# samon - the software

January 12, 2015

# Background

- Randomized study with outcome measurements taken at fixed time-points
- Monotone missing data pattern
- Interest is in a comparison of treatment arm means at the last scheduled time-point
- Outcomes are coded as positive integers
- Missing values are coded as -1
- Rows indicate individuals and columns indicate time-points
- Data at the first time-point (the baseline) is never missing

# Background

$$\textit{subject}\left\{\begin{array}{rrrrrrrr}
 & & & \overset{\textit{time-point 3}}{\swarrow} & & & & \\
82 & 88 & 81 & . & . & . & . & . \\
71 & 75 & 69 & 66 & 62 & 58 & 51 & 48 \\
62 & 63 & . & 55 & 61 & 66 & 68 & . \\
113 & 110 & 104 & 97 & . & . & . & . \\
88 & 92 & 99 & 70 & . & . & . & . \\
66 & 71 & 71 & 71 & 75 & 75 & 71 & 71 \\
90 & 88 & 88 & 88 & 77 & . & . & . \\
88 & 91 & 92 & 91 & 95 & 90 & 88 & . \\
. & 102 & 103 & 99 & 87 & 88 & . & .
\end{array}\right.$$

# Case Study: Chronic Schizophrenia

- RIS-INT-3 (Marder and Meibach, 1994, Chouinard *et al.*, 1993) was a multi-center study designed to assess the effectiveness and adverse experiences of four fixed doses of risperidone compared to haliperidol and placebo in the treatment of chronic schizophrenia.
- At selection, patients were required to have a PANSS (Positive and Negative Syndrome Scale) score between 60 and 120.

# RIS-INT-3

- Prior to randomization, there was a single-blind, one-week washout phase during which all anti-psychotic medications were to be discontinued.
- If acute psychotic symptoms occurred, patients were randomized to a double-blind treatment phase, scheduled to last 8 weeks.
- Patients were randomized to one of 6 treatment groups: risperidone 2, 6, 10 or 16 mg, haliperidol 20 mg, or placebo.
- Dose titration occurred during the first week of the double-blind phase.

# RSIP-INT-3

- Patients scheduled for 5 post-baseline assessements at weeks 1,2,4,6, and 8 of the double-blind phase.
- Primary efficiacy variable: PANSS score
- 521 patients randomized to receive placebo ($n = 88$), haliperidol 20 mg ($n = 87$), risperidone 2mg ($n = 87$), risperidone 6mg ($n = 86$), risperidone 10 mg ($n = 86$), or risperidone 16 mg ($n = 87$).
- Here we compare placebo (treatment 1) with risperidone 6mg (treatment 2).
- Data distributed with samon are simulated from this trial (they are not the original data).

# Central Question

*What is the difference in the mean PANSS scores at week 8 between risperidone at a dose of 6mg versus placebo in the counterfactual world in which all patients were followed to that week?*

# R samon library

- R interface to underlying C code provided by the samon library
- The library function loads the library, typically:

  library(samon, lib.loc=”location of library”)
- This will also let you load the datasets:

  data(samonPANSS1)

  data(samonPANSS2)

| function | description |
|---|---|
| samon | The main function. Takes data, estimates optimal $\sigma_P$ and $\sigma_Q$, produces (one-step IF) estimate for the input data and for pairs of parametric bootstrap samples. |
| samonCombine | Combines the outputs from samon into one samonMat object. Takes a list of filenames and combines them. |
| samonDiff | Takes two samonMat objects and produces a samonMat object for the difference in influence function estimates |
| samonBiasCorrection | Takes a samonMat object and produces corrected influence estimates |
| samonXBiasCorrection | Takes two samonMat objects (one from each treatment groups and for each pair of alphas produces an estimate of the difference in Influence function estimates. |

# The samon library

```
> library(samon,
+        lib.loc="../../../Rver0.1/samlib")
> data("samonPANSS1")
>
> print(samonPANSS1)
    V1  V2  V3  V4  V5  V6
1   90  87  86  93  72  87
2  112  -1  -1  -1  -1  -1
3   99  76  62  52  57  49
4   86  78  91 113  89  68
5   80  85  -1  -1  -1  -1
6   72  64  78 113  -1  -1
7   67  -1  -1  -1  -1  -1
8   96  -1  -1  -1  -1  -1
9   93  90  -1  -1  -1  -1
10  78  70  53  85  -1  -1
```

# The samonDataCheck function

- The samonDataCheck function can be used to check and describe data to ensure it is in samon canonical form
- The function takes a dataframe or matrix as its sole argument:

  ```
  samonDataCheck(samonPANSS1)
  ```

- Prints a small report on the data and returns a list with some useful objects. We will ignore this.

