# bsts_problems

### First error

The olddata parameter don´t combine with olddata.tiemstamp parameter, and don´t change enough the results.

### Fit model

In this step just fit a bsts model that include Local linear trend and seasonal components.

The parameters have been adjusted as shown bellow.

### Predictions

Visualization of the prediction output without applying the olddata parameter.

*Error* obtained when trying to include the olddata and olddata.timestamps parameters. Also, include de validation of that errors it is not real, number of observations and lenght of timestamp are equal.

```
pred_wolddata_ts <- try(predict(fit,
                                horizon = 100,
                                burn = burnin,
                                quantiles = c(.05, .95),
                                olddata = claims[50:80],
                                olddata.timestamps = index(claims[50:80]),
                                seed = 2408))

print(paste0("Difference between length of data and timestamp: ",length(index(claims[50:80])) - length(
```

```
## [1] "Difference between length of data and timestamp: 0"
```

```
pred_wolddata_ts
```

```
## [1] "Error : number.of.observations == length(timestamps) is not TRUE\n"
## attr(,"class")
## [1] "try-error"
## attr(,"condition")
## <simpleError: number.of.observations == length(timestamps) is not TRUE>
```

Second experiment, in this case removing timestamp, just olddata parameter has been defined using the 50:80 observations from training data.

```
pred_wolddata <- try(predict(fit,
                                horizon = 100,
                                burn = burnin,
                                quantiles = c(.05, .95),
                                olddata = claims[50:80],
                                seed = 2408))

length(index(claims[50:80])) - length(claims[50:80])
```

```
## [1] 0
```

Third experiment, in this case removing timestamp, just olddata parameter has been defined using the 20:30 observations from training data.

It can be seen, that there are low difference between results using or not the oldata parameter. However, there are any diference between select differents samples of training data as olddata.

```
print(paste0("Acumulated diference between without olddata prediction and training[50:80] data:", sum(pr
```

```
## [1] "Acumulated diference between without olddata prediction and training[50:80] data:17.89225168721
```

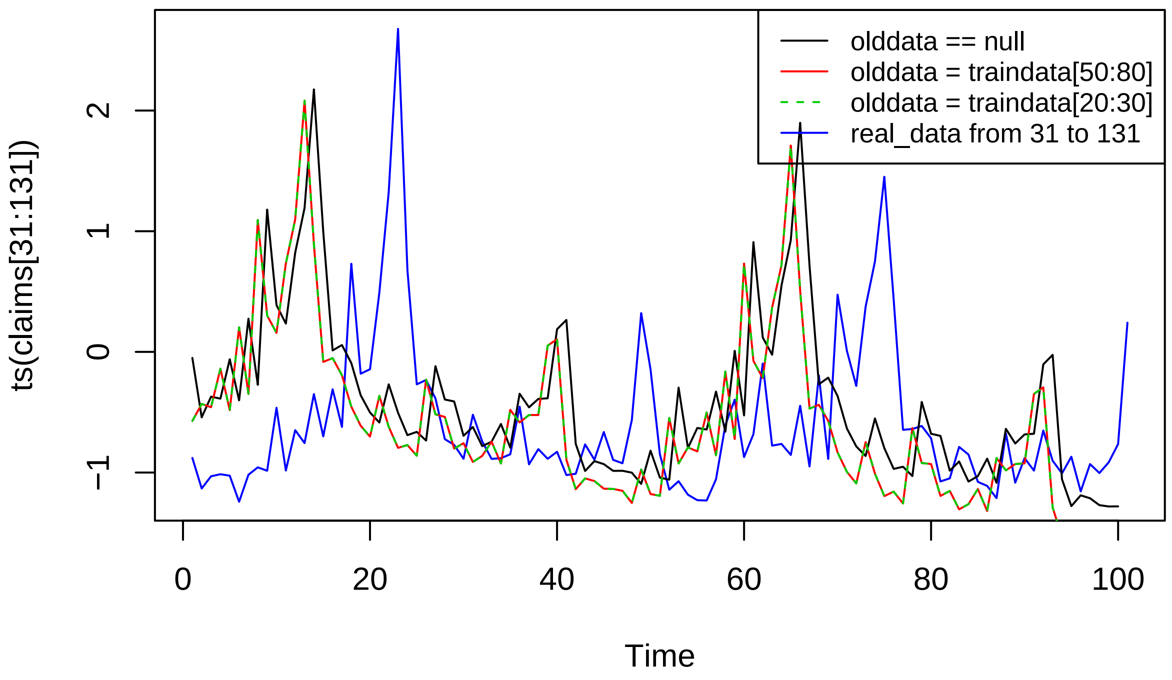
```
print(paste0("Acumulated diference between without olddata prediction and training[20:30] data:",sum(pre
```

```
## [1] "Acumulated diference between without olddata prediction and training[20:30] data:17.89225168721
```

```
print(paste0("Acumulated diference between training[20:30] and training[50:80] data:",sum(pred_wolddata
```

```
## [1] "Acumulated diference between training[20:30] and training[50:80] data:0"
```

```
plot(ts(claims[31:131]), col = 4)
lines(ts(pred_woolddata$mean), col = 1)
lines(ts(pred_wolddata$mean), col = 2)
lines(ts(pred_wolddata2$mean), col = 3, lty = 2)


legend("topright", legend=c("olddata == null", "olddata = traindata[50:80]", "olddata = traindata[20:30]
       col=c(1,2,3,4), lty = c(1,1,2,1), cex=0.8)
```



