  2.6e-06 ***
I((W > 4) * (W - 4)^3) -0.05067  0.00913  30.82  2.8e-08 ***
I((W > 8) * (W - 8)^3)  0.02441  0.00344  50.39  1.3e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.389  0.0366

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha     0.81  0.0217
Number of clusters:   73   Maximum cluster size: 20
```

Table 13: Parameter Estimates for Naive Full Logistic-Binomial Model in Class 3

```
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + W + I(W^2) + I(W^3) +
    I((W > 4) * (W - 4)^3) + I((W > 8) * (W - 8)^3), family = "binomial",
    data = NDA2, subset = (Cluster == "Class3"), id = ID, waves = W,
    corstr = "ar1")

 Coefficients:
                       Estimate  Std.err   Wald Pr(>|W|)
(Intercept)            -4.92475  0.44961 119.98  < 2e-16 ***
W                       0.54563  0.66858   0.67     0.41
I(W^2)                 -0.18804  0.22943   0.67     0.41
I(W^3)                  0.02733  0.02225   1.51     0.22
I((W > 4) * (W - 4)^3) -0.04545  0.02594   3.07     0.08 .
I((W > 8) * (W - 8)^3)  0.02137  0.00487  19.23  1.2e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.414   0.152

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha    0.781  0.0798
Number of clusters:   68   Maximum cluster size: 20
```

```
12   103 S03       5    11  4.44  4.37  4.06  4.67     2 Class1
13   103 S04       5    12  4.51  4.49  4.19  4.79     2 Class1
14   103 S05       3    13  4.61  4.57  4.26  4.86     2 Class1
15   103 S06       5    14  4.62  4.60  4.29  4.90     2 Class1
16   103 S07       6    15  4.63  4.63  4.31  4.93     2 Class1
17   103 S08       5    16  4.61  4.65  4.31  4.96     2 Class1
18   103 S09       4    17  4.74  4.69  4.35  5.00     2 Class1
19   103 S10       6    18  4.74  4.76  4.43  5.06     2 Class1
20   103 S11       6    19  4.87  4.87  4.54  5.17     2 Class1
> head(NDA2[NDA2$Cluster=="Class2",], n=20)
# A tibble: 20 x 10
      ID week    NDA     W   Obs   Fit  UFit  LFit     K Cluster
   <dbl> <chr> <dbl> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <fct>
 1   108 H0        0     0  1.88  1.86  1.60  2.12     1 Class2
 2   108 H1        0     1  1.86  1.93  1.66  2.23     1 Class2
 3   108 H2        0     2  2.01  1.89  1.61  2.19     1 Class2
 4   108 H3        0     3  1.90  1.84  1.58  2.13     1 Class2
 5   108 H4        0     4  1.98  1.88  1.62  2.17     1 Class2
 6   108 P1        0     5  2.09  2.12  1.85  2.41     1 Class2
 7   108 P2        4     6  2.37  2.52  2.23  2.82     1 Class2
 8   108 B1        3     7  3.05  3.02  2.73  3.31     1 Class2
 9   108 B2        3     8  3.55  3.51  3.20  3.81     1 Class2
10   108 S01       3     9  3.98  3.90  3.58  4.21     1 Class2
11   108 S02       4    10  4.14  4.18  3.86  4.49     1 Class2
12   108 S03       2    11  4.44  4.37  4.06  4.67     1 Class2
13   108 S04       3    12  4.51  4.49  4.19  4.79     1 Class2
14   108 S05       5    13  4.61  4.57  4.26  4.86     1 Class2
15   108 S06       5    14  4.62  4.60  4.29  4.90     1 Class2
16   108 S07       6    15  4.63  4.63  4.31  4.93     1 Class2
17   108 S08       5    16  4.61  4.65  4.31  4.96     1 Class2
18   108 S09       7    17  4.74  4.69  4.35  5.00     1 Class2
19   108 S10       5    18  4.74  4.76  4.43  5.06     1 Class2
20   108 S11       7    19  4.87  4.87  4.54  5.17     1 Class2
> head(NDA2[NDA2$Cluster=="Class3",], n=20)
# A tibble: 20 x 10
      ID week    NDA     W   Obs   Fit  UFit  LFit     K Cluster
   <dbl> <chr> <dbl> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <fct>
 1   106 H0        0     0  1.88  1.86  1.60  2.12     3 Class3
 2   106 H1        0     1  1.86  1.93  1.66  2.23     3 Class3
 3   106 H2        0     2  2.01  1.89  1.61  2.19     3 Class3
 4   106 H3        0     3  1.90  1.84  1.58  2.13     3 Class3
 5   106 H4        0     4  1.98  1.88  1.62  2.17     3 Class3
 6   106 P1        0     5  2.09  2.12  1.85  2.41     3 Class3
 7   106 P2        0     6  2.37  2.52  2.23  2.82     3 Class3
 8   106 B1        0     7  3.05  3.02  2.73  3.31     3 Class3
 9   106 B2        5     8  3.55  3.51  3.20  3.81     3 Class3
10   106 S01       5     9  3.98  3.90  3.58  4.21     3 Class3
11   106 S02       5    10  4.14  4.18  3.86  4.49     3 Class3
12   106 S03       7    11  4.44  4.37  4.06  4.67     3 Class3
13   106 S04       5    12  4.51  4.49  4.19  4.79     3 Class3
14   106 S05       7    13  4.61  4.57  4.26  4.86     3 Class3
```

```
15   106 S06       7    14  4.62  4.60  4.29  4.90     3 Class3
16   106 S07       7    15  4.63  4.63  4.31  4.93     3 Class3
17   106 S08       6    16  4.61  4.65  4.31  4.96     3 Class3
18   106 S09       7    17  4.74  4.69  4.35  5.00     3 Class3
19   106 S10       7    18  4.74  4.76  4.43  5.06     3 Class3
20   106 S11       5    19  4.87  4.87  4.54  5.17     3 Class3
```

13.2 Create Data Frame for Fitted and Observed Means

```
> NDA2$NDAobs <- 0
> NDA2$NDAfit <- 0
> NDA2$NDAlb <- 0
> NDA2$NDAub <- 0
> ok <- NDA2$Cluster=="Class1"
> NDA2$NDAobs[ok] <- 7*predict(MM1, type="response")
> NDA2$NDAfit[ok] <- 7*predict(HH1, type="response")
> b <- coef(HH1)
> names(b) <- NULL
> Sigma <- HH1$geese$vbeta
> X <- cbind(1, HH1$model$`bs(W, knots = c(4, 8))`)
> #X <- cbind(1, HH1$model$`bs(W`)
> K <- 1000
> R <- mvrnorm(K, mu=b, Sigma=Sigma)
> U <- 7*plogis(X %*% (t(R)))
> u <- apply(U,1,quantile, prob=.025)
> v <- apply(U,1,quantile, prob=.975)
> NDA2$NDAlb[ok] <- u
> NDA2$NDAub[ok] <- v
> ok <- NDA2$Cluster=="Class2"
> NDA2$NDAobs[ok] <- 7*predict(MM2, type="response")
> NDA2$NDAfit[ok] <- 7*predict(HH2, type="response")
> b <- coef(HH2)
> names(b) <- NULL
> Sigma <- HH2$geese$vbeta
> X <- cbind(1, HH2$model$`bs(W, knots = c(4, 8))`)
> K <- 1000
> R <- mvrnorm(K, mu=b, Sigma=Sigma)
> U <- 7*plogis(X %*% (t(R)))
> u <- apply(U,1,quantile, prob=.025)
> v <- apply(U,1,quantile, prob=.975)
> NDA2$NDAlb[ok] <- u
> NDA2$NDAub[ok] <- v
> ok <- NDA2$Cluster=="Class3"
> NDA2$NDAobs[ok] <- 7*predict(MM3, type="response")
> NDA2$NDAfit[ok] <- 7*predict(HH3, type="response")
> b <- coef(HH3)
> names(b) <- NULL
> Sigma <- HH3$geese$vbeta
> X <- cbind(1, HH3$model$`bs(W, knots = c(4, 8))`)
> K <- 1000
```

```
> R <- mvrnorm(K, mu=b, Sigma=Sigma)
> U <- 7*plogis(X %*% (t(R)))
> u <- apply(U,1,quantile, prob=.025)
> v <- apply(U,1,quantile, prob=.975)
> NDA2$NDAlb[ok] <- u
> NDA2$NDAub[ok] <- v
> head(as.data.frame(NDA2[NDA2$Cluster=="Class1",]), n=20)
    ID week NDA  W  Obs  Fit UFit LFit K Cluster NDAobs NDAfit NDAlb
1  103   H0   3  0 1.88 1.86 1.60 2.12 2  Class1   4.30   4.30  3.99
2  103   H1   0  1 1.86 1.93 1.66 2.23 2  Class1   4.17   4.28  3.81
3  103   H2   6  2 2.01 1.89 1.61 2.19 2  Class1   4.53   4.36  3.93
4  103   H3   3  3 1.90 1.84 1.58 2.13 2  Class1   4.45   4.50  4.18
5  103   H4   3  4 1.98 1.88 1.62 2.17 2  Class1   4.72   4.69  4.37
6  103   P1   4  5 2.09 2.12 1.85 2.41 2  Class1   4.84   4.90  4.57
7  103   P2   4  6 2.37 2.52 2.23 2.82 2  Class1   5.08   5.09  4.78
8  103   B1   2  7 3.05 3.02 2.73 3.31 2  Class1   5.27   5.27  4.94
9  103   B2   6  8 3.55 3.51 3.20 3.81 2  Class1   5.27   5.40  5.03
10 103  S01   5  9 3.98 3.90 3.58 4.21 2  Class1   5.44   5.49  5.08
11 103  S02   5 10 4.14 4.18 3.86 4.49 2  Class1   5.71   5.54  5.12
12 103  S03   5 11 4.44 4.37 4.06 4.67 2  Class1   5.61   5.57  5.13
13 103  S04   5 12 4.51 4.49 4.19 4.79 2  Class1   5.69   5.58  5.14
14 103  S05   3 13 4.61 4.57 4.26 4.86 2  Class1   5.67   5.58  5.11
15 103  S06   5 14 4.62 4.60 4.29 4.90 2  Class1   5.60   5.58  5.08
16 103  S07   6 15 4.63 4.63 4.31 4.93 2  Class1   5.58   5.59  5.05
17 103  S08   5 16 4.61 4.65 4.31 4.96 2  Class1   5.70   5.61  5.04
18 103  S09   4 17 4.74 4.69 4.35 5.00 2  Class1   5.62   5.65  5.10
19 103  S10   6 18 4.74 4.76 4.43 5.06 2  Class1   5.53   5.71  5.20
20 103  S11   6 19 4.87 4.87 4.54 5.17 2  Class1   5.82   5.81  5.29
   NDAub
1   4.63
2   4.73
3   4.78
4   4.82
5   4.99
6   5.21
7   5.39
8   5.56
9   5.72
10  5.83
11  5.89
12  5.91
13  5.94
14  5.95
15  5.96
16  5.99
17  6.01
18  6.04
19  6.10
20  6.17
> head(as.data.frame(NDA2[NDA2$Cluster=="Class2",]), n=20)
    ID week NDA  W  Obs  Fit UFit LFit K Cluster NDAobs NDAfit NDAlb
```

