

Testing tspred

Discrepancies with predict.ARIMA seem consistent with regularization using a Kalman filter

We will test regularization by shortening or tapering the pi sequence. This should work fine with AR series but not with an MA sequence that is close to non-invertible.

Variables:

- type of series:
 - stationary and invertible
 - nearly non-stationary 2nd order
 - nearly non-invertible 2nd order
 - both X
 - integrated or not X
 - seasonal
- models
 - 202 101 212 111 + same seasonal 12
- prediction
 - arima
 - tspred
 - tspred with taper

```
library(knitr)
opts_chunk$set(error=TRUE,fig.dim=c(8,7))
# hook_plot = knitr_hooks$get('plot')
# knitr_hooks$set(plot = function(x, options) paste('\n', hook_plot(x, options), sep = ''))
```

```
library(cv)
```

```
## Loading required package: doParallel
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
library(latticeExtra)
```

```
## Loading required package: lattice
```

```
source('tspred.R')
```

```
## Loading required package: stats4
```

```
##
```

```
## Attaching package: 'sarima'
```

```
## The following object is masked from 'package:stats':
```

```
##
```

```
##      spectrum
```

```
predplot <- function(N = 9999, ar = numeric(0), ma = numeric(0), int = 0, xcoef = 0,
                     half_taper = 50,
                     period = numeric(0), sar = numeric(0), sma = numeric(0),
                     sint = 0, sigma = 1,
                     main = '', seed = NA,
                     ylim = NULL,
                     sub = paste0(
                       "ar=", paste(ar, collapse = ' '),
```

```

      "ma=", paste(ma, collapse = ' '),
      "int=", int,
      "period=", sint,
      "sar=", paste(sar, collapse = ' '),
      "sma=", paste(sma, collapse = ' '),
      "sint=", sint)) {
{
  print(polyroot(c(1,-ar)))
  print(Mod(polyroot(c(1,-ar))))
  print(2*pi/Arg(polyroot(c(1,-ar))))
  if(!is.na(seed)) set.seed(seed)
  dd <- rts(N, ar = ar, ma = ma, int = int,
            period = period, sar = sar, sma = sma, sint = sint, xcoef= xcoef)
  # print(cyplot(y ~ time, dd, type = 'l'))
}
models <- within(
  list(),
  {
    `101` <- print(cv::Arima(y ~ x, data = dd, order = c(1,0,1)))
    `202` <- print(cv::Arima(y ~ x, data = dd, order = c(2,0,2)))
    `111` <- print(cv::Arima(y ~ x, data = dd, order = c(1,1,1)))
    `212` <- print(cv::Arima(y ~ x, data = dd, order = c(2,1,2)))
    `101/101(3)` <- print(cv::Arima(y ~ 1 +x, data = dd,
                                     order = c(1,0,1),
                                     seasonal = list(order = c(1,0,1), period = 3)))
    `202/202(3)` <- print(cv::Arima(y ~ x, data = dd,
                                     order = c(2,0,2),
                                     seasonal = list(order = c(2,0,2), period = 3)))
    `111/111(3)` <- print(cv::Arima(y ~ 1 +x, data = dd,
                                     order = c(1,1,1),
                                     seasonal = list(order = c(1,1,1), period = 3)))
    `212/212(3)` <- print(cv::Arima(y ~ x, data = dd,
                                     order = c(2,1,2),
                                     seasonal = list(order = c(2,1,2), period = 3)))

  }
) %>% rev

Show_pred(models, dd, last = 30, show = 50, main = main, sub = sub, ylim = ylim,
           half_taper = half_taper)
print(lapply(models, summary))
}

cols <- c('#88000088', '#00880088', '#00008888')

trellis.par.set('superpose.symbol',
  list(pch=c(0,2,6,1),
       col = cols ))
trellis.par.set('superpose.line', list(lty=1:4, col = cols ))

```

AR .8, MA .8

```
replicate(3, predplot(ar=.8, ma=.8, main = '101'))

## [1] 1.25+0i
## [1] 1.25
## [1] Inf

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Estimates:
##          ar1          ma1 (Intercept)          x
##    0.798888    0.798418    0.108817   -0.008465

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Estimates:
##          ar1          ar2          ma1          ma2 (Intercept)          x
##    0.315207    0.388010    1.279396    0.382399    0.110834   -0.008412

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Estimates:
##          ar1          ma1          x
##   -0.12997    0.82289  -0.00824

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##
## Estimates:
##          ar1          ar2          ma1          ma2          x
##    0.792748    0.007170  -0.198899  -0.801101  -0.008434

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##           seasonal = list(order = c(1, 0, 1), period = 3))
##
## Estimates:
##          ar1          ma1          sar1          sma1 (Intercept)          x
##    0.795680    0.797897    0.896862   -0.889463    0.109985   -0.008448

## Note: 'data' coerced to 'ts_data_frame'
```

```

## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##      0, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2
##  0.495759  0.237271  1.096970  0.239823  0.195491  0.622637
##      sma1      sma2 (Intercept)      x
##  -0.184134  -0.620733   0.104024  -0.008327
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##      seasonal = list(order = c(1, 1, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1      x
## -0.182942  0.861837 -0.123180 -0.999999 -0.007547
## Note: 'data' coerced to 'ts_data_frame'
##
## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
##      1, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2      sma1      sma2
##  0.792308  0.008302 -0.197871 -0.802127 -0.927232  0.002793 -0.073188 -0.926811
##      x
## -0.008994
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
##
## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1
## Note: 'data' coerced to 'ts_data_frame'
##
## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
##
## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1

```

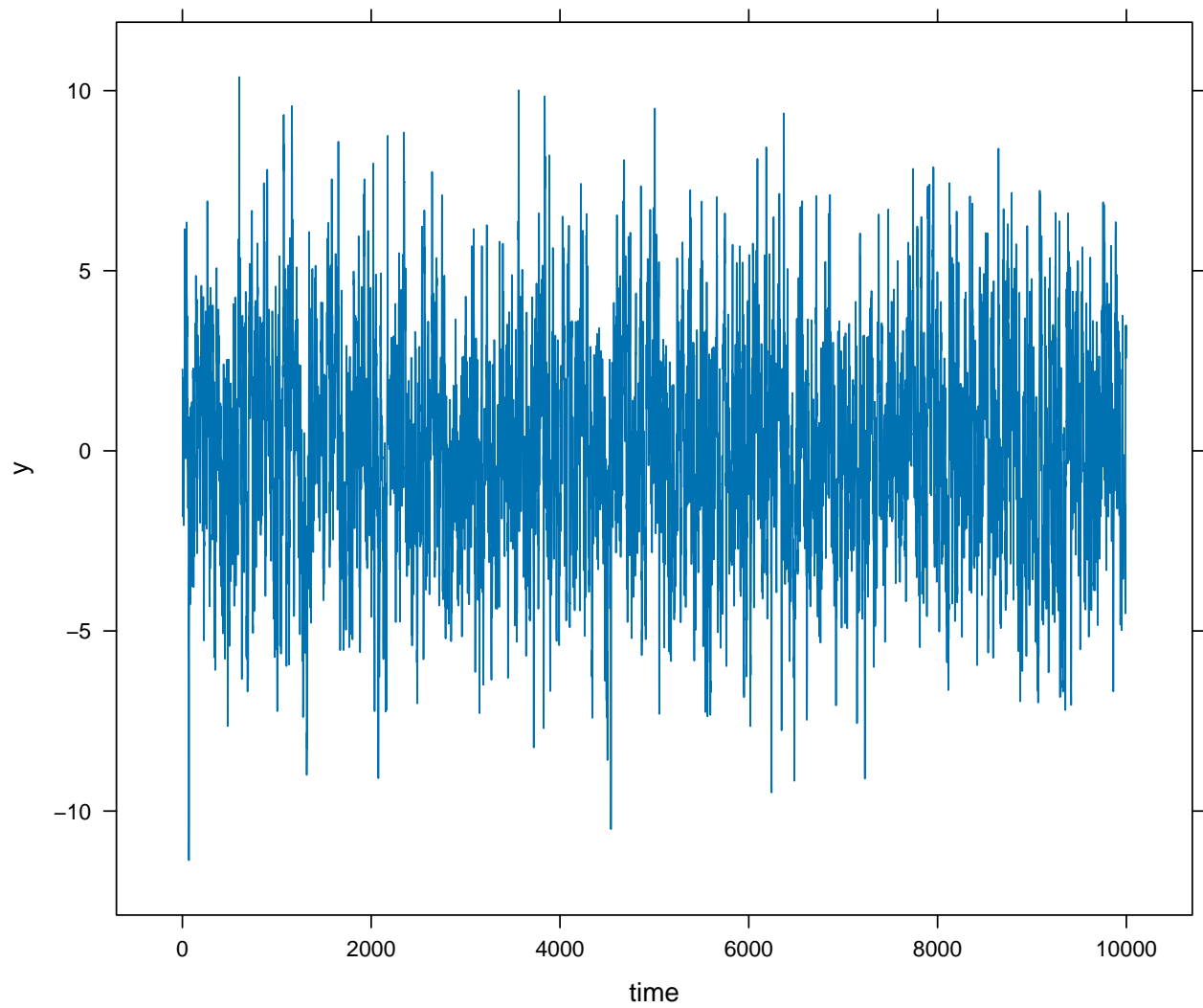
```

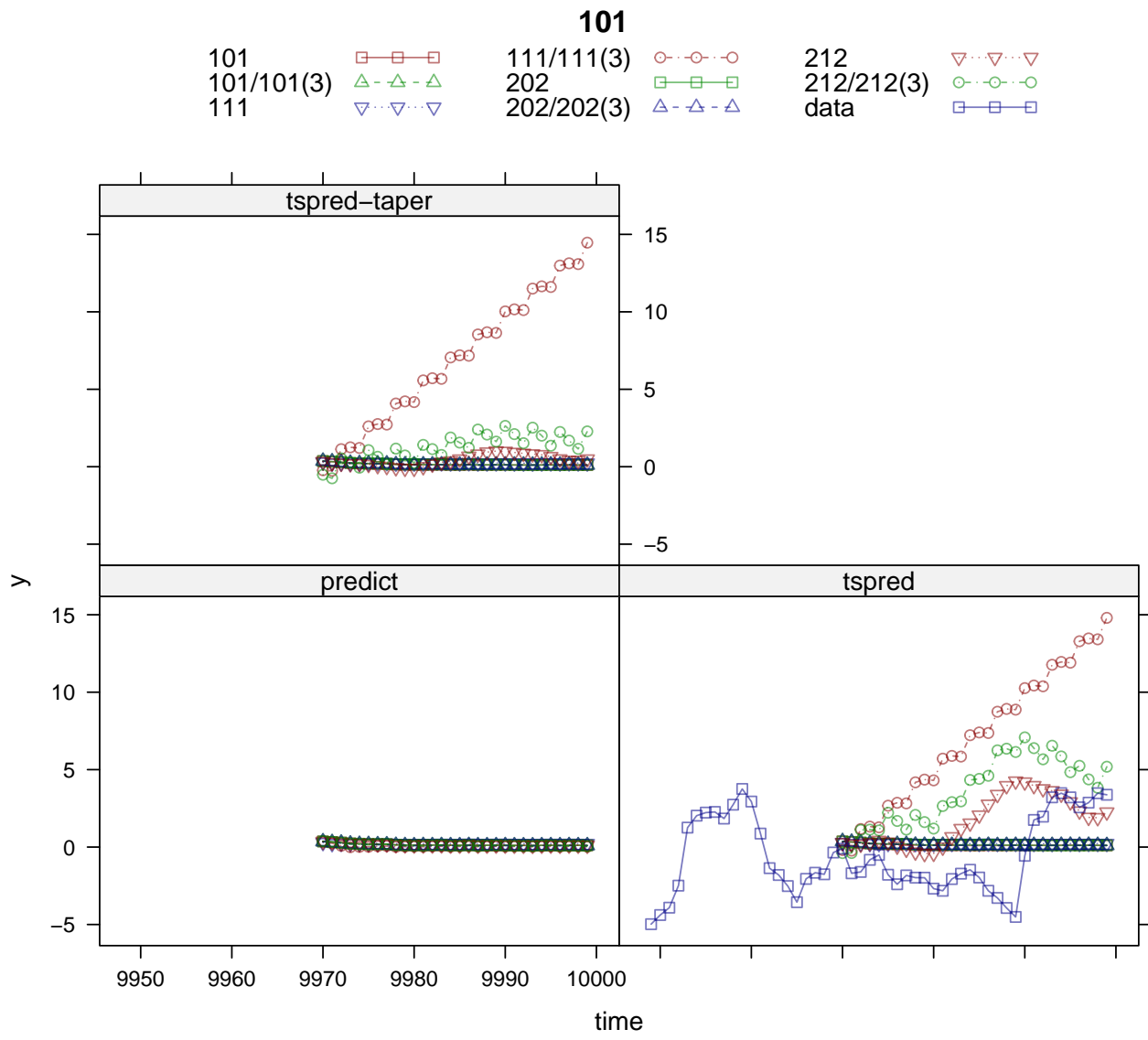
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained

##              model
## method      101 101/101(3) 111 111/111(3) 202 202/202(3) 212 212/212(3) data
##   predict      30      30 30      30 30      30 30      30 0
##   tspred      30      30 30      30 30      30 30      30 51
##   tspred-taper 30      30 30      30 30      30 30      30 0
##   Total      90      90 90      90 90      90 90      90 51

##              model
## method      Total
##   predict      240
##   tspred      291
##   tspred-taper 240
##   Total      771

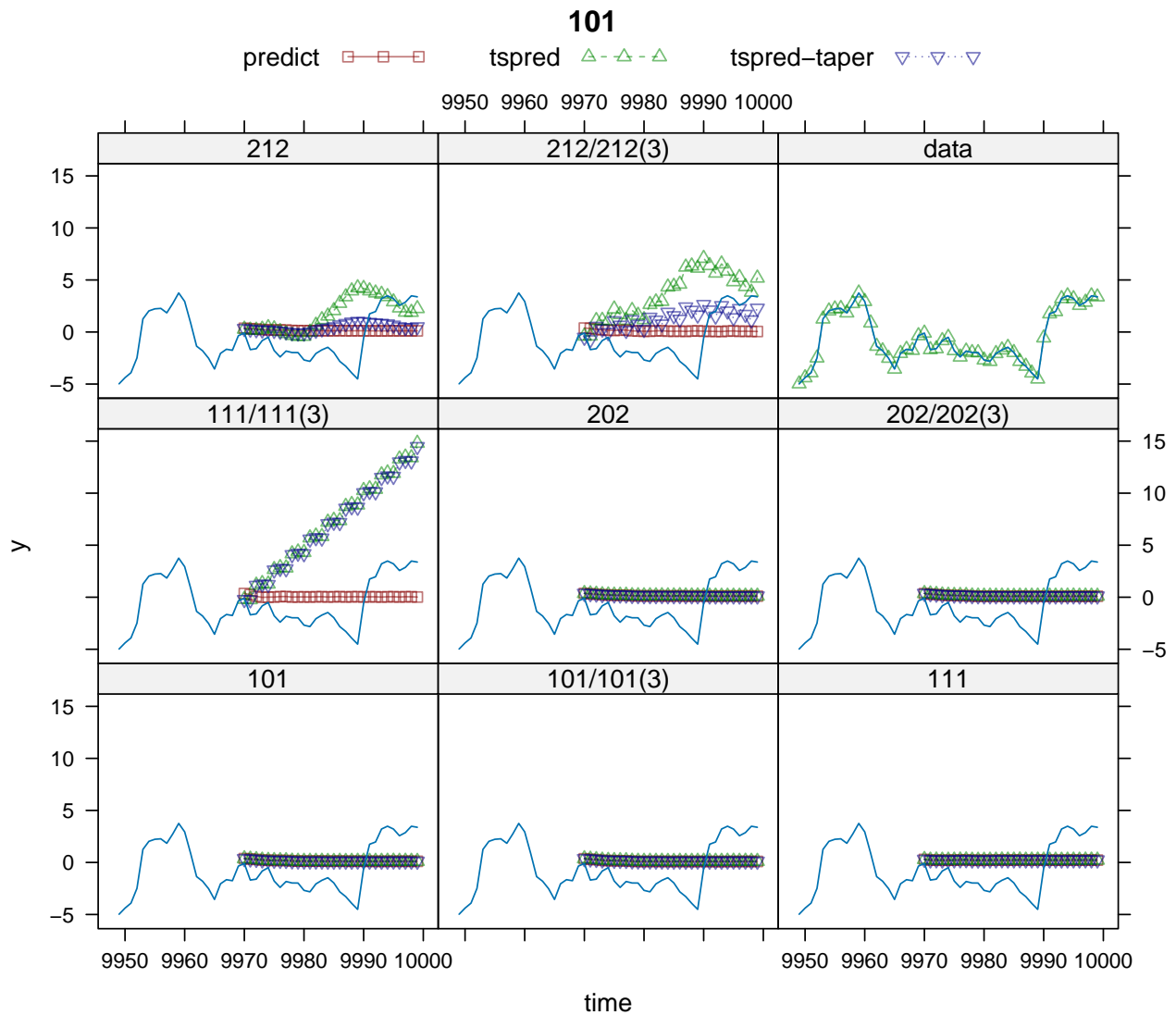
```





ar=0.8 ma=0.8 int=0 period=0sar=sma=sint=0

[1] " "



ar=0.8 ma=0.8 int=0 period=0sar=sma=sint=0

```
## [1] " "
## $`101`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -4.20847 -0.66208 -0.00378  0.67525  3.68157
##
## Estimates:
##              Estimate Std. Error z value Pr(>|z|)
## ar1            0.798888   0.006165 129.583 <2e-16 ***
## ma1            0.798418   0.006253 127.685 <2e-16 ***
## (Intercept)    0.108817   0.089635   1.214  0.2247
## x             -0.008465   0.003677  -2.302  0.0213 *
## ---
```



```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.003
## Log-likelihood = -14220
## AIC = 28440
##
## $`202`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -4.209652 -0.662498 -0.005235  0.674174  3.677240
##
## Warning in sqrt(diag(x$vcov)): NaNs produced
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1      0.315207      NaN      NaN      NaN
## ar2      0.388010      NaN      NaN      NaN
## ma1      1.279396      NaN      NaN      NaN
## ma2      0.382399      NaN      NaN      NaN
## (Intercept) 0.110834  0.089899  1.233  0.2176
## x        -0.008412  0.003665 -2.295  0.0217 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.003
## Log-likelihood = -14220
## AIC = 28450
##
## $`111`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -4.765535 -0.705038 -0.005174  0.704497  4.069185
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.129966  0.012549 -10.357 <2e-16 ***
## ma1  0.822889  0.006989 117.744 <2e-16 ***
## x   -0.008240  0.003686  -2.235  0.0254 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.051
## Log-likelihood = -14680
## AIC = 29380
##
## $`212`
##

```

```

## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -4.21682 -0.66842 -0.01012  0.66284  3.66860
##
## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1  0.792748   0.012683   62.504 <2e-16 ***
## ar2  0.007170   0.012573    0.570  0.5685
## ma1 -0.198899   0.007785  -25.549 <2e-16 ***
## ma2 -0.801101   0.007779 -102.981 <2e-16 ***
## x   -0.008434   0.003677   -2.294  0.0218 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.003
## Log-likelihood = -14220
## AIC = 28450
##
## $`101/101(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##      seasonal = list(order = c(1, 0, 1), period = 3))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -4.214820 -0.661177 -0.006431  0.674837  3.693301
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1          0.795680   0.006580 120.916 <2e-16 ***
## ma1          0.797897   0.006274 127.184 <2e-16 ***
## sar1         0.896862   0.081748  10.971 <2e-16 ***
## sma1        -0.889463   0.084648 -10.508 <2e-16 ***
## (Intercept)  0.109985   0.094501   1.164  0.2445
## x          -0.008448   0.003677   -2.298  0.0216 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.003
## Log-likelihood = -14220
## AIC = 28450
##
## $`202/202(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##      0, 2), period = 3))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max

```

```
## -4.206725 -0.660097 -0.006251 0.674470 3.692030
## Warning in sqrt(diag(x$vcov)): NaNs produced
## Estimates:
##           Estimate Std. Error z value Pr(>|z|
## ar1         0.495759      NaN      NaN      NaN
## ar2         0.237271      NaN      NaN      NaN
## ma1         1.096970      NaN      NaN      NaN
## ma2         0.239823      NaN      NaN      NaN
## sar1        0.195491    0.301118    0.649 0.51620
## sar2        0.622637    0.231552    2.689 0.00717 **
## sma1       -0.184134    0.297835   -0.618 0.53642
## sma2       -0.620733    0.225506   -2.753 0.00591 **
## (Intercept) 0.104024    0.094107    1.105 0.26900
## x          -0.008327    0.003677   -2.264 0.02354 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.003
## Log-likelihood = -14220
## AIC = 28450
##
## $`111/111(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##           seasonal = list(order = c(1, 1, 1), period = 3))
##
## Residuals:
##           Min           1st Q       Median        3rd Q          Max
## -4.753483 -0.699131 -0.002692  0.698499  4.045386
##
## Estimates:
##           Estimate Std. Error  z value Pr(>|z|
## ar1   -0.1829418   0.0124997  -14.636 <2e-16 ***
## ma1    0.8618366   0.0065247   132.089 <2e-16 ***
## sar1  -0.1231798   0.0114298   -10.777 <2e-16 ***
## sma1  -0.9999991   0.0007074 -1413.527 <2e-16 ***
## x     -0.0075475   0.0037254    -2.026  0.0428 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.045
## Log-likelihood = -14630
## AIC = 29280
##
## $`212/212(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
##           1, 2), period = 3))
##
## Residuals:
##           Min           1st Q       Median        3rd Q          Max
```

```
## -4.2218942 -0.6504118 0.0007795 0.6805615 3.6886474
## Warning in sqrt(diag(x$vcov)): NaNs produced
## Estimates:
##      Estimate Std. Error z value Pr(>|z|
## ar1  0.792308  0.014806  53.512 <2e-16 ***
## ar2  0.008302  0.017340   0.479 0.6321
## ma1 -0.197871  0.010935 -18.095 <2e-16 ***
## ma2 -0.802127  0.010927 -73.409 <2e-16 ***
## sar1 -0.927232      NaN      NaN      NaN
## sar2  0.002793  0.014788   0.189 0.8502
## sma1 -0.073188      NaN      NaN      NaN
## sma2 -0.926811      NaN      NaN      NaN
## x    -0.008994  0.003678  -2.445 0.0145 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.003
## Log-likelihood = -14230
## AIC = 28480
##
## [1] 1.25+0i
## [1] 1.25
## [1] Inf
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Estimates:
##      ar1      ma1 (Intercept)      x
## 0.7950005 0.8032510 -0.0288438 -0.0003846
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Estimates:
##      ar1      ar2      ma1      ma2 (Intercept)      x
## 1.0892634 -0.2360224 0.5125463 -0.2315641 -0.0278620 -0.0003879
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Estimates:
##      ar1      ma1      x
## -0.1170223 0.8217551 -0.0004183
## Note: 'data' coerced to 'ts_data_frame'
##
```

```

## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##
## Estimates:
##      ar1      ar2      ma1      ma2      x
## 0.803063 -0.009123 -0.200146 -0.799276 -0.001016

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##      seasonal = list(order = c(1, 0, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1 (Intercept)      x
## 0.7942497 0.8028071 0.3061078 -0.3029806 -0.0313832 -0.0003899

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##      0, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2
## 0.7317338 0.0478261 0.8708532 0.0534843 -1.2620832 -0.9110800
##      sma1      sma2 (Intercept)      x
## 1.2616153 0.9229541 -0.0314550 -0.0007194

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##      seasonal = list(order = c(1, 1, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1      x
## -0.1691510 0.8581549 -0.1220885 -0.9992917 -0.0006584

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
##      1, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2      sma1
## 0.8027633 -0.0141509 -0.2008186 -0.7991801 -0.2958756 0.0186377 -0.6944613
##      sma2      x
## -0.3046397 -0.0003847

## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