# samonDataCheck

```
> # Check treatment 1 data
> chk1 <- samonDataCheck( samonPANSS1 )


Samon Data Check:
--------------------------------
Number of time-points:         6
Number of subjects:           88
Minimum value:                44
Maximum value:               153

No Samon problems found
```

# samonDataCheck

```
Missing Patterns:
                   N   proportion
*_____    :        8       0.0909
**____    :       10       0.1136
***___    :       25       0.2841
****__    :       15       0.1705
*****_    :        7       0.0795
******    :       23       0.2614
```

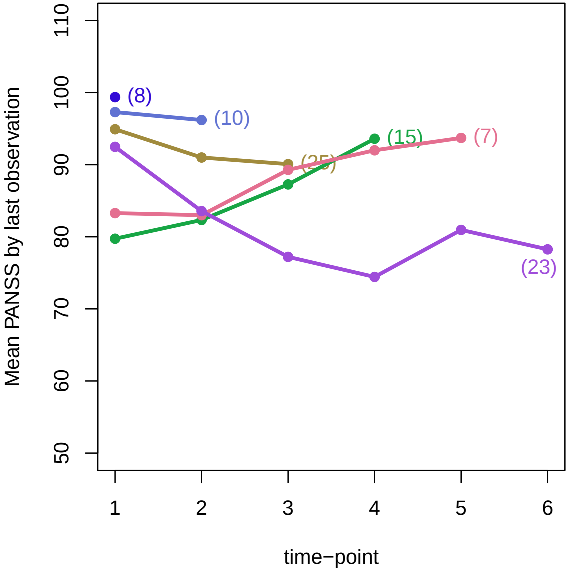
# samonDataCheck

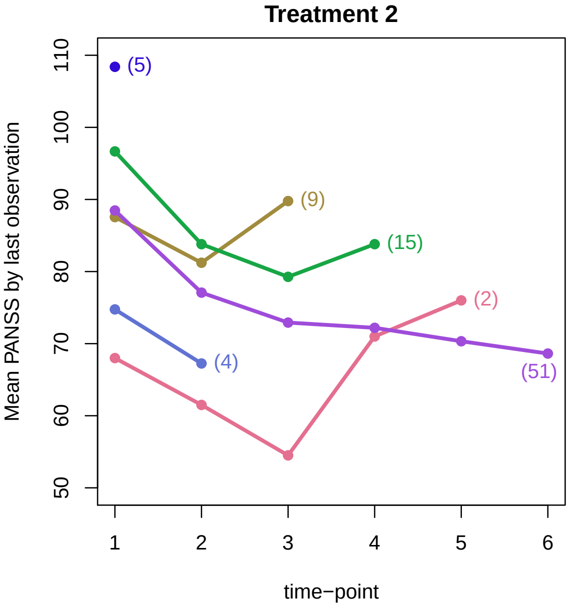
```
> # Check treatment 2 data
> chk2 <- samonDataCheck( samonPANSS2 )


Samon Data Check:
--------------------------------
Number of time-points:         6
Number of subjects:           86
Minimum value:                37
Maximum value:               135

No Samon problems found
```

# samonDataCheck

```
Missing Patterns:
                      N   proportion
*_____   :            5       0.0581
**____   :            4       0.0465
***___   :            9       0.1047
****__   :           15       0.1744
*****_   :            2       0.0233
******   :           51       0.5930
```

## Treatment 1


Mean PANSS by last observation
time−point
50
60
70
80
90
100
110
1
2
3
4
5
6
(8)
(10)
(25)
(15)
(7)
(23)


Treatment 2
Mean PANSS by last observation
time−point
50
60
70
80
90
100
110
1
2
3
4
5
6
(5)
(9)
(15)
(2)
(4)
(51)

# notation

- Consider each treatment group separately.
- Let $K$ denote the number of time-points after baseline and $n$ denote the number of individuals in a treatment group.
- $R_{t,j}$ indicates if individual $j$ is on study at time-point $t$, $1 \leq j \leq n$ and $0 \leq t \leq K$. That is $R_{t,j} = 1$ if individual $j$ is on study at time $t$ and $R_{t,j} = 0$ otherwise.
- If $R_{t,j} = 1$ then $Y_{t,j}$ denotes individual $j$'s outcome at time $t$.

# notation

- We sometimes wish to refer to the condition of being on-study without specifying which individual is involved. In this case we drop the subscript $j$, so that, for example, $Prob(R_t = 1)$ refers to the probability of being on study at time-point $t$.
- In a similar fashion $Y_t$ denotes an outcome value at time $t$.

# notation

The data in a treatment group are partitioned as follows. Given $N_P$ the number of partitions, the first partition $Prn_1$ is formed from individuals $1, 2, \ldots, [n/N_P]$, where the bracket notation, $[x]$, denotes the greatest integer less than or equal to $x$. If $2 < n \mid N_P$ the next $[n/N_P]$ set of individuals are assigned to partion $Prn_2$, otherwise the next $[n/N_P] + 1$ set of individuals are assigned to partion $Prn_2$. This process is continued until the whole treatment group is partitioned into $N_P$ partitions.

# The probability of continuing on study, P

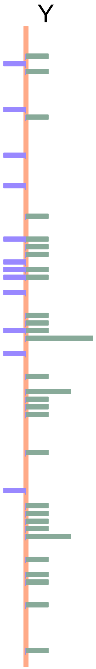
For a partition $Prn_s$, $s = 1, 2, \dots N_P$, the probability of staying on study at time $t$ given that an individual is on study at time $t-1$ and has outcome $Y_{t-1}$ at time $t-1$ is denoted by $P_s(R_t = 1 \mid R_{t-1} = 1, Y_{t-1})$. Invoking a first order Markov assumption we model this probability as follows:

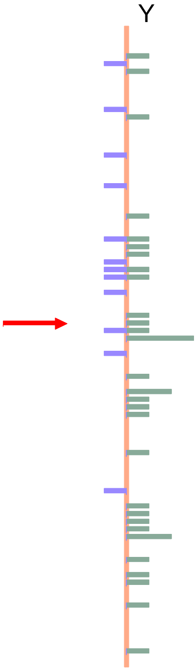
$$P_s(R_t = 1 \mid R_{t-1} = 1, Y_{t-1}) =$$

