Methods

Maria Kuruvilla

2023-05-15

## River

The Dungeness River is a 28 mile river located in northwest part of Washington State. It originates in the Olympic mountains and drains into the Strait of Juan de Fuca. The Dungeness river is fed through melting snowpack from the Olympics as well as the rain. The lowest flows are in August-September whereas the highest temperature is around 15 degree Celsius in July-August. The Dungeness River supports five species of salmonids - Chinook, Coho, Chum, Pink, and Steelhead. There are multiple hatcheries that release Chinook, Coho and Steelhead salmon into the river at various stages. The Dungeness River hatchery, Gray Wolf, Hurd Creek, and Upper Dungeness hatchery are the main hatcheries on the Dungeness River.

## Data

### Smolt trap data

Rotatry screw traps along with a variety of other techniques are used to monitor salmon smolts that migrate from their natal sites to the ocean. The smolts that are caught in the trap and enumerated and and recorded twice a day. In this paper, I have used data collected using a rotatory screw trap monitored by the Washington Department of Fish and Wildlife (WDFW) in the Dungeness river. To test the ‘pied piper hypothesis’ or the influence of hatchery salmon on the migration timing of wild salmon, we used a subset of the data of species with a hatchery program in the Dungeness river. Namely, we collated data representing Coho yearlings, Chinook subyearlings and yearlings, and Steelhead smolts. The hatchery fish are identified by the code wire tag, the adipose fin clipped or by their size when the fish are unmarked and unclipped (<10%). Since the salmon smolts caught in the trap are counted at a different time everyday and since the number of hours for which the trap is operating in the river varies everyday, we took the sum of the all the salmon smolts migrating on a given day to give a daily count along with the corresponding total number of hours for which the trap was operating that day.

### Environmental data

To account for the effect of environmental variables that affect the migration of wild salmon, we collated environmental data from various websites.

#### Temperature

We collected temperature data for years 2005-2013 from the Washington Department of Ecology website <https://ecology.wa.gov/>. For the years 2014-2020, we used temperature data collected by a logger near the trap. However, data from 2015 was missing.

#### Flow

We collected flow data in cubic feet per second for the years 2005-2020 from the Washington Department of Ecology.

#### Accumulated thermal units

We calculated ‘accumulated thermal units solstice’ for years 2005-2013 as the cumulative sum of the temperature experienced since Winter Solstice. For the rest of the years, we have temperature data only during migration season. Therefore, we calculated ‘accumulated thermal units april’ as the cumulative sum of the temperature experienced since April 1st.

#### Photoperiod

We calculated day light hours for a given day using <https://api.sunrise-sunset.org> for latitude 48.11591 and longitude -123.14318.

## Analysis

We used the multivariate auto regressive state-space (MARSS) model to estimate the influence of the number of hatchery salmon in the river, and the different environmental variables on the number of wild salmon migrating (Hinrichsen and Holmes 2009). MARSS models have been used to analyse time series data that are auto correlated. In the MARSS framework, is used to describe the observed number of wild salmon at time t and represents the true number of wild salmon at time t. They are related by the following equations:

where are the covariates affecting the states, are the effects of the covariates on , are the covariates affecting the observations, are the effects of the covariates on the observation. and are the normally distributed errors. is the variance-covariance matrix of the observation errors and is the variance-covariance matrix of the process errors. We assume that the observations and states are independent between years. Therefore and are identity matrices in our model. We assume that the observation errors are the same for all years. Therefore, is a diagonal matrix with all diagonal elements set to be equal. We assume the process errors to be different among years. Therefore, is a diagonal matrix with the diagonal elements unequal.

$$$$

where,

.

where,

.

### Model selection

Akaike Information Criterion (AIC) is often used to select the best model among models with different combinations of covariates. However, with autocorreleated data, the sample size is much lower than what is accounted for by the small sample correction for AIC (AICc). Thus, we used AICpb which uses parametric bootstrapping and was designed for MARSS models to select the best model.

data <- read.csv(here("data","dungeness\_subset\_covariates.csv"), header = TRUE)

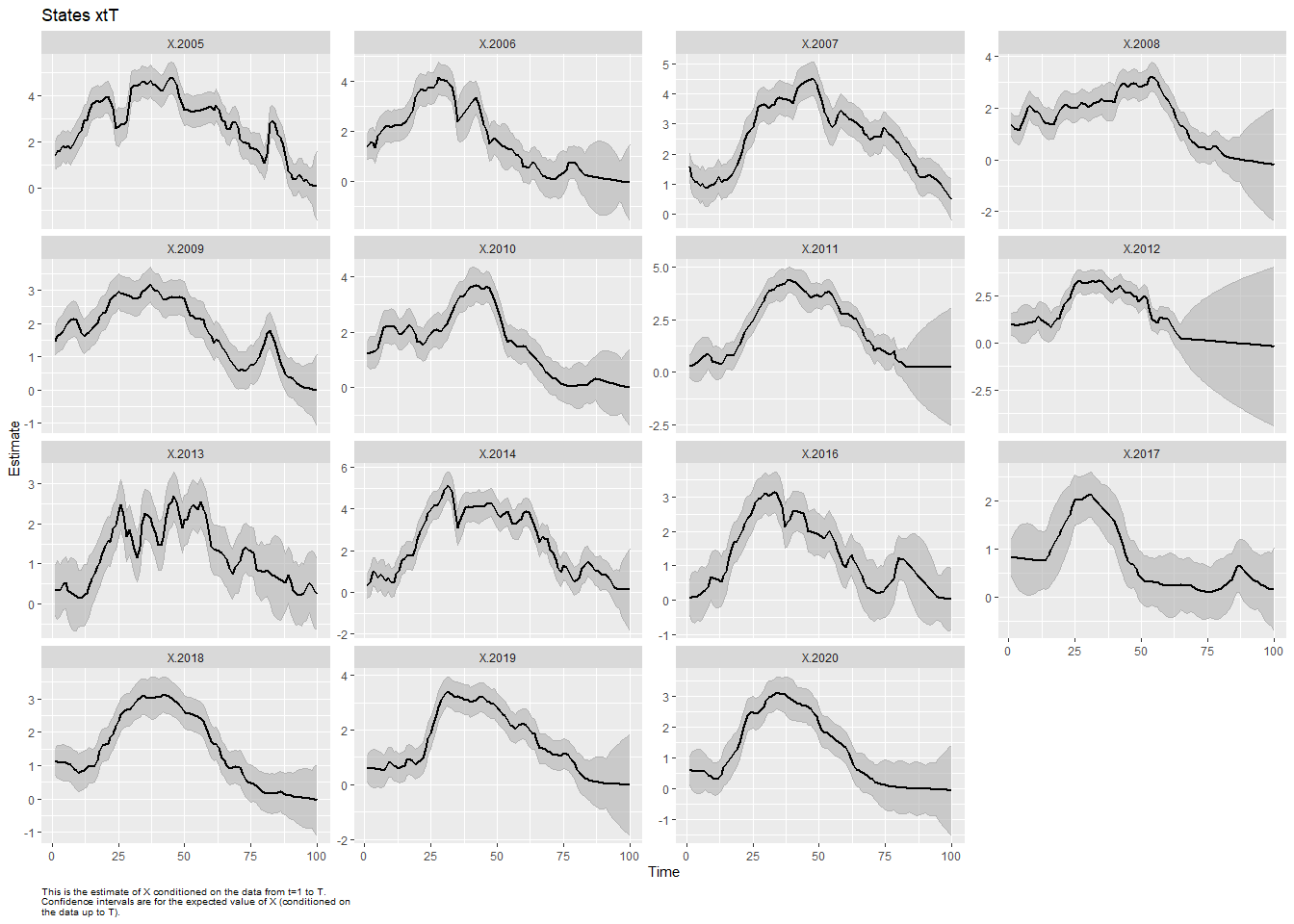
Wrangling data for coho - I am leaving out 2015 data because I do not have temperature data for 2015. Since most of the migration for coho happens between day of year 100 and 200, I am filtering on day of year. I am converting 0’s to NA and logging the rest of the data.

coho <- arrange(data,doy) %>%  
 filter(year != 2015 & doy > 100 & doy <=200) %>%  
 mutate(log.value = ifelse(coho1\_wild\_num == 0, NA, log(coho1\_wild\_num))) %>%  
 select(log.value,year,doy) %>%  
 pivot\_wider(names\_from = year, values\_from = log.value) %>%  
 column\_to\_rownames(var = "doy") %>%  
 as.matrix() %>%  
 t()

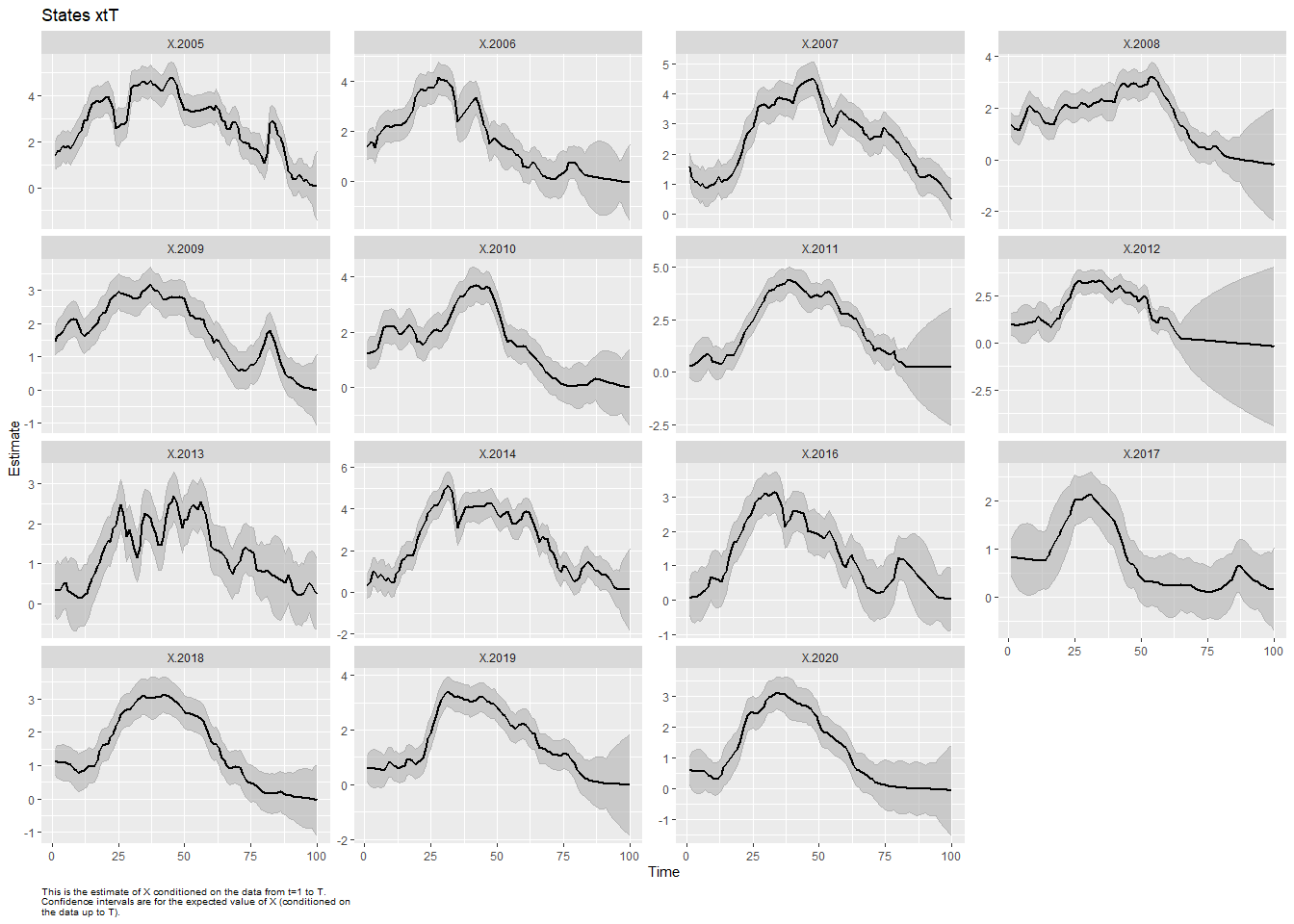
mod.list\_all\_years <- list(  
 U = "unequal",  
 R = "diagonal and equal",  
 Q = "diagonal and unequal"  
)  
  
fit\_all\_years <- MARSS(coho, model = mod.list\_all\_years, method = "BFGS")

## Success! Converged in 198 iterations.  
## Function MARSSkfas used for likelihood calculation.  
##   
## MARSS fit is  
## Estimation method: BFGS   
## Estimation converged in 198 iterations.   
## Log-likelihood: -1260.627   
## AIC: 2613.254 AICc: 2617.399   
##   
## Estimate  
## R.diag 0.324113  
## U.X.2005 -0.013504  
## U.X.2006 -0.014727  
## U.X.2007 -0.011003  
## U.X.2008 -0.015661  
## U.X.2009 -0.014848  
## U.X.2010 -0.012328  
## U.X.2011 -0.000323  
## U.X.2012 -0.011885  
## U.X.2013 -0.001055  
## U.X.2014 -0.001846  
## U.X.2016 -0.000342  
## U.X.2017 -0.006813  
## U.X.2018 -0.011591  
## U.X.2019 -0.006201  
## U.X.2020 -0.006518  
## Q.(X.2005,X.2005) 0.211416  
## Q.(X.2006,X.2006) 0.142670  
## Q.(X.2007,X.2007) 0.087939  
## Q.(X.2008,X.2008) 0.080299  
## Q.(X.2009,X.2009) 0.067363  
## Q.(X.2010,X.2010) 0.087366  
## Q.(X.2011,X.2011) 0.111461  
## Q.(X.2012,X.2012) 0.128519  
## Q.(X.2013,X.2013) 0.134372  
## Q.(X.2014,X.2014) 0.199039  
## Q.(X.2016,X.2016) 0.105745  
## Q.(X.2017,X.2017) 0.040287  
## Q.(X.2018,X.2018) 0.050315  
## Q.(X.2019,X.2019) 0.075993  
## Q.(X.2020,X.2020) 0.061311  
## x0.X.2005 1.411411  
## x0.X.2006 1.424423  
## x0.X.2007 1.565302  
## x0.X.2008 1.375251  
## x0.X.2009 1.473848  
## x0.X.2010 1.250844  
## x0.X.2011 0.297534  
## x0.X.2012 1.007455  
## x0.X.2013 0.340885  
## x0.X.2014 0.306163  
## x0.X.2016 0.055290  
## x0.X.2017 0.832968  
## x0.X.2018 1.130543  
## x0.X.2019 0.606077  
## x0.X.2020 0.597314  
## Initial states (x0) defined at t=0  
##   
## Standard errors have not been calculated.   
## Use MARSSparamCIs to compute CIs and bias estimates.