```

```

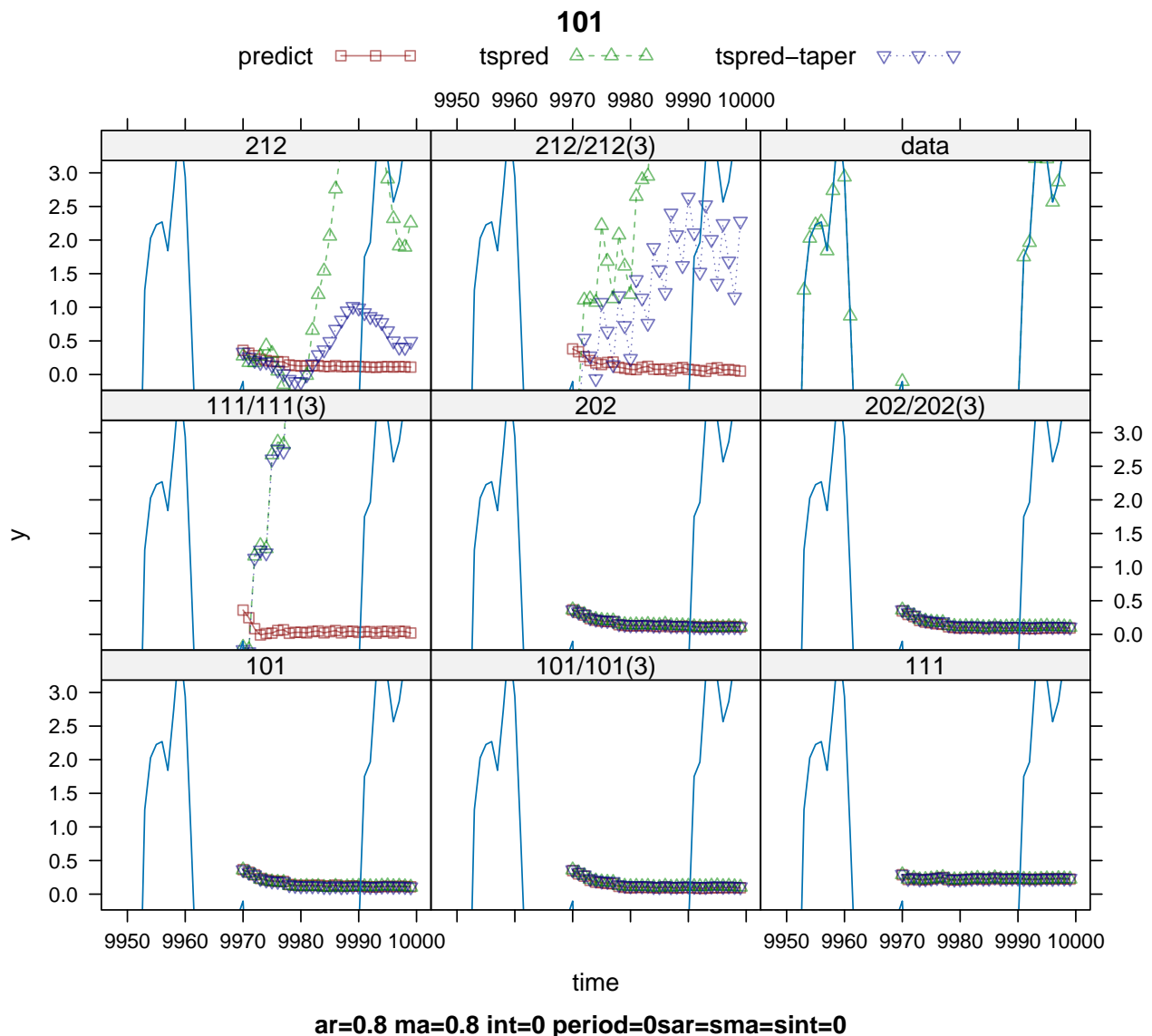
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1

## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained

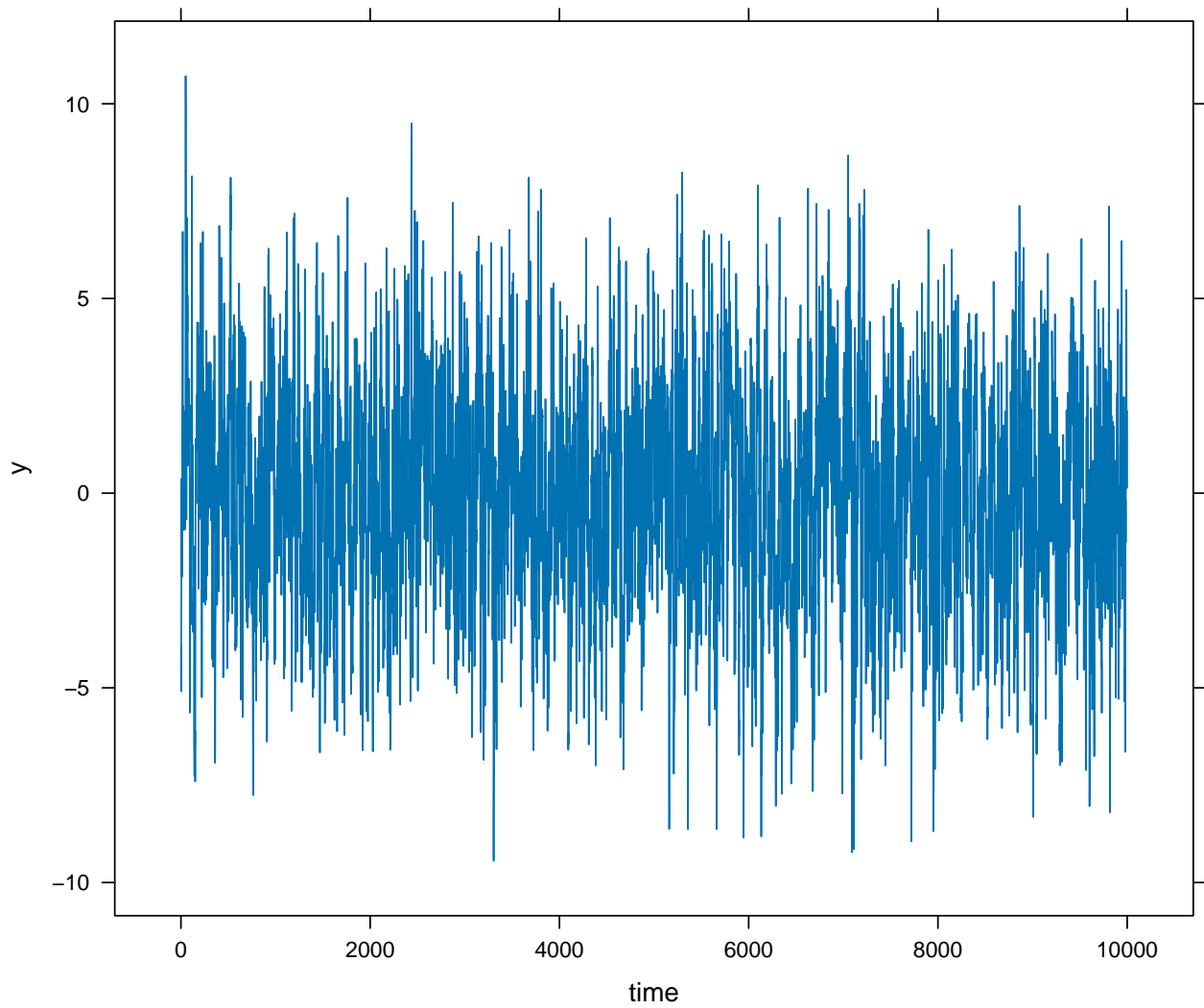
```



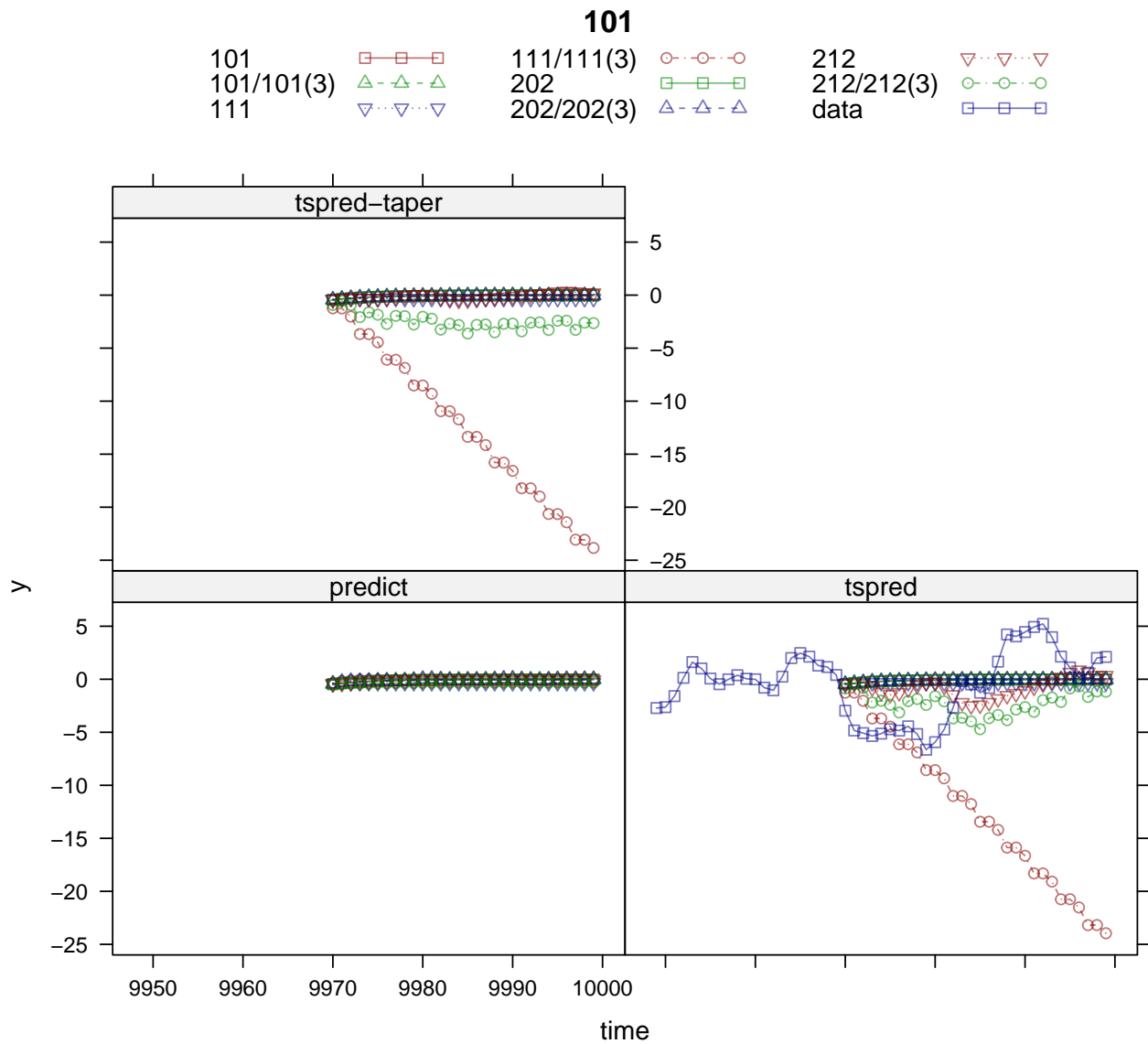
```

##          model
## method    101 101/101(3) 111 111/111(3) 202 202/202(3) 212 212/212(3) data
## predict      30      30 30      30 30      30 30      30 0
## tspred       30      30 30      30 30      30 30      30 51
## tspred-taper 30      30 30      30 30      30 30      30 0
## Total        90      90 90      90 90      90 90      90 51
##          model
## method      Total
## predict      240
## tspred       291
## tspred-taper 240
## Total        771

```

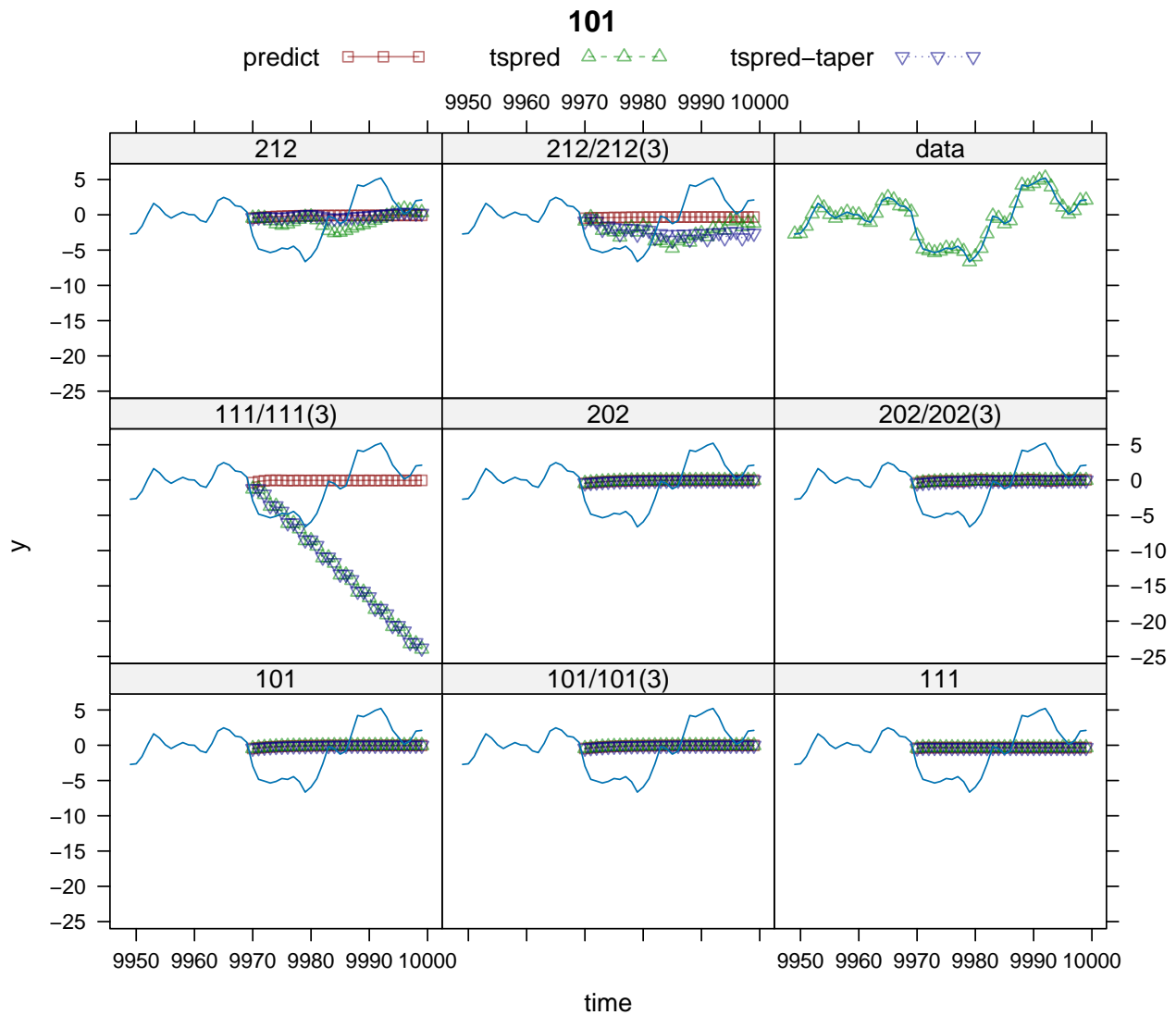


ar=0.8 ma=0.8 int=0 period=0sar=sma=sint=0



ar=0.8 ma=0.8 int=0 period=0sar=sma=sint=0

[1] " "



ar=0.8 ma=0.8 int=0 period=0sar=sma=sint=0

```
## [1] "    "
## $`101`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Residuals:
##      Min      1st Q   Median     3rd Q      Max
## -3.38504 -0.65673 -0.00515  0.65795  3.61808
##
## Estimates:
##              Estimate Std. Error z value Pr(>|z|)
## ar1           0.7950005   0.0062071 128.079  <2e-16 ***
## ma1           0.8032510   0.0059860 134.189  <2e-16 ***
## (Intercept) -0.0288438   0.0867233  -0.333   0.739
## x            -0.0003846   0.0034009  -0.113   0.910
## ---
```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.9863
## Log-likelihood = -14050
## AIC = 28110
##
## $`202`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -3.386132 -0.656823 -0.005758  0.656281  3.606968
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1      1.0892634   1.4739547   0.739   0.460
## ar2     -0.2360224   1.1692391  -0.202   0.840
## ma1      0.5125463   1.4791693   0.347   0.729
## ma2     -0.2315641   1.1909430  -0.194   0.846
## (Intercept) -0.0278620  0.0860532  -0.324   0.746
## x        -0.0003879  0.0034010  -0.114   0.909
##
## Residual standard deviation: 0.9863
## Log-likelihood = -14050
## AIC = 28120
##
## $`111`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -4.00771 -0.69098  0.00563  0.69317  3.67991
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.1170223  0.0123711  -9.459 <2e-16 ***
## ma1  0.8217551  0.0067567 121.620 <2e-16 ***
## x   -0.0004183  0.0034062  -0.123   0.902
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.036
## Log-likelihood = -14540
## AIC = 29080
##
## $`212`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##

```

```

## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -3.41775 -0.67625 -0.02063  0.64229  3.59286
##
## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1  0.803063    0.012472   64.390 <2e-16 ***
## ar2 -0.009123    0.012370   -0.737  0.461
## ma1 -0.200146    0.007561  -26.470 <2e-16 ***
## ma2 -0.799276    0.007551 -105.847 <2e-16 ***
## x   -0.001016    0.003405   -0.298  0.765
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.9864
## Log-likelihood = -14050
## AIC = 28120
##
## $`101/101(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##      seasonal = list(order = c(1, 0, 1), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -3.38358 -0.65579 -0.00521  0.65852  3.62627
##
## Warning in sqrt(diag(x$vcov)): NaNs produced
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1      0.7942497      NaN      NaN      NaN
## ma1      0.8028071  0.0058484 137.269 <2e-16 ***
## sar1      0.3061078      NaN      NaN      NaN
## sma1     -0.3029806      NaN      NaN      NaN
## (Intercept) -0.0313832  0.0867745  -0.362  0.718
## x         -0.0003899  0.0033979  -0.115  0.909
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.9863
## Log-likelihood = -14050
## AIC = 28120
##
## $`202/202(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##      0, 2), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -3.359087 -0.659464 -0.005393  0.652150  3.640722

```

```
## Warning in sqrt(diag(x$vcov)): NaNs produced

## Estimates:
##           Estimate Std. Error z value Pr(>|z|)
## ar1         0.7317338      NaN      NaN      NaN
## ar2         0.0478261      NaN      NaN      NaN
## ma1         0.8708532      NaN      NaN      NaN
## ma2         0.0534843      NaN      NaN      NaN
## sar1        -1.2620832  0.1049322 -12.028 < 2e-16 ***
## sar2        -0.9110800  0.1133639  -8.037 9.22e-16 ***
## sma1         1.2616153  0.1014170  12.440 < 2e-16 ***
## sma2         0.9229541  0.1038709   8.886 < 2e-16 ***
## (Intercept) -0.0314550  0.0863070  -0.364   0.716
## x           -0.0007194  0.0034024  -0.211   0.833
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.9855
## Log-likelihood = -14040
## AIC = 28110
##
## $`111/111(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##           seasonal = list(order = c(1, 1, 1), period = 3))
##
## Residuals:
##           Min           1st Q       Median           3rd Q           Max
## -3.90875 -0.69874  0.01367  0.68423  3.65713
##
## Estimates:
##           Estimate Std. Error z value Pr(>|z|)
## ar1    -0.1691510  0.0124981 -13.534 <2e-16 ***
## ma1     0.8581549  0.0064737 132.561 <2e-16 ***
## sar1   -0.1220885  0.0114664 -10.648 <2e-16 ***
## sma1   -0.9992917  0.0010424 -958.601 <2e-16 ***
## x      -0.0006584  0.0034504  -0.191   0.849
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.03
## Log-likelihood = -14490
## AIC = 28990
##
## $`212/212(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
##           1, 2), period = 3))
##
## Residuals:
##           Min           1st Q       Median           3rd Q           Max
## -3.411702 -0.653166  0.001104  0.667949  3.644818
```

```

##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1   0.8027633  0.0150242  53.432 <2e-16 ***
## ar2  -0.0141509  0.0177958  -0.795  0.427
## ma1  -0.2008186  0.0113153 -17.747 <2e-16 ***
## ma2  -0.7991801  0.0113094 -70.665 <2e-16 ***
## sar1 -0.2958756  0.4426659  -0.668  0.504
## sar2  0.0186377  0.0115701   1.611  0.107
## sma1 -0.6944613  0.4402221  -1.578  0.115
## sma2 -0.3046397  0.4380191  -0.695  0.487
## x    -0.0003847  0.0034046  -0.113  0.910
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.9863
## Log-likelihood = -14060
## AIC = 28150
##
## [1] 1.25+0i
## [1] 1.25
## [1] Inf

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Estimates:
##      ar1      ma1 (Intercept)      x
##  0.820141  0.799003  0.052344 -0.001518

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Estimates:
##      ar1      ar2      ma1      ma2 (Intercept)      x
##  0.134871  0.559578  1.489085  0.554095  0.053250 -0.001455

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Estimates:
##      ar1      ma1      x
## -0.108398  0.817691 -0.001479

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))

```

```

##
## Estimates:
##      ar1      ar2      ma1      ma2      x
## 0.821518 -0.001333 -0.201505 -0.798495 -0.001514
## Note: 'data' coerced to 'ts_data_frame'
## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##      seasonal = list(order = c(1, 0, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1 (Intercept)      x
## 0.818544 0.797471 -0.048074 0.056598 0.043331 -0.001495
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##      0, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2
## 0.341290 0.384676 1.280176 0.390203 -0.006849 0.008312
##      sma1      sma2 (Intercept)      x
## 0.022874 -0.008253 0.049298 -0.001407
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##      seasonal = list(order = c(1, 1, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1      x
## -0.153426 0.851294 -0.105018 -1.000000 -0.001782
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
##      1, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2      sma1      sma2
## 0.828431 -0.012423 -0.208425 -0.791575 -0.943022 0.016422 -0.040905 -0.959094
##      x
## -0.001457
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

```

```

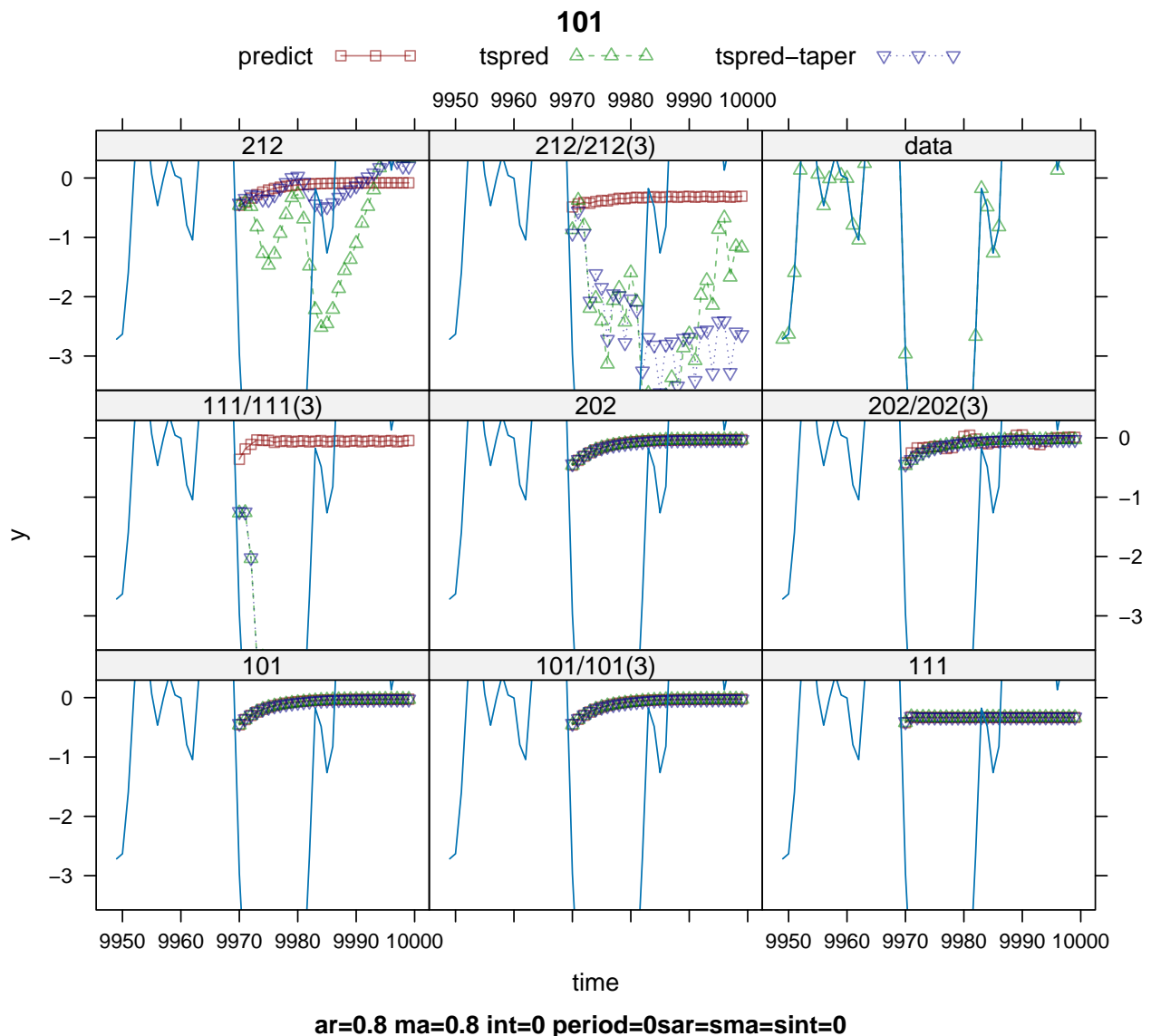
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1

## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained

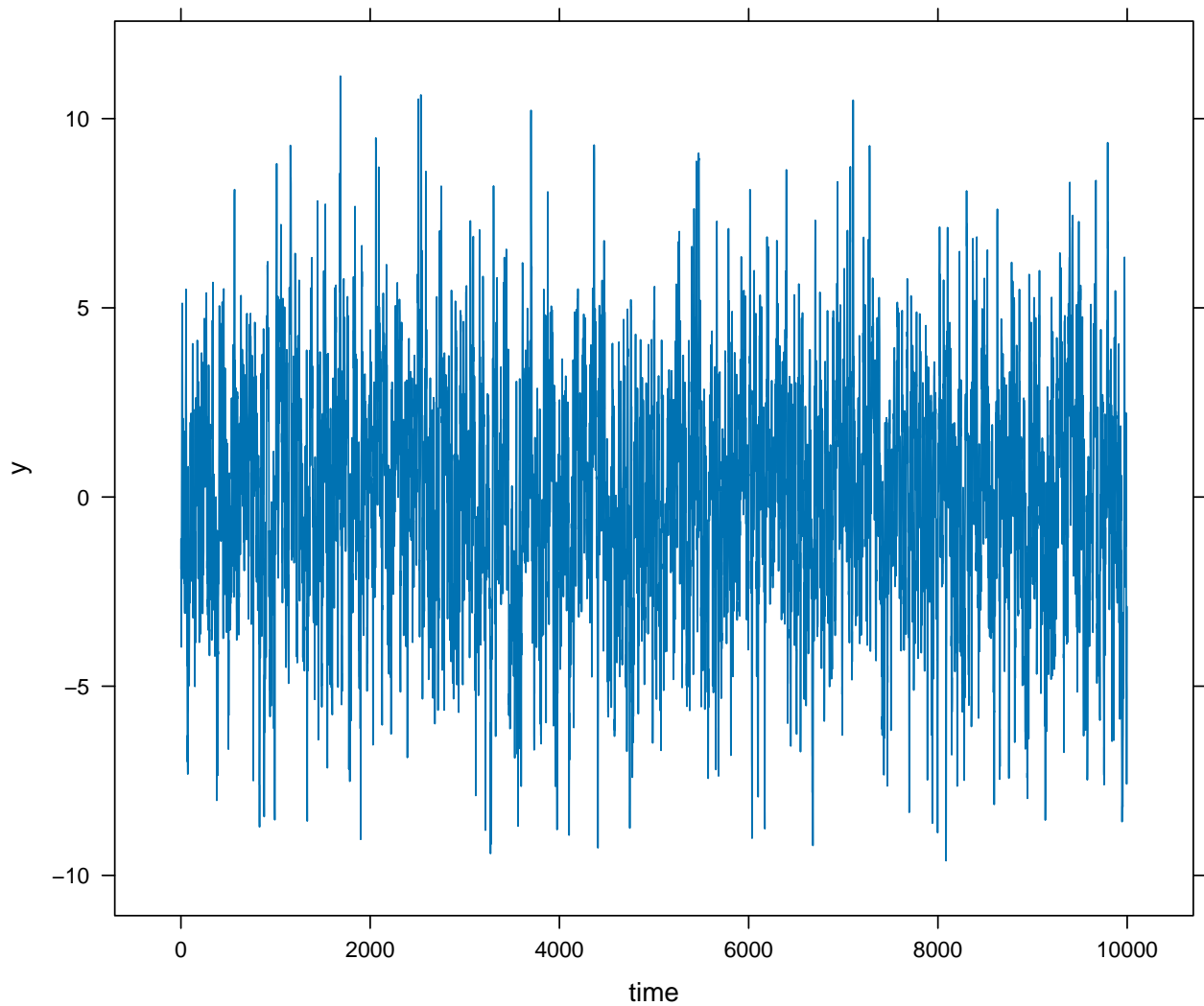
```



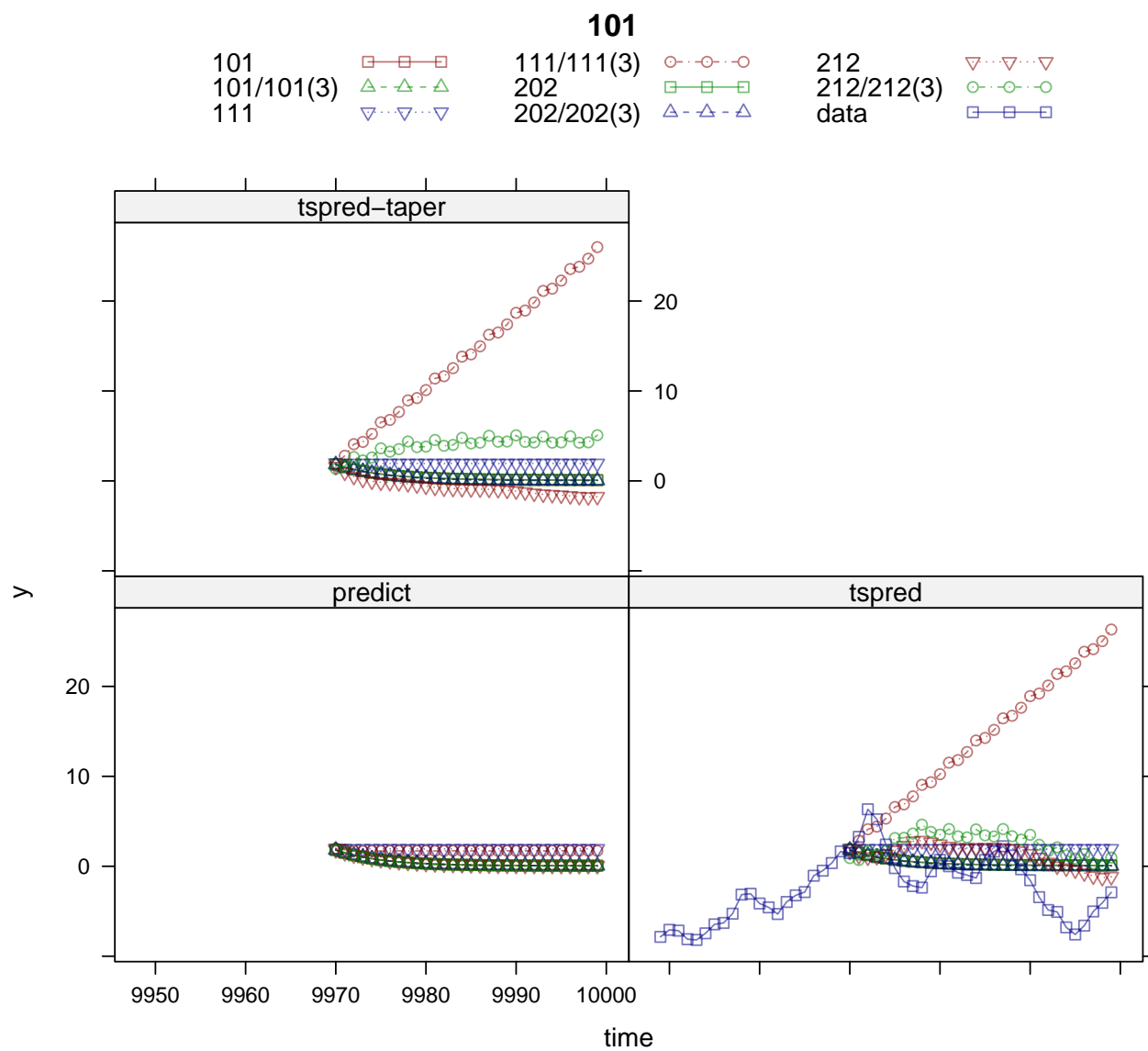
```

##          model
## method    101 101/101(3) 111 111/111(3) 202 202/202(3) 212 212/212(3) data
## predict      30      30 30      30 30      30 30      30 0
## tspred       30      30 30      30 30      30 30      30 51
## tspred-taper 30      30 30      30 30      30 30      30 0
## Total        90      90 90      90 90      90 90      90 51
##          model
## method      Total
## predict      240
## tspred       291
## tspred-taper 240
## Total        771

```

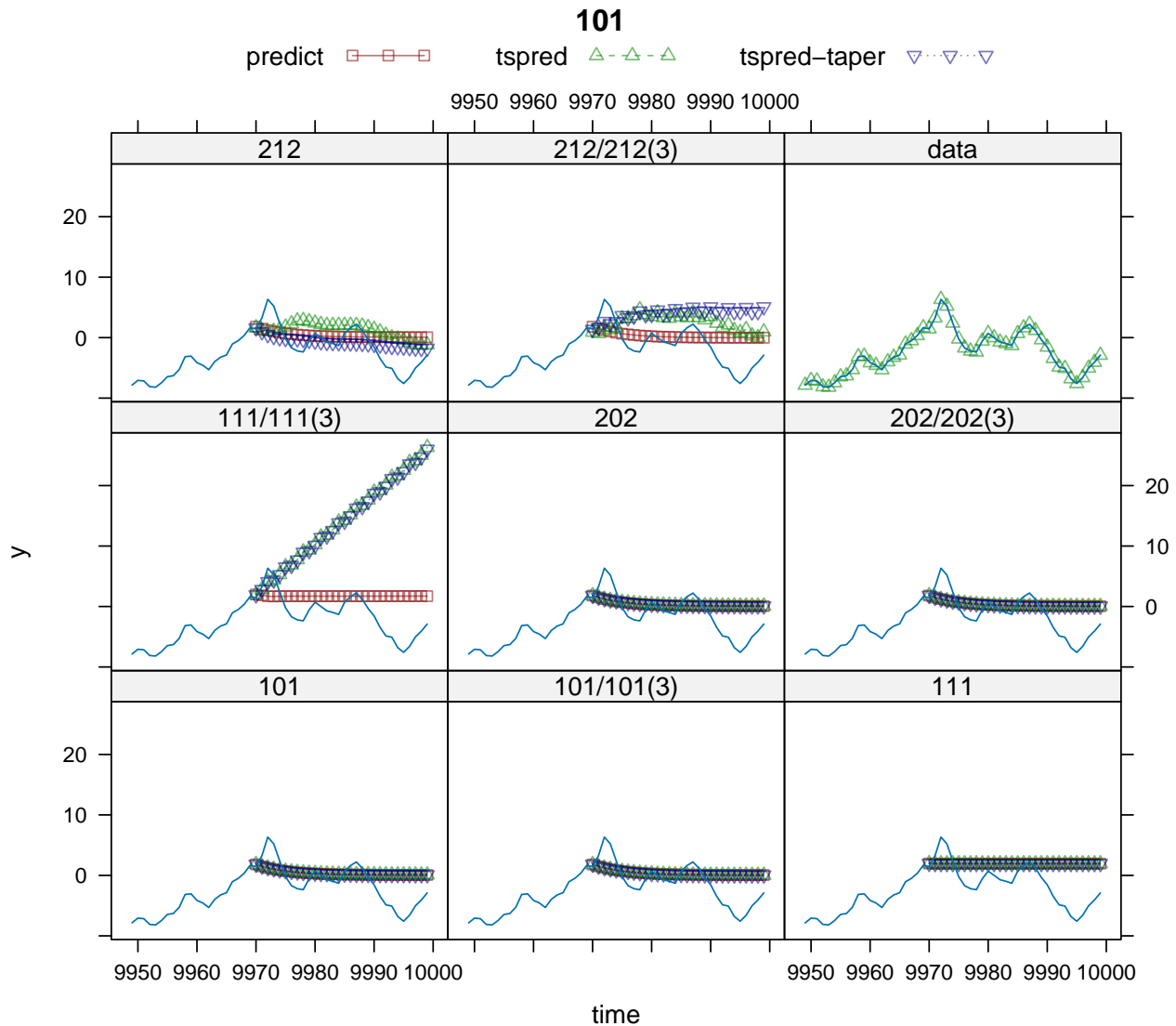


ar=0.8 ma=0.8 int=0 period=0sar=sma=sint=0



ar=0.8 ma=0.8 int=0 period=0sar=sma=sint=0

[1] " "



ar=0.8 ma=0.8 int=0 period=0sar=sma=sint=0

```
## [1] " "
## $`101`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -3.717127 -0.675618 -0.002989  0.684380  4.175928
##
## Estimates:
##              Estimate Std. Error z value Pr(>|z|)
## ar1           0.820141   0.005840 140.431  <2e-16 ***
## ma1           0.799003   0.006068 131.668  <2e-16 ***
## (Intercept)  0.052344   0.100533   0.521    0.603
## x           -0.001518   0.003501  -0.434    0.665
## ---
```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.006
## Log-likelihood = -14240
## AIC = 28500
##
## $`202`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -3.718077 -0.678699 -0.004183  0.684010  4.164553
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1      0.134871   0.241491   0.558  0.57651
## ar2      0.559578   0.198718   2.816  0.00486 **
## ma1      1.489085   0.240217   6.199 5.68e-10 ***
## ma2      0.554095   0.191333   2.896  0.00378 **
## (Intercept) 0.053250   0.100100   0.532  0.59475
## x      -0.001455   0.003503  -0.415  0.67778
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.006
## Log-likelihood = -14240
## AIC = 28500
##
## $`111`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -3.50512 -0.70913  0.00416  0.70435  4.67310
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.108398   0.012452  -8.705 <2e-16 ***
## ma1  0.817691   0.006916 118.228 <2e-16 ***
## x   -0.001479   0.003506  -0.422  0.673
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.05
## Log-likelihood = -14670
## AIC = 29350
##
## $`212`
##
## Call:

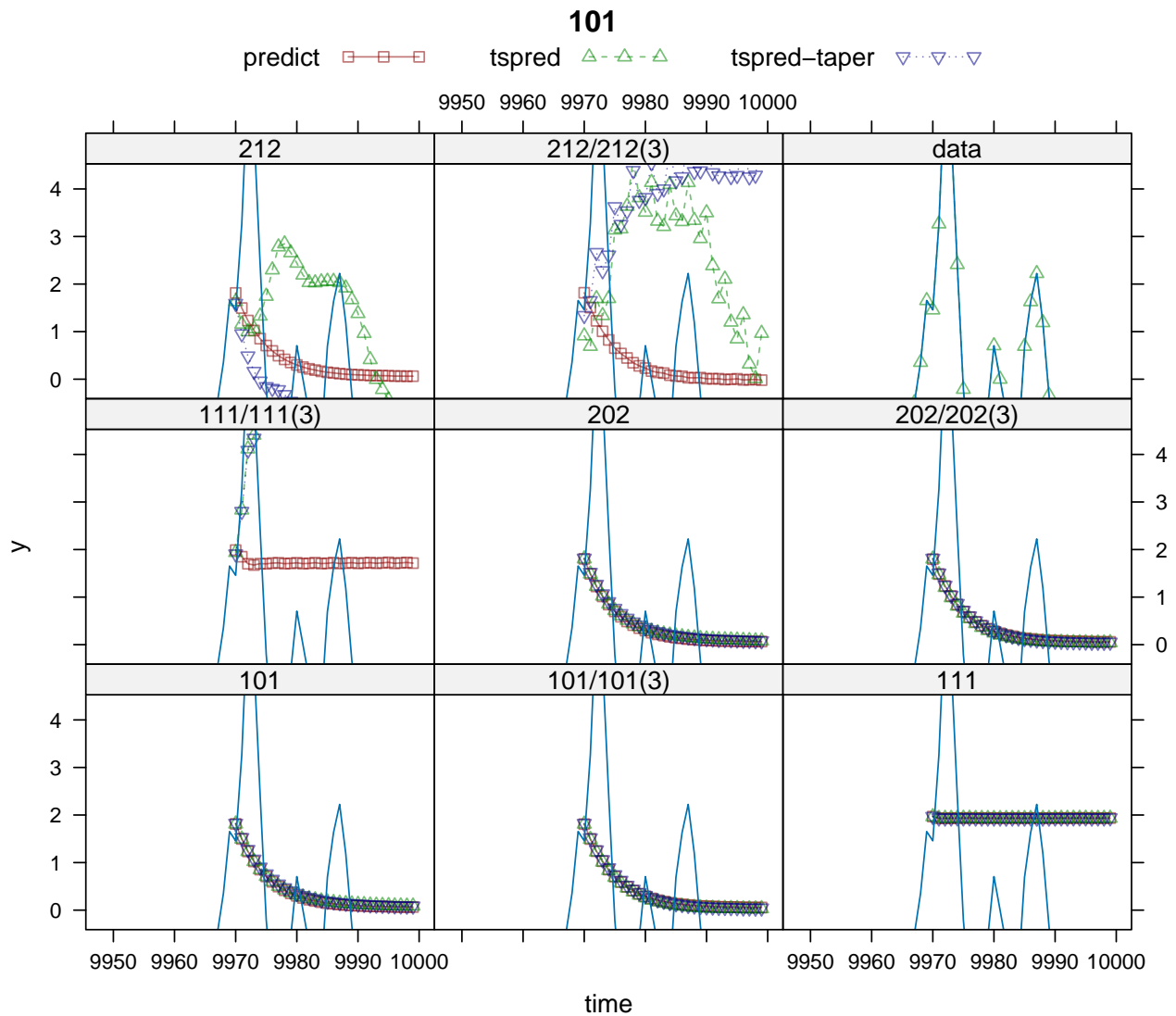
```

```

## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -3.718164 -0.676009 -0.001213  0.682300  4.179253
##
## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1  0.821518   0.012523   65.600 <2e-16 ***
## ar2 -0.001333   0.012431   -0.107  0.915
## ma1 -0.201505   0.007576  -26.598 <2e-16 ***
## ma2 -0.798495   0.007568 -105.503 <2e-16 ***
## x   -0.001514   0.003501   -0.432  0.665
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.006
## Log-likelihood = -14250
## AIC = 28500
##
## $`101/101(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##      seasonal = list(order = c(1, 0, 1), period = 3))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -3.710765 -0.674651 -0.005381  0.682823  4.183246
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1          0.818544   0.006462 126.678 <2e-16 ***
## ma1          0.797471   0.006487 122.935 <2e-16 ***
## sar1        -0.048074   0.866101  -0.056  0.956
## sma1         0.056598   0.861734   0.066  0.948
## (Intercept) 0.043331   0.100372   0.432  0.666
## x          -0.001495   0.003500  -0.427  0.669
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.006
## Log-likelihood = -14240
## AIC = 28500
##
## $`202/202(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##      0, 2), period = 3))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -3.714162 -0.678556 -0.001288  0.683504  4.172973

```

```
## Warning in sqrt(diag(x$vcov)): NaNs produced
```



ar=0.8 ma=0.8 int=0 period=0sar=sma=sint=0

```
## Estimates:
##           Estimate Std. Error z value Pr(>|z|)
## ar1      0.341290   0.437064   0.781 0.43488
## ar2      0.384676   0.361059   1.065 0.28669
## ma1      1.280176   0.436828   2.931 0.00338 **
## ma2      0.390203   0.345783   1.128 0.25913
## sar1     -0.006849      NaN      NaN      NaN
## sar2      0.008312      NaN      NaN      NaN
## sma1      0.022874      NaN      NaN      NaN
## sma2     -0.008253      NaN      NaN      NaN
## (Intercept) 0.049298  0.099512   0.495 0.62032
## x        -0.001407  0.003503  -0.402 0.68786
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.005
```

```

## Log-likelihood = -14240
## AIC = 28510
##
## $`111/111(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##     seasonal = list(order = c(1, 1, 1), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -3.545127 -0.703210 -0.003602  0.704918  4.556834
##
## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1  -0.1534257  0.0126591  -12.120 <2e-16 ***
## ma1   0.8512940  0.0067949  125.285 <2e-16 ***
## sar1 -0.1050178  0.0115034   -9.129 <2e-16 ***
## sma1 -0.9999996  0.0007179 -1393.005 <2e-16 ***
## x    -0.0017821  0.0035349   -0.504  0.614
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.045
## Log-likelihood = -14640
## AIC = 29290
##
## $`212/212(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
##     1, 2), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -3.74340 -0.68475 -0.01155  0.67147  4.21967
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1   0.828431  0.014555  56.916 <2e-16 ***
## ar2  -0.012423  0.016889  -0.736  0.462
## ma1  -0.208425  0.010681 -19.513 <2e-16 ***
## ma2  -0.791575  0.010668 -74.201 <2e-16 ***
## sar1 -0.943022  0.067388 -13.994 <2e-16 ***
## sar2  0.016422  0.014992  1.095  0.273
## sma1 -0.040905  0.065877  -0.621  0.535
## sma2 -0.959094  0.065877 -14.559 <2e-16 ***
## x    -0.001457  0.003503  -0.416  0.678
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.006
## Log-likelihood = -14260
## AIC = 28540

```

```
##           [,1]           [,2]           [,3]
## 101      summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 202      summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 111      summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 212      summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 101/101(3) summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 202/202(3) summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 111/111(3) summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 212/212(3) summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
```

AR -1.95,-, MA .8

```
replicate(3,predplot(ar=c(1.95,-.97), ma =.8, main = '201'))
```

```
## [1] 1.005155+0.1434991i 1.005155-0.1434991i
## [1] 1.015346 1.015346
## [1] 44.30864 -44.30864

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Estimates:
##      ar1      ma1 (Intercept)      x
## 0.989536 0.978359 -0.444294 0.005774

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Estimates:
##      ar1      ar2      ma1      ma2 (Intercept)      x
## 1.9396901 -0.9595543 0.8087860 0.0083633 -0.4459863 -0.0007366

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Estimates:
##      ar1      ma1      x
## 0.9486460 0.8363886 -0.0004965

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##
## Estimates:
##      ar1      ar2      ma1      ma2      x
## 1.9393659 -0.9592880 -0.1980354 -0.8015082 -0.0005311
```

```

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##     seasonal = list(order = c(1, 0, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1 (Intercept)      x
##  0.97417   0.90713   0.71616   0.27469   -0.44481   0.00127

## Note: 'data' coerced to 'ts_data_frame'

## Warning in log(s2): NaNs produced

## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##     0, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2
##  1.9383168 -0.9582431  0.8236570  0.0289013  0.0762413  0.9083645
##      sma1      sma2 (Intercept)      x
## -0.0722855 -0.9105716 -0.3617242 -0.0007668

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##     seasonal = list(order = c(1, 1, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1      x
##  0.9377710  0.8083828  0.1439363 -0.9999998 -0.0006805

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
##     1, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2      sma1
##  0.8823991 -0.6220273  1.0004969  0.9996672 -0.1610200  0.0694409  0.7461133
##      sma2      x
##  0.1207922 -0.0008881

## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

```



```

## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced

## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1

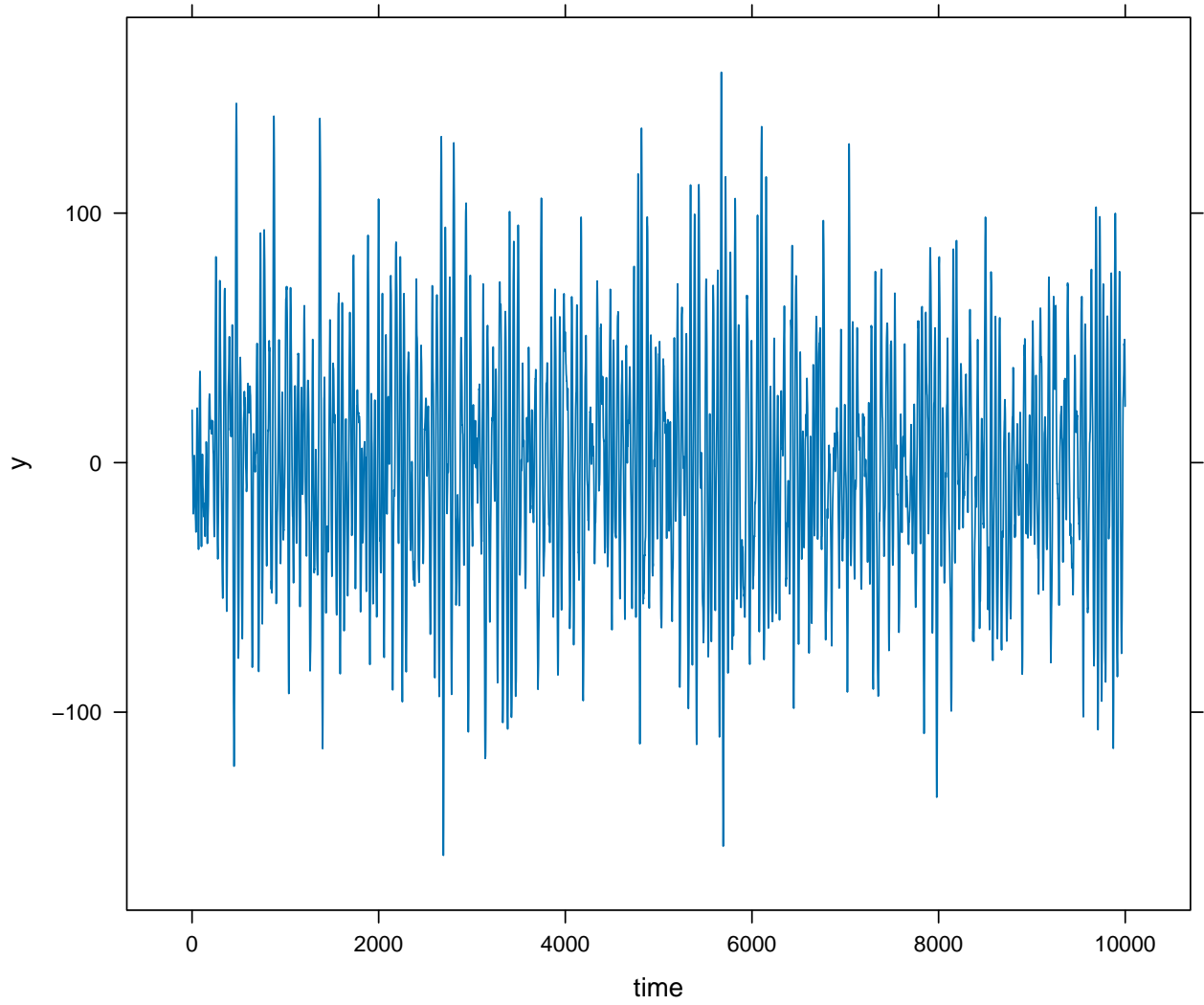
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained

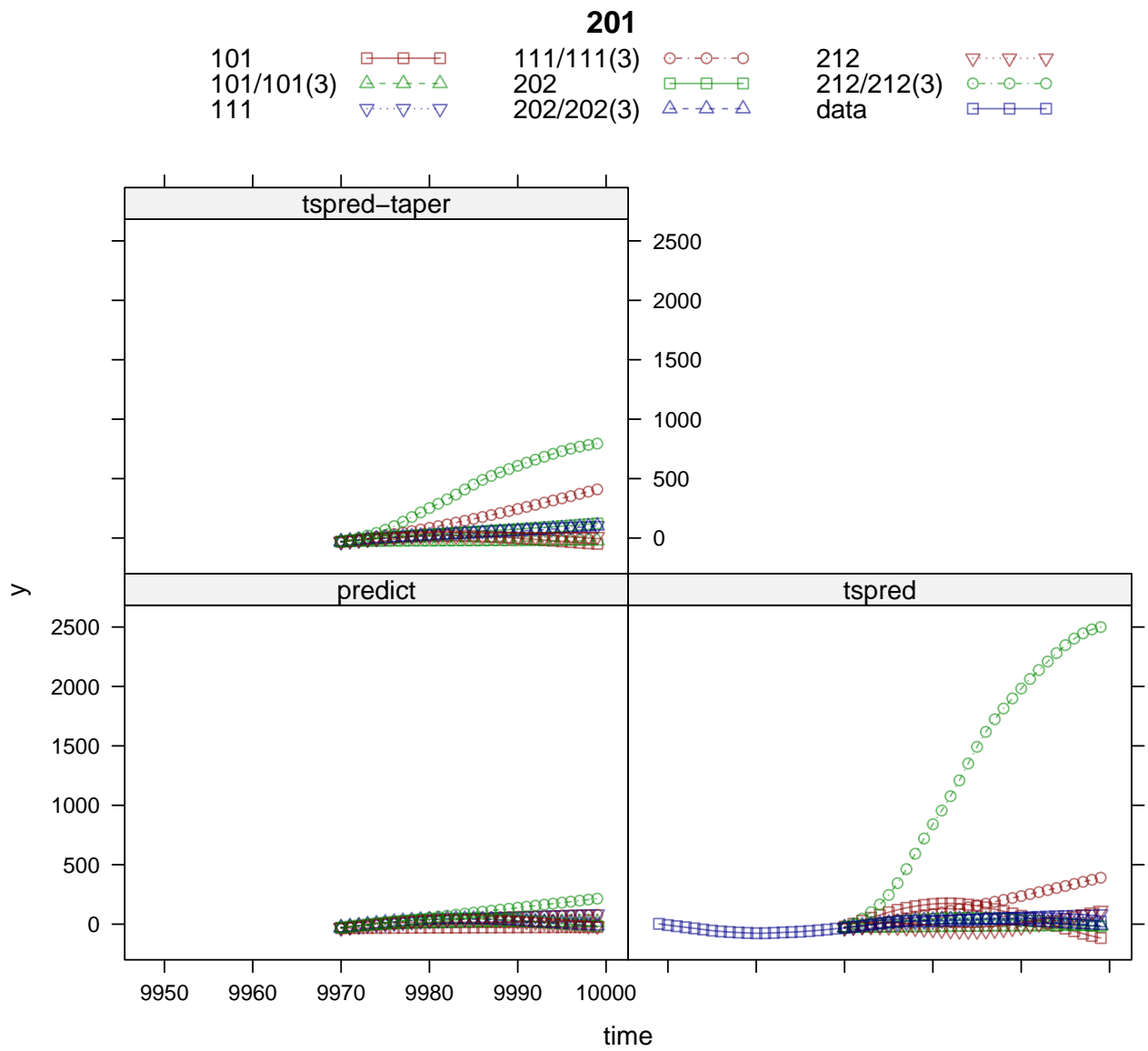
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained

##
##          model
## method      101 101/101(3) 111 111/111(3) 202 202/202(3) 212 212/212(3) data
## predict      30      30 30      30 30      30 30      30 0
## tspred       30      30 30      30 30      30 30      30 51
## tspred-taper 30      30 30      30 30      30 30      30 0
## Total        90      90 90      90 90      90 90      90 51
##
##          model
## method      Total
## predict      240
## tspred       291
## tspred-taper 240
## Total        771

