```
load("synthetic_raw.Rda")
library(SmoothHOOI)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

This is an illustration of how to use our functions on synthetic data

```
set.seed(32123)
```

Generate synthetic data

```
synthetic <- synthetic_data(L_tilde, R_tilde, mean_G, cov_G, E, p=207, noise_level=1, pattern="random",

synthetic_dat <- synthetic$sim_Mmiss
```

Make the second order difference matrix

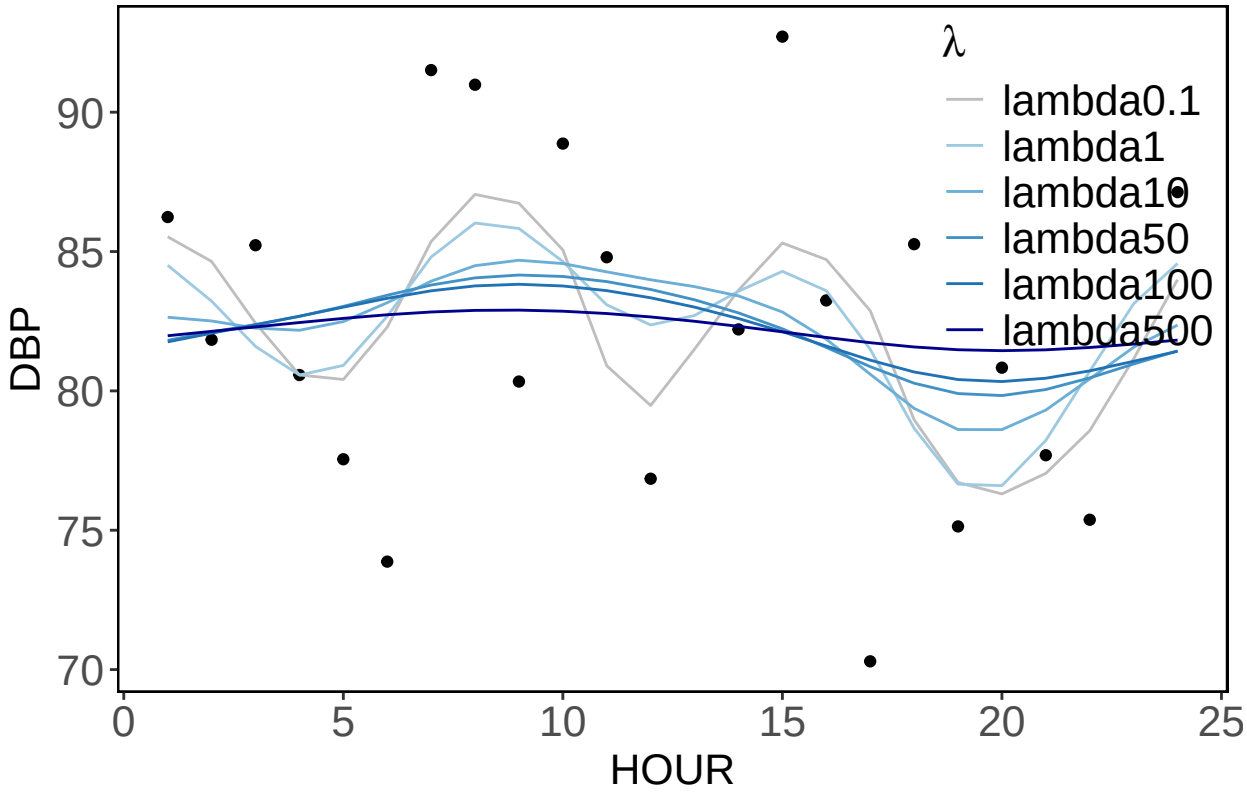
```
D2 <- SecDiffMat(24)
```

Coarse-to-fine grid search (Example)

```
original <- (synthetic_dat@data[ , 1, 180])*12+70
lambda_seq <- c(0.1, 1, 10, 50, 100, 500)
est_df <- matrix(NA, nrow=24, ncol=length(lambda_seq))
for (i in 1:length(lambda_seq)){
  lambda <- lambda_seq[i]
  res <-  mglram(tnsr = synthetic_dat@data, ranks = c(6, 3), init=0, D = D2,
                 lambda = lambda, max_iter = 500, tol = 1e-5, L0 = NULL)
  est_df[,i] <- (res$est[,1,180])*12 +70
}
org_df <- data.frame(cbind(c(1:24), original))
est_df <- data.frame(cbind(c(1:24), est_df))
```

```r
colnames(org_df) <- c("hour", "original")
colnames(est_df) <- c("hour", paste0("lambda",lambda_seq))
est_long <- pivot_longer(est_df, cols = starts_with("lambda"),
                         names_to = "lambda", values_to = "y")
est_long$lambda <- factor(est_long$lambda, levels=c("lambda0.1","lambda1","lambda10","lambda50","lambda
```

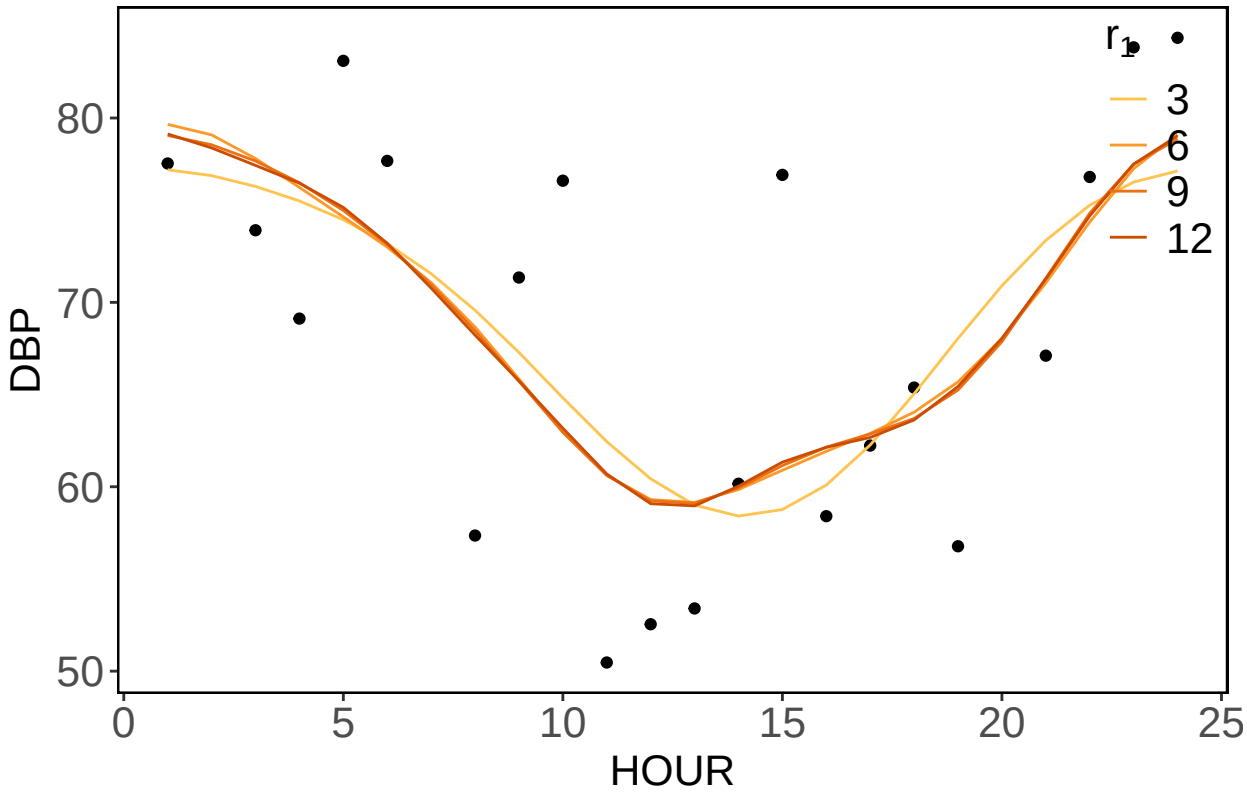
```r
custom_colors <- c("grey","#9ECAE1","#6BAED6", "#4292C6", "#2171B5","darkblue")
names(custom_colors) <- levels(factor(est_long$lambda))

ggplot() +
  geom_point(data = org_df, aes(x = hour, y = original), color = "black") +  # Original data
  geom_line(data = est_long, aes(x = hour, y =y, color = lambda)) +  # Estimated curves
  labs(title = "",
       x = "HOUR", y = "DBP") +
  scale_color_manual(values = custom_colors,
                     name = expression(lambda)) +
  theme(legend.position = c(1, 1),
    legend.justification = c(1, 1),
    panel.grid = element_blank(),
    panel.background = element_blank(),
    plot.background = element_blank(),
    panel.border = element_rect(color = "black", fill = NA, linewidth = 0.8),  # add border rectangle
    axis.line = element_blank(),
    legend.background = element_blank(),  # removes the outer background & border
    legend.key = element_blank(),
    axis.text = element_text(size = 16),
    axis.title = element_text(size = 16),
    legend.text = element_text(size = 16,hjust=0),
    legend.title = element_text(size = 16)
    )
```




```
original <- (synthetic_dat@data[ , 1, 1])*12 + 70
r1_seq <- c(3,6,9,12)
est_df <- matrix(NA, nrow=24, ncol=length(r1_seq))

for (i in 1:length(r1_seq)){
  r1 <- r1_seq[i]
  res <-  mglram(tnsr = synthetic_dat@data, ranks = c(r1, 3), init=0, D = D2,
                 lambda = 10, max_iter = 500, tol = 1e-5, L0 = NULL)
  est_df[,i] <- (res$est[,1,1])*12 + 70
}

colnames(est_df) <- paste0("r", r1_seq)
est_df <- cbind(hour = 1:24, est_df)
est_df <- data.frame(est_df)
org_df <- data.frame(hour = 1:24, original = original)
est_long <- pivot_longer(est_df, cols = starts_with("r"),
                         names_to = "r", values_to = "y")
est_long$r <- factor(est_long$r, levels = paste0("r", r1_seq))
```

```
custom_colors <- c("#FEC44F", "#FE9929","#EC7014","#CC4C02")
names(custom_colors) <- levels(factor(est_long$r))

ggplot() +
  geom_point(data = org_df, aes(x = hour, y = original), color = "black") +  # Original data
  geom_line(data = est_long, aes(x = hour, y =y, color = r)) +  # Estimated curves
  labs(title = "",
       x = "HOUR", y = "DBP") +
  scale_color_manual(values = custom_colors,
```

```
                    labels = c("r3" = 3, "r6" = 6,
                               "r9" = 9, "r12" = 12),
                    name =  expression(r[1])) +
  theme(legend.position = c(1, 1),
    legend.justification = c(1, 1),
    panel.grid = element_blank(),
    panel.background = element_blank(),
    plot.background = element_blank(),
    panel.border = element_rect(color = "black", fill = NA, linewidth = 0.8),  # add border rectangle
    axis.line = element_blank(),
    legend.background = element_blank(),  # removes the outer background & border
    legend.key = element_blank(),
    axis.text = element_text(size = 16),
    axis.title = element_text(size = 16),
    legend.text = element_text(size = 16,hjust=0),
    legend.title = element_text(size = 16)
    )
```



Find optimal hyperparameter using k-fold cross-validation

```
start_time <- Sys.time()

kcv_res <- kcv(tnsr=synthetic_dat@data, rank_grid=as.matrix(expand.grid(r1<-seq(3,6,by=1), r2<-c(2,3))),
               k=5, L0=NULL, D=D2, tol=0.01, max_iter=500, init=0)
```

```r
end_time <- Sys.time()

exec_time <- end_time - start_time
print(exec_time)
```

```
## Time difference of 1.414425 mins
```

```r
kcv_res
```

```
## $MSE_mat
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## [1,] 0.4574180 0.4563962 0.4558467 0.4555202 0.4553195 0.4552173 0.4551636
## [2,] 0.4775572 0.4719454 0.4680733 0.4652939 0.4632029 0.4616088 0.4603470
## [3,] 0.5052776 0.4951932 0.4882944 0.4832599 0.4794436 0.4764485 0.4740727
## [4,] 0.5066719 0.4907224 0.4817348 0.4757795 0.4715111 0.4683146 0.4658241
## [5,] 0.4965277 0.4944524 0.4931901 0.4922805 0.4915904 0.4910496 0.4905767
## [6,] 0.5295827 0.5193708 0.5126108 0.5077092 0.5039570 0.5007705 0.4984549
## [7,] 0.5789946 0.5571674 0.5431017 0.5331473 0.5256973 0.5199305 0.5153394
## [8,] 0.5928744 0.5604811 0.5432430 0.5321033 0.5241661 0.5182251 0.5135974
##           [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
## [1,] 0.4551326 0.4551244 0.4551439 0.4551701 0.4552117 0.4552667 0.4553312
## [2,] 0.4593436 0.4585294 0.4578785 0.4573330 0.4568877 0.4565140 0.4562151
## [3,] 0.4721354 0.4705406 0.4691923 0.4680766 0.4671293 0.4663148 0.4656094
## [4,] 0.4638502 0.4622501 0.4609377 0.4598513 0.4589337 0.4581659 0.4575206
## [5,] 0.4901459 0.4897800 0.4894066 0.4890946 0.4888057 0.4885112 0.4882590
## [6,] 0.4965749 0.4951471 0.4938129 0.4926566 0.4916556 0.4907766 0.4899978
## [7,] 0.5116133 0.5084970 0.5058608 0.5036371 0.5017288 0.5000214 0.4985696
## [8,] 0.5098484 0.5068146 0.5042310 0.5020782 0.5002382 0.4986037 0.4972117
##          [,15]     [,16]     [,17]     [,18]     [,19]     [,20]
## [1,] 0.4553971 0.4554801 0.4555720 0.4556727 0.4557824 0.4558967
## [2,] 0.4559722 0.4557744 0.4556154 0.4554983 0.4554167 0.4553640
## [3,] 0.4650140 0.4645015 0.4640610 0.4636583 0.4633362 0.4630628
## [4,] 0.4569723 0.4565108 0.4561205 0.4557936 0.4555215 0.4552853
## [5,] 0.4880244 0.4878030 0.4875884 0.4874057 0.4872396 0.4870744
## [6,] 0.4892984 0.4886779 0.4881303 0.4876424 0.4871869 0.4868047
## [7,] 0.4972893 0.4961368 0.4951133 0.4942073 0.4934043 0.4926708
## [8,] 0.4959866 0.4949133 0.4939401 0.4930648 0.4923132 0.4916336
##
## $SE_mat
##             [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
## [1,] 0.007079625 0.007152598 0.007182404 0.007197249 0.007219006 0.007228729
## [2,] 0.007580448 0.007445164 0.007469977 0.007536817 0.007596464 0.007659833
## [3,] 0.007791452 0.007228970 0.007070370 0.006956167 0.006863179 0.006784328
## [4,] 0.004182263 0.004424419 0.004483074 0.004519083 0.004542536 0.004557748
## [5,] 0.002900679 0.003332232 0.003653965 0.003915526 0.004127480 0.004303933
## [6,] 0.008445266 0.008389684 0.008434280 0.008478762 0.008509812 0.008502717
## [7,] 0.003709473 0.004052917 0.004304787 0.004483349 0.004669279 0.004812448
## [8,] 0.003793029 0.004073149 0.004067770 0.004051058 0.004001664 0.003983291
##             [,7]        [,8]        [,9]       [,10]       [,11]       [,12]
## [1,] 0.007237180 0.007252604 0.007256321 0.007263398 0.007270757 0.007277066
## [2,] 0.007719313 0.007776541 0.007827171 0.007877782 0.007925712 0.007968782
## [3,] 0.006708197 0.006649344 0.006587940 0.006540831 0.006493205 0.006449711
## [4,] 0.004568805 0.004579479 0.004578978 0.004585142 0.004582756 0.004584865
```

```
## [5,] 0.004458022 0.004599596 0.004717829 0.004833640 0.004932592 0.005026773
## [6,] 0.008538614 0.008571694 0.008631409 0.008658689 0.008693197 0.008715447
## [7,] 0.004939714 0.005046498 0.005156473 0.005267013 0.005347081 0.005428778
## [8,] 0.003991826 0.003995687 0.004019768 0.004034492 0.004072125 0.004105859
##           [,13]       [,14]       [,15]       [,16]       [,17]       [,18]
## [1,] 0.007284213 0.007291364 0.007304484 0.007311313 0.007318143 0.007324963
## [2,] 0.008014654 0.008055343 0.008095148 0.008128883 0.008167724 0.008203463
## [3,] 0.006405584 0.006374389 0.006340905 0.006310179 0.006281714 0.006256063
## [4,] 0.004577630 0.004570991 0.004568908 0.004560028 0.004550408 0.004540187
## [5,] 0.005120913 0.005201923 0.005278794 0.005351852 0.005425316 0.005492460
## [6,] 0.008740143 0.008763632 0.008781022 0.008802664 0.008818752 0.008837827
## [7,] 0.005519196 0.005593505 0.005665125 0.005748712 0.005815601 0.005876507
## [8,] 0.004129882 0.004173095 0.004217686 0.004257287 0.004294386 0.004338080
##           [,19]       [,20]
## [1,] 0.007331758 0.007335321
## [2,] 0.008236304 0.008268152
## [3,] 0.006231805 0.006209379
## [4,] 0.004529455 0.004521022
## [5,] 0.005557137 0.005622322
## [6,] 0.008852143 0.008865307
## [7,] 0.005939041 0.006012995
## [8,] 0.004379894 0.004425899
##
## $opt_para
##      [,1] [,2] [,3]
## [1,]    3    2    9
```

```
res <- mglram(tnsr = synthetic_dat@data, ranks = c(3, 2), init=0, D = D2,
       lambda = 9, max_iter = 500, tol = 1e-5, L0 = NULL)

res$conv
```

```
## [1] TRUE
```

```
tilde <- MakeIdent(L=res$L, G=res$G, R=res$R)

L_tilde <- tilde$L_tilde
R_tilde <- tilde$R_tilde
G_tilde <- tilde$G_tilde

L_tilde
```

```
##             [,1]        [,2]         [,3]
##  [1,] -0.2284550  0.19490268 -0.123075378
##  [2,] -0.2252760  0.22023097 -0.052538782
##  [3,] -0.2213523  0.23471693  0.013567028
##  [4,] -0.2172047  0.23933754  0.072049630
##  [5,] -0.2132657  0.23285233  0.122500721
##  [6,] -0.2093088  0.21374184  0.165663650
##  [7,] -0.2048836  0.17901423  0.202433012
##  [8,] -0.1992570  0.12803180  0.233613855
##  [9,] -0.1919382  0.06136624  0.258437758
## [10,] -0.1840533 -0.01669524  0.274947798
```

```
## [11,] -0.1769982 -0.09925009  0.277988773
## [12,] -0.1722254 -0.17926375  0.262254311
## [13,] -0.1699296 -0.24962487  0.224716510
## [14,] -0.1706224 -0.30130781  0.166633827
## [15,] -0.1749215 -0.32884452  0.091067448
## [16,] -0.1816160 -0.32885157  0.003158485
## [17,] -0.1900241 -0.30156835 -0.089135701
## [18,] -0.1994224 -0.25014511 -0.176086801
## [19,] -0.2091245 -0.18251772 -0.246781599
## [20,] -0.2180782 -0.10645109 -0.291464379
## [21,] -0.2250422 -0.02897456 -0.306019934
## [22,] -0.2291644  0.04411753 -0.291615530
## [23,] -0.2307116  0.10734692 -0.251627219
## [24,] -0.2303858  0.15783298 -0.192078077
```

```
R_tilde
```

```
##            [,1]       [,2]
## [1,] -0.6271955  0.2553330
## [2,] -0.5602656  0.5083655
## [3,] -0.5410437 -0.8224169
```

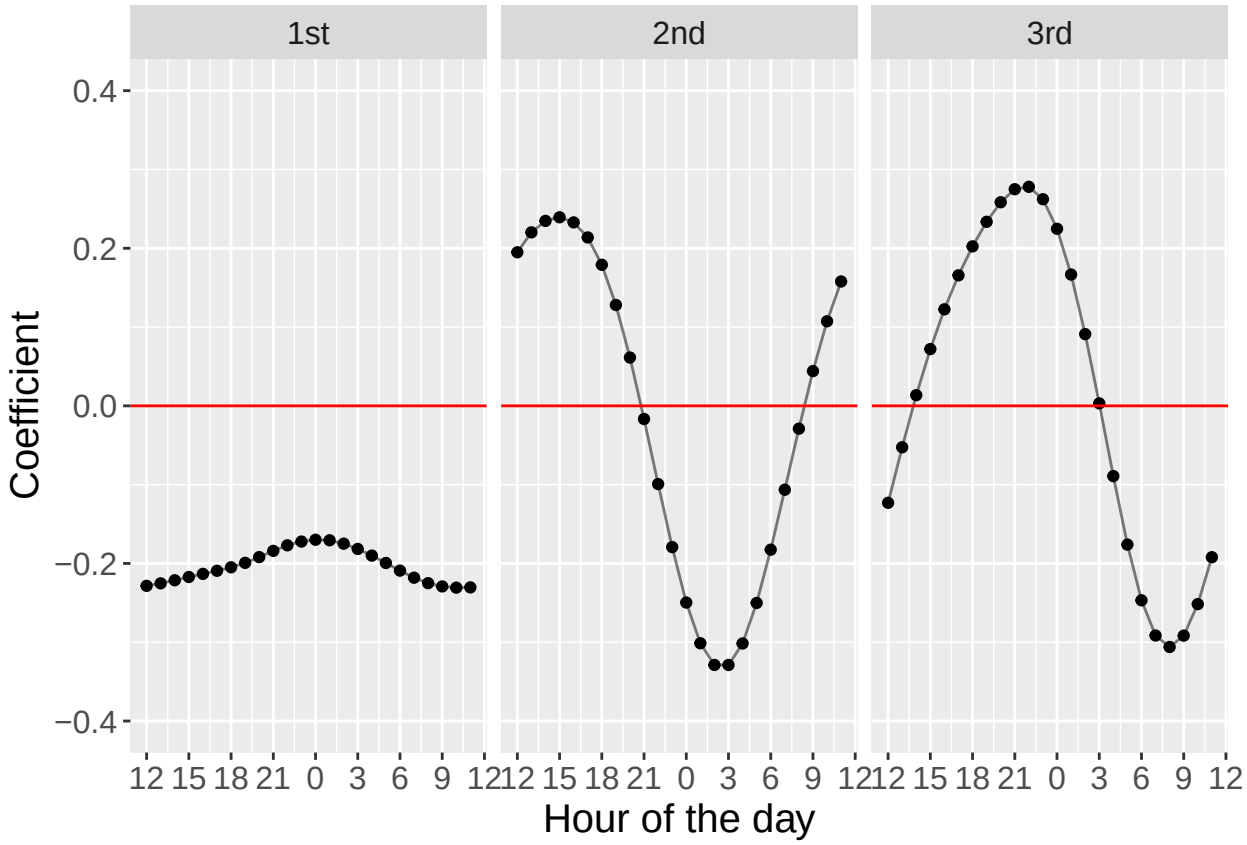
```
G_tilde[ , ,1:10]
```

```
## , , 1
##
##           [,1]       [,2]
## [1,] -1.224538 -0.2936235
## [2,] -5.477462 -1.5609323
## [3,]  1.156635 -0.2345109
##
## , , 2
##
##            [,1]       [,2]
## [1,]  2.7305815  7.9176982
## [2,] -0.4715415 -0.8113450
## [3,]  2.6630224 -0.1505596
##
## , , 3
##
##           [,1]       [,2]
## [1,]  2.557276 -0.4902649
## [2,] -1.853480  2.6625636
## [3,] -1.401344  0.6274245
##
## , , 4
##
##            [,1]      [,2]
## [1,] -5.5118146 -5.355439
## [2,] -4.3004987  3.048665
## [3,] -0.4820475 -1.534459
##
## , , 5
```

```
##
##            [,1]      [,2]
## [1,]  0.2530558  2.110150
## [2,] -6.1821776  2.187105
## [3,] -0.8232639 -0.523512
##
## , , 6
##
##            [,1]      [,2]
## [1,] -10.276152 -4.171788
## [2,]  -4.433035  1.236427
## [3,]  -3.742673 -1.453785
##
## , , 7
##
##           [,1]       [,2]
## [1,]  4.207740  0.9310597
## [2,] -1.154908  0.2251016
## [3,]  1.194034 -0.1306769
##
## , , 8
##
##            [,1]      [,2]
## [1,] -9.4138593  5.425209
## [2,] -0.5446551 -1.456186
## [3,] -0.3344780  1.264845
##
## , , 9
##
##           [,1]      [,2]
## [1,] -3.633431 2.2456202
## [2,] -1.659963 0.9721529
## [3,]  1.837248 1.7961851
##
## , , 10
##
##           [,1]       [,2]
## [1,] -1.130647  4.7265850
## [2,] -1.894931  0.7981547
## [3,] -3.022278 -0.2726418
```

```
dataL = data.frame(loading = c(L_tilde[, 1], L_tilde[, 2], L_tilde[, 3]),
                   component = c(rep("1st", 24), rep("2nd", 24), rep("3rd", 24)),
                   hour = rep(12:35, 3))

pL = dataL %>%
  ggplot(aes(x = hour, y = loading)) + geom_line(alpha = 0.5) + geom_point() + geom_hline(yintercept = 0
  scale_x_continuous(breaks = c(12, 15, 18, 21, 24, 27, 30, 33, 36), labels = c(12, 15, 18, 21, 0, 3, 6
  xlab("Hour of the day") +
  facet_grid(~ component) +
  ylab("Coefficient") +
  theme(text = element_text(size = 15)) +
  scale_y_continuous(limits = c(-0.4, 0.4))
```

```
pL
```




```
R_tilde[, 1] # first component of R tilde
```

```
## [1] -0.6271955 -0.5602656 -0.5410437
```

```
R_tilde[, 2] # second component of R tilde
```

```
## [1]  0.2553330  0.5083655 -0.8224169
```

```
## 37/100 DBP + 33/50 SBP + 29/100 HR (Seem to be joint effect of the three)
R_tilde[1, 1]/sum(R_tilde[, 1]) # DBP: 37/100
```

```
## [1] 0.3628543
```

```
R_tilde[2, 1]/sum(R_tilde[, 1]) # SBP: 33/100
```

```
## [1] 0.3241331
```

```
R_tilde[3, 1]/sum(R_tilde[, 1]) # HR: 30/100 = 3/10
```

```
## [1] 0.3130125
```

```r
## -13/100 DBP - 33/100 SBP + 27/50 HR (Contrast between blood pressure and heart rate?)
R_tilde[1, 2]/sum(abs(R_tilde[, 2])) # DBP: -13/100
```