#### Second problem

It is not possible combine newdata parameter for regression and add the olddata parameter to define the initial situation and values to generate a prediction.

When we combine olddata with new data to get a prediction, the function shows the next error:

*Error*: number.of.time.points > 0 is not TRUE In addition: Warning message: In is.na(observed.data$response) : is.na() applied to non-(list or vector) of type 'NULL'

We have debugged the code and it seems like the predict function removes the response variable on the .FormatObservedDataForPredictions step. Looking into this function, the function .ExtractResponse doesn´t get the response variable name from the formula parameter.

### Fit with lags

After show some problems with the olddata parameter, let's explain the second issue.

We tried use newdata as a alternative to olddata adding some lags of the objective variable, that contain information from past. This alternative solve in some way the forecast problem for diferents timestamps but return and error in combination with olddata parameter.

#### Train model

```r
model_components <- list()
summary(model_components <- AddLocalLinearTrend(model_components,
                                                 y = claims))
```

```
##      Length Class            Mode
## [1,] 6      LocalLinearTrend list
```

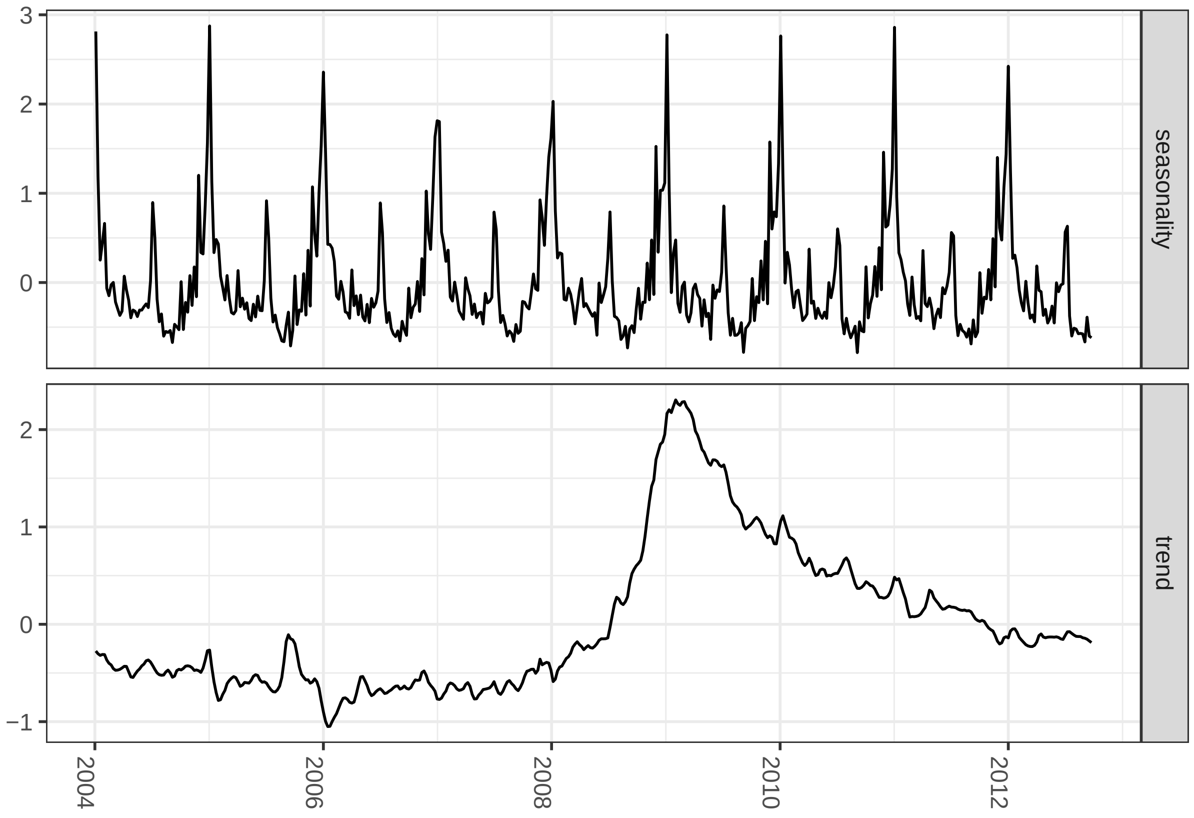
```r
summary(model_components <- AddSeasonal(model_components,
                                        y = claims,
                                        nseasons  = 52))
```

```
##      Length Class            Mode
## [1,] 6      LocalLinearTrend list
## [2,] 6      Seasonal         list
```

```r
fit_reg <- bsts(data = claims_dt[1:400,],
            formula = x ~.,
            state.specification = model_components,
            niter = 2000)
```

```
## =-=-=-=-= Iteration 0 Thu Mar 21 19:21:21 2019
##  =-=-=-=-=
## =-=-=-=-= Iteration 200 Thu Mar 21 19:21:27 2019
##  =-=-=-=-=
## =-=-=-=-= Iteration 400 Thu Mar 21 19:21:32 2019
##  =-=-=-=-=
## =-=-=-=-= Iteration 600 Thu Mar 21 19:21:38 2019
##  =-=-=-=-=
## =-=-=-=-= Iteration 800 Thu Mar 21 19:21:43 2019
##  =-=-=-=-=
## =-=-=-=-= Iteration 1000 Thu Mar 21 19:21:49 2019
##  =-=-=-=-=
## =-=-=-=-= Iteration 1200 Thu Mar 21 19:21:54 2019
##  =-=-=-=-=
## =-=-=-=-= Iteration 1400 Thu Mar 21 19:22:00 2019
##  =-=-=-=-=
## =-=-=-=-= Iteration 1600 Thu Mar 21 19:22:05 2019
##  =-=-=-=-=
## =-=-=-=-= Iteration 1800 Thu Mar 21 19:22:10 2019
##  =-=-=-=-=
```