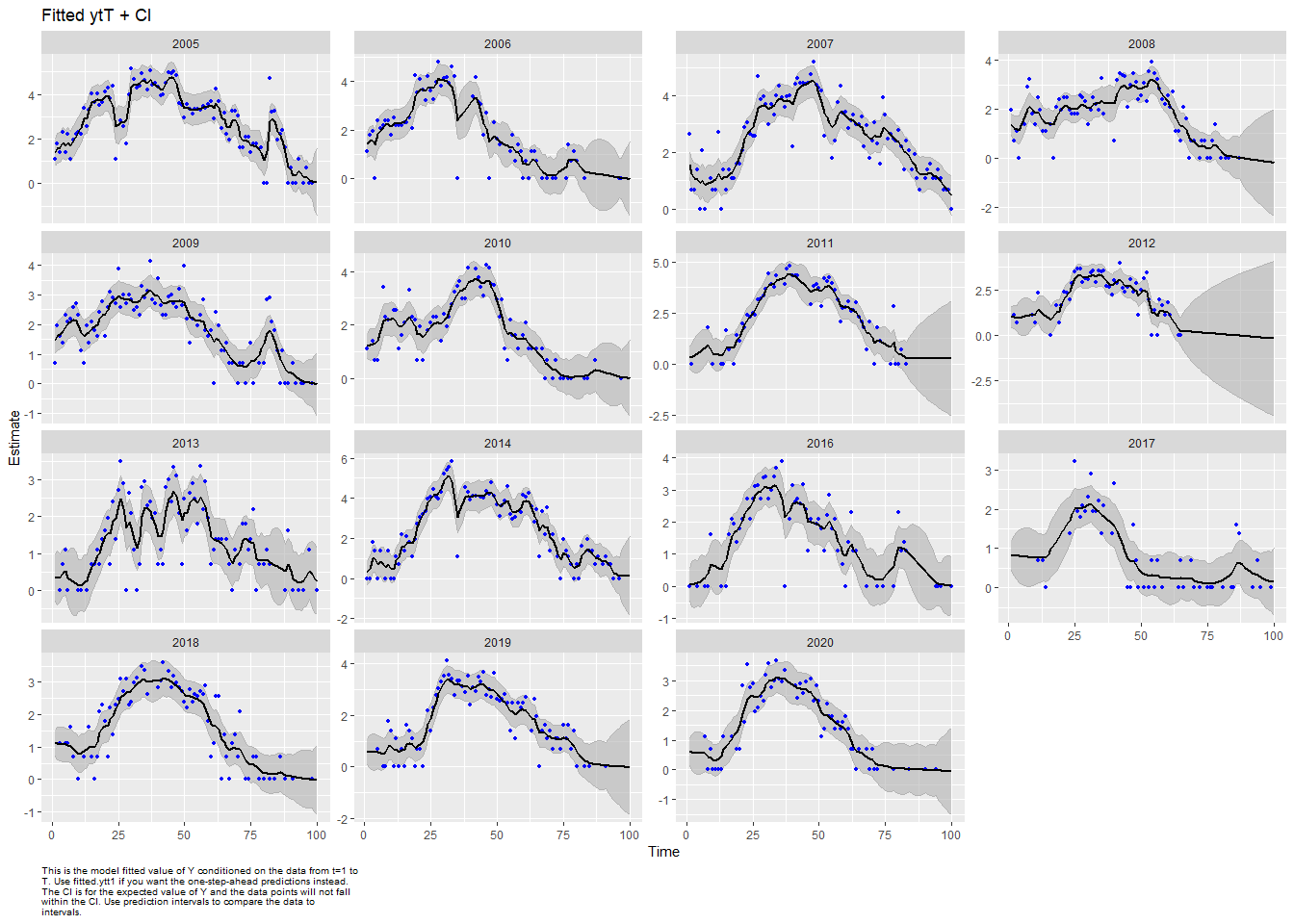
autoplot(fit\_all\_years)



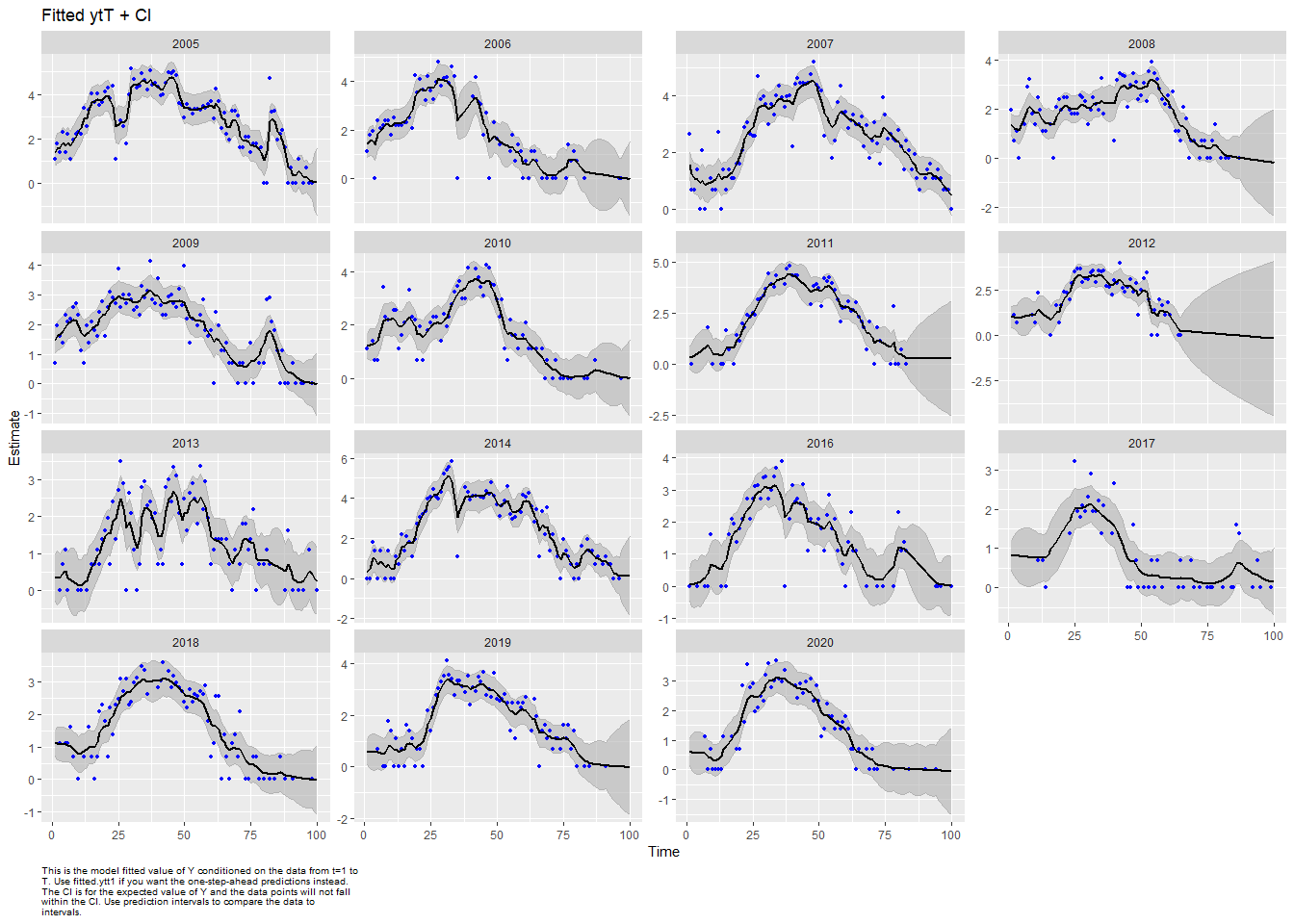
## plot.type = xtT



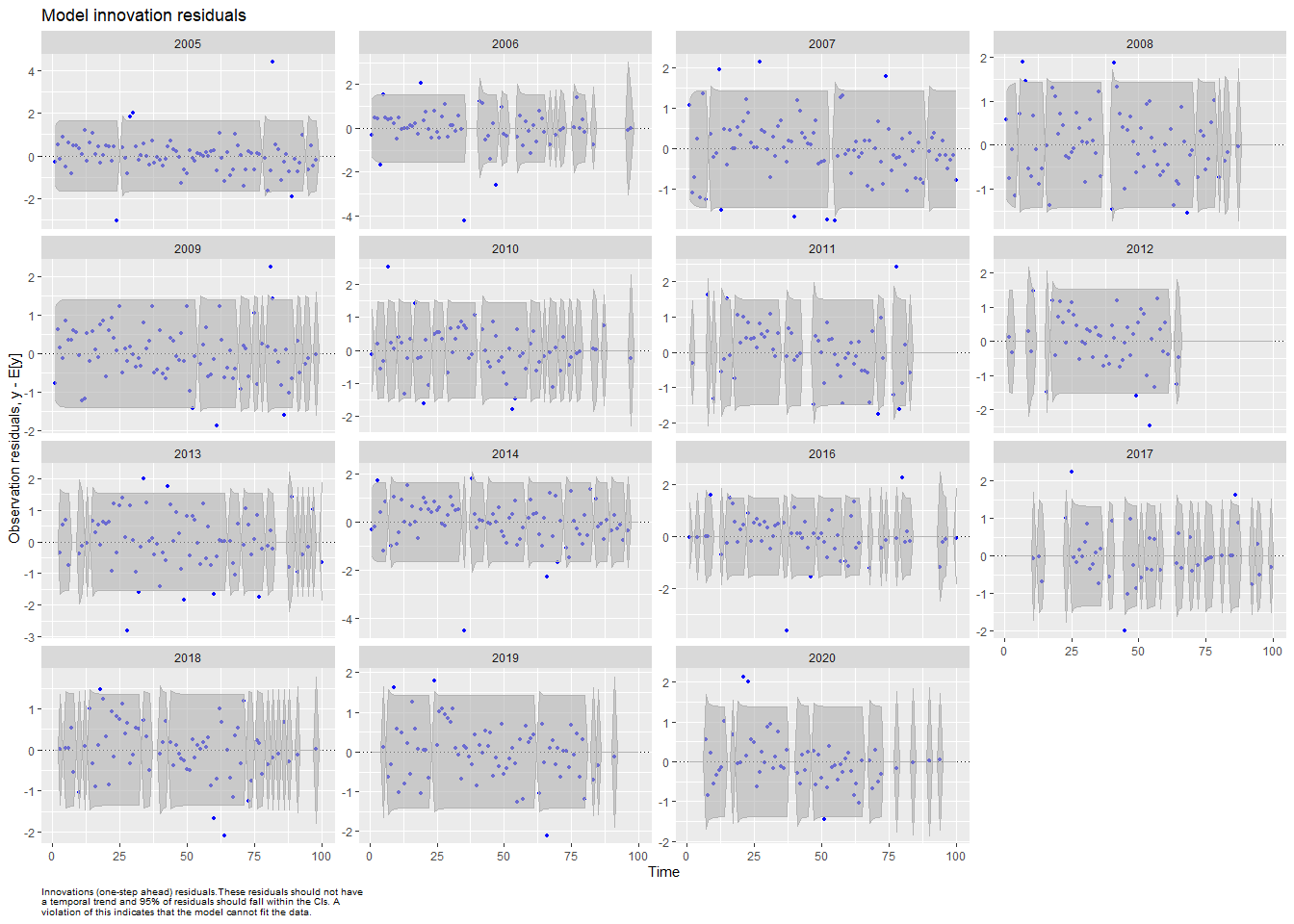
## Hit <Return> to see next plot (q to exit):