```

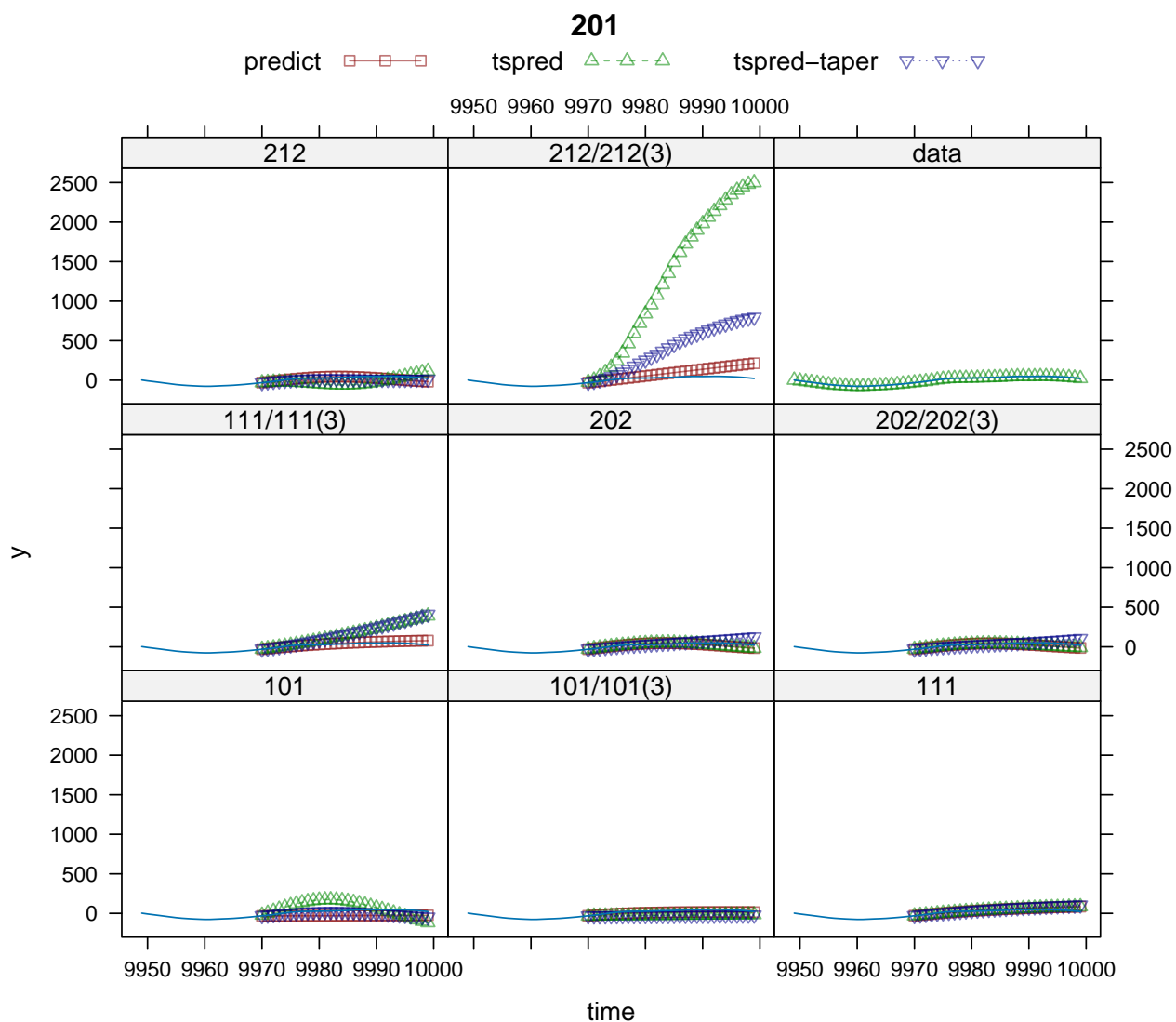


ar=1.95 -0.97 ma=0.8 int=0 period=0sar=sma=sint=0



ar=1.95 -0.97 ma=0.8 int=0 period=0sar=sma=sint=0

[1] " "



ar=1.95 -0.97 ma=0.8 int=0 period=0sar=sma=sint=0

```
## [1] "    "
## $`101`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Residuals:
##      Min       1st Q   Median       3rd Q       Max
## -12.75866  -2.28745  -0.01451   2.26800  13.64189
##
## Estimates:
##              Estimate Std. Error z value Pr(>|z|)
## ar1            0.989536   0.001426  694.115  <2e-16 ***
## ma1            0.978359   0.001554  629.521  <2e-16 ***
## (Intercept) -0.444294    6.234930  -0.071    0.943
## x              0.005774   0.003568   1.619    0.106
## ---
```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 3.328
## Log-likelihood = -26210
## AIC = 52440
##
## $`202`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -4.155470 -0.668384 -0.001393  0.678464  3.955295
##
## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1      1.9396901   0.0029409   659.557 <2e-16 ***
## ar2     -0.9595543   0.0029381  -326.590 <2e-16 ***
## ma1      0.8087860   0.0105063    76.981 <2e-16 ***
## ma2      0.0083633   0.0104801     0.798  0.425
## (Intercept) -0.4459863  0.9225514    -0.483  0.629
## x      -0.0007366   0.0016871    -0.437  0.662
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.008
## Log-likelihood = -14270
## AIC = 28560
##
## $`111`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -4.94403 -0.75083  0.00925  0.75511  4.05865
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.9486460   0.0031696  299.298 <2e-16 ***
## ma1  0.8363886   0.0049181  170.063 <2e-16 ***
## x   -0.0004965   0.0017031   -0.292  0.771
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.125
## Log-likelihood = -15360
## AIC = 30740
##
## $`212`
##
## Call:

```

```

## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -4.14624 -0.68222 -0.01464  0.66838  3.95208
##
## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1  1.9393659  0.0027689  700.414 <2e-16 ***
## ar2 -0.9592880  0.0027683 -346.531 <2e-16 ***
## ma1 -0.1980354  0.0059298  -33.397 <2e-16 ***
## ma2 -0.8015082  0.0059184 -135.426 <2e-16 ***
## x   -0.0005311  0.0016867   -0.315  0.753
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.008
## Log-likelihood = -14270
## AIC = 28560
##
## $`101/101(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##      seasonal = list(order = c(1, 0, 1), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -7.7993 -1.3095  0.0224  1.3210  7.3360
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1      0.974174  0.002278 427.645 <2e-16 ***
## ma1      0.907132  0.003276 276.931 <2e-16 ***
## sar1     0.716157  0.007922  90.405 <2e-16 ***
## sma1     0.274686  0.009513  28.875 <2e-16 ***
## (Intercept) -0.444811  6.580516 -0.068  0.946
## x         0.001270  0.002071  0.613  0.540
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.981
## Log-likelihood = -21030
## AIC = 42070
##
## $`202/202(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##      0, 2), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -4.200589 -0.666764 -0.003176  0.677804  3.935543

```

```

##
## Estimates:
##           Estimate Std. Error  z value Pr(>|z|
## ar1         1.9383168  0.0030325  639.183 < 2e-16 ***
## ar2        -0.9582431  0.0030292 -316.338 < 2e-16 ***
## ma1         0.8236570  0.0105878  77.793 < 2e-16 ***
## ma2         0.0289013  0.0108685   2.659 0.00783 **
## sar1        0.0762413  0.1066317   0.715 0.47461
## sar2        0.9083645  0.0965304   9.410 < 2e-16 ***
## sma1       -0.0722855  0.1073094  -0.674 0.50055
## sma2       -0.9105716  0.0972242  -9.366 < 2e-16 ***
## (Intercept) -0.3617242  1.0429833  -0.347 0.72873
## x          -0.0007668  0.0016936  -0.453 0.65072
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.008
## Log-likelihood = -14280
## AIC = 28570
##
## $`111/111(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##           seasonal = list(order = c(1, 1, 1), period = 3))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -4.843502 -0.741518  0.008169   0.749786  4.343262
##
## Estimates:
##           Estimate Std. Error  z value Pr(>|z|
## ar1    0.9377710  0.0036523  256.760 <2e-16 ***
## ma1    0.8083828  0.0056789  142.347 <2e-16 ***
## sar1   0.1439363  0.0111023  12.965 <2e-16 ***
## sma1 -0.9999998  0.0007500 -1333.396 <2e-16 ***
## x     -0.0006805  0.0016839  -0.404  0.686
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.116
## Log-likelihood = -15290
## AIC = 30590
##
## $`212/212(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
## 1, 2), period = 3))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -4.959687 -0.749611  0.007967   0.755928  4.659822
##

```

```

## Estimates:
##      Estimate Std. Error  z value Pr(>|z|
## ar1    0.8823991  0.0081275  108.570 < 2e-16 ***
## ar2   -0.6220273  0.0117792  -52.807 < 2e-16 ***
## ma1    1.0004969  0.0004719 2120.362 < 2e-16 ***
## ma2    0.9996672  0.0010237  976.490 < 2e-16 ***
## sar1  -0.1610200  0.2325875   -0.692 0.48875
## sar2   0.0694409  0.0398335    1.743 0.08128 .
## sma1   0.7461133  0.2330836    3.201 0.00137 **
## sma2   0.1207922  0.1008545    1.198 0.23104
## x      -0.0008881  0.0017395   -0.511 0.60966
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.122
## Log-likelihood = -15340
## AIC = 30710
##
## [1] 1.005155+0.1434991i 1.005155-0.1434991i
## [1] 1.015346 1.015346
## [1] 44.30864 -44.30864

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Estimates:
##      ar1      ma1 (Intercept)      x
## 0.989349 0.980799 0.555167 -0.004071

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Estimates:
##      ar1      ar2      ma1      ma2 (Intercept)      x
## 1.947706 -0.968092 0.797049 -0.010895 0.571535 -0.001974

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Estimates:
##      ar1      ma1      x
## 0.955580 0.847396 -0.001793

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##

```



```

## Estimates:
##      ar1      ar2      ma1      ma2      x
## 1.947009 -0.967397 -0.193995 -0.806005 -0.001991

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##      seasonal = list(order = c(1, 0, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1 (Intercept)      x
## 0.974768 0.907311 0.734641 0.323048 0.554058 -0.001486

## Note: 'data' coerced to 'ts_data_frame'

## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##      0, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2
## 1.946364 -0.966875 0.798344 -0.007246 0.516107 -0.144743
##      sma1      sma2 (Intercept)      x
## -0.503529 0.147947 0.545151 -0.001989

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##      seasonal = list(order = c(1, 1, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1      x
## 0.941745 0.807061 0.202121 -1.000000 -0.001726

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
##      1, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2      sma1      sma2
## 0.882422 -0.605312 0.999771 0.999419 -0.518256 0.186580 1.129479 0.266875
##      x
## -0.002054

## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

```

```

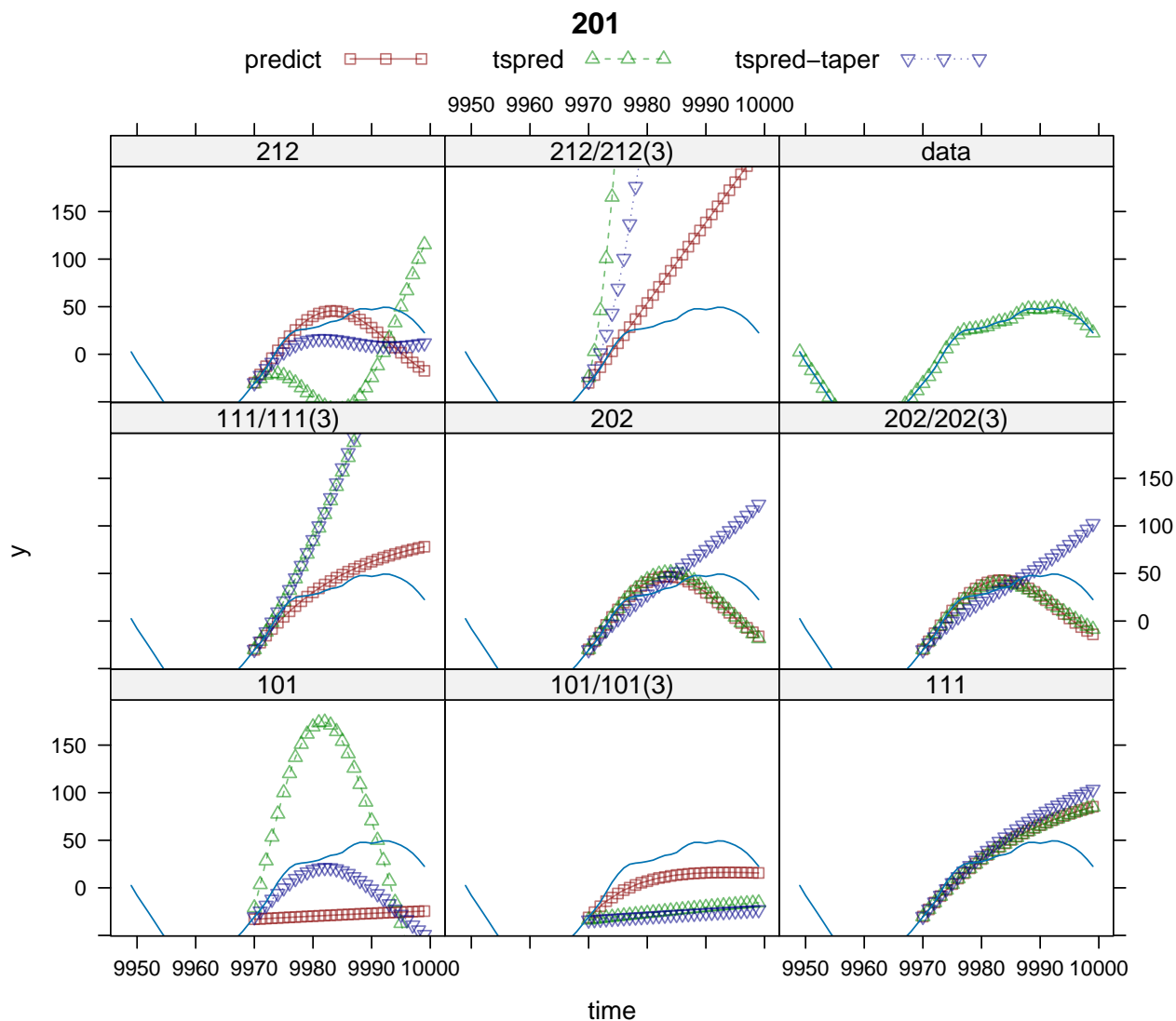
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1

## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

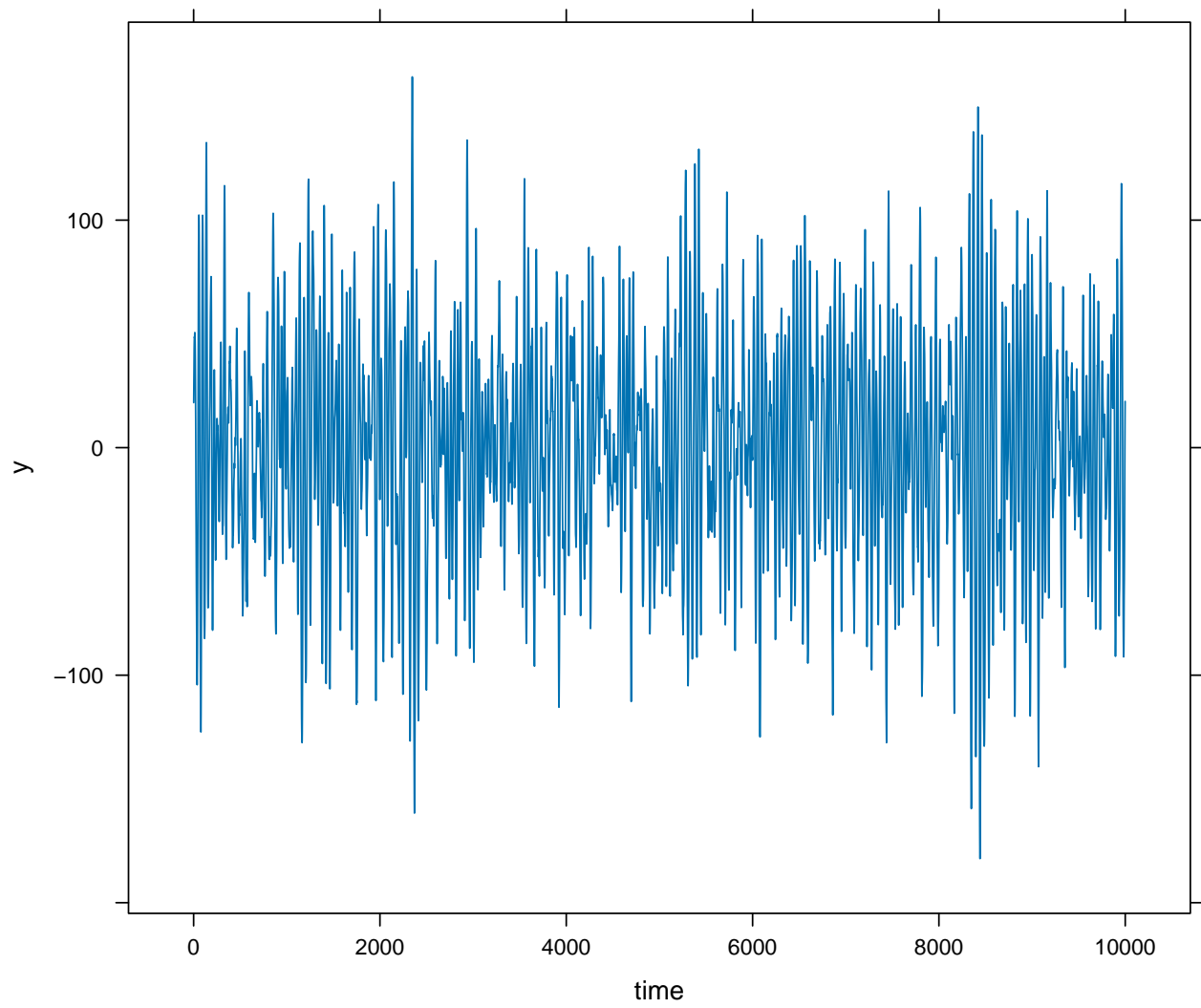
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained

```



ar=1.95 -0.97 ma=0.8 int=0 period=0sar=sma=sint=0

```
##          model
## method    101 101/101(3) 111 111/111(3) 202 202/202(3) 212 212/212(3) data
## predict      30      30 30      30 30      30 30      30 0
## tspred       30      30 30      30 30      30 30      30 51
## tspred-taper 30      30 30      30 30      30 30      30 0
## Total        90      90 90      90 90      90 90      90 51
##          model
## method      Total
## predict      240
## tspred       291
## tspred-taper 240
## Total        771
```



201

101
101/101(3)
111

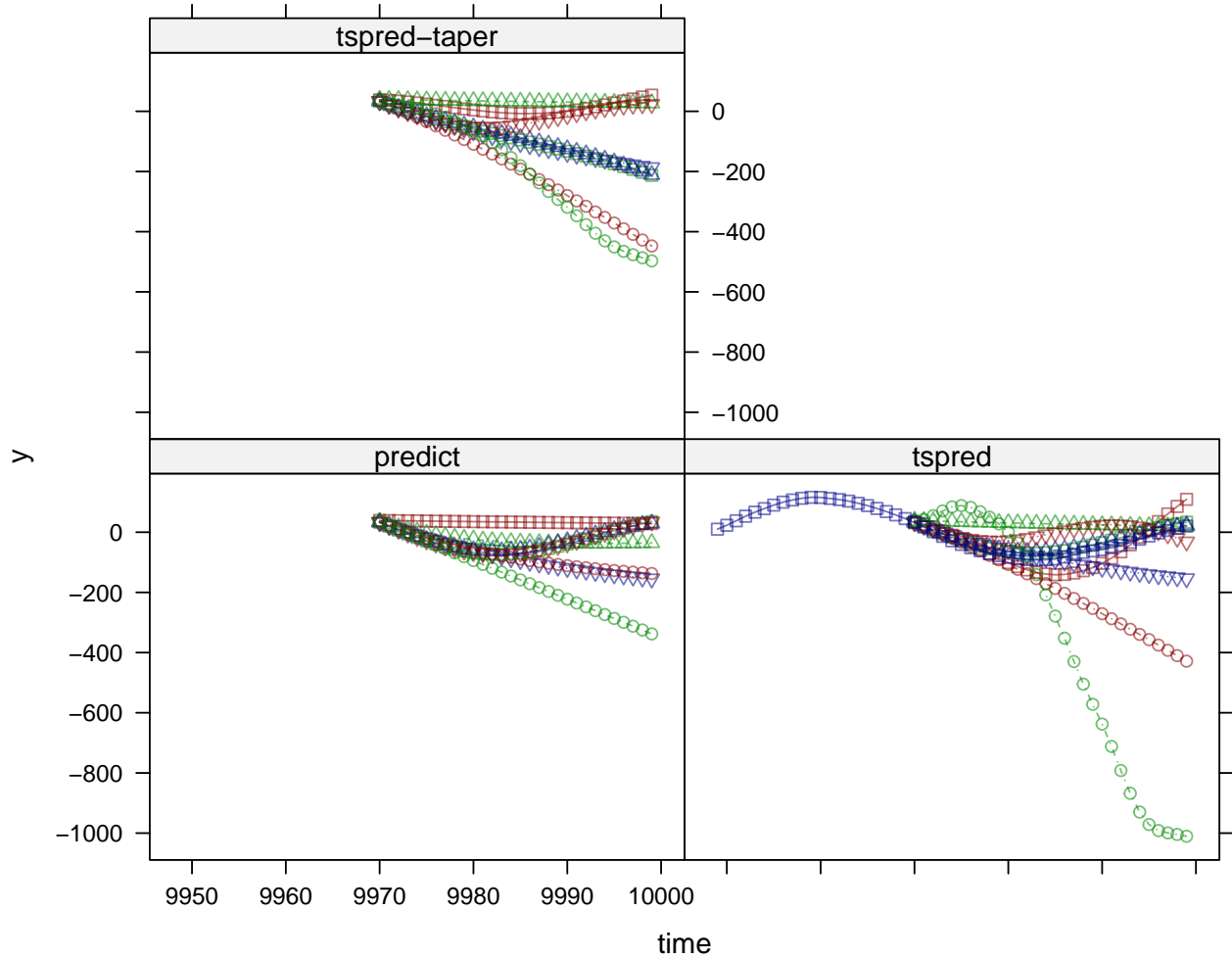
□-□-□
△-△-△
▽-▽-▽

111/111(3)
202
202/202(3)

○-○-○
□-□-□
△-△-△

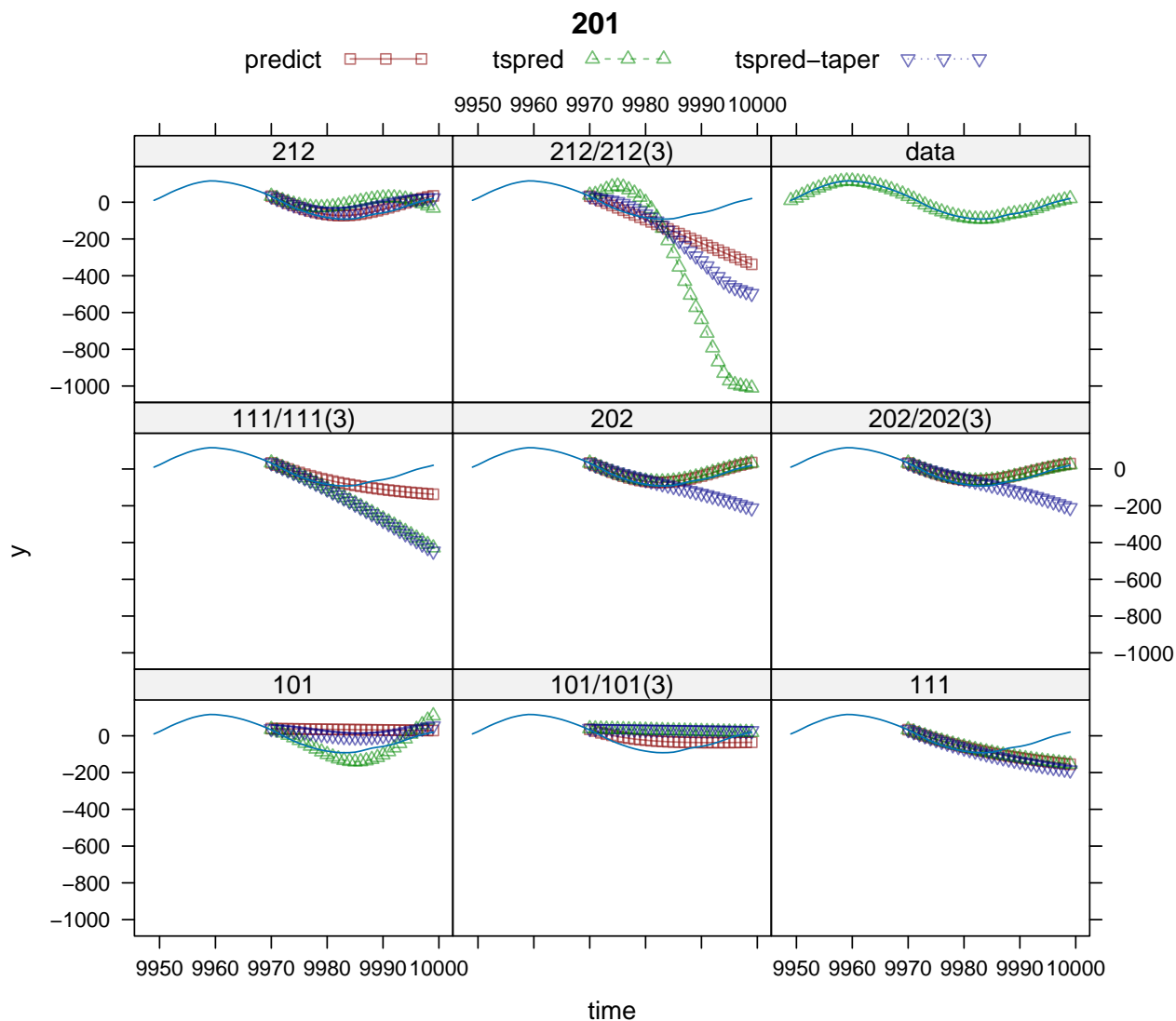
212
212/212(3)
data

▽-▽-▽
○-○-○
□-□-□



ar=1.95 -0.97 ma=0.8 int=0 period=0sar=sma=sint=0

[1] " "



ar=1.95 -0.97 ma=0.8 int=0 period=0sar=sma=sint=0

