```r
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
```

```r
library(tidyr)
```

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

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

Generate synthetic data

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

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

synthetic_dat <- synthetic$sim_Mmiss

#save(synthetic_dat, file = "synthetic_data.Rda")
```

Make the second order difference matrix

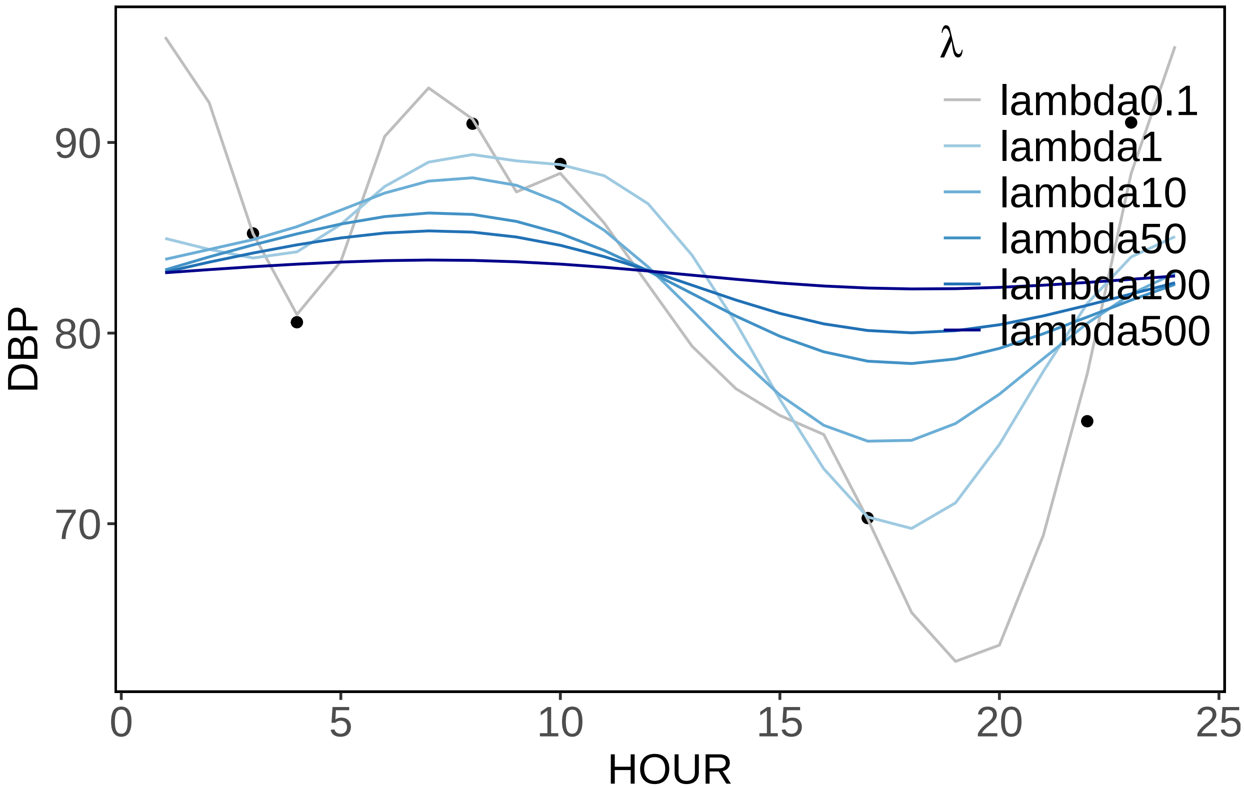
```r
D2 <- SecDiffMat(24)
```

Coarse-to-fine grid search (Example)

```r
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]
```

```r
  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$est[,1,180])*12 +70
}
org_df <- data.frame(cbind(c(1:24), original))
est_df <- data.frame(cbind(c(1:24), est_df))
colnames(org_df) <- c("hour", "original")
colnames(est_df) <- c("hour", paste0("lambda",lambda_seq))
est_long <- pivot_longer(est_df, cols = starts_with("lambda"),
                         names_to = "lambda", values_to = "y")
est_long$lambda <- factor(est_long$lambda, levels=c("lambda0.1","lambda1","lambda10","lambda50","lambda
```

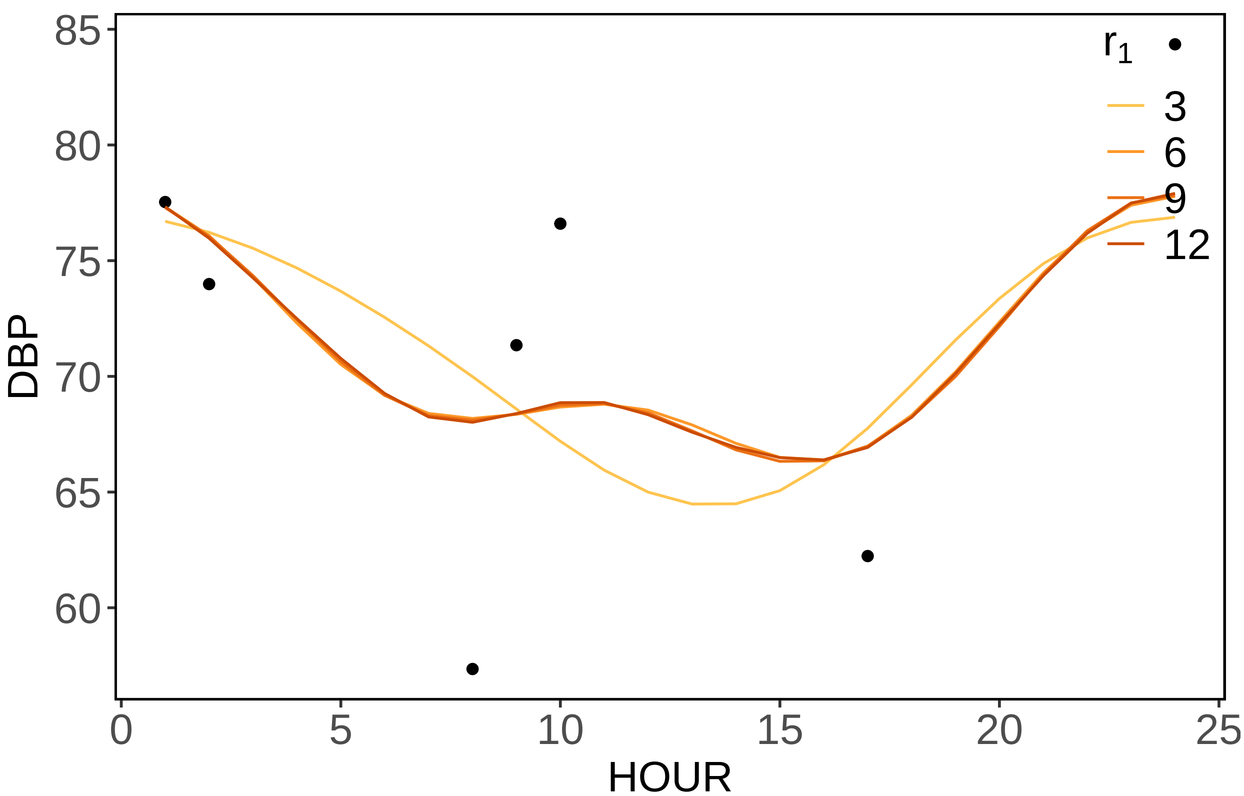
```r
custom_colors <- c("grey","#9ECAE1","#6BAED6", "#4292C6", "#2171B5","darkblue")
names(custom_colors) <- levels(factor(est_long$lambda))

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)
    )
```




```r
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))
```

```r
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, LO_ =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 8.2455 mins
```

```r
kcv_res
```

```
## $MSE_mat
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## [1,] 0.4942339 0.4882951 0.4848214 0.4824064 0.4805549 0.4790842 0.4778581
## [2,] 0.5172700 0.5017996 0.4934665 0.4876554 0.4834435 0.4802149 0.4776675
## [3,] 0.5397529 0.5177960 0.5060908 0.4985144 0.4931221 0.4890912 0.4859702
## [4,] 0.5447999 0.5176707 0.5047372 0.4968205 0.4913981 0.4874115 0.4843402
## [5,] 0.5646849 0.5518426 0.5443432 0.5390976 0.5351042 0.5318821 0.5292060
## [6,] 0.6075049 0.5785536 0.5635194 0.5539042 0.5470290 0.5417890 0.5377811
## [7,] 0.6315316 0.5956107 0.5765610 0.5641732 0.5552948 0.5485696 0.5432737
## [8,] 0.6228361 0.5833545 0.5645573 0.5528576 0.5447089 0.5386532 0.5339401
##           [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
## [1,] 0.4768395 0.4759561 0.4752172 0.4745559 0.4740048 0.4735143 0.4731151
## [2,] 0.4756702 0.4741165 0.4728828 0.4719019 0.4710991 0.4704462 0.4699209
## [3,] 0.4835067 0.4814981 0.4798734 0.4785218 0.4774181 0.4764862 0.4757101
## [4,] 0.4819339 0.4799920 0.4784129 0.4771001 0.4760126 0.4751024 0.4743490
## [5,] 0.5269268 0.5249520 0.5232362 0.5217300 0.5203963 0.5191894 0.5181516
## [6,] 0.5342709 0.5314099 0.5289830 0.5269053 0.5251284 0.5235871 0.5222183
## [7,] 0.5389780 0.5354016 0.5324249 0.5298842 0.5277019 0.5258031 0.5241486
## [8,] 0.5301958 0.5271173 0.5245666 0.5224184 0.5205874 0.5190082 0.5176714
##          [,15]     [,16]     [,17]     [,18]     [,19]     [,20]     [,21]
## [1,] 0.4727614 0.4724730 0.4722266 0.4720378 0.4718797 0.4717680 0.4716799
## [2,] 0.4694822 0.4691425 0.4688775 0.4686918 0.4685456 0.4684560 0.4684089
## [3,] 0.4750601 0.4745289 0.4740963 0.4737409 0.4734643 0.4732420 0.4730740
## [4,] 0.4737209 0.4731902 0.4727625 0.4724075 0.4721291 0.4719069 0.4717379
## [5,] 0.5172068 0.5163561 0.5156091 0.5149405 0.5143398 0.5138076 0.5133326
## [6,] 0.5210053 0.5199790 0.5190328 0.5181936 0.5174681 0.5168314 0.5162542
## [7,] 0.5227179 0.5214489 0.5203059 0.5193289 0.5184648 0.5176815 0.5169988
## [8,] 0.5165079 0.5154802 0.5146078 0.5138253 0.5131669 0.5125817 0.5120774
##          [,22]     [,23]     [,24]     [,25]     [,26]     [,27]     [,28]
## [1,] 0.4716344 0.4716159 0.4716135 0.4716544 0.4717160 0.4717882 0.4718977
## [2,] 0.4684047 0.4684417 0.4685007 0.4685901 0.4687281 0.4688644 0.4690402
## [3,] 0.4729560 0.4728888 0.4728515 0.4728489 0.4729012 0.4729589 0.4730520
## [4,] 0.4716161 0.4715401 0.4715022 0.4714986 0.4715397 0.4715957 0.4716776
## [5,] 0.5129286 0.5125541 0.5122236 0.5119387 0.5117244 0.5115122 0.5113530
## [6,] 0.5157543 0.5153130 0.5149354 0.5145957 0.5143002 0.5140948 0.5138977
## [7,] 0.5163957 0.5158687 0.5153975 0.5149828 0.5146329 0.5143332 0.5140763
## [8,] 0.5116609 0.5112816 0.5109749 0.5107194 0.5105068 0.5103296 0.5101900
##          [,29]     [,30]
## [1,] 0.4720135 0.4721507
## [2,] 0.4692211 0.4694247
## [3,] 0.4731735 0.4733121
## [4,] 0.4717999 0.4719271
## [5,] 0.5112157 0.5111004
## [6,] 0.5137148 0.5135921
```