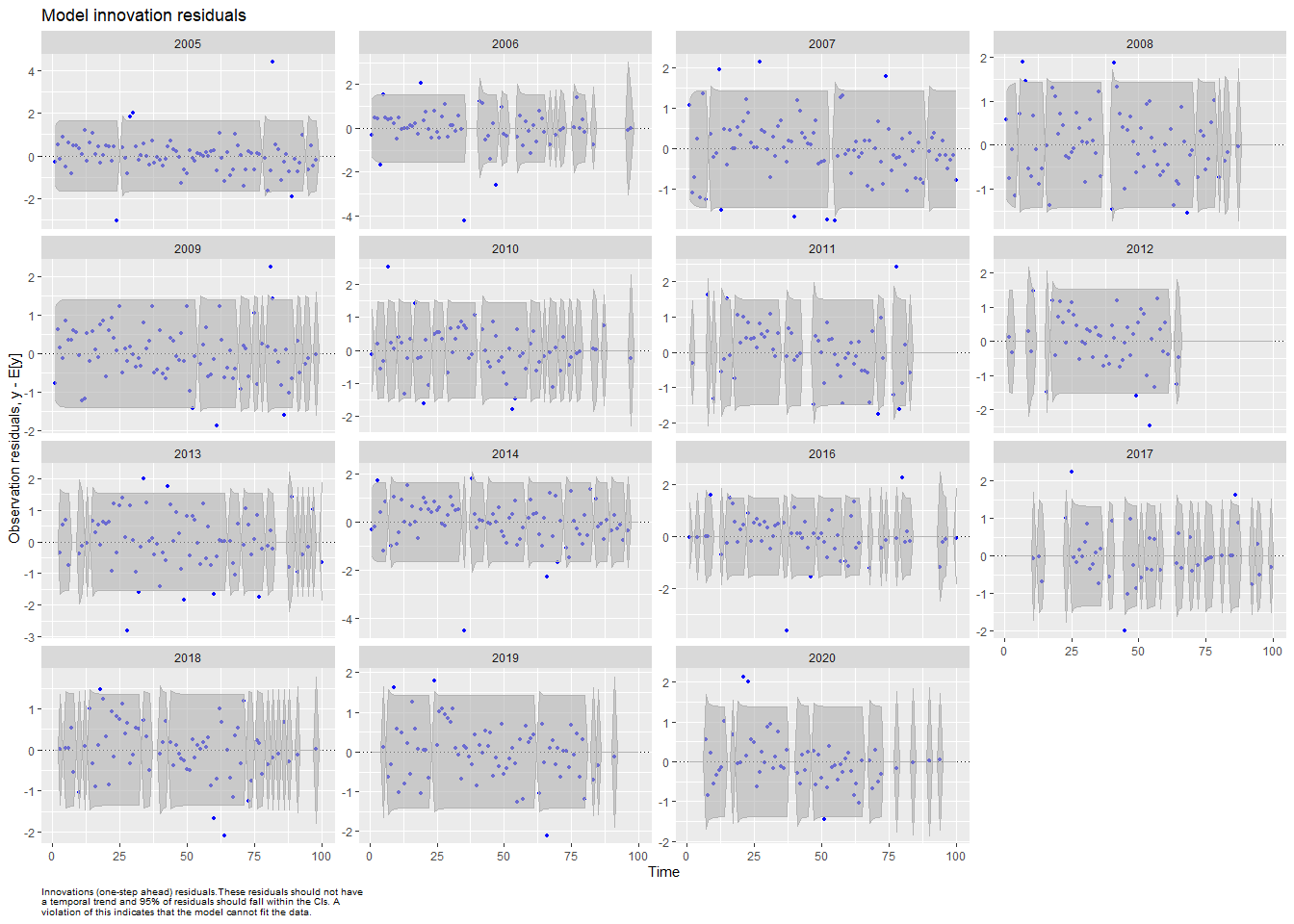
## plot.type = fitted.ytT



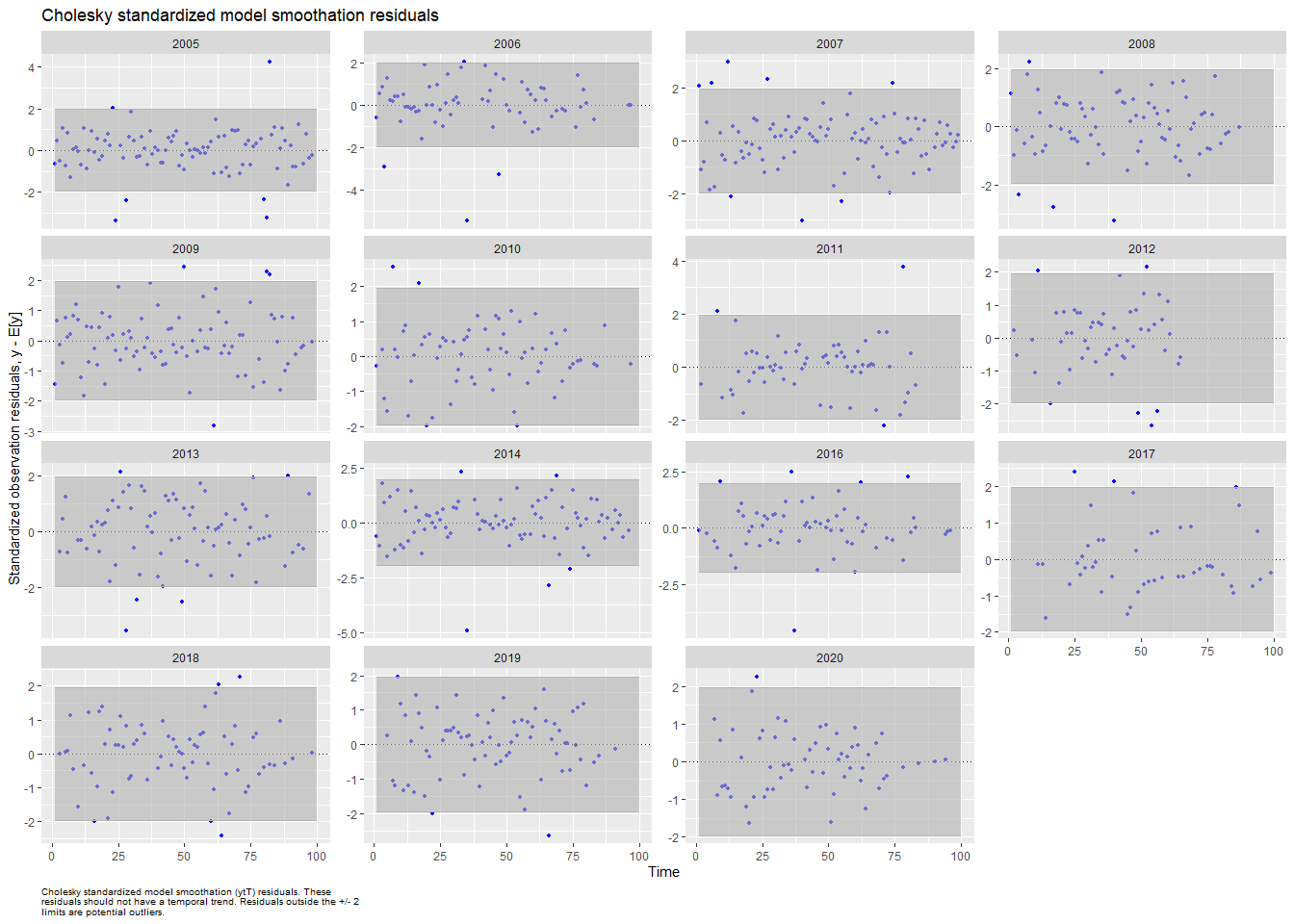
## Hit <Return> to see next plot (q to exit):