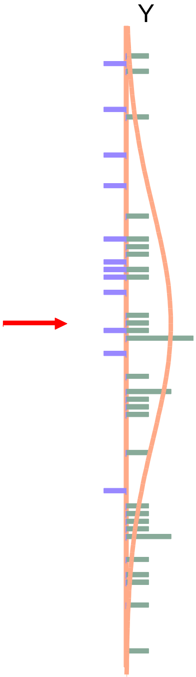
$$\frac{\sum_{j=1}^{N} I(j \notin Prn_s) R_{t-1,j} K(Y_{t-1,j}, Y_{t-1}, \sigma)}{\sum_{j=1}^{N} I(j \notin Prn_s) K(Y_{t-1,j}, Y_{t-1}, \sigma)}$$

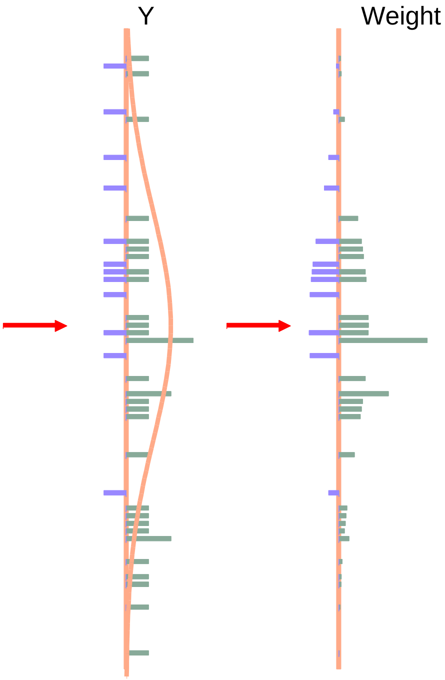
where we take the kernel $K$, to be normal

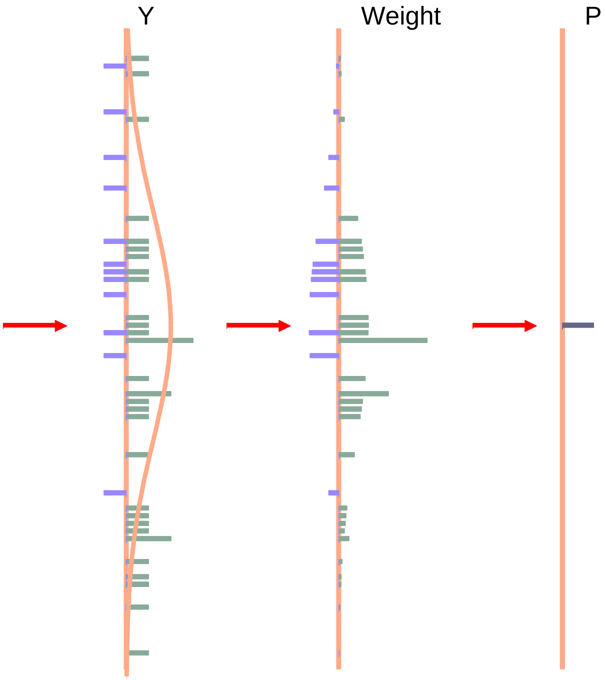
$$K(Y_{t-1,j}, Y_{t-1}, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{1}{2}\left(\frac{Y_{t-1,j} - Y_{t-1}}{\sigma}\right)^2}$$