```
## [1] 0.1609801
```

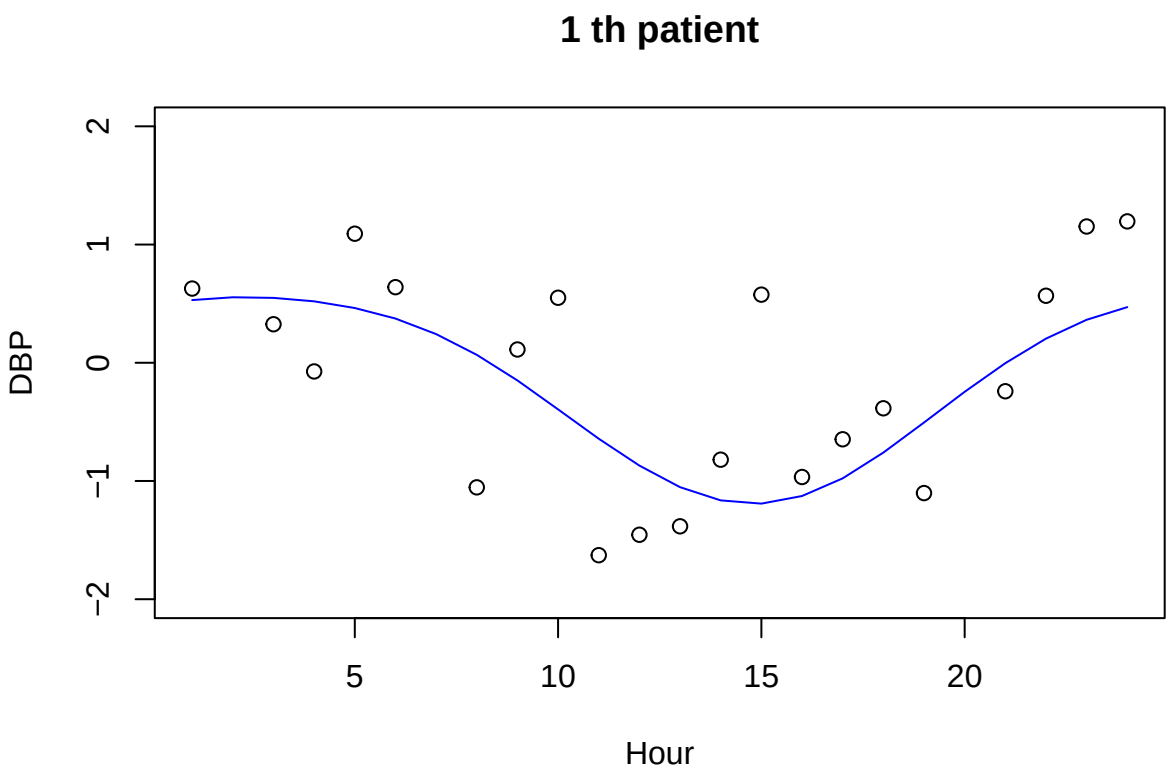
```r
R_tilde[2, 2]/sum(abs(R_tilde[, 2])) # SBP: -33/100
```

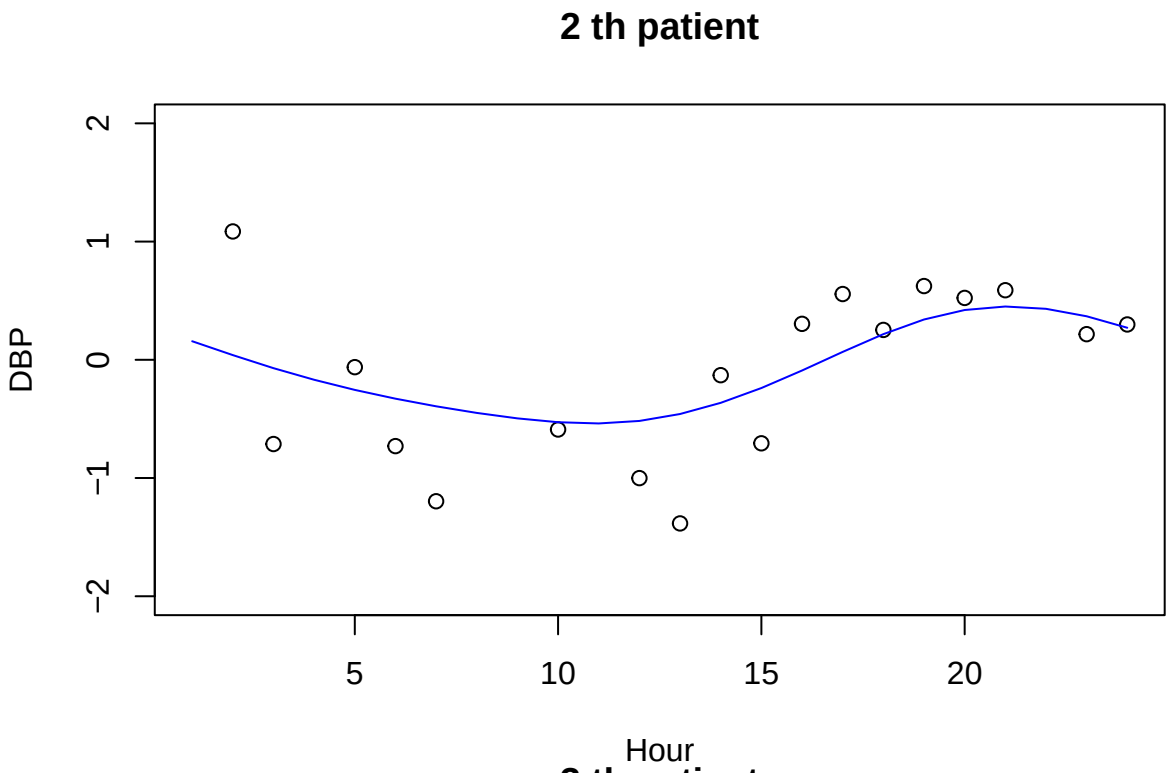
```
## [1] 0.3205098
```

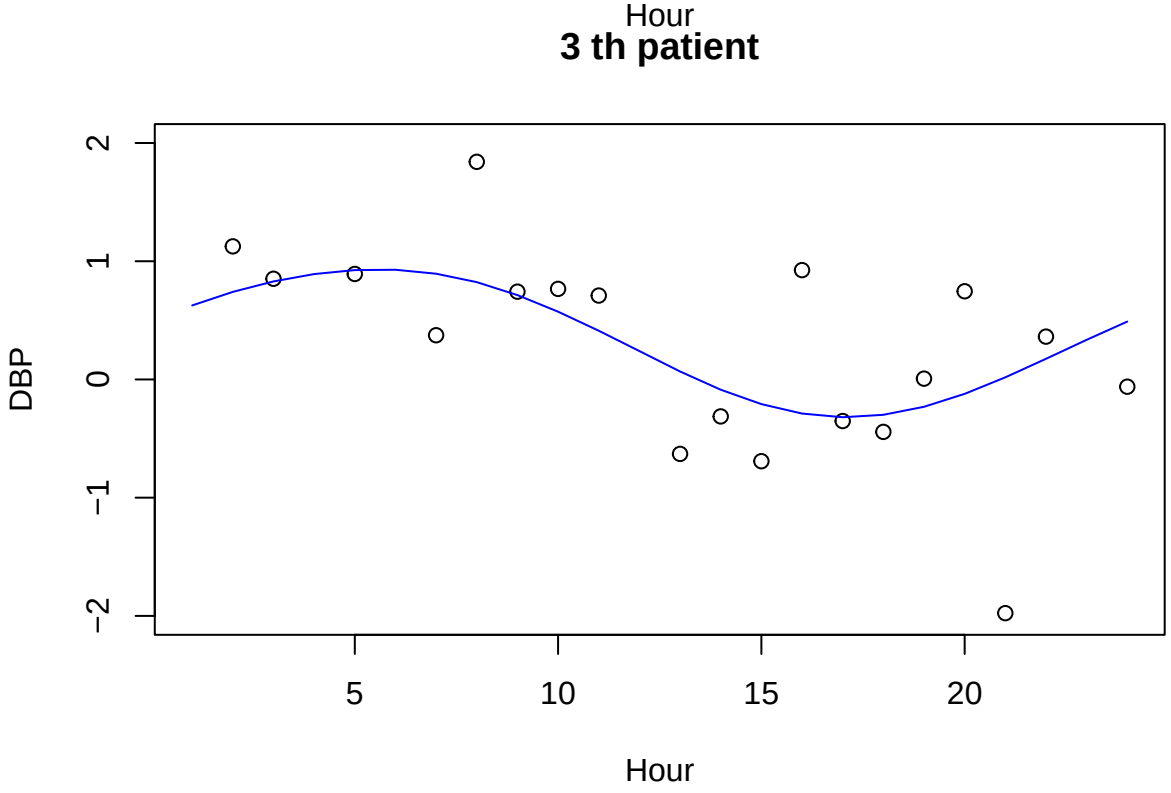
```r
R_tilde[3, 2]/sum(abs(R_tilde[, 2])) # HR: 54/100 = 27/50
```

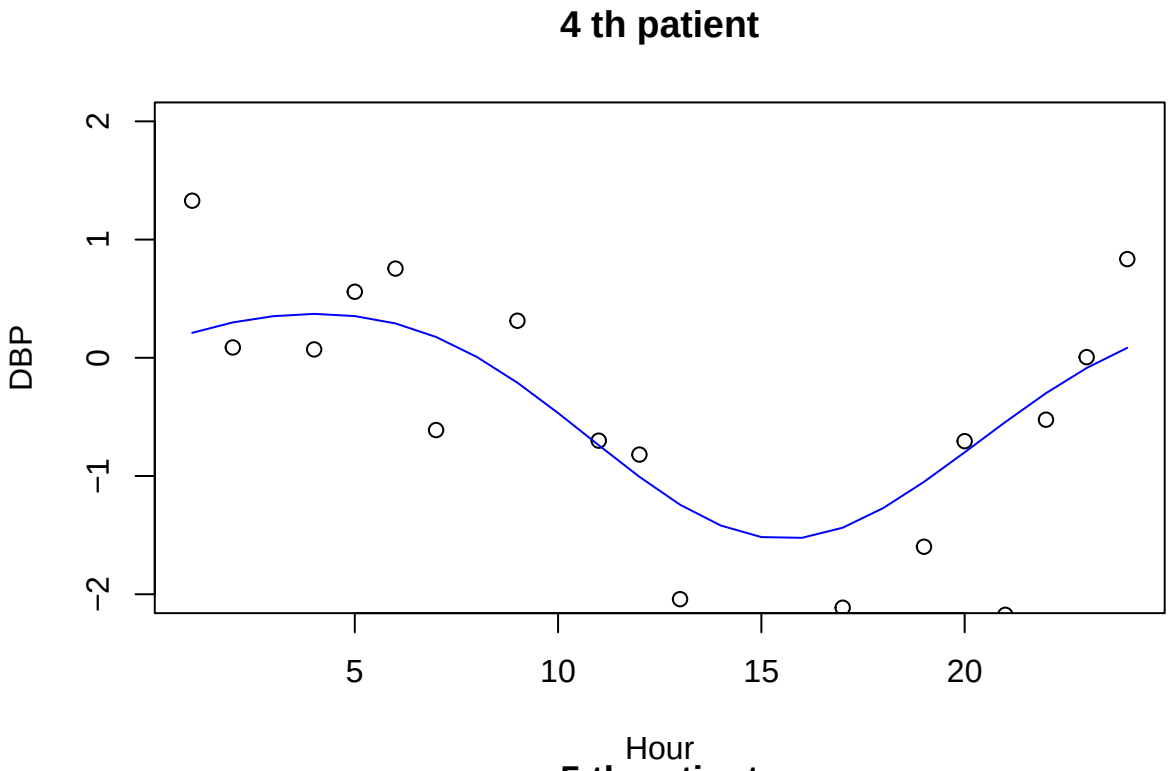
```
## [1] -0.5185101
```

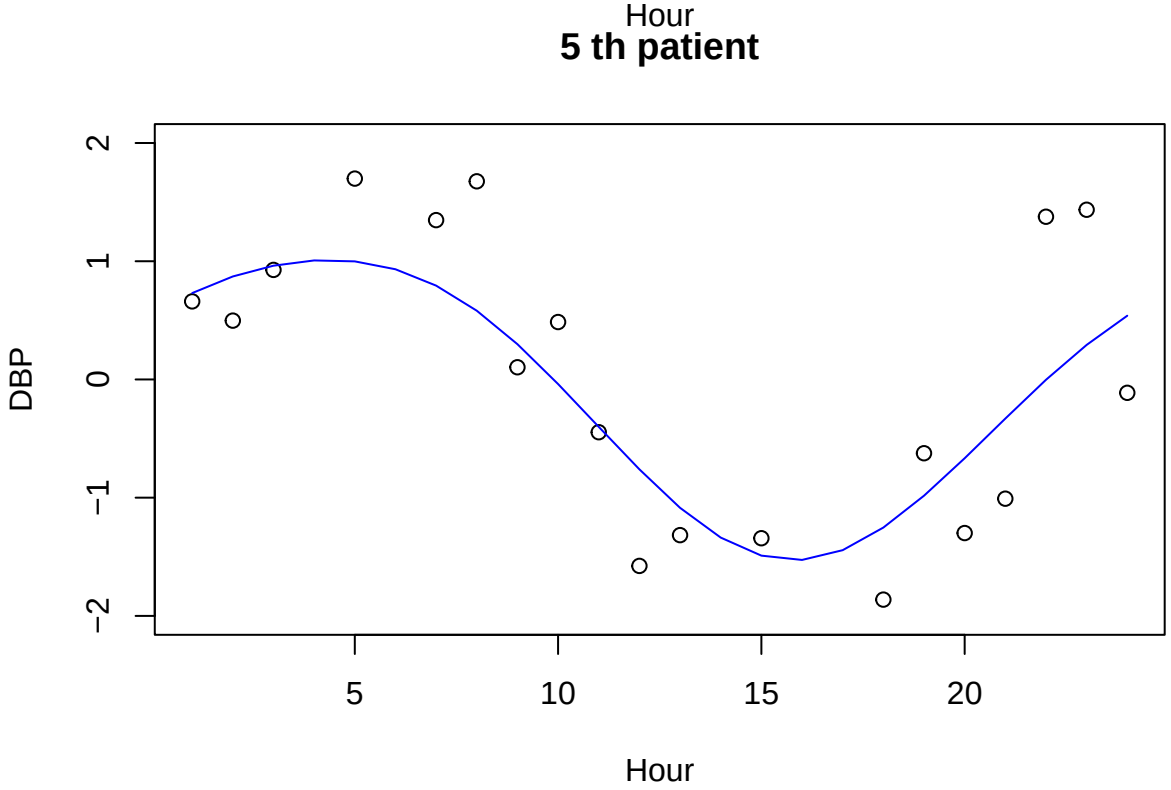
```r
for (i in 1:25){
  plot(synthetic_dat@data[ , 1, i], main=paste(i,"th patient"), xlab="Hour", ylab="DBP", ylim=c(-2,2))
  lines(res$est[ , 1, i], col="blue") # estimated curve by our algorithm
}
```

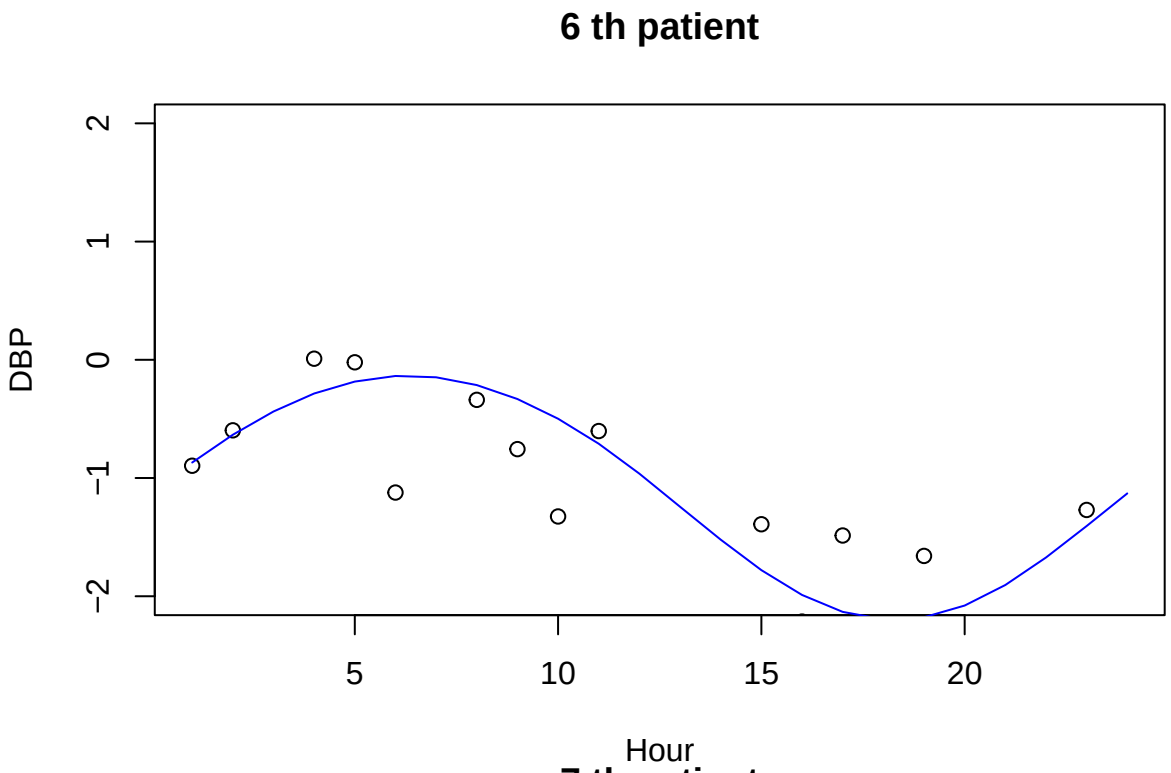


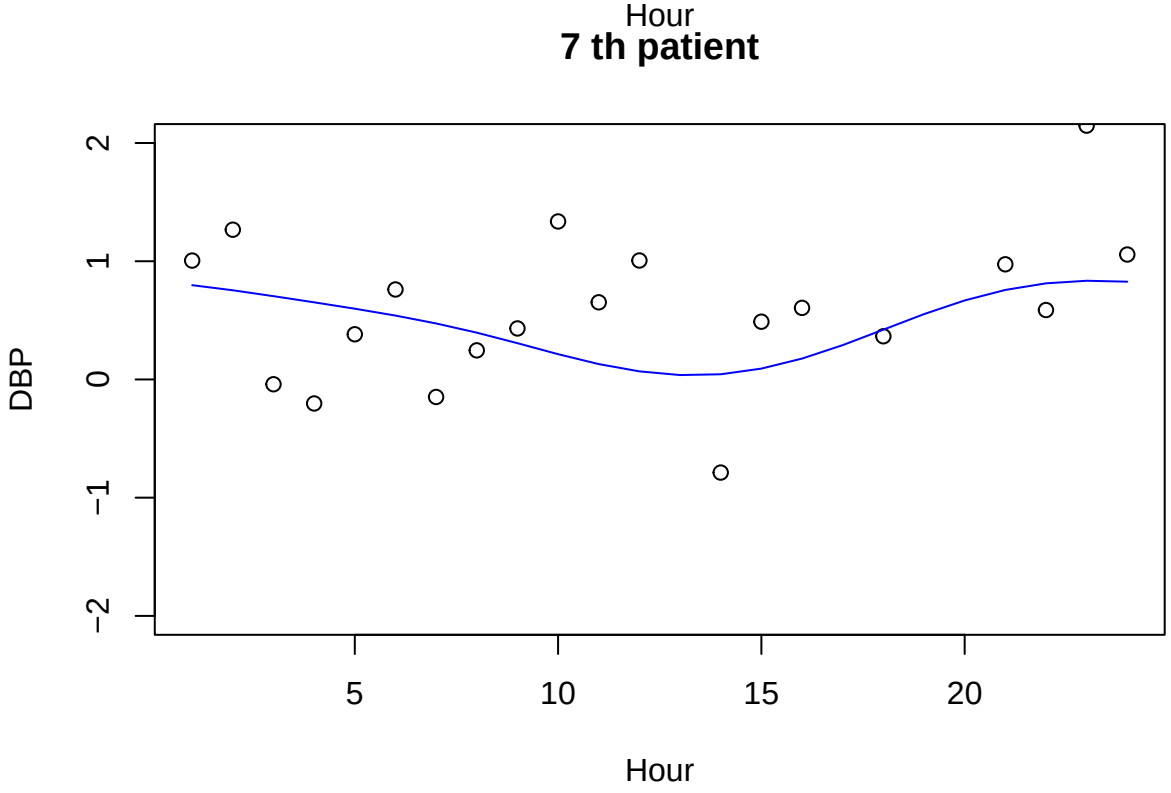

2 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

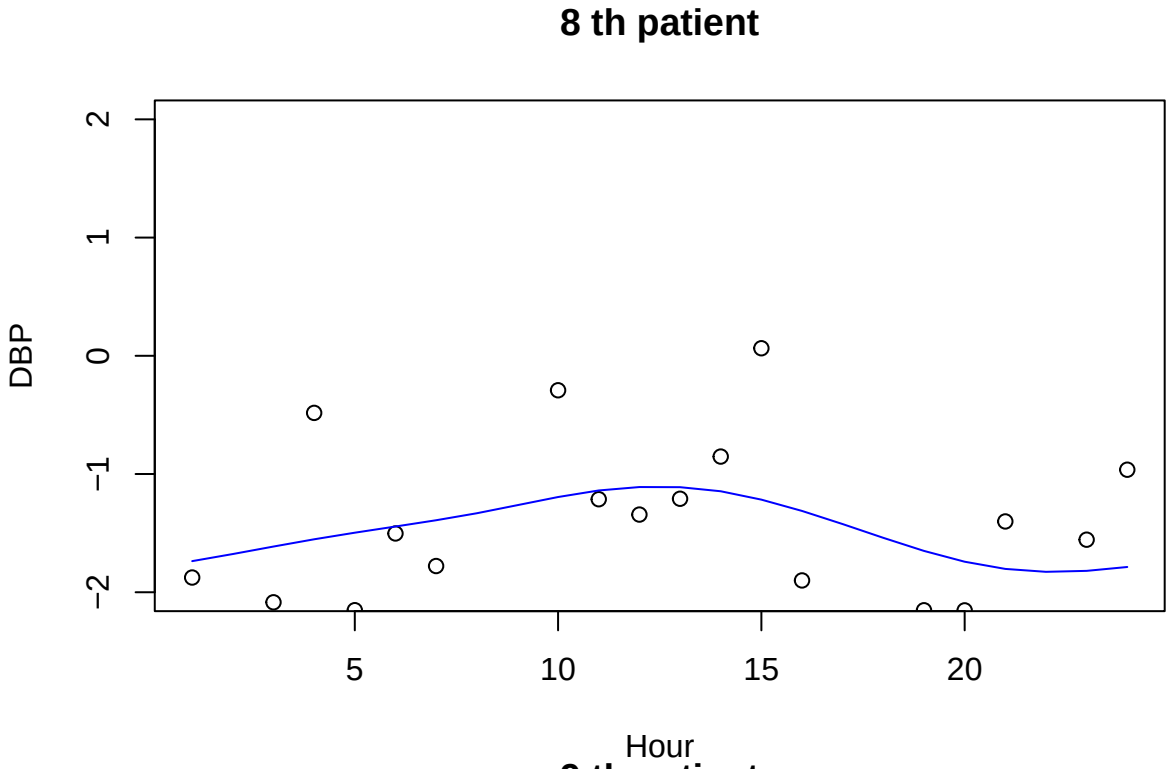

3 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

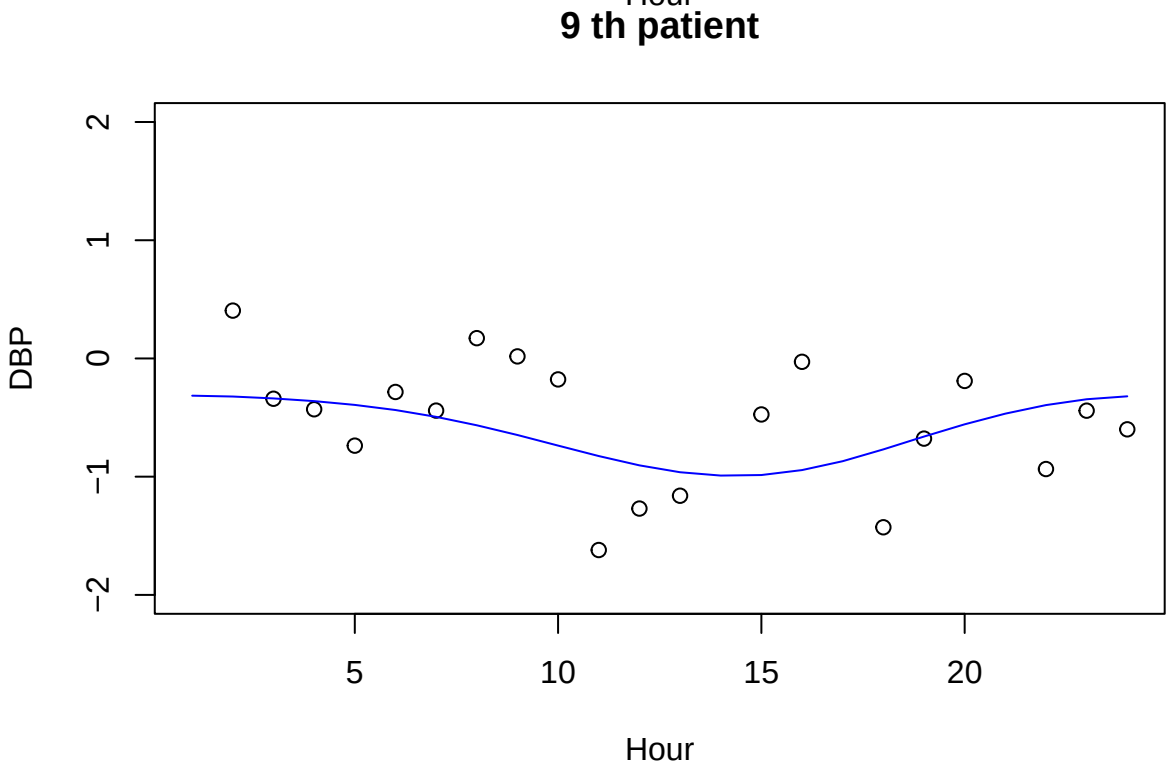

4 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

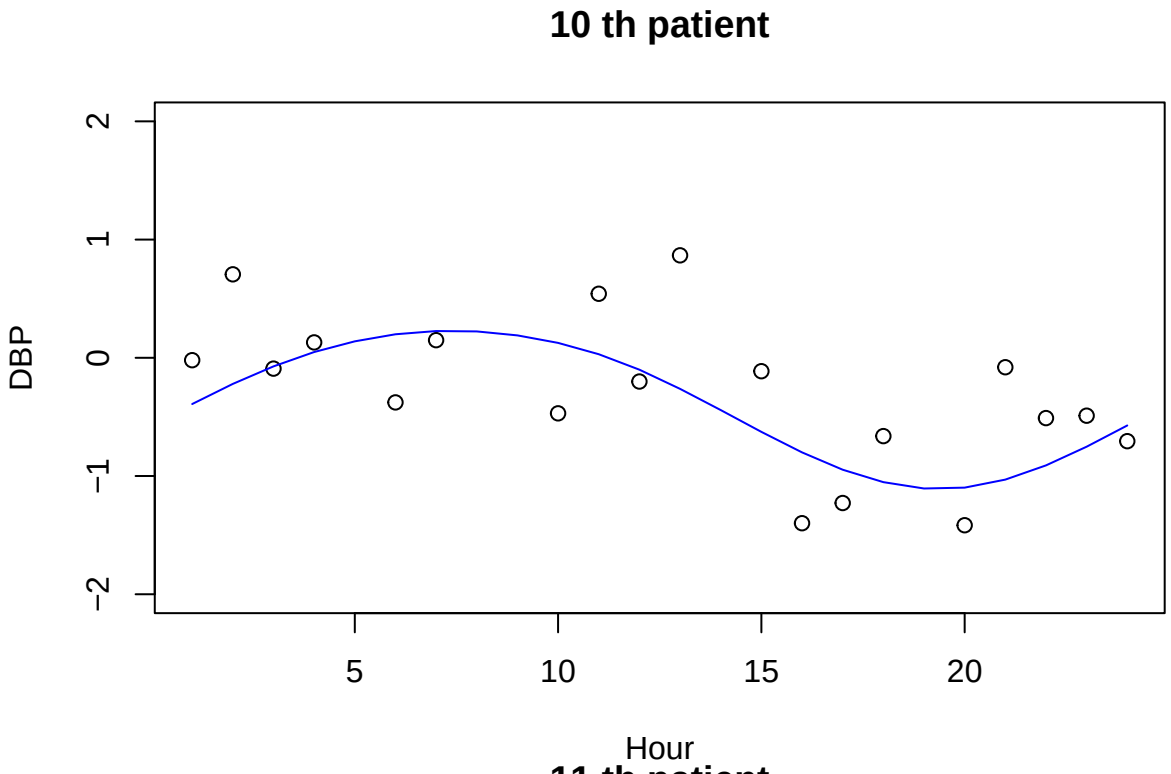

5 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

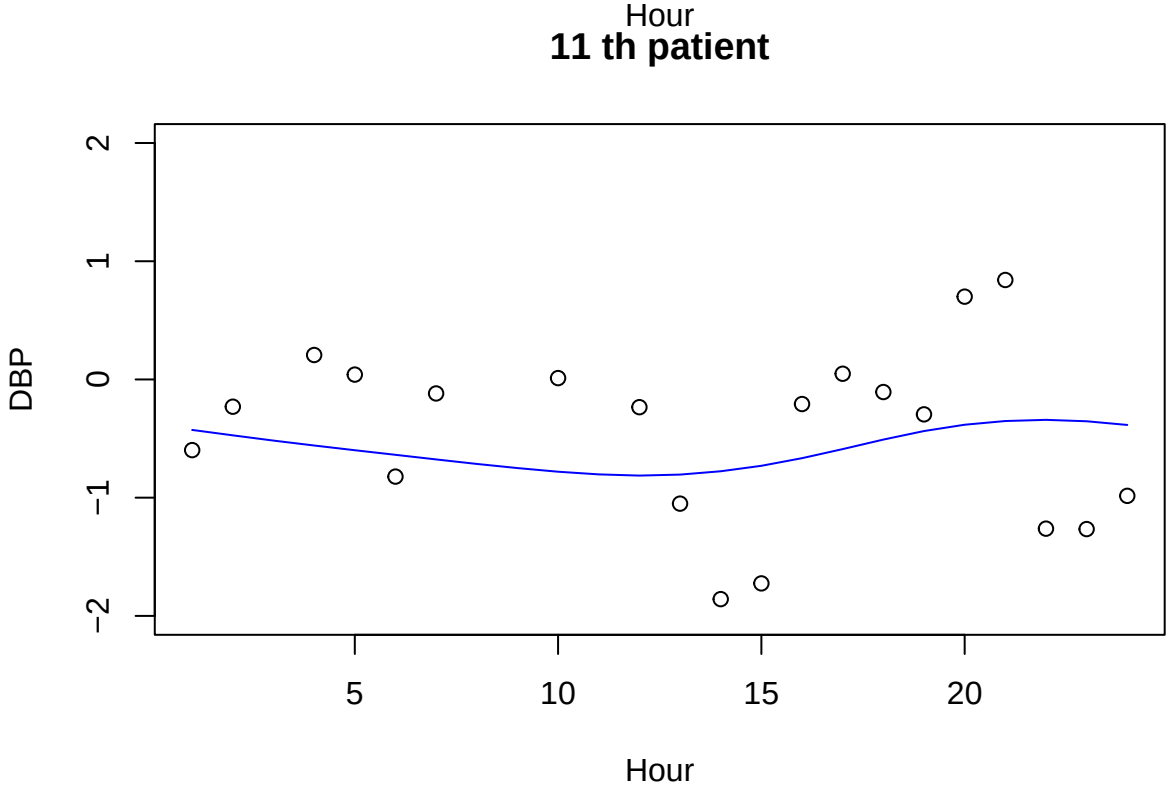

6 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

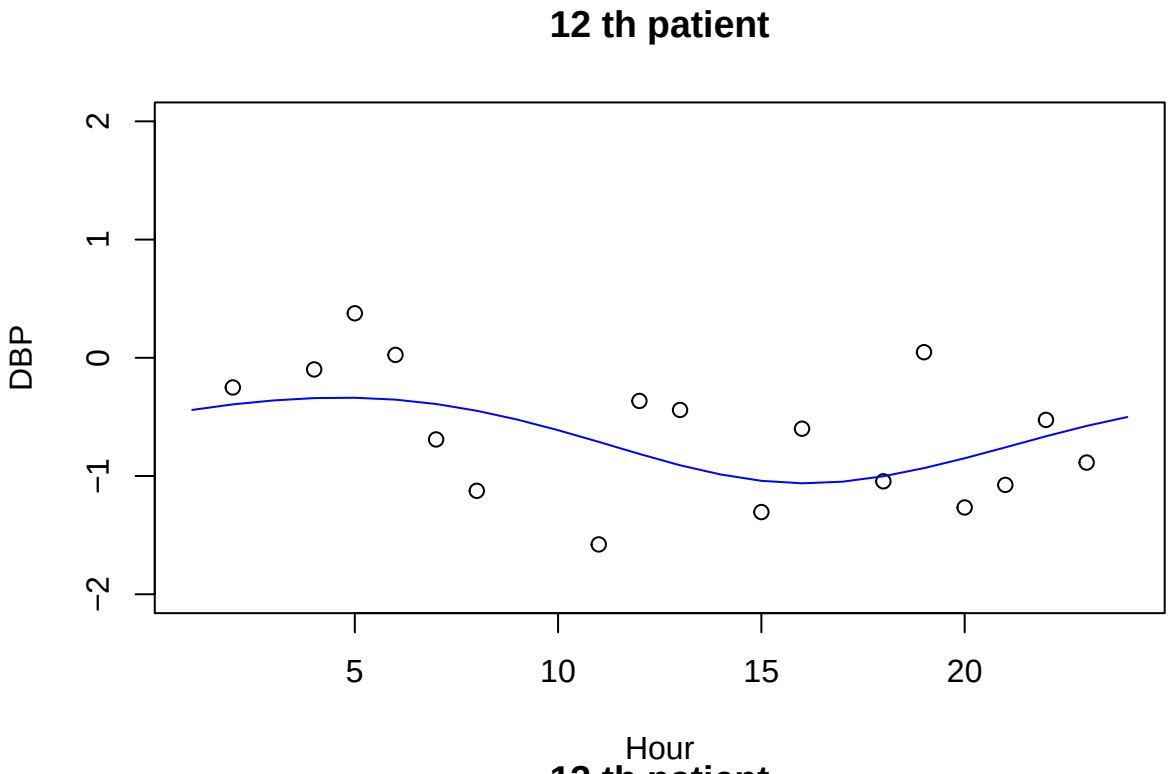

7 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

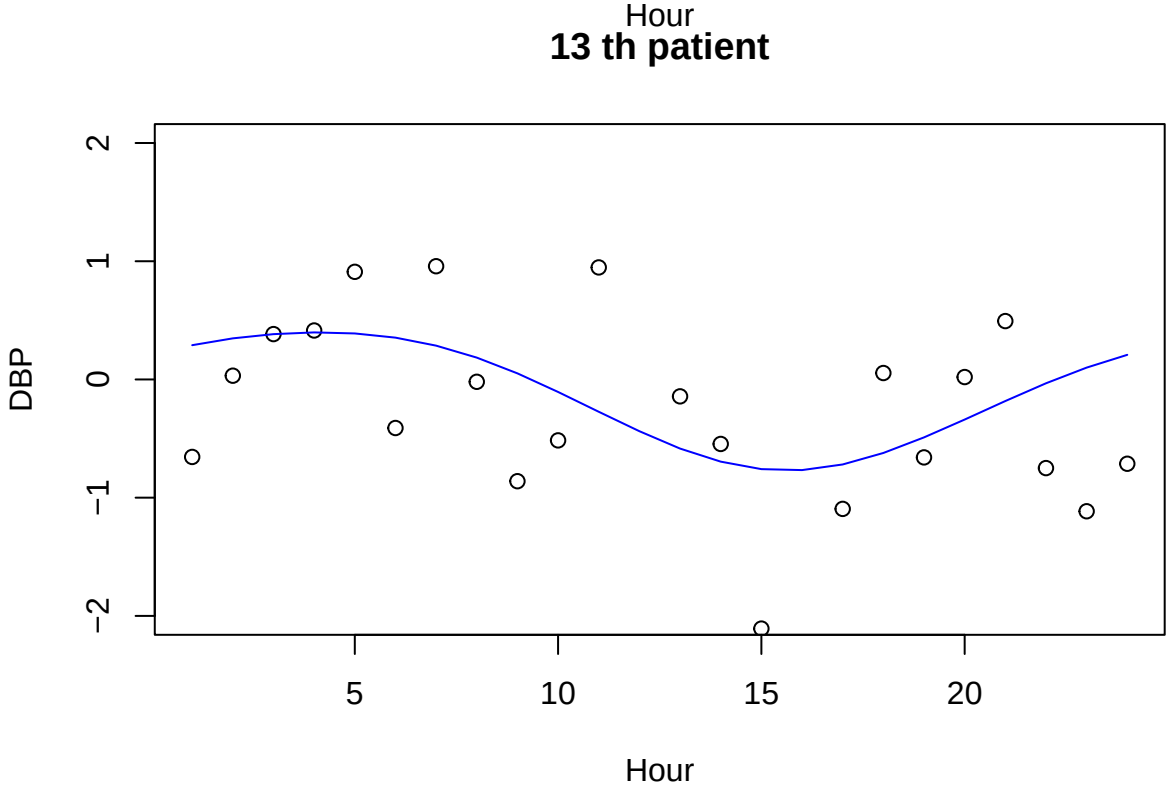

8 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

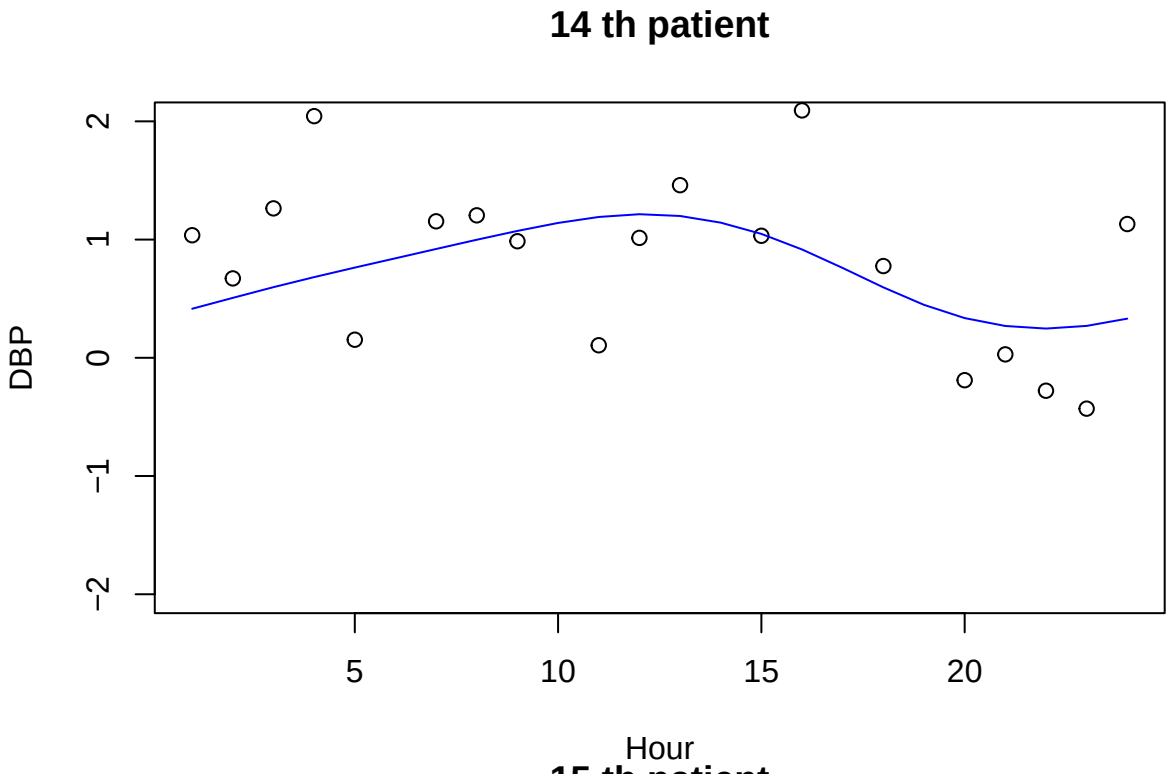

9 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