```
## [7,] 0.5138644 0.5136768
## [8,] 0.5101109 0.5100463
##
## $SE_mat
##             [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
## [1,] 0.005441095 0.005126904 0.004894346 0.004716007 0.004561539 0.004454061
## [2,] 0.006754331 0.006472516 0.006057254 0.005987266 0.005959665 0.005960373
## [3,] 0.010396753 0.008703820 0.007766835 0.007196275 0.006826144 0.006592351
## [4,] 0.004306314 0.005201497 0.005731699 0.006057875 0.006273192 0.006422112
## [5,] 0.009345320 0.007799071 0.006968207 0.006415920 0.005999329 0.005697511
## [6,] 0.012242430 0.010955413 0.010117427 0.009482061 0.008988382 0.008567198
## [7,] 0.011374452 0.011994112 0.012248611 0.012351412 0.012378504 0.012372032
## [8,] 0.014173052 0.013603698 0.013402407 0.013261999 0.013154587 0.013066553
##             [,7]        [,8]        [,9]       [,10]       [,11]       [,12]
## [1,] 0.004346959 0.004277600 0.004205694 0.004153980 0.004098417 0.004057039
## [2,] 0.005970609 0.005963952 0.005935055 0.005893712 0.005846181 0.005805135
## [3,] 0.006445750 0.006366711 0.006323264 0.006318900 0.006332458 0.006370691
## [4,] 0.006529540 0.006618656 0.006688559 0.006742665 0.006793293 0.006835874
## [5,] 0.005454150 0.005250234 0.005078567 0.004944907 0.004824667 0.004729986
## [6,] 0.008244075 0.007923551 0.007669462 0.007438637 0.007237974 0.007058415
## [7,] 0.012347304 0.012300821 0.012268041 0.012236323 0.012182921 0.012144148
## [8,] 0.012962491 0.012880849 0.012801890 0.012724842 0.012654687 0.012586364
##            [,13]       [,14]       [,15]       [,16]       [,17]       [,18]
## [1,] 0.004023603 0.003989926 0.003961850 0.003935295 0.003911812 0.003898461
## [2,] 0.005763585 0.005728812 0.005696210 0.005670292 0.005647141 0.005622140
## [3,] 0.006414941 0.006469192 0.006528194 0.006596789 0.006667796 0.006740017
## [4,] 0.006874762 0.006916430 0.006954355 0.006986019 0.007020100 0.007057003
## [5,] 0.004650625 0.004587723 0.004544206 0.004488279 0.004454090 0.004426432
## [6,] 0.006906551 0.006772223 0.006652168 0.006545213 0.006448524 0.006349873
## [7,] 0.012125539 0.012086745 0.012039896 0.012015330 0.011980857 0.011951720
## [8,] 0.012518977 0.012451178 0.012398279 0.012337741 0.012282895 0.012238409
##            [,19]       [,20]       [,21]       [,22]       [,23]       [,24]
## [1,] 0.003879573 0.003863083 0.003848384 0.003839564 0.003828464 0.003818733
## [2,] 0.005605447 0.005588757 0.005576635 0.005569075 0.005556522 0.005547876
## [3,] 0.006817035 0.006890816 0.006964632 0.007036660 0.007113050 0.007184965
## [4,] 0.007089335 0.007121154 0.007152379 0.007183044 0.007212852 0.007238352
## [5,] 0.004405648 0.004388994 0.004377005 0.004383072 0.004378864 0.004377760
## [6,] 0.006275971 0.006202373 0.006128582 0.006067772 0.006012284 0.005965414
## [7,] 0.011923684 0.011903483 0.011878070 0.011853694 0.011829384 0.011807197
## [8,] 0.012183953 0.012137159 0.012097969 0.012057030 0.012012792 0.011984458
##            [,25]       [,26]       [,27]       [,28]       [,29]       [,30]
## [1,] 0.003811008 0.003804056 0.003803693 0.003793513 0.003794922 0.003790193
## [2,] 0.005539264 0.005534330 0.005529614 0.005526790 0.005523814 0.005518994
## [3,] 0.007255835 0.007332035 0.007400559 0.007471283 0.007540223 0.007608221
## [4,] 0.007266965 0.007302473 0.007330102 0.007353411 0.007387143 0.007409453
## [5,] 0.004378759 0.004387074 0.004391507 0.004401499 0.004412560 0.004433647
## [6,] 0.005919235 0.005877017 0.005842843 0.005805288 0.005772975 0.005750410
## [7,] 0.011786013 0.011778638 0.011754988 0.011729259 0.011723148 0.011702856
## [8,] 0.011948201 0.011915786 0.011881696 0.011860948 0.011831561 0.011811291
##
## $opt_para
##      [,1] [,2] [,3]
## [1,]    4    2   22
```

Check explained variability

```
opt_res <- mglram(tnsr = synthetic_dat@data, ranks = c(4, 2), init=0, D = D2,
       lambda = 22, max_iter = 500, tol = 1e-5, L0 = NULL)
```

```
# Cumulative explained variability when r1 increases
tilde <- MakeIdent(opt_res$L, opt_res$G, opt_res$R)
L_tilde <- tilde$L_tilde
G_tilde <- tilde$G_tilde
R_tilde <- tilde$R_tilde

nmiss_idx <- which(!is.na(synthetic_dat@data))

cum_var_L <- rep(NA, 4)
for (i in 1:4){
  comp <- array(NA, c(24,3,207))
  for (j in 1:207){
    comp[ , , j] <- matrix(L_tilde[, 1:i], ncol=i) %*% matrix(G_tilde[1:i, , j], nrow=i) %*% t(R_tilde)
  }
  cum_var_L[i] <- sum(comp[nmiss_idx]^2)/sum(synthetic_dat@data[nmiss_idx]^2)
}

cum_var_L # 0.4472346 0.5592966 0.6085780 0.6106854
```

```
## [1] 0.4472346 0.5592966 0.6085780 0.6106854
```

```
# Separate explained variability for different L ranks
cum_var_L0 <- c(0, cum_var_L[1:3])
sep_var_L <- cum_var_L - cum_var_L0

sep_var_L # 0.447234584 0.112062011 0.049281371 0.002107439
```

```
## [1] 0.447234584 0.112062011 0.049281371 0.002107439
```

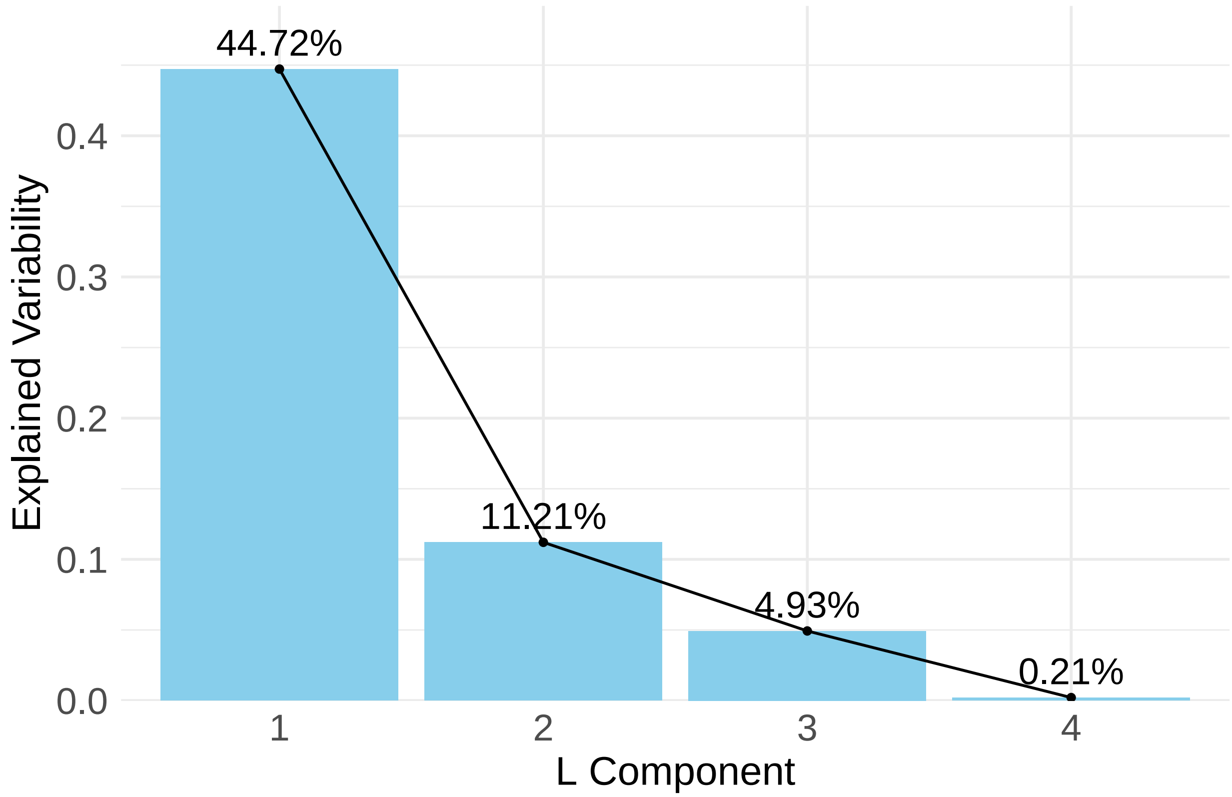
```
# Cumulative explained variability when r2 increases
cum_var_R <- rep(NA, 2)
for (i in 1:2){
  comp <- array(NA, dim(synthetic_dat@data))
  for (j in 1:207){
    comp[ , , j] <- L_tilde %*% matrix(G_tilde[,1:i, j], ncol=i) %*% matrix(t(R_tilde[, 1:i]), nrow=i)
  }
  cum_var_R[i] <- sum(comp[nmiss_idx]^2)/sum(synthetic_dat@data[nmiss_idx]^2)
}

cum_var_R # 0.3812621 0.6106854
```

```
## [1] 0.3812621 0.6106854
```

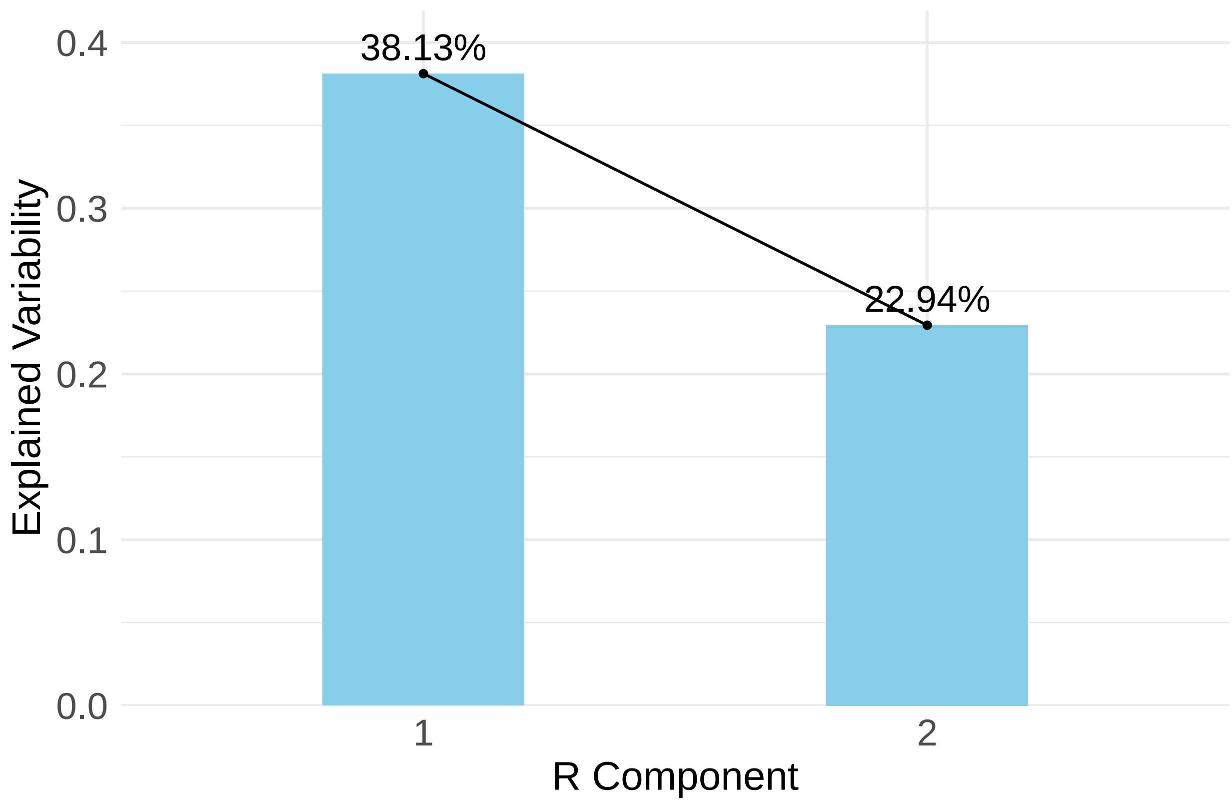
```r
# Separate explained variability for different R ranks
cum_var_R0 <- c(0, cum_var_R[1:1])
sep_var_R <- cum_var_R - cum_var_R0

sep_var_R # 0.3812621 0.2294233
```

```
## [1] 0.3812621 0.2294233
```

```r
# Plot the explained variability
var_data_L <- data.frame(
  component = factor(seq_along(sep_var_L), levels = seq_along(sep_var_L)),
  value = sort(sep_var_L, decreasing = TRUE)
)

var_data_L$percentage <- paste0(round(var_data_L$value * 100, 2), "%")

ggplot(var_data_L, aes(x = component, y = value)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  geom_text(aes(label = percentage), vjust = -0.5, size = 5) +
  geom_line(aes(group = 1), color = "black", linewidth = 0.5) +
  geom_point(color = "black", size = 1) +
  labs(y = "Explained Variability", x="L Component", title="") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.title.y = element_text(size = 15),
        axis.title.x = element_text(size = 15),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 14)) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.1)))
```