```
1  108   H0   0  0 1.88 1.86 1.60 2.12 1  Class2   1.45   1.46  1.24
2  108   H1   0  1 1.86 1.93 1.66 2.23 1  Class2   1.52   1.62  1.32
3  108   H2   0  2 2.01 1.89 1.61 2.19 1  Class2   1.59   1.49  1.24
4  108   H3   0  3 1.90 1.84 1.58 2.13 1  Class2   1.30   1.32  1.13
5  108   H4   0  4 1.98 1.88 1.62 2.17 1  Class2   1.19   1.30  1.11
6  108   P1   0  5 2.09 2.12 1.85 2.41 1  Class2   1.41   1.59  1.36
7  108   P2   4  6 2.37 2.52 2.23 2.82 1  Class2   2.05   2.20  1.94
8  108   B1   3  7 3.05 3.02 2.73 3.31 1  Class2   3.37   3.05  2.71
9  108   B2   3  8 3.55 3.51 3.20 3.81 1  Class2   3.89   3.90  3.45
10 108  S01   3  9 3.98 3.90 3.58 4.21 1  Class2   4.65   4.48  3.97
11 108  S02   4 10 4.14 4.18 3.86 4.49 1  Class2   4.59   4.81  4.28
12 108  S03   2 11 4.44 4.37 4.06 4.67 1  Class2   4.65   4.97  4.43
13 108  S04   3 12 4.51 4.49 4.19 4.79 1  Class2   4.81   5.00  4.47
14 108  S05   5 13 4.61 4.57 4.26 4.86 1  Class2   4.78   4.95  4.44
15 108  S06   5 14 4.62 4.60 4.29 4.90 1  Class2   4.78   4.87  4.35
16 108  S07   6 15 4.63 4.63 4.31 4.93 1  Class2   4.70   4.77  4.24
17 108  S08   5 16 4.61 4.65 4.31 4.96 1  Class2   4.74   4.72  4.18
18 108  S09   7 17 4.74 4.69 4.35 5.00 1  Class2   4.91   4.74  4.18
19 108  S10   5 18 4.74 4.76 4.43 5.06 1  Class2   4.96   4.88  4.33
20 108  S11   7 19 4.87 4.87 4.54 5.17 1  Class2   5.10   5.16  4.59
   NDAub
1   1.70
2   1.93
3   1.75
4   1.53
5   1.52
6   1.83
7   2.46
8   3.39
9   4.32
10  4.96
11  5.29
12  5.42
13  5.44
14  5.39
15  5.31
16  5.24
17  5.20
18  5.23
19  5.36
20  5.63
> head(as.data.frame(NDA2[NDA2$Cluster=="Class3",]), n=20)
    ID week NDA  W  Obs  Fit UFit LFit K Cluster NDAobs NDAfit  NDAlb
1  106   H0   0  0 1.88 1.86 1.60 2.12 3  Class3 0.0588 0.0505 0.0210
2  106   H1   0  1 1.86 1.93 1.66 2.23 3  Class3 0.0588 0.0739 0.0433
3  106   H2   0  2 2.01 1.89 1.61 2.19 3  Class3 0.1029 0.0877 0.0464
4  106   H3   0  3 1.90 1.84 1.58 2.13 3  Class3 0.1471 0.0992 0.0577
5  106   H4   0  4 1.98 1.88 1.62 2.17 3  Class3 0.2353 0.1257 0.0841
6  106   P1   0  5 2.09 2.12 1.85 2.41 3  Class3 0.2206 0.2000 0.1456
7  106   P2   0  6 2.37 2.52 2.23 2.82 3  Class3 0.1618 0.3721 0.2841
8  106   B1   0  7 3.05 3.02 2.73 3.31 3  Class3 0.6176 0.7167 0.5759
```

```
9  106   B2   5  8 3.55 3.51 3.20 3.81 3  Class3 1.5735 1.2684 1.0220
10 106  S01   5  9 3.98 3.90 3.58 4.21 3  Class3 1.9254 1.9118 1.5306
11 106  S02   5 10 4.14 4.18 3.86 4.49 3  Class3 2.1692 2.5046 2.0161
12 106  S03   7 11 4.44 4.37 4.06 4.67 3  Class3 3.1077 2.9675 2.4373
13 106  S04   5 12 4.51 4.49 4.19 4.79 3  Class3 3.0317 3.2821 2.7129
14 106  S05   7 13 4.61 4.57 4.26 4.86 3  Class3 3.3810 3.4661 2.8802
15 106  S06   7 14 4.62 4.60 4.29 4.90 3  Class3 3.5000 3.5504 2.9511
16 106  S07   7 15 4.63 4.63 4.31 4.93 3  Class3 3.6290 3.5689 2.9685
17 106  S08   6 16 4.61 4.65 4.31 4.96 3  Class3 3.3770 3.5557 2.9117
18 106  S09   7 17 4.74 4.69 4.35 5.00 3  Class3 3.6667 3.5452 2.9091
19 106  S10   7 18 4.74 4.76 4.43 5.06 3  Class3 3.6833 3.5715 2.9382
20 106  S11   5 19 4.87 4.87 4.54 5.17 3  Class3 3.6500 3.6686 3.0218
   NDAub
1  0.121
2  0.131
3  0.175
4  0.173
5  0.184
6  0.272
7  0.474
8  0.889
9  1.569
10 2.361
11 3.050
12 3.527
13 3.817
14 4.004
15 4.117
16 4.146
17 4.154
18 4.148
19 4.186
20 4.326
```

Here I present the observed and fitted means and proportions. First, the observed means. And then the observed proportions. Now the fitted means. And the fitted proportions.

```
> # cbind(week=1:19, round(tapply(NDA5$NDAObs, list(NDA5$W, NDA5$Cluster), mean),2))
> # cbind(week=1:19, round(tapply(NDA5$NDAObs/7, list(NDA5$W, NDA5$Cluster), mean),2))
> # cbind(week=1:19, round(tapply(NDA5$NDAfit, list(NDA5$W, NDA5$Cluster), mean),2))
> # cbind(week=1:19,round(tapply(NDA5$NDAfit/7, list(NDA5$W, NDA5$Cluster), mean),2))
```

13.3 Plot Results

Plot the fitted and observed means for each cluster.

```
> NDAPlot3 <-ggplot(data=NDA2) +
   xlab("Week Interval") + ylab("Number of Days Abstinent per Week") +
   geom_vline(xintercept = c(4,8), color="gray60") +
   geom_ribbon(aes(x=W, ymin=LFit, ymax=UFit), fill="gray40", alpha=.09) +
   geom_ribbon(aes(x=W, ymin=NDAlb, ymax=NDAub, group=Cluster, fill=Cluster), alpha=.1) +
   geom_line( aes(x=W, y=Fit), color="gray80") +
```

```
  geom_line(  aes(x=W, y=NDAfit,               group=Cluster, color=Cluster)) +
  geom_point( aes(x=W, y=Obs), color="gray80") +
  geom_point( aes(x=W, y=NDAobs,               group=Cluster, color=Cluster)) +
  scale_color_manual(values=c("red","blue", "green"), name="Class") +
  scale_fill_manual( values=c("red","blue", "green"), name="Class") +
  scale_x_continuous(limits=c(0,19), breaks = 0:19, minor_breaks = NULL,
        labels=c(paste0("H",0:4), "P1", "P2", "B1", "B2", paste0("S0", 1:9), "S10", "S11")) +
  scale_y_continuous(breaks=0:7, limits=c(0,7),
                     sec.axis=sec_axis(~./7, name="Percent Days Abstinent")) +
  guides(fill=guide_legend(title=NULL)) +
  guides(color=guide_legend(title=NULL)) +
  theme_minimal() +
  theme(legend.position=c(.8,.2), legend.text=element_text(lineheight = 2))
> pdf("NDA/NDAPlot3.pdf")
> print(NDAPlot3)
> dev.off()
null device
          1
```

Plot the results within cluster

```
> NDAPlot4 <- ggplot(data=NDA2, aes(x=W, y=NDAfit, group=Cluster, color=Cluster)) +
  xlab("Weekly Interval") + ylab("Number of Days Abstinent per Week") +
  geom_vline(xintercept = c(4,8), color="gray60") +
  geom_line(size=1) +
  facet_wrap(vars(Cluster), ncol=1) +
  geom_point(aes(x=W, y=NDAobs, group=Cluster, color=Cluster)) +
  geom_line(data=NDA2, aes(x=W, y=NDA, group=ID, color=Cluster), alpha=.1) +
 scale_color_manual(values=c("red","blue", "green")) +
 scale_x_continuous(limits=c(0,19), breaks = 0:19, minor_breaks = NULL,
    labels=c(paste0("H",0:4), "P1", "P2", "B1", "B2", paste0("S0", 1:9), "S10", "S11")) +
 scale_y_continuous(breaks=0:7, limits=c(0,7)) +
 theme_gray()
> pdf("NDA/NDAPlot4.pdf")
> print(NDAPlot4)
> dev.off()
null device
          1
```

The fitted and observed means for each cluster are displayed in Figure 12 on the following page.
The fitted and observed data within each cluster are displayed in Figure 13 on page 59.

14 Follow-up Analyses

Here I conduct the analyses of the followup data at 3 and 6 months.

14.1 Get Follow-up Data