```
## [1] "    "
## $`101`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Residuals:
##      Min       1st Q   Median       3rd Q       Max
## -14.64111  -2.50820   0.07628   2.51649  12.99907
##
## Estimates:
##              Estimate Std. Error z value Pr(>|z|)
## ar1            0.989349   0.001442  686.030 <2e-16 ***
## ma1            0.980799   0.001390  705.661 <2e-16 ***
## (Intercept)    0.555167   6.705078   0.083   0.934
## x             -0.004071   0.003382  -1.204   0.229
## ---
```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 3.636
## Log-likelihood = -27100
## AIC = 54210
##
## $`202`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -3.9478536 -0.6768275  0.0001911  0.6860088  3.5371617
##
## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1      1.947706   0.002588  752.492 <2e-16 ***
## ar2     -0.968092   0.002586 -374.324 <2e-16 ***
## ma1      0.797049   0.010228  77.929 <2e-16 ***
## ma2     -0.010895   0.010166  -1.072  0.284
## (Intercept) 0.571535   0.875952   0.652  0.514
## x       -0.001974   0.001615  -1.222  0.222
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.9988
## Log-likelihood = -14180
## AIC = 28380
##
## $`111`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -4.347369 -0.772027 -0.001993  0.784317  3.930312
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.955580   0.002950 323.900 <2e-16 ***
## ma1  0.847396   0.004775 177.466 <2e-16 ***
## x   -0.001793   0.001619  -1.108  0.268
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.141
## Log-likelihood = -15510
## AIC = 31030
##
## $`212`
##
## Call:

```

```

## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -3.925471 -0.677463  0.001788  0.684832  3.533820
##
## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1  1.947009    0.002534  768.498 <2e-16 ***
## ar2 -0.967397    0.002533 -381.843 <2e-16 ***
## ma1 -0.193995    0.006023  -32.209 <2e-16 ***
## ma2 -0.806005    0.006016 -133.971 <2e-16 ***
## x   -0.001991    0.001613   -1.235   0.217
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.9989
## Log-likelihood = -14180
## AIC = 28380
##
## $`101/101(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##      seasonal = list(order = c(1, 0, 1), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -7.48500 -1.35930  0.03834  1.37190  8.04713
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1          0.974768    0.002246 433.910 <2e-16 ***
## ma1          0.907311    0.003270 277.443 <2e-16 ***
## sar1         0.734641    0.007511  97.807 <2e-16 ***
## sma1         0.323048    0.009029  35.779 <2e-16 ***
## (Intercept)  0.554058    7.859829   0.070   0.944
## x          -0.001486    0.001977  -0.752   0.452
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 2.032
## Log-likelihood = -21290
## AIC = 42590
##
## $`202/202(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##      0, 2), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -3.956602 -0.678278  0.001695  0.688749  3.541834

```



```
## Warning in sqrt(diag(x$vcov)): NaNs produced

## Estimates:
##           Estimate Std. Error  z value Pr(>|z|)
## ar1         1.946364   0.002801  695.003 <2e-16 ***
## ar2        -0.966875   0.002773 -348.727 <2e-16 ***
## ma1         0.798344   0.010286  77.614 <2e-16 ***
## ma2        -0.007246   0.010997  -0.659  0.510
## sar1         0.516107         NaN      NaN      NaN
## sar2        -0.144743         NaN      NaN      NaN
## sma1        -0.503529         NaN      NaN      NaN
## sma2         0.147947         NaN      NaN      NaN
## (Intercept)  0.545151   0.894829   0.609  0.542
## x           -0.001989   0.001616  -1.231  0.218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.9987
## Log-likelihood = -14180
## AIC = 28380
##
## $`111/111(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##           seasonal = list(order = c(1, 1, 1), period = 3))
##
## Residuals:
##           Min           1st Q       Median        3rd Q          Max
## -4.313018 -0.758250 -0.007852  0.764069  3.815499
##
## Estimates:
##           Estimate Std. Error  z value Pr(>|z|)
## ar1    0.9417447   0.0035370  266.254 <2e-16 ***
## ma1    0.8070609   0.0057595  140.128 <2e-16 ***
## sar1   0.2021211   0.0109546   18.451 <2e-16 ***
## sma1  -0.9999998   0.0007711 -1296.909 <2e-16 ***
## x     -0.0017264   0.0015969   -1.081  0.28
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.123
## Log-likelihood = -15350
## AIC = 30710
##
## $`212/212(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
##           1, 2), period = 3))
##
## Residuals:
##           Min           1st Q       Median        3rd Q          Max
## -4.28606 -0.76681 -0.00602  0.77188  3.95191
```

```

## Warning in sqrt(diag(x$vcov)): NaNs produced

## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1    0.8824215  0.0082289  107.235 <2e-16 ***
## ar2   -0.6053117  0.0125303  -48.308 <2e-16 ***
## ma1    0.9997706  0.0004520 2212.053 <2e-16 ***
## ma2    0.9994185  0.0008596 1162.618 <2e-16 ***
## sar1  -0.5182564      NaN      NaN      NaN
## sar2   0.1865801      NaN      NaN      NaN
## sma1   1.1294785      NaN      NaN      NaN
## sma2   0.2668754      NaN      NaN      NaN
## x     -0.0020540  0.0017035  -1.206   0.228
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.126
## Log-likelihood = -15380
## AIC = 30780
##
## [1] 1.005155+0.1434991i 1.005155-0.1434991i
## [1] 1.015346 1.015346
## [1] 44.30864 -44.30864

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Estimates:
##      ar1      ma1 (Intercept)      x
## 0.9895592 0.9820547 1.9633077 0.0005238

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Estimates:
##      ar1      ar2      ma1      ma2 (Intercept)      x
## 1.9519994 -0.9722557 0.7879633 -0.0136643 1.9991681 -0.0008512

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Estimates:
##      ar1      ma1      x
## 0.959581 0.846355 -0.001103

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:

```

```

## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##
## Estimates:
##      ar1      ar2      ma1      ma2      x
##  1.9510881 -0.9713507 -0.2009557 -0.7988592 -0.0008215
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##      seasonal = list(order = c(1, 0, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1 (Intercept)      x
##    0.975638    0.905256    0.749088    0.337995    1.963009   -0.001864
## Note: 'data' coerced to 'ts_data_frame'
## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##      0, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2
##  1.9794036 -0.9998714    0.8268963    0.0500693    0.6735075   -0.4106530
##      sma1      sma2 (Intercept)      x
## -0.6836357    0.3799768    2.0434794   -0.0003122
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##      seasonal = list(order = c(1, 1, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1      x
##  0.945862    0.802645    0.213609 -1.000000 -0.001222
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
##      1, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2      sma1      sma2
##  0.887775 -0.589147    1.000197    0.999835 -0.154274    0.115396    0.754365    0.101805
##      x
## -0.001011
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

```

```

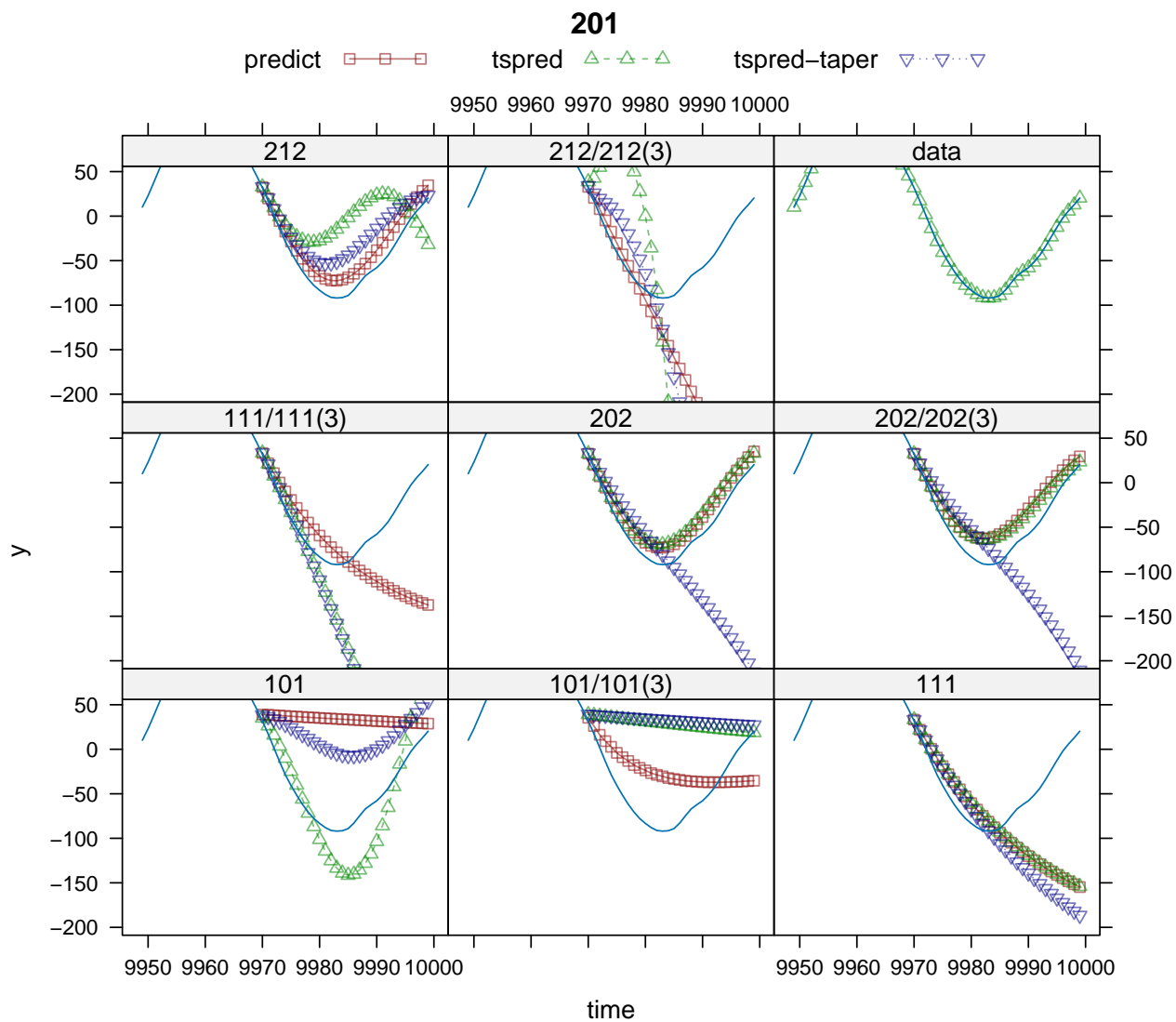
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1

## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

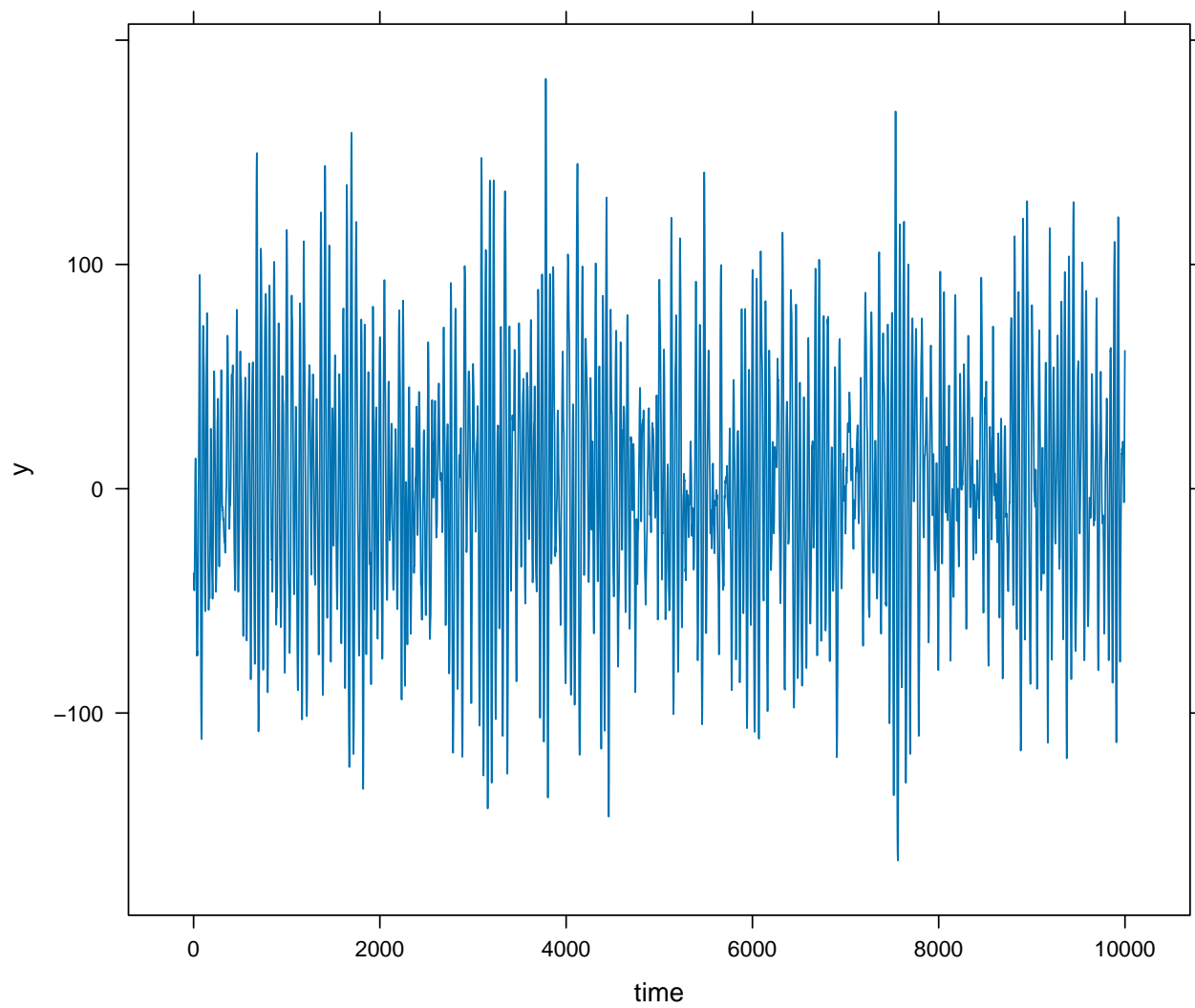
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained

```

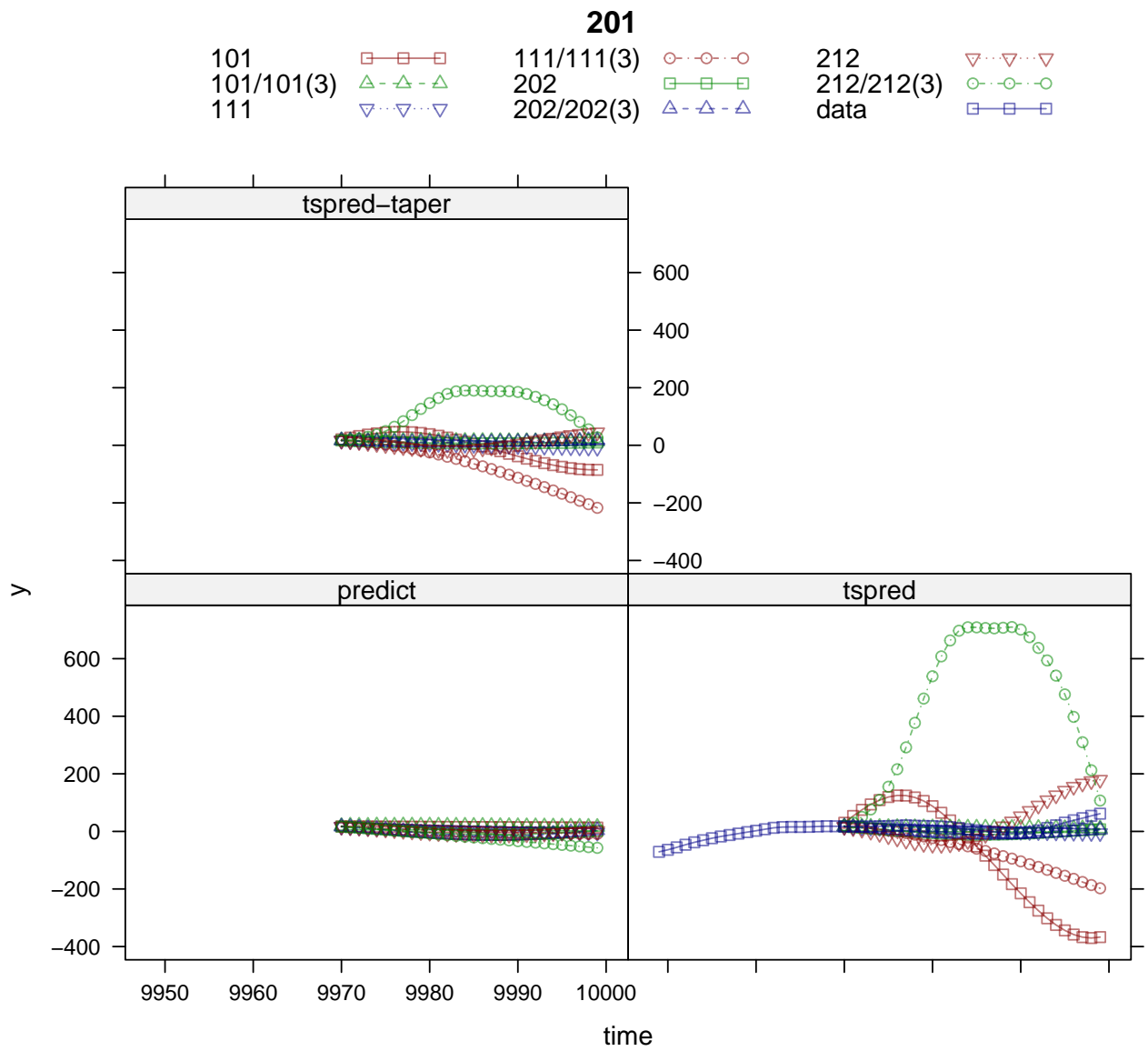


ar=1.95 -0.97 ma=0.8 int=0 period=0sar=sma=sint=0

```
##          model
## method   101 101/101(3) 111 111/111(3) 202 202/202(3) 212 212/212(3) data
## predict      30      30 30      30 30      30 30      30 0
## tspred       30      30 30      30 30      30 30      30 51
## tspred-taper 30      30 30      30 30      30 30      30 0
## Total        90      90 90      90 90      90 90      90 51
##          model
## method   Total
## predict      240
## tspred       291
## tspred-taper 240
## Total        771
```

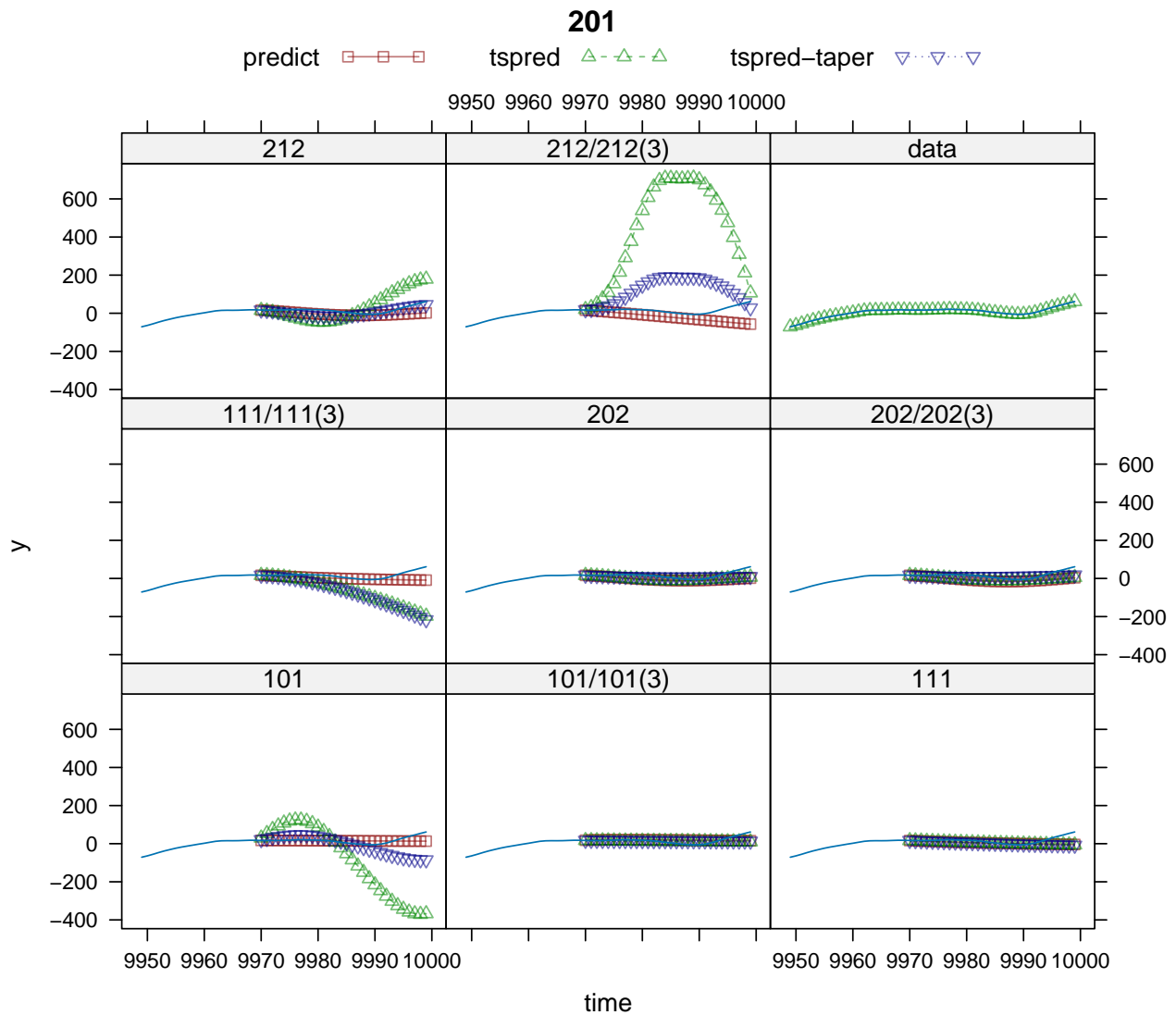


ar=1.95 -0.97 ma=0.8 int=0 period=0sar=sma=sint=0



ar=1.95 -0.97 ma=0.8 int=0 period=0sar=sma=sint=0

[1] " "