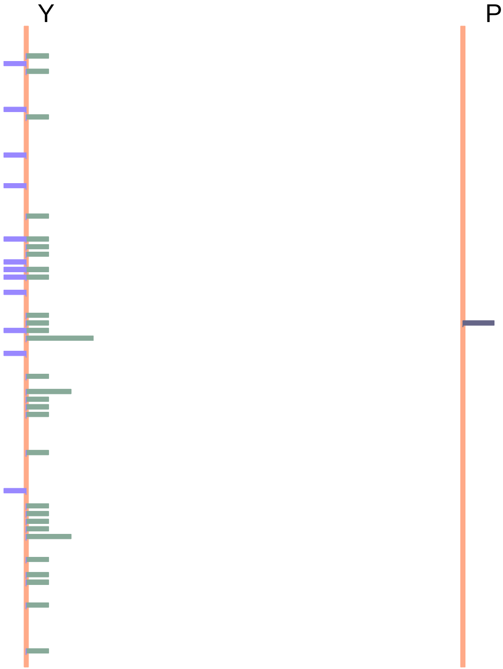

Y


Y

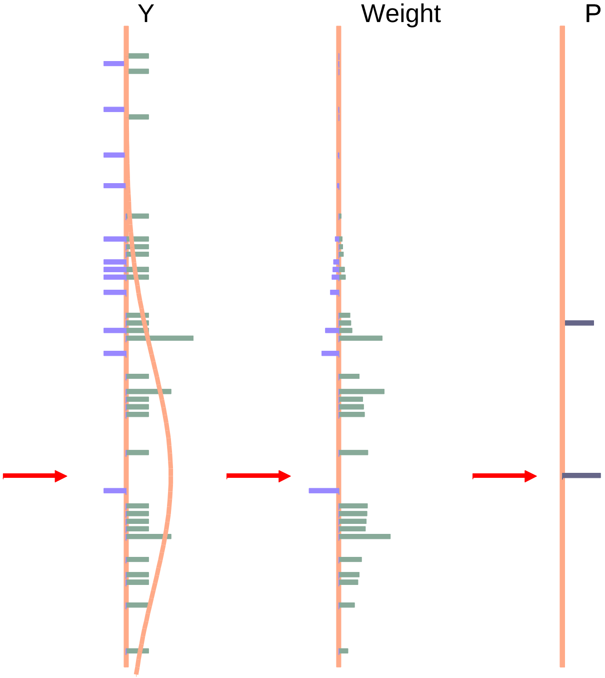

Y

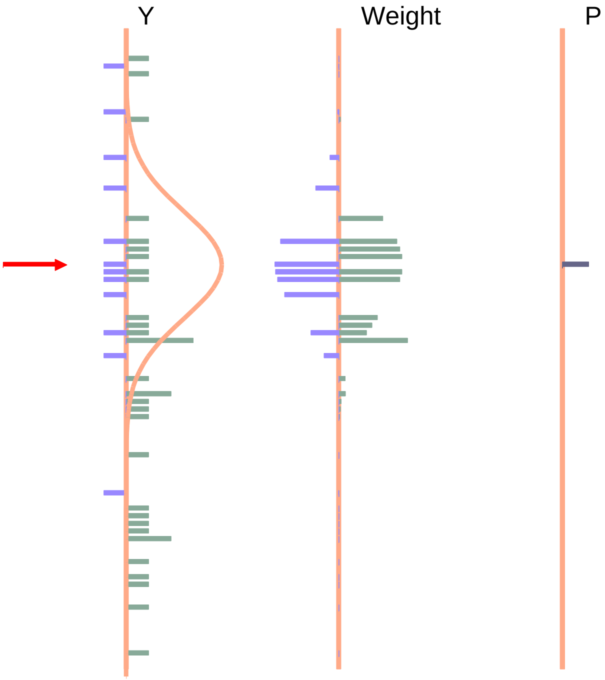

Y
Weight

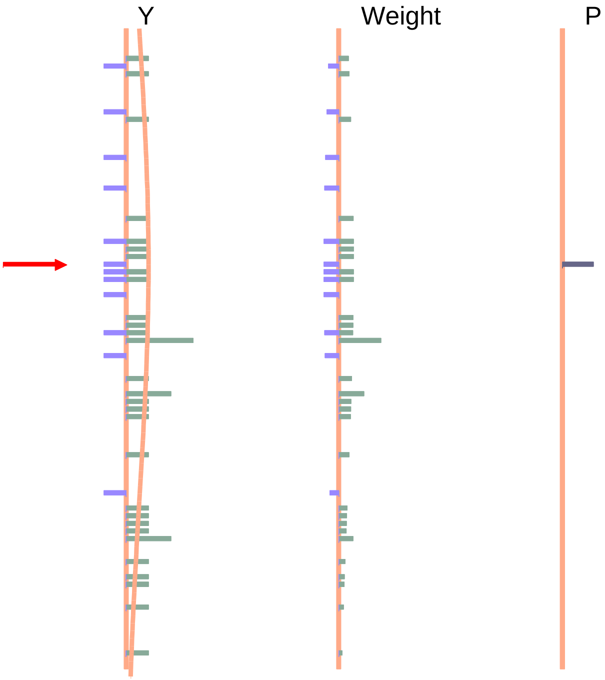

Y
Weight
P



SAMON


Y
Weight
P


Y
Weight
P



Frame: 30 of 62

SAMON

# The loss function

We choose an optimal $\sigma_P$ by minimizing the loss function $L_P$, which up to a constant is

$$L_P(\sigma) \ = \ \sum_{t=2}^{N_t} \sum_{j=1}^{n} R_{t-1,j} \left(R_{t,j} - P_{-j}(R_{t,j} = 1 \mid Y_{t-1,j})\right)^2$$

where $P_{-j}$ is the propability distribution derived from the data after removing the partition to which $j$ belongs.

# samonevalP and samonevalQ

- The functions samonevalP and samonevalQ can be used to compute the loss function for a range of $\sigma$.
- Takes five arguments:

```
samonevalP(
   mat = samonPANSS1,
   Npart = 10,
   LowSigmaP = 0.1,
   HighSigmaP = 100,
   IncrementSigmaP = 0.1
)
```

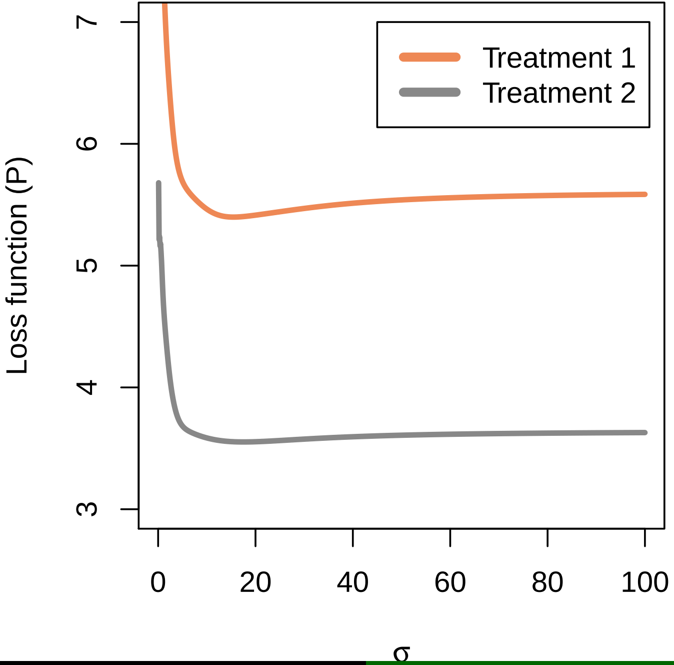
- returns a two column matrix, the first column containing $\sigma_P$ and the second the corresponding loss function value.