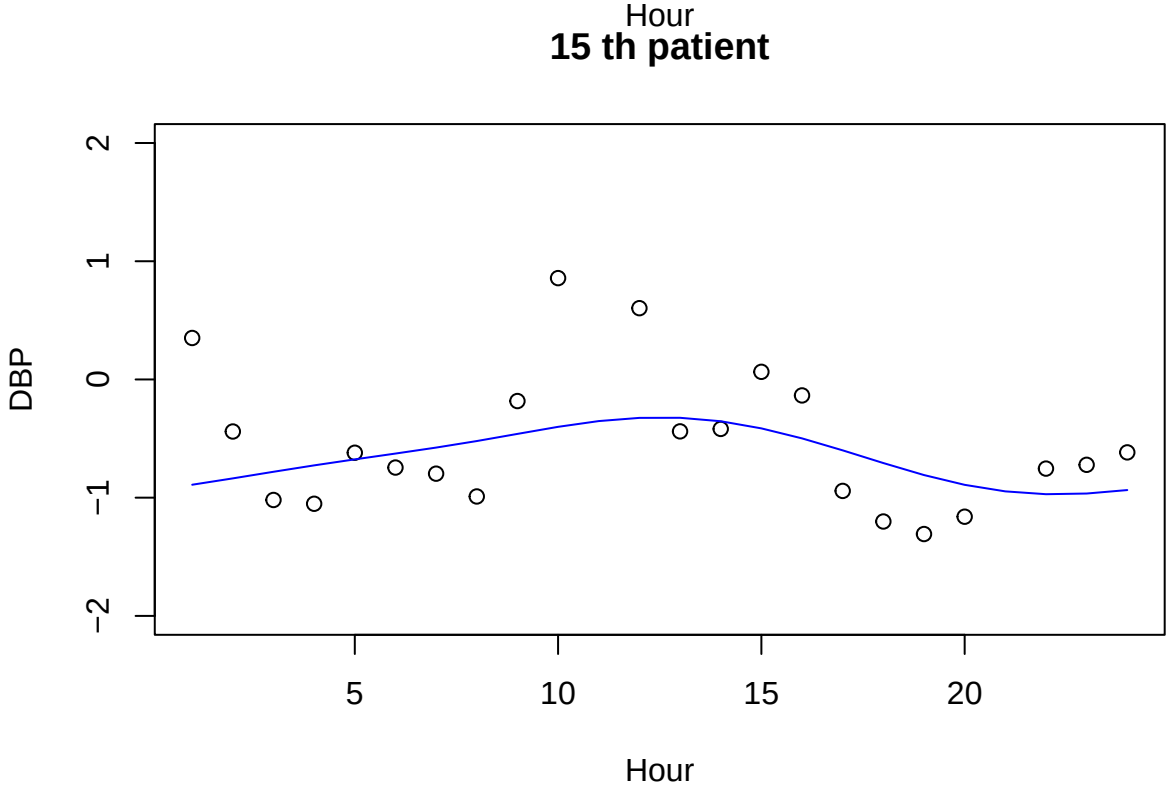

10 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

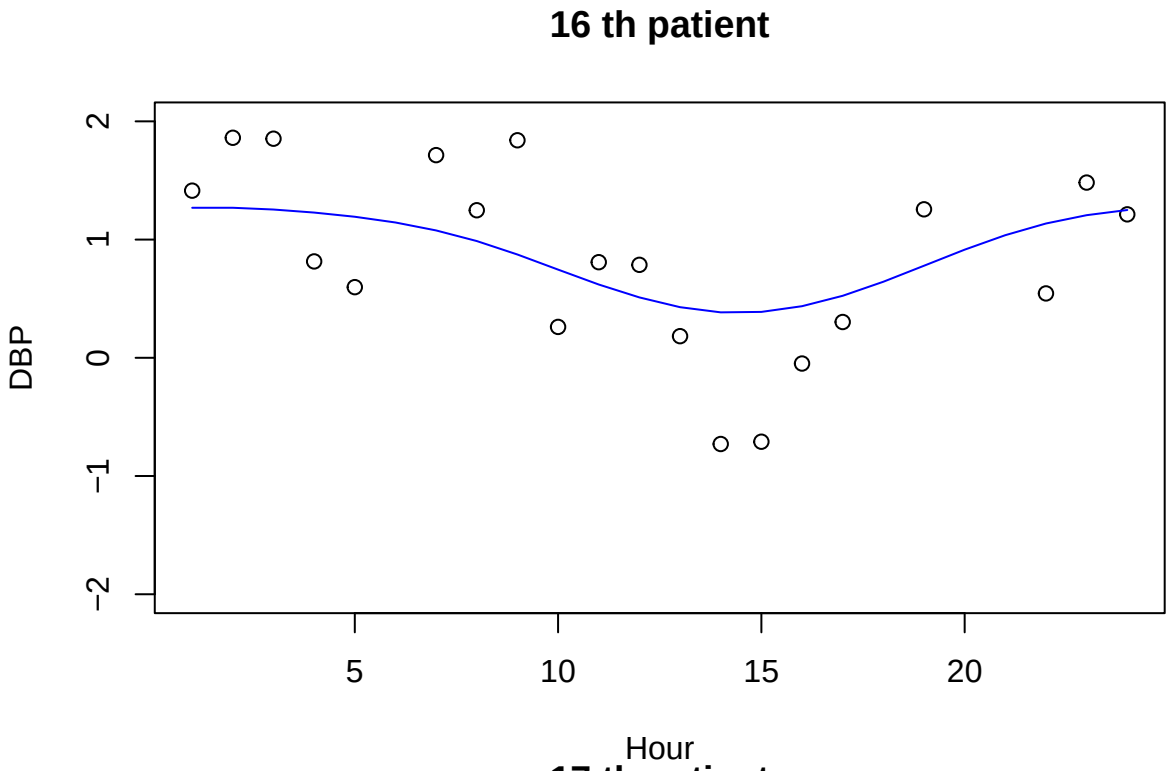

11 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

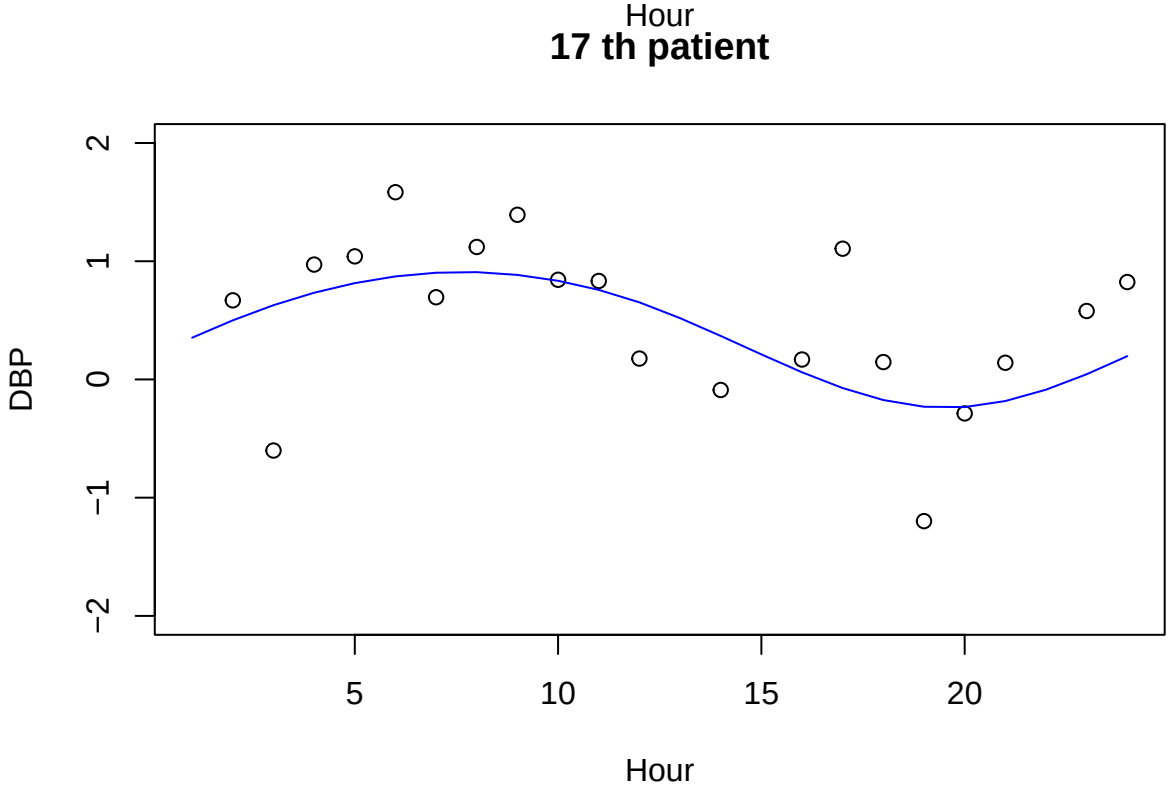

12 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

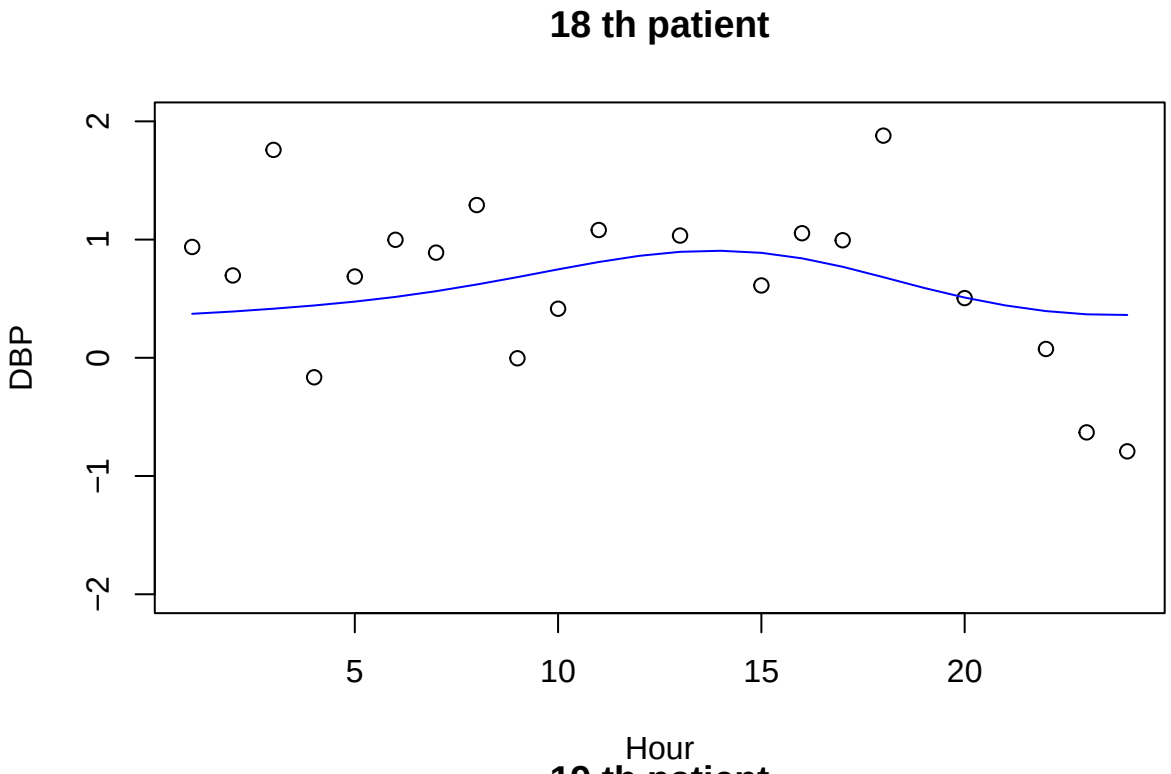

13 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

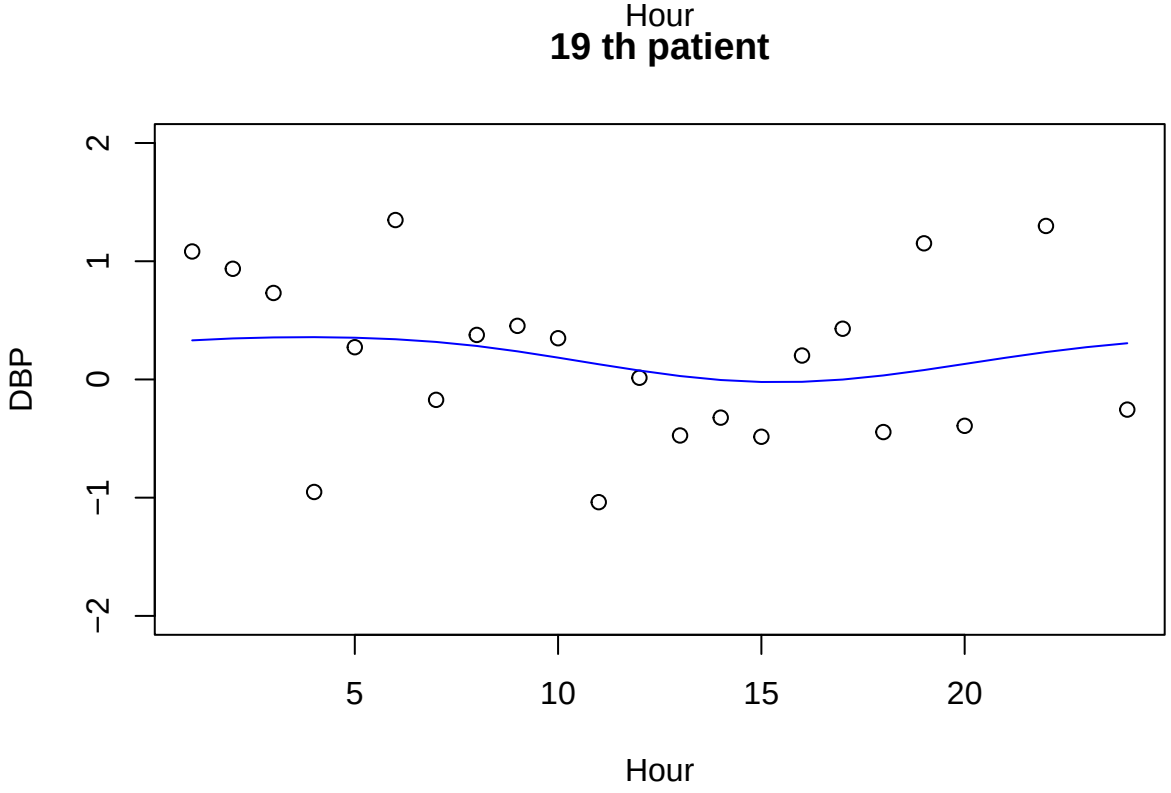

14 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

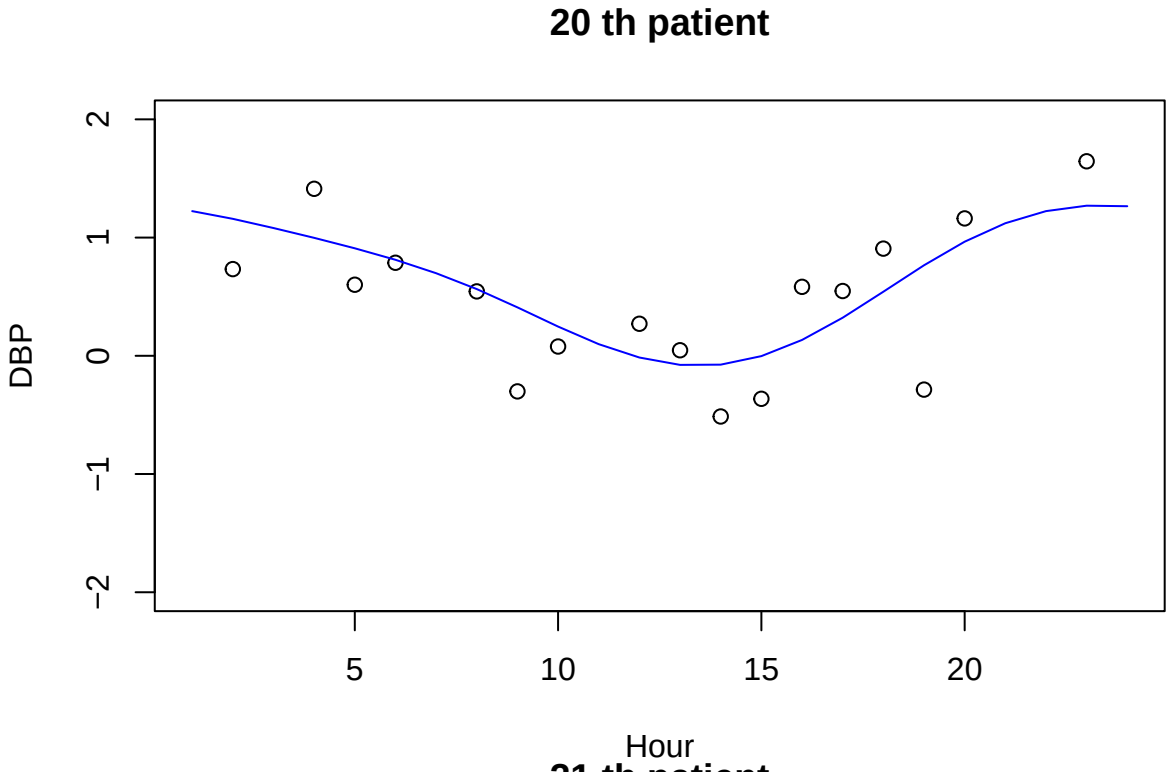

15 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

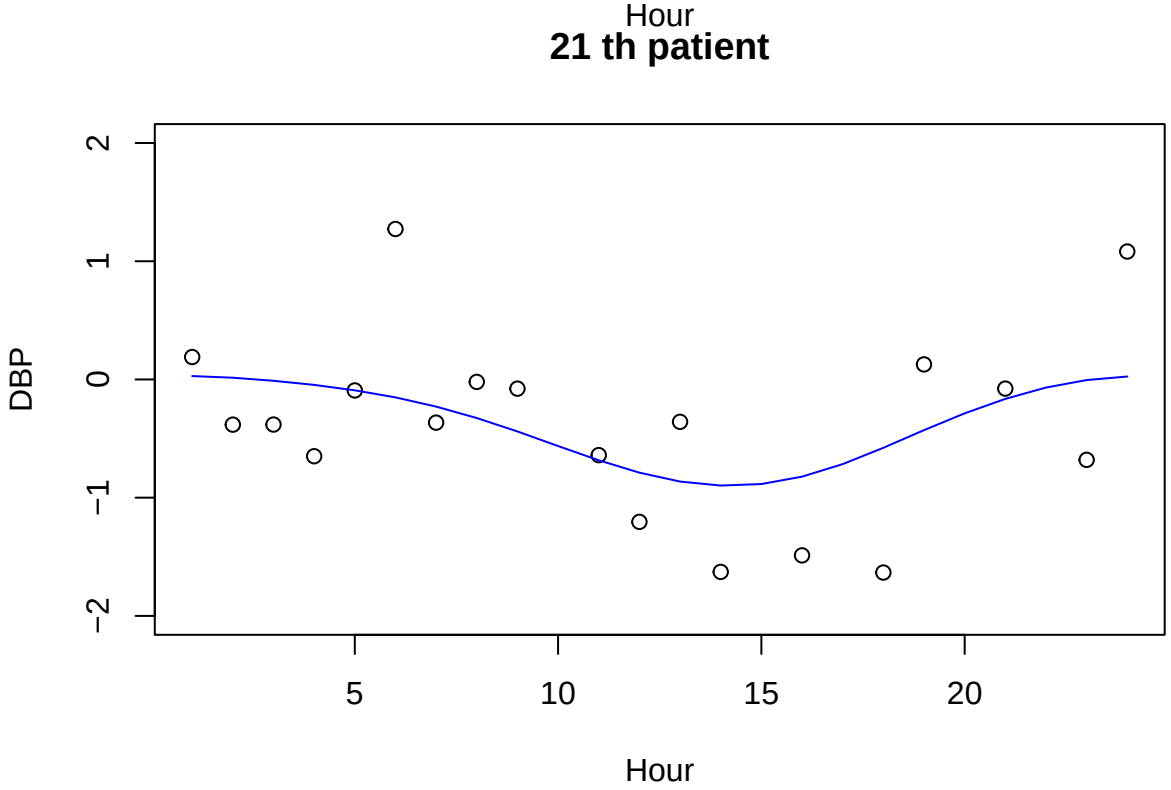

16 th patient
DBP
Hour
2
1
0
−1
−2
5
10
15
20

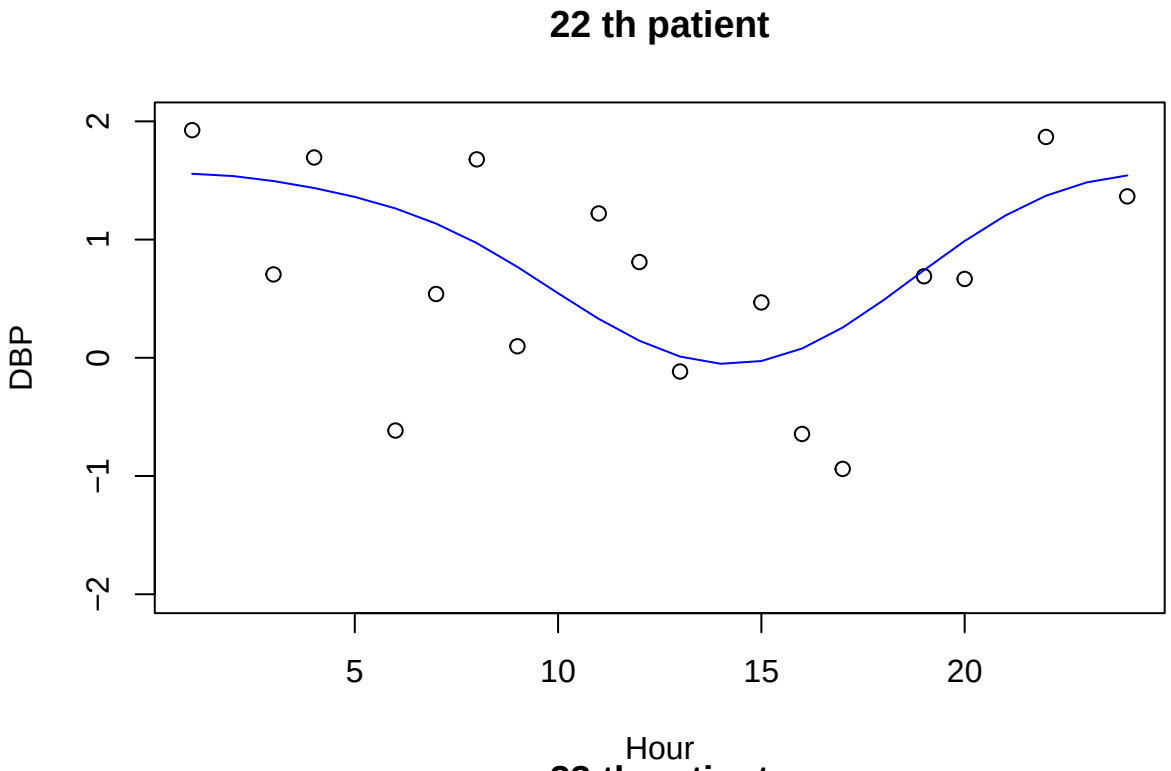

17 th patient
DBP
Hour
2
1
0
−1
−2
5
10
15
20

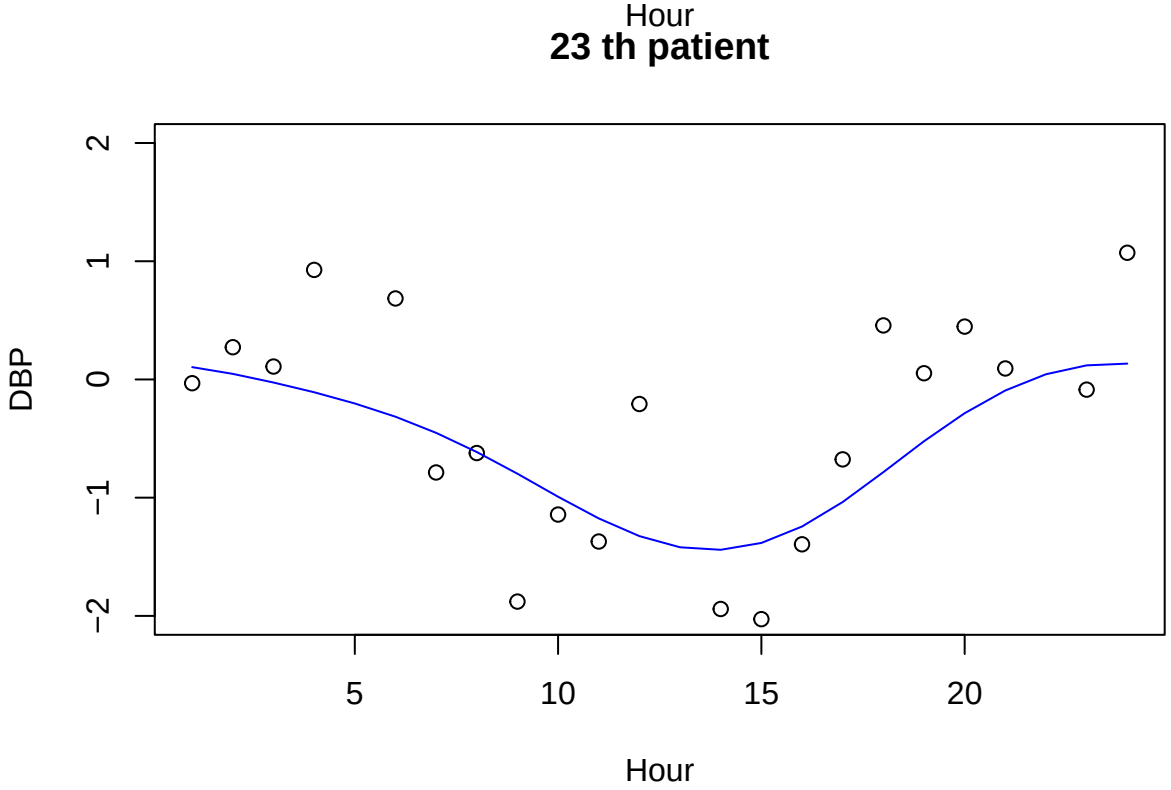

18 th patient
DBP
Hour
2
1
0
−1
−2
5
10
15
20

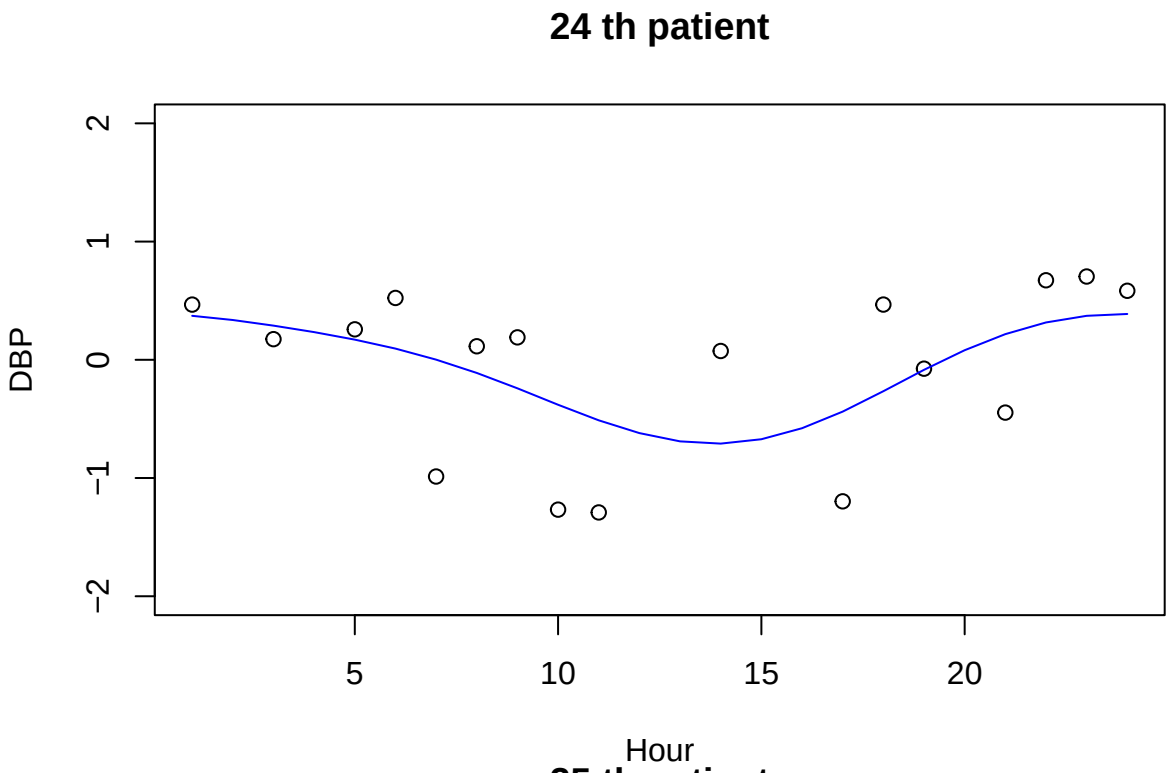

19 th patient
DBP
Hour
2
1
0
−1
−2
5
10
15
20

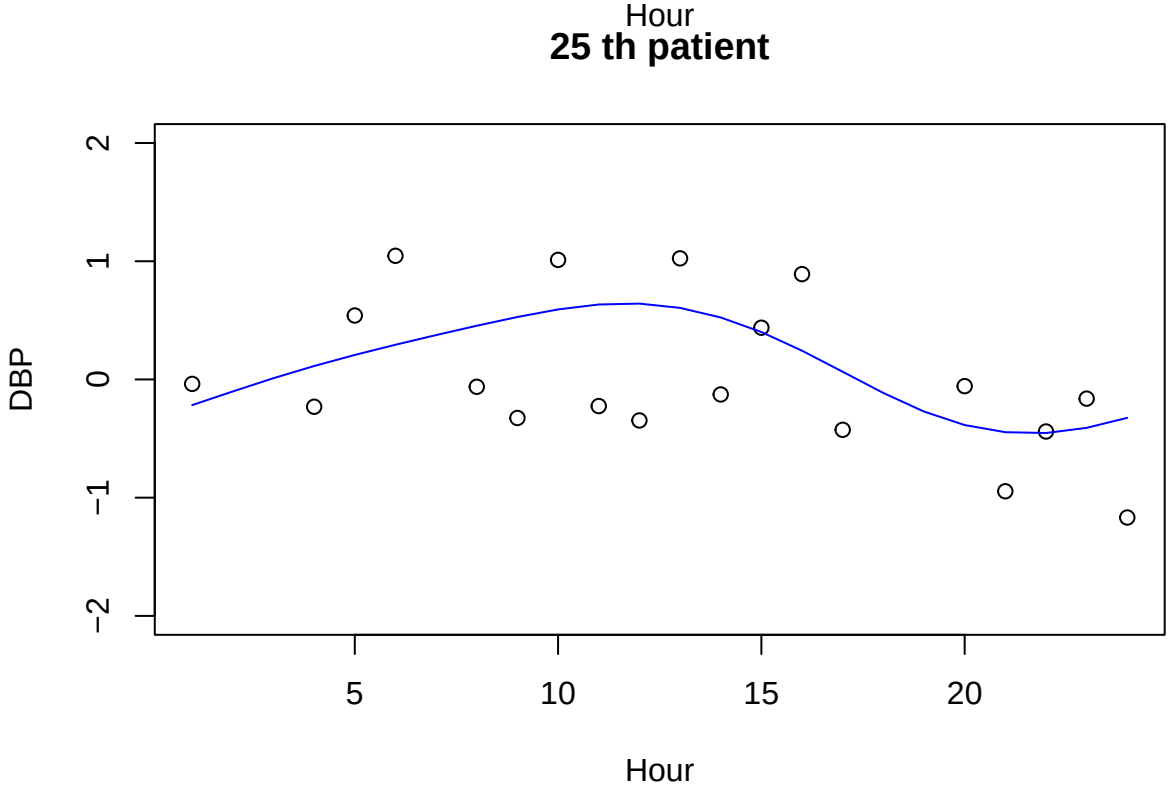

20 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour


21 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

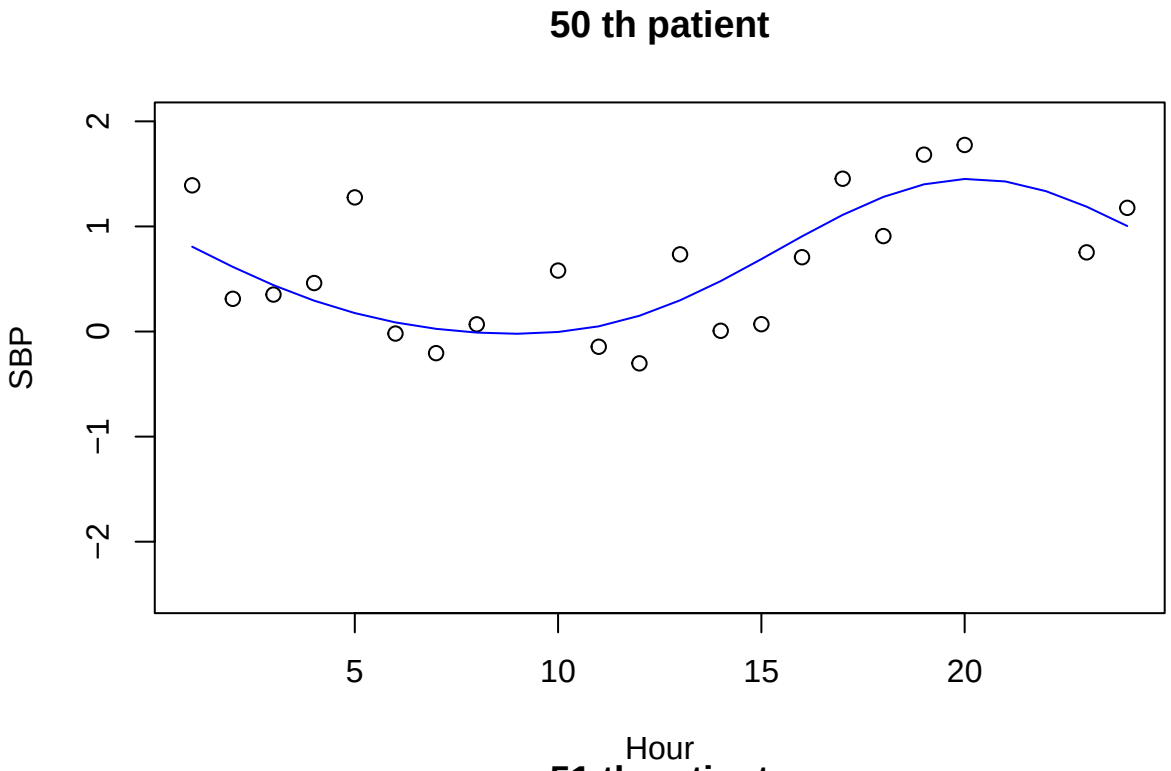

22 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

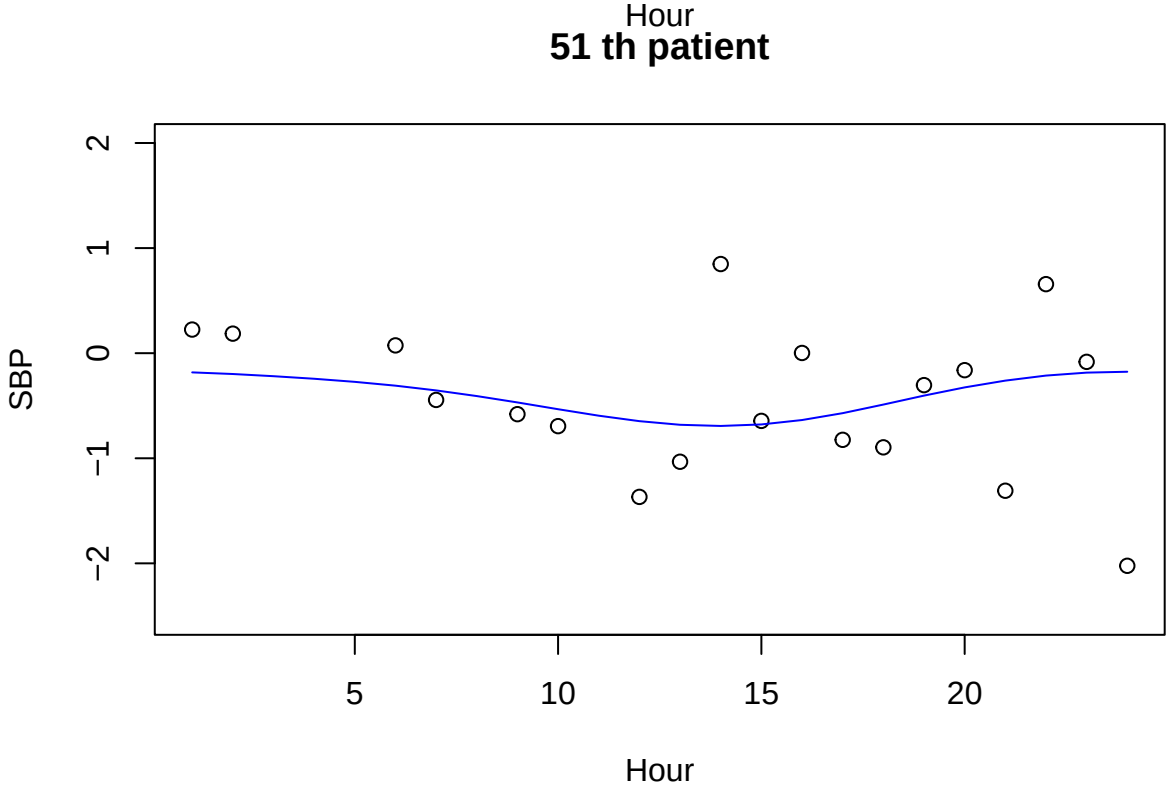

23 th patient
DBP
2
1
0
−1
−2
5
10
15
20
Hour

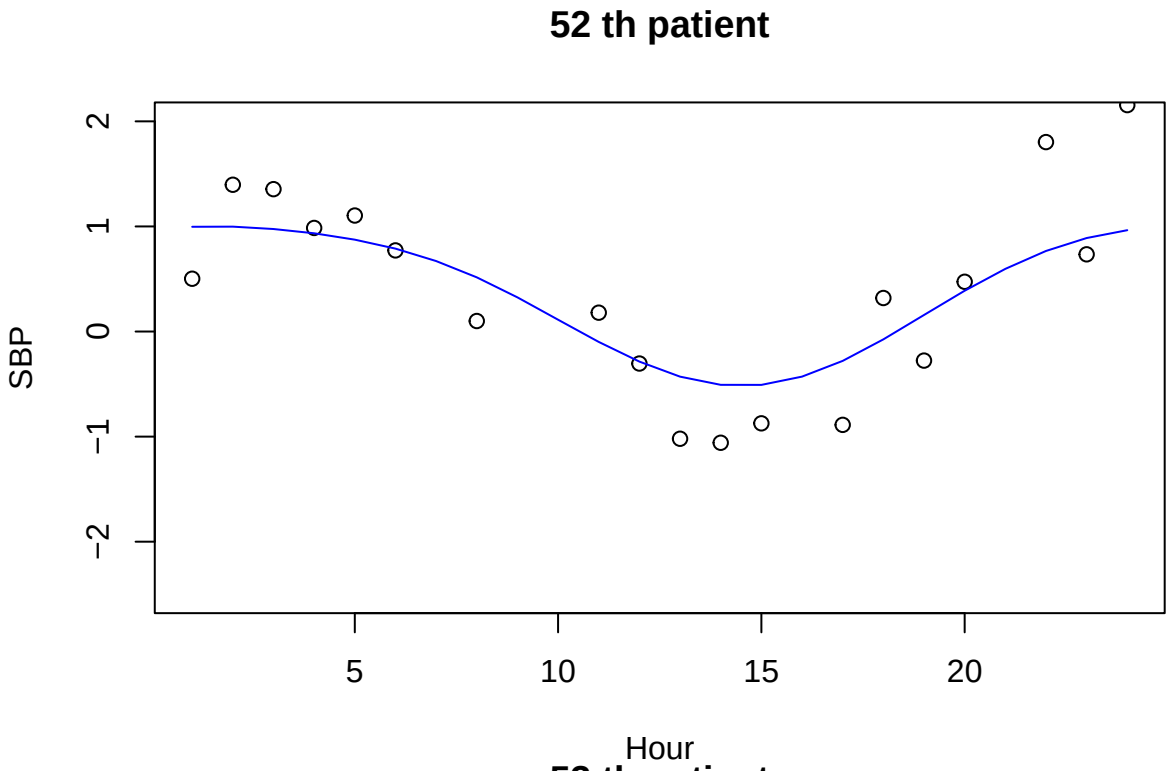


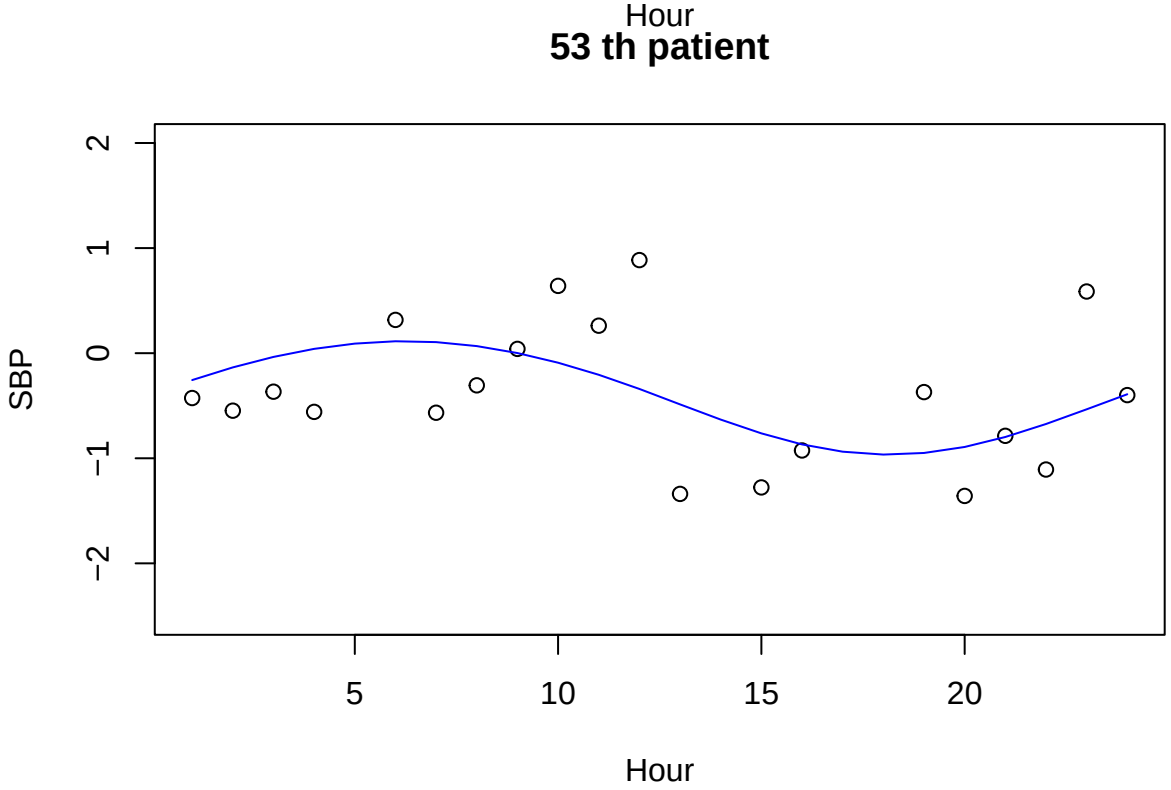


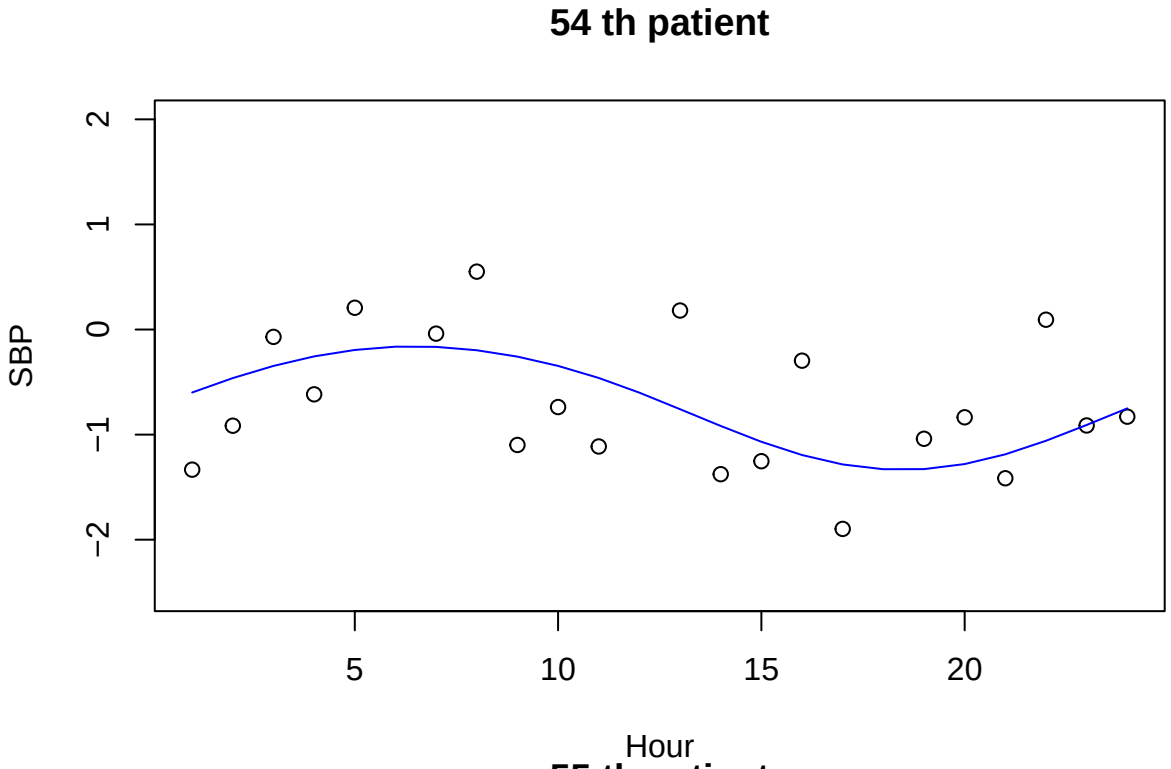