```r
# Plot the explained variability
var_data_R <- data.frame(
  component = factor(seq_along(sep_var_R), levels = seq_along(sep_var_R)),
  value = sort(sep_var_R, decreasing = TRUE)
)

var_data_R$percentage <- paste0(round(var_data_R$value * 100, 2), "%")

ggplot(var_data_R, aes(x = component, y = value)) +
  geom_bar(stat = "identity", fill = "skyblue", width = 0.4) +
  geom_text(aes(label = percentage), vjust = -0.5, size = 5) +
  geom_line(aes(group = 1), color = "black", linewidth = 0.5) +
  geom_point(color = "black", size = 1) +
  labs(title = "", x = "R Component", y = "Explained Variability") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.title.y = element_text(size = 15),
        axis.title.x = element_text(size = 15),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 14)) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.1)))
```



Update hyperparameters for parsimony

```
start_time <- Sys.time()

res <- mglram(tnsr = synthetic_dat@data, ranks = c(3, 2), init=0, D = D2,
       lambda = 24, max_iter = 500, tol = 1e-5, L0 = NULL)

res$conv
```

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

```
end_time <- Sys.time()

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

```
## Time difference of 0.6714299 secs
```

```
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,] -0.2204613  0.20790067 -0.158708877
##  [2,] -0.2187223  0.24239650 -0.090922030
##  [3,] -0.2162770  0.26192713 -0.020413022
##  [4,] -0.2132307  0.26587141  0.048651843
##  [5,] -0.2097717  0.25367011  0.112952800
##  [6,] -0.2059251  0.22552909  0.169652718
##  [7,] -0.2017157  0.18184422  0.216097005
##  [8,] -0.1972099  0.12438980  0.251419427
##  [9,] -0.1927431  0.05559529  0.274333297
## [10,] -0.1890365 -0.02037933  0.283497019
## [11,] -0.1864162 -0.09795309  0.276680286
## [12,] -0.1850316 -0.17091767  0.252059441
## [13,] -0.1849966 -0.23328148  0.209148229
## [14,] -0.1863944 -0.27970048  0.149603041
## [15,] -0.1891757 -0.30586134  0.076973881
## [16,] -0.1929878 -0.30857926 -0.003751707
## [17,] -0.1976431 -0.28788620 -0.086259446
## [18,] -0.2026861 -0.24615115 -0.163124479
## [19,] -0.2077854 -0.18806534 -0.226761346
## [20,] -0.2125947 -0.11930103 -0.270883649
## [21,] -0.2166590 -0.04505208 -0.291904259
## [22,] -0.2194912  0.02939243 -0.288916086
## [23,] -0.2210065  0.09908498 -0.263400790
## [24,] -0.2213056  0.15954759 -0.218227108
```

```
R_tilde
```

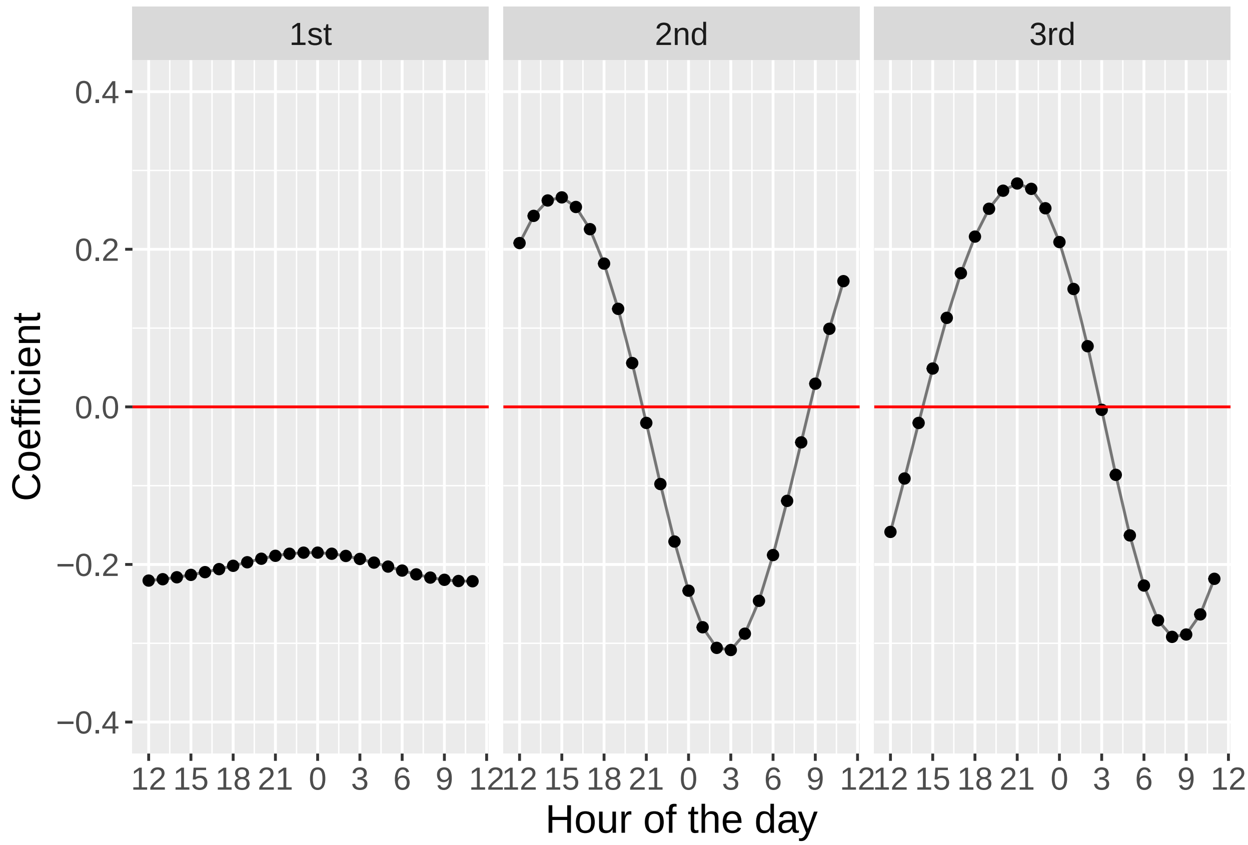
```
##             [,1]       [,2]
## [1,] -0.6205801  0.2656008
## [2,] -0.5411394  0.5358726
## [3,] -0.5674932 -0.8014342
```

```
G_tilde[ , ,1:5]
```

```
## , , 1
##
##             [,1]       [,2]
## [1,] -0.7964393  0.2822846
## [2,] -3.8410252 -0.4805376
## [3,] -0.1891807  0.2660925
##
## , , 2
##
##             [,1]       [,2]
## [1,]  2.9733025  8.4163397
## [2,] -0.9184376 -0.5826965
## [3,]  2.0209947 -0.1556882
##
## , , 3
##
##             [,1]        [,2]
## [1,]  2.9257257 -1.33227227
```

```
## [2,] -2.1303616  1.54452017
## [3,] -0.1153615 -0.03559759
##
## , , 4
##
##           [,1]      [,2]
## [1,] -7.453151 -4.399917
## [2,] -3.310160  2.266604
## [3,] -1.036860  1.017397
##
## , , 5
##
##            [,1]       [,2]
## [1,] -0.3813837  2.1377674
## [2,] -5.8909622  1.9275214
## [3,] -0.5018295 -0.3126235
```

```r
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] -0.6205801 -0.5411394 -0.5674932
```

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

```
## [1]  0.2656008  0.5358726 -0.8014342
```

```
## 36/100 DBP + 31/50 SBP + 33/100 HR (Seem to be joint effect of the three)
R_tilde[1, 1]/sum(R_tilde[, 1]) # DBP: 36/100
```

```
## [1] 0.3588801
```

```
R_tilde[2, 1]/sum(R_tilde[, 1]) # SBP: 31/100
```

```
## [1] 0.3129398
```

```
R_tilde[3, 1]/sum(R_tilde[, 1]) # HR: 33/100
```

```
## [1] 0.3281801
```

```r
## 17/100 DBP + 33/100 SBP - 50/100 HR (Contrast between blood pressure and heart rate)
R_tilde[1, 2]/sum(abs(R_tilde[, 2])) # DBP: 17/100
```

```
## [1] 0.1656994
```

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

```
## [1] 0.3343129
```

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

```
## [1] -0.4999877
```

```r
# Prepare to plot effect of time components
n = 207

Ri = array(NA, dim = c(3, 3, n))
for (i in 1:n){
  Ri[, , i] = G_tilde[, , i] %*% t(R_tilde)
}

RiL1 = Ri[1, , ]
apply(RiL1, 1, sd) # 2.789581 2.997533 3.661661 (sd for DBP, SBP, and HR for 1st time component)
```

```
## [1] 2.979859 3.102938 3.972821
```

```r
RiL2 = Ri[2, , ]
apply(RiL2, 1, sd) # 1.379921 1.401255 1.444989 (sd for DBP, SBP, and HR for 2nd time component)
```

```
## [1] 1.104060 1.158456 1.300945
```

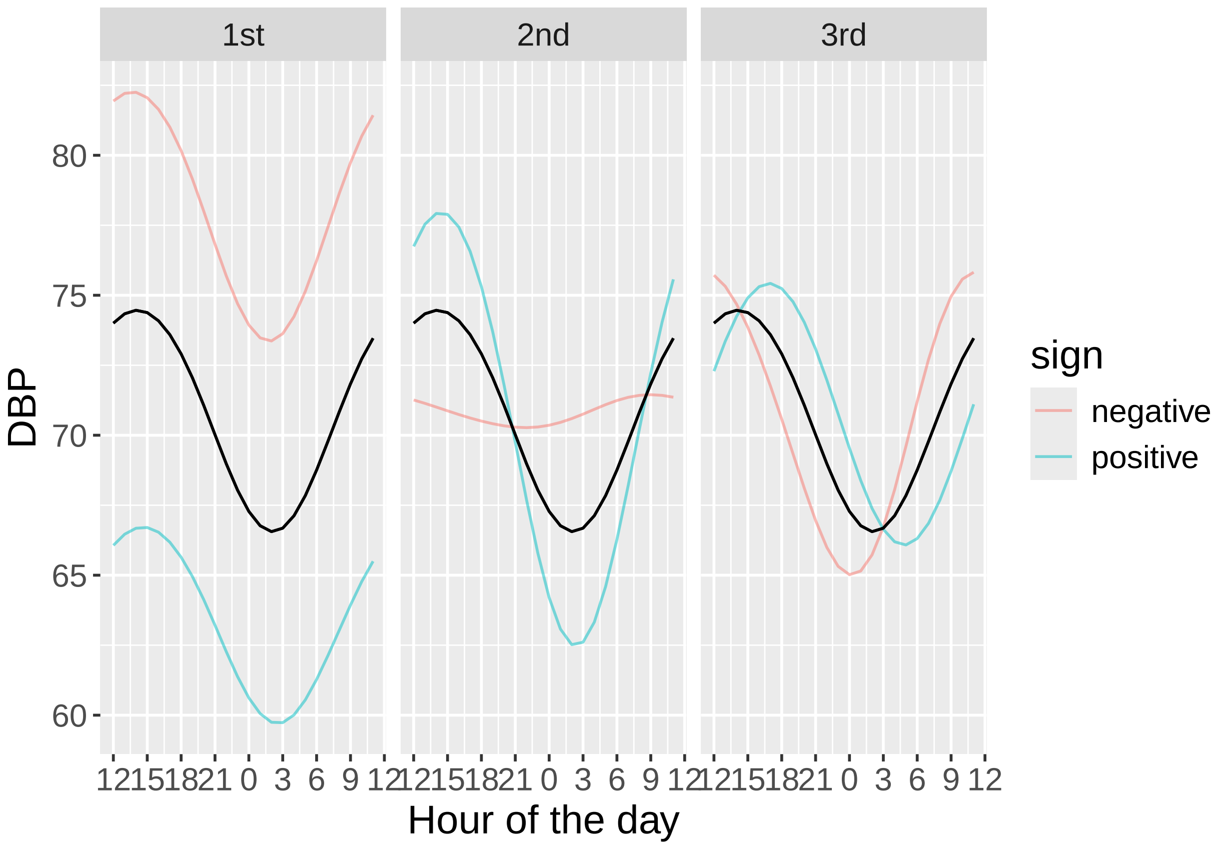
```r
RiL3 = Ri[3, , ]
apply(RiL3, 1, sd) # 1.067016 1.115904 1.072485 (sd for DBP, SBP, and HR for 3rd time component)
```

```
## [1] 0.9594026 1.0334819 1.1442265
```