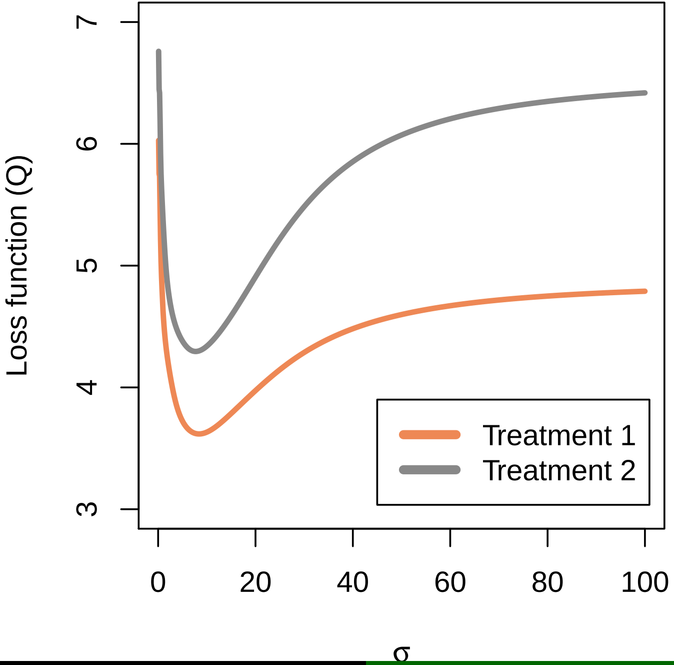
# The samon library

```
> library(samon,
+         lib.loc="../../../Rver0.1/samlib")
> data("samonPANSS1")
> data("samonPANSS2")
>
> ResultsP1 <- samonevalP(
+     mat             = samonPANSS1,
+     Npart           = 10,
+
+     IncrementSigmaP = 0.1,
+     LowSigmaP       = 0.1,
+     HighSigmaP      = 100.0
+                      )
```

# The samon library

```
> ResultsP2 <- samonevalP(
+     mat             = samonPANSS2,
+     Npart           = 10,
+
+     IncrementSigmaP = 0.1,
+     LowSigmaP       = 0.1,
+     HighSigmaP      = 100.0
+                      )
> ResultsP <- cbind(ResultsP1,ResultsP2)
> PQPlot(ResultsP,
+        "P.pdf", 4.2, 4.2,
+        "Loss function (P)",
+         c(0,100), c( 3, 7), c(45,7.0) )
```


Treatment 1
Treatment 2
Loss function (P)
3
4
5
6
7
0
20
40
60
80
100
σ


Loss function (Q)
7
6
5
4
3
0
20
40
60
80
100
σ
Treatment 1
Treatment 2

# samon

The samon function can be used to find the optimal values of $\sigma_P$ and $\sigma_Q$. Aguments include:

| | |
|---|---|
| mat | data matrix |
| Npart | Number of partitions |
| InitialSigmaP | initial value for sigmaP |
| LowSigmaP | lower bound |
| HighSigmaP | upper bound |
| InitialSigmaQ | initial value for sigmaQ |
| LowSigmaQ | lower bound |
| HighSigmaQ | upper bound |

```
# Example1.R
# Finding optimal Sigma_p and Sigma_q.
# ----------------------------------------
library(samon, lib.loc="../../samlib")
data("samonPANSS1")

samonResults <- samon(
    mat             = samonPANSS1,
    Npart           = 10,

    InitialSigmaP   = 10.0,
    LowSigmaP       = 0.001,
    HighSigmaP      = 50.0,

    InitialSigmaQ   = 8.0,
    LowSigmaQ       = 0.001,
    HighSigmaQ      = 50.0,

    nametype        = "PQ" )

# print the output
print(samonResults)
```

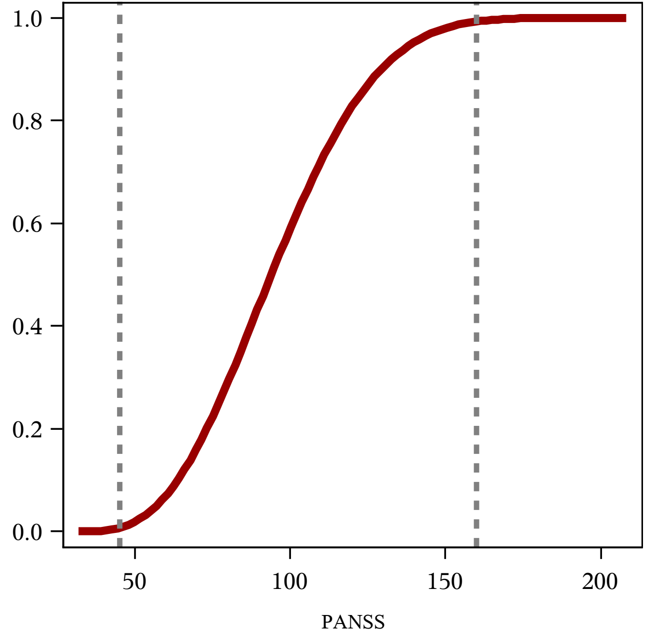
```
> print(samonResults)
     ResultType1 ResultType2 Sample
[1,]           0           1      0
[2,]           0           2      0

     Converged Iterations  Estimate  LossMin
[1,]         0         18 15.451858 5.398571
[2,]         0         14  8.399289 3.618053

      inputLB inputUB inputInitial
[1,]    0.001      50           10
[2,]    0.001      50            8
>
```