```r
# Plot Estimated Curves for SBP
for (i in 50:75){
  plot(synthetic_dat@data[ , 2, i], main=paste(i,"th patient"), xlab="Hour", ylab="SBP", ylim=c(-2.5,2))
  lines(res$est[ , 2, i], col="blue")
}
```

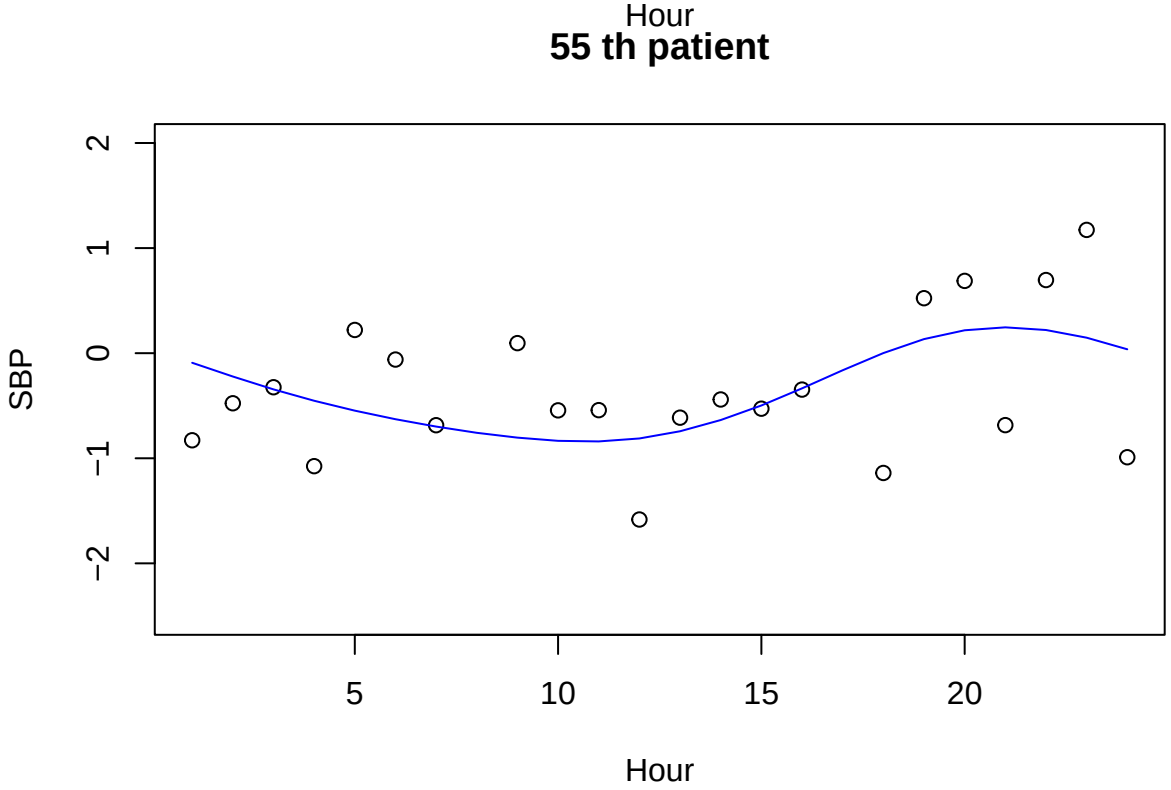

50 th patient
SBP
2
1
0
−1
−2
5
10
15
20
Hour

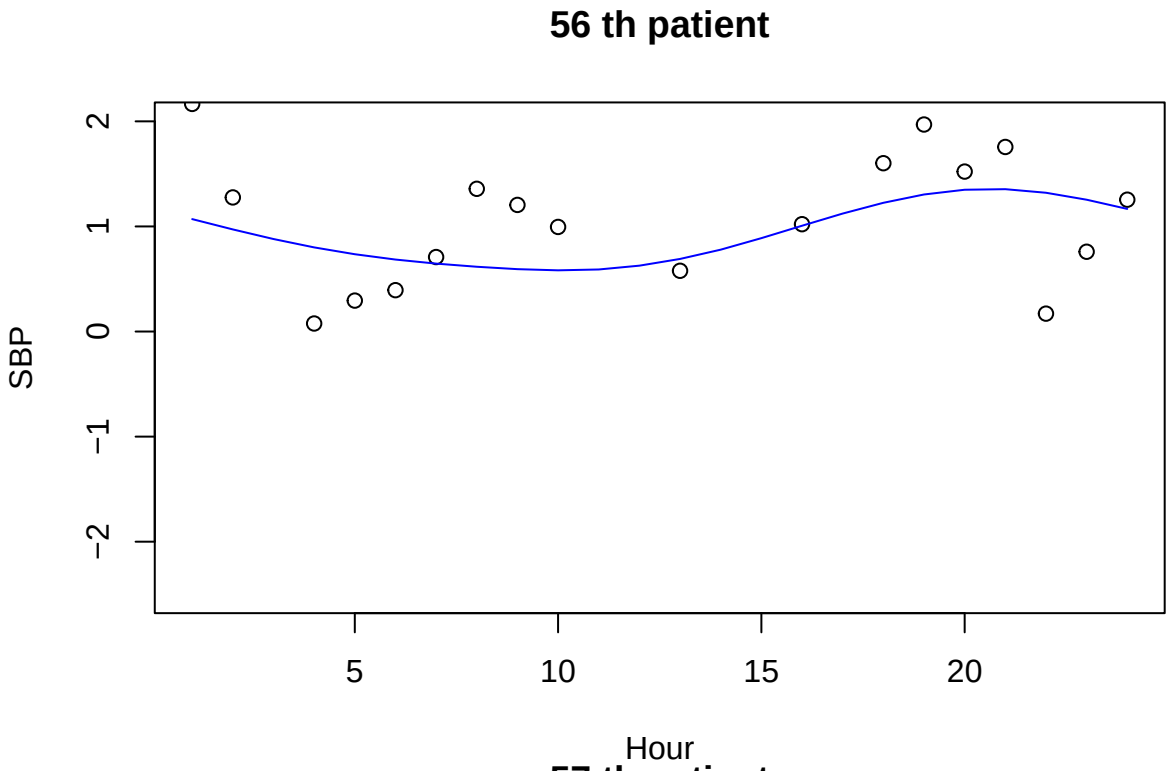

51 th patient
SBP
2
1
0
−1
−2
5
10
15
20
Hour

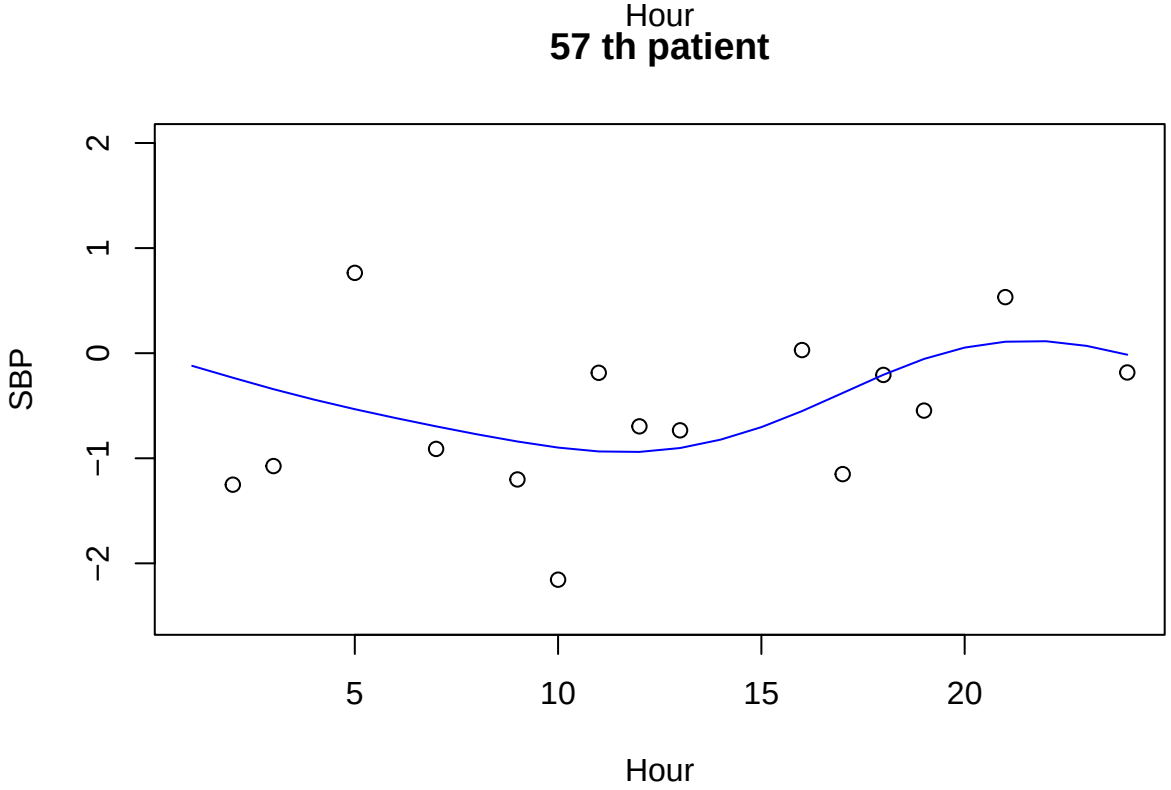

52 th patient
SBP
2
1
0
−1
−2
5
10
15
20
Hour

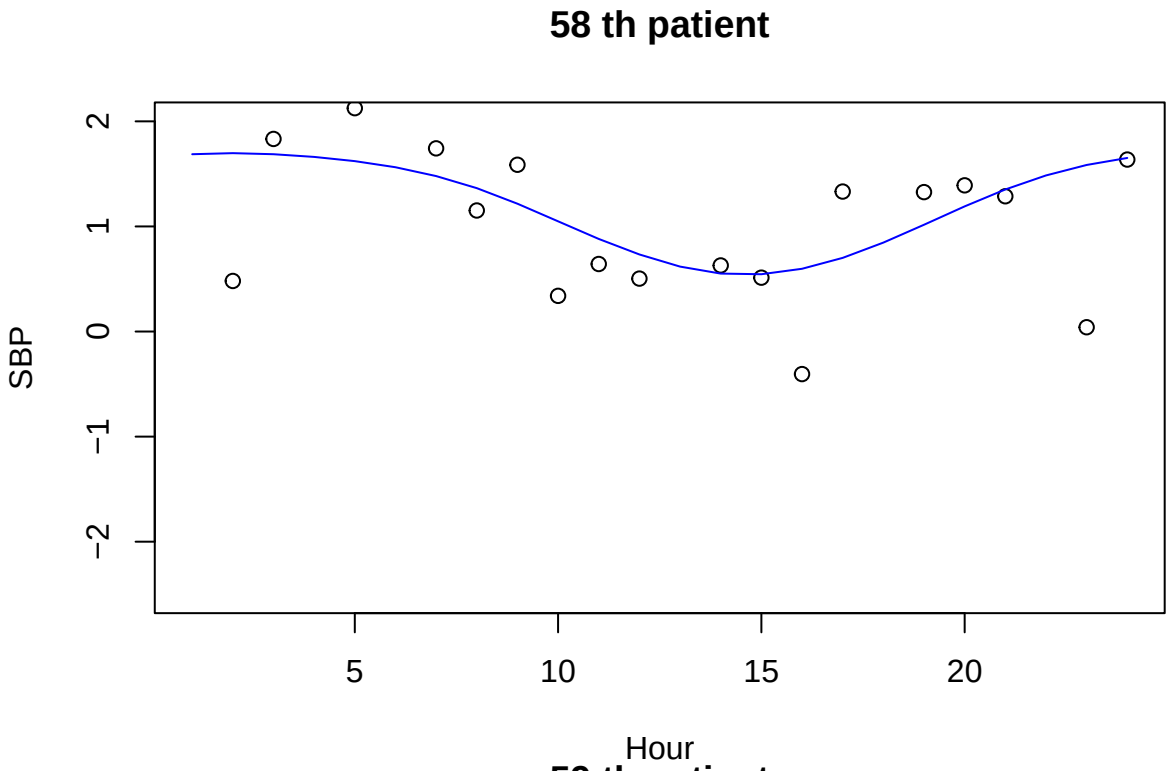

53 th patient
SBP
2
1
0
−1
−2
5
10
15
20
Hour

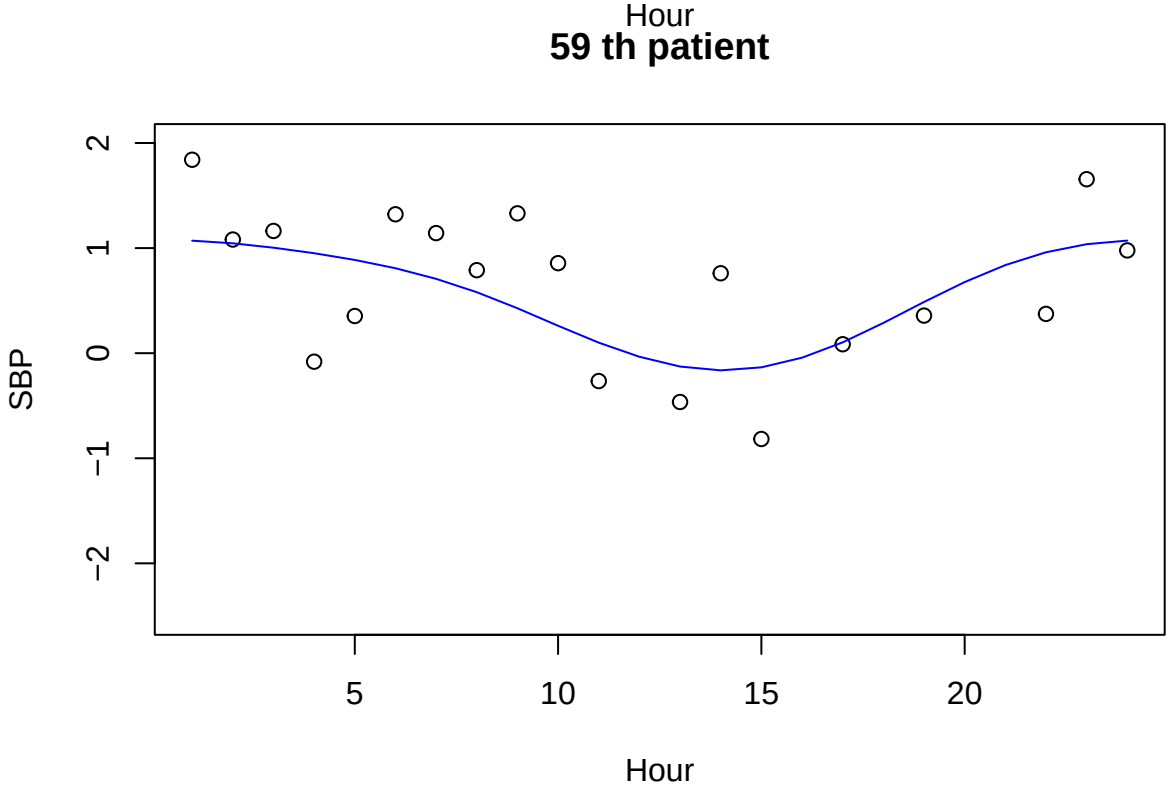

54 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

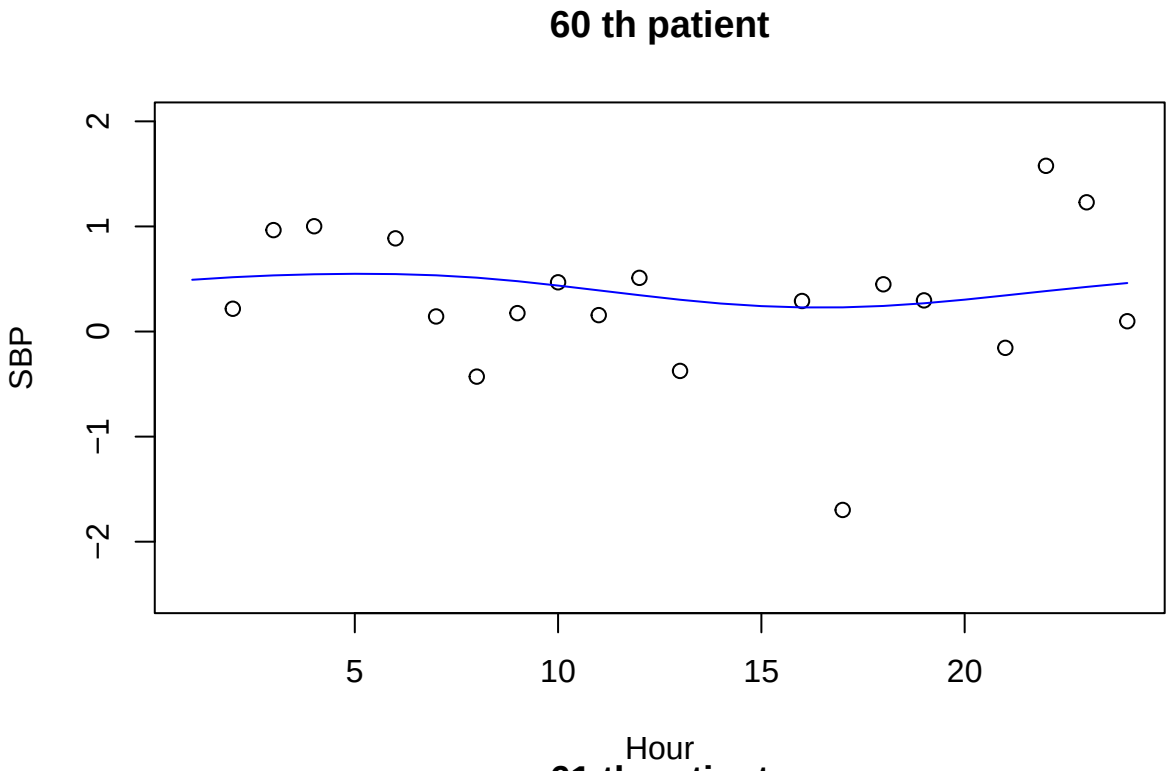

55 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

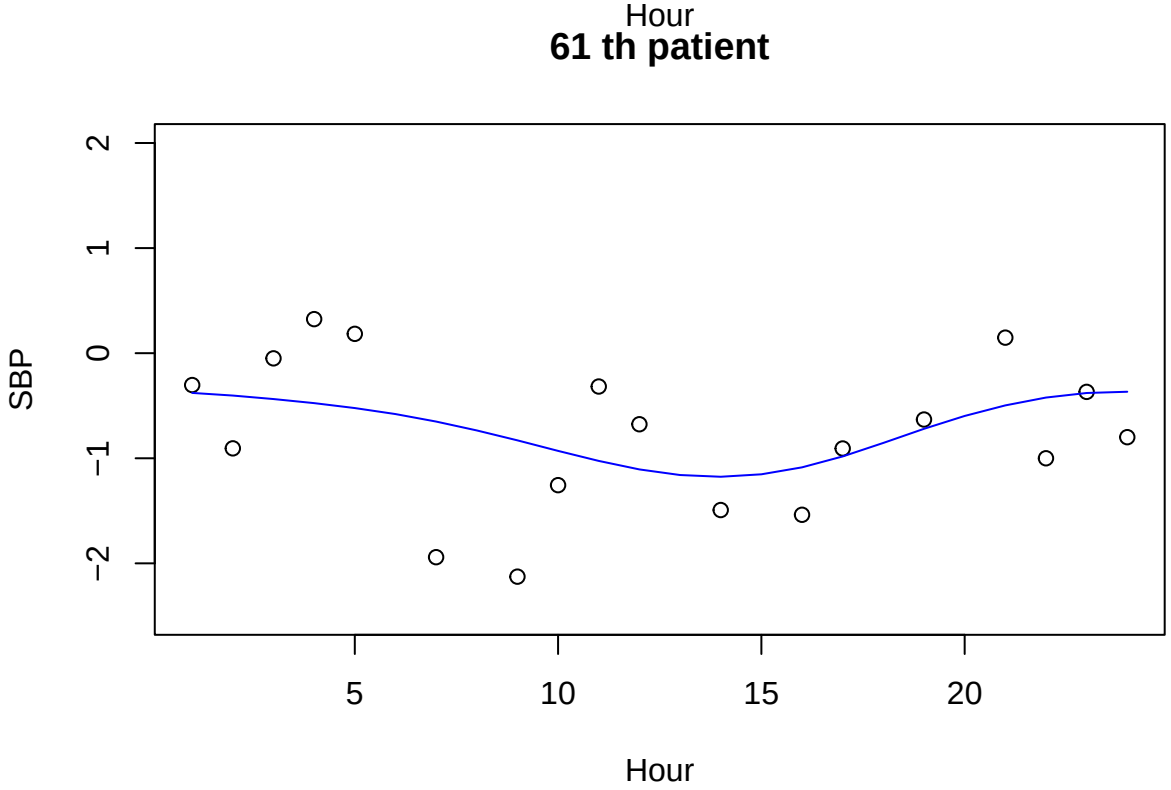

56 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

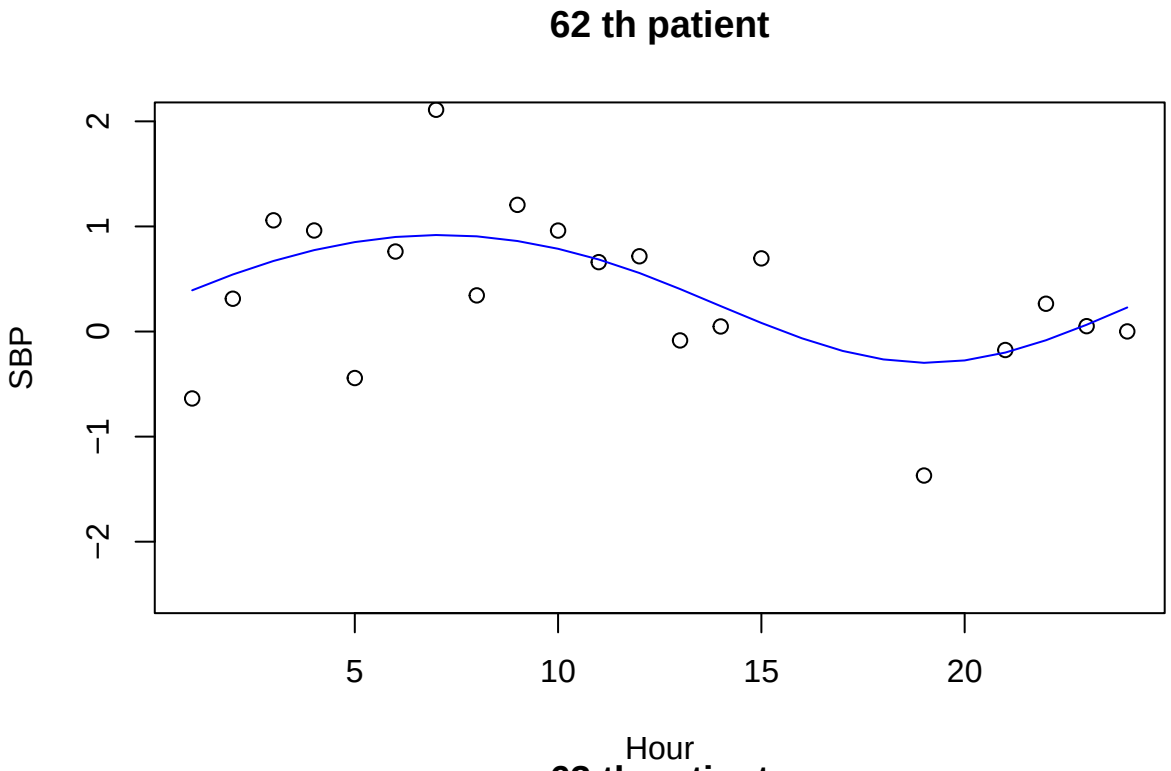

57 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

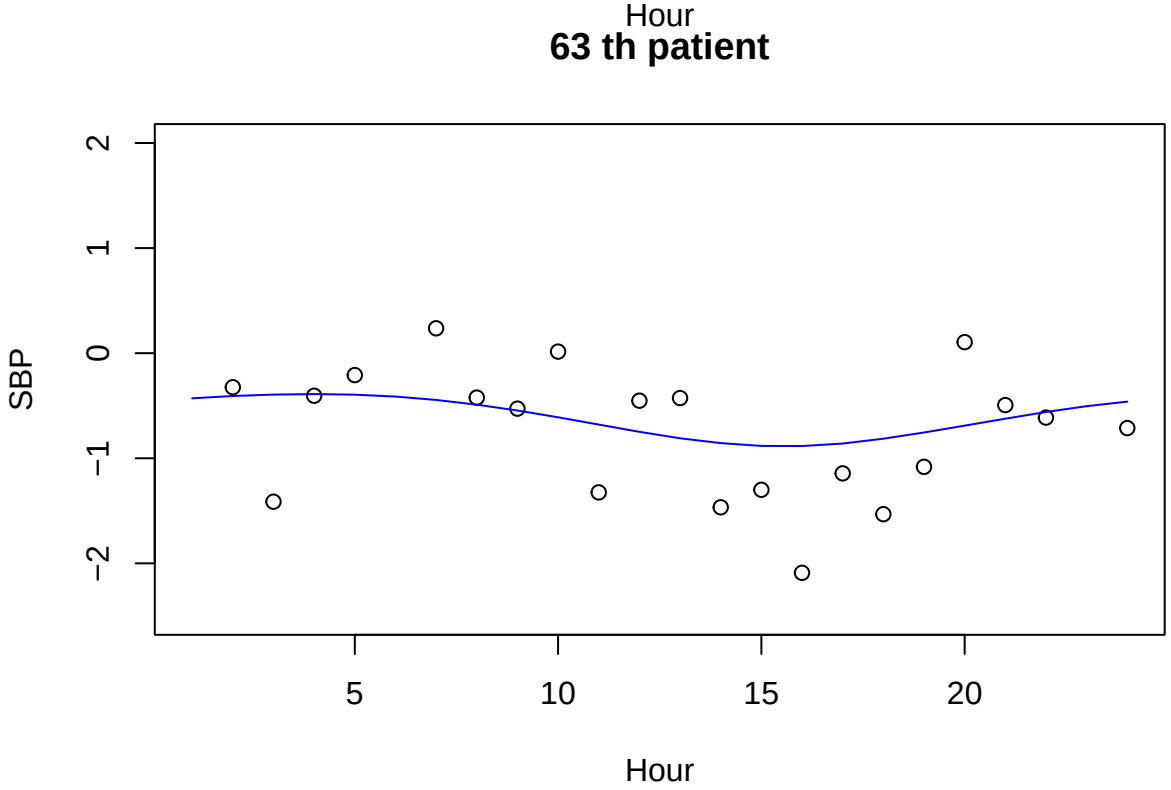

58 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

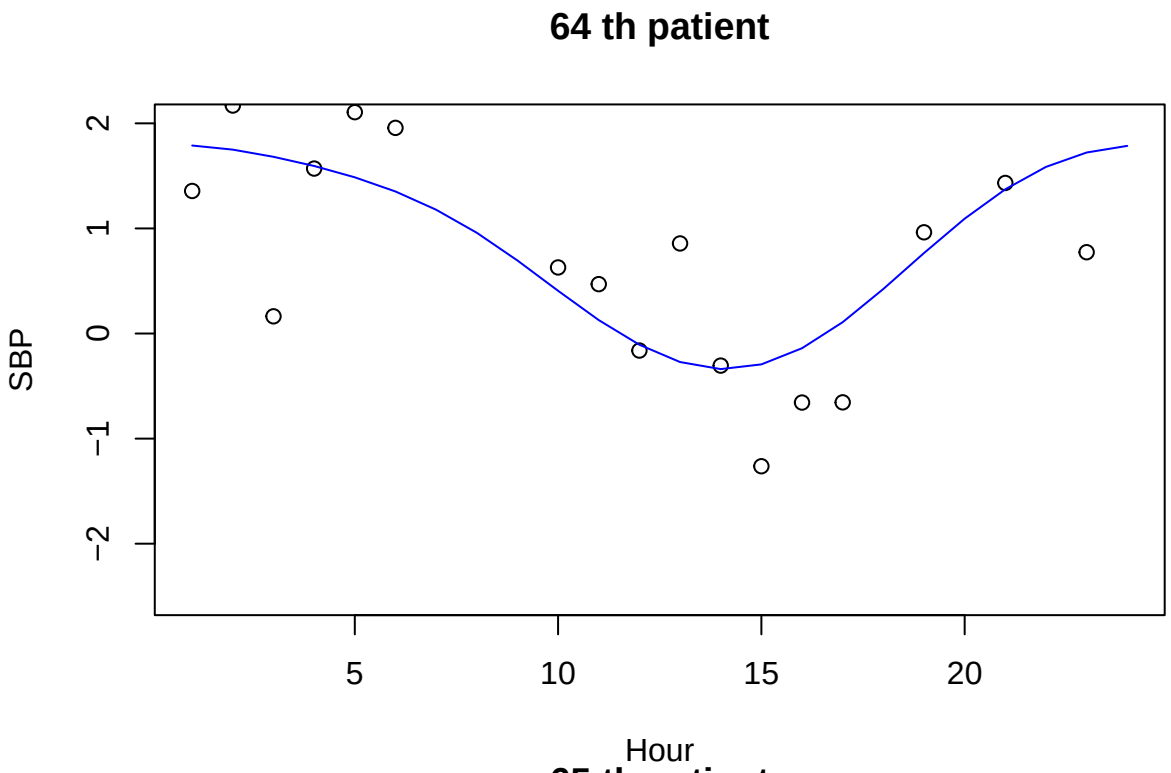

59 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

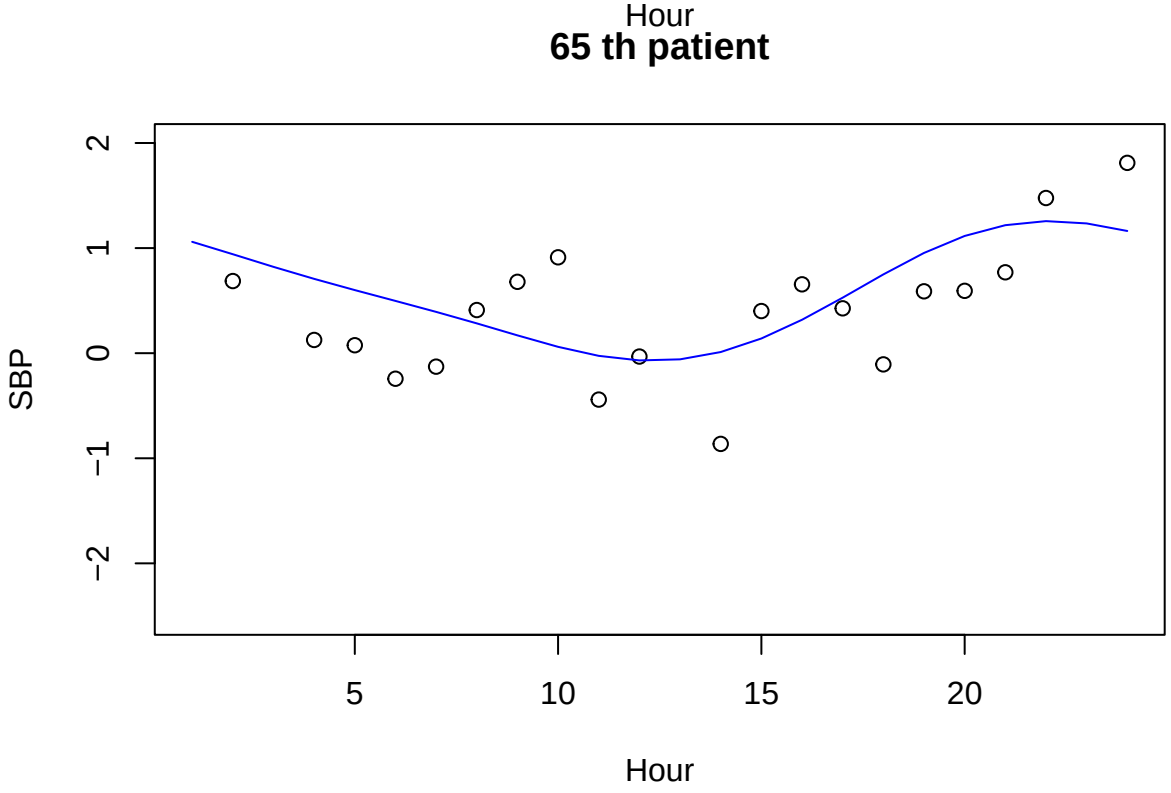

60 th patient
SBP
2
1
0
−1
−2
5
10
15
20
Hour

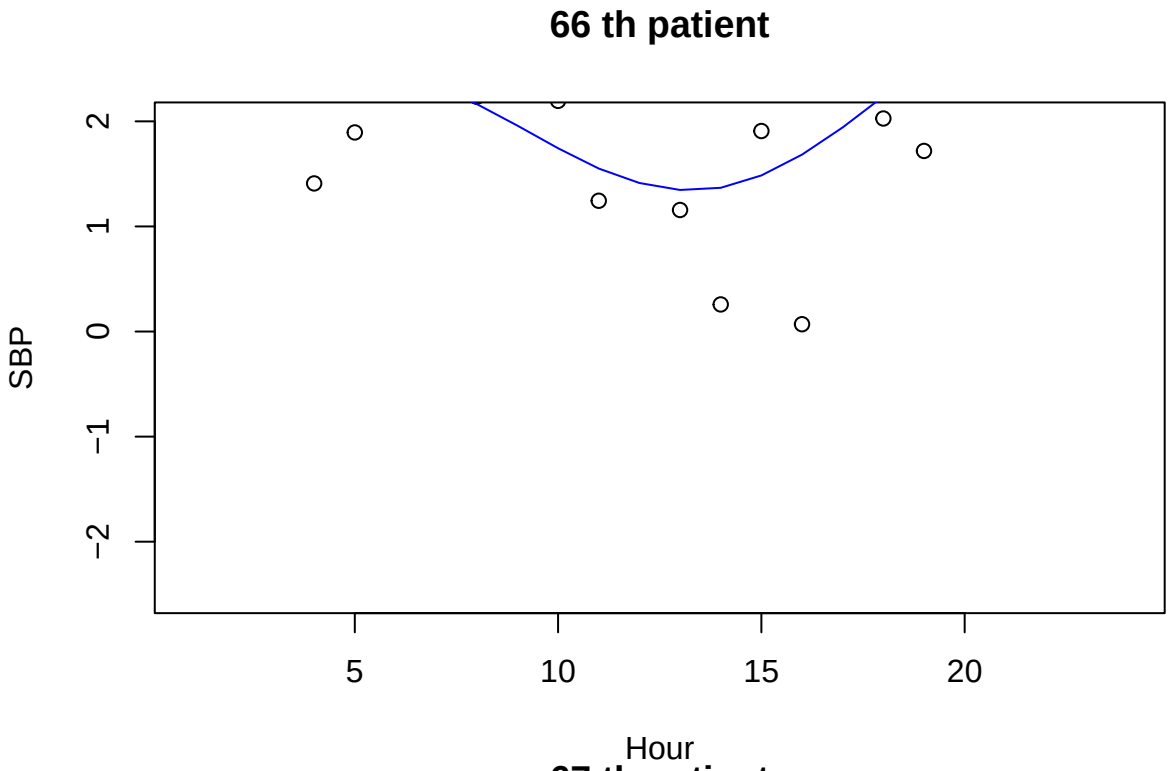