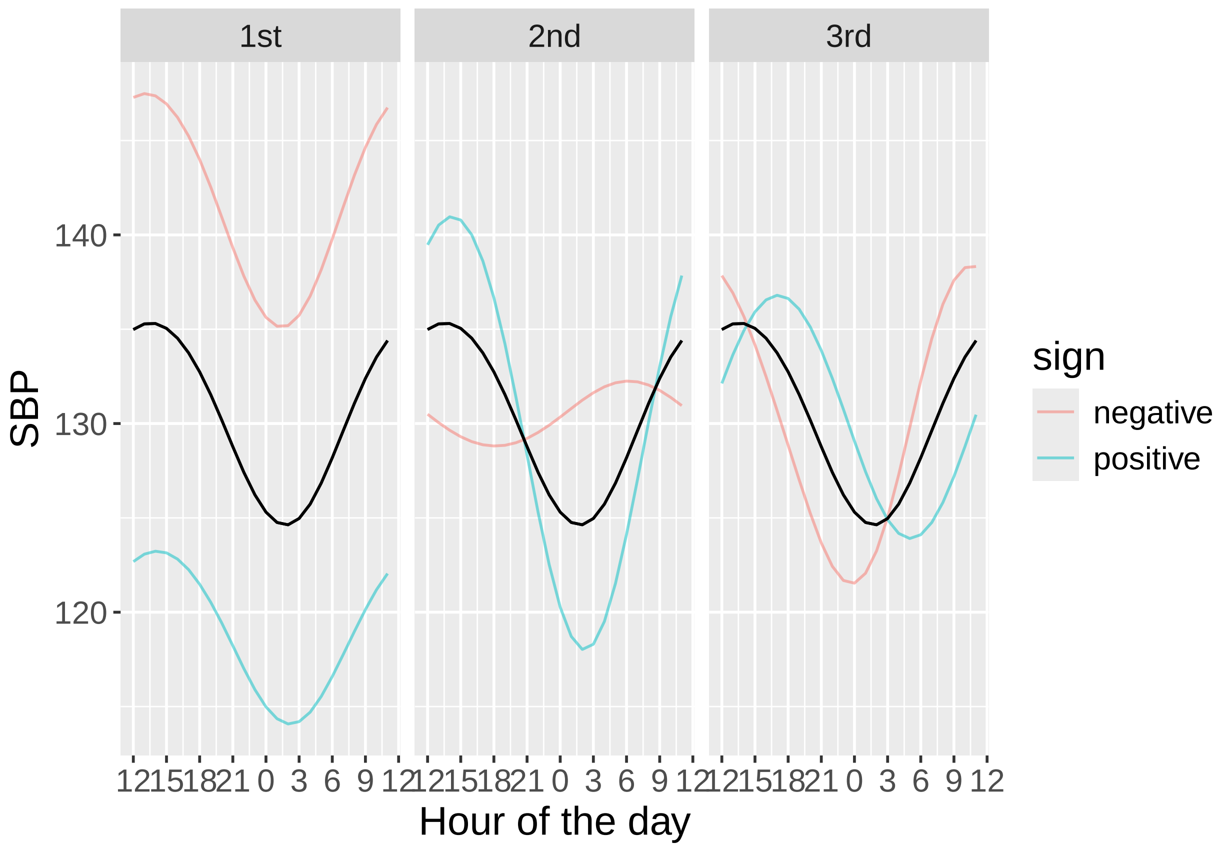
```r
DBPmean = rep(70, 24) + (L_tilde[, 1] * mean(RiL1[1,]) + L_tilde[, 2] * mean(RiL2[1,]) + L_tilde[, 3] *
SBPmean = rep(129, 24) + (L_tilde[, 1] * mean(RiL1[2,]) + L_tilde[, 2] * mean(RiL2[2,]) + L_tilde[, 3]
HRmean = rep(76, 24) + (L_tilde[, 1] * mean(RiL1[3,]) + L_tilde[, 2] * mean(RiL2[3,]) + L_tilde[, 3] *
```

```r
# Plot effect of time components on DBP
dataLeffects = data.frame(DBPmean = rep(DBPmean, 6),
                          value = c(DBPmean + L_tilde[, 1] * 3*12, DBPmean - L_tilde[, 1] * 3*12,
                                    DBPmean + L_tilde[, 2] *1.1*12, DBPmean - L_tilde[, 2] * 1.1*12,
                                    DBPmean + L_tilde[, 3] * 0.9*12, DBPmean - L_tilde[, 3] * 0.9*12),
                          component = c(rep("1st", 24 * 2), rep("2nd", 24 * 2), rep("3rd", 24 * 2)),
                          sign = rep(rep(c("positive", "negative"), each = 24), 3),
                          hour = rep(12:35, 6))
```

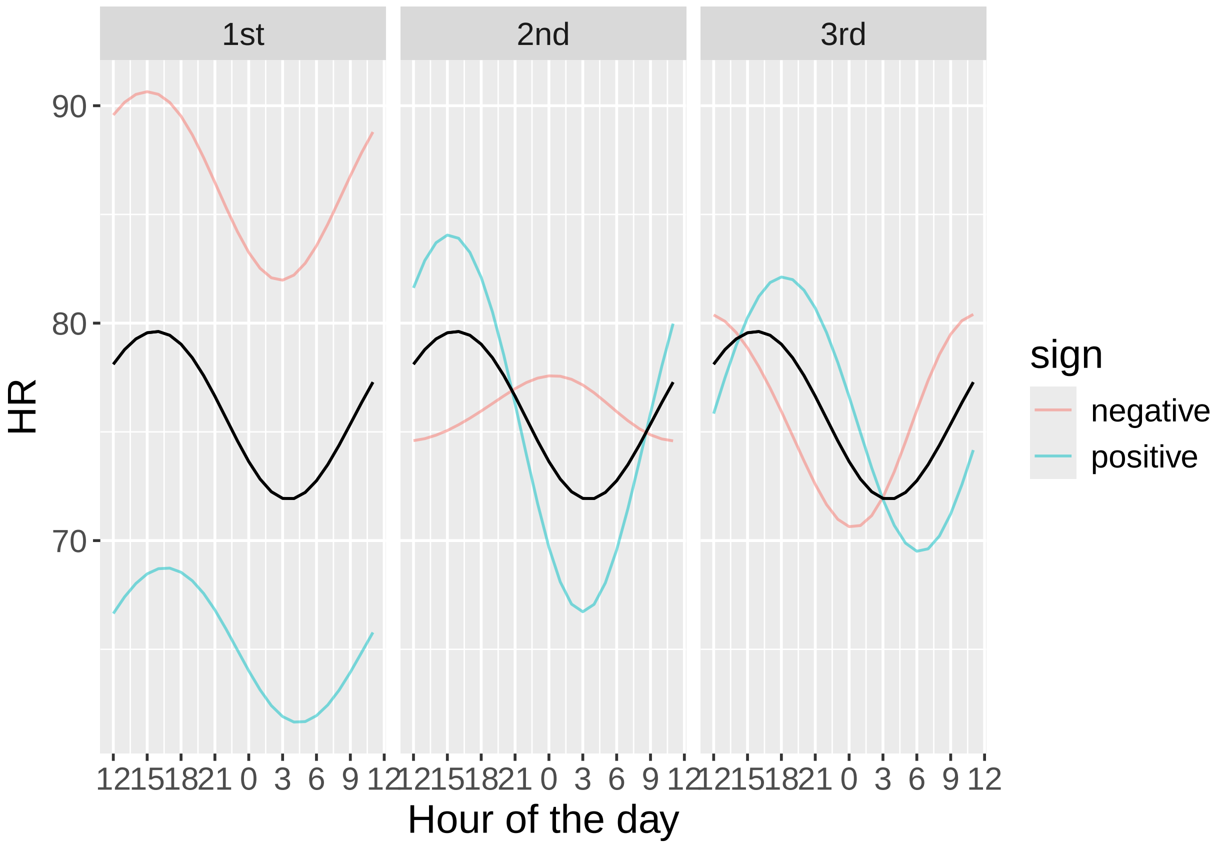
```
pdataL = dataLeffects %>%
  ggplot(aes(x = hour, y = value, group = sign, col = sign)) + geom_line(alpha = 0.5) +
  scale_x_continuous(breaks = c(12, 15, 18, 21, 24, 27, 30, 33, 36), labels = c(12, 15, 18, 21, 0, 3, 6
  xlab("Hour of the day") +
  facet_grid(~ component) +
  ylab("DBP") + geom_line(aes(x = hour, y = DBPmean), col = "black") +
  theme(text = element_text(size = 15))

pdataL
```




```
# Plot effect of time components on SBP
dataLeffects = data.frame(SBPmean = rep(SBPmean, 6),
                          value = c(SBPmean + L_tilde[, 1] * 3.1*18, SBPmean - L_tilde[, 1] * 3.1*18,
                                    SBPmean + L_tilde[, 2] * 1.2*18, SBPmean - L_tilde[, 2] * 1.2*18,
                                    SBPmean + L_tilde[, 3] * 1*18, SBPmean - L_tilde[, 3] * 1*18),
                          component = c(rep("1st", 24 * 2), rep("2nd", 24 * 2), rep("3rd", 24 * 2)),
                          sign = rep(rep(c("positive", "negative"), each = 24), 3),
                          hour = rep(12:35, 6))

pdataL = dataLeffects %>%
  ggplot(aes(x = hour, y = value, group = sign, col = sign)) + geom_line(alpha = 0.5) +
  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("SBP") + geom_line(aes(x = hour, y = SBPmean
  theme(text = element_text(size = 15))

pdataL
```