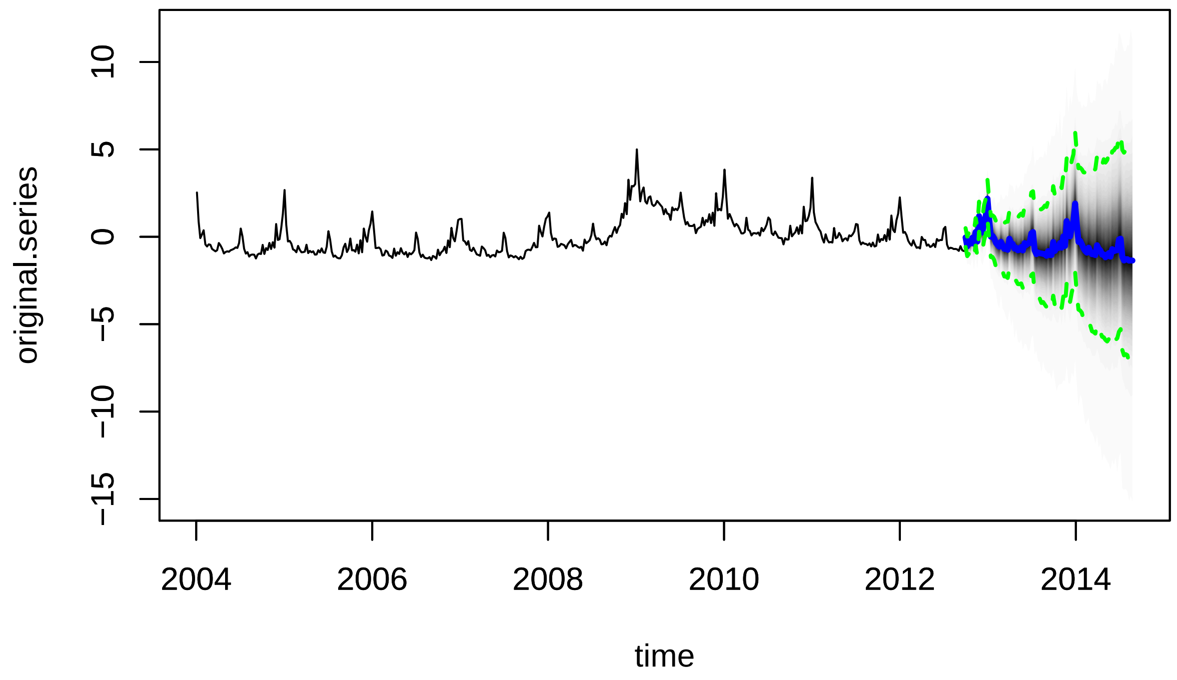
```r
burnin <- 500 # Throw away first 500
tibble(
  date = as.Date(time(claims)),
  trend = colMeans(fit$state.contributions[-(1:burnin),"trend",]),
  seasonality = colMeans(fit$state.contributions[-(1:burnin),"seasonal.52.1",])) %>%
  gather("component", "value", trend, seasonality) %>%
  ggplot(aes(x = date, y= value)) +
  geom_line() + theme_bw() +
  theme(legend.title = element_blank()) + ylab("") + xlab("") +
  facet_grid(component ~ ., scales="free") + guides(colour=FALSE) +
  theme(axis.text.x=element_text(angle = -90, hjust = 0))
```