```
> NDA4 <- NDA1[, c(1, 22:24)]
> names(NDA4) <- c("ID", "F3", "F6", "Cluster")
```

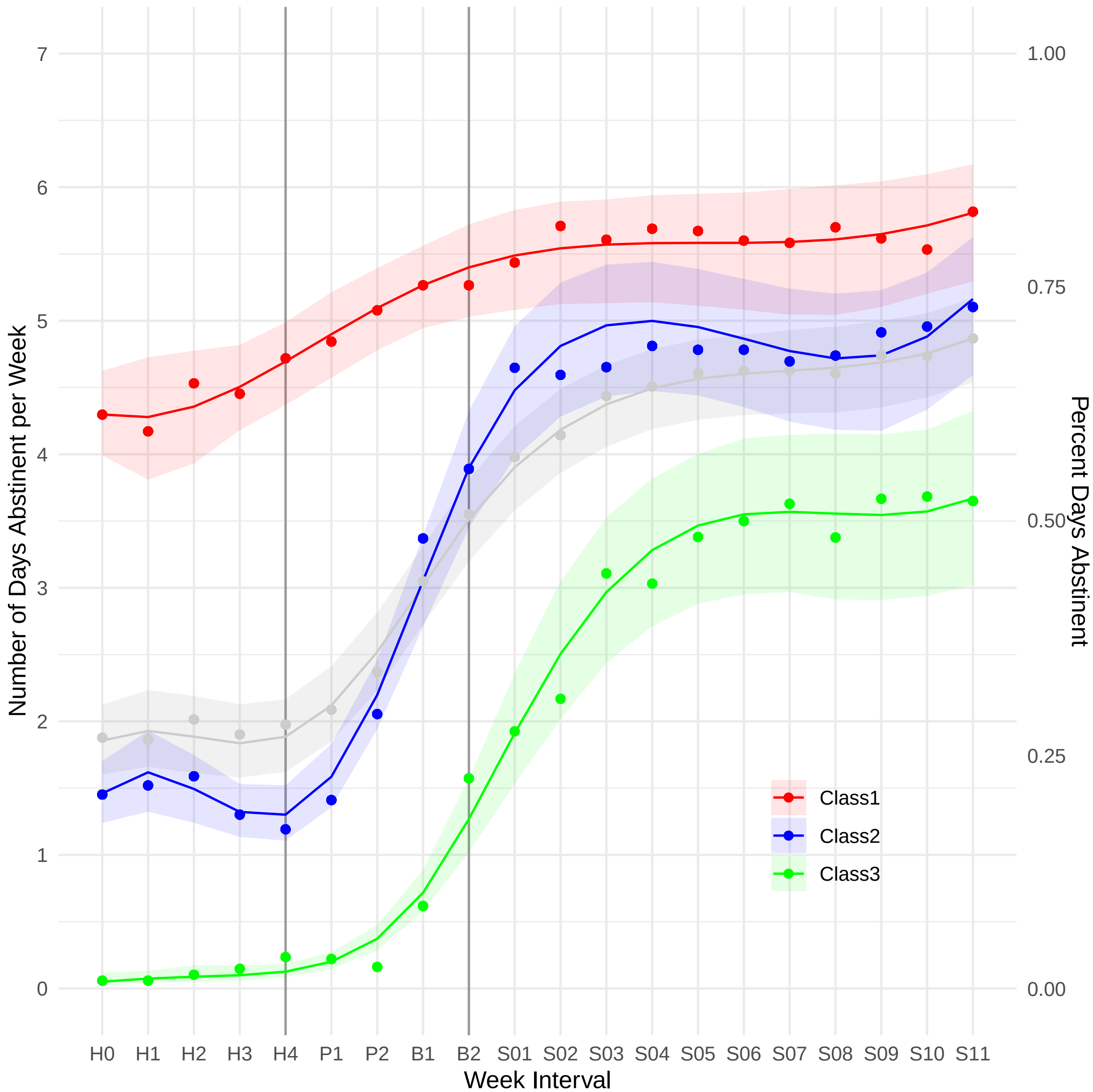

Figure 12: Observed (points) and fitted (lines) means for pretreatment and treatment for each class with corresponding 95% probability ribbons. The gray points, line, and ribbon show the results for the entire sample.

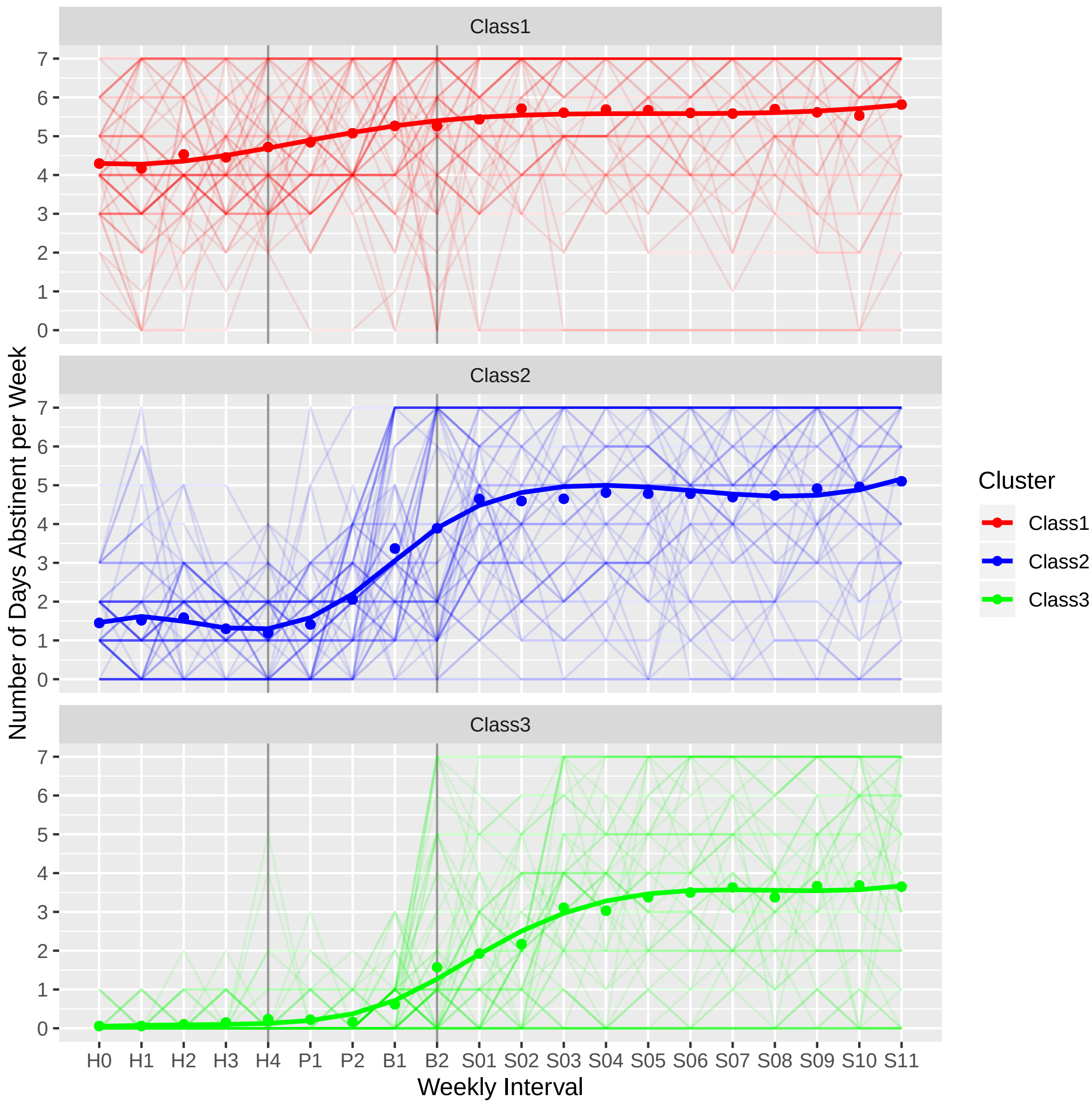

Figure 13: Observed (points) and fitted (lines) means for pretreatment and treatment within each class. The ghosted lines are the actual data.

14.2 Missing Data

```
> N <- nrow(NDA4)
> table(apply(is.na(NDA4),1,sum))
  0   1   2
170  13  22
> round(table(apply(is.na(NDA4),1,sum))/N,3)
    0     1     2
0.829 0.063 0.107
```

14.3 Data Summaries

The first data summary is

```
> options(width=72)
> N <- nrow(NDA4)
> summary(NDA4)
       ID             F3              F6            Cluster
 Min.   :103   Min.   :0.00   Min.   :0.00   Class1:64
 1st Qu.:194   1st Qu.:0.44   1st Qu.:0.42   Class2:73
 Median :290   Median :0.83   Median :0.74   Class3:68
 Mean   :289   Mean   :0.67   Mean   :0.66
 3rd Qu.:384   3rd Qu.:0.99   3rd Qu.:1.00
 Max.   :478   Max.   :1.00   Max.   :1.00
               NA's   :26     NA's   :31
```

The second summary is

```
> options(width=72)
> describe((NDA4))
(NDA4)

 4  Variables      205  Observations
------------------------------------------------------------------------
ID
       n  missing distinct     Info     Mean      Gmd      .05      .10
     205        0      205        1    288.7    125.5    129.4    143.4
     .25      .50      .75      .90      .95
   194.0    290.0    384.0    437.6    455.6

lowest : 103 106 108 110 111, highest: 472 474 476 477 478
------------------------------------------------------------------------
F3
       n  missing distinct     Info     Mean      Gmd      .05      .10
     179       26       57    0.991   0.6741   0.3808    0.000    0.008
     .25      .50      .75      .90      .95
   0.445    0.830    0.990    1.000    1.000

lowest : 0.00 0.01 0.03 0.04 0.08, highest: 0.96 0.97 0.98 0.99 1.00
------------------------------------------------------------------------
F6
```

```
       n  missing distinct     Info     Mean      Gmd      .05      .10
     174       31       65    0.983   0.6646   0.3849   0.0000   0.0130
     .25      .50      .75      .90      .95
  0.4225   0.7350   0.9975   1.0000   1.0000

lowest : 0.00 0.01 0.02 0.04 0.05, highest: 0.96 0.97 0.98 0.99 1.00
------------------------------------------------------------------------
Cluster
       n  missing distinct
     205        0        3

Value      Class1 Class2 Class3
Frequency      64     73     68
Proportion  0.312  0.356  0.332
------------------------------------------------------------------------
```

14.4 Data Preparation

The dataframe NDA4 was created by converting NDA4 and converted from wide to long format. The variable NDA for the number of days abstinent as created from the orginal proportion data. The week number was appended. The data frame NDA4 was reduced to all non-missing values.

```
> NDA4 <- gather(NDA4, week, NDA, F3:F6)
> NDA4 <- arrange(NDA4, ID, factor(week, levels=c("F3", "F6")))
> NDA4$NDA <- round(NDA4$NDA*7)
> NDA4$W <- c(21,22)
> head(NDA4, n=12)
# A tibble: 12 x 5
      ID Cluster week    NDA     W
   <dbl> <fct>   <chr> <dbl> <dbl>
 1   103 Class1  F3        5    21
 2   103 Class1  F6        5    22
 3   106 Class3  F3       NA    21
 4   106 Class3  F6       NA    22
 5   108 Class2  F3        5    21
 6   108 Class2  F6        5    22
 7   110 Class2  F3        7    21
 8   110 Class2  F6        7    22
 9   111 Class1  F3        7    21
10   111 Class1  F6        6    22
11   113 Class1  F3        7    21
12   113 Class1  F6       NA    22
> tail(NDA4, n=12)
# A tibble: 12 x 5
      ID Cluster week    NDA     W
   <dbl> <fct>   <chr> <dbl> <dbl>
 1   470 Class2  F3        7    21
 2   470 Class2  F6       NA    22
 3   472 Class1  F3        6    21
 4   472 Class1  F6        7    22
 5   474 Class3  F3       NA    21
```

```
 6   474 Class3  F6        4    22
 7   476 Class2  F3        2    21
 8   476 Class2  F6        1    22
 9   477 Class3  F3        0    21
10   477 Class3  F6        0    22
11   478 Class2  F3        4    21
12   478 Class2  F6        4    22
```

```
> NDA4 <- na.omit(NDA4)
> UID <- unique(NDA4$ID)
> table(NDA4$ID)
103 108 110 111 113 114 118 128 129 131 132 133 134 137 138 139 140 143
  2   2   2   2   1   2   2   2   2   2   2   2   2   2   2   2   2   2
144 145 146 150 151 154 155 156 160 161 165 166 167 168 170 171 175 176
  2   1   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
178 179 181 182 186 187 188 189 191 193 194 195 196 197 198 201 205 206
  2   2   2   2   2   2   2   2   2   2   2   2   1   2   2   2   2   2
207 208 209 210 211 214 215 217 218 223 228 231 233 236 237 240 244 249
  2   1   2   2   2   2   2   2   2   2   2   1   2   2   2   2   2   2
251 252 253 256 257 258 259 263 265 266 269 271 274 276 279 280 281 286
  2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
288 290 291 292 295 297 301 302 306 307 308 309 311 314 315 317 319 322
  2   1   1   2   2   2   1   2   2   2   2   2   2   2   2   2   2   2
323 324 326 327 328 330 331 334 335 339 340 341 343 346 351 353 355 356
  2   2   2   2   2   2   2   2   2   1   1   2   2   2   2   2   2   2
361 364 367 370 373 374 376 377 378 379 383 384 386 387 388 389 390 391
  2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
395 396 398 400 403 404 411 413 415 416 418 421 423 425 429 430 433 434
  2   2   2   2   2   2   2   2   2   2   1   2   2   2   2   2   2   2
435 436 437 440 441 444 446 447 450 451 454 456 457 463 469 470 472 474
  2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   1   2   1
476 477 478
  2   2   2
> UID[which(table(NDA4$ID) < 2)]
 [1] 113 145 196 208 231 290 291 301 339 340 418 470 474
```

14.5 Observed Means

Here I obtained the observed=fitted means and the respective 95% probability intervals. The within-subject observations are now assumed to be independent.