61 th patient
SBP
2
1
0
−1
−2
5
10
15
20
Hour

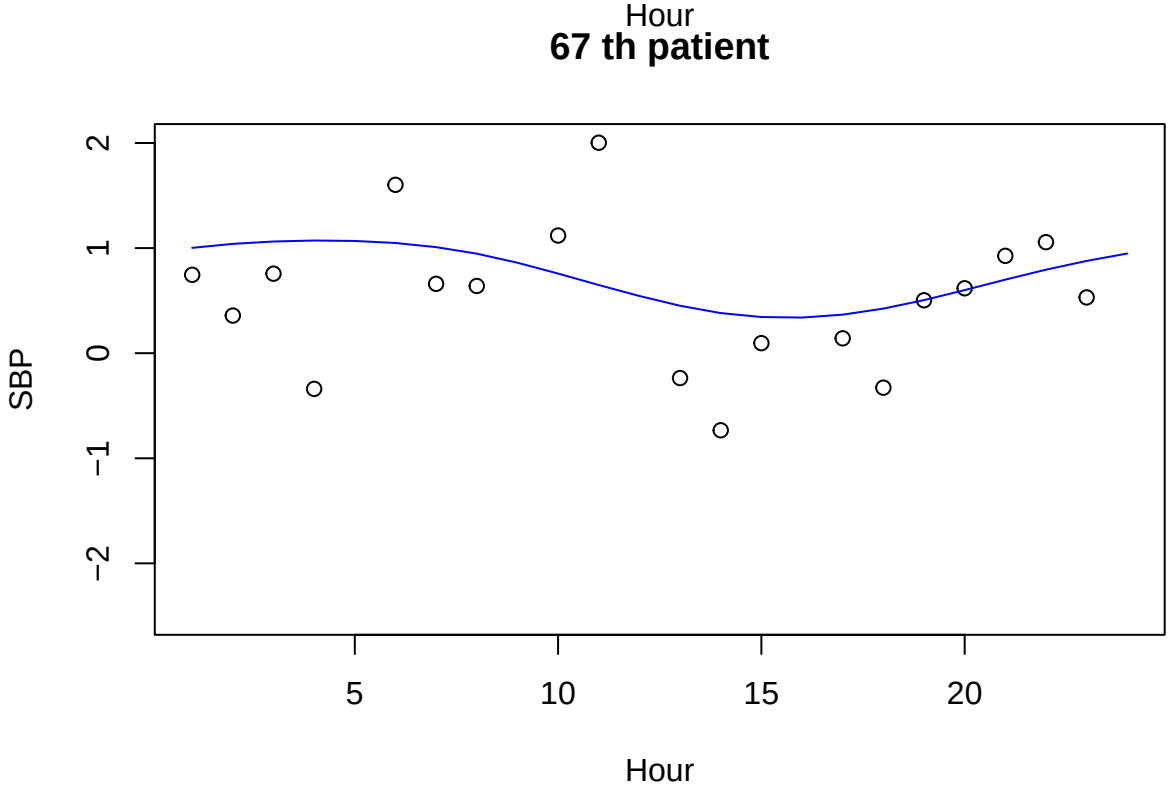

62 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

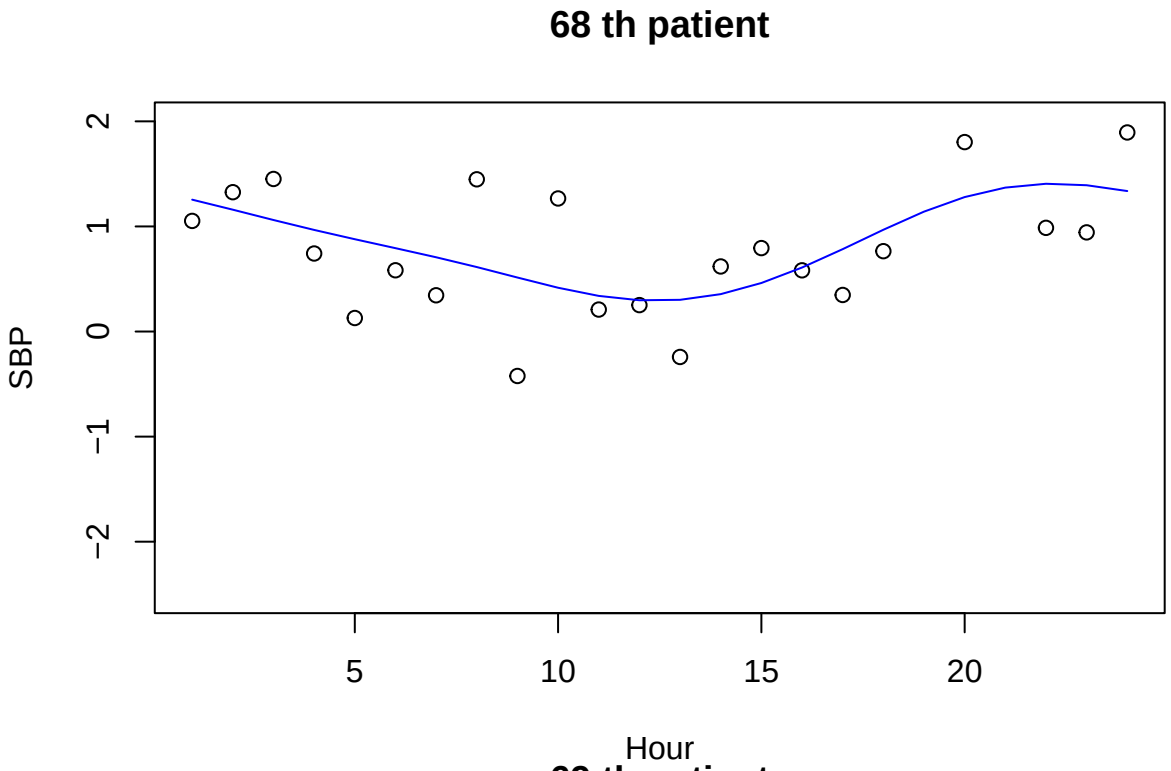

63 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

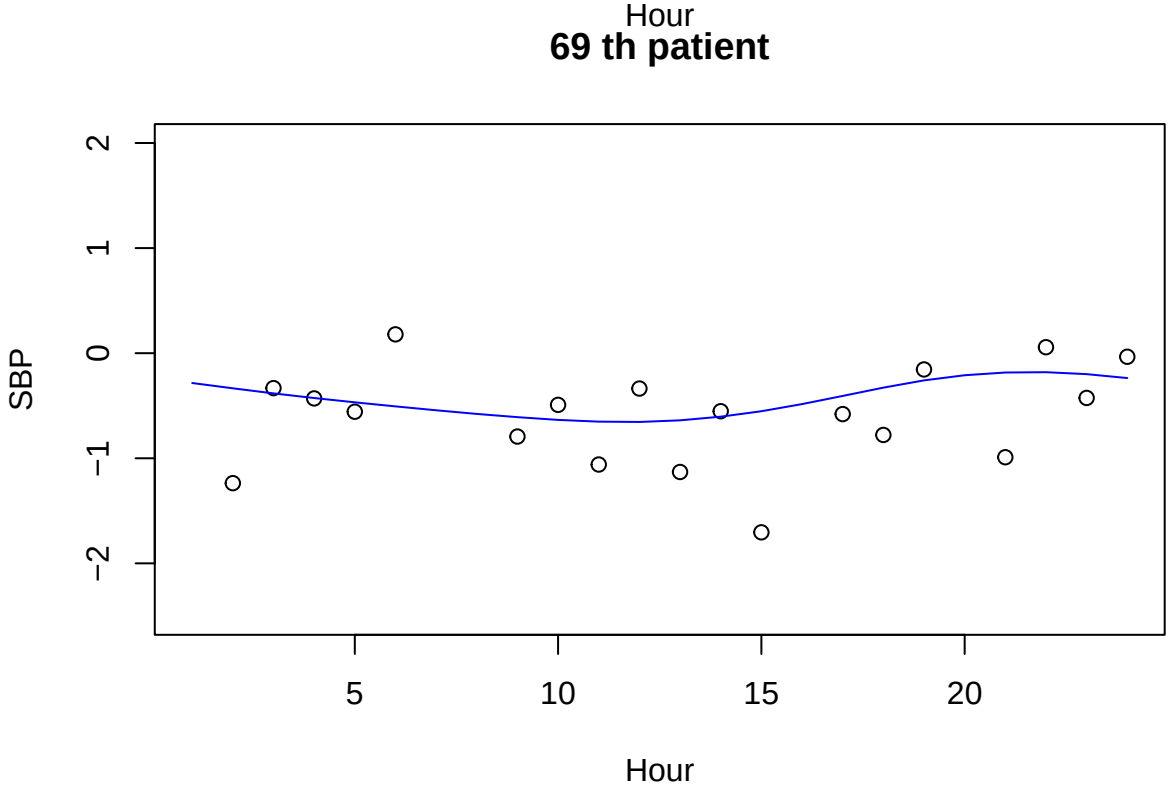

64 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

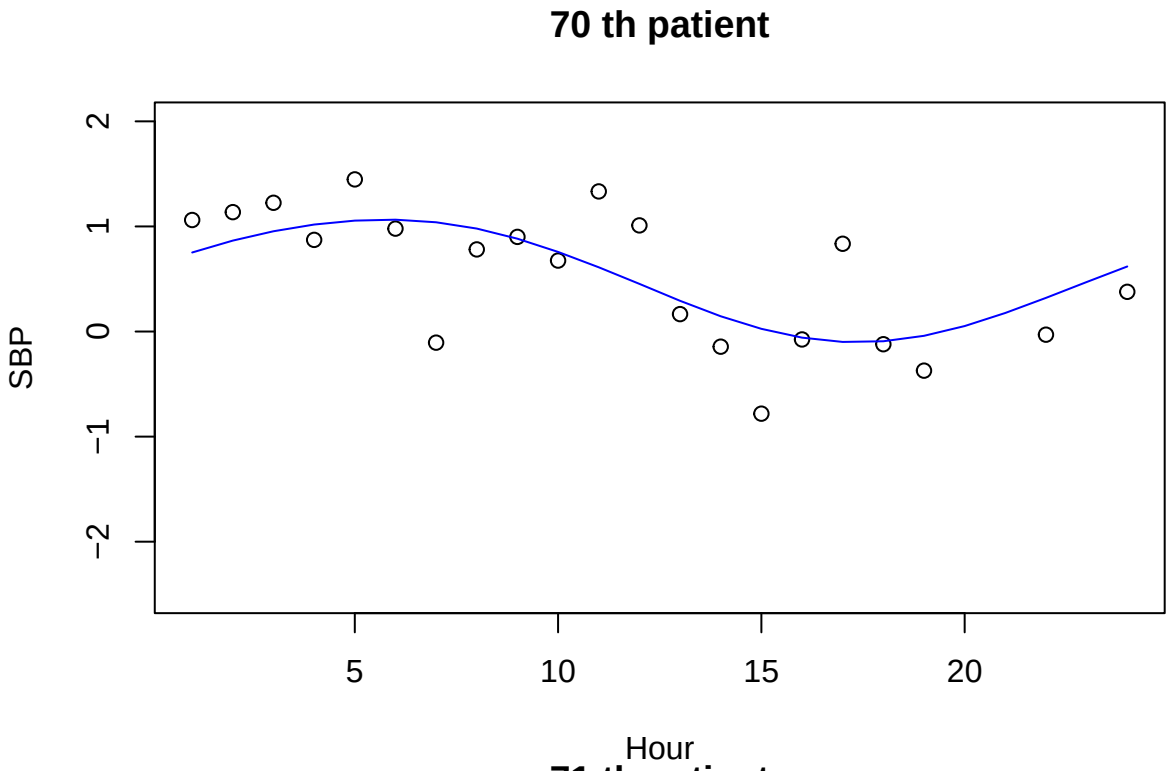

65 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

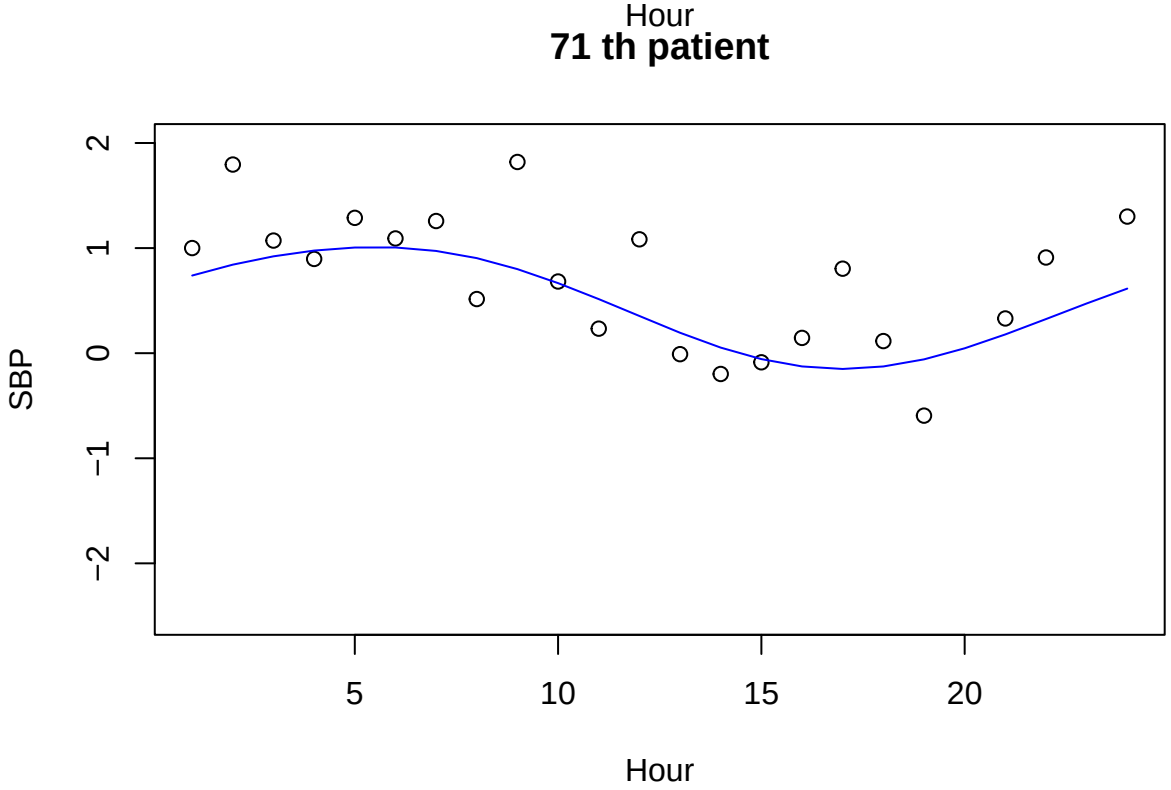

66 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

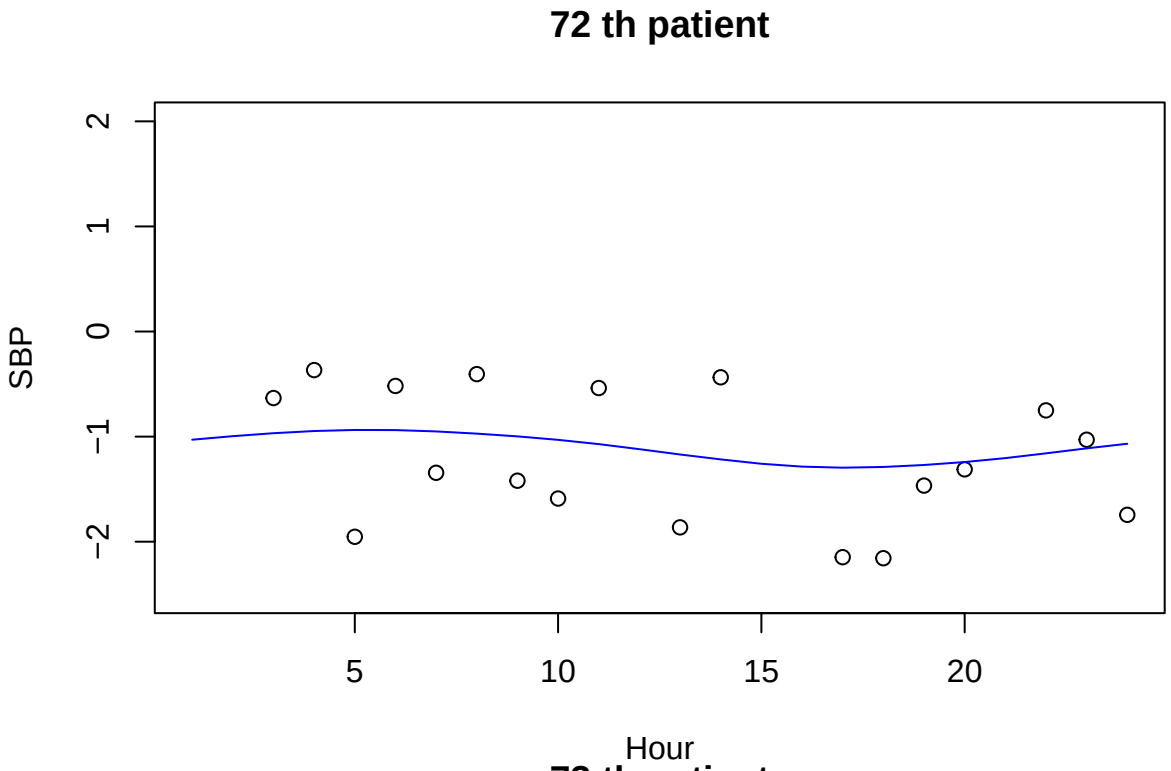

67 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

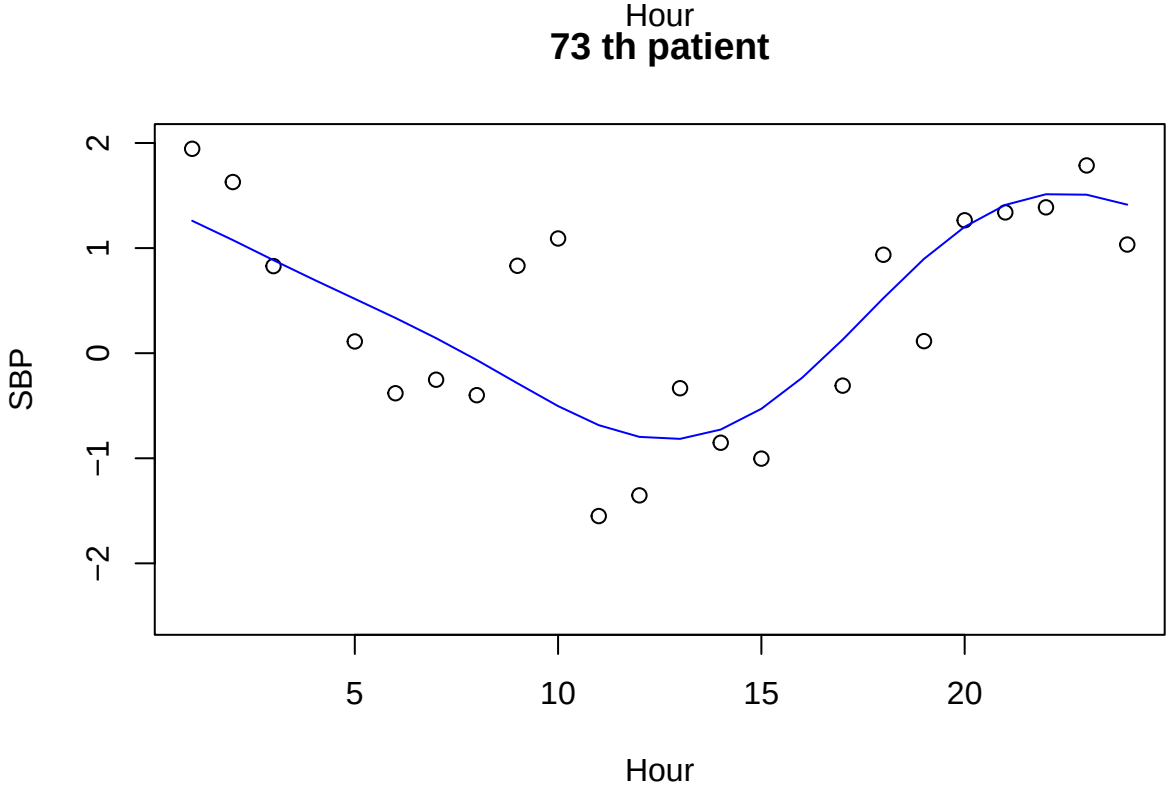

68 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

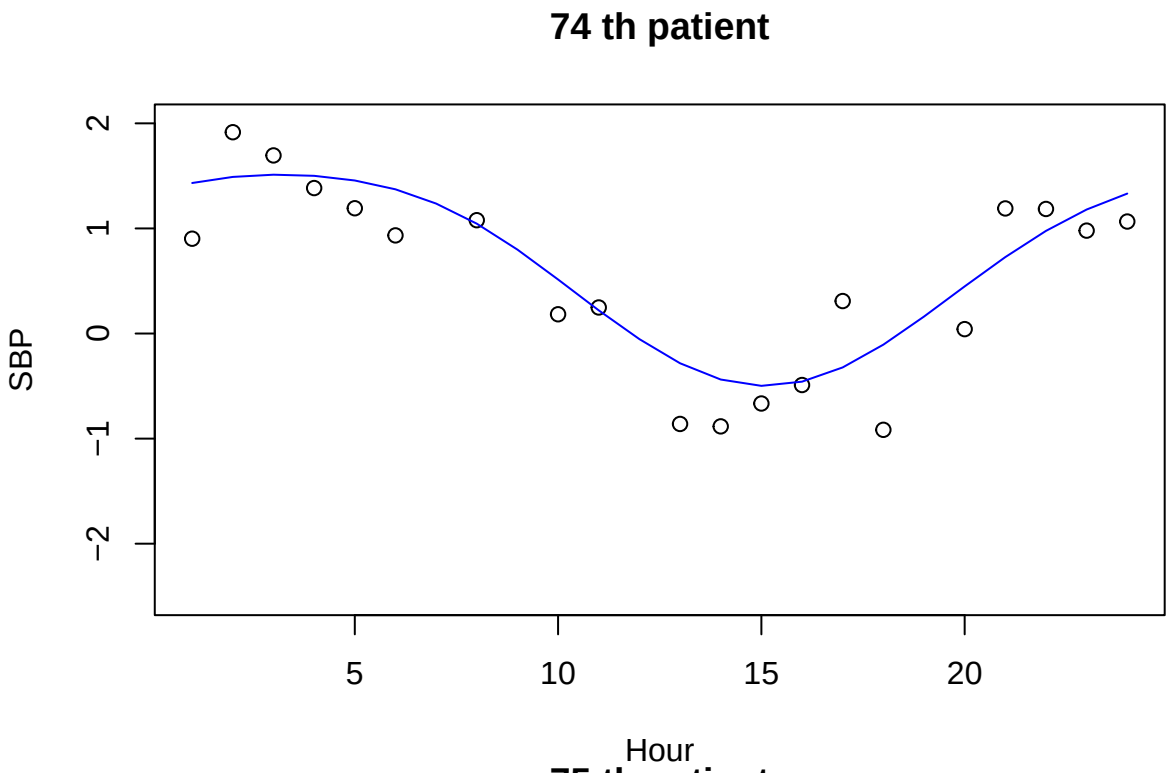

69 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

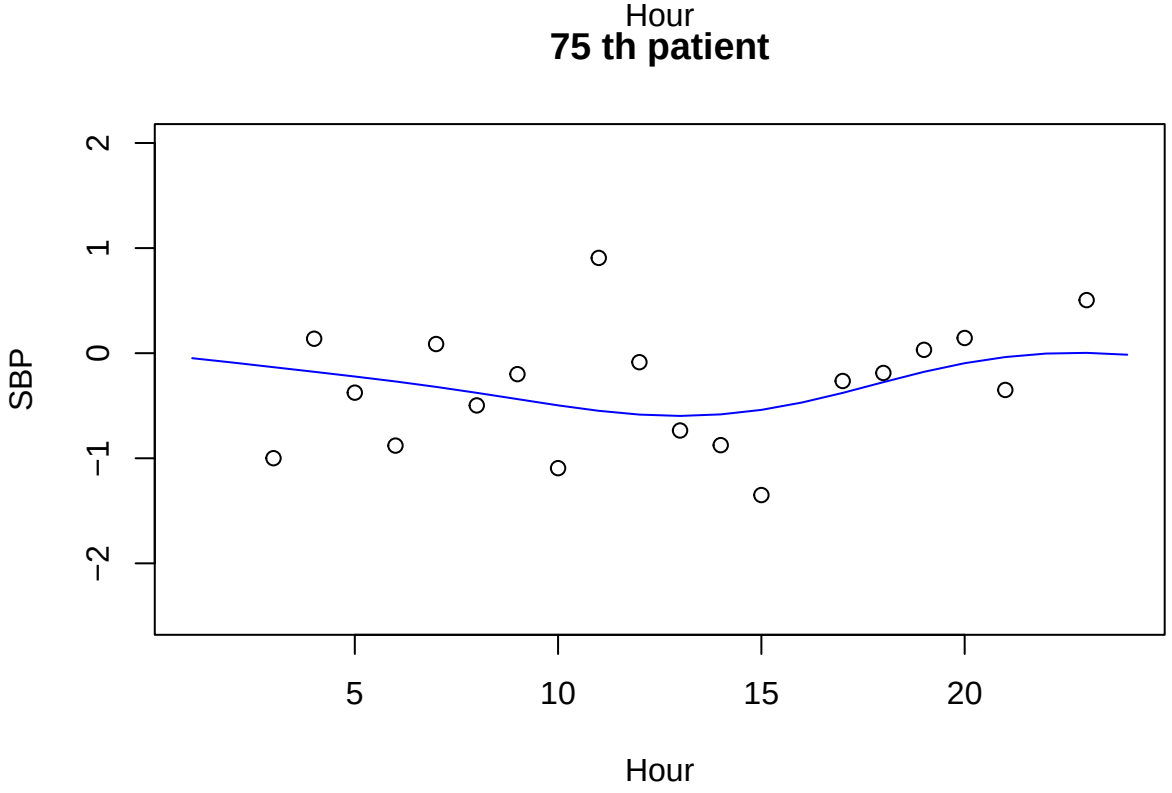

70 th patient
SBP
2
1
0
−1
−2
5
10
15
20
Hour


71 th patient
SBP
2
1
0
−1
−2
5
10
15
20
Hour

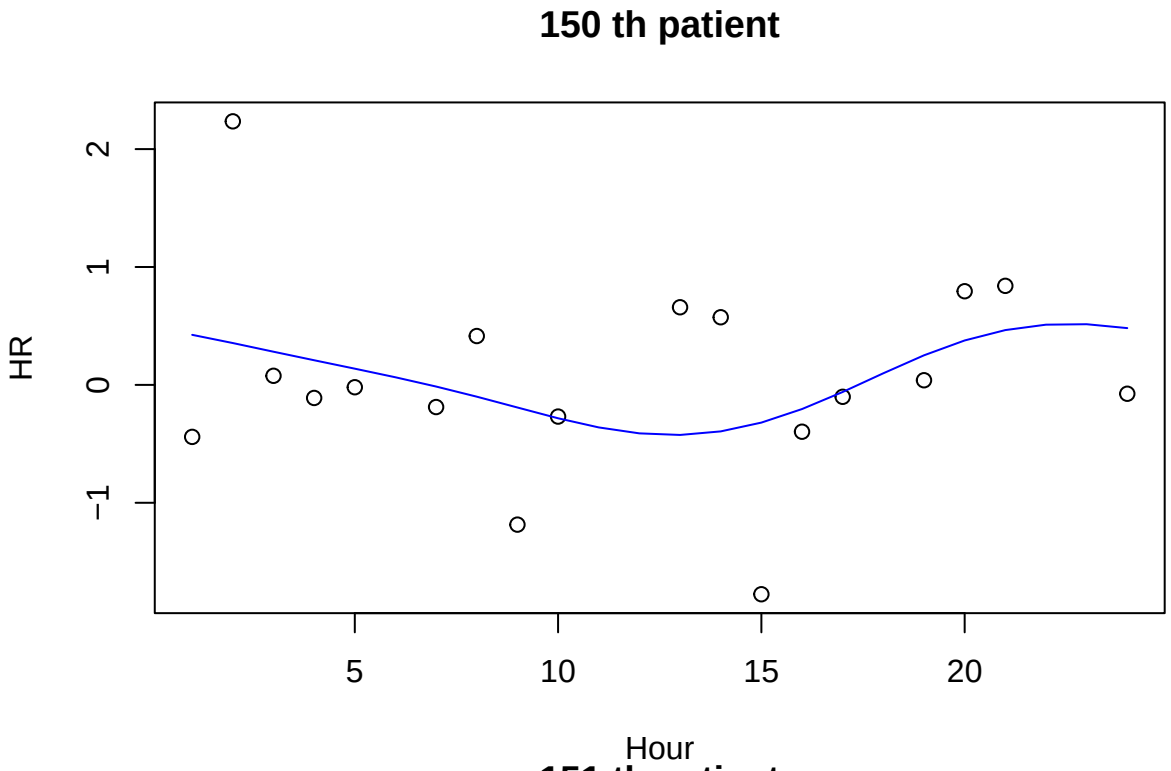

72 th patient
SBP
2
1
0
−1
−2
5
10
15
20
Hour

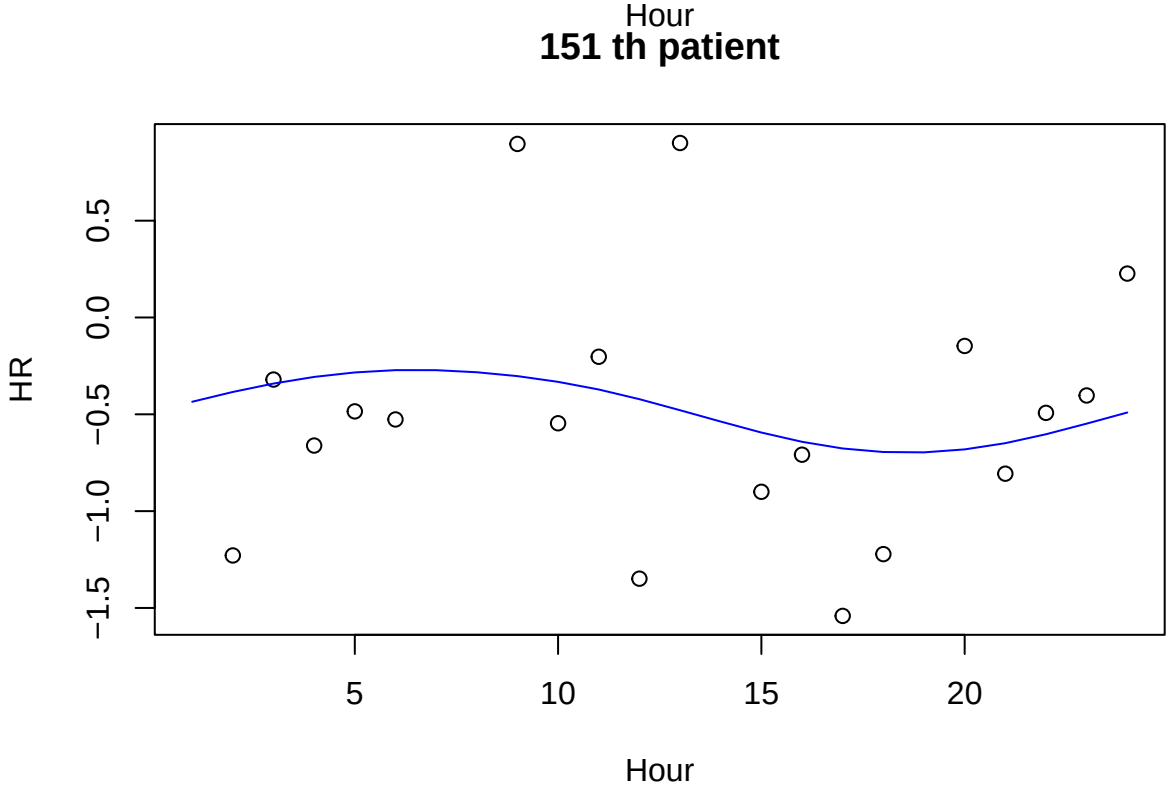

73 th patient
SBP
2
1
0
−1
−2
5
10
15
20
Hour

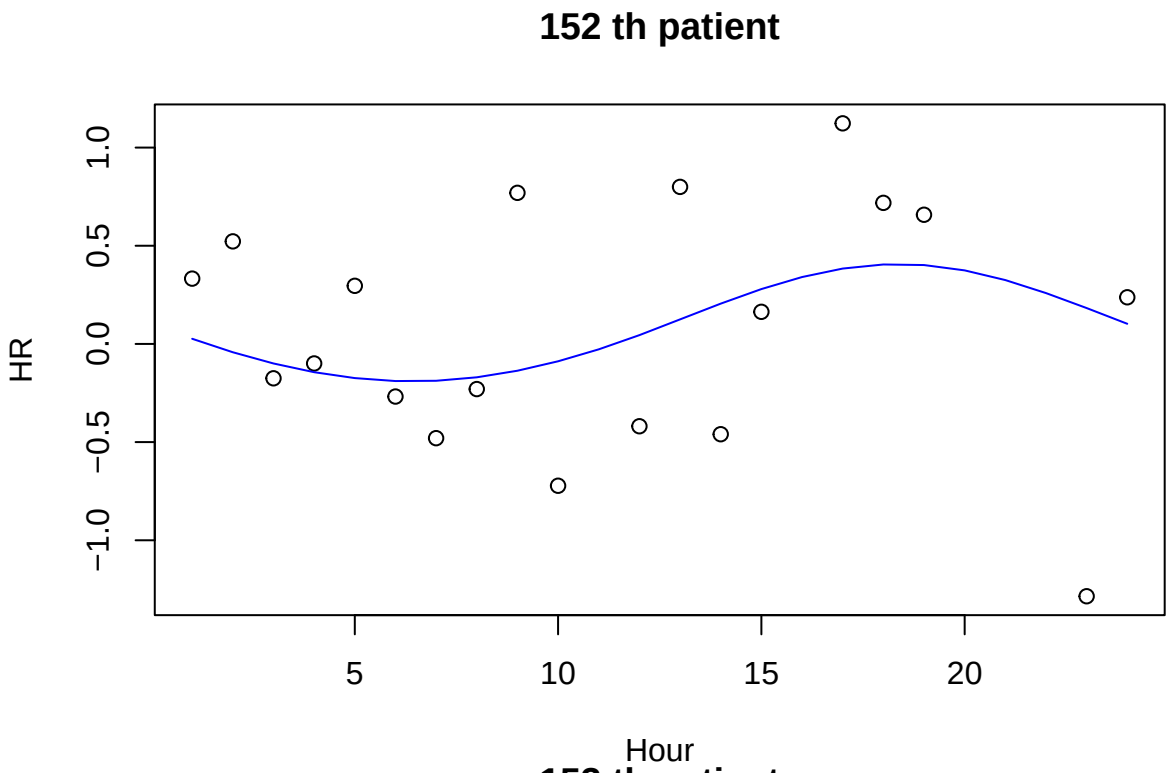


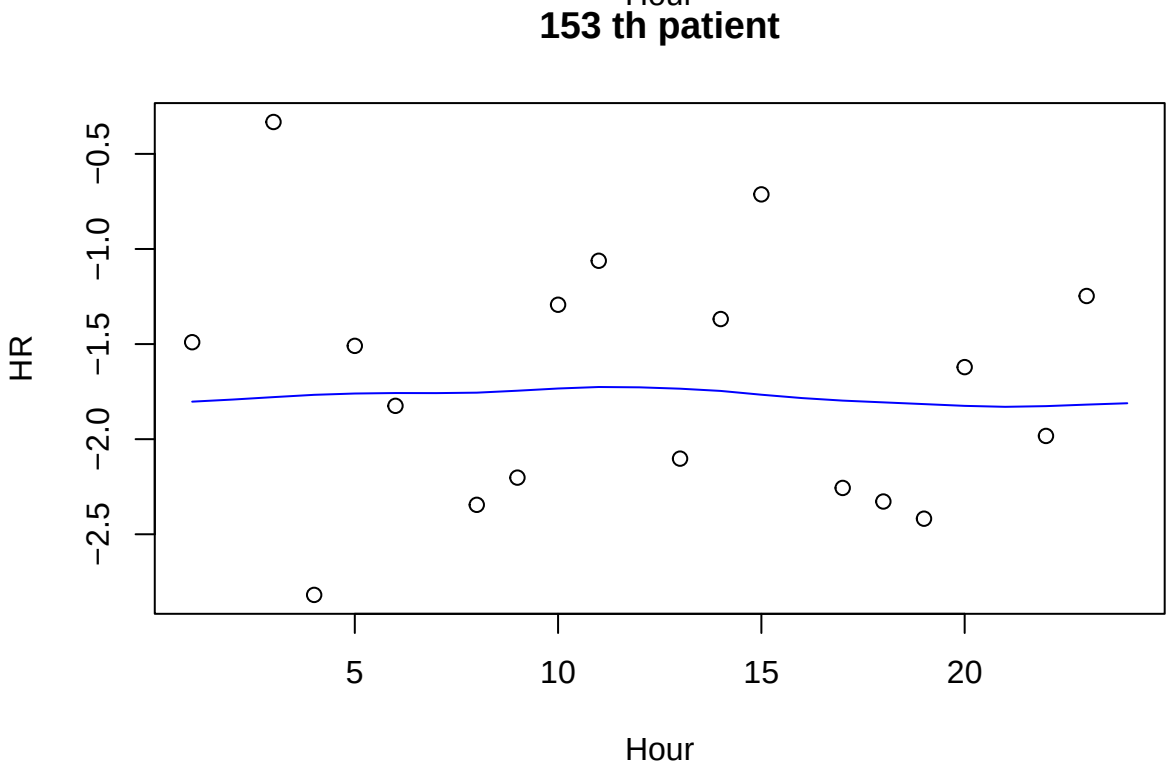


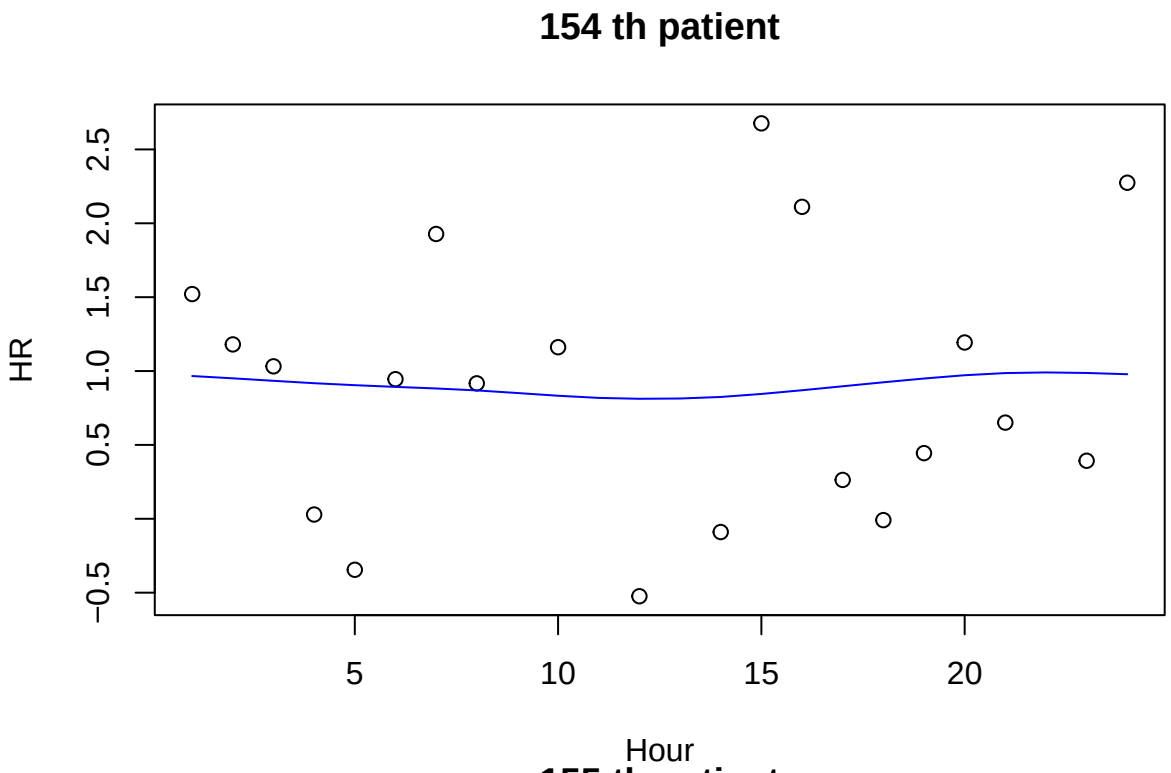