ar=1.95 -0.97 ma=0.8 int=0 period=0sar=sma=sint=0

```
## [1] " "
## $`101`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Residuals:
##      Min       1st Q   Median       3rd Q       Max
## -13.05118  -2.65358   0.03864   2.70457  12.20511
##
## Estimates:
##              Estimate Std. Error z value Pr(>|z|)
## ar1            0.9895592  0.0014389  687.742  <2e-16 ***
## ma1            0.9820547  0.0013538  725.380  <2e-16 ***
## (Intercept)    1.9633077  7.3260934   0.268    0.789
## x              0.0005238  0.0035058   0.149    0.881
## ---
```



```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 3.895
## Log-likelihood = -27790
## AIC = 55590
##
## $`202`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -3.99705 -0.68350  0.01781  0.67575  3.74086
##
## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1      1.951994   0.0024180  807.279 <2e-16 ***
## ar2     -0.9722557   0.0024165 -402.339 <2e-16 ***
## ma1      0.7879633   0.0102435   76.923 <2e-16 ***
## ma2     -0.0136643   0.0102604   -1.332  0.1829
## (Intercept) 1.9991681  0.8841172    2.261  0.0237 *
## x       -0.0008512   0.0016639   -0.512  0.6089
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.009
## Log-likelihood = -14280
## AIC = 28580
##
## $`111`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -3.888536 -0.801758  0.005387  0.797362  4.514543
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.959581   0.002821 340.211 <2e-16 ***
## ma1  0.846355   0.004790 176.707 <2e-16 ***
## x   -0.001103   0.001669  -0.661  0.509
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.17
## Log-likelihood = -15760
## AIC = 31530
##
## $`212`
##
## Call:

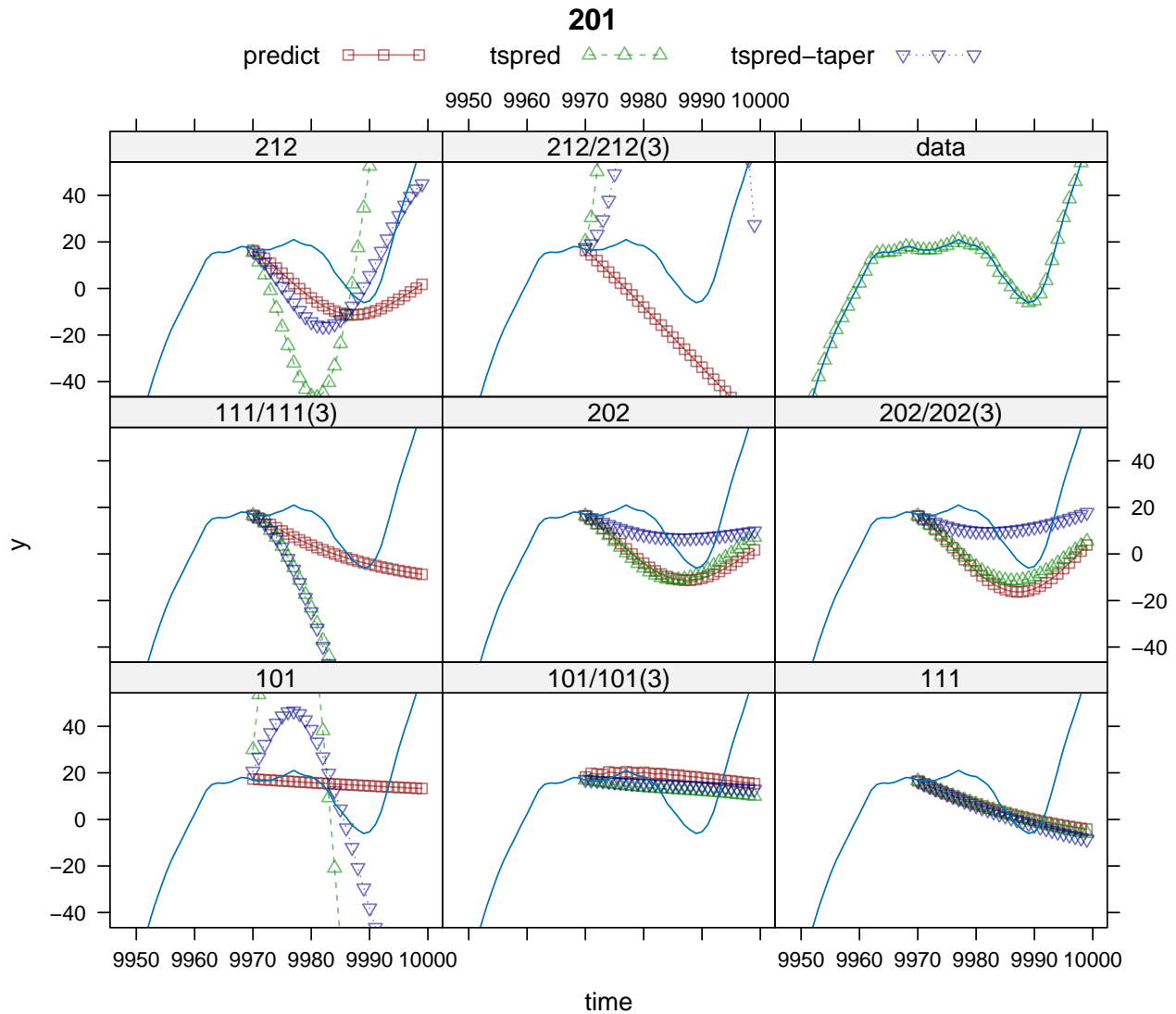
```

```

## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -3.99168 -0.67185  0.02842  0.68986  3.73629
##
## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1  1.9510881  0.0023759  821.188 <2e-16 ***
## ar2 -0.9713507  0.0023761 -408.809 <2e-16 ***
## ma1 -0.2009557  0.0062214  -32.301 <2e-16 ***
## ma2 -0.7988592  0.0062056 -128.732 <2e-16 ***
## x   -0.0008215  0.0016656  -0.493  0.622
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.009
## Log-likelihood = -14290
## AIC = 28580
##
## $`101/101(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##      seasonal = list(order = c(1, 0, 1), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -7.573e+00 -1.448e+00 -5.753e-05  1.470e+00  6.810e+00
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1      0.975638  0.002207 442.021 <2e-16 ***
## ma1      0.905256  0.003341 270.985 <2e-16 ***
## sar1     0.749088  0.007263 103.131 <2e-16 ***
## sma1     0.337995  0.008937  37.819 <2e-16 ***
## (Intercept) 1.963009  8.732816  0.225  0.822
## x        -0.001864  0.001912  -0.975  0.330
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 2.101
## Log-likelihood = -21620
## AIC = 43260
##
## $`202/202(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##      0, 2), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -3.96257 -0.68710  0.01025  0.68003  3.87816

```

```
## Warning in sqrt(diag(x$vcov)): NaNs produced
```



ar=1.95 -0.97 ma=0.8 int=0 period=0sar=sma=sint=0

```
## Estimates:
```

```
##           Estimate Std. Error  z value Pr(>|z|)
## ar1      1.9794036   0.0003079 6429.448 < 2e-16 ***
## ar2     -0.9998714         NaN      NaN      NaN
## ma1      0.8268963   0.0104555  79.087 < 2e-16 ***
## ma2      0.0500693   0.0107645   4.651 3.3e-06 ***
## sar1      0.6735075         NaN      NaN      NaN
## sar2     -0.4106530         NaN      NaN      NaN
## sma1     -0.6836357         NaN      NaN      NaN
## sma2      0.3799768         NaN      NaN      NaN
## (Intercept) 2.0434794   0.8827025   2.315 0.0206 *
## x        -0.0003122   0.0017088  -0.183 0.8550
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Residual standard deviation: 1.019
```

```

## Log-likelihood = -14370
## AIC = 28770
##
## $`111/111(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##     seasonal = list(order = c(1, 1, 1), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -4.05944 -0.77684  0.01004  0.78091  4.48164
##
## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1    0.9458618  0.0034071   277.615 <2e-16 ***
## ma1    0.8026452  0.0058286   137.707 <2e-16 ***
## sar1   0.2136095  0.0109240    19.554 <2e-16 ***
## sma1  -0.9999995  0.0006734 -1484.998 <2e-16 ***
## x     -0.0012221  0.0016490    -0.741  0.459
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.149
## Log-likelihood = -15580
## AIC = 31170
##
## $`212/212(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
##     1, 2), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -4.227128 -0.778278  0.009571  0.774281  4.184878
##
## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1    0.8877747  0.0083643   106.139 < 2e-16 ***
## ar2   -0.5891474  0.0126908   -46.423 < 2e-16 ***
## ma1    1.0001970  0.0004406  2270.306 < 2e-16 ***
## ma2    0.9998353  0.0006831 1463.581 < 2e-16 ***
## sar1  -0.1542741  0.1562295    -0.987 0.323406
## sar2   0.1153960  0.0332160     3.474 0.000513 ***
## sma1   0.7543646  0.1562027     4.829 1.37e-06 ***
## sma2   0.1018054  0.0669965     1.520 0.128621
## x     -0.0010114  0.0017593    -0.575 0.565364
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.148
## Log-likelihood = -15570
## AIC = 31170

```

```
##           [,1]           [,2]           [,3]
## 101      summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 202      summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 111      summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 212      summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 101/101(3) summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 202/202(3) summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 111/111(3) summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 212/212(3) summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
```

AR .8, MA .8 PERIOD 3 MAR .8 SMA .8

```
replicate(3, predplot(
  ar=.8, ma=.8,
  period = 3, sar = .8, sma = .8,
  main = '101/101(3)'))

## [1] 1.25+0i
## [1] 1.25
## [1] Inf

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Estimates:
##      ar1      ma1 (Intercept)      x
## 0.93640 0.99417 -0.33314 0.00218

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Estimates:
##      ar1      ar2      ma1      ma2 (Intercept)      x
## 0.788319 0.190179 0.695962 -0.298787 -0.334244 0.002469

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Estimates:
##      ar1      ma1      x
## 0.014187 -0.099528 -0.005395

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##
## Estimates:
```

```

##          ar1          ar2          ma1          ma2          x
## -0.9475041 -0.7857207  1.7536826  0.7677939  0.0003943

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##          seasonal = list(order = c(1, 0, 1), period = 3))
##
## Estimates:
##          ar1          ma1          sar1          sma1 (Intercept)          x
##  0.8059088  0.8041513  0.7896331  0.8064140 -0.3618415  0.0004326

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
## 0, 2), period = 3))
##
## Estimates:
##          ar1          ar2          ma1          ma2          sar1          sar2
##  0.4779041  0.2680171  1.1317316  0.2609881  1.2316175 -0.3465912
##          sma1          sma2 (Intercept)          x
##  0.3563456 -0.3633668 -0.2532164  0.0004357

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##          seasonal = list(order = c(1, 1, 1), period = 3))
##
## Estimates:
##          ar1          ma1          sar1          sma1          x
## -0.1463450  0.8581094 -0.2289347  0.8296516  0.0005505

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
## 1, 2), period = 3))
##
## Estimates:
##          ar1          ar2          ma1          ma2          sar1          sar2          sma1
##  0.8012421  0.0080640 -0.1910588 -0.8089411  0.7743655  0.0135186 -0.1891579
##          sma2          x
## -0.8081705  0.0004432

## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

```

```

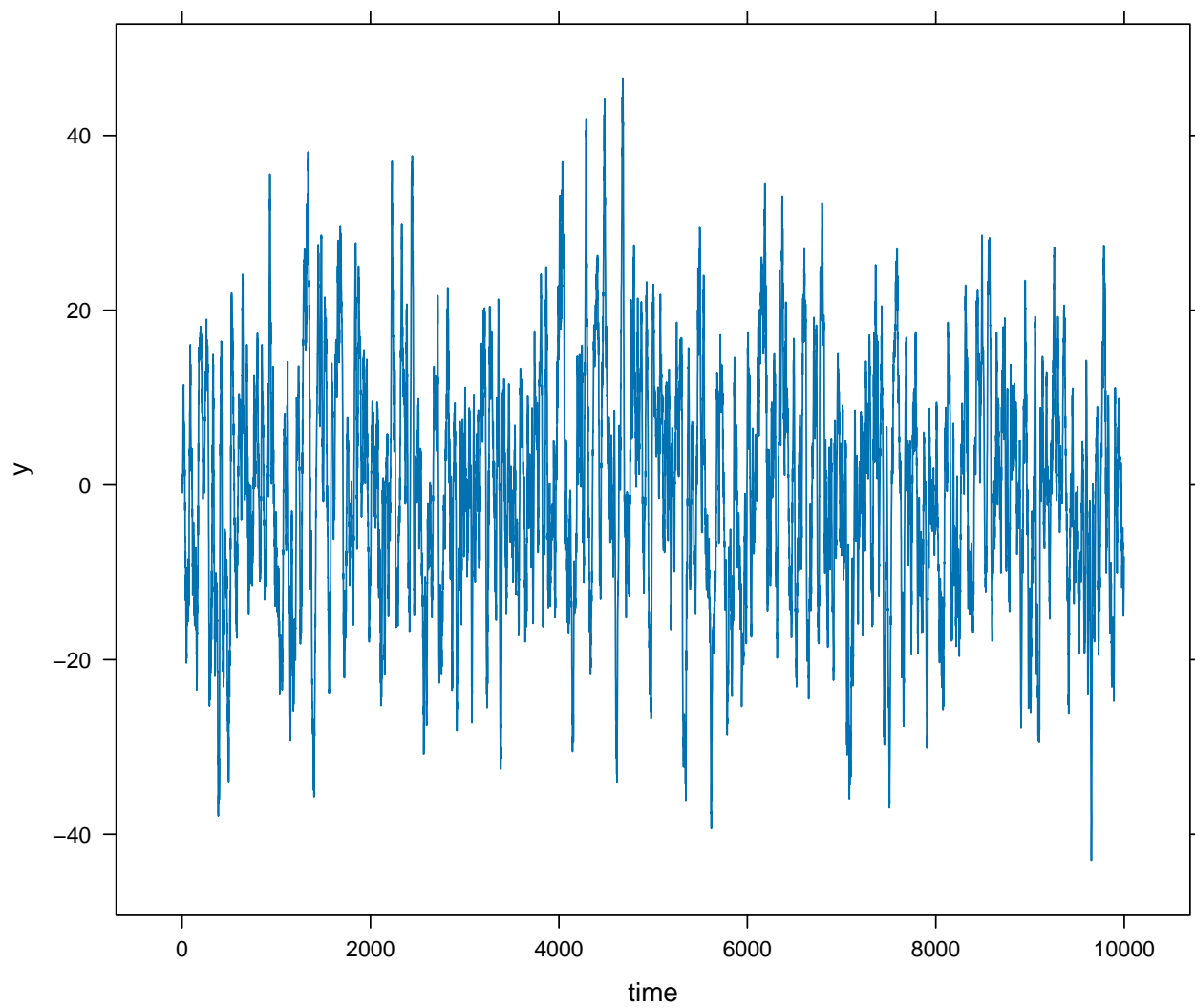
## Note: 'data' coerced to 'ts_data_frame'

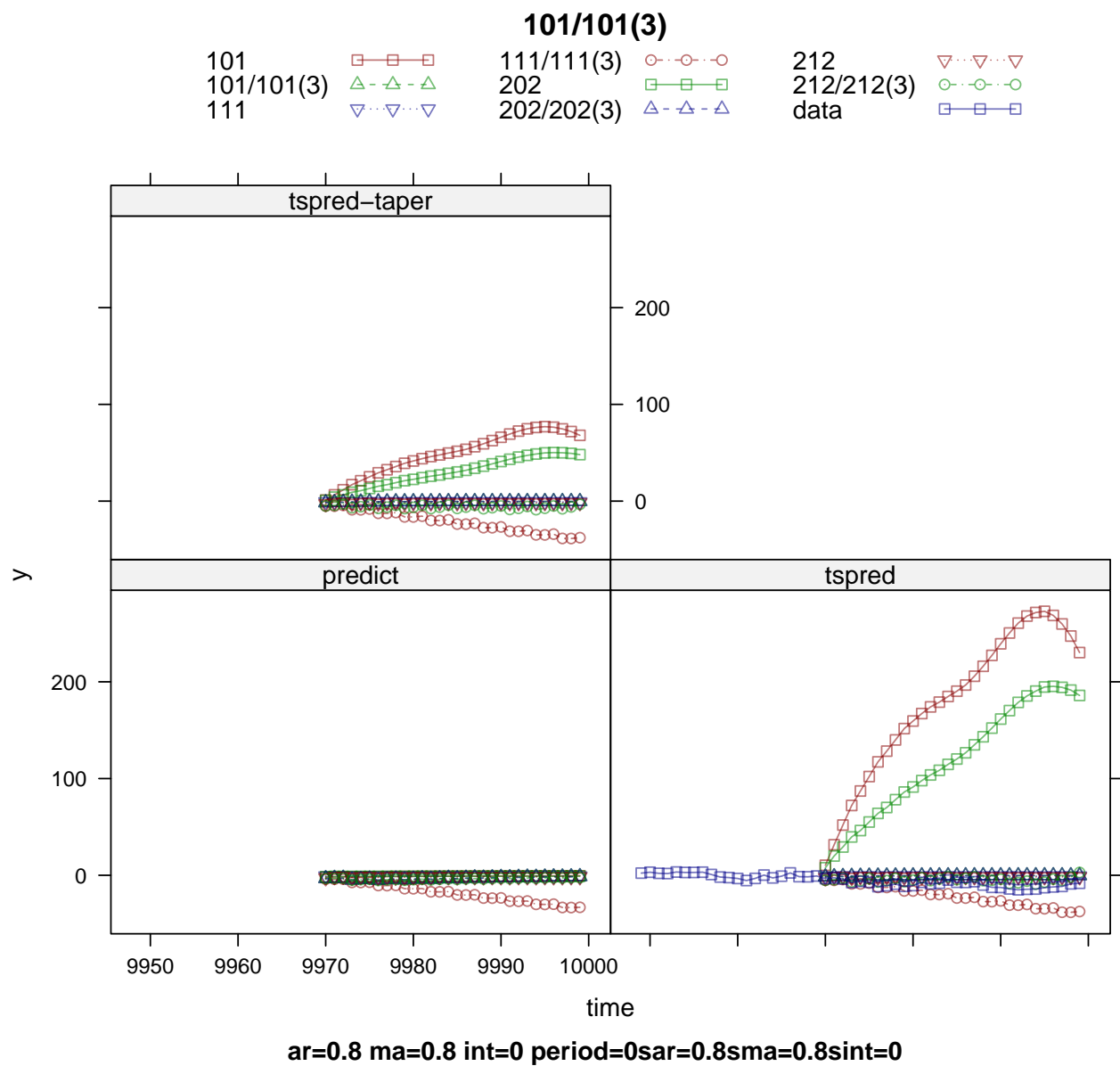
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained

##              model
## method      101 101/101(3) 111 111/111(3) 202 202/202(3) 212 212/212(3) data
## predict      30          30 30          30 30          30 30          30 0
## tspred       30          30 30          30 30          30 30          30 51
## tspred-taper 30          30 30          30 30          30 30          30 0
## Total        90          90 90          90 90          90 90          90 51

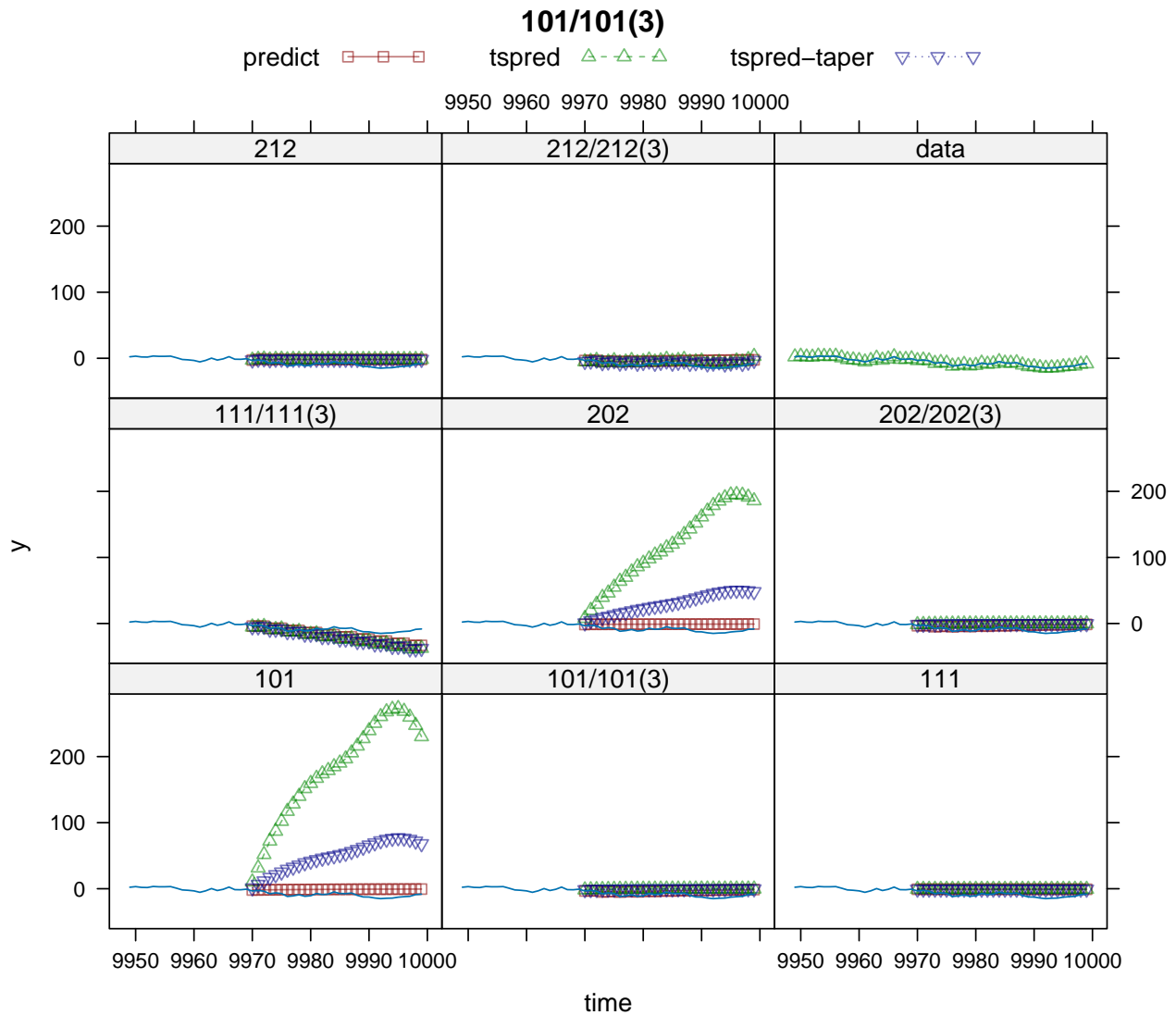
##              model
## method      Total
## predict      240
## tspred       291
## tspred-taper 240
## Total        771

```





[1] " "



ar=0.8 ma=0.8 int=0 period=0sar=0.8sma=0.8sint=0

```
## [1] "    "
## $`101`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -10.03952  -1.61081  -0.03039    1.61174    8.83381
##
## Estimates:
##              Estimate Std. Error  z value Pr(>|z|)
## ar1          0.9364001   0.0035050  267.161  <2e-16 ***
## ma1          0.9941750   0.0007581 1311.442  <2e-16 ***
## (Intercept) -0.3331352   0.7432159  -0.448   0.6540
## x            0.0021805   0.0012471   1.748   0.0804 .
## ---
```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 2.374
## Log-likelihood = -22840
## AIC = 45680
##
## $`202`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -7.81658 -1.47100 -0.01337  1.45428  9.79171
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1      0.788319   0.013477  58.494 <2e-16 ***
## ar2      0.190179   0.013175  14.435 <2e-16 ***
## ma1      0.695962   0.010082  69.027 <2e-16 ***
## ma2     -0.298787   0.010011 -29.845 <2e-16 ***
## (Intercept) -0.334244  1.382537  -0.242  0.8090
## x          0.002469   0.001488   1.659  0.0971 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 2.139
## Log-likelihood = -21800
## AIC = 43610
##
## $`111`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -10.030801 -1.789496  0.003127  1.812988  11.729483
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.014187   0.028256   0.502 0.615593
## ma1 -0.099528   0.026392  -3.771 0.000163 ***
## x   -0.005395   0.019531  -0.276 0.782386
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 2.675
## Log-likelihood = -24020
## AIC = 48050
##
## $`212`
##
## Call:

```

```

## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -5.46238 -0.98256 -0.01549  0.97561  6.20305
##
## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1 -0.9475041  0.0065091 -145.566 <2e-16 ***
## ar2 -0.7857207  0.0062458 -125.799 <2e-16 ***
## ma1  1.7536826  0.0064390  272.352 <2e-16 ***
## ma2  0.7677939  0.0064320  119.371 <2e-16 ***
## x    0.0003943  0.0006996   0.564  0.573
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.451
## Log-likelihood = -17910
## AIC = 35830
##
## $`101/101(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##      seasonal = list(order = c(1, 0, 1), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -3.47346 -0.66376 -0.01144  0.67410  3.96441
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1      0.8059088  0.0065318 123.383 <2e-16 ***
## ma1      0.8041513  0.0065872 122.078 <2e-16 ***
## sar1     0.7896331  0.0067878 116.330 <2e-16 ***
## sma1     0.8064140  0.0067132 120.124 <2e-16 ***
## (Intercept) -0.3618415  0.7843673  -0.461  0.645
## x          0.0004326  0.0007003   0.618  0.537
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.988
## Log-likelihood = -14070
## AIC = 28160
##
## $`202/202(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##      0, 2), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -3.47930 -0.66486 -0.01269  0.67089  3.95628

```