```
> M <- geeglm(cbind(NDA, 7-NDA) ~ 1 + factor(W) * Cluster, id=ID, data=NDA4,
             family="binomial", waves=W)
> summary(M)
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + factor(W) * Cluster,
    family = "binomial", data = NDA4, id = ID, waves = W)

 Coefficients:
                          Estimate Std.err  Wald Pr(>|W|)
```

```
(Intercept)                 1.5860  0.2201 51.92  5.8e-13 ***
factor(W)22                 0.1219  0.1876  0.42   0.5158
ClusterClass2              -0.8733  0.2862  9.31   0.0023 **
ClusterClass3              -1.5704  0.3083 25.95  3.5e-07 ***
factor(W)22:ClusterClass2  -0.2516  0.2223  1.28   0.2578
factor(W)22:ClusterClass3  -0.0283  0.2297  0.02   0.9020
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.507  0.0389

Correlation: Structure = independenceNumber of clusters:   183   Maximum cluster size: 2
> NDA4$Obs <- 7*fitted(M, type="response")
> NDA4$NDAlb <- 0
> NDA4$NDAub <- 0
> b <- coef(M)
> names(b) <- NULL
> Sigma <- M$geese$vbeta
> X <- model.matrix(M)
> K <- 1000
> R <- mvrnorm(K, mu=b, Sigma=Sigma)
> U <- 7*plogis(X %*% (t(R)))
> u <- apply(U,1,quantile, prob=.025)
> v <- apply(U,1,quantile, prob=.975)
> NDA4$NDAlb <- u
> NDA4$NDAub <- v
> head(as.data.frame(NDA4[NDA4$Cluster=="Class1",]), n=20)
    ID Cluster week NDA  W  Obs NDAlb NDAub
1  103  Class1   F3   5 21 5.81  5.32  6.19
2  103  Class1   F6   5 22 5.93  5.48  6.27
3  111  Class1   F3   7 21 5.81  5.32  6.19
4  111  Class1   F6   6 22 5.93  5.48  6.27
5  113  Class1   F3   7 21 5.81  5.32  6.19
6  114  Class1   F3   6 21 5.81  5.32  6.19
7  114  Class1   F6   6 22 5.93  5.48  6.27
8  129  Class1   F3   7 21 5.81  5.32  6.19
9  129  Class1   F6   4 22 5.93  5.48  6.27
10 140  Class1   F3   7 21 5.81  5.32  6.19
11 140  Class1   F6   5 22 5.93  5.48  6.27
12 144  Class1   F3   7 21 5.81  5.32  6.19
13 144  Class1   F6   6 22 5.93  5.48  6.27
14 145  Class1   F3   7 21 5.81  5.32  6.19
15 146  Class1   F3   7 21 5.81  5.32  6.19
16 146  Class1   F6   7 22 5.93  5.48  6.27
17 150  Class1   F3   7 21 5.81  5.32  6.19
18 150  Class1   F6   7 22 5.93  5.48  6.27
19 151  Class1   F3   7 21 5.81  5.32  6.19
20 151  Class1   F6   7 22 5.93  5.48  6.27
> head(as.data.frame(NDA4[NDA4$Cluster=="Class2",]), n=20)
```

```
    ID Cluster week NDA  W  Obs NDAlb NDAub
1  108  Class2   F3   5 21 4.70  4.15  5.22
2  108  Class2   F6   5 22 4.49  3.90  5.09
3  110  Class2   F3   7 21 4.70  4.15  5.22
4  110  Class2   F6   7 22 4.49  3.90  5.09
5  133  Class2   F3   6 21 4.70  4.15  5.22
6  133  Class2   F6   7 22 4.49  3.90  5.09
7  134  Class2   F3   7 21 4.70  4.15  5.22
8  134  Class2   F6   7 22 4.49  3.90  5.09
9  138  Class2   F3   7 21 4.70  4.15  5.22
10 138  Class2   F6   7 22 4.49  3.90  5.09
11 143  Class2   F3   7 21 4.70  4.15  5.22
12 143  Class2   F6   7 22 4.49  3.90  5.09
13 154  Class2   F3   7 21 4.70  4.15  5.22
14 154  Class2   F6   7 22 4.49  3.90  5.09
15 160  Class2   F3   7 21 4.70  4.15  5.22
16 160  Class2   F6   6 22 4.49  3.90  5.09
17 161  Class2   F3   7 21 4.70  4.15  5.22
18 161  Class2   F6   7 22 4.49  3.90  5.09
19 165  Class2   F3   3 21 4.70  4.15  5.22
20 165  Class2   F6   4 22 4.49  3.90  5.09
> head(as.data.frame(NDA4[NDA4$Cluster=="Class3",]), n=20)
    ID Cluster week NDA  W  Obs NDAlb NDAub
1  118  Class3   F3   6 21 3.53  2.80  4.16
2  118  Class3   F6   6 22 3.69  2.99  4.37
3  128  Class3   F3   0 21 3.53  2.80  4.16
4  128  Class3   F6   0 22 3.69  2.99  4.37
5  131  Class3   F3   4 21 3.53  2.80  4.16
6  131  Class3   F6   5 22 3.69  2.99  4.37
7  132  Class3   F3   0 21 3.53  2.80  4.16
8  132  Class3   F6   2 22 3.69  2.99  4.37
9  137  Class3   F3   7 21 3.53  2.80  4.16
10 137  Class3   F6   7 22 3.69  2.99  4.37
11 139  Class3   F3   4 21 3.53  2.80  4.16
12 139  Class3   F6   5 22 3.69  2.99  4.37
13 156  Class3   F3   1 21 3.53  2.80  4.16
14 156  Class3   F6   5 22 3.69  2.99  4.37
15 166  Class3   F3   0 21 3.53  2.80  4.16
16 166  Class3   F6   0 22 3.69  2.99  4.37
17 168  Class3   F3   5 21 3.53  2.80  4.16
18 168  Class3   F6   5 22 3.69  2.99  4.37
19 178  Class3   F3   3 21 3.53  2.80  4.16
20 178  Class3   F6   3 22 3.69  2.99  4.37


> NDAPlot7 <-ggplot(data=NDA2) +
  theme_classic() +
  xlab("Weekly Interval") + ylab("Number of Days Abstinent per Week") +
  geom_vline(xintercept = c(4,8), color="gray60") +
  geom_rect(xmin=19.7, xmax=20.3, ymin=-Inf, ymax=Inf, fill="white") +
  geom_vline(xintercept = c(20), color="gray60", linetype="dotted") +
  scale_color_manual(values=c("red","blue", "green"), name="Class") +
```

```
    scale_fill_manual( values=c("red","blue", "green"), name="Class") +
    scale_x_continuous(limits=c(0,22.5), breaks = c(0:19, 21, 22),
          minor_breaks = NULL, expand=expand_scale(add=c(.5,.1)),
          labels=c(paste0("H",0:4), "P1", "P2", "B1", "B2",
                   paste0("S0", 1:9), "S10", "S11", "F3", "F6")) +
    scale_y_continuous(breaks=0:7, limits=c(0,7),
                       sec.axis=sec_axis(~./7, name="Percent Days Abstinent")) +
    annotate("text", x=2, y=6.6, label="Distal") +
    annotate("text", x=6, y=6.6, label="Proximal") +
    annotate("text", x=14, y=6.6, label="Treatment") +
    annotate("text", x=21.5, y=6.6, label="Followup") +
    annotate("text", x=2, y=5.2, label="Class 1", fontface="italic") +
    annotate("text", x=2, y=2.2, label="Class 2", fontface="italic") +
    annotate("text", x=2, y=0.5, label="Class 3", fontface="italic") +
    geom_point(data=NDA2, aes(x=W, y=NDAobs, group=Cluster, color=Cluster)) +
    geom_ribbon(data=NDA2,
                aes(x=W, ymin=NDAlb, ymax=NDAub, group=Cluster, fill=Cluster), alpha=.1) +
    geom_line(data=NDA2, aes(x=W, y=NDAfit, group=Cluster, color=Cluster)) +
    geom_point(data=NDA4, aes(x=W, y=Obs, group=Cluster, color=Cluster),
               size=2, position=position_dodge(0.5)) +
    geom_linerange(data=NDA4, aes(x=W, ymin=NDAlb, ymax=NDAub, group=Cluster, color=Cluster),
                   position=position_dodge(0.5))+
    guides(color=FALSE) + guides(fill=FALSE)
> pdf("NDA/NDAPlot7.pdf")
> print(NDAPlot7)
> dev.off()
null device
          1
```

The fitted and observed data within each class are displayed in Figure 14 on the following page.

15 Additional Analyses

There were two additional analyses.

15.1 Polynomial Analysis of Proximal-Pretreatment Phase

Here I examime the trajectory structure of each class during the Proximal-Pretreatment phase, fromw Week −4 to Week 0. Models consisting of logistic-binomial, AR(1), quadratic polynomials with class-by-polynomial interactions were fitted to these data. Splines were unnecessary as there were no knots. There was no missing data.