```
# Plot Estimated Curves for HR
for (i in 150:175){
  plot(synthetic_dat@data[ , 3, i], main=paste(i,"th patient"), xlab="Hour", ylab="HR")
  lines(res$est[ , 3, i], col="blue")
}
```

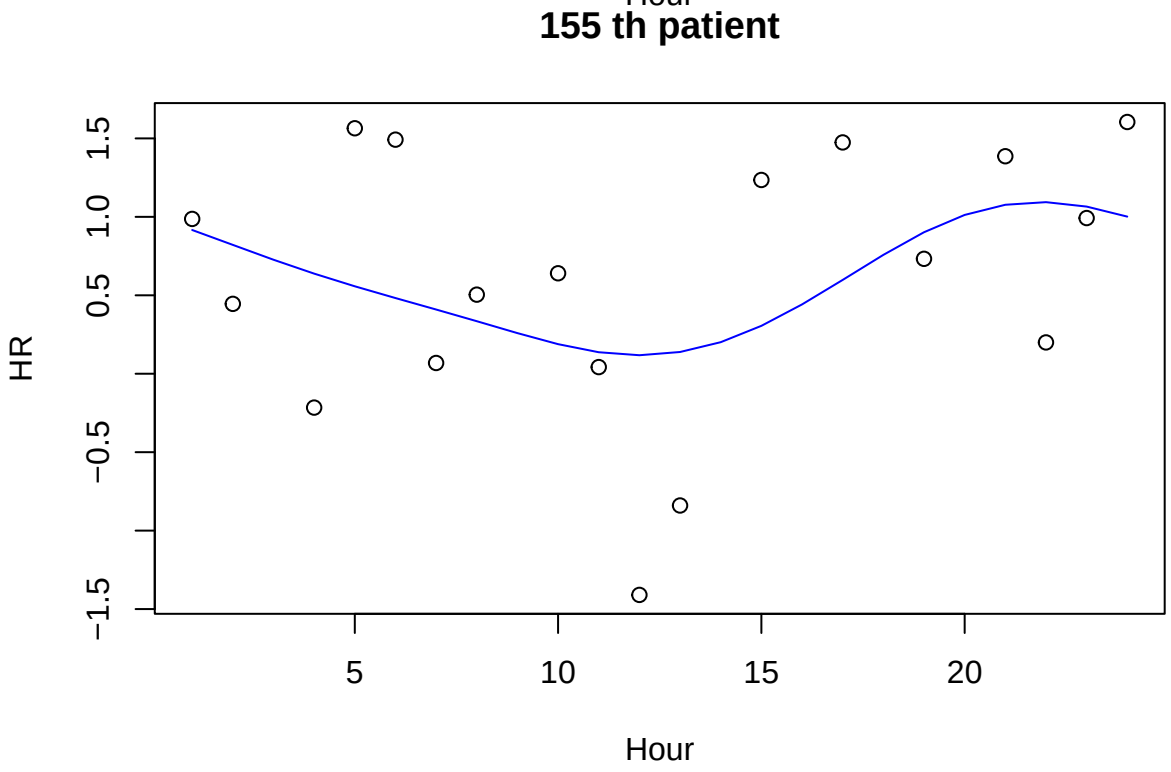

150 th patient
HR
2
1
0
−1
5
10
15
20
Hour

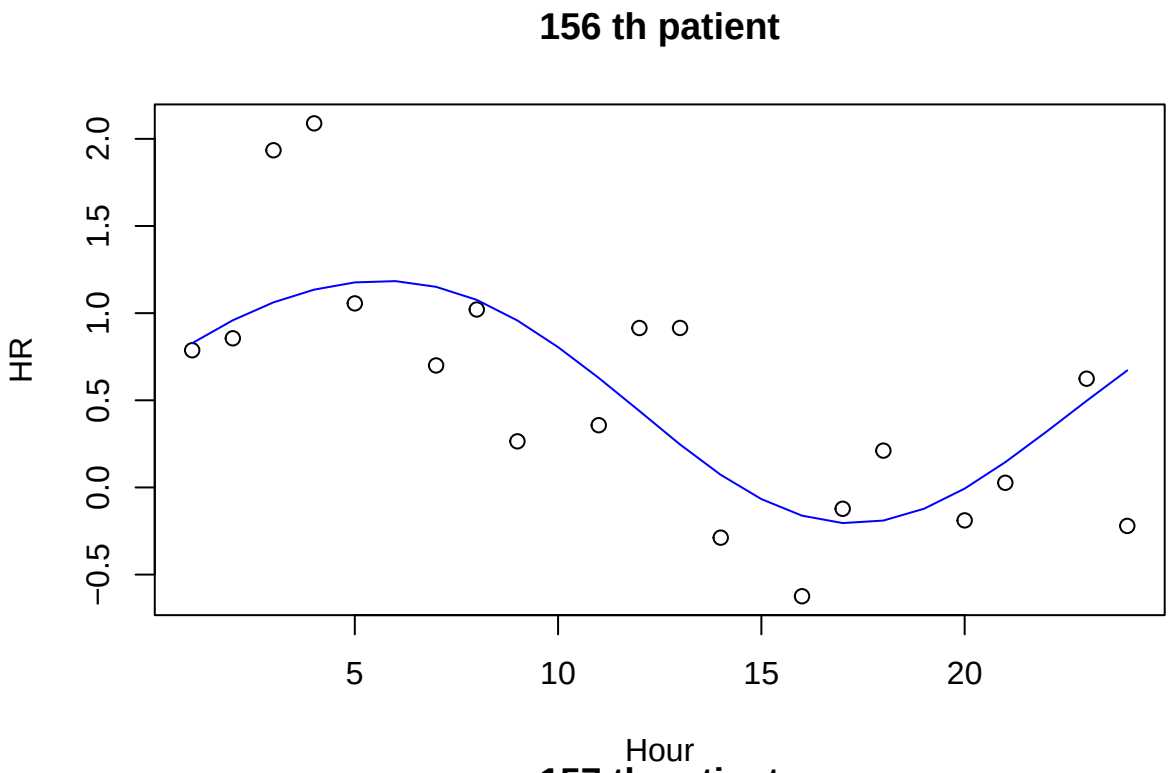

151 th patient
HR
0.5
0.0
−0.5
−1.0
−1.5
5
10
15
20
Hour

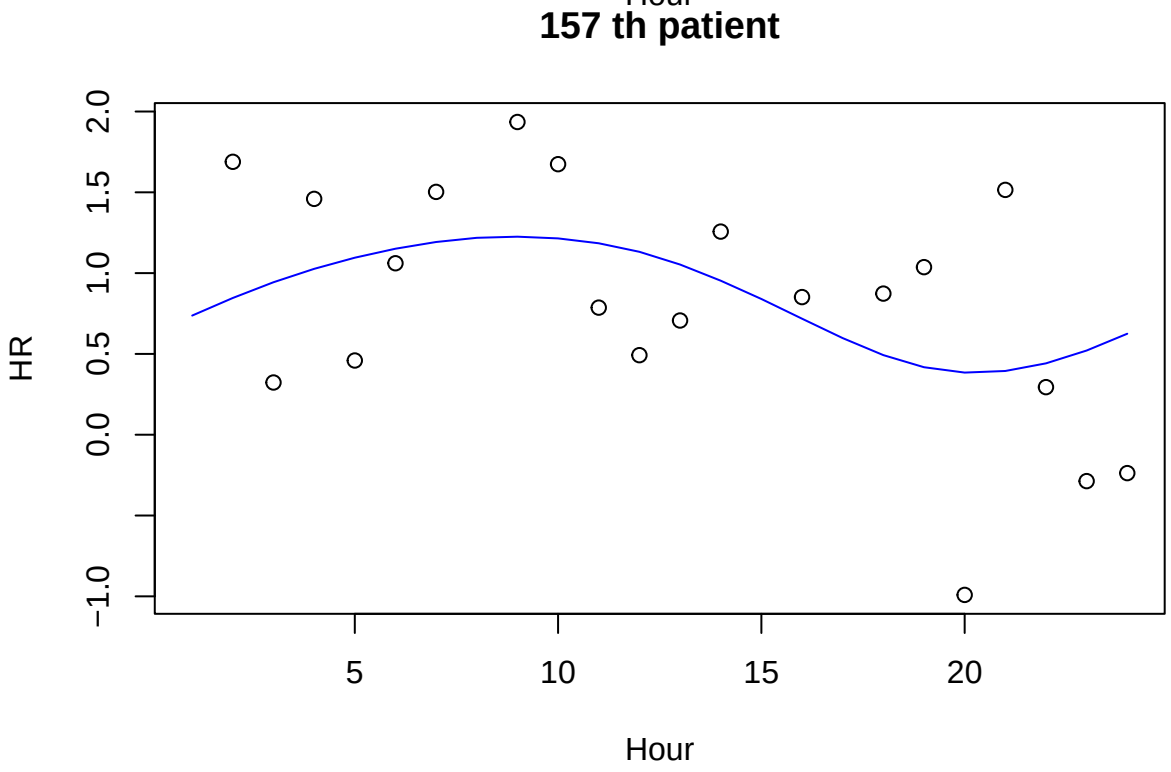

152 th patient
HR
1.0
0.5
0.0
−0.5
−1.0
5
10
15
20
Hour

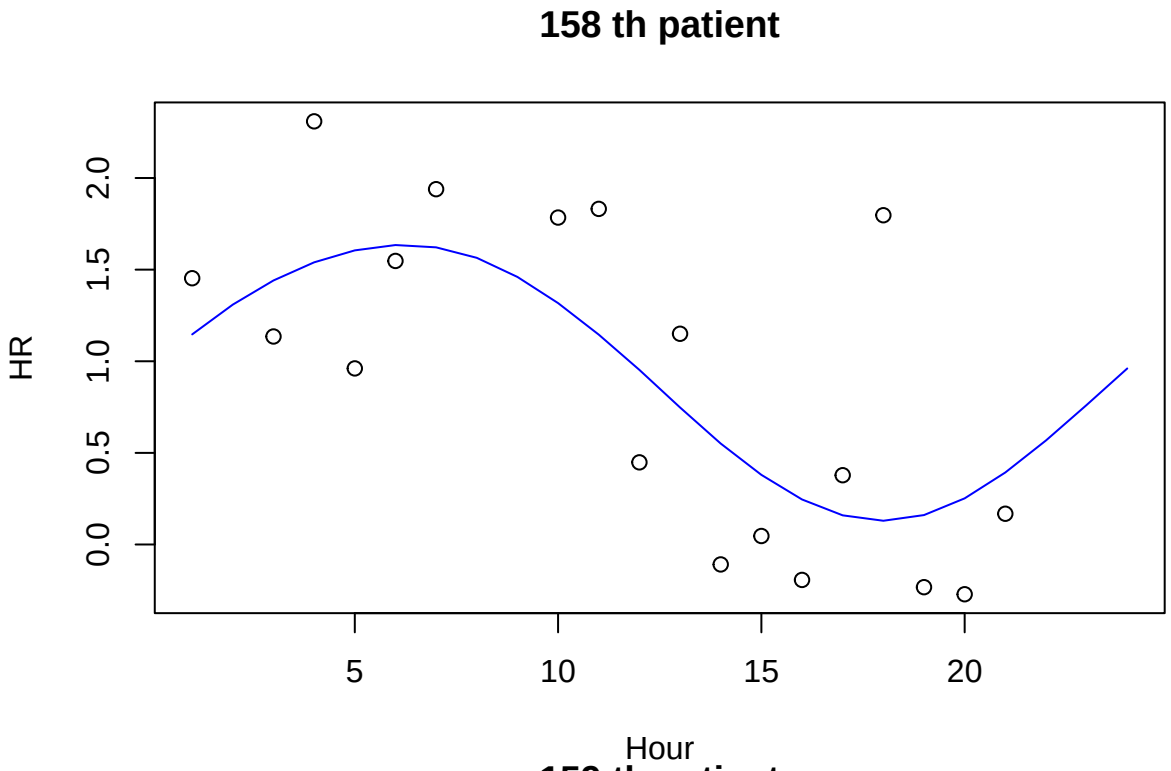

153 th patient
HR
−0.5
−1.0
−1.5
−2.0
−2.5
5
10
15
20
Hour

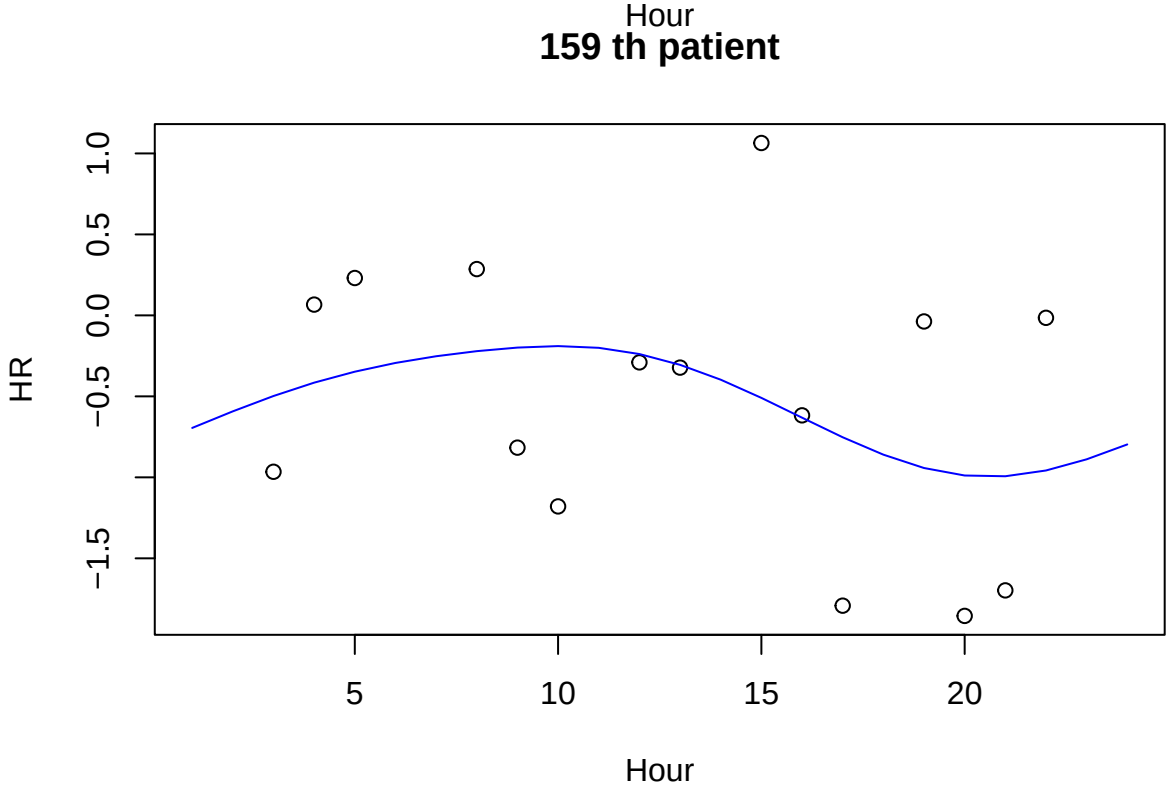

154 th patient
HR
−0.5
0.5
1.0
1.5
2.0
2.5
5
10
15
20
Hour

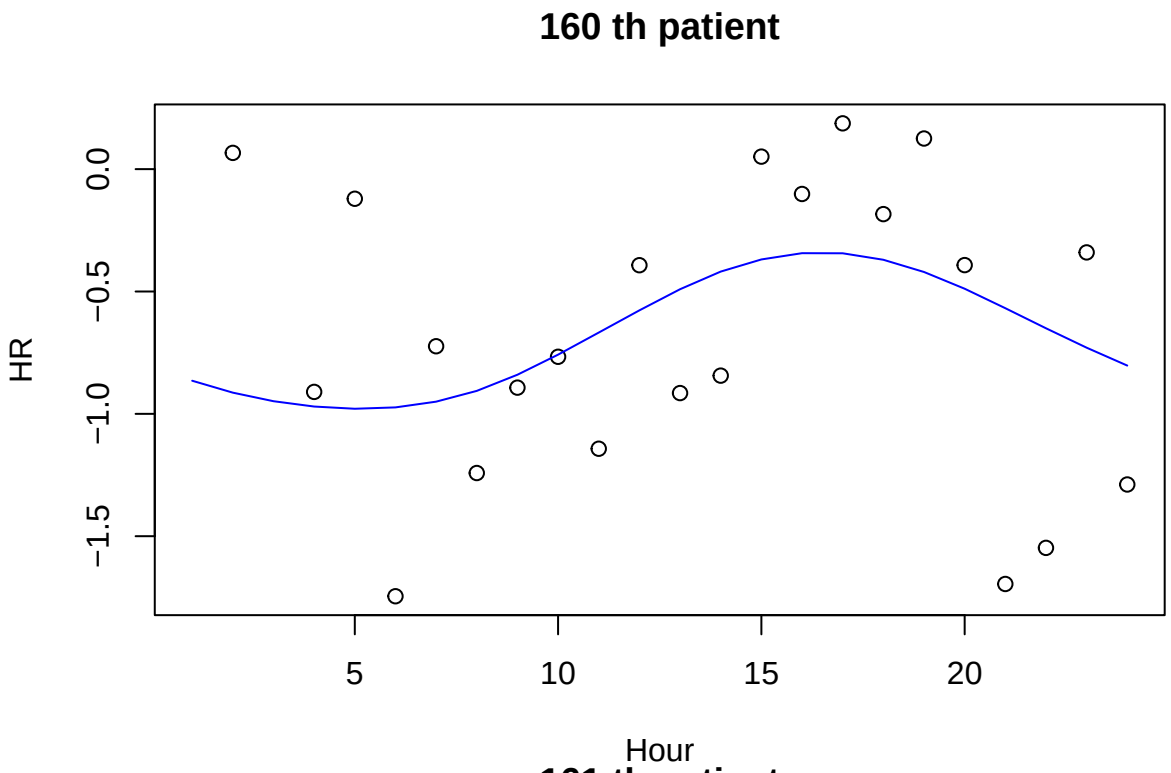

155 th patient
HR
−1.5
−0.5
0.5
1.0
1.5
5
10
15
20
Hour

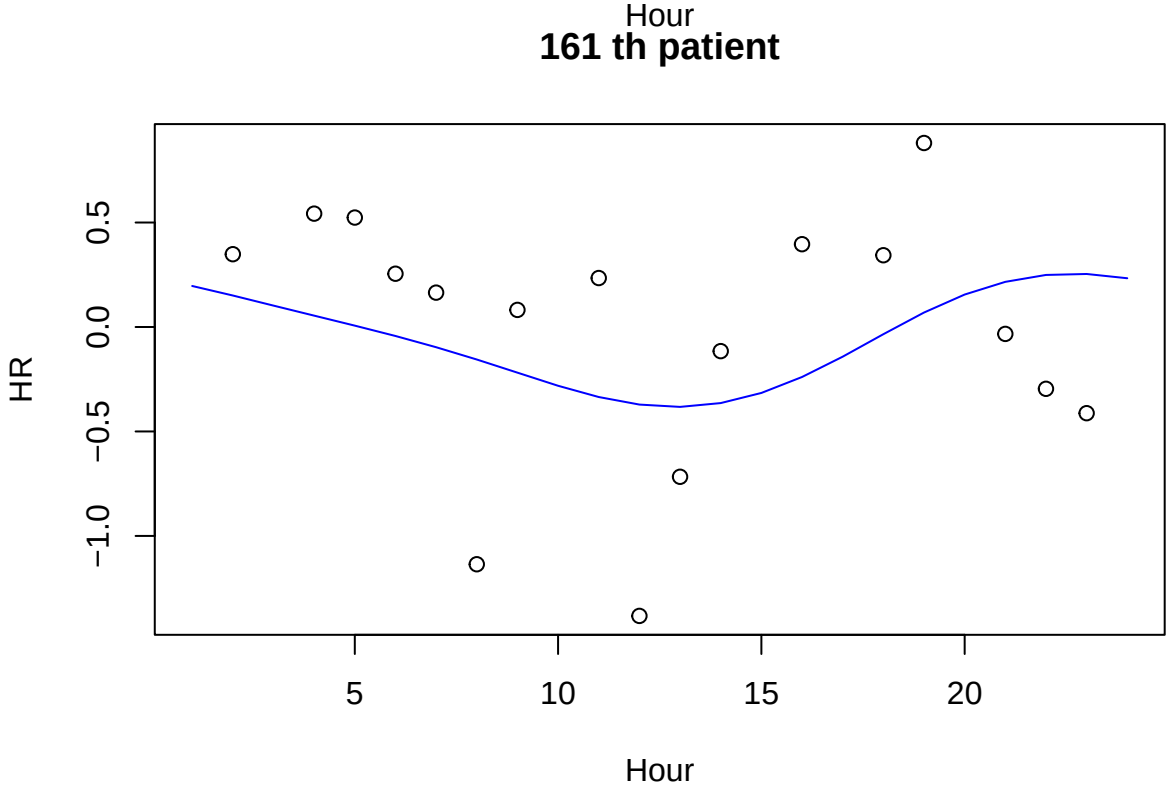

156 th patient
HR
2.0
1.5
1.0
0.5
0.0
−0.5
5
10
15
20
Hour

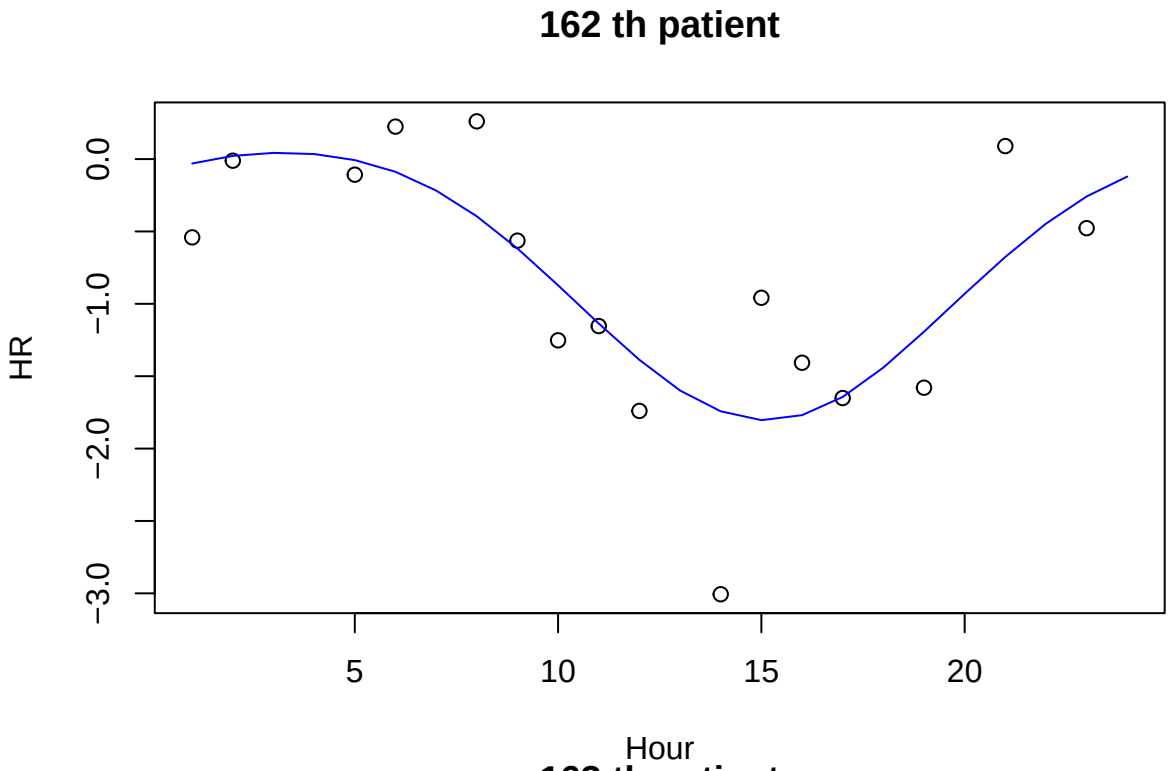

157 th patient
HR
2.0
1.5
1.0
0.5
0.0
−1.0
5
10
15
20
Hour

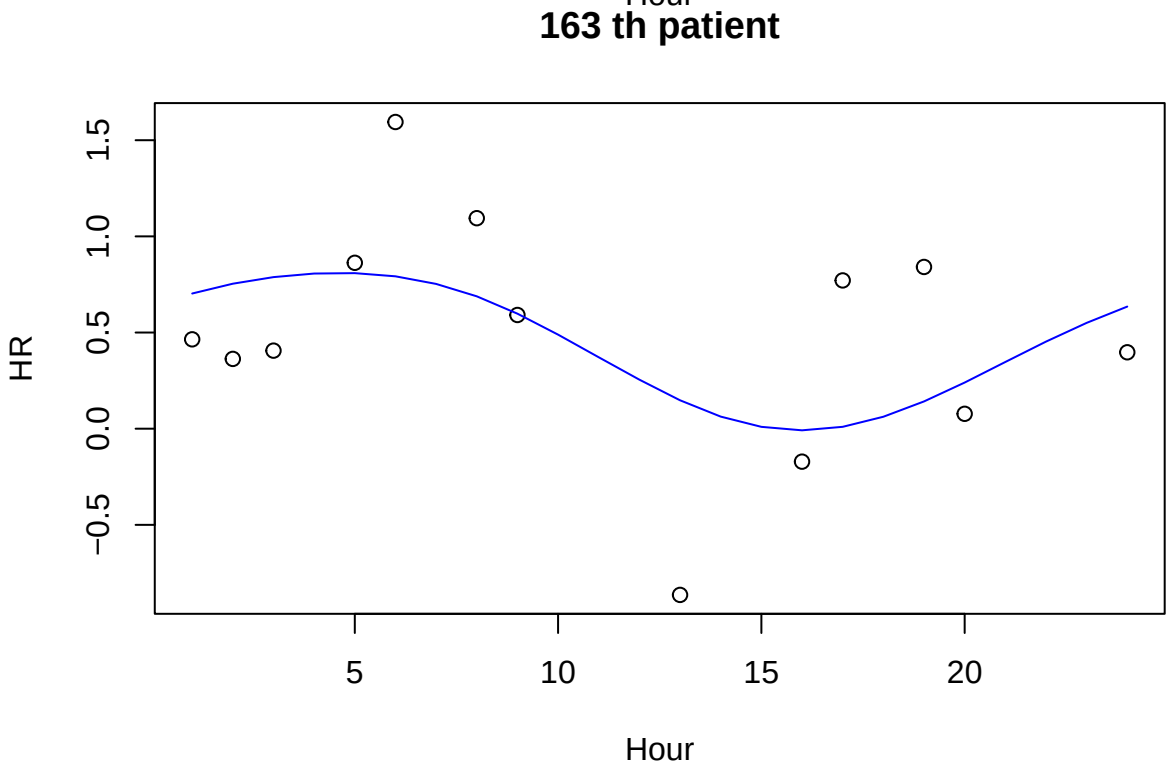

158 th patient
HR
0.0
0.5
1.0
1.5
2.0
5
10
15
20
Hour

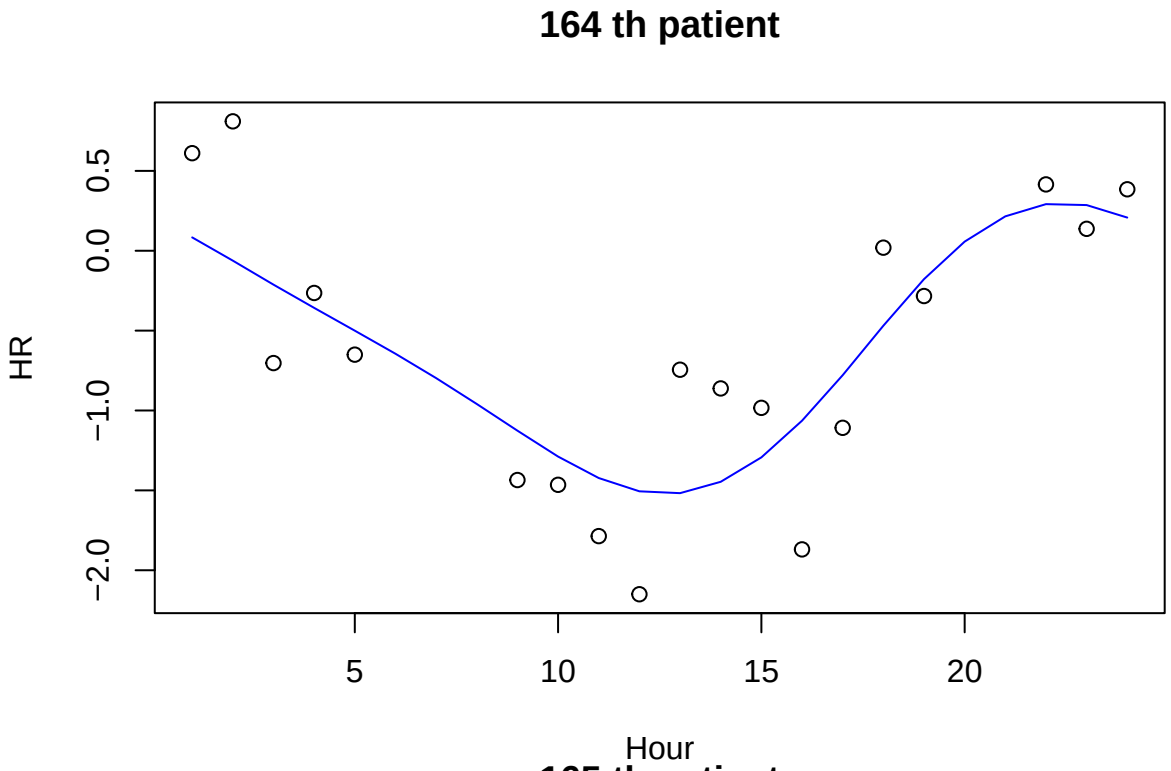

159 th patient
HR
−1.5
−0.5
0.0
0.5
1.0
5
10
15
20
Hour

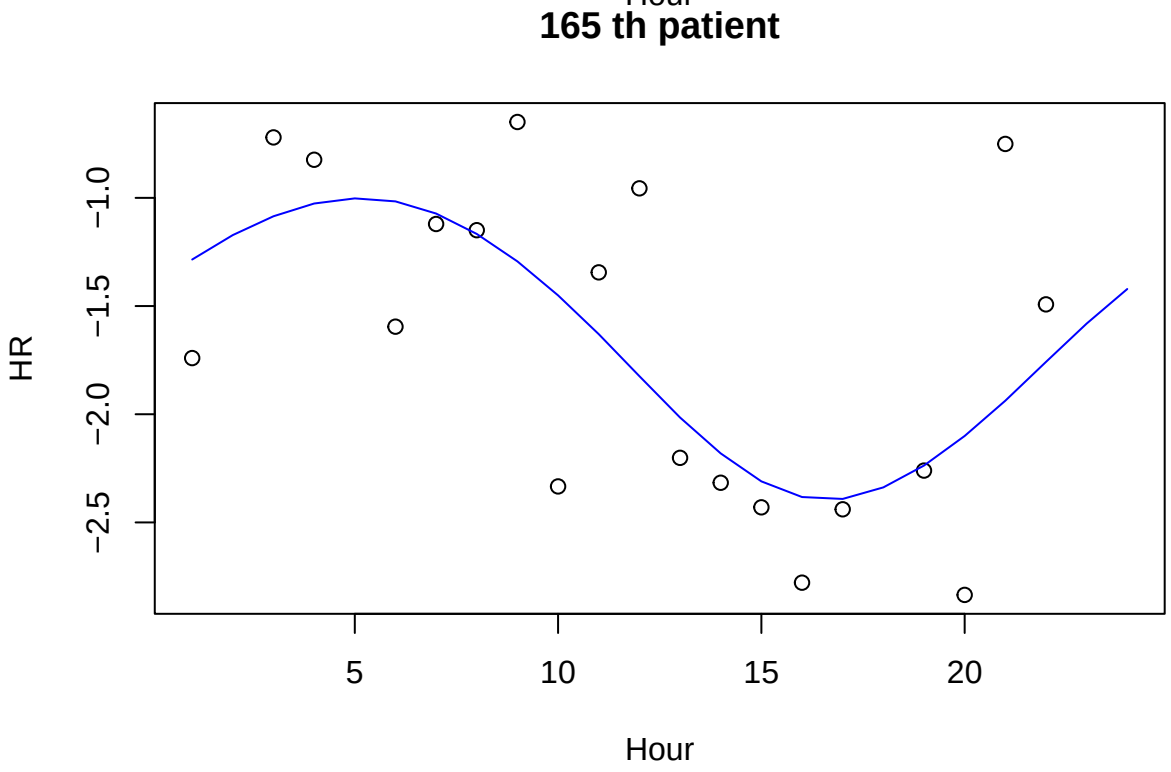

160 th patient
HR
0.0
−0.5
−1.0