# Sensitivity Analysis

- Within samon the sensitivity bias function is the cumulative function of the beta distribution, a flexible function with bounded support.
- This together with the sensitivity analysis parameter $\alpha$ provides the mechanism by which we measure the sensitivity of the results to informative drop-out.
- $\alpha = 0$ is missing at random
- $\alpha$ quantifies the influence of $Y_{t+1}$ on the decision to drop-out between $t$ and $t + 1$.

# Sensitivity Analysis

- The cumulative beta function is defined on the interval (0,1) and in order to use it as the sensitivity bias function we need to map the range of our data into (0,1).
- In the case of PANSS data there are theoretical limits in that PANSS scores range between 30 and 210.
- Clinical practice gives a range of values over which a change in PANSS noticeable effect. This translates to parameters for the cumulative beta function $\zeta_1$ and $\zeta_2$.
- Another strategy might be to fit a beta distribution to the data (after suitible transformation) to determine $\zeta_1$ and $\zeta_2$.

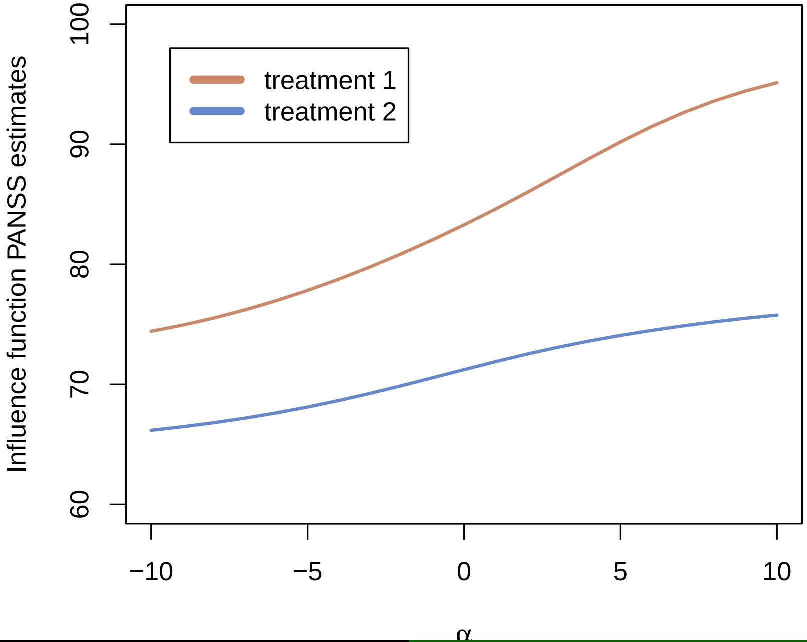

1.0
0.8
0.6
0.4
0.2
0.0
50
100
150
200
PANSS

```
# Example2_partA.R
# Produce one-step influence function estimates
# ----------------------------------------------
library(samon, lib.loc="../../samlib")
data(samonPANSS1)

Results1 <- samon(
    mat                = samonPANSS1,
    Npart              = 10,

    # initial value, lower and upper
    # bounds for sigmaP
    InitialSigmaP      = 10.0,
    LowSigmaP          = 0.001,
    HighSigmaP         = 50.0,

    # initial value, lower and upper
    # bounds for sigmaQ
    InitialSigmaQ      = 8.0,
    LowSigmaQ          = 0.001,
    HighSigmaQ         = 50.0,
    LowAlpha        = -10,     # alphas
    HighAlpha       =  10,
    IncrementAlpha  =   1,

    lb              = 30,      # parameters for
    ub              = 210,     # cumulative
    zeta1           = 4.0,     # beta distribution
    zeta2           = 7.0,

    nametype        = "Alpha" )

print(Results1)
saveRDS(Results1,"Results1.rds")
```

```
> print(Results1[ Results1[,2] == 3,
+                c( 6, 7, 8, 9, 10)])
                 PlugIn          Var         IF       Var
       Alpha   Estimate       PlugIn   Estimate        IF
 1       -10     66.571     0.053319    66.1752    5.6039
 2        -9     66.831     0.055501    66.4692    5.6463
 3        -8     67.134     0.058060    66.8041    5.7037
 4        -7     67.487     0.061065    67.1863    5.7787
 5        -6     67.898     0.064575    67.6211    5.8720
 6        -5     68.371     0.068629    68.1113    5.9814
 7        -4     68.907     0.073227    68.6562    6.0991
 8        -3     69.502     0.078313    69.2513    6.2110
 9        -2     70.146     0.083753    69.8875    6.2988
10        -1     70.824     0.089315    70.5515    6.3455
11         0     71.517     0.094625    71.2249    6.3437
12         1     72.203     0.099170    71.8857    6.3012
13         2     72.866     0.102389    72.5120    6.2374
14         3     73.492     0.103843    73.0884    6.1700
15         4     74.072     0.103363    73.6086    6.1066
16         5     74.603     0.101143    74.0746    6.0467
17         6     75.085     0.097727    74.4928    5.9872
```

```
# Example2_partC.R
# Retrieve the estimates and plot them
# -------------------------------------------------
Results1 <- readRDS("Results1.rds")
Results2 <- readRDS("Results2.rds")

# Select the IF results from the results matrices
Trt1Results <- Results1[ Results1[,2] == 3, ]
Trt2Results <- Results2[ Results2[,2] == 3, ]

pdf(file="Example2.pdf", height=5.5, width=5)
par(mar=c(4,4,2,2))

plot.new()
plot.window( xlim = c(-5, 5), ylim = c( 60, 100 ))
```

```
lines( x = Trt1Results[,6], y = Trt1Results[,9],
       lwd=2, col = "#CC8866FF")
lines( x = Trt2Results[,6], y = Trt2Results[,9],
       lwd=2, col = "#6688CCFF")

axis(1); axis(2)

legend( -4.4, 98,
         legend = c("treatment 1", "treatment 2"),
         xjust=0, yjust=1,
         lty = c("solid","solid"), lwd=c(5,5),
         col=c("#CC8866FF","#6688CCFF"))

title( xlab = expression(alpha),
       ylab = "Influence function PANSS estimates")
box()
dev.off()
```