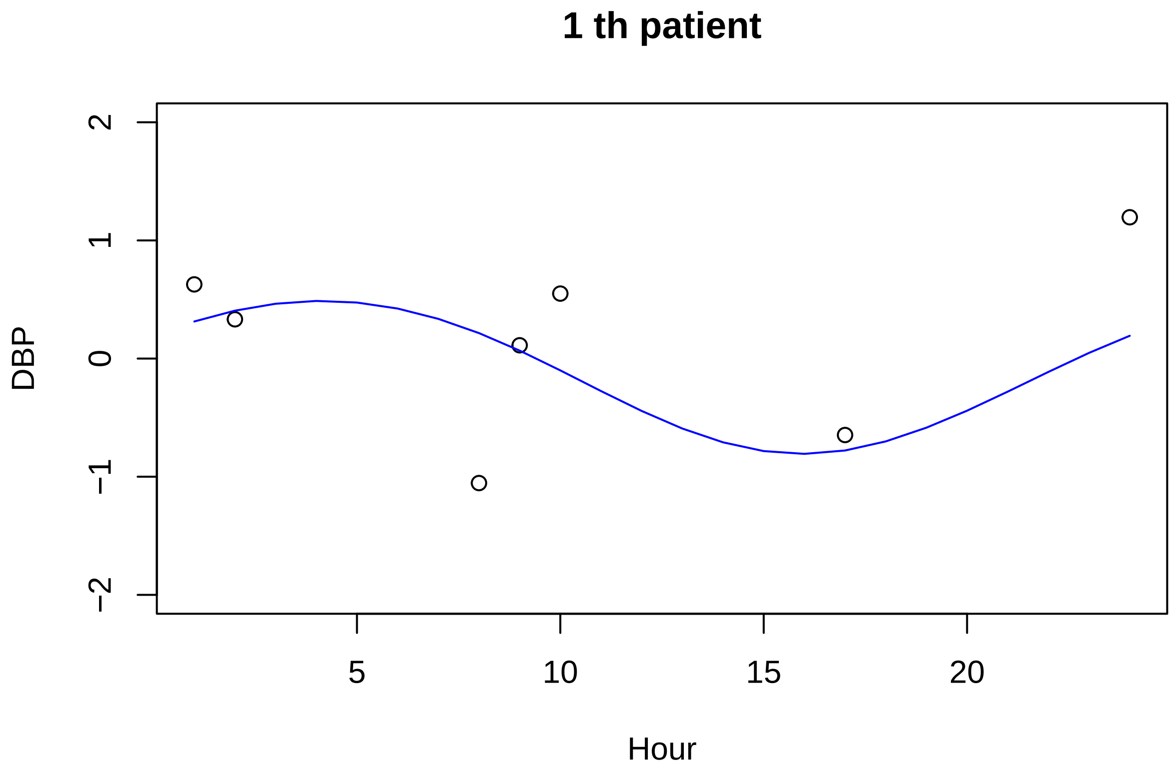



```r
# Plot effect of time components on HR
dataLeffects = data.frame(HRmean = rep(HRmean, 6),
                          value = c(HRmean + L_tilde[, 1] * 4*13, HRmean - L_tilde[, 1] *4*13,
                                    HRmean + L_tilde[, 2] * 1.3*13, HRmean - L_tilde[, 2] * 1.3*13,
                                    HRmean + L_tilde[, 3] * 1.1*13, HRmean - L_tilde[, 3] * 1.1*13),
                          component = c(rep("1st", 24 * 2), rep("2nd", 24 * 2), rep("3rd", 24 * 2)),
                          sign = rep(rep(c("positive", "negative"), each = 24), 3),
                          hour = rep(12:35, 6))

pdataL = dataLeffects %>%
  ggplot(aes(x = hour, y = value, group = sign, col = sign)) + geom_line(alpha = 0.5) +
  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("HR") + geom_line(aes(x = hour, y = HRmean),
  theme(text = element_text(size = 15))

pdataL
```

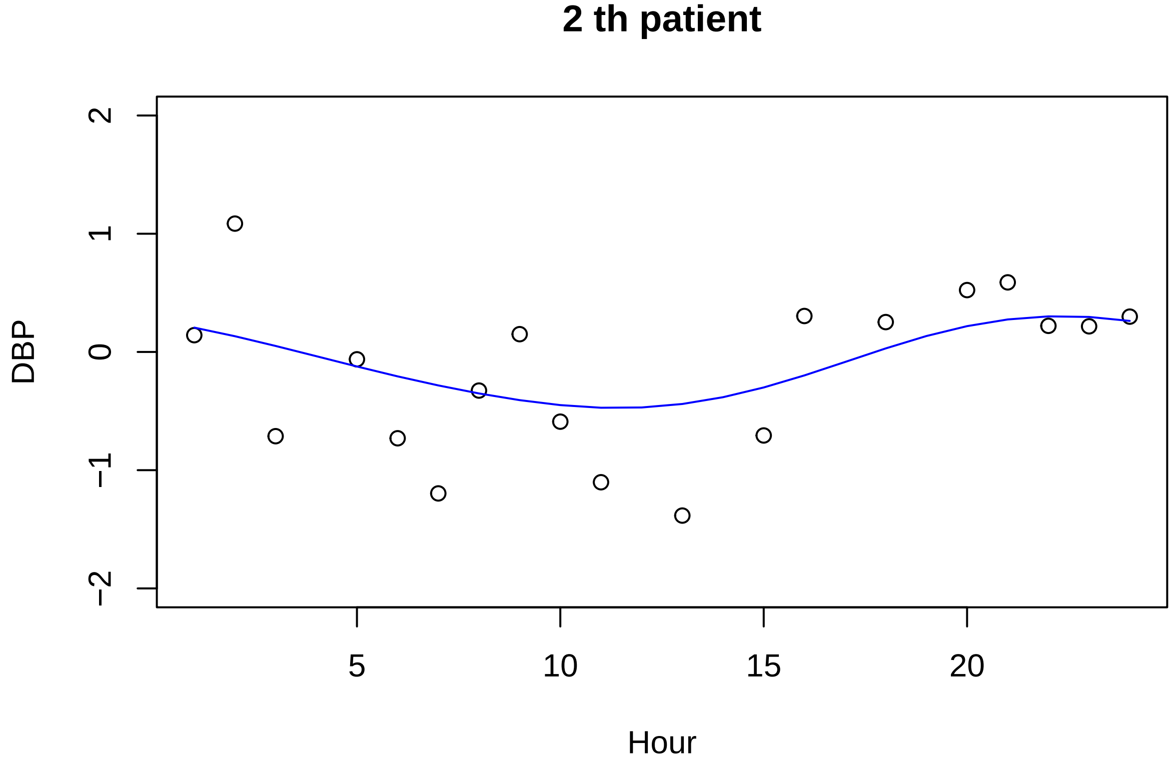


Plot estimated curves for some patients

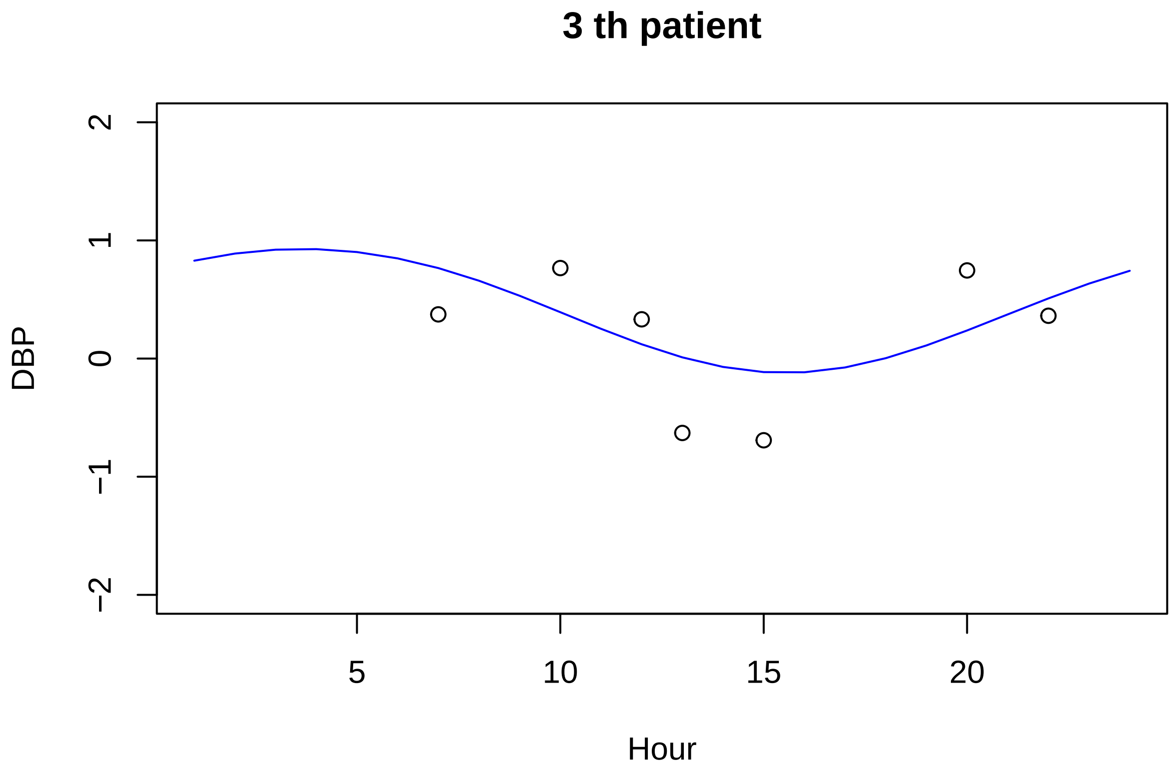
```
for (i in 1:5){
  plot(synthetic_dat@data[ , 1, i], main=paste(i,"th patient"), xlab="Hour", ylab="DBP", ylim=c(-2,2))
  lines(res$est[ , 1, i], col="blue") # estimated curve by our algorithm
}
```