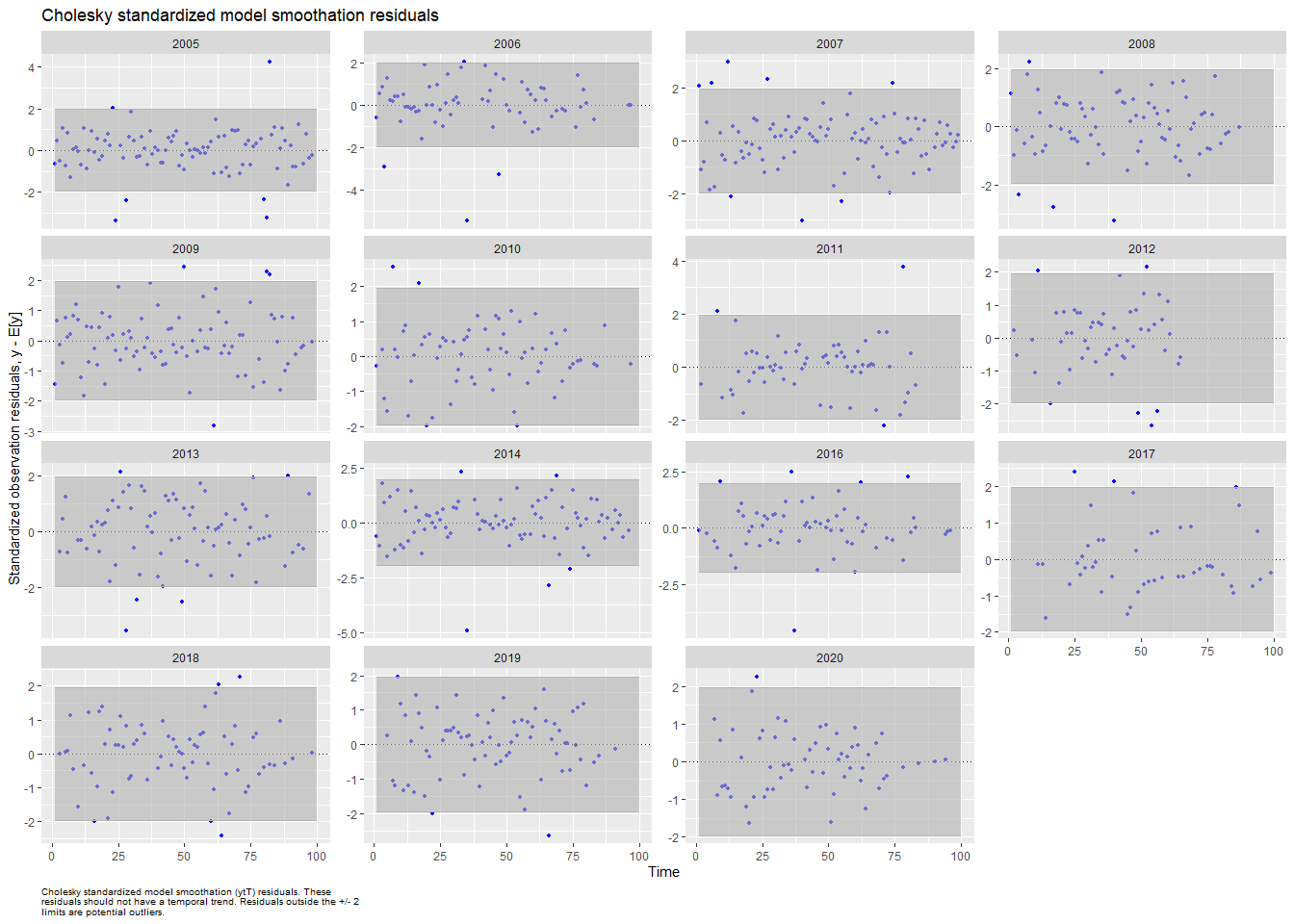
## plot.type = model.resids.ytt1



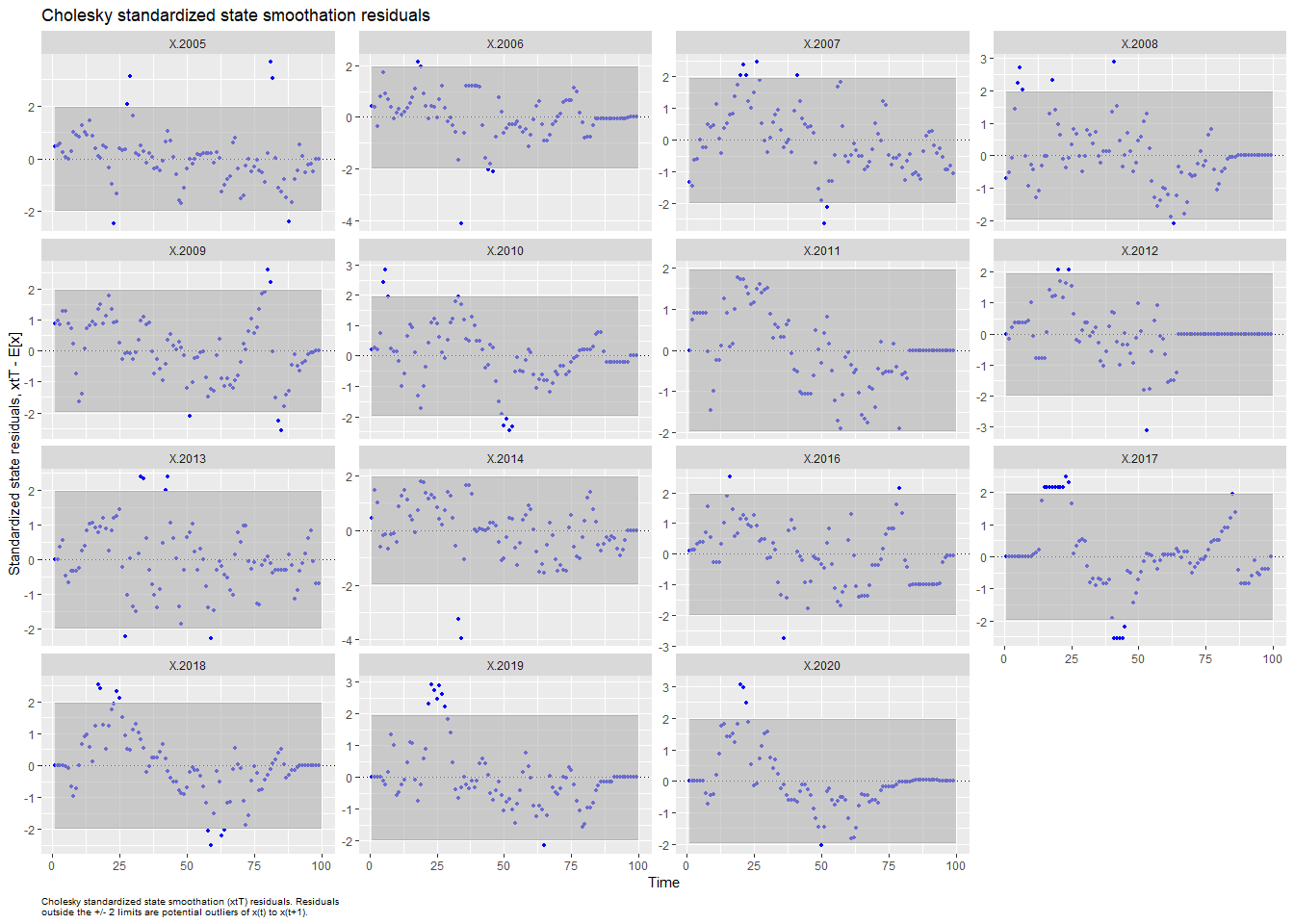
## Hit <Return> to see next plot (q to exit):



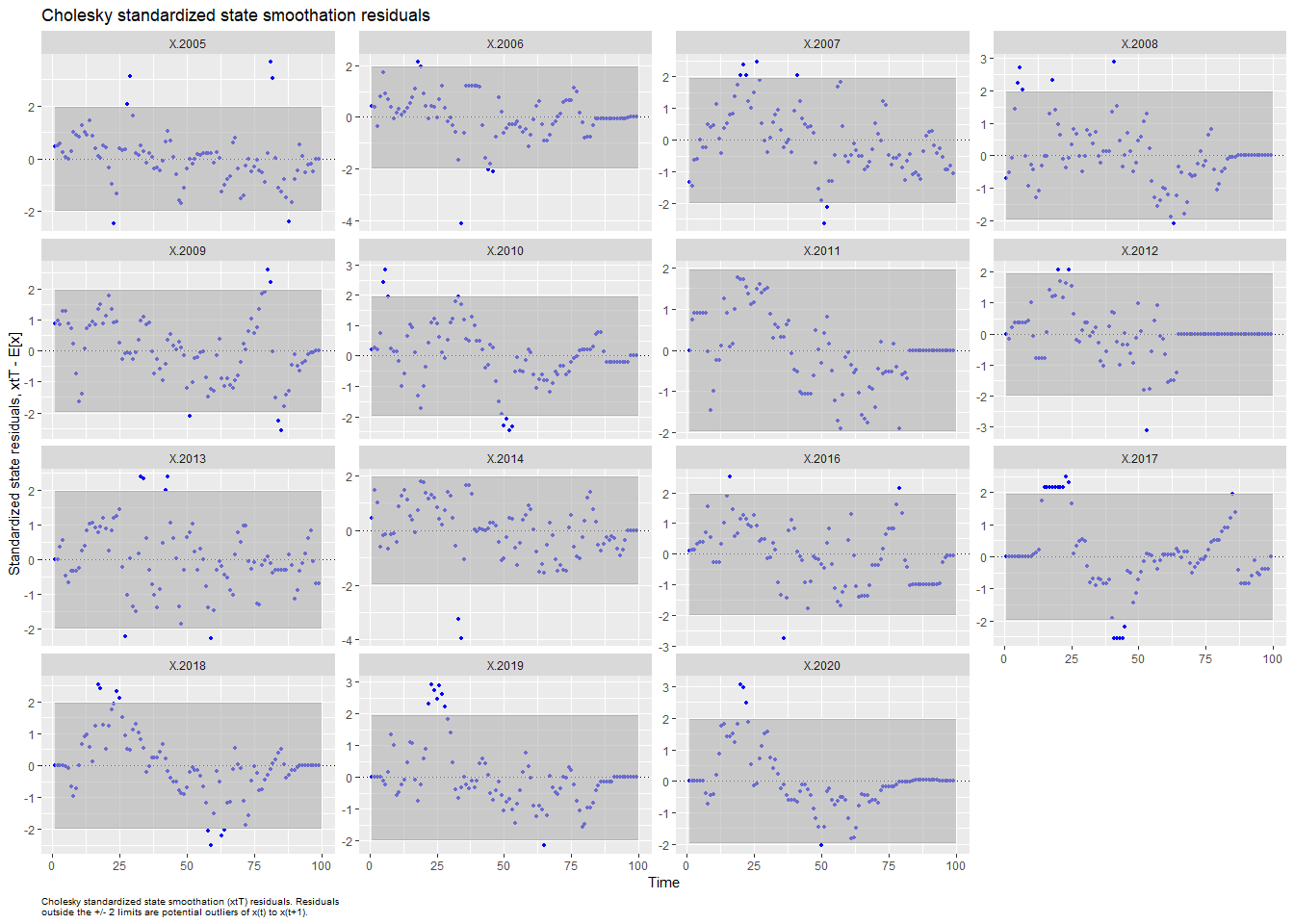
## plot.type = std.model.resids.ytT



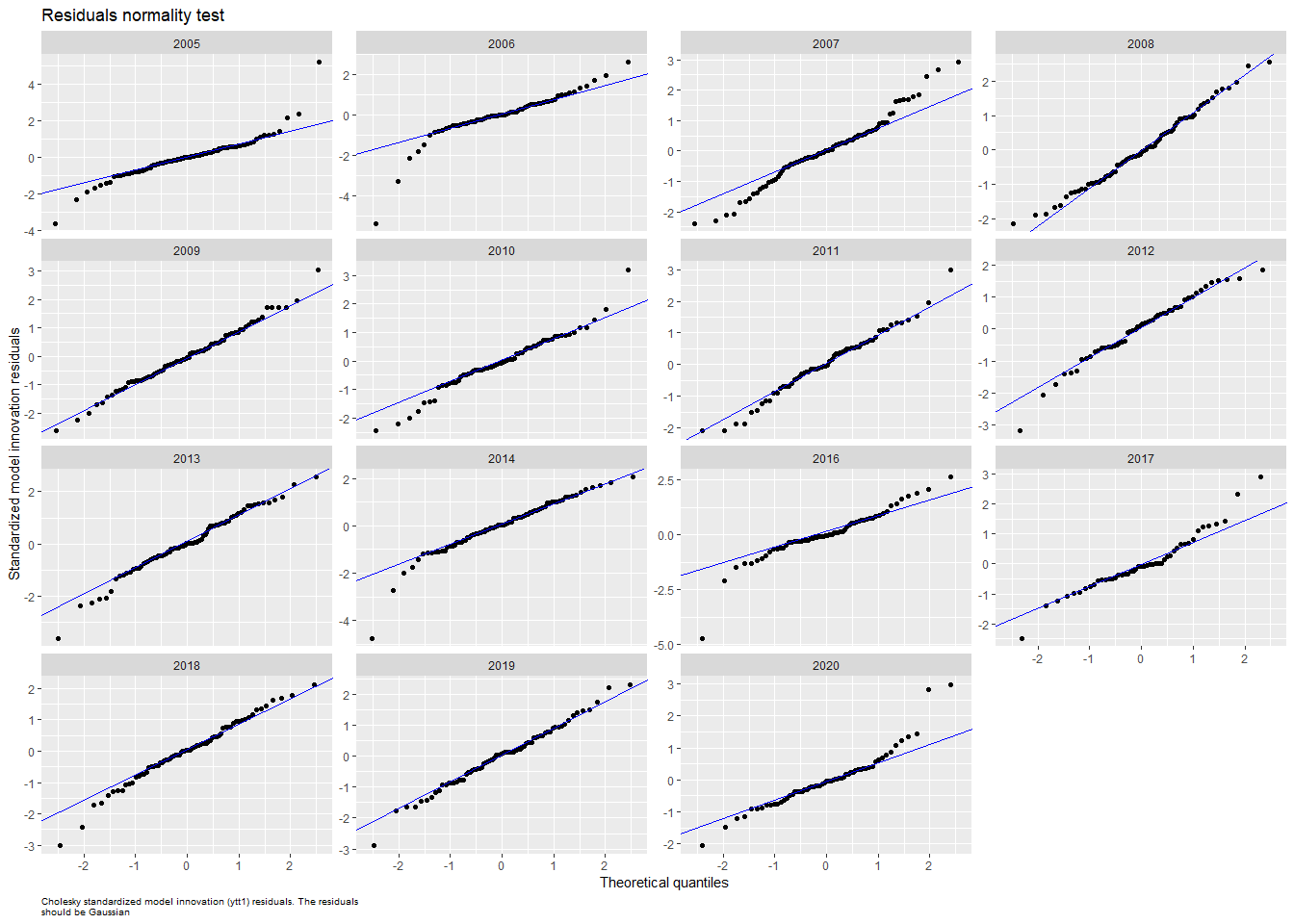
## Hit <Return> to see next plot (q to exit):