### Predictions

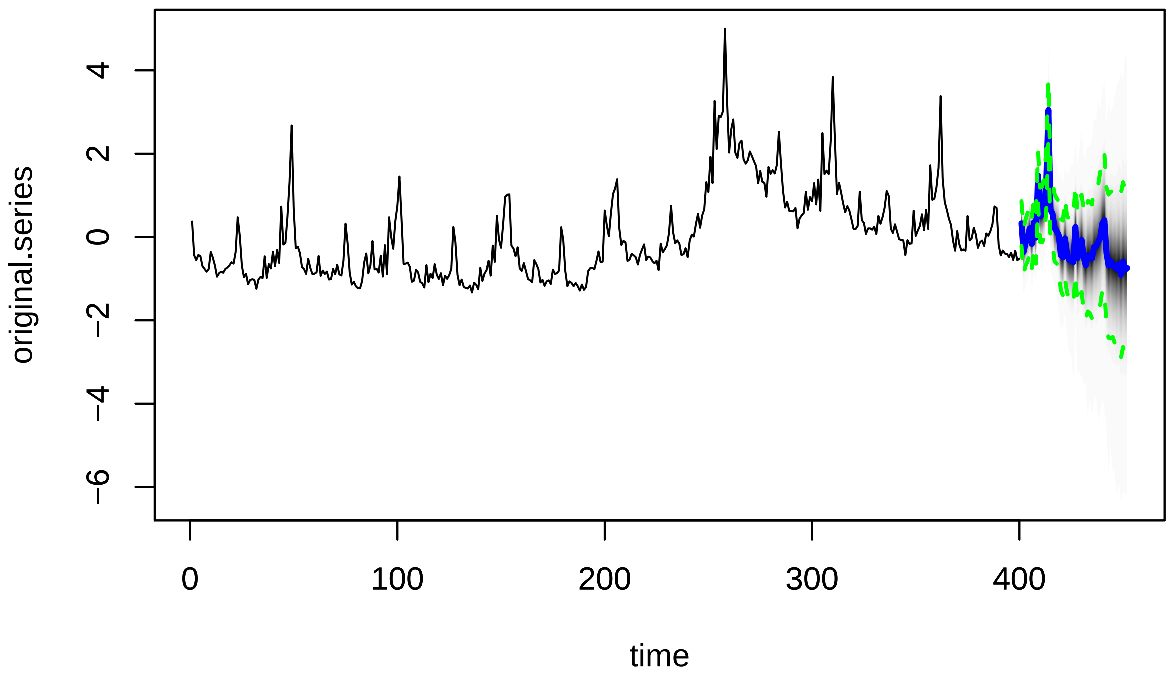
Visualization of the prediction output without applying the new model with lags and newdata.

```r
pred_wo_newolddata <- predict(fit,
                              horizon = 100,
                              burn = burnin,
                              quantiles = c(.05, .95),
                              seed = 2408)
plot(pred_wo_newolddata)
```



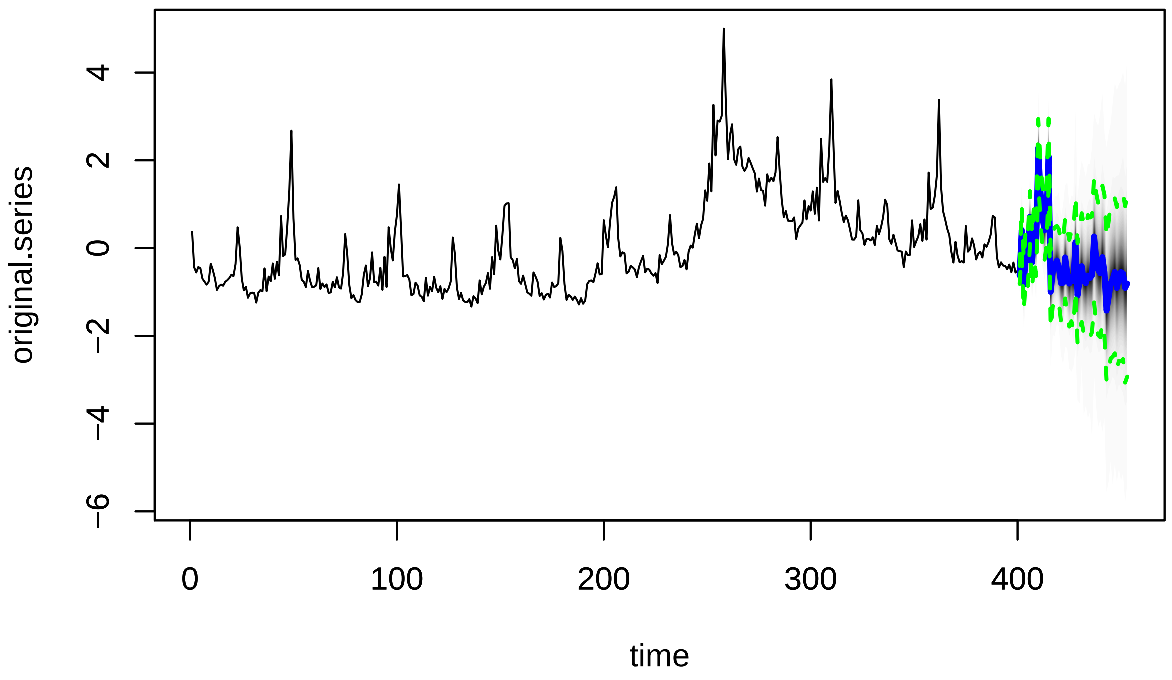
Second experiment, results of the second model using lags as regressors, so including newdata parameter between 401 and 452 timestamps, the end of the training data.

```
pred_wo_olddata <- try(predict(fit_reg,
                               horizon = 100,
                               burn = burnin,
                               quantiles = c(.05, .95),
                               seed = 2408,
                               newdata = claims_dt[401:452],
                               timestamps = 401:452))
try(plot(pred_wo_olddata))
```