treatment 1
treatment 2
Influence function PANSS estimates
100
90
80
70
60
−10
−5
0
5
10
α

- Use double bootstrap to compute bias corrected IF estimates and standard errors.
- Using the optimal $\sigma_P$ and $\sigma_Q$ the original data is used to create a sample, sampleA say. Optimal $\sigma_P$ and $\sigma_Q$ and then found for sampleA and in turn a new sample is created, sampleB. The two samples sampleA and sampleB are a single sample pair of the original data.
- The one-step IF estimate is created for each alpha for both sampleA and sampleB.
- samon is used to generate multiple sample pairs.

```
# Example3_1a.R
# Produce influence function estimates and 500
# pairs of bootstrap estimates for treatment 1.
# ----------------------------------------------
library(samon, lib.loc="../../samlib")
data("samonPANSS1")

Results1a <- samon(
    mat             = samonPANSS1,
    Npart           = 10,

    InitialSigmaP   = 10.0,
    LowSigmaP       = 0.001,
    HighSigmaP      = 50.0,

    InitialSigmaQ   = 8.0,
    LowSigmaQ       = 0.001,
    HighSigmaQ      = 50.0,
  LowAlpha        = -5,       # alphas
  HighAlpha       =  5,
  IncrementAlpha  = 0.5,

  lb              = 30,       # parameters to
  ub              = 210,      # cumulative
  zeta1           = 4.0,      # beta distribution
  zeta2           = 7.0,

  NBootstrapPairs = 500,      # number of bootstrap
  seed0           = 826847827 )

# save results for latter use
saveRDS(Results1a, "Results1a.rds")
```

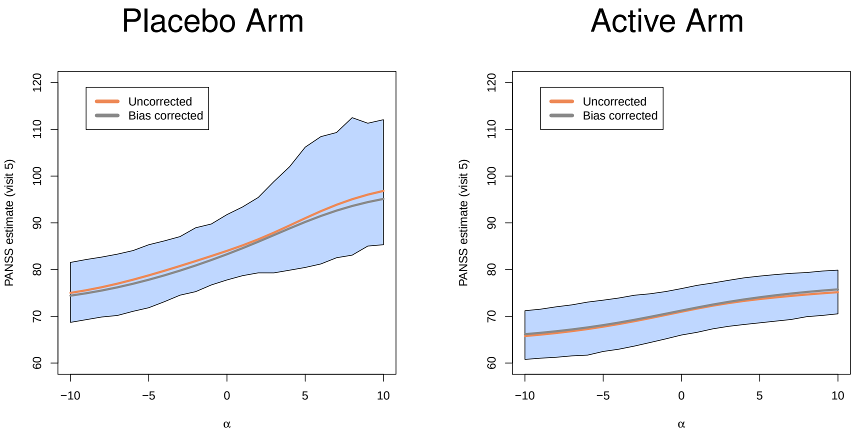
| function | description |
|---|---|
| samonCombine | combines the outputs from samon into one samonMat object.<br>The results are stored in rds files. samonCombine takes a list of such files and combines them. |
| samonDiff | Takes two samonMat objects and produces a samonMat object for the difference in influence function estimates |
| samonBiasCorrection | Takes a samonMat object and produces corrected influence estimates |
| samonXBiasCorrection | Takes two samonMat objects (one from each treatment groups and for each pair of alphas produces an estimate of the difference in Influence function estimates. |

```
# Example3_plots.R
#
# Results have previously been stored in rds files
# treatment 1:   Results3a.rds,  Results3b.rds
# treatment 2:   Results3c.rds,  Results3d.rds
#
# 1. put the results together
# 2. create a difference object
# 3. create bias corrected estimates
# 4. plot some results
# --------------------------------------------------
library(samon,lib.loc="../../samlib")

filenames1 <- c("Results1a.rds", "Results1b.rds")
filenames2 <- c("Results2a.rds", "Results2b.rds")

trt1Results <- samonCombine( filenames1 )
trt2Results <- samonCombine( filenames2 )

# the difference trt2 - trt1
diffResults <- samonDiff( trt1Results, trt2Results)
```

```
# correct each result
BiasCorrection1 <- samonBiasCorrection(
                    trt1Results, -10:10,  1)
BiasCorrection2 <- samonBiasCorrection(
                    trt2Results, -10:10,  2)
BiasCorrectionD <- samonBiasCorrection(
                    diffResults, -10:10, -1)

samonPlot <- function( Res, file, height,
                      width, ylab, xlim,
                      ylim, legpos) {

  pdf(file=file, height=height, width=width)
  par(mar=c(4,4,2,2))

  # bands for CIs
  polyRes <- rbind( Res[, c(1,4)],
                    Res[ nrow(Res):1, c(1,5)] )

  plot.new()
  plot.window( xlim = xlim, ylim = ylim )
  lines( x = Res[,1], y = Res[,3], lwd=3,
          col = "#EE8855FF")
  lines( x = Res[,1], y = Res[,2], lwd=3,
          col = "#888888FF")

  legend( legpos[1], legpos[2],
          legend = c("Uncorrected", "Bias corrected"),
          xjust=0, yjust=1,
          lty = c("solid","solid"), lwd=c(5,5),
          col=c("#EE8855FF","#888888FF"))

  axis(1)
  axis(2)

  title( xlab = expression(alpha),  ylab = ylab)
  box()

  dev.off()
  invisible(return())
}
```