```

##
## Estimates:
##           Estimate Std. Error z value Pr(>|z|)
## ar1         0.4779041  0.8163988   0.585   0.558
## ar2         0.2680171  0.6568550   0.408   0.683
## ma1         1.1317316  0.8171402   1.385   0.166
## ma2         0.2609881  0.6597453   0.396   0.692
## sar1        1.2316175  1.0361016   1.189   0.235
## sar2       -0.3465912  0.8196633  -0.423   0.672
## sma1        0.3563456  1.0296360   0.346   0.729
## sma2       -0.3633668  0.8267541  -0.440   0.660
## (Intercept) -0.2532164  0.7989868  -0.317   0.751
## x           0.0004357  0.0006963   0.626   0.531
##
## Residual standard deviation: 0.9879
## Log-likelihood = -14070
## AIC = 28170
##
## $`111/111(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##           seasonal = list(order = c(1, 1, 1), period = 3))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -3.821357 -0.736966 -0.003199  0.732859  4.673943
##
## Estimates:
##           Estimate Std. Error z value Pr(>|z|)
## ar1   -0.1463450   0.0125811 -11.632 <2e-16 ***
## ma1    0.8581094   0.0070202 122.235 <2e-16 ***
## sar1  -0.2289347   0.0133333 -17.170 <2e-16 ***
## sma1   0.8296516   0.0076780 108.056 <2e-16 ***
## x      0.0005505   0.0006579   0.837   0.403
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.09
## Log-likelihood = -15040
## AIC = 30100
##
## $`212/212(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
## 1, 2), period = 3))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -3.52217 -0.68181 -0.02131  0.66143  3.94128
##
## Estimates:
##           Estimate Std. Error z value Pr(>|z|)

```

```

## ar1    0.8012421  0.0145917  54.911  <2e-16 ***
## ar2    0.0080640  0.0171473   0.470   0.638
## ma1   -0.1910588  0.0105628 -18.088  <2e-16 ***
## ma2   -0.8089411  0.0105555 -76.637  <2e-16 ***
## sar1   0.7743655  0.0165035  46.921  <2e-16 ***
## sar2   0.0135186  0.0142733   0.947   0.344
## sma1  -0.1891579  0.0081473 -23.217  <2e-16 ***
## sma2  -0.8081705  0.0081587 -99.056  <2e-16 ***
## x      0.0004432  0.0006935   0.639   0.523
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.9884
## Log-likelihood = -14080
## AIC = 28180
##
## [1] 1.25+0i
## [1] 1.25
## [1] Inf

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Estimates:
##          ar1          ma1 (Intercept)          x
##    0.935747    0.994990   -0.347881   -0.001085

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Estimates:
##          ar1          ar2          ma1          ma2 (Intercept)          x
##    0.783246    0.195984    0.684299   -0.311286   -0.351217   -0.001106

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Estimates:
##          ar1          ma1          x
##    0.004853  -0.111846  -0.029334

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##
## Estimates:
##          ar1          ar2          ma1          ma2          x

```

```

## -0.9518600 -0.7946147 1.7369815 0.7502696 -0.0005943
## Note: 'data' coerced to 'ts_data_frame'
## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##     seasonal = list(order = c(1, 0, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1 (Intercept)      x
## 0.7973974 0.8023156 0.7924429 0.8000443 -0.3118965 -0.0006539
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##     0, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2
## 0.5951047 0.1713949 1.0060765 0.1536412 0.4893888 0.2475008
##      sma1      sma2 (Intercept)      x
## 1.0708299 0.2076375 -0.3621781 -0.0006383
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##     seasonal = list(order = c(1, 1, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1      x
## -0.160312 0.865358 -0.255035 0.829494 -0.000595
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
##     1, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2      sma1      sma2
## 0.785435 0.019855 -0.184450 -0.815520 0.748735 0.042424 -0.189410 -0.810016
##      x
## -0.000636
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

```

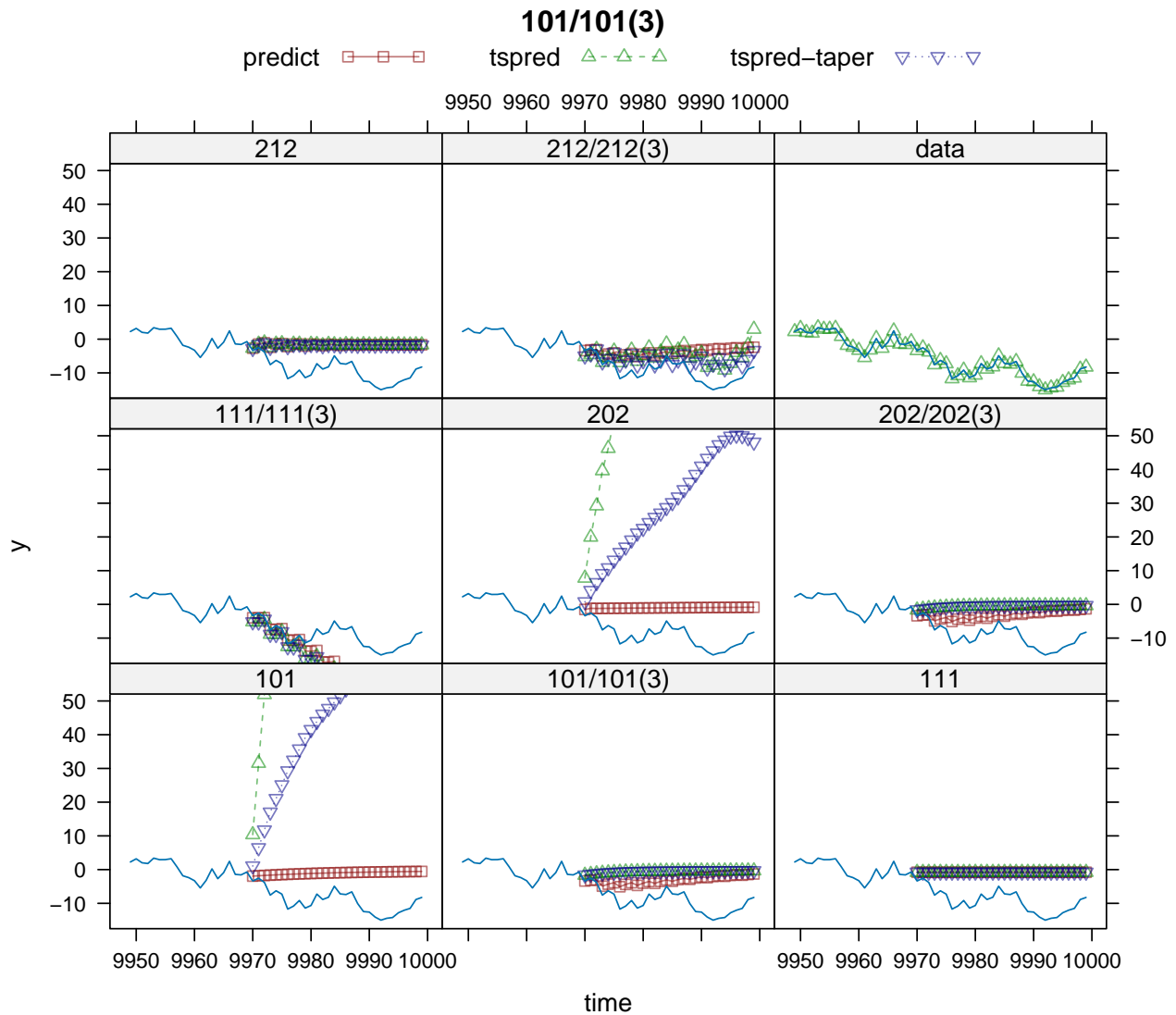
```

## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1

## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

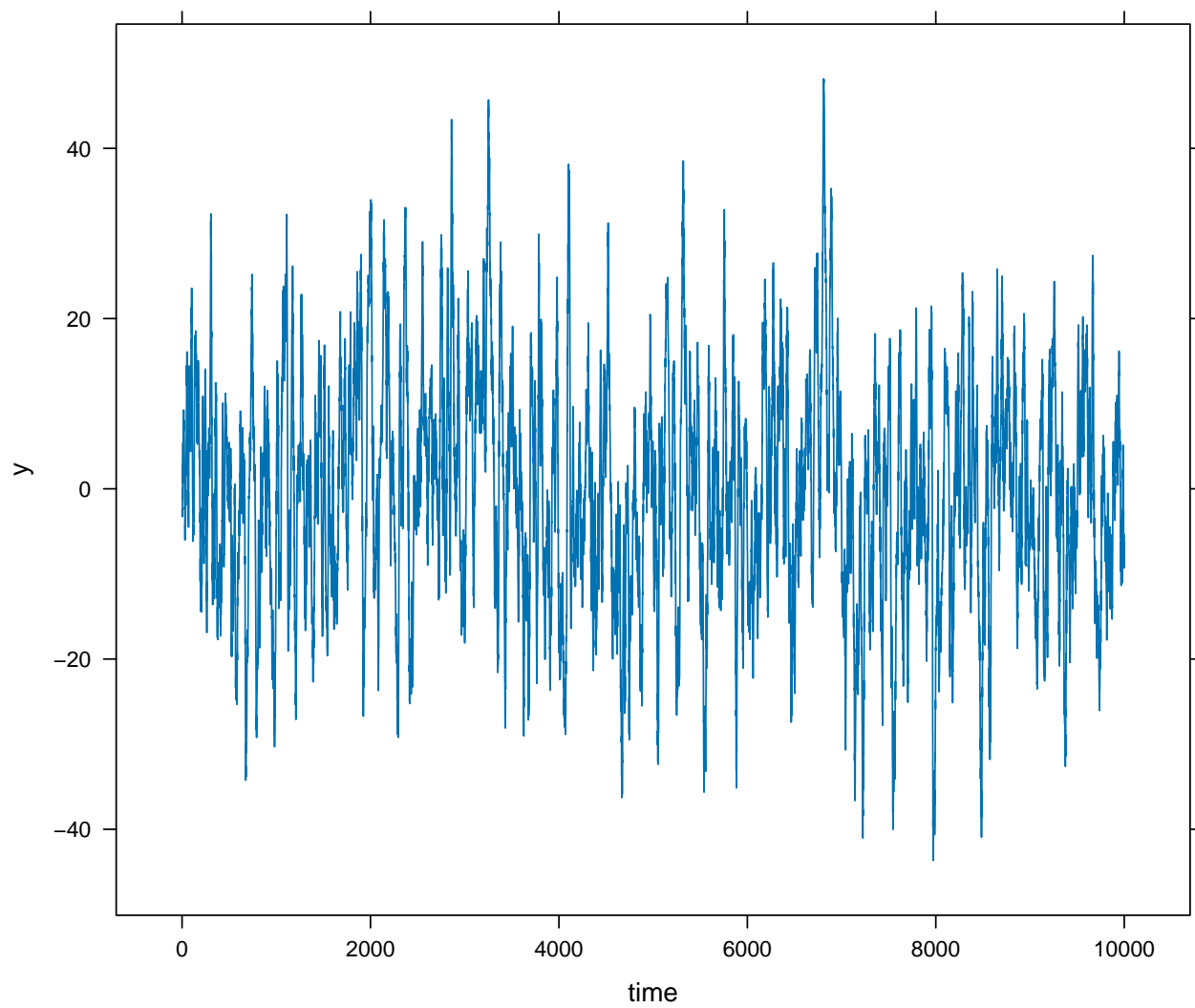
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained

```

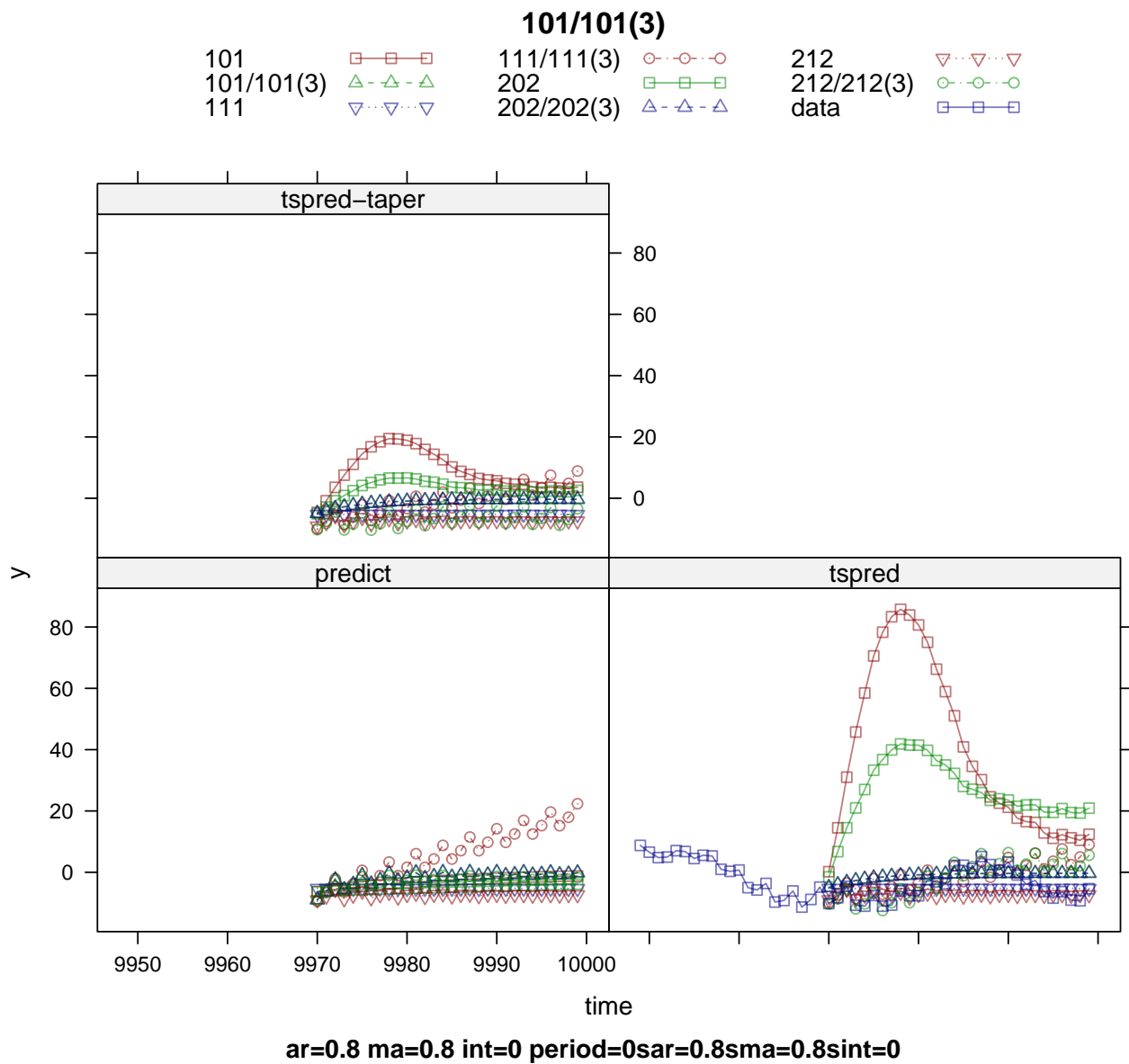



ar=0.8 ma=0.8 int=0 period=0sar=0.8sma=0.8sint=0

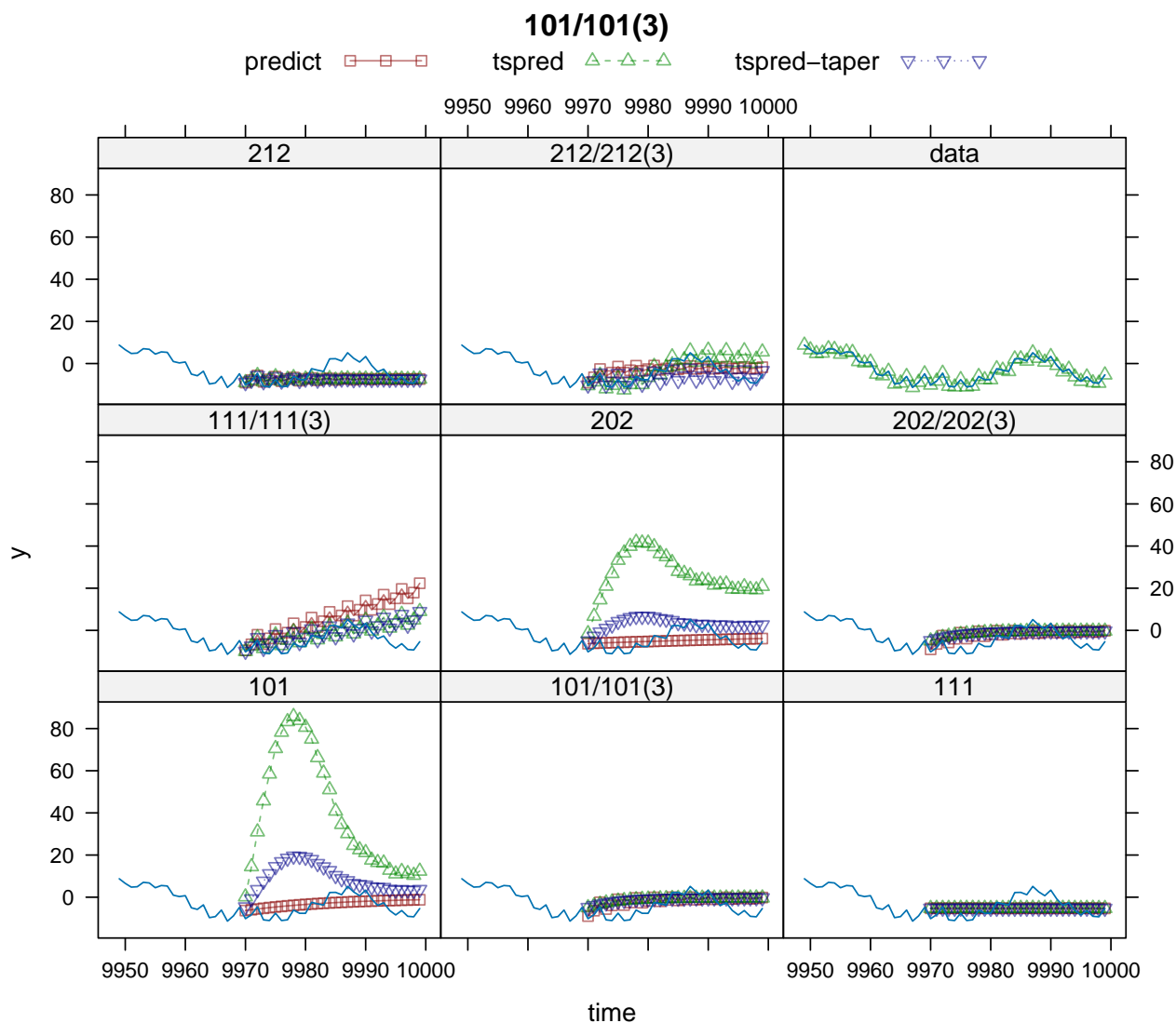
```
##          model
## method   101 101/101(3) 111 111/111(3) 202 202/202(3) 212 212/212(3) data
## predict      30          30 30          30 30          30 30          30 0
## tspred       30          30 30          30 30          30 30          30 51
## tspred-taper 30          30 30          30 30          30 30          30 0
## Total        90          90 90          90 90          90 90          90 51
##          model
## method   Total
## predict      240
## tspred       291
## tspred-taper 240
## Total        771
```



ar=0.8 ma=0.8 int=0 period=0sar=0.8sma=0.8sint=0



[1] " "



ar=0.8 ma=0.8 int=0 period=0sar=0.8sma=0.8sint=0

```
## [1] "      "
## $`101`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -7.88639 -1.63881  0.01219  1.63561  8.96997
##
## Estimates:
##              Estimate Std. Error  z value Pr(>|z|)
## ar1          0.9357475   0.0035274  265.276  <2e-16 ***
## ma1          0.9949902   0.0007251 1372.165  <2e-16 ***
## (Intercept) -0.3478810   0.7415377  -0.469    0.639
## x           -0.0010853   0.0010675  -1.017    0.309
## ---
```

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 2.393
## Log-likelihood = -22920
## AIC = 45840
##
## $`202`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -7.542256 -1.443696  0.008315  1.424609  7.992065
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1      0.783246   0.013302  58.882 <2e-16 ***
## ar2      0.195984   0.013013  15.060 <2e-16 ***
## ma1      0.684299   0.009880  69.261 <2e-16 ***
## ma2     -0.311286   0.009827 -31.677 <2e-16 ***
## (Intercept) -0.351217  1.406051  -0.250  0.803
## x        -0.001106   0.001216  -0.909  0.363
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 2.14
## Log-likelihood = -21800
## AIC = 43620
##
## $`111`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -9.4385 -1.7683  0.0397  1.8004 10.1533
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.004853   0.025554   0.190   0.849
## ma1 -0.111846   0.023504  -4.759 1.95e-06 ***
## x   -0.029334   0.019721  -1.487   0.137
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 2.674
## Log-likelihood = -24020
## AIC = 48050
##
## $`212`
##
## Call:

```

```

## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -5.595608 -0.981131  0.003514  0.994077  5.260522
##
## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1 -0.9518600  0.0064102 -148.492 <2e-16 ***
## ar2 -0.7946147  0.0061308 -129.610 <2e-16 ***
## ma1  1.7369815  0.0066214  262.330 <2e-16 ***
## ma2  0.7502696  0.0066093  113.516 <2e-16 ***
## x   -0.0005943  0.0007046  -0.843  0.399
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.443
## Log-likelihood = -17860
## AIC = 35730
##
## $`101/101(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##      seasonal = list(order = c(1, 0, 1), period = 3))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -3.882111 -0.684962  0.005126  0.679357  3.779648
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1      0.7973974  0.0067257 118.560 <2e-16 ***
## ma1      0.8023156  0.0066581 120.501 <2e-16 ***
## sar1     0.7924429  0.0068091 116.380 <2e-16 ***
## sma1     0.8000443  0.0067479 118.561 <2e-16 ***
## (Intercept) -0.3118965  0.7675141  -0.406  0.684
## x        -0.0006539  0.0007407  -0.883  0.377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.001
## Log-likelihood = -14200
## AIC = 28420
##
## $`202/202(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##      0, 2), period = 3))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -3.817773 -0.683093  0.004311  0.677257  3.773487

```

```

##
## Estimates:
##           Estimate Std. Error z value Pr(>|z|)
## ar1         0.5951047  0.6745514   0.882  0.3777
## ar2         0.1713949  0.5363281   0.320  0.7493
## ma1         1.0060765  0.6761076   1.488  0.1367
## ma2         0.1536412  0.5502873   0.279  0.7801
## sar1        0.4893888  0.2552426   1.917  0.0552 .
## sar2        0.2475008  0.2011558   1.230  0.2185
## sma1        1.0708299  0.2587898   4.138 3.51e-05 ***
## sma2        0.2076375  0.2069177   1.003  0.3156
## (Intercept) -0.3621781  0.7953726  -0.455  0.6489
## x           -0.0006383  0.0007181  -0.889  0.3741
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1
## Log-likelihood = -14200
## AIC = 28420
##
## $`111/111(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##           seasonal = list(order = c(1, 1, 1), period = 3))
##
## Residuals:
##      Min       1st Q   Median       3rd Q       Max
## -4.131470 -0.742877  0.009803  0.723992  4.115726
##
## Estimates:
##           Estimate Std. Error z value Pr(>|z|)
## ar1    -0.1603125  0.0125583 -12.765 <2e-16 ***
## ma1     0.8653580  0.0069279 124.909 <2e-16 ***
## sar1   -0.2550353  0.0133944 -19.040 <2e-16 ***
## sma1    0.8294941  0.0075572 109.761 <2e-16 ***
## x      -0.0005950  0.0006772  -0.879   0.38
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.102
## Log-likelihood = -15150
## AIC = 30320
##
## $`212/212(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
## 1, 2), period = 3))
##
## Residuals:
##      Min       1st Q   Median       3rd Q       Max
## -3.790022 -0.691545  0.002637  0.671610  3.753067
##

```

```

## Estimates:
##      Estimate Std. Error z value Pr(>|z|
## ar1   0.7854346  0.0145803  53.870 < 2e-16 ***
## ar2   0.0198553  0.0171569   1.157 0.24716
## ma1  -0.1844502  0.0106552 -17.311 < 2e-16 ***
## ma2  -0.8155196  0.0106008 -76.930 < 2e-16 ***
## sar1  0.7487348  0.0166535  44.960 < 2e-16 ***
## sar2  0.0424244  0.0144006   2.946 0.00322 **
## sma1 -0.1894099  0.0082302 -23.014 < 2e-16 ***
## sma2 -0.8100164  0.0081536 -99.345 < 2e-16 ***
## x    -0.0006360  0.0007199  -0.883 0.37701
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1
## Log-likelihood = -14200
## AIC = 28430
##
## [1] 1.25+0i
## [1] 1.25
## [1] Inf

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Estimates:
##      ar1      ma1 (Intercept)      x
##  0.9287032  0.9937170 -1.1504982 -0.0004603

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Estimates:
##      ar1      ar2      ma1      ma2 (Intercept)      x
##  0.783316  0.193412  0.684583 -0.309687 -1.151327 -0.001014

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Estimates:
##      ar1      ma1      x
##  0.006194 -0.111065  0.003939

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##

```



```

## Estimates:
##      ar1      ar2      ma1      ma2      x
## -0.950296 -0.795742  1.744931  0.761327 -0.000018

## Note: 'data' coerced to 'ts_data_frame'

## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1

##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##      seasonal = list(order = c(1, 0, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1 (Intercept)      x
##  7.998e-01  7.945e-01  7.948e-01  8.027e-01 -1.105e+00 -9.856e-05

## Note: 'data' coerced to 'ts_data_frame'

## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##      0, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2
##  1.158e+00 -2.968e-01  4.410e-01 -2.668e-01  2.171e-01  4.640e-01
##      sma1      sma2 (Intercept)      x
##  1.403e+00  4.866e-01 -1.132e+00 -7.617e-05

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##      seasonal = list(order = c(1, 1, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1      x
## -0.1386779  0.8446956 -0.2058290  0.8194297 -0.0001784

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
##      1, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2      sma1
##  8.181e-01 -2.410e-02 -2.189e-01 -7.811e-01  8.161e-01 -1.924e-02 -1.979e-01
##      sma2      x
## -8.010e-01 -8.091e-05

## Note: 'data' coerced to 'ts_data_frame'