```
> NDA5 <- filter(NDA2, W>3, W<9)
> P0 <- geeglm(cbind(NDA, 7-NDA)~1+ factor(W)*Cluster, id= ID, data=NDA5,
              family="binomial", waves=W)
> P1 <- geeglm(cbind(NDA, 7-NDA)~1+poly(W,2)*Cluster, id= ID, data=NDA5,
              family="binomial", waves=W, corstr = "ar1")
> Q1 <- geeglm(cbind(NDA, 7-NDA)~1+poly(W,2), id= ID,
              data=NDA5, subset=(Cluster=="Class1"),
              family="binomial", waves=W, corstr = "ar1")
> Q2 <- geeglm(cbind(NDA, 7-NDA)~1+poly(W,2), id= ID,
```

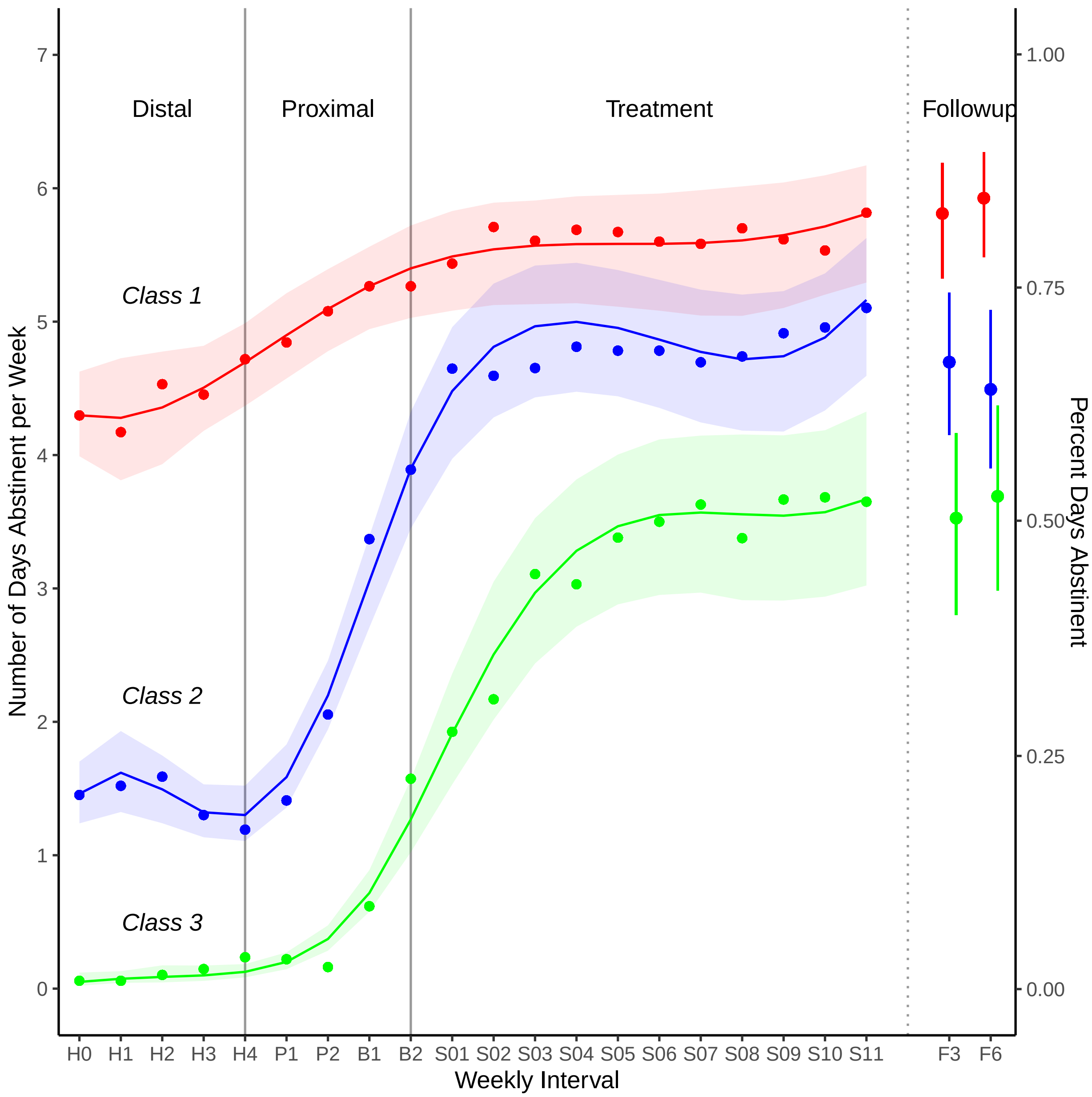

Figure 14: Observed and fitted means with 95% probability intervals for pretreatment, treatment, and followup for each class.

```
              data=NDA5, subset=(Cluster=="Class2"),
              family="binomial", waves=W, corstr = "ar1")
> Q3 <- geeglm(cbind(NDA, 7-NDA)~1+poly(W,2), id= ID,
              data=NDA5, subset=(Cluster=="Class3"),
              family="binomial", waves=W, corstr = "ar1")
> summary(P0)
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + factor(W) * Cluster,
    family = "binomial", data = NDA5, id = ID, waves = W)

 Coefficients:
                         Estimate Std.err   Wald Pr(>|W|)
(Intercept)                0.7268  0.1273  32.61  1.1e-08 ***
factor(W)5                 0.0825  0.1193   0.48    0.489
factor(W)6                 0.2448  0.1301   3.54    0.060 .
factor(W)7                 0.3837  0.1946   3.89    0.049 *
factor(W)8                 0.3837  0.2074   3.42    0.064 .
ClusterClass2             -2.3106  0.1805 163.91  < 2e-16 ***
ClusterClass3             -4.0855  0.4669  76.58  < 2e-16 ***
factor(W)5:ClusterClass2   0.1248  0.1979   0.40    0.528
factor(W)6:ClusterClass2   0.4608  0.2074   4.94    0.026 *
factor(W)7:ClusterClass2   1.1257  0.2839  15.72  7.4e-05 ***
factor(W)8:ClusterClass2   1.4241  0.3185  19.99  7.8e-06 ***
factor(W)5:ClusterClass3  -0.1492  0.5197   0.08    0.774
factor(W)6:ClusterClass3  -0.6303  0.5576   1.28    0.258
factor(W)7:ClusterClass3   0.6395  0.5387   1.41    0.235
factor(W)8:ClusterClass3   1.7369  0.5615   9.57    0.002 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
           Estimate Std.err
(Intercept)     0.31   0.068

Correlation: Structure = independenceNumber of clusters:   205   Maximum cluster size: 5
> summary(P1)
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + poly(W, 2) * Cluster,
    family = "binomial", data = NDA5, id = ID, waves = W, corstr = "ar1")

 Coefficients:
                          Estimate Std.err   Wald Pr(>|W|)
(Intercept)                  0.945   0.113  70.27  < 2e-16 ***
poly(W, 2)1                  4.623   2.283   4.10   0.0428 *
poly(W, 2)2                 -0.671   1.761   0.15   0.7029
ClusterClass2               -1.677   0.136 151.82  < 2e-16 ***
ClusterClass3               -3.720   0.183 412.11  < 2e-16 ***
poly(W, 2)1:ClusterClass2   17.171   3.661  22.00  2.7e-06 ***
poly(W, 2)2:ClusterClass2    1.707   2.416   0.50   0.4798
poly(W, 2)1:ClusterClass3   19.637   6.229   9.94   0.0016 **
poly(W, 2)2:ClusterClass3   13.206   4.743   7.75   0.0054 **
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.311  0.0725

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha    0.345  0.0733
Number of clusters:   205   Maximum cluster size: 5
> NDA5$Pobs <- 7*fitted(P0)
> NDA5$Pfit <- 7*fitted(P1)
> NDAPlot9 <- ggplot(data=NDA5) +
   xlab("Weekly Interval") + ylab("Number of Days Abstinent per Week") +
   scale_x_continuous(limits=c(4,8), breaks = 4:8, minor_breaks = NULL,
         labels=c("H4", "P1", "P2", "B1", "B2")) +
   scale_y_continuous(breaks=0:7, limits=c(0,7)) +
   geom_point(aes(x=W, y=Pobs, group=Cluster, color=Cluster)) +
   geom_line(aes(x=W, y=Pfit, group=Cluster, color=Cluster))
> pdf("NDA/NDAPlot9.pdf")
> print(NDAPlot9)
> dev.off()
null device
          1
```

The polynomial order of the Proximal-Pretest in each class is given in Tables 14 to 16 on pages 69–71: Class 1 shows a slight linear but no quadratic effect. Class 2 shows a larger linear effect but no quadratic effect. Class 3 shows both linear and quadratic effects.

The fitted and observed data withing each class are displayed in Figure 15 on page 72.

15.2 Tests of Changes from Basleline to Various End Points

The second analysis compared the change in NDAs from the beginning of treatment $\boldsymbol{B}2$ to the end of treatment at $\boldsymbol{S}11$, $\boldsymbol{F}3$ and $\boldsymbol{F}6$ among the three classes. These analyses use the orginal data frame `NDA1`.

The test of significance from $\boldsymbol{B}2$ to $\boldsymbol{S}11$ is given in Table 17 on page 73.

The tests with Class 2 and Class 3 as reference are given in Table 18 on page 74.

The test of significance from $\boldsymbol{B}2$ to $\boldsymbol{F}3$ is given in Table 19 on page 75.

The tests with Class 2 and Class 3 as reference are given in Table 20 on page 76.

The test of significance from $\boldsymbol{B}2$ to $\boldsymbol{F}6$ is given in Table 21 on page 77.

The tests with Class 2 and Class 3 as reference are given in Table 22 on page 78.

16 Publication Graphs

Here I collect some graphics that have been converted to black-and-white for publication. Figure 16 on page 80 is a modified version of Figure 14 on page 66.