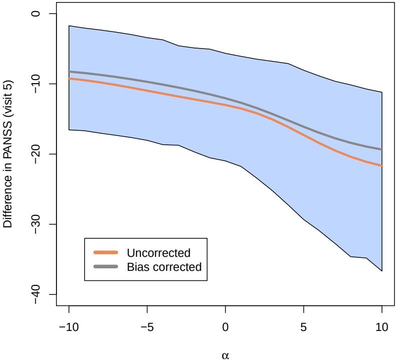
```
# treatment 1
samonPlot(BiasCorrection1[, c(2,3,8,6,7)],
          "Trt1Est.pdf",
          5.5, 6,
          "PANSS estimate (visit 5)",
          c(-10,10), c( 60, 120),
          legpos = c( -9, 119) )

# treatment 2
samonPlot(BiasCorrection2[, c(2,3,8,6,7)],
          "Trt2Est.pdf",
          5.5, 6,
          "PANSS estimate (visit 5)",
          c(-10,10), c( 60, 120),
          legpos = c( -9, 119) )

# and treatment 2 minus treatment 1
samonPlot(BiasCorrectionD[, c(2,3,8,6,7)],
          "TrtDEst.pdf",
          5.5, 6,
         "Difference in PANSS (visit 5)",
```

# Estimated PANSS score at visit 5



# Estimated PANSS score at visit 5

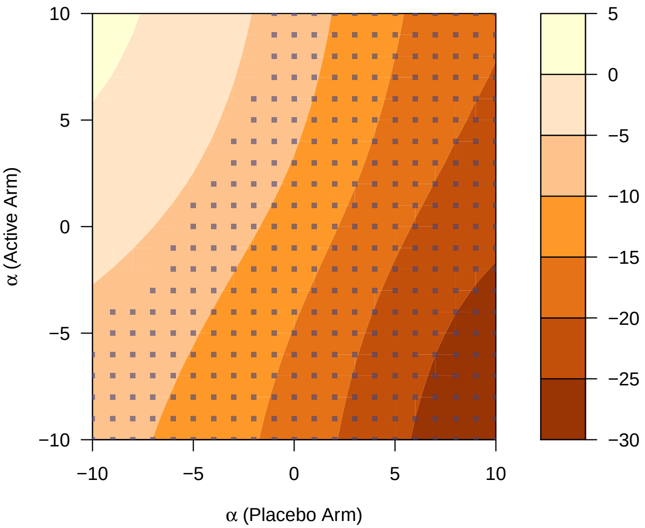
## Difference (active - placebo)



Another useful plot is a surface plot of the difference in the estimated mean value in the two treatment groups given as a function of the two alpha parameters. We use the samonXBiasCorrection function to compute the difference in estimates for each pair of alpha. The plotting is done with the filled.contour function.

```
# Example3_contourPlot.R
# A contour plot of the difference in estimates.
# ------------------------------------------------
library(samon,lib.loc="../../samlib")

filenames1 <- c("Results1a.rds", "Results1b.rds")
filenames2 <- c("Results2a.rds", "Results2b.rds")

trt1Results <- samonCombine( filenames1 )
trt2Results <- samonCombine( filenames2 )

XRes <- samonXBiasCorrection( trt1Results,
                              trt2Results, -10:10 )
```

```
pdf(file="Example3_contour.pdf", height=5, width=6)
par(mar=c(4,4,2,2))

filled.contour(
  x       = -10:10,
  y       = -10:10,
  z       = matrix(XRes[,3],21,21, byrow=TRUE),
  xlab    = expression(paste(alpha, " (Placebo Arm)")),
  ylab    = expression(paste(alpha, " (Active Arm)")),
  nlevels = 8,
  color.palette = colorRampPalette(c( "#993404","
     #D95F0E","#FE9929", "#FFD9BE","#FFFFD4"),
     space="rgb"),
  plot.axis = ( points(
        XRes[ sign(XRes[,6]) == sign(XRes[,7]),
        c(1,2)], pch=15, cex=0.6,
        col = c("#44447799")))
              )
dev.off()
```



10
5
0
−5
−10
−10
−5
0
5
10
α (Active Arm)
α (Placebo Arm)
5
0
−5
−10
−15
−20
−25
−30