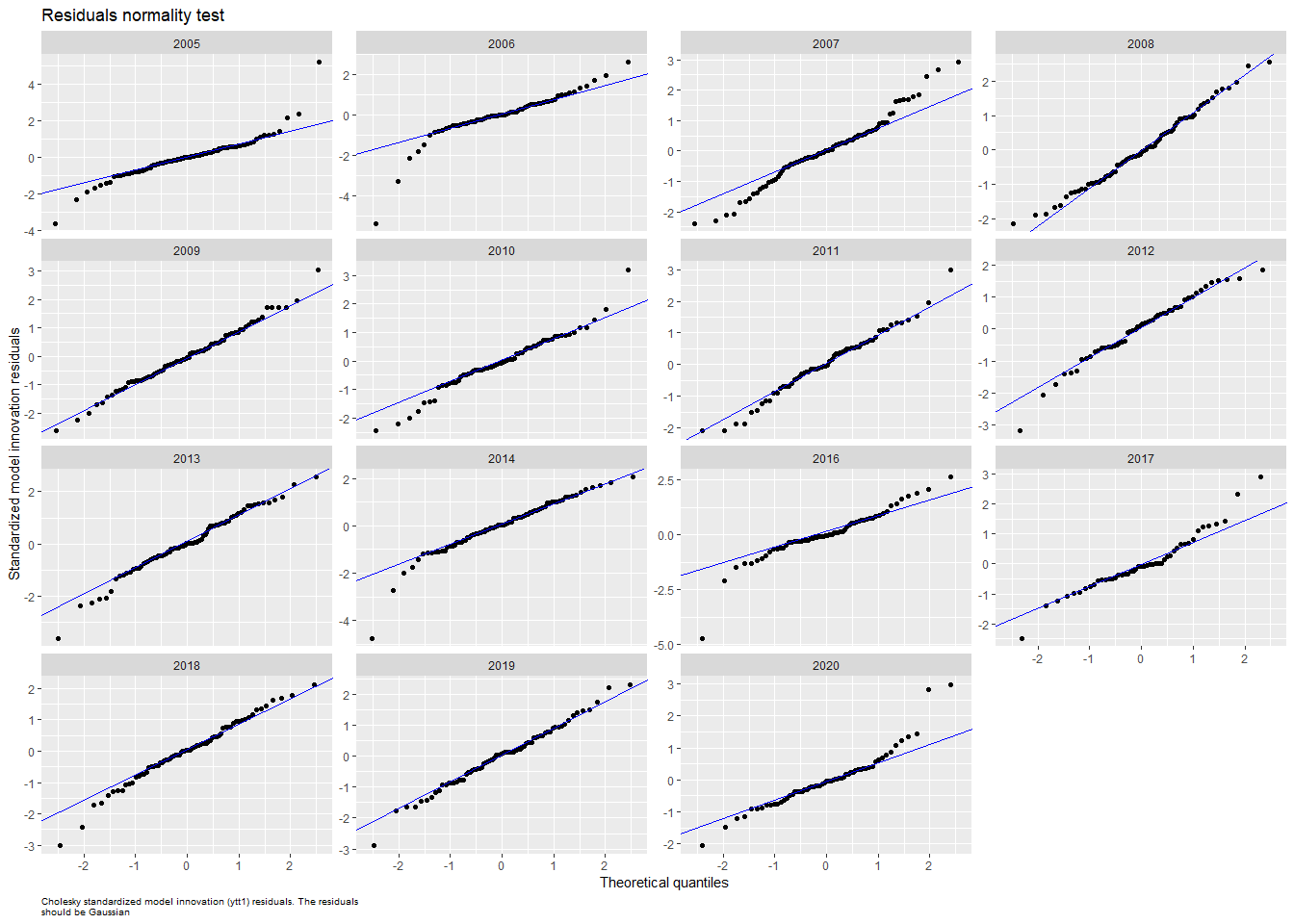
## plot.type = std.state.resids.xtT



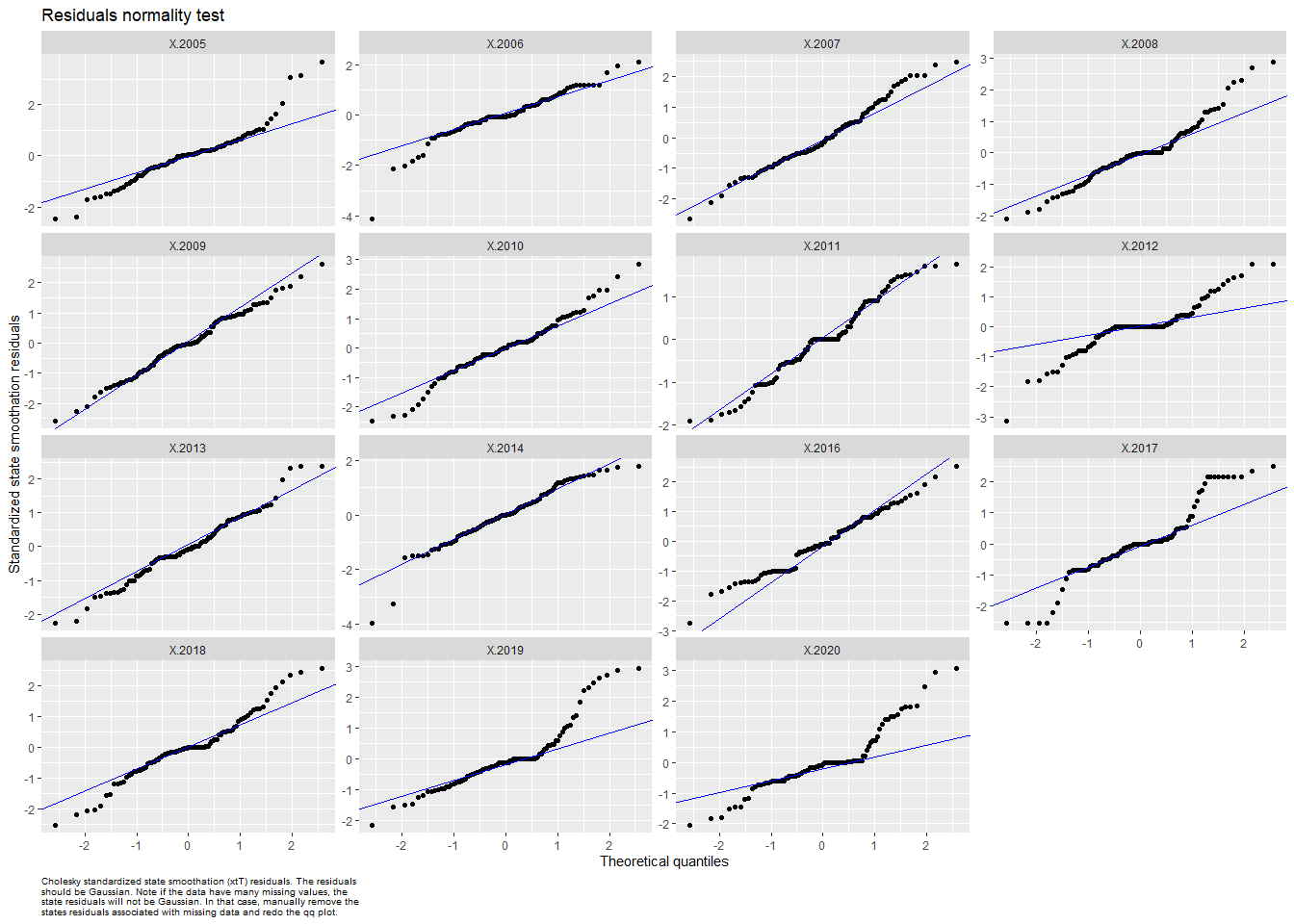
## Hit <Return> to see next plot (q to exit):



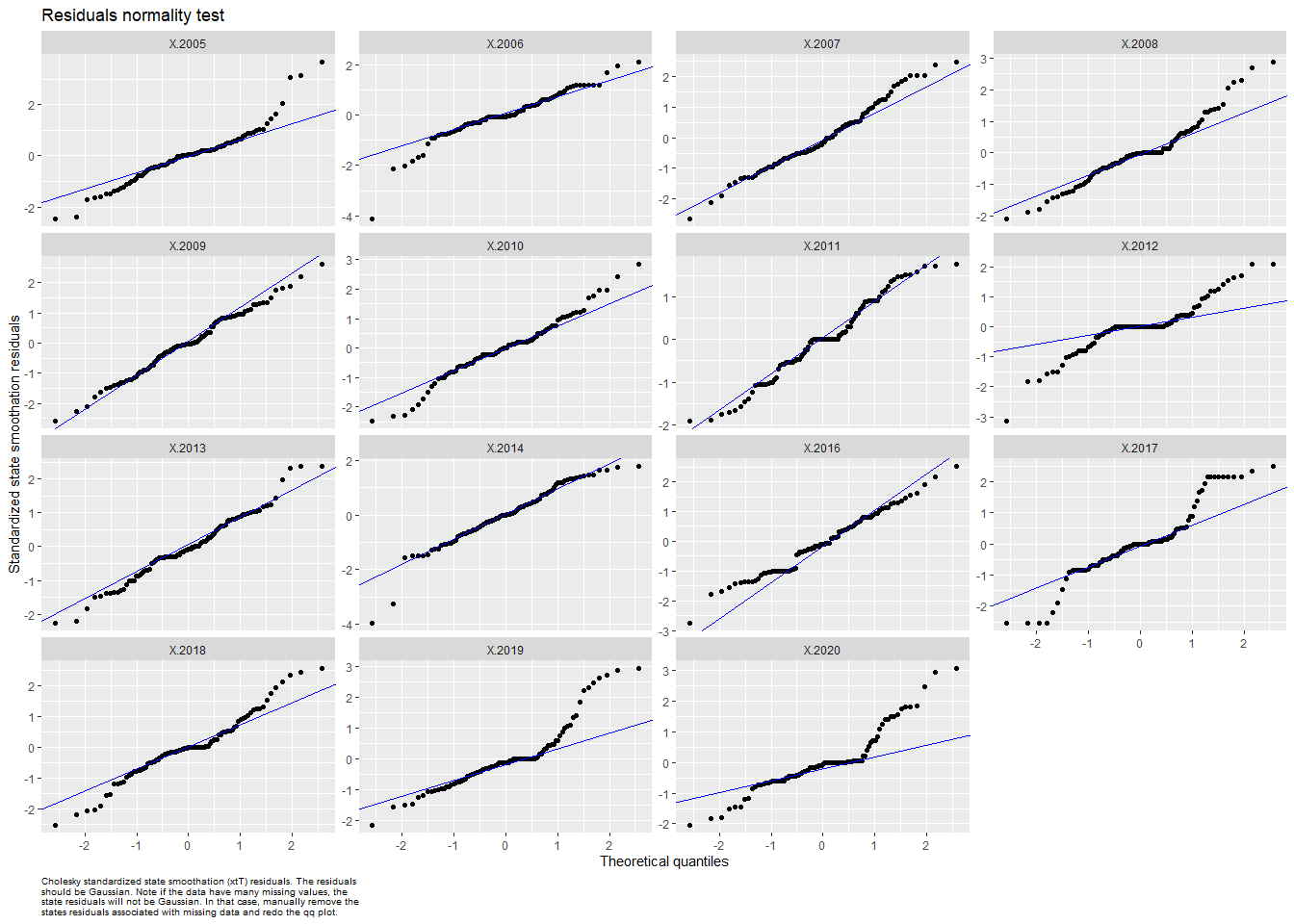
## plot.type = qqplot.std.model.resids.ytt1



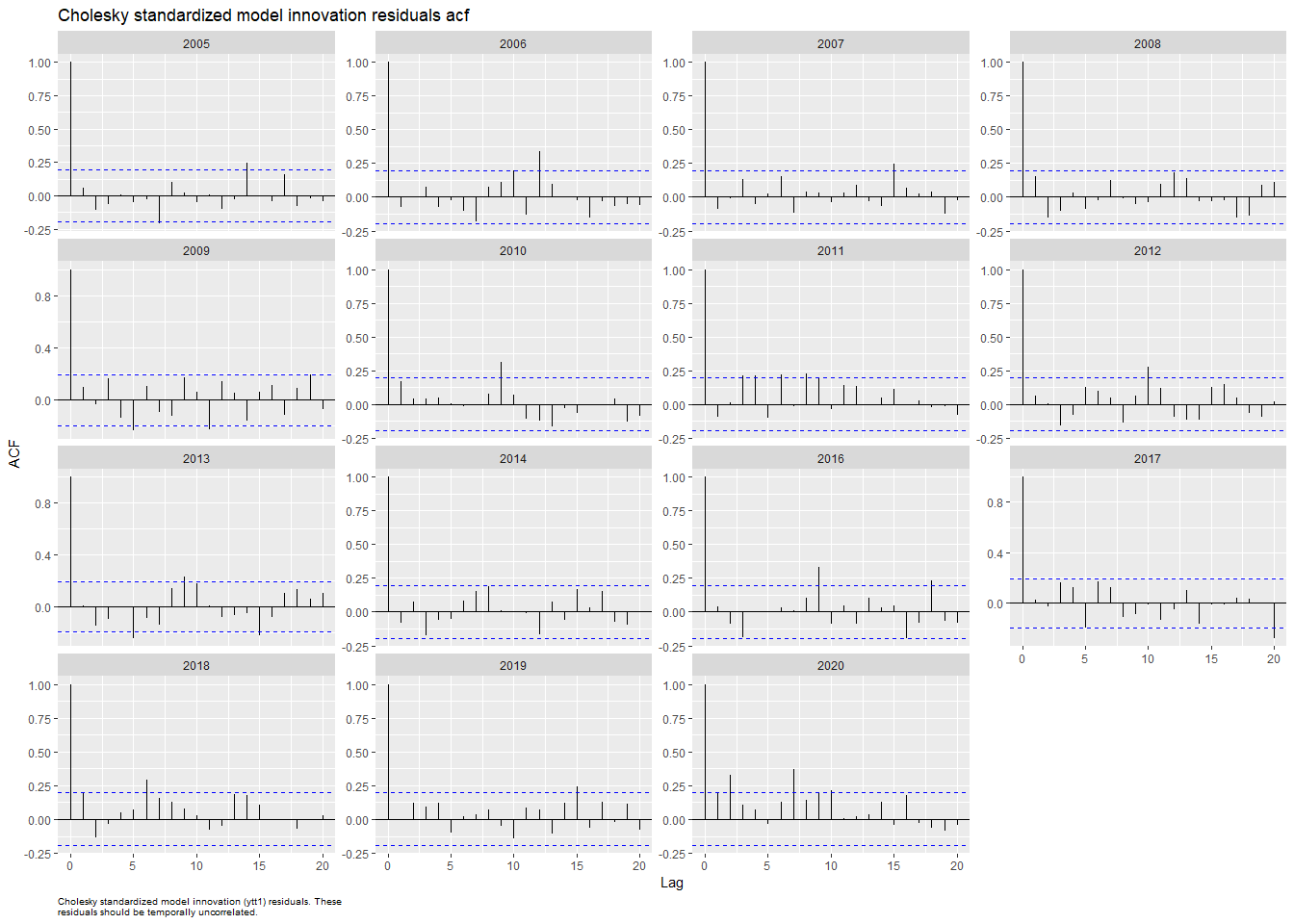
## Hit <Return> to see next plot (q to exit):