Third experiment, results of the second model using lags as regressors, so including newdata parameter between 40 and 92. We validate that modification of data it´s not just stochastic aproximation, data modifies prediction.

```r
pred_wo_olddata2 <- try(predict(fit_reg,
                                horizon = 100,
                                burn = burnin,
                                quantiles = c(.05, .95),
                                seed = 2408,
                                newdata = claims_dt[40:92],
                                timestamps = 40:92))
try(plot(pred_wo_olddata2))
```



*Error*: Fourth experiment, results of the second model using lags as regressors, so including newdata parameter with data from training between 401 and 452, and oldata too. Same first error.

```
pred_w_newolddata_ts <- try(predict(fit,
                          horizon = 100,
                          burn = burnin,
                          quantiles = c(.05, .95),
                          olddata = claims_dt[50:80,],
                          olddata.timestamps = index(claims_dt[50:80,]),
                          newdata = claims_dt[401:452,!"x",with = F],
                          timestamps = 401:452,
                          seed = 2408))

pred_w_newolddata_ts
```

```
## [1] "Error : number.of.observations == length(timestamps) is not TRUE\n"
## attr(,"class")
## [1] "try-error"
## attr(,"condition")
## <simpleError: number.of.observations == length(timestamps) is not TRUE>
```

```
print(paste0("Difference of newdata length and newdata.timestamp length):",length(401:452) - length(cla
```

```
## [1] "Difference of newdata length and newdata.timestamp length):0"
```

```r
print(paste0("Difference of olddata length and olddata.timestamp length):",length(index(claims[50:80]))
```

```
## [1] "Difference of olddata length and olddata.timestamp length):0"
```

*Error*: Visualization of the prediction output applying the new model with lags and newdata, olddata without timestamp.

```r
pred_w_newolddata <- try(predict(fit_reg,
                         horizon = 100,
                         burn = burnin,
                         quantiles = c(.05, .95),
                         olddata = claims_dt[20:30,],
                         newdata = claims_dt[401:452,!"x",with = F],
                         timestamps = 401:452,
                         seed = 2408))
```

```
## Warning in is.na(observed.data$response): is.na() aplicado a un objeto que
## no es (lista o vector) de tipo 'NULL
```