```

```

## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1

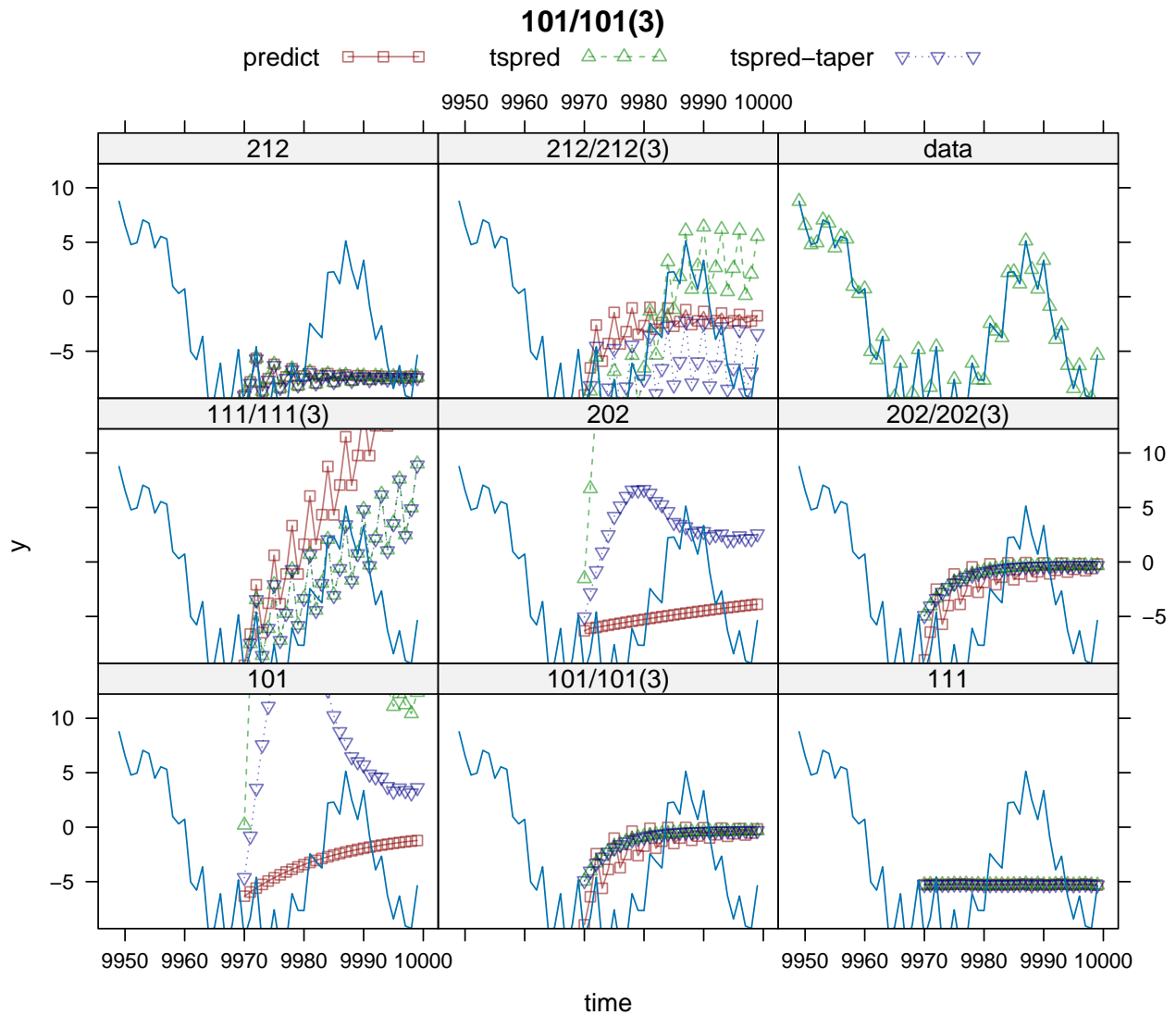
## Note: 'data' coerced to 'ts_data_frame'

## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1

## Note: 'data' coerced to 'ts_data_frame'
## Note: 'data' coerced to 'ts_data_frame'

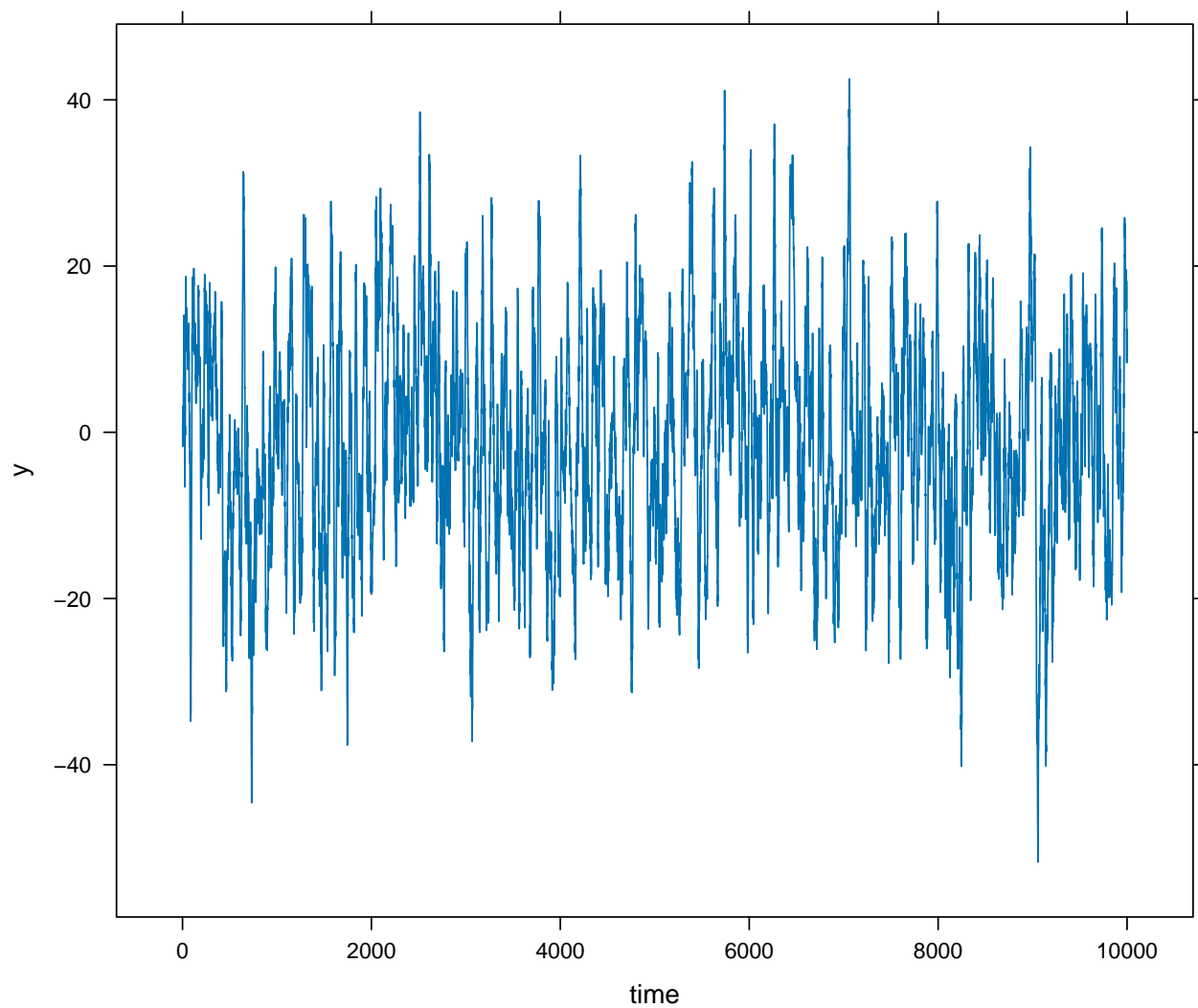
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained
## Warning in arma2pi(ar = ar, ma = ma, ar.seasonal = sar, ma.seasonal = sma, :
## all 100 pi weights retained

```

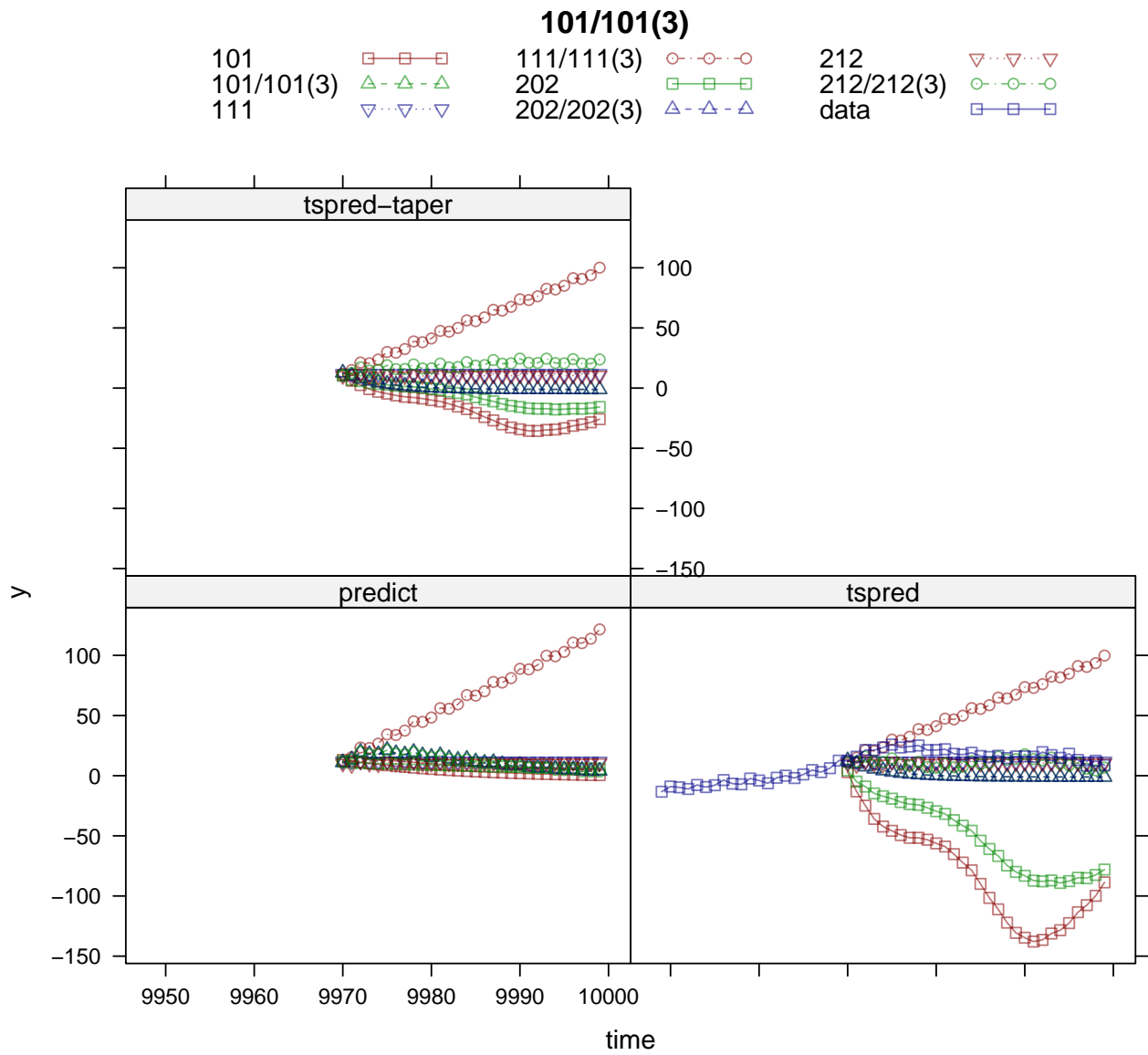


ar=0.8 ma=0.8 int=0 period=0sar=0.8sma=0.8sint=0

```
##          model
## method    101 101/101(3) 111 111/111(3) 202 202/202(3) 212 212/212(3) data
## predict      30      30  30      30  30      30  30      30  0
## tspred       30      30  30      30  30      30  30      30  51
## tspred-taper 30      30  30      30  30      30  30      30  0
## Total        90      90  90      90  90      90  90      90  51
##          model
## method      Total
## predict      240
## tspred       291
## tspred-taper 240
## Total        771
```

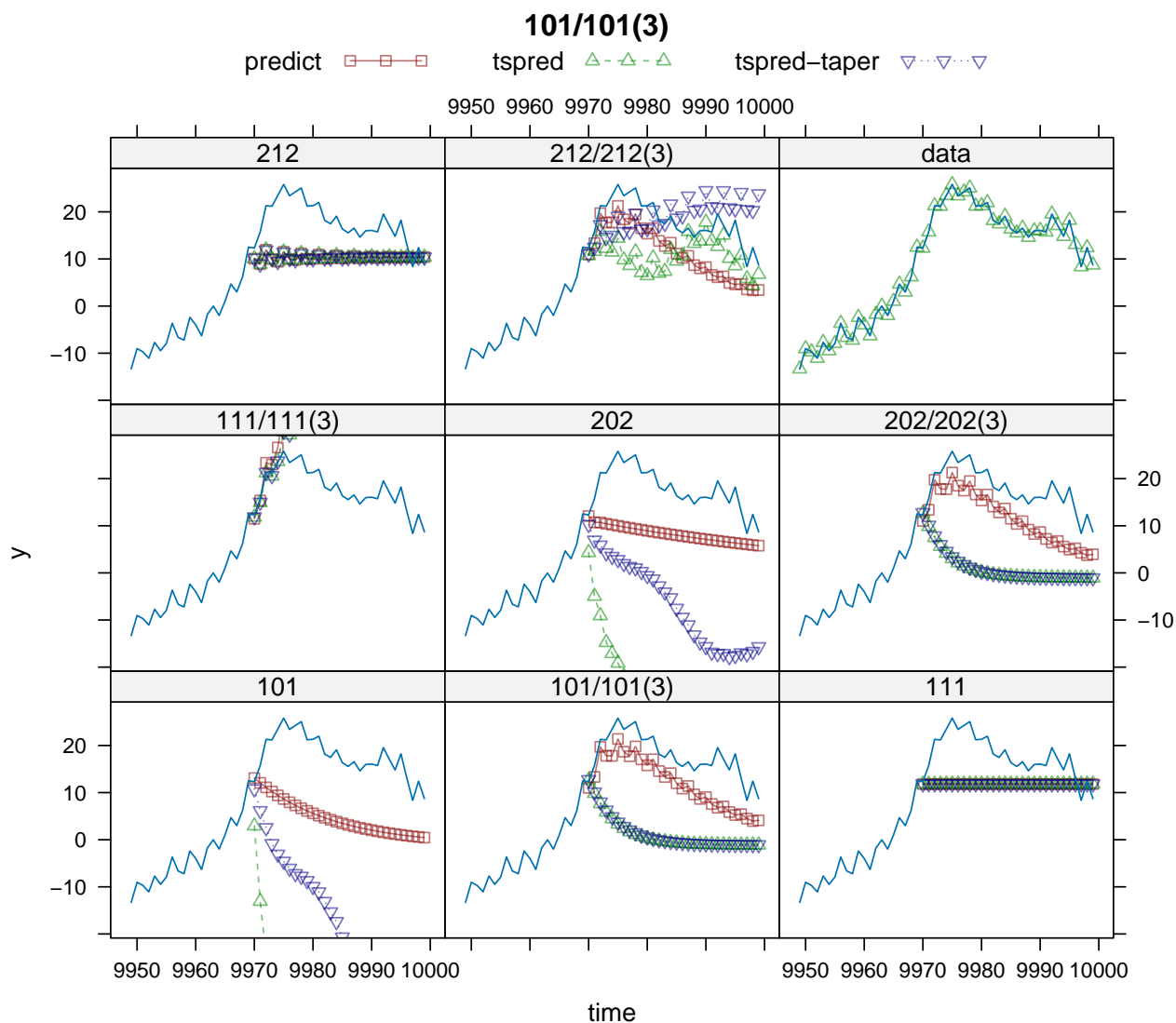


ar=0.8 ma=0.8 int=0 period=0sar=0.8sma=0.8sint=0



ar=0.8 ma=0.8 int=0 period=0sar=0.8sma=0.8sint=0

[1] " "



ar=0.8 ma=0.8 int=0 period=0sar=0.8sma=0.8sint=0

```
## $`101`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Residuals:
##      Min      1st Q  Median      3rd Q       Max
## -9.822  -1.623  -0.010   1.659   8.400
##
## Estimates:
##              Estimate Std. Error  z value Pr(>|z|)
## ar1          0.9287032   0.0037020  250.864  <2e-16 ***
## ma1          0.9937170   0.0007692 1291.946  <2e-16 ***
## (Intercept) -1.1504982   0.6809064  -1.690   0.0911 .
## x           -0.0004603   0.0013697  -0.336   0.7368
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```

## Residual standard deviation: 2.438
## Log-likelihood = -23100
## AIC = 46220
##
## $`202`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -8.46746 -1.46317 -0.01899  1.44261  7.96335
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1      0.783316   0.013365  58.611 <2e-16 ***
## ar2      0.193412   0.013041  14.832 <2e-16 ***
## ma1      0.684583   0.009944  68.840 <2e-16 ***
## ma2     -0.309687   0.009870 -31.377 <2e-16 ***
## (Intercept) -1.151327  1.285485  -0.896  0.370
## x         -0.001014   0.001609  -0.630  0.529
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 2.187
## Log-likelihood = -22010
## AIC = 44040
##
## $`111`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -10.77029 -1.78796 -0.02036  1.80856  10.92658
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.006194   0.025761   0.240   0.810
## ma1 -0.111065   0.023734  -4.680 2.87e-06 ***
## x    0.003939   0.020604   0.191   0.848
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 2.734
## Log-likelihood = -24240
## AIC = 48490
##
## $`212`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##

```



```

## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -5.7140 -0.9741 -0.0149  0.9725  5.6972
##
## Estimates:
##      Estimate Std. Error  z value Pr(>|z|)
## ar1 -0.9502962  0.0063831 -148.877 <2e-16 ***
## ar2 -0.7957418  0.0061237 -129.944 <2e-16 ***
## ma1  1.7449307  0.0065405  266.787 <2e-16 ***
## ma2  0.7613270  0.0065277  116.630 <2e-16 ***
## x   -0.0000180  0.0007795   -0.023  0.982
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.468
## Log-likelihood = -18030
## AIC = 36080
##
## $`101/101(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##      seasonal = list(order = c(1, 0, 1), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -3.595803 -0.687831  0.004072  0.660395  4.045446
##
## Estimates:
##      Estimate Std. Error z value Pr(>|z|)
## ar1      7.998e-01  6.528e-03 122.521 <2e-16 ***
## ma1      7.945e-01  6.613e-03 120.139 <2e-16 ***
## sar1      7.948e-01  6.601e-03 120.412 <2e-16 ***
## sma1      8.027e-01  6.446e-03 124.532 <2e-16 ***
## (Intercept) -1.105e+00  7.857e-01  -1.407  0.159
## x          -9.856e-05  7.595e-04  -0.130  0.897
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1
## Log-likelihood = -14200
## AIC = 28420
##
## $`202/202(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,
##      0, 2), period = 3))
##
## Residuals:
##      Min      1st Q      Median      3rd Q      Max
## -3.603128 -0.686844  0.002189  0.662920  4.049908
##
## Estimates:

```

```

##           Estimate Std. Error z value Pr(>|z|
## ar1      1.158e+00  3.271e-01   3.541 0.000399 ***
## ar2     -2.968e-01  2.580e-01  -1.150 0.249947
## ma1      4.410e-01  3.277e-01   1.346 0.178372
## ma2     -2.668e-01  2.636e-01  -1.012 0.311367
## sar1      2.171e-01  2.532e-01   0.857 0.391321
## sar2      4.640e-01  2.038e-01   2.277 0.022793 *
## sma1      1.403e+00  2.496e-01   5.621 1.89e-08 ***
## sma2      4.866e-01  2.005e-01   2.427 0.015219 *
## (Intercept) -1.132e+00  7.667e-01  -1.476 0.139917
## x         -7.617e-05  7.723e-04  -0.099 0.921434
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1
## Log-likelihood = -14200
## AIC = 28420
##
## $`111/111(3)`
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 1, 1),
##           seasonal = list(order = c(1, 1, 1), period = 3))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -4.118109 -0.760307  0.006684  0.744838  4.299877
##
## Estimates:
##           Estimate Std. Error z value Pr(>|z|
## ar1   -0.1386779   0.0128359 -10.804  <2e-16 ***
## ma1    0.8446956   0.0073945 114.234  <2e-16 ***
## sar1  -0.2058290   0.0132812 -15.498  <2e-16 ***
## sma1   0.8194297   0.0075004 109.252  <2e-16 ***
## x     -0.0001784   0.0007306  -0.244   0.807
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1.106
## Log-likelihood = -15190
## AIC = 30390
##
## $`212/212(3)`
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2), seasonal = list(order = c(2,
## 1, 2), period = 3))
##
## Residuals:
##      Min       1st Q       Median       3rd Q       Max
## -3.6133638 -0.6853311  0.0006308  0.6657272  4.0963432
##
## Estimates:
##           Estimate Std. Error z value Pr(>|z|

```

```
## ar1 8.181e-01 1.562e-02 52.383 <2e-16 ***
## ar2 -2.410e-02 1.870e-02 -1.289 0.197
## ma1 -2.189e-01 1.209e-02 -18.101 <2e-16 ***
## ma2 -7.811e-01 1.207e-02 -64.702 <2e-16 ***
## sar1 8.161e-01 1.721e-02 47.414 <2e-16 ***
## sar2 -1.924e-02 1.474e-02 -1.305 0.192
## sma1 -1.979e-01 7.796e-03 -25.381 <2e-16 ***
## sma2 -8.010e-01 7.769e-03 -103.111 <2e-16 ***
## x -8.091e-05 7.695e-04 -0.105 0.916
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 1
## Log-likelihood = -14200
## AIC = 28430

##          [,1]          [,2]          [,3]
## 101      summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 202      summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 111      summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 212      summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 101/101(3) summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 202/202(3) summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 111/111(3) summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
## 212/212(3) summary.ARIMA,8 summary.ARIMA,8 summary.ARIMA,8
```

AR -1.95,-.97, MA .8, PERIOD 3 SAR -1.95,-.97 SMA .8

```
replicate(3,
  predplot(
    ar=c(1.95,-.97), ma =.8,
    period = 3, sar=c(1.95,-.97), sma =.8,
    main ='201/201(3)') )
```

```
## [1] 1.005155+0.1434991i 1.005155-0.1434991i
## [1] 1.015346 1.015346
## [1] 44.30864 -44.30864

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Estimates:
##      ar1      ma1 (Intercept)      x
## 0.99852 1.00000 218.92844 0.00165

## Note: 'data' coerced to 'ts_data_frame'

##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2))
##
## Estimates:
##      ar1      ar2      ma1      ma2 (Intercept)      x
```

```

##    1.607e+00  -6.099e-01   1.991e+00   9.927e-01   1.812e+02   6.250e-05
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 1, 1))
##
## Estimates:
##      ar1      ma1      x
## 0.88910 1.00000 0.00161
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 1, 2))
##
## Estimates:
##      ar1      ar2      ma1      ma2      x
## 0.683808 0.298238 0.561085 -0.438915 0.001629
## Note: 'data' coerced to 'ts_data_frame'
## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1
##
## Call:
## cv::Arima(formula = y ~ 1 + x, data = dd, order = c(1, 0, 1),
##      seasonal = list(order = c(1, 0, 1), period = 3))
##
## Estimates:
##      ar1      ma1      sar1      sma1 (Intercept)      x
## 9.962e-01 9.653e-01 9.648e-01 9.930e-01 2.283e+02 4.288e-04
## Note: 'data' coerced to 'ts_data_frame'
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in log(s2): NaNs produced
## Warning in stats::arima(y, order = order, seasonal = seasonal, xreg = x, :
## possible convergence problem: optim gave code = 1
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(2, 0, 2), seasonal = list(order = c(2,

```

```
##      0, 2), period = 3))
##
## Estimates:
##      ar1      ar2      ma1      ma2      sar1      sar2
##  3.617e-01  6.087e-01  1.747e+00  7.955e-01  1.874e+00 -9.251e-01
##      sma1      sma2 (Intercept)      x
##  1.498e+00  5.937e-01  2.447e+02 -7.287e-05
## Note: 'data' coerced to 'ts_data_frame'
## Error in optim(init[mask], armafn, method = optim.method, hessian = TRUE, : non-finite finite-differ
```

AR -1.95,-.97, MA .8, INT: 1, PERIOD 3 SAR -1.95,-.97 SMA .8

```
replicate(3,
  predplot(
    ar=c(1.95,-.97), int = 1, ma =.8,
    period = 3, sar=c(1.95,-.97), sma =.8,
    main='211/201(3)' ))
```

```
## [1] 1.005155+0.1434991i 1.005155-0.1434991i
## [1] 1.015346 1.015346
## [1] 44.30864 -44.30864
## Note: 'data' coerced to 'ts_data_frame'
## Error in stats::arima(y, order = order, seasonal = seasonal, xreg = x, : non-stationary AR part from
```

AR -1.95,-.97, MA .8, INT: 1, PERIOD 3 SAR -1.95,-.97 SMA .8

```
replicate(3,predplot(
  ar=c(1.95,-.97), ma =.8, period = 3,
  sar=c(1.95,-.97), sma =.8, sint = 1,
  main = '201/211(3)'))
```

```
## [1] 1.005155+0.1434991i 1.005155-0.1434991i
## [1] 1.015346 1.015346
## [1] 44.30864 -44.30864
## Note: 'data' coerced to 'ts_data_frame'
##
## Call:
## cv::Arima(formula = y ~ x, data = dd, order = c(1, 0, 1))
##
## Estimates:
##      ar1      ma1 (Intercept)      x
##  9.997e-01  1.000e+00 -1.811e+05  2.203e-03
## Note: 'data' coerced to 'ts_data_frame'
## Error in stats::arima(y, order = order, seasonal = seasonal, xreg = x, : non-stationary AR part from
knitr::knit_exit()
```