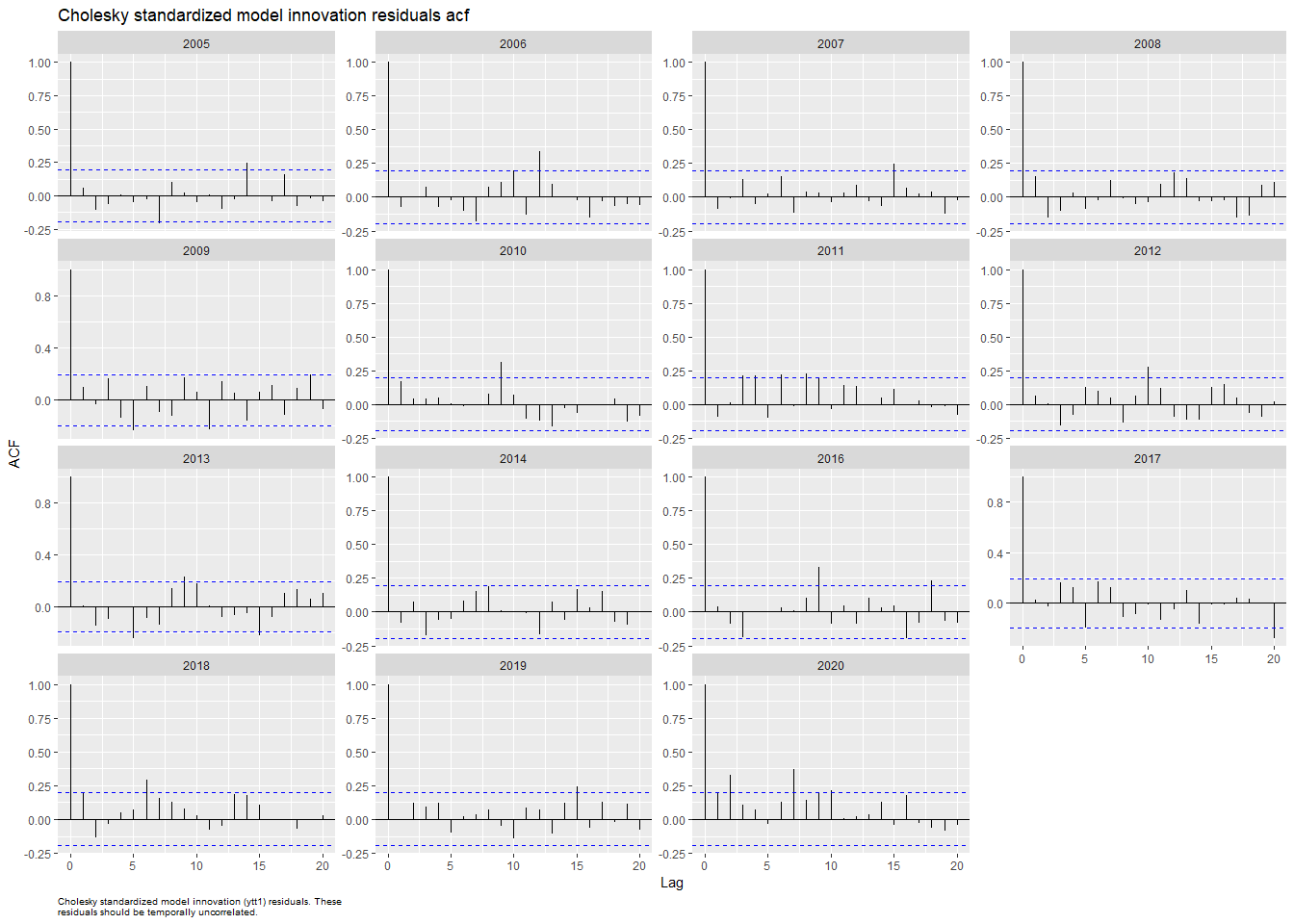
## plot.type = qqplot.std.state.resids.xtT



## Finished plots.



## plot.type = acf.std.model.resids.ytt1



## Hit <Return> to see next plot (q to exit):

Wrangling covariate data - I am interpolating hatchery data to fill in the missing values. I am making a dataframe with interpolated hatchery values and temperature values.

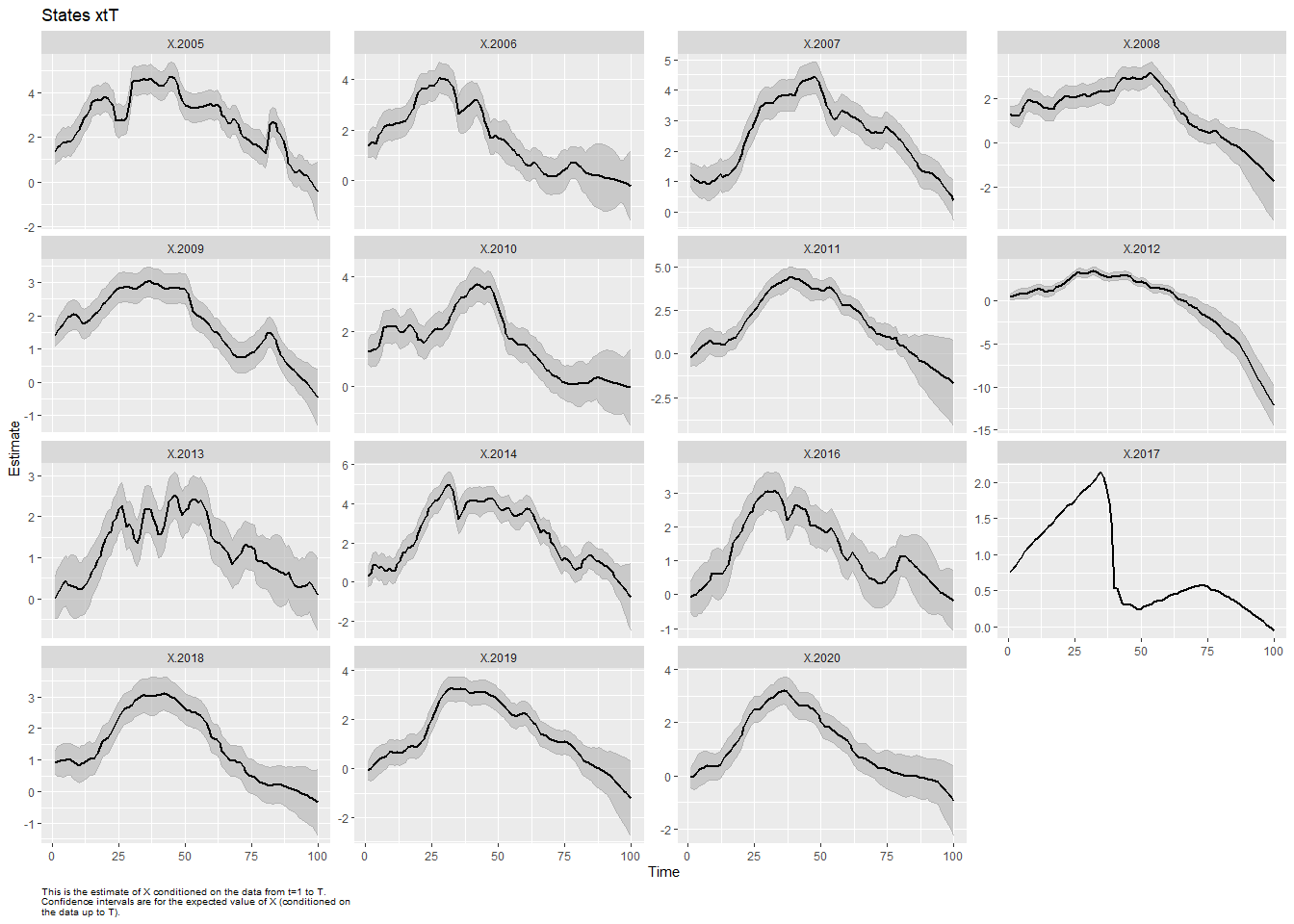
data$coho1\_hatchery\_num\_interpolate <- na.approx(data$coho1\_hatchery\_num)  
  
covariates\_coho\_hatchery\_temp\_all\_years <- arrange(data,doy) %>%  
 filter(year != 2015 & doy > 100 & doy <=200) %>%  
 select(coho1\_hatchery\_num\_interpolate,year,doy, temp) %>%  
 pivot\_wider(names\_from = year, values\_from = c(coho1\_hatchery\_num\_interpolate,temp)) %>%  
 column\_to\_rownames(var = "doy") %>%  
 as.matrix() %>%  
 t()

mod.list.hatchery\_temp\_all\_years <- list(  
 U = "unequal",  
 R = "diagonal and equal",  
 Q = "diagonal and unequal",  
 C = matrix(list("h2005",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2005",0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
 0,"h2006",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2006",0,0,0,0,0,0,0,0,0,0,0,0,0,  
 0,0,"h2007",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2007",0,0,0,0,0,0,0,0,0,0,0,0,  
 0,0,0,"h2008",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2008",0,0,0,0,0,0,0,0,0,0,0,  
 0,0,0,0,"h2009",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2009",0,0,0,0,0,0,0,0,0,0,  
 0,0,0,0,0,"h2010",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2010",0,0,0,0,0,0,0,0,0,  
 0,0,0,0,0,0,"h2011",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2011",0,0,0,0,0,0,0,0,  
 0,0,0,0,0,0,0,"h2012",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2012",0,0,0,0,0,0,0,  
 0,0,0,0,0,0,0,0,"h2013",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2013",0,0,0,0,0,0,  
 0,0,0,0,0,0,0,0,0,"h2014",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2014",0,0,0,0,0,  
 0,0,0,0,0,0,0,0,0,0,"h2016",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2016",0,0,0,0,  
 0,0,0,0,0,0,0,0,0,0,0,"h2017",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2017",0,0,0,  
 0,0,0,0,0,0,0,0,0,0,0,0,"h2018",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2018",0,0,  
 0,0,0,0,0,0,0,0,0,0,0,0,0,"h2019",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2019",0,  
 0,0,0,0,0,0,0,0,0,0,0,0,0,0,"h2020",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2020"),  
 15,30, byrow = TRUE),  
 c = covariates\_coho\_hatchery\_temp\_all\_years  
)  
  
  
  
fit\_temp\_hatchery\_all\_years <- MARSS(coho, model = mod.list.hatchery\_temp\_all\_years, method = "BFGS")

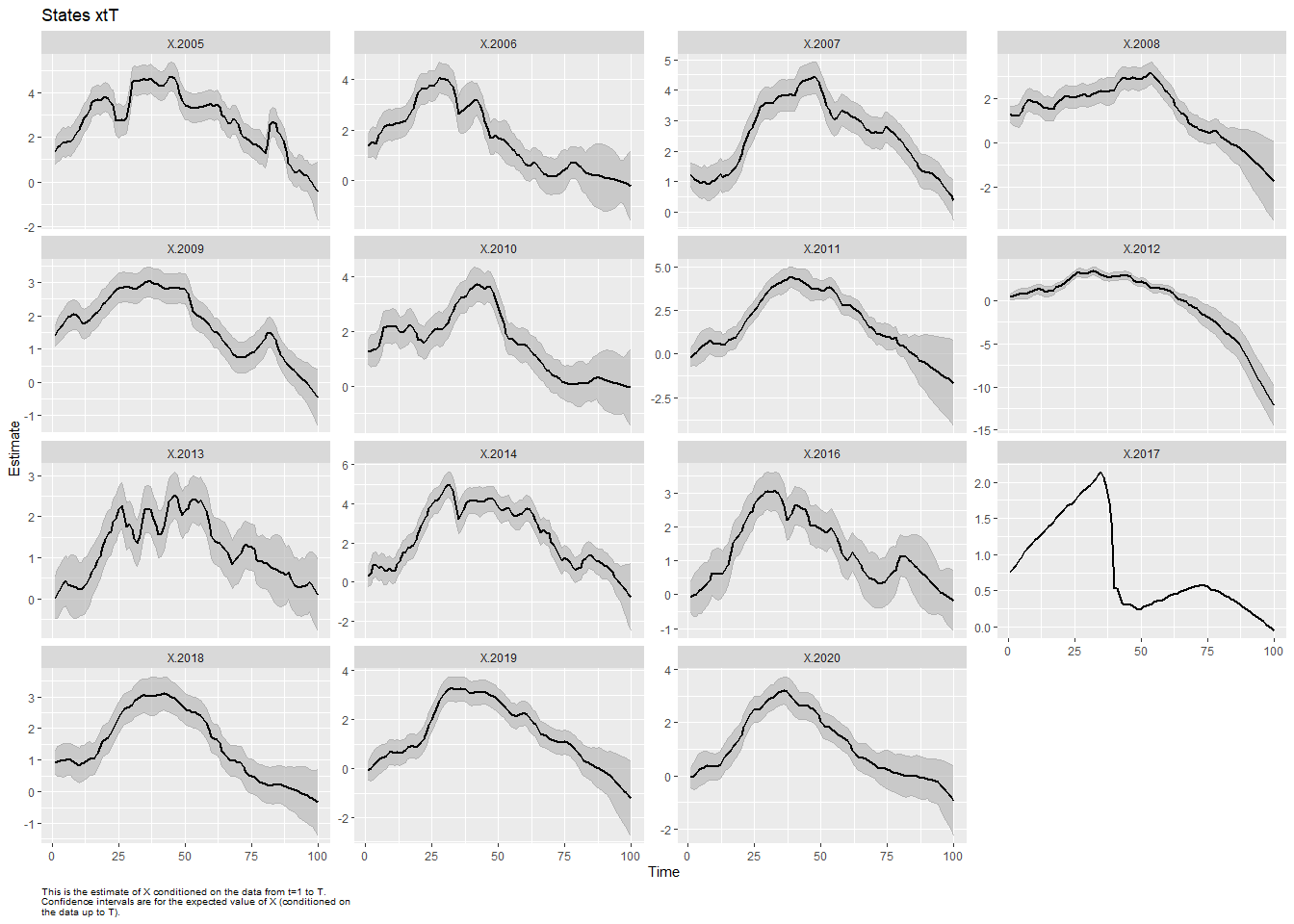
## Success! Converged in 955 iterations.  
## Function MARSSkfas used for likelihood calculation.  
##   
## MARSS fit is  
## Estimation method: BFGS   
## Estimation converged in 955 iterations.   
## Log-likelihood: -1235.898   
## AIC: 2623.796 AICc: 2635.35   
##   
## Estimate  
## R.diag 3.61e-01  
## U.X.2005 3.09e-01  
## U.X.2006 1.40e-01  
## U.X.2007 2.80e-01  
## U.X.2008 1.71e-01  
## U.X.2009 2.17e-01  
## U.X.2010 1.20e-02  
## U.X.2011 5.38e-01  
## U.X.2012 1.18e+00  
## U.X.2013 2.29e-01  
## U.X.2014 3.27e-01  
## U.X.2016 2.09e-01  
## U.X.2017 1.56e-01  
## U.X.2018 1.13e-01  
## U.X.2019 3.80e-01  
## U.X.2020 3.74e-01  
## Q.(X.2005,X.2005) 1.42e-01  
## Q.(X.2006,X.2006) 1.05e-01  
## Q.(X.2007,X.2007) 5.45e-02  
## Q.(X.2008,X.2008) 5.13e-02  
## Q.(X.2009,X.2009) 3.00e-02  
## Q.(X.2010,X.2010) 8.24e-02  
## Q.(X.2011,X.2011) 8.19e-02  
## Q.(X.2012,X.2012) 3.69e-02  
## Q.(X.2013,X.2013) 8.23e-02  
## Q.(X.2014,X.2014) 1.40e-01  
## Q.(X.2016,X.2016) 7.55e-02  
## Q.(X.2017,X.2017) 2.87e-12  
## Q.(X.2018,X.2018) 4.32e-02  
## Q.(X.2019,X.2019) 4.66e-02  
## Q.(X.2020,X.2020) 4.28e-02  
## x0.X.2005 1.26e+00  
## x0.X.2006 1.37e+00  
## x0.X.2007 1.11e+00  
## x0.X.2008 1.24e+00  
## x0.X.2009 1.34e+00  
## x0.X.2010 1.28e+00  
## x0.X.2011 -2.54e-01  
## x0.X.2012 4.32e-01  
## x0.X.2013 -5.08e-02  
## x0.X.2014 2.06e-01  
## x0.X.2016 -1.43e-01  
## x0.X.2017 6.91e-01  
## x0.X.2018 9.00e-01  
## x0.X.2019 -1.48e-01  
## x0.X.2020 -1.25e-01  
## C.h2005 3.22e-04  
## C.h2006 -1.14e-03  
## C.h2007 -5.72e-05  
## C.h2008 -5.26e-05  
## C.h2009 -1.21e-04  
## C.h2010 9.46e-05  
## C.h2011 -6.35e-05  
## C.h2012 1.71e-04  
## C.h2013 -1.53e-04  
## C.h2014 2.45e-04  
## C.h2016 8.24e-05  
## C.h2017 -6.18e-04  
## C.h2018 -1.37e-05  
## C.h2019 -7.81e-05  
## C.h2020 -2.90e-04  
## C.t2005 -3.34e-02  
## C.t2006 -1.64e-02  
## C.t2007 -3.12e-02  
## C.t2008 -2.43e-02  
## C.t2009 -2.34e-02  
## C.t2010 -3.69e-03  
## C.t2011 -7.27e-02  
## C.t2012 -1.67e-01  
## C.t2013 -2.49e-02  
## C.t2014 -3.61e-02  
## C.t2016 -1.80e-02  
## C.t2017 -1.44e-02  
## C.t2018 -1.26e-02  
## C.t2019 -3.61e-02  
## C.t2020 -3.80e-02  
## Initial states (x0) defined at t=0  
##   
## Standard errors have not been calculated.   
## Use MARSSparamCIs to compute CIs and bias estimates.