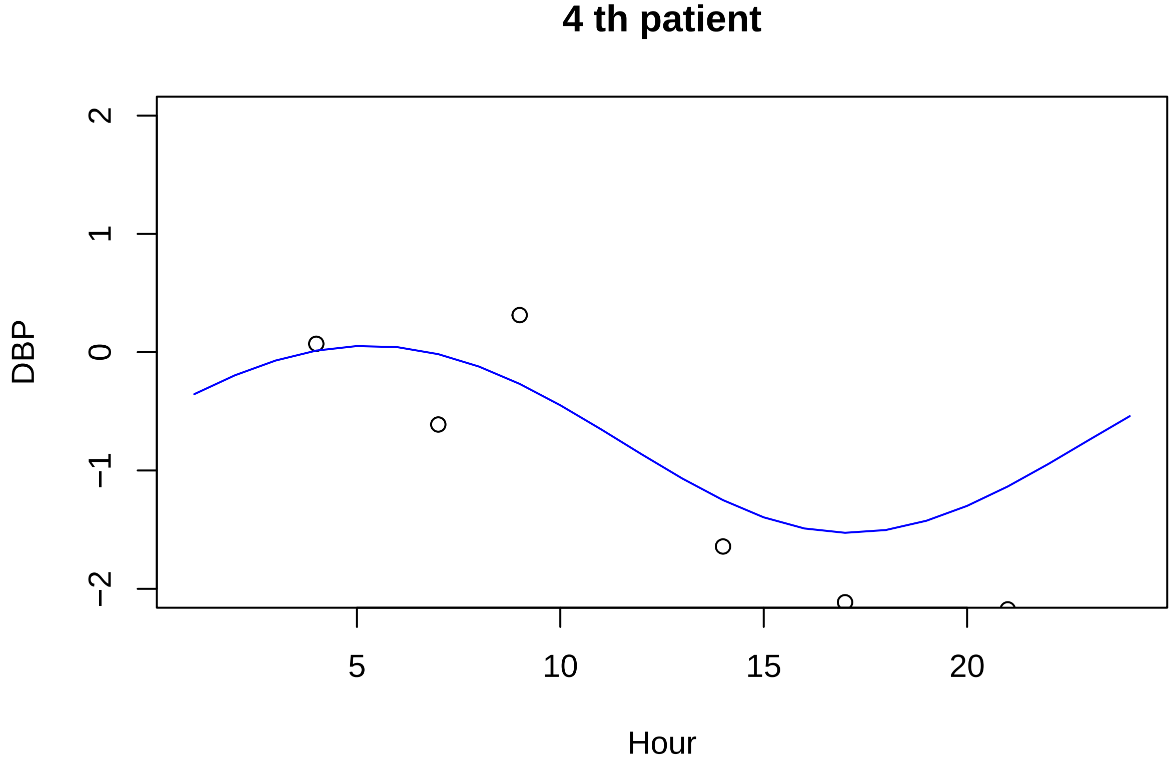


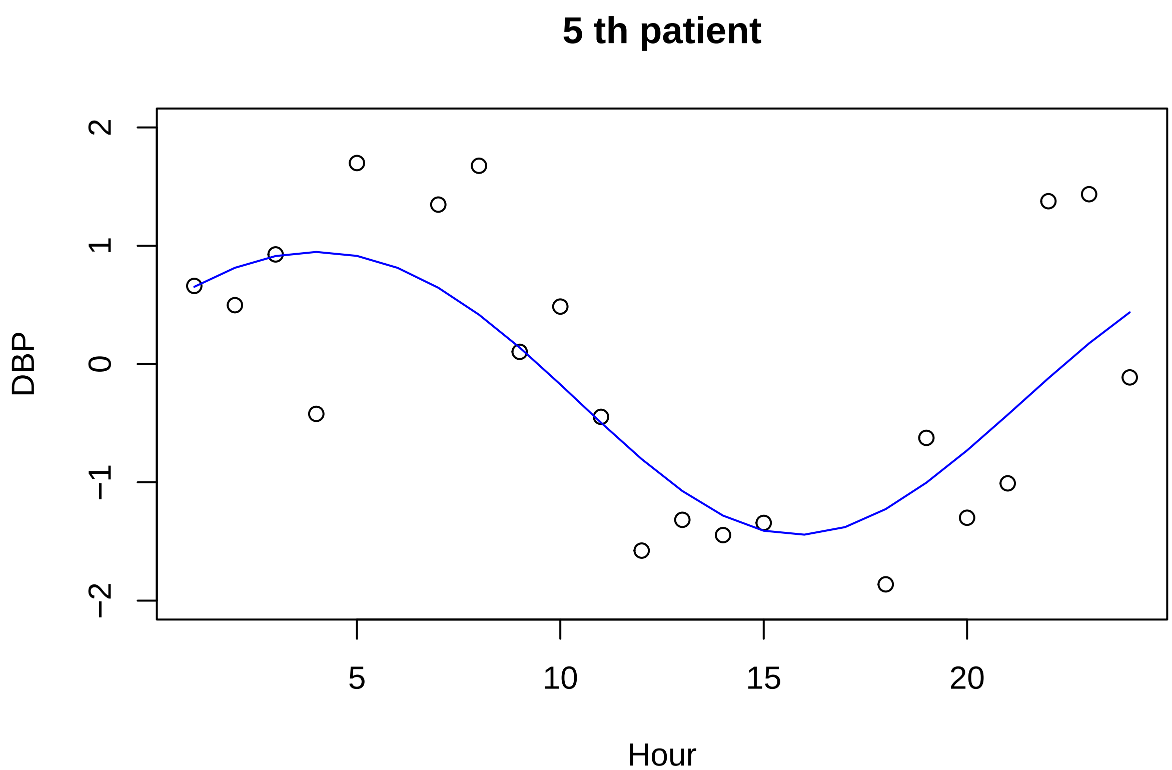
1 th patient
DBP
Hour



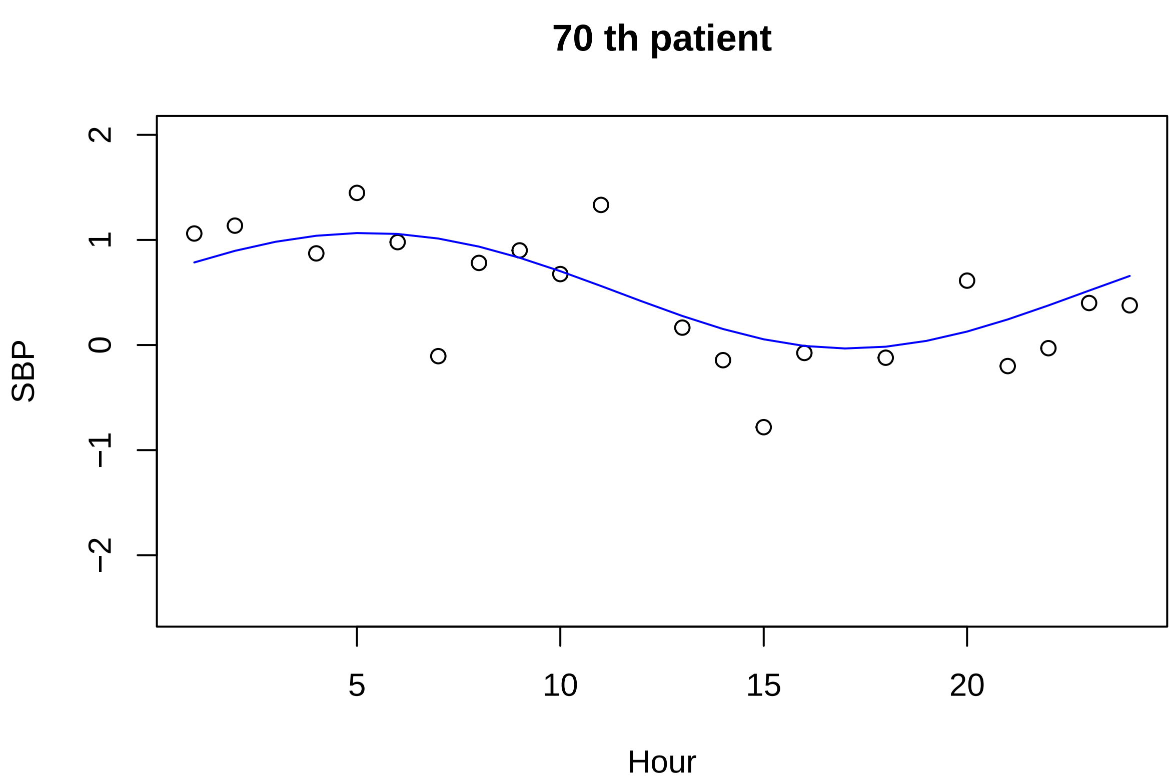
2 th patient
DBP
Hour

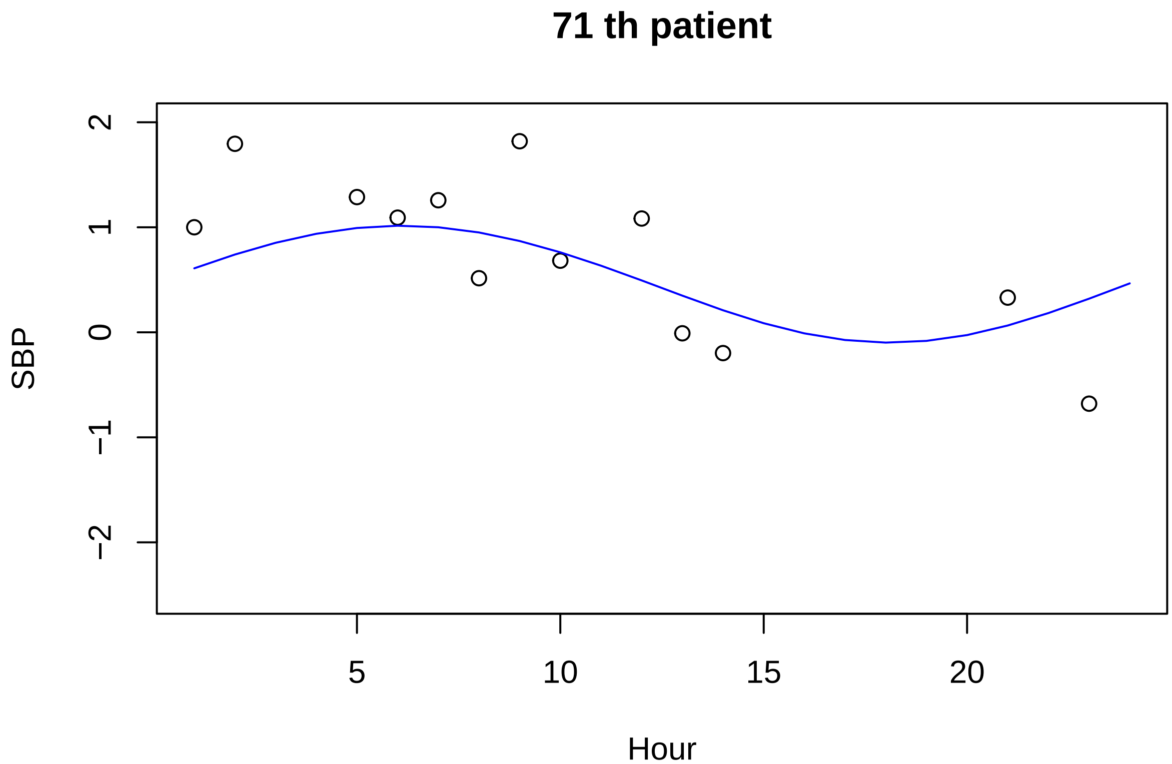

3 th patient
2
1
0
−1
−2
DBP
5
10
15
20
Hour

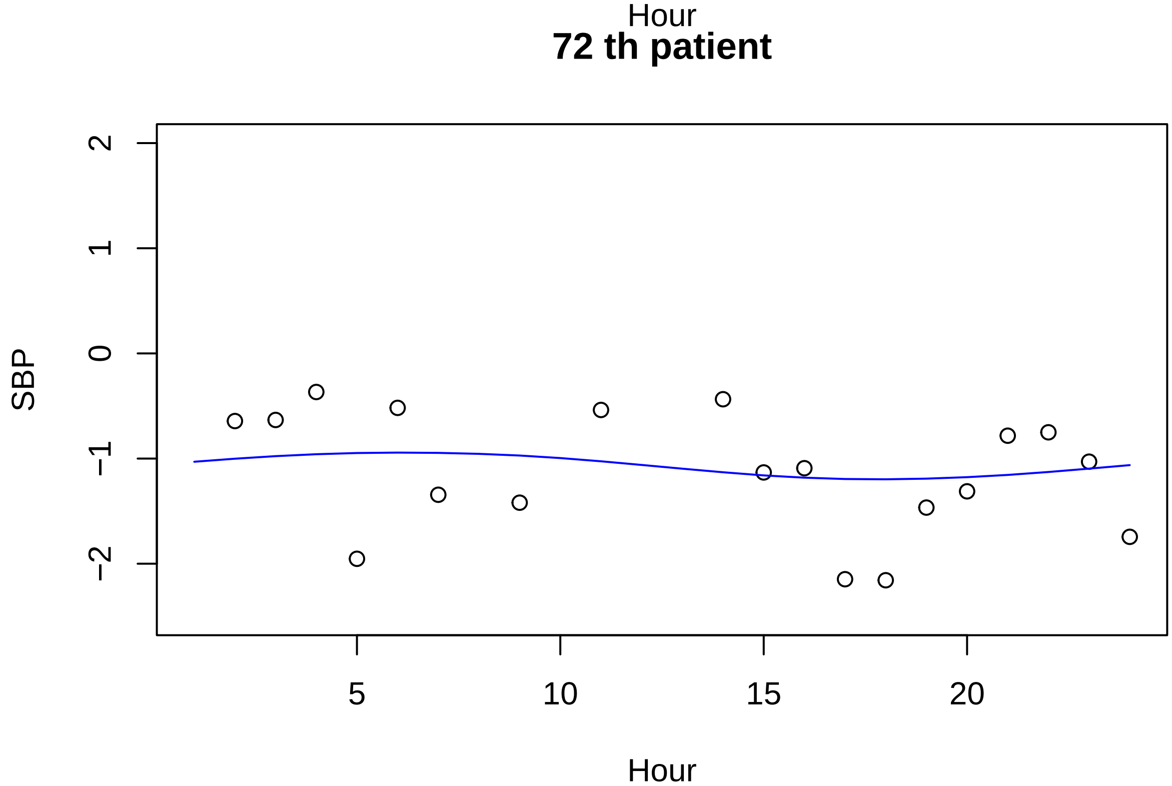

4 th patient
2
1
0
−1
−2
DBP
5
10
15
20
Hour

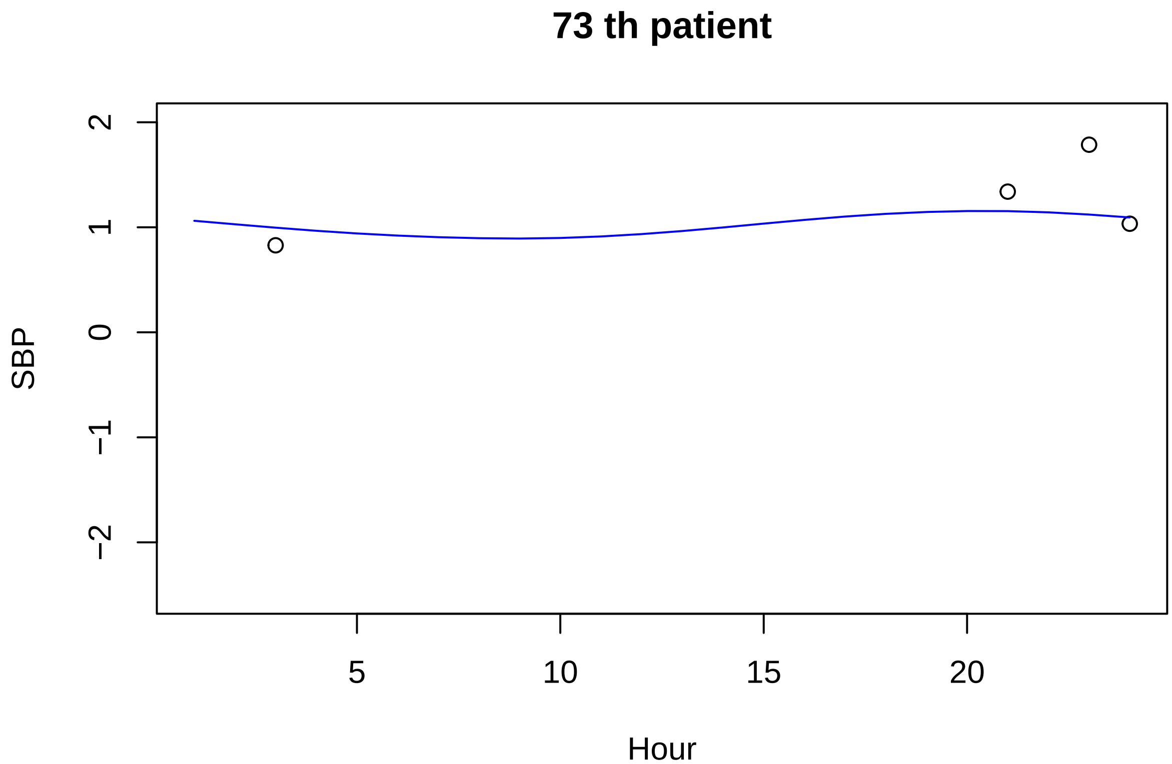


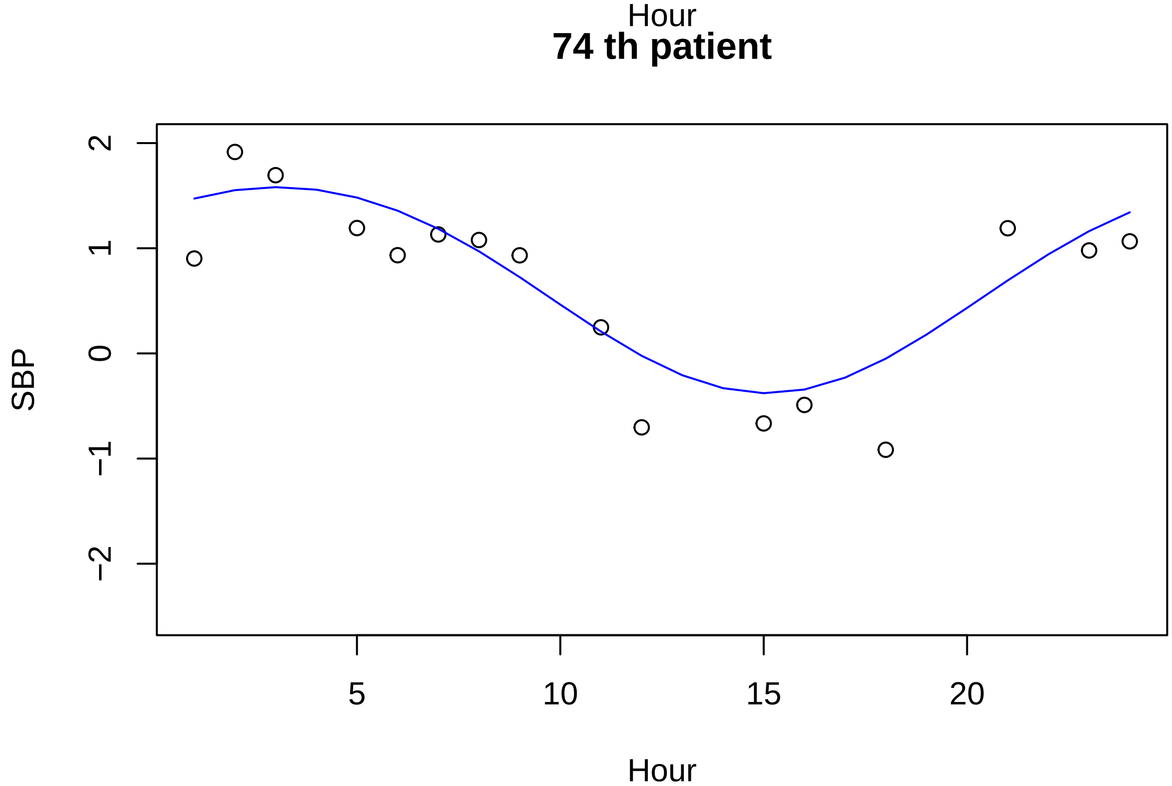

```
# Plot Estimated Curves for SBP
for (i in 70:75){
  plot(synthetic_dat@data[ , 2, i], main=paste(i,"th patient"), xlab="Hour", ylab="SBP", ylim=c(-2.5,2)
  lines(res$est[ , 2, i], col="blue")
}
```