−1.5
5
10
15
20
Hour

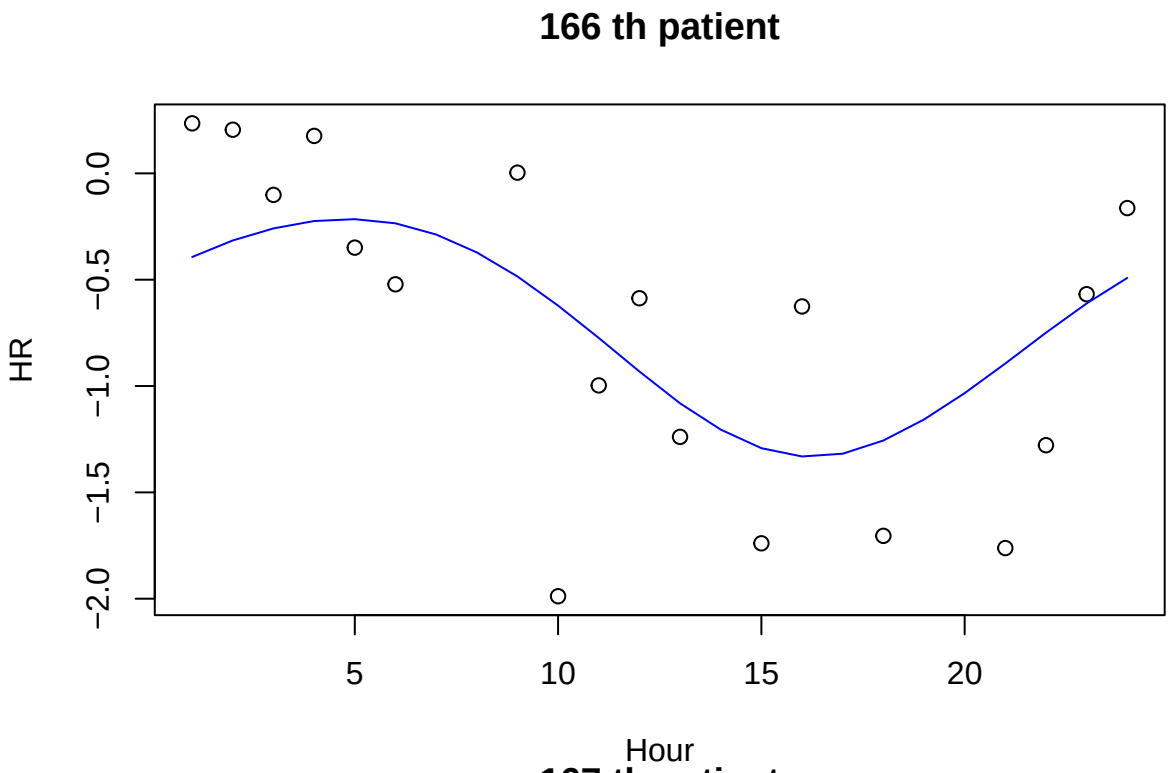

161 th patient
HR
0.5
0.0
−0.5
−1.0
5
10
15
20
Hour

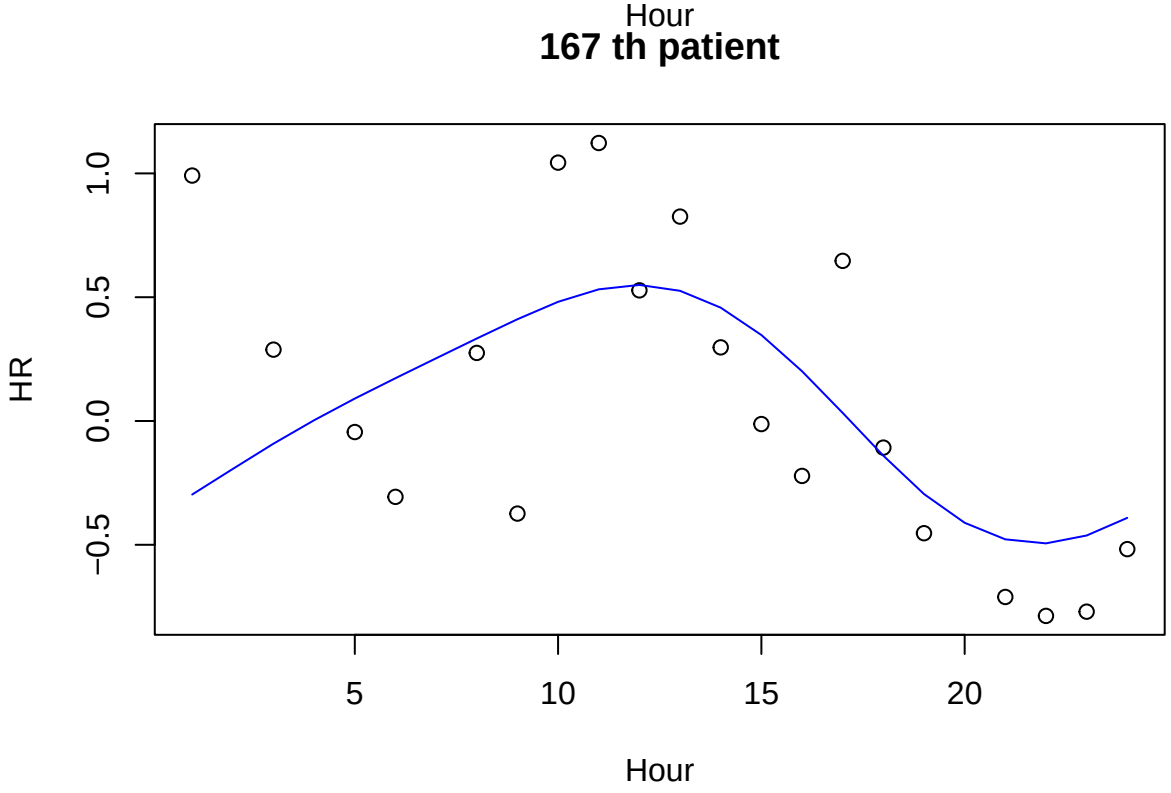

162 th patient
HR
0.0
−1.0
−2.0
−3.0
5
10
15
20
Hour

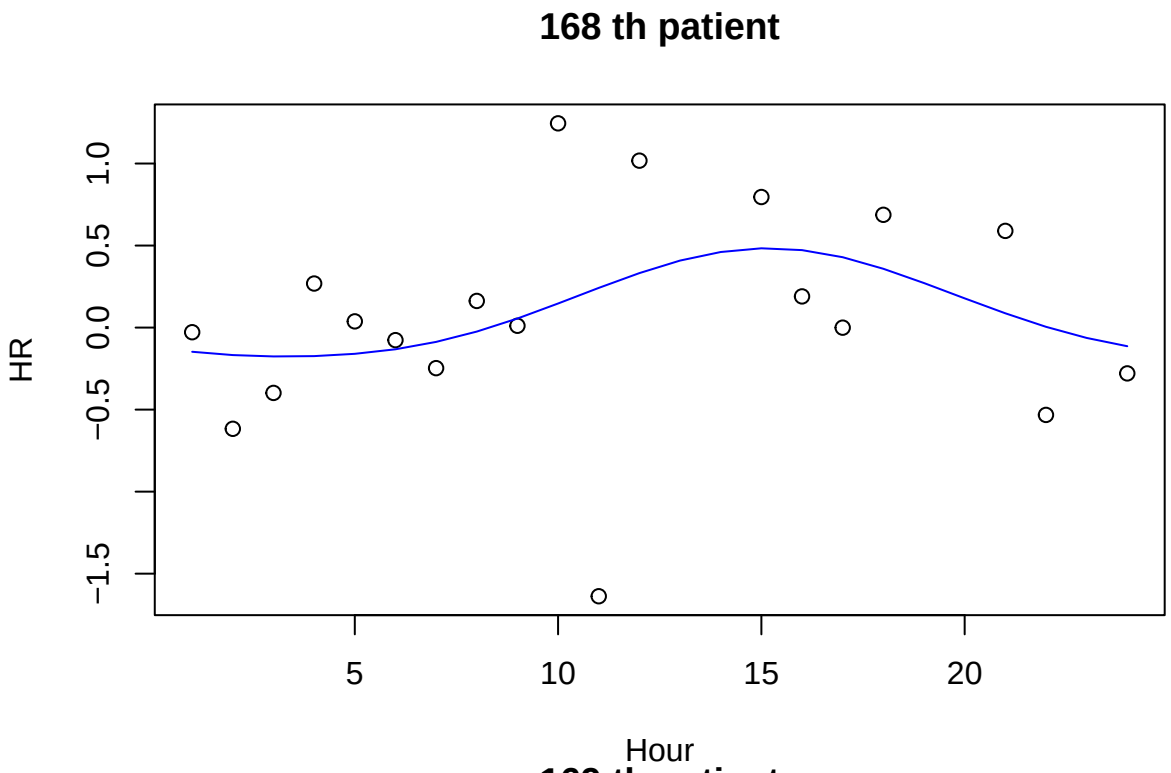

163 th patient
HR
1.5
1.0
0.5
0.0
−0.5
5
10
15
20
Hour

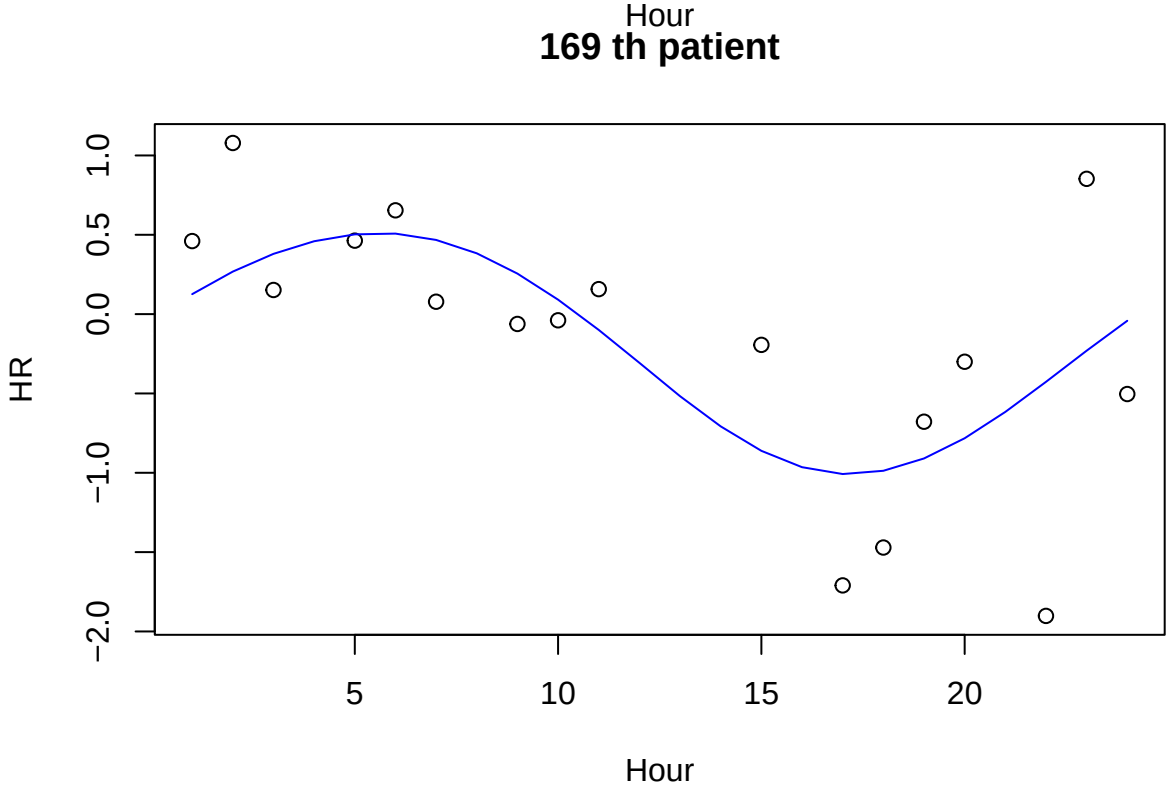

164 th patient
HR
0.5
0.0
−1.0
−2.0
5
10
15
20
Hour

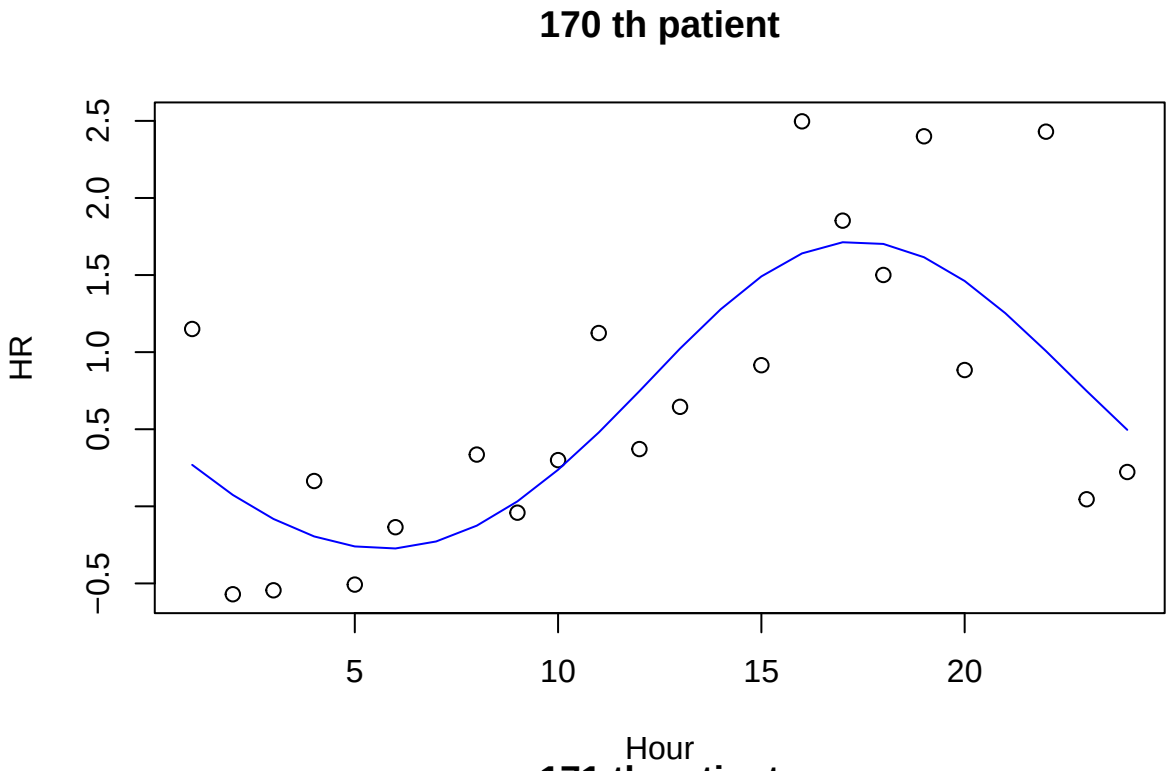

165 th patient
HR
−1.0
−1.5
−2.0
−2.5
5
10
15
20
Hour

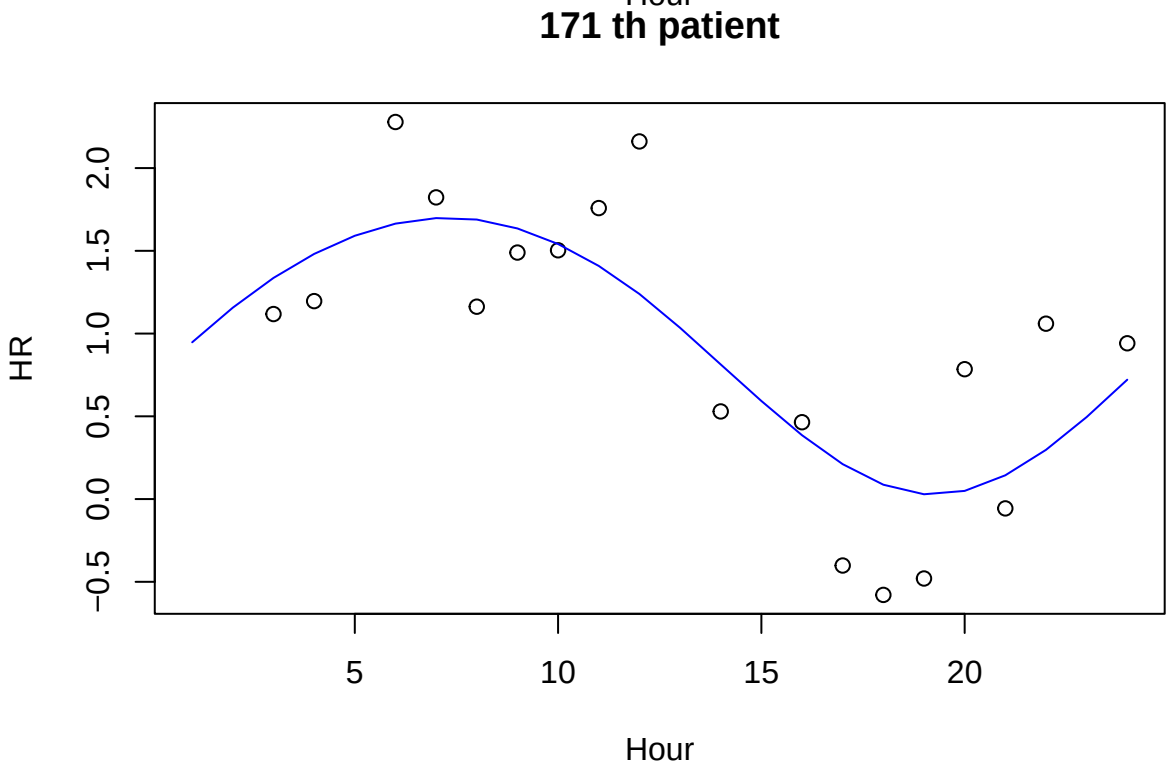

166 th patient
HR
0.0
−0.5
−1.0
−1.5
−2.0
5
10
15
20
Hour

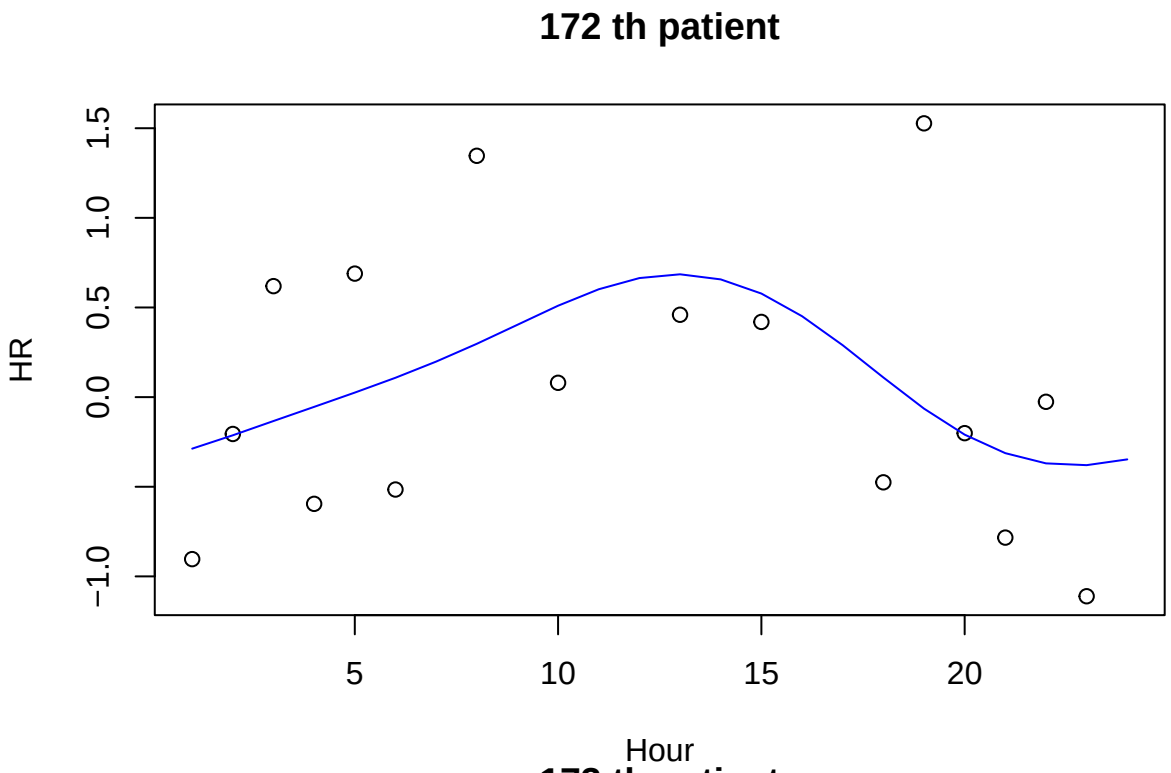

167 th patient
HR
1.0
0.5
0.0
−0.5
5
10
15
20
Hour

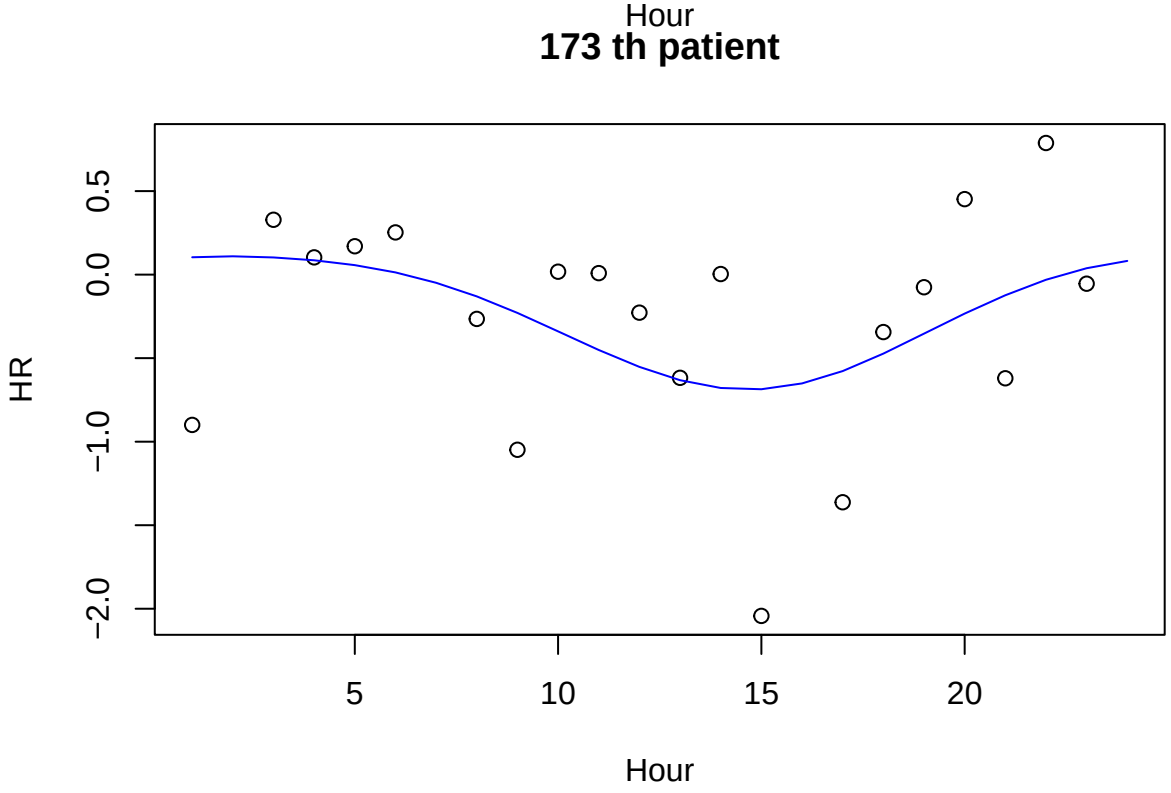

168 th patient
HR
1.0
0.5
0.0
−0.5
−1.5
5
10
15
20
Hour

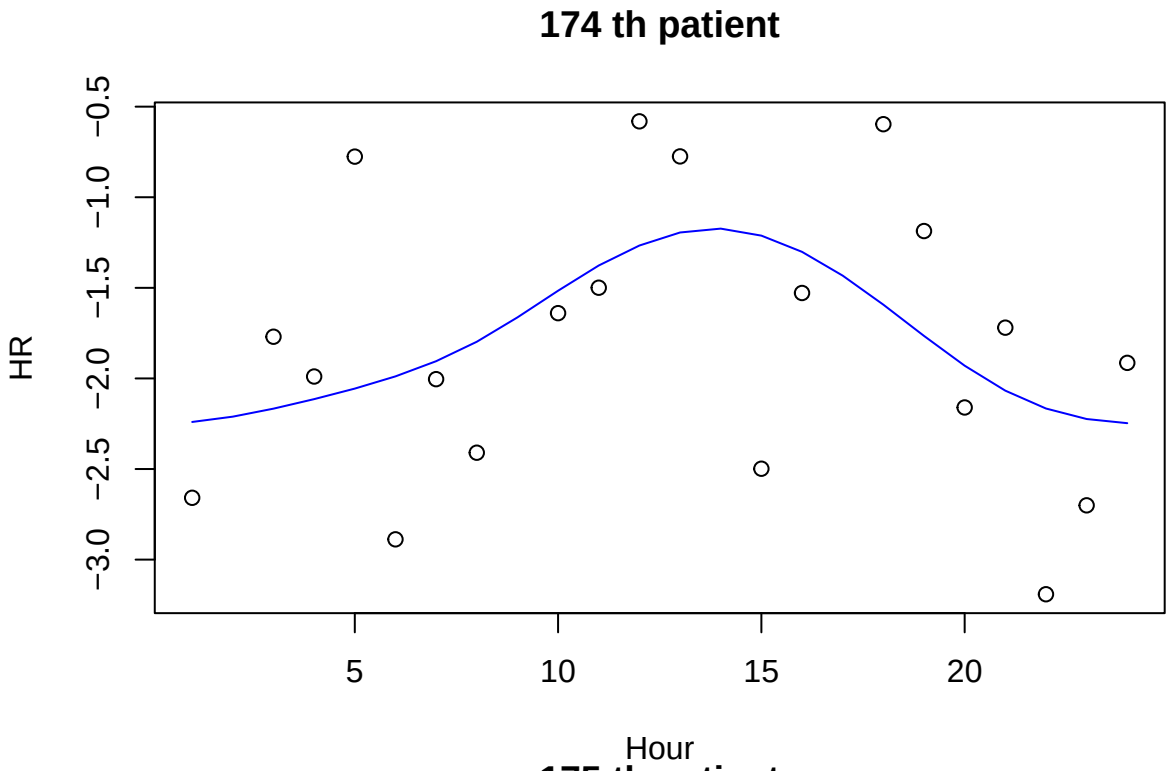

169 th patient
HR
1.0
0.5
0.0
−1.0
−2.0
5
10
15
20
Hour

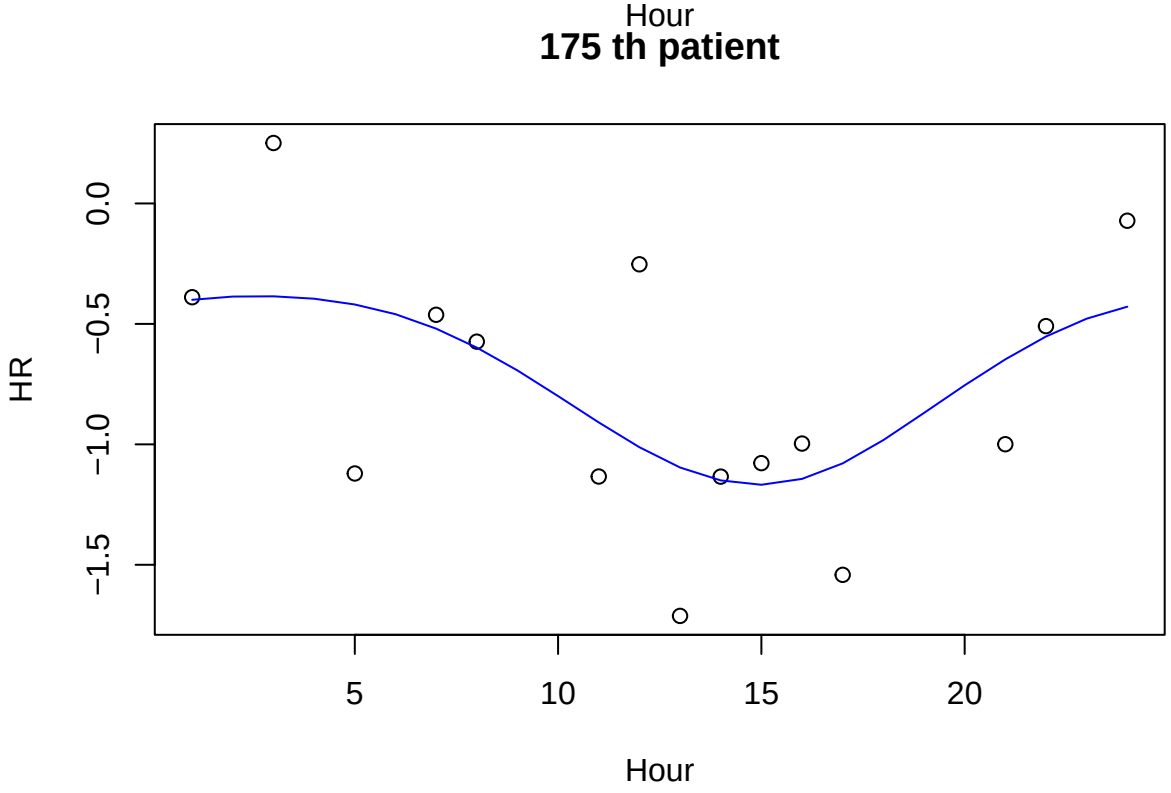

170 th patient
HR
−0.5
0.5
1.0
1.5
2.0
2.5
5
10
15
20
Hour


171 th patient
HR
−0.5
0.0
0.5
1.0
1.5
2.0
5
10
15
20
Hour


172 th patient
HR
1.5
1.0
0.5
0.0
−1.0
5
10
15
20
Hour


173 th patient
HR
0.5
0.0
−1.0
−2.0
5
10
15
20
Hour

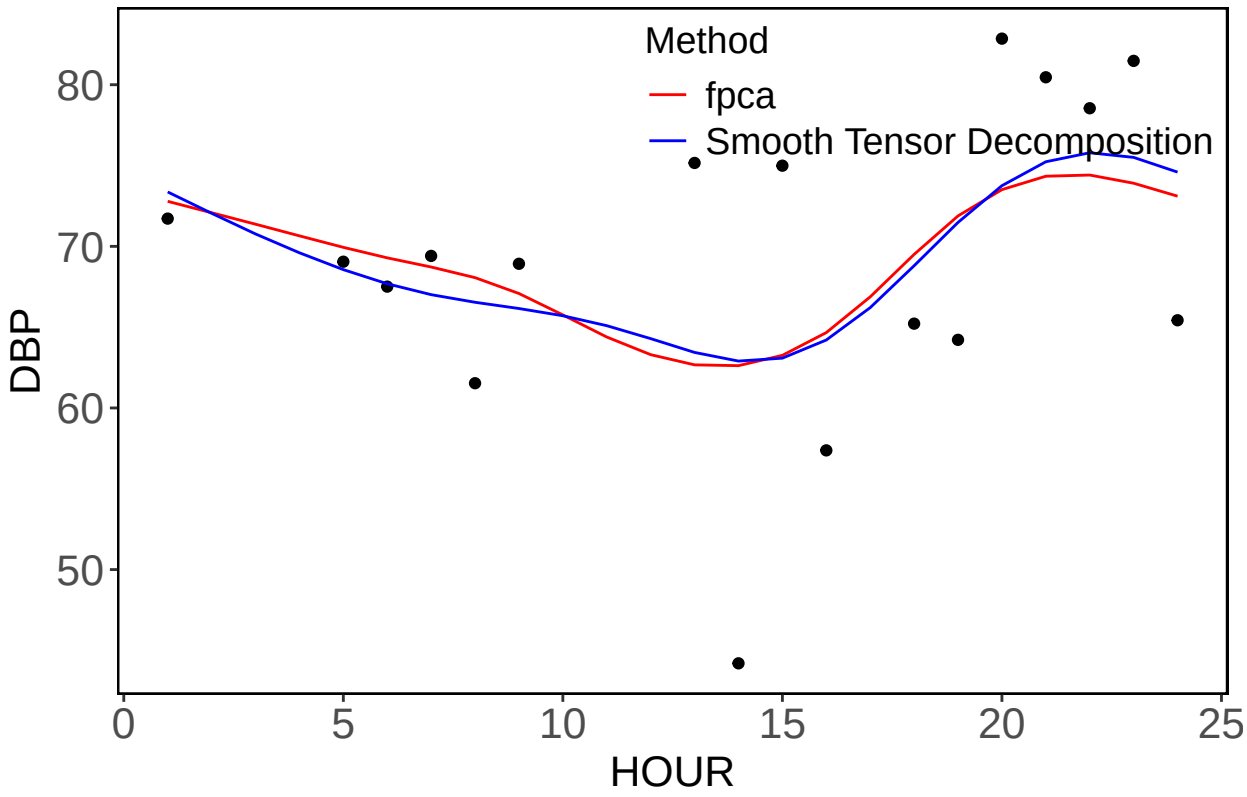




Comparison with FPCA (Example)