Plotting model diagnostics.

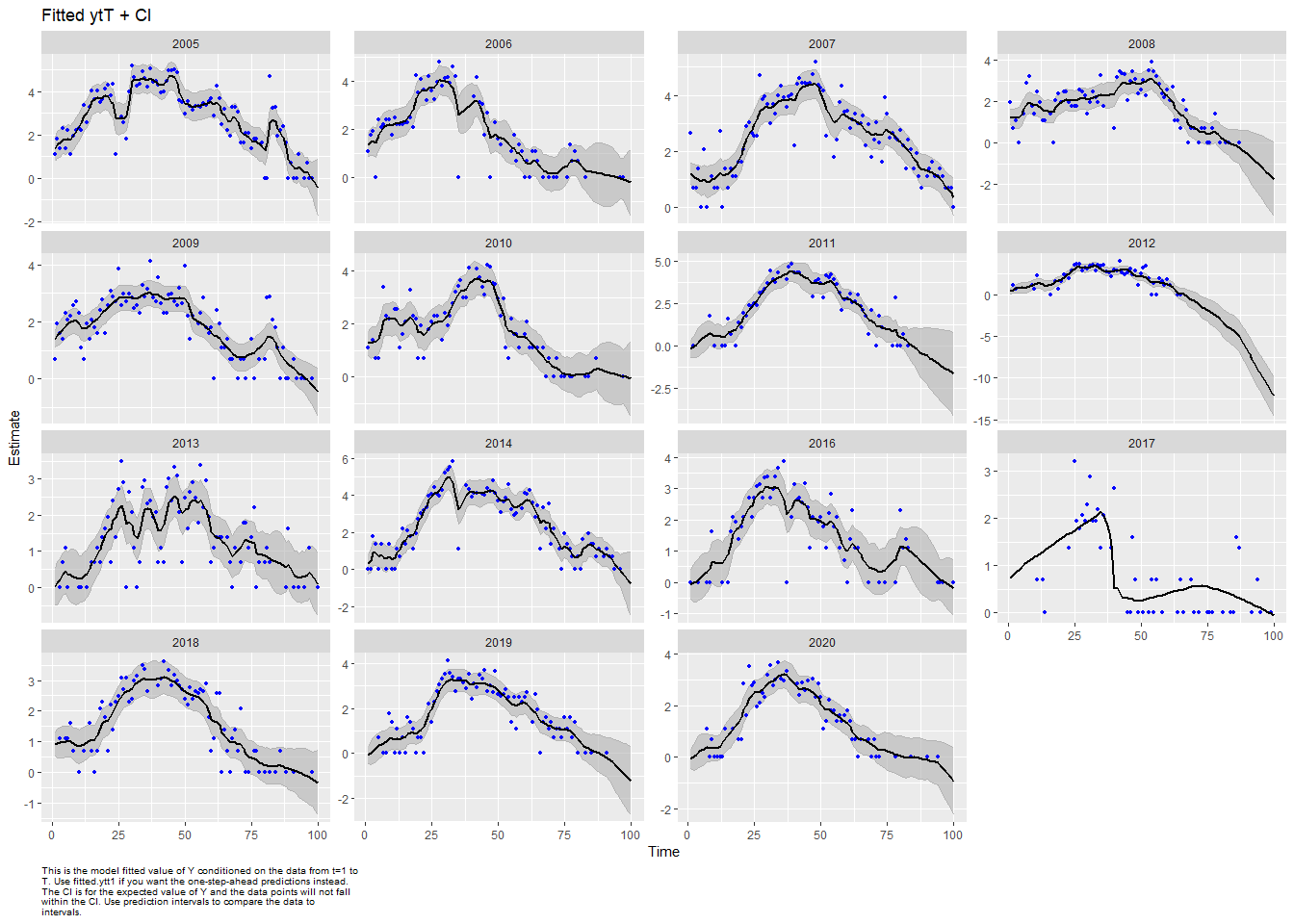
autoplot(fit\_temp\_hatchery\_all\_years)



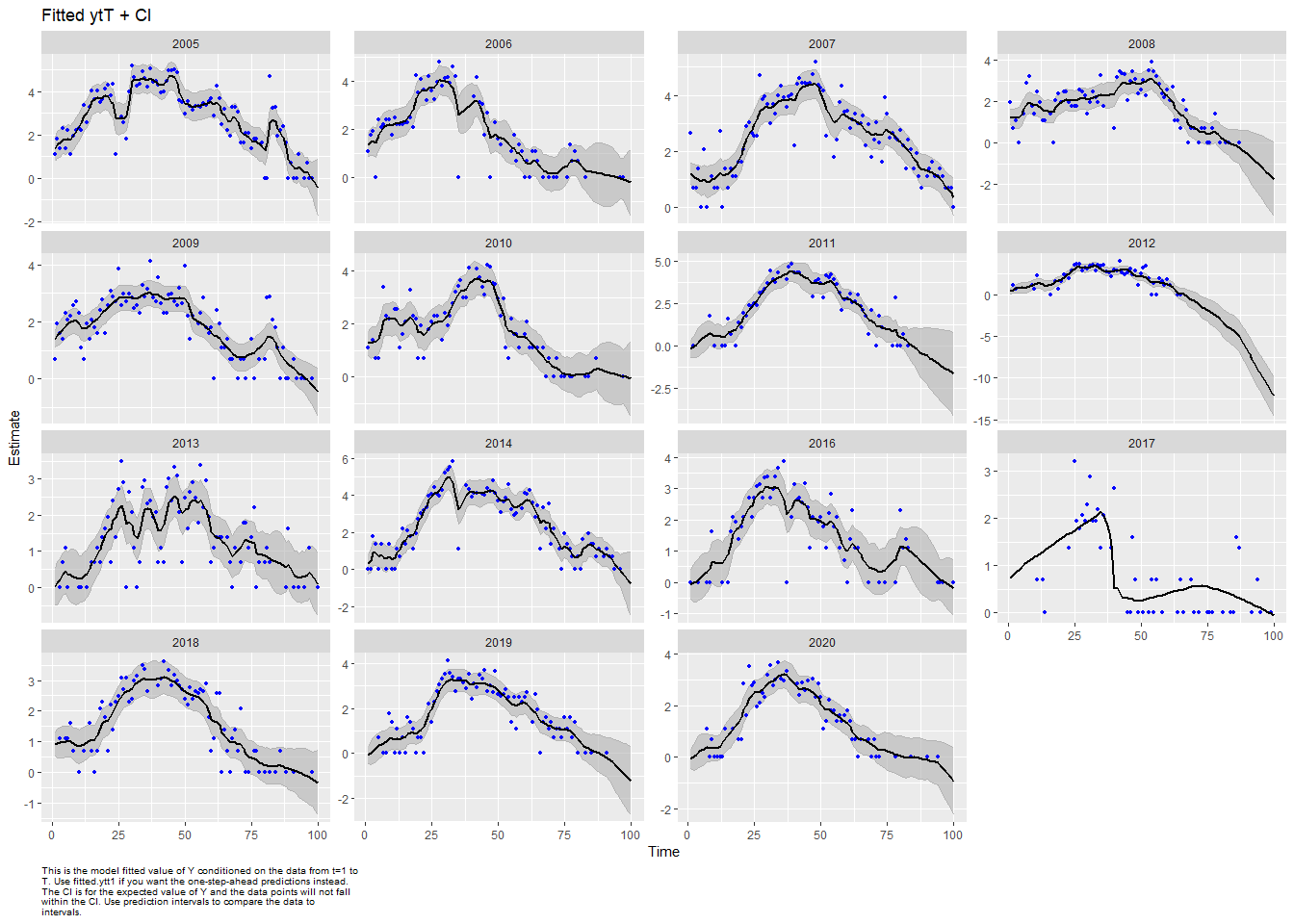
## plot.type = xtT



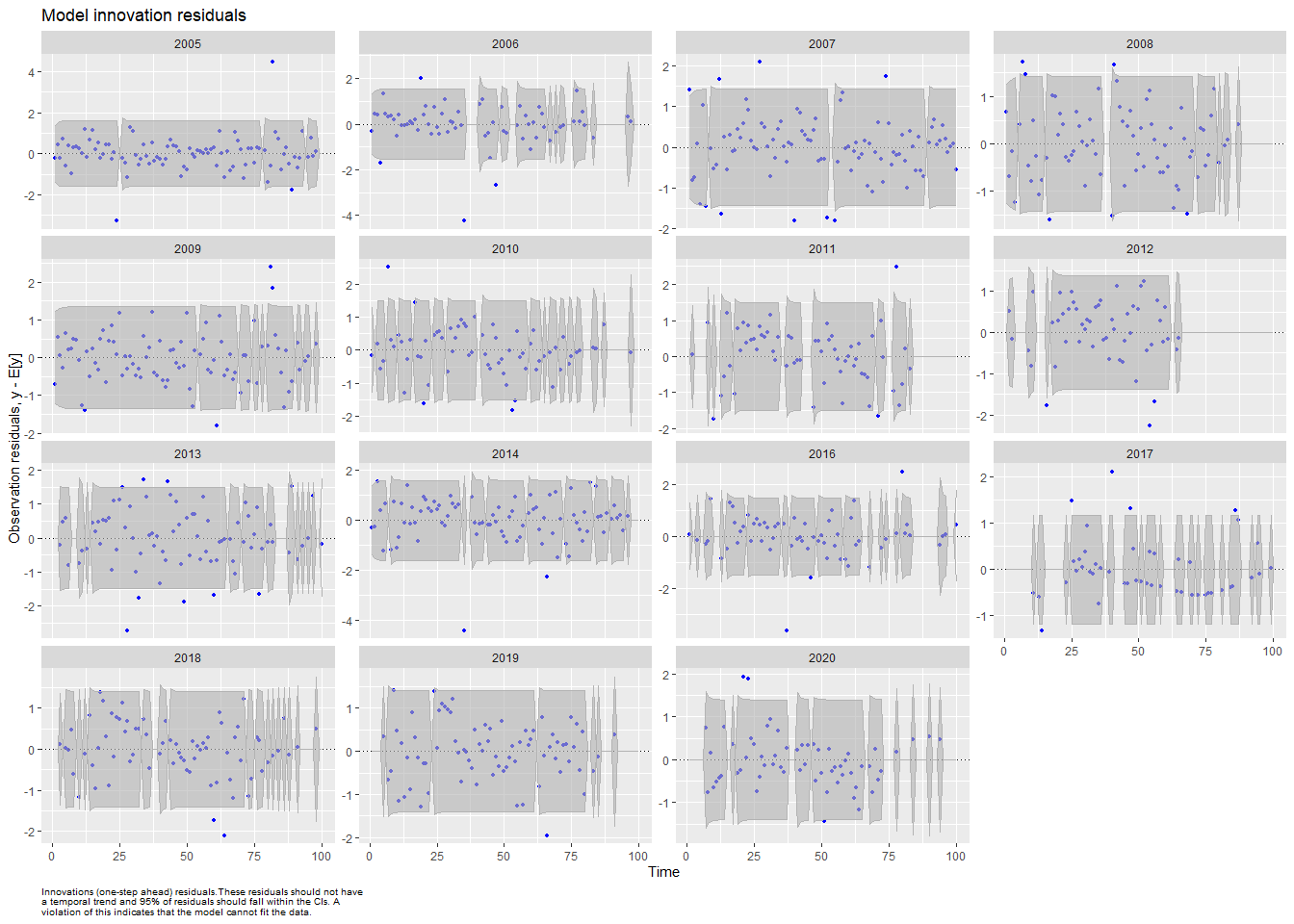
## Hit <Return> to see next plot (q to exit):