```
> NDAPlot8 <- ggplot(data=NDA2) +
   theme_bw() +
```

Table 14: Proximal-Pretest Polynomial for Class 1

```
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + poly(W, 2), family = "binomial",
    data = NDA5, subset = (Cluster == "Class1"), id = ID, waves = W,
    corstr = "ar1")

 Coefficients:
            Estimate Std.err  Wald Pr(>|W|)
(Intercept)    0.945   0.113 70.24   <2e-16 ***
poly(W, 2)1    4.465   2.312  3.73    0.053 .
poly(W, 2)2   -0.670   1.797  0.14    0.709
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.305  0.0392

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha    0.613  0.0608
Number of clusters:   64   Maximum cluster size: 5
```

Table 15: Proximal-Pretest Polynomial for Class 2

```
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + poly(W, 2), family = "binomial",
    data = NDA5, subset = (Cluster == "Class2"), id = ID, waves = W,
    corstr = "ar1")

 Coefficients:
            Estimate Std.err Wald Pr(>|W|)
(Intercept)  -0.7323  0.0762 92.3  < 2e-16 ***
poly(W, 2)1  21.8446  2.8690 58.0  2.7e-14 ***
poly(W, 2)2   1.0505  1.6539  0.4     0.53
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.322  0.0293

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha     0.32  0.0629
Number of clusters:   73   Maximum cluster size: 5
```

Table 16: Proximal-Pretest Polynomial for Class 3

```
Call:
geeglm(formula = cbind(NDA, 7 - NDA) ~ 1 + poly(W, 2), family = "binomial",
    data = NDA5, subset = (Cluster == "Class3"), id = ID, waves = W,
    corstr = "ar1")

 Coefficients:
            Estimate Std.err   Wald Pr(>|W|)
(Intercept)   -2.786   0.144 373.23  < 2e-16 ***
poly(W, 2)1   24.277   5.624  18.64  1.6e-05 ***
poly(W, 2)2   13.100   4.335   9.13   0.0025 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)    0.304   0.207

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate Std.err
alpha    0.128  0.0908
Number of clusters:   68   Maximum cluster size: 5
```

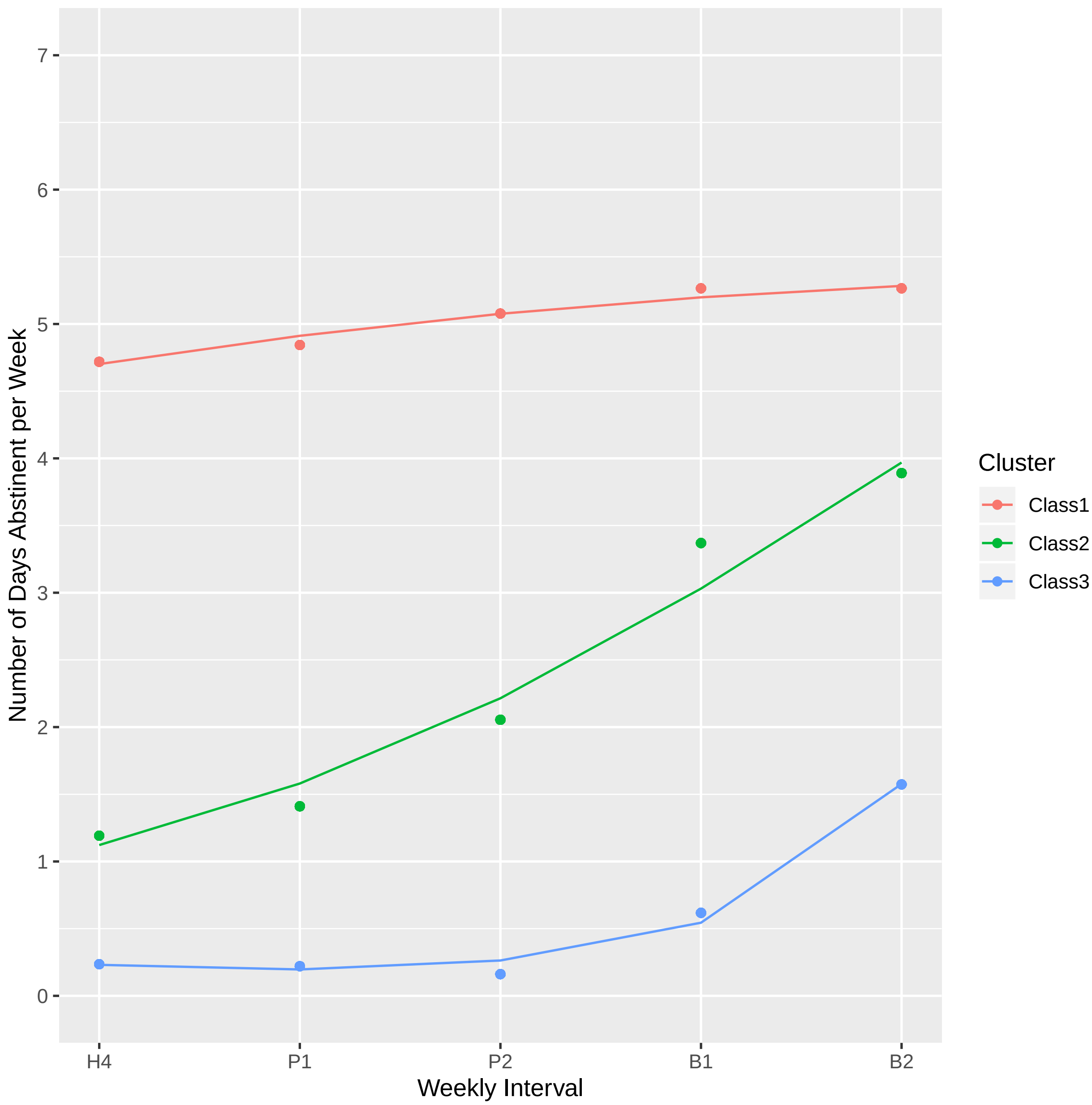

Figure 15: Observed means and quadratic polynomial fitted values for the three classes during the Proximal-Pretreatment phase.

Table 17: Test of Differences from $\boldsymbol{B}2$ to $\boldsymbol{S}11$: The upper panel displays the absolute difference among classes. The lower panel compares the relative difference among classes with Class 1 as the reference.

```
> summary(geeglm(I(round(7*S11))~offset(I(round(7*B2))) + 0 + Cluster, id=ID, data=NDA1))
Call:
geeglm(formula = I(round(7 * S11)) ~ offset(I(round(7 * B2))) +
    0 + Cluster, data = NDA1, id = ID)

 Coefficients:
              Estimate Std.err  Wald Pr(>|W|)
ClusterClass1    0.550   0.266  4.27  0.03872 *
ClusterClass2    1.221   0.323 14.24  0.00016 ***
ClusterClass3    2.217   0.413 28.80    8e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)      7.2    0.78

Correlation: Structure = independenceNumber of clusters:   188   Maximum cluster size: 1
```

```
> summary(geeglm(I(round(7*S11))~offset(I(round(7*B2))) + 1 + Cluster, id=ID, data=NDA1))
Call:
geeglm(formula = I(round(7 * S11)) ~ offset(I(round(7 * B2))) +
    1 + Cluster, data = NDA1, id = ID)

 Coefficients:
              Estimate Std.err  Wald Pr(>|W|)
(Intercept)      0.550   0.266  4.27  0.03872 *
ClusterClass2    0.671   0.419  2.56  0.10933
ClusterClass3    1.667   0.491 11.51  0.00069 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)      7.2    0.78

Correlation: Structure = independenceNumber of clusters:   188   Maximum cluster size: 1
```

Table 18: Test of Differences from $\boldsymbol{B}2$ to $\boldsymbol{S}11$: The upper panel uses Class 2 as reference. The lower panel uses Class 3 as reference.

```
> Cluster2 <- factor(NDA1$Cluster, levels=c(paste0("Class", c(2,1,3))))
> summary(geeglm(I(round(7*S11))~offset(I(round(7*B2))) + 1 + Cluster2, id=ID, data=NDA1))
Call:
geeglm(formula = I(round(7 * S11)) ~ offset(I(round(7 * B2))) +
    1 + Cluster2, data = NDA1, id = ID)

 Coefficients:
               Estimate Std.err  Wald Pr(>|W|)
(Intercept)       1.221   0.323 14.24  0.00016 ***
Cluster2Class1   -0.671   0.419  2.56  0.10933
Cluster2Class3    0.996   0.525  3.61  0.05760 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)      7.2    0.78

Correlation: Structure = independenceNumber of clusters:   188   Maximum cluster size: 1
```

```
> Cluster3 <- factor(NDA1$Cluster, levels=c(paste0("Class", c(3,1,2))))
> summary(geeglm(I(round(7*S11))~offset(I(round(7*B2))) + 1 + Cluster3, id=ID, data=NDA1))
Call:
geeglm(formula = I(round(7 * S11)) ~ offset(I(round(7 * B2))) +
    1 + Cluster3, data = NDA1, id = ID)

 Coefficients:
               Estimate Std.err  Wald Pr(>|W|)
(Intercept)       2.217   0.413 28.80    8e-08 ***
Cluster3Class1   -1.667   0.491 11.51  0.00069 ***
Cluster3Class2   -0.996   0.525  3.61  0.05760 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)      7.2    0.78

Correlation: Structure = independenceNumber of clusters:   188   Maximum cluster size: 1
```

Table 19: Test of Differences from $\boldsymbol{B}2$ to $\boldsymbol{F}3$: The upper panel displays the absolute difference among classes. The lower panel compares the relative difference among classes with Class 1 as the reference.

```
> summary(geeglm(I(round(7*F3))~offset(I(round(7*B2))) + 0 + Cluster, id=ID, data=NDA1))
Call:
geeglm(formula = I(round(7 * F3)) ~ offset(I(round(7 * B2))) +
    0 + Cluster, data = NDA1, id = ID)

 Coefficients:
              Estimate Std.err  Wald Pr(>|W|)
ClusterClass1    0.569   0.287  3.94   0.0471 *
ClusterClass2    0.909   0.326  7.77   0.0053 **
ClusterClass3    2.236   0.394 32.19  1.4e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)     6.76   0.679

Correlation: Structure = independenceNumber of clusters:   179   Maximum cluster size: 1
```

```
> summary(geeglm(I(round(7*F3))~offset(I(round(7*B2))) + 1 + Cluster, id=ID, data=NDA1))
Call:
geeglm(formula = I(round(7 * F3)) ~ offset(I(round(7 * B2))) +
    1 + Cluster, data = NDA1, id = ID)

 Coefficients:
              Estimate Std.err  Wald Pr(>|W|)
(Intercept)      0.569   0.287  3.94  0.04708 *
ClusterClass2    0.340   0.434  0.61  0.43340
ClusterClass3    1.667   0.487 11.71  0.00062 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)     6.76   0.679

Correlation: Structure = independenceNumber of clusters:   179   Maximum cluster size: 1
```

Table 20: Test of Differences from $\boldsymbol{B}2$ to $\boldsymbol{F}3$: The upper panel uses Class 2 as reference. The lower panel uses Class 3 as reference.

```
> Cluster2 <- factor(NDA1$Cluster, levels=c(paste0("Class", c(2,1,3))))
> summary(geeglm(I(round(7*F3))~offset(I(round(7*B2))) + 1 + Cluster2, id=ID, data=NDA1))
Call:
geeglm(formula = I(round(7 * F3)) ~ offset(I(round(7 * B2))) +
    1 + Cluster2, data = NDA1, id = ID)

 Coefficients:
               Estimate Std.err Wald Pr(>|W|)
(Intercept)       0.909   0.326 7.77   0.0053 **
Cluster2Class1   -0.340   0.434 0.61   0.4334
Cluster2Class3    1.327   0.512 6.73   0.0095 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)     6.76   0.679

Correlation: Structure = independenceNumber of clusters:   179   Maximum cluster size: 1
```

```
> Cluster3 <- factor(NDA1$Cluster, levels=c(paste0("Class", c(3,1,2))))
> summary(geeglm(I(round(7*F3))~offset(I(round(7*B2))) + 1 + Cluster3, id=ID, data=NDA1))
Call:
geeglm(formula = I(round(7 * F3)) ~ offset(I(round(7 * B2))) +
    1 + Cluster3, data = NDA1, id = ID)

 Coefficients:
               Estimate Std.err  Wald Pr(>|W|)
(Intercept)       2.236   0.394 32.19  1.4e-08 ***
Cluster3Class1   -1.667   0.487 11.71  0.00062 ***
Cluster3Class2   -1.327   0.512  6.73  0.00948 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)     6.76   0.679

Correlation: Structure = independenceNumber of clusters:   179   Maximum cluster size: 1
```

Table 21: Test of Differences from $\boldsymbol{B}2$ to $\boldsymbol{F}6$: The upper panel displays the absolute difference among classes. The lower panel compares the relative difference among classes with Class 1 as the reference.