```
library(refund)
h1 <- rTensor::unfold(synthetic_dat[,1,], row_idx = 2, col_idx = 1)@data
```

```r
outfpca1 <- refund::fpca.sc(Y = as.matrix(h1), npc = 3, center=FALSE)

res <-  mglram(tnsr = synthetic_dat@data, ranks = c(6, 3), init=0, D = D2,
                 lambda = 12, max_iter = 500, tol = 1e-5, L0 = NULL)

for (i in c(90)){
df <- data.frame(cbind(c(1:24), (res$est[,1,i])*12+70, (outfpca1$Yhat[i,])*12+70))
colnames(df) <- c("hour", "my", "fpca")
original <- (synthetic_dat@data[ , 1, i])*12+70
org_df <- data.frame(cbind(c(1:24), original))
colnames(org_df) <- c("hour", "original")

df_long <- pivot_longer(df, cols=c("my", "fpca"), names_to = "method")

custom_colors <- c("blue","red")
names(custom_colors) <- c("my", "fpca")

p <- ggplot() +
  geom_point(data = org_df, aes(x = hour, y = original), color = "black") +  # Original data
  geom_line(data = df_long, aes(x = hour, y =value, color = method)) +
  labs(title = "",
       x = "HOUR", y = "DBP") +
  scale_color_manual(values = custom_colors,
                     labels = c("my" = "Smooth Tensor Decomposition", "fPCA" = "Functional Principal Co
                    name =  "Method") +
  theme(legend.position = c(1, 1),
    legend.justification = c(1, 1),
    panel.grid = element_blank(),
    panel.background = element_blank(),
    plot.background = element_blank(),
    panel.border = element_rect(color = "black", fill = NA, linewidth = 0.8),  # add border rectangle
    axis.line = element_blank(),
    legend.background = element_blank(),  # removes the outer background & border
    legend.key = element_blank(),
    axis.text = element_text(size = 16),
    axis.title = element_text(size = 16),
    legend.text = element_text(size = 14,hjust=0),
    legend.title = element_text(size = 14)
    )
print(p)
}
```


Method
fpca
Smooth Tensor Decomposition
DBP
HOUR
80
70
60
50
0
5
10
15
20
25