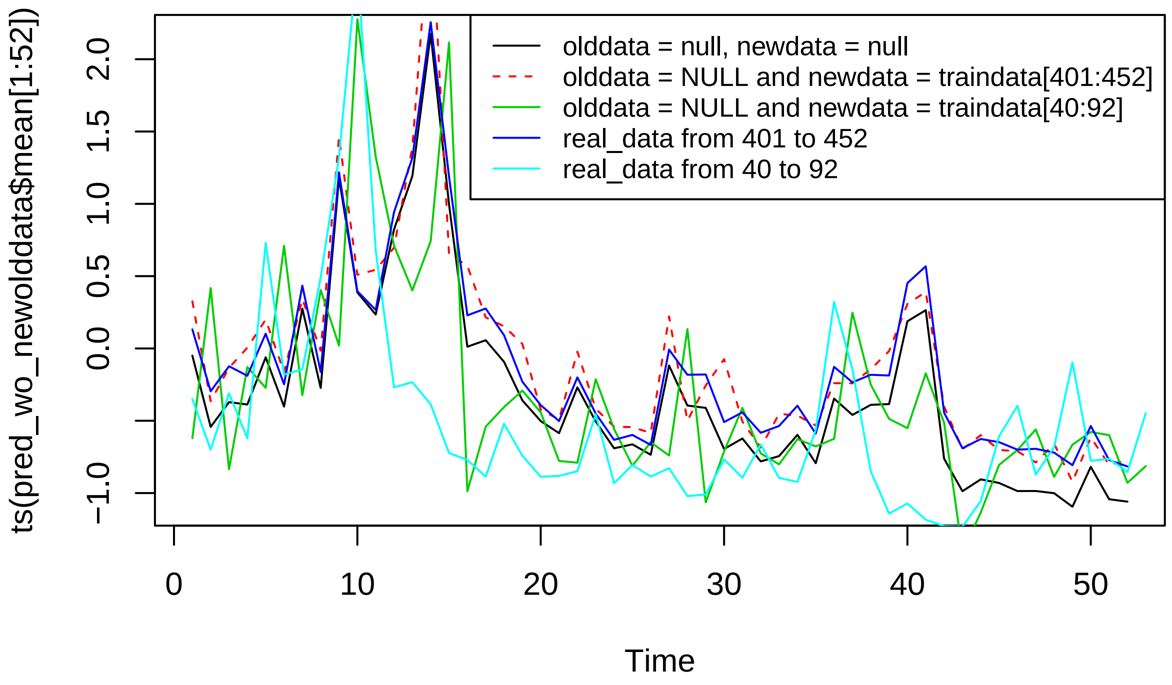
```r
pred_w_newolddata
```

```
## [1] "Error : number.of.time.points > 0 is not TRUE\n"
## attr(,"class")
## [1] "try-error"
## attr(,"condition")
## <simpleError: number.of.time.points > 0 is not TRUE>
```

We have debugged the code and it seems like the predict function removes the response variable on the .FormatObservedDataForPredictions step. Looking into this function, the function .ExtractResponse doesn´t get the response variable name from the formula parameter.

```
## [1] "Acumulated diference between without olddata prediction and just newdata from 401 to 452:-11.61

## Warning in pred_wo_newolddata$mean[1:52] - pred_wo_olddata2$mean: longitud
## de objeto mayor no es múltiplo de la longitud de uno menor

## [1] "Acumulated diference between without olddata prediction and just newdata from 40 to 92:0.506956

## Warning in pred_wo_olddata$mean[1:52] - pred_wo_olddata2$mean: longitud de
## objeto mayor no es múltiplo de la longitud de uno menor

## [1] "Acumulated diference between newdata from 401 to 452 prediction and just newdata from 40 to 92:
```

```r
plot(ts(pred_wo_newolddata$mean[1:52]), col = 1)
lines(ts(pred_wo_olddata$mean), col = 2, lty = 2)
lines(ts(pred_wo_olddata2$mean), col = 3)
lines(ts(claims_dt[401:452]$x), col = 4)
lines(ts(claims_dt[40:92]$x), col = 5)

legend("topright", legend=c("olddata = null, newdata = null", "olddata = NULL and newdata = traindata[4
       col=c(1,2,3,4,5), lty = c(1,2,1,1,1), cex=0.8)
```


olddata = null, newdata = null
olddata = NULL and newdata = traindata[401:452]
olddata = NULL and newdata = traindata[40:92]
real_data from 401 to 452
real_data from 40 to 92
ts(pred_wo_newolddata$mean[1:52])
−1.0
0.0
0.5
1.0
1.5
2.0
0
10
20
30
40
50
Time