```
> summary(geeglm(I(round(7*F6))~offset(I(round(7*B2))) + 0 + Cluster, id=ID, data=NDA1))
Call:
geeglm(formula = I(round(7 * F6)) ~ offset(I(round(7 * B2))) +
    0 + Cluster, data = NDA1, id = ID)

 Coefficients:
              Estimate Std.err  Wald Pr(>|W|)
ClusterClass1    0.630   0.269  5.49    0.019 *
ClusterClass2    0.692   0.366  3.58    0.059 .
ClusterClass3    2.309   0.400 33.34  7.7e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)     7.24   0.782

Correlation: Structure = independenceNumber of clusters:   174   Maximum cluster size: 1
```

```
> summary(geeglm(I(round(7*F6))~offset(I(round(7*B2))) + 1 + Cluster, id=ID, data=NDA1))
Call:
geeglm(formula = I(round(7 * F6)) ~ offset(I(round(7 * B2))) +
    1 + Cluster, data = NDA1, id = ID)

 Coefficients:
              Estimate Std.err  Wald Pr(>|W|)
(Intercept)     0.6296  0.2687  5.49  0.01913 *
ClusterClass2   0.0627  0.4540  0.02  0.89020
ClusterClass3   1.6795  0.4818 12.15  0.00049 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)     7.24   0.782

Correlation: Structure = independenceNumber of clusters:   174   Maximum cluster size: 1
```

Table 22: Test of Differences from $\boldsymbol{B}2$ to $\boldsymbol{F}6$: The upper panel uses Class 2 as reference. The lower panel uses Class 3 as reference.

```
> Cluster2 <- factor(NDA1$Cluster, levels=c(paste0("Class", c(2,1,3))))
> summary(geeglm(I(round(7*F6))~offset(I(round(7*B2))) + 1 + Cluster2, id=ID, data=NDA1))
Call:
geeglm(formula = I(round(7 * F6)) ~ offset(I(round(7 * B2))) +
    1 + Cluster2, data = NDA1, id = ID)

 Coefficients:
               Estimate Std.err Wald Pr(>|W|)
(Intercept)      0.6923  0.3660 3.58   0.0585 .
Cluster2Class1  -0.0627  0.4540 0.02   0.8902
Cluster2Class3   1.6168  0.5421 8.90   0.0029 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)     7.24   0.782

Correlation: Structure = independenceNumber of clusters:   174   Maximum cluster size: 1
```

```
> Cluster3 <- factor(NDA1$Cluster, levels=c(paste0("Class", c(3,1,2))))
> summary(geeglm(I(round(7*F6))~offset(I(round(7*B2))) + 1 + Cluster3, id=ID, data=NDA1))
Call:
geeglm(formula = I(round(7 * F6)) ~ offset(I(round(7 * B2))) +
    1 + Cluster3, data = NDA1, id = ID)

 Coefficients:
               Estimate Std.err Wald Pr(>|W|)
(Intercept)       2.309   0.400 33.3  7.7e-09 ***
Cluster3Class1   -1.679   0.482 12.2  0.00049 ***
Cluster3Class2   -1.617   0.542  8.9  0.00286 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Estimated Scale Parameters:
            Estimate Std.err
(Intercept)     7.24   0.782

Correlation: Structure = independenceNumber of clusters:   174   Maximum cluster size: 1
```

```
  scale_x_continuous(limits=c(0,22.5), breaks = c(0:19, 21, 22),
         minor_breaks = NULL, expand=expand_scale(add=c(.5,.1)),
         labels=c(paste0("-", 8:1), "0", paste0("+", 1:11), "F3", "F6")) +
  scale_y_continuous(breaks=0:7, limits=c(0,7),
                     sec.axis=sec_axis(~./7, name="Percent Days Abstinent per Week")) +
  xlab("Weekly Interval") + ylab("Number of Days Abstinent per Week") +
  geom_vline(xintercept = c(4,8), color="gray50", alpha=.4) +
  geom_rect(xmin=19.5, xmax=20.5, ymin=-Inf, ymax=Inf, fill="white") +
  geom_vline(xintercept = c(20), color="gray50", linetype="dotted") +
  theme(legend.position=c(.7,.2),
        legend.background = element_blank(), legend.key=element_rect()) +
  annotate("text", x=4.0, y=6.8, label="Pre-Treatment") +
  annotate("text", x=2, y=6.4, label="Distal") +
  annotate("text", x=6, y=6.4, label="Proximal") +
  annotate("text", x=14, y=6.6, label="Treatment") +
  annotate("text", x=21.5, y=6.6, label="Followup") +
  annotate("text", x=6, y=0, label="PS", size=3) +
  annotate("text", x=7, y=0, label="BL", size=3) +
    geom_ribbon(data=NDA2, aes(x=W, ymin=NDAlb, ymax=NDAub, group=Cluster, linetype=Cluster),
              color="grey60", fill="gray90", alpha=.2) +
  geom_point(data=NDA2, aes(x=W, y=NDAobs, group=Cluster, shape=Cluster), size=2) +
  geom_line(data=NDA2, aes(x=W, y=NDAfit, group=Cluster, linetype=Cluster)) +
  geom_point(data=NDA4, aes(x=W, y=Obs, group=Cluster, shape=Cluster),
             size=2, position=position_dodge(0.5)) +
  geom_linerange(data=NDA4, aes(x=W, ymin=NDAlb, ymax=NDAub, group=Cluster, linetype=Cluster),
                 position=position_dodge(0.5)) +
  scale_shape_discrete(
    labels=c("High Abstinence, Minimal Increase (n=64)",
             "Low Abstinence, Stable Increase (n=73)",
             "Non-abstinent, Accelerated Increase (n=68)"),
    guide=FALSE) +
  scale_linetype_discrete(
    labels=c("High Abstinence, Minimal Increase (n=64)",
             "Low Abstinence, Stable Increase (n=73)",
             "Non-abstinent, Accelerated Increase (n=68)"),
    guide=FALSE) +
  guides(linetype=guide_legend(title=NULL),
         shape=guide_legend(title=NULL))
> pdf("NDA/NDAPlot8.pdf")
> print(NDAPlot8)
> dev.off()
null device
          1
```

Figure 17 on page 82 is a modification of Figure 12 on page 58.

```
> NDAPlot10 <- ggplot(data=NDA2) +
  theme_bw() +
  scale_x_continuous(limits=c(0,19), breaks = c(0:19), minor_breaks = NULL,
         labels=c(paste0("-", 8:1), "0", paste0("+", 1:11))) +
  scale_y_continuous(breaks=0:7, limits=c(0,7),
                     sec.axis=sec_axis(~./7, name="Percent Days Abstinent")) +
```

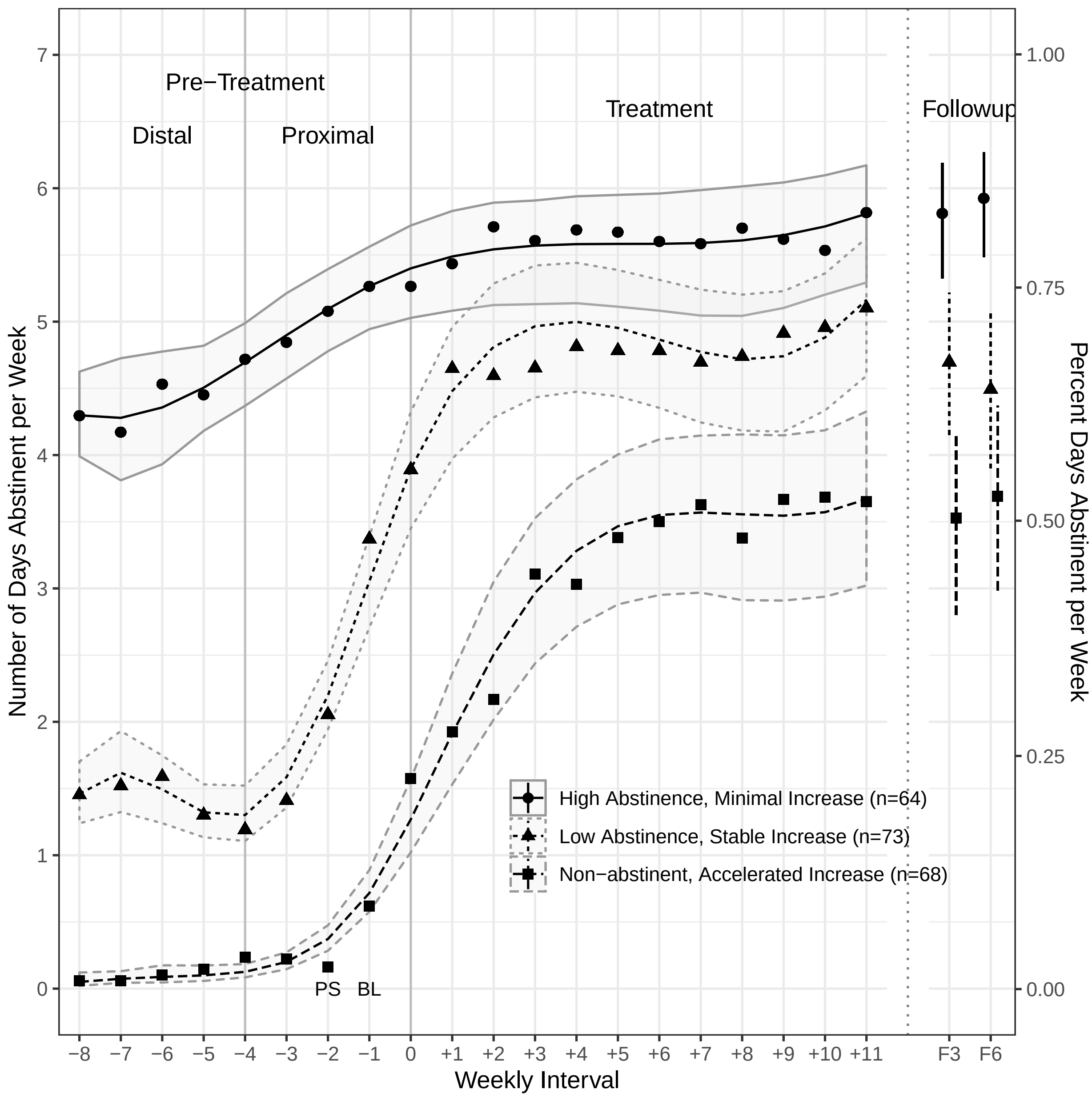

Figure 16: Observed and fitted means with 95% probability intervals for Distal-, Proximal-pretreatment, Treatment, and Follow-up phases for each class.