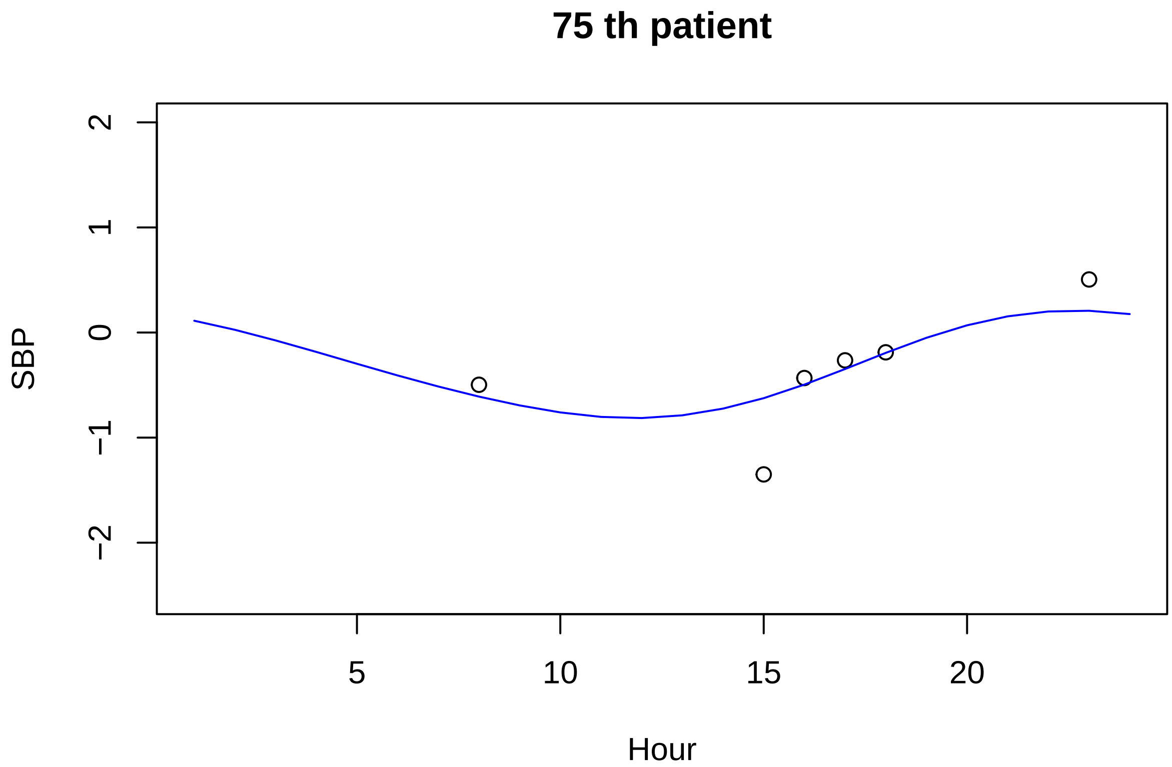



71 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

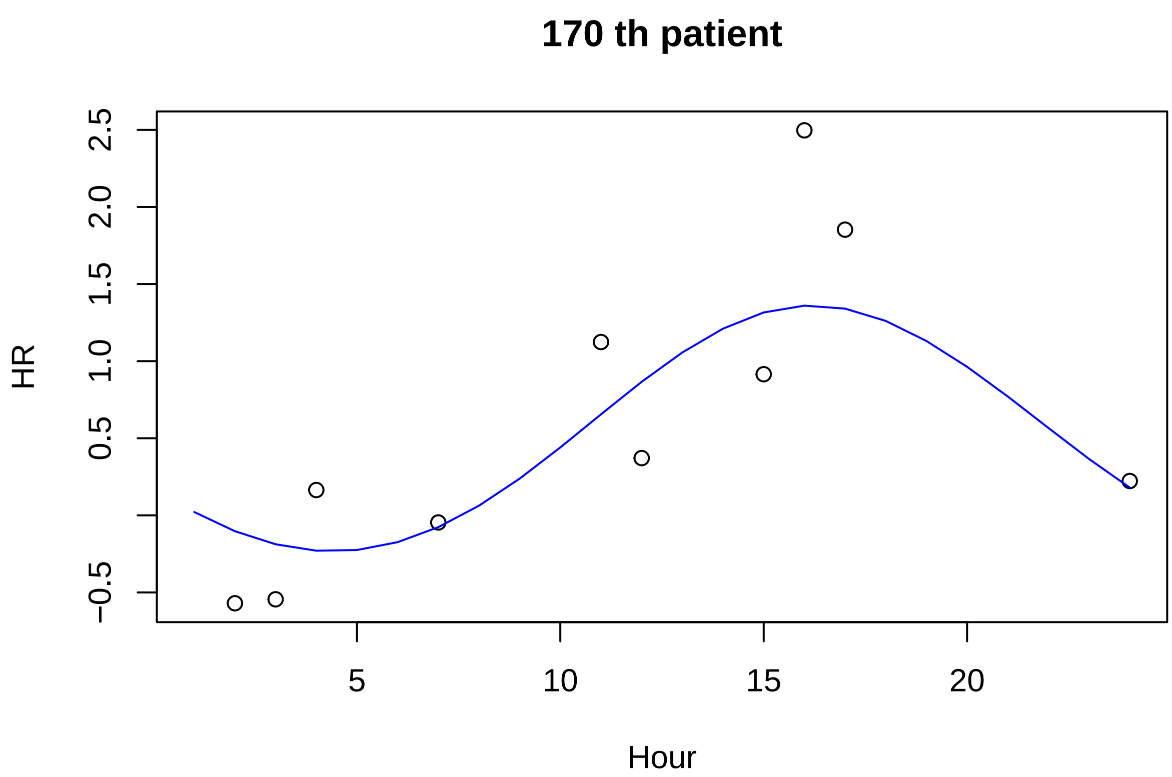

72 th patient
SBP
Hour
2
1
0
−1
−2
5
10
15
20

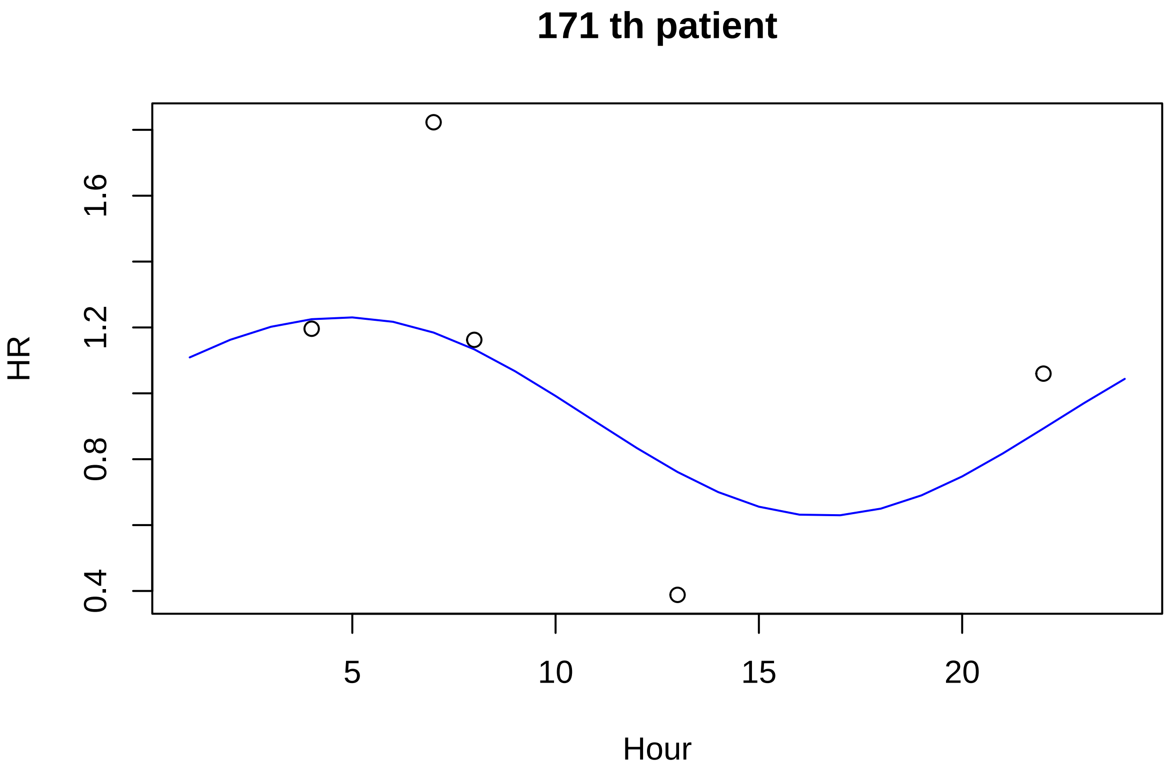

73 th patient
SBP
2
1
0
−1
−2
5
10
15
20
Hour


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

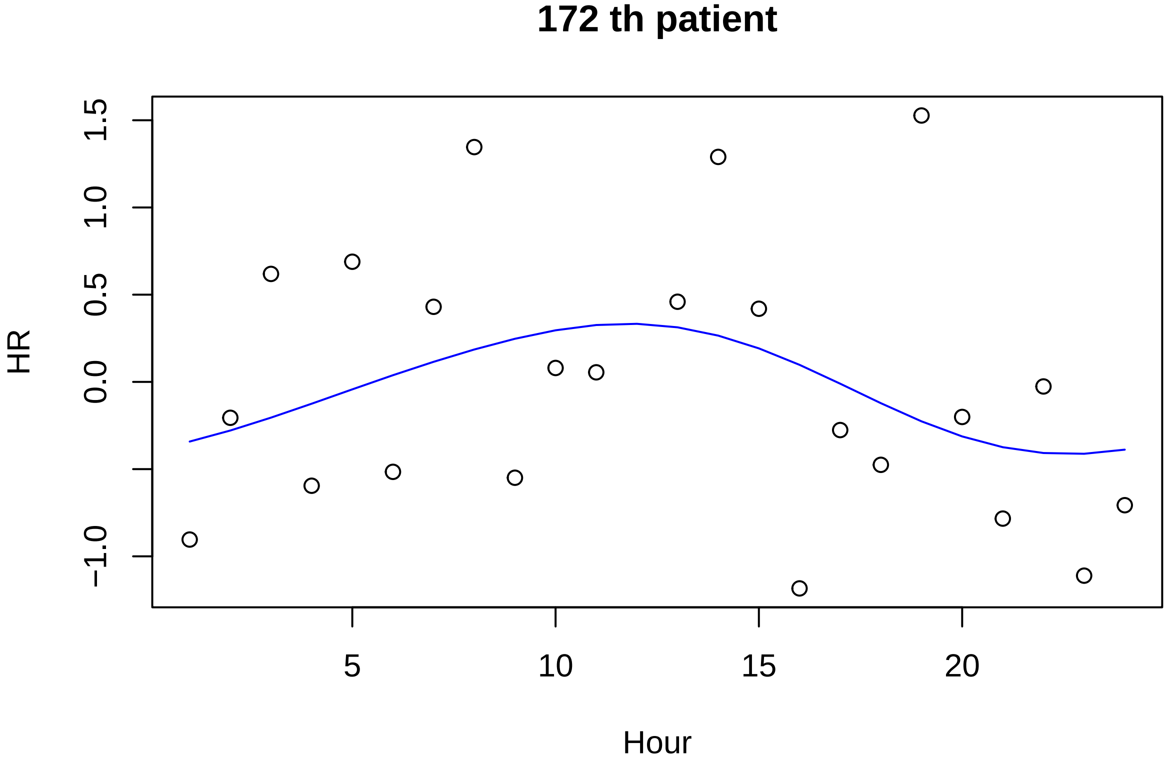



```
# Plot Estimated Curves for HR
for (i in 170:175){
  plot(synthetic_dat@data[ , 3, i], main=paste(i,"th patient"), xlab="Hour", ylab="HR")
  lines(res$est[ , 3, i], col="blue")
}
```