```
  xlab("Weekly Interval") + ylab("Number of Days Abstinent per Week") +
  geom_vline(xintercept = c(4,8), color="gray50") +
  theme(legend.position=c(.7,.2), legend.background = element_blank(),
        legend.key=element_rect()) +
  annotate("text", x=4, y=6.8, label="Pre-Treatment") +
  annotate("text", x=2, y=6.4, label="Distal") +
  annotate("text", x=6, y=6.4, label="Proximal") +
  annotate("text", x=14, y=6.6, label="Treatment") +
  annotate("text", x=6, y=0, label="PS", size=3) +
  annotate("text", x=7, y=0, label="BL", size=3) +
  annotate("text", x=12.1, y=.2, label="O   Combined N=205" ) +
  geom_line(data=NDA2, aes(x=W, y=Fit), linetype="solid", color="gray80" ) +
  geom_line(data=NDA2, aes(x=W, y=NDAfit, group=Cluster), color="gray80") +
  #scale_shape_manual(values=c(15,17,18)) +
  geom_point(data=NDA2, aes(x=W, y=NDAobs, group=Cluster, shape=Cluster), size=3) +
  geom_point(data=NDA2, aes(x=W, y=Obs), shape=1, size=3) +
  scale_shape_discrete(
    labels=c("High Abstinence, Minimal Increase (n=64)",
             "Low Abstinence, Stable Increase (n=73)",
             "Non-abstinent, Accelerated Increase (n=68)"),
    guide=FALSE) +
  scale_linetype_discrete(
    labels=c("High Abstinence, Minimal Increase (n=64)",
             "Low Abstinence, Stable Increase (n=73)",
             "Non-abstinent, Accelerated Increase (n=68)"),
    guide=FALSE) +
  guides(linetype=guide_legend(title=NULL), shape=guide_legend(title=NULL))
> pdf("NDA/NDAPlot10.pdf")
> print(NDAPlot10)
> dev.off()
null device
          1
```

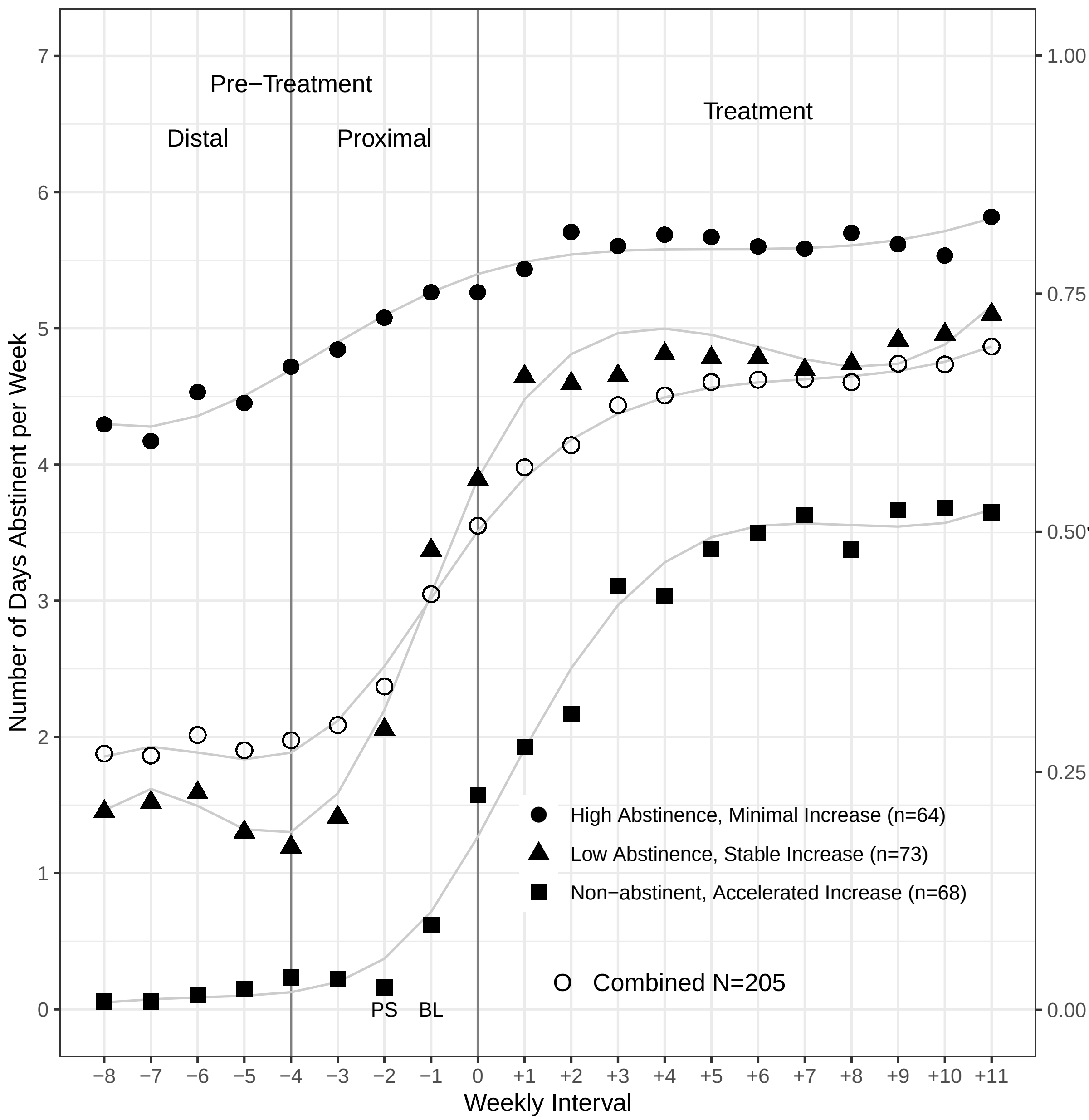

Figure 17: Observed (points) and fitted (lines) means for distal-, proximal-pretreatment, and treatment for each class. The gray points and line show the results for the entire sample.

References

Baggerly, K. A. & Coombes, K. R. (2009). Deriving chemosensitivity from cell lines: forensic bioinformatics and reproducible research in high-throughput biology. *The Annals of Applied Statistics*, *3*(4), 1309–1334.

Barba, L. A. (2018). Terminologies for reproducible research. *arXiv e-prints*, arXiv:1802.03311.

Diggle, P. J., Liang, K.-Y., & Zeger, S. L. (1994). *Analysis of longitudinal data*. Oxford, UK: Clarendon Press.

Donoho, D. L. (2010). An invitation to reproducible computational research. *Biostatistics*, *11*(3), 385–388.

Fraley, C. & Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, *97*(458), 611–631.

Gandrud, C. (2016). *Reproducible research with R and RStudio* (2nd ed.). Boca Raton, FL: Chapman & Hall/CRC.

Gentleman, R. & Lang, D. T. (2007). Statistical analyses and reproducible research. *Journal of Computational and Graphical Statistics*, *16*(1), 1–23.

Grün, B. & Leisch, F. (2007). Fitting finite mixtures of generalized linear regressions in R. *Computational Statistics & Data Analysis*, *51*(11), 5247–5252.

Grün, B. & Leisch, F. (2007). FlexMix: An R package for finite mixture modelling. *R News*, *7*(1), 8–13.

Grün, B. & Leisch, F. (2008). FlexMix version 2: Finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software*, *28*(4), 1–35.

Halekoh, U., Højsgaard, S., & Yan, J. (2006). The R package geepack for generalized estimating equations. *Journal of Statistical Software*, *15*(2), 1–11.

Hardin, J. W. & Hilbe, J. M. (2013). *Generalized estimating equations* (2nd ed.). Boca Raton, FL: CRC Press.

Harrell, Jr, F. E. (2019). *Hmisc: Harrell Miscellaneous*. R package version 4.2-0.

Hastie, T. J., Tibshirani, R. J., & Friedman, J. (2001). *The elements of statistical learning: data mining, inference, and prediction*. New York, NY: Springer.

Hemelrijk, J. (1966). Underlining random variables. *Statistica Neerlandica*, *20*(1), 1–7.

Herndon, T., Ash, M., & Pollin, R. (2013). Does high public debt consistently stifle economic growth? a critique of Reinhart and Rogoff. Technical Report 322, Political Economy Research Institute, University of Massachusetts Amherst.

Ihaka, R. (2010). R: lessons learned, directions for the future. In *Proceedings of the 2010 Joint Statistical Meetings*, Alexandria, VA. American Statistical Association, American Statistical Association.

Ihaka, R. & Gentleman, R. (1996). R: a language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, *5*(3), 299–314.

Laird, N. (2004). *Analysis of Longitudinal and Cluster-Correlated Data*. Beachwood, OH: Institute of Mathematical Sciences and the American Statistical Association.

Leisch, F. (2002). Sweave, part I: Mixing R and LaTeX. *R News*, *2*(3), 28–31.

Leisch, F. (2004). Flexmix: A general framework for finite mixture models and latent class regression in r. *Journal of Statistical Software, Articles*, *11*(8), 1–18.

Leisch, F. & R-Core (2015). *Sweave user manual*.

Little, R. J. A. & Rubin, D. B. (2002). *Statistical analysis with missing data* (2nd ed.). New York, NY: John Wiley & Sons.

McLachlan, G. & Peel, D. (2000). *Finite mixture models*. New York, NY: Wiley.

Mittelbach, F. & Goossens, M. (2004). *The LaTeX companion* (2nd ed.). Boston, MA: Addison-Wesley.

Peng, R. D. (2009). Reproducible research and Biostatistics. *Biostatistics*, *10*(3), 405–408.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Robins, J. M., Rotnitzky, A., & Zhao, L. P. (1995). Analysis of semiparametric regression models for repeated outcomes in the presence of missing data. *Journal of the American Statistical Association*, *90*(429), 106–121.

RStudio Team, (2015). *Rstudio: integrated development environment for R*. Boston, MA: RStudio, Inc.

Schafer, J. L. (2006). Marginal modeling of intensive longitudinaldata by generalized estimating equations. In T. A. Walls & J. L. Schafer (Eds.), *Models for intensive longitudinal data*. New York: Oxford Univerisity Press.

Schenk, C. (2019). *MiKTeX 2.9 Manual*.

Thieme, N. (2018). R generation 25. *Significance*, *15*(4), 14–19.

van Buuren, S. & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, *45*(3), 1–67.

Venables, W. N., Smith, D. M., & The R Development Core Team (2019). *An introduction to R. Notes on R: a programming environment for data analysis and graphics.* (3.6.0 ed.).

Wickham, H. (2017). *tidyverse: Easily Install and Load the 'Tidyverse'*. R package version 1.2.1.

Wickham, H. & Bryan, J. (2018). *readxl: Read Excel Files*. R package version 1.1.0.

Wickham, H. & Grolemund, G. (2017). *R for data science: import, tidy, transform, visualize, and model data*. Sebastopol, CA: O'Reilly Media.

Yan, J. (2002). geepack: yet another package for generalized estimating equations. *R-News*, *2/3*, 12–14.

Yan, J. & Fine, J. (2004). Estimating equations for association structures. *Statistics in Medicine*, *23*(6), 859–874.