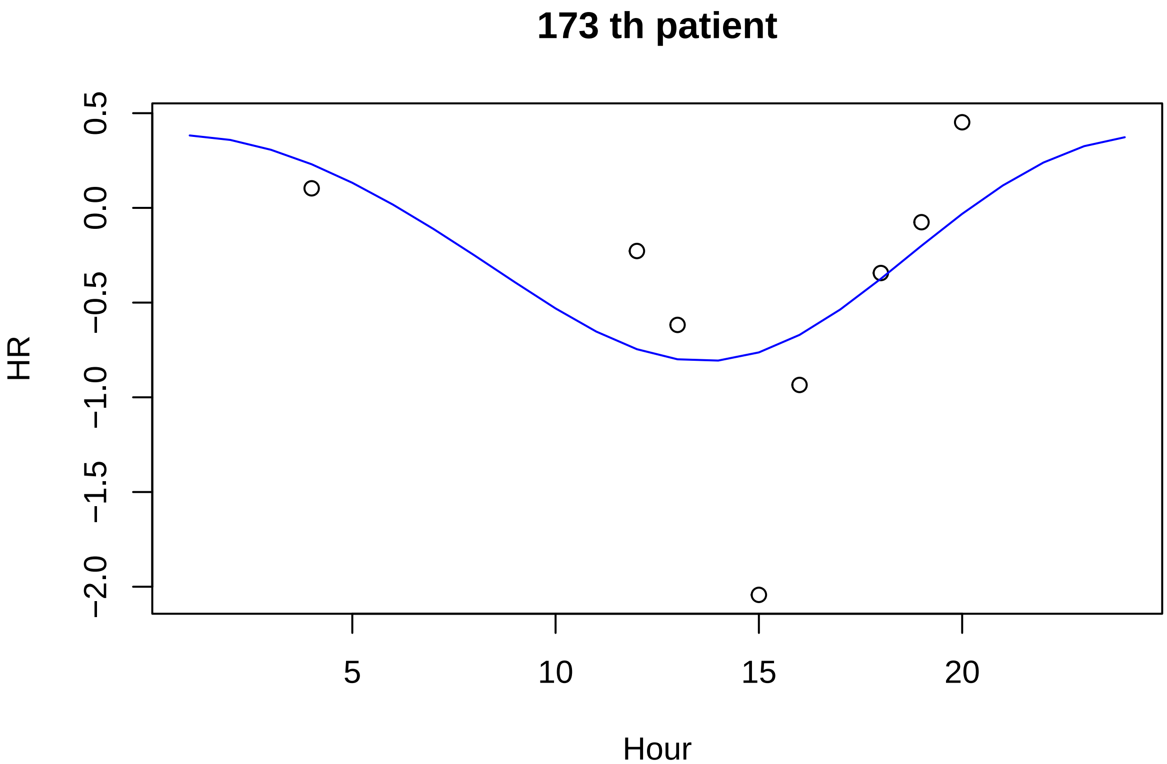


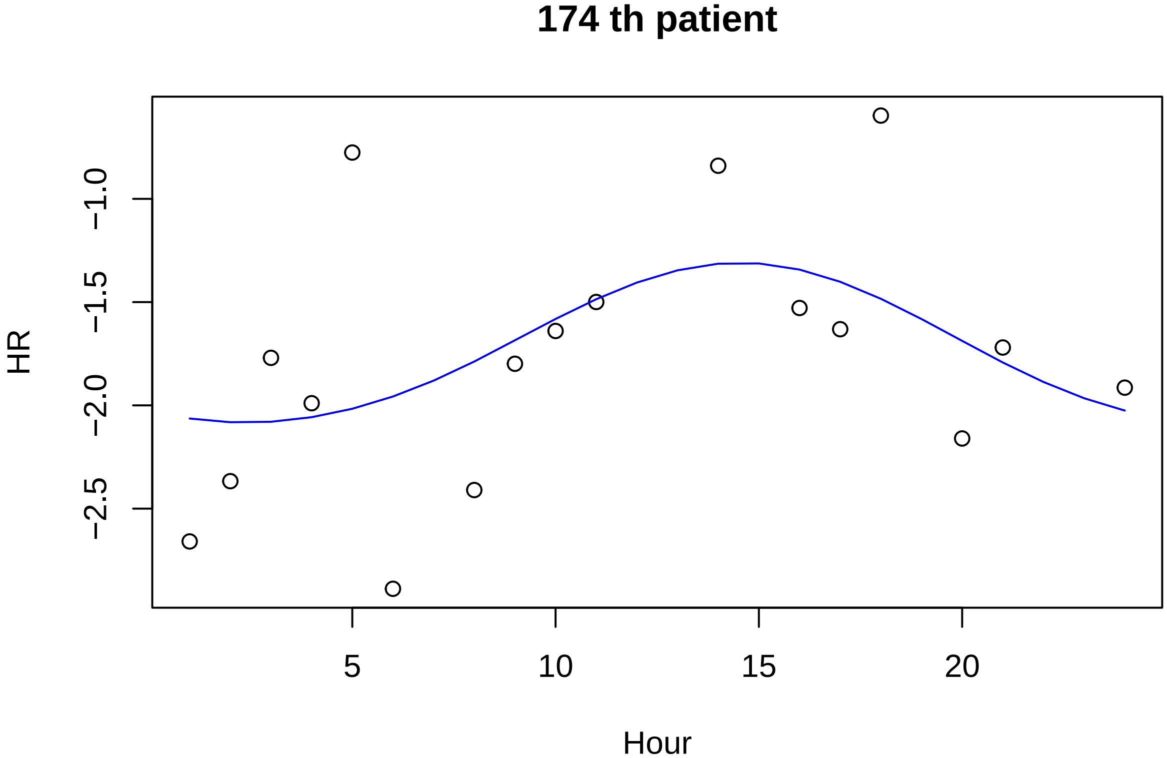


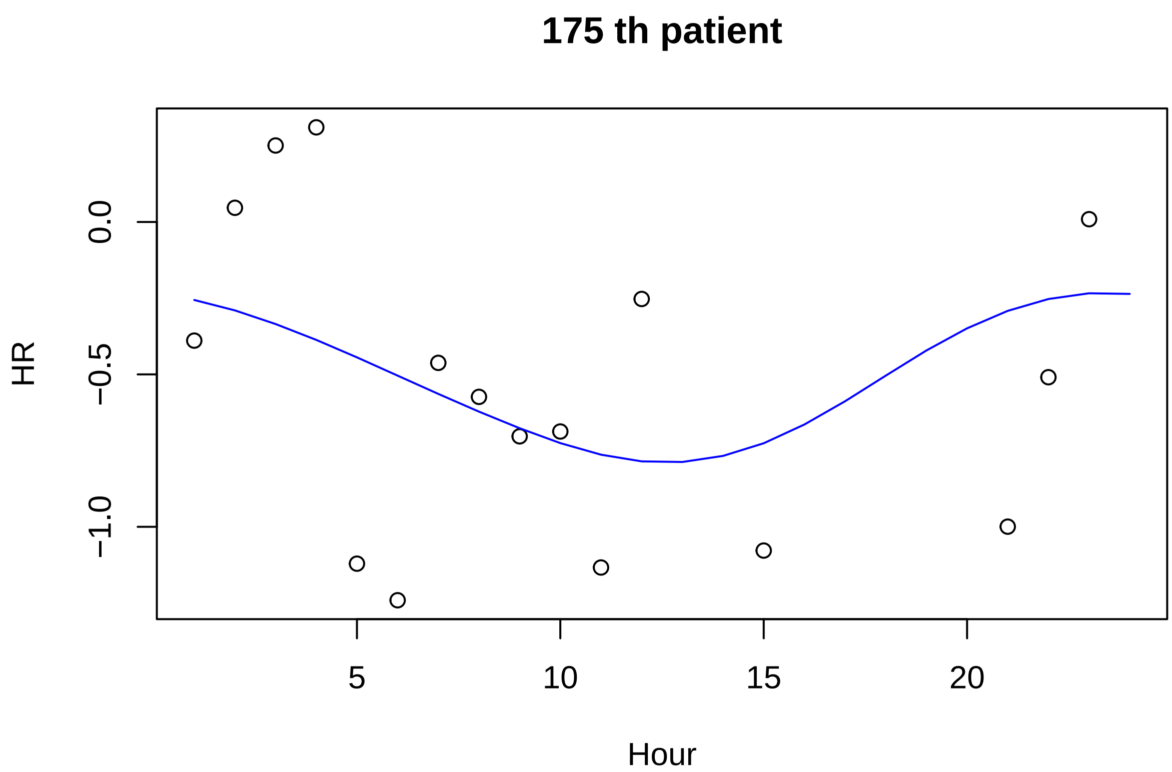
171 th patient
HR
0.4
0.8
1.2
1.6
5
10
15
20
Hour



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


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


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



Comparison with FPCA (Example)

```
library(refund)
h1 <- rTensor::unfold(synthetic_dat[,1,], row_idx = 2, col_idx = 1)@data
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(111)){
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)
}
```