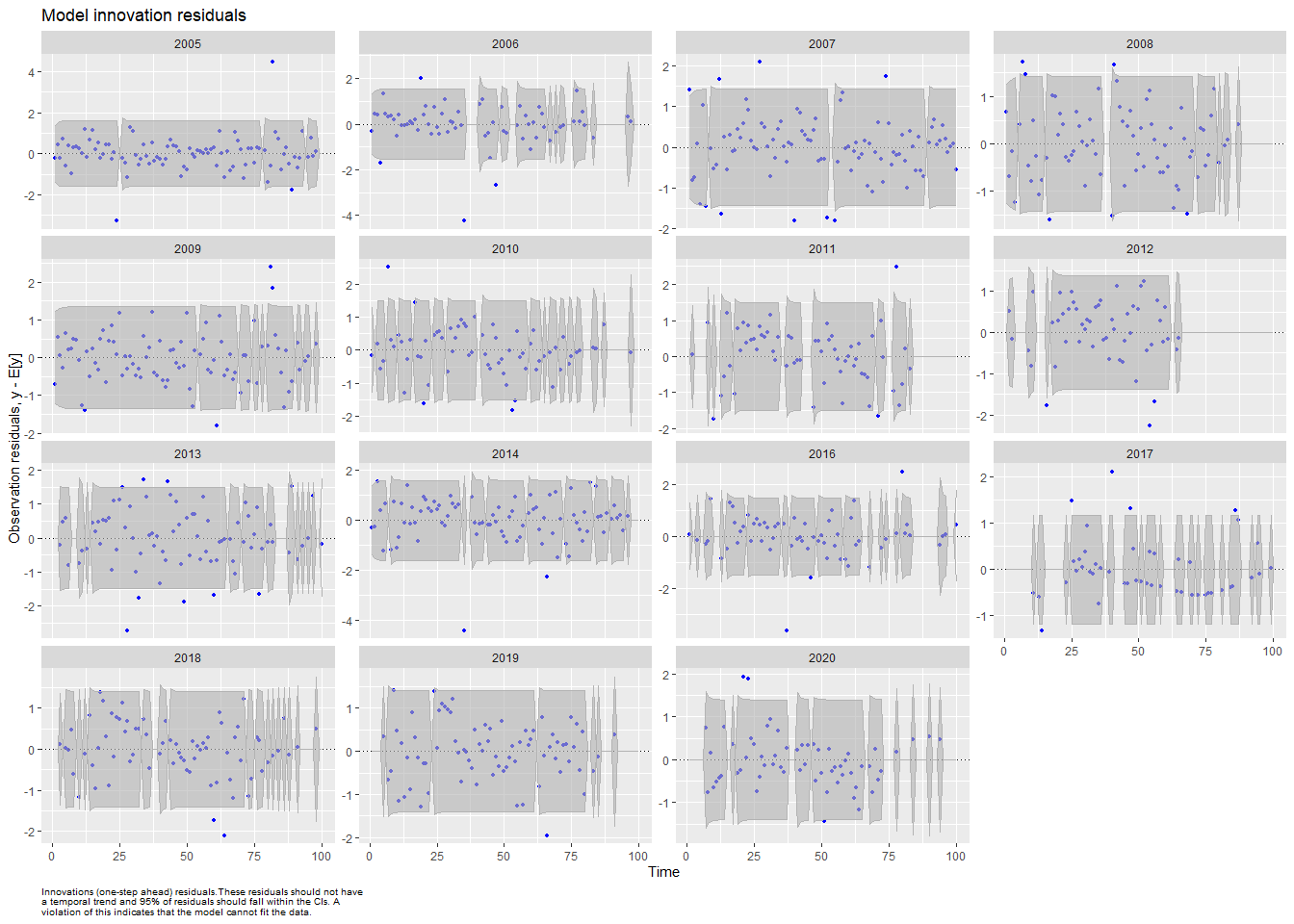
## plot.type = fitted.ytT



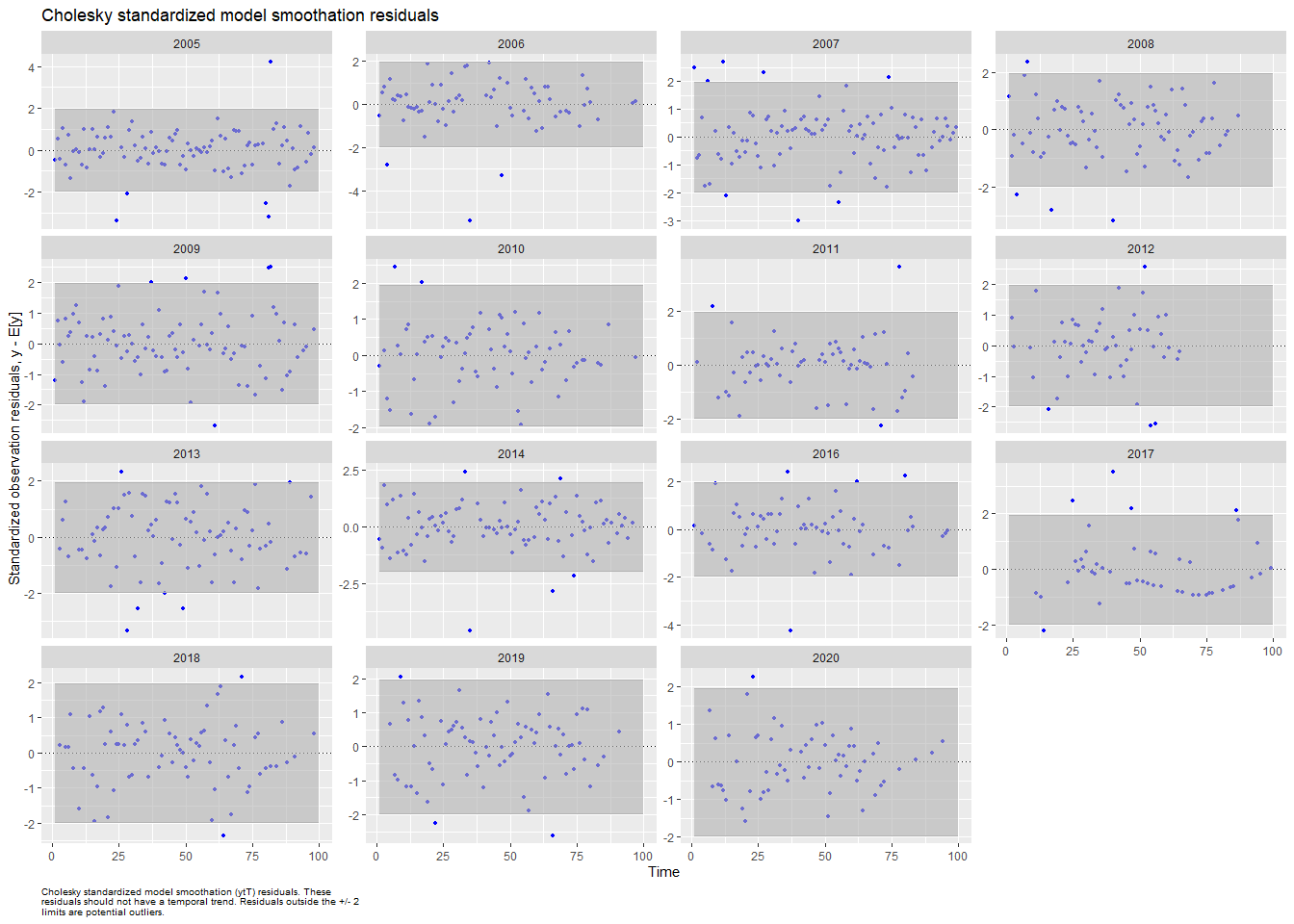
## Hit <Return> to see next plot (q to exit):



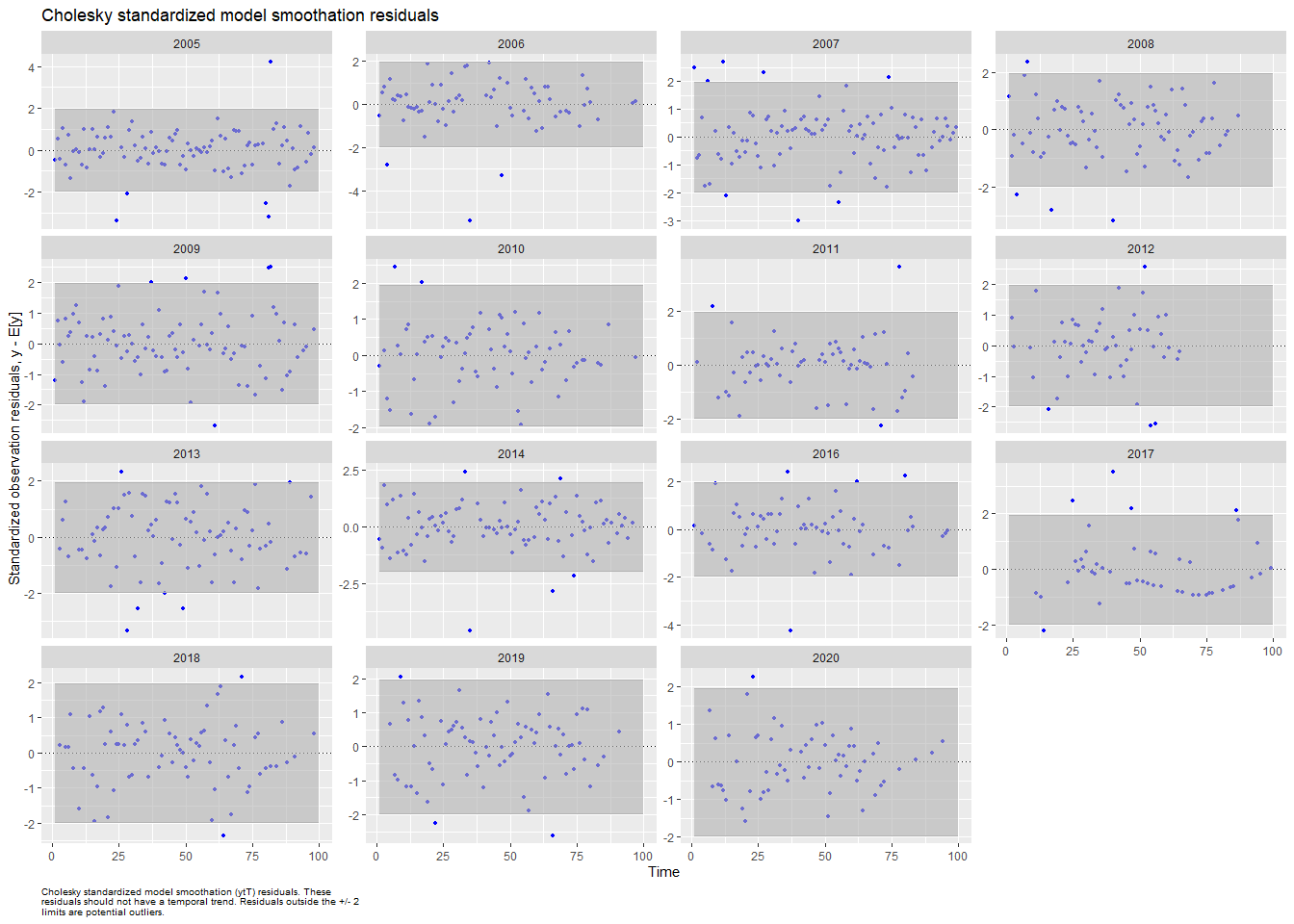
## plot.type = model.resids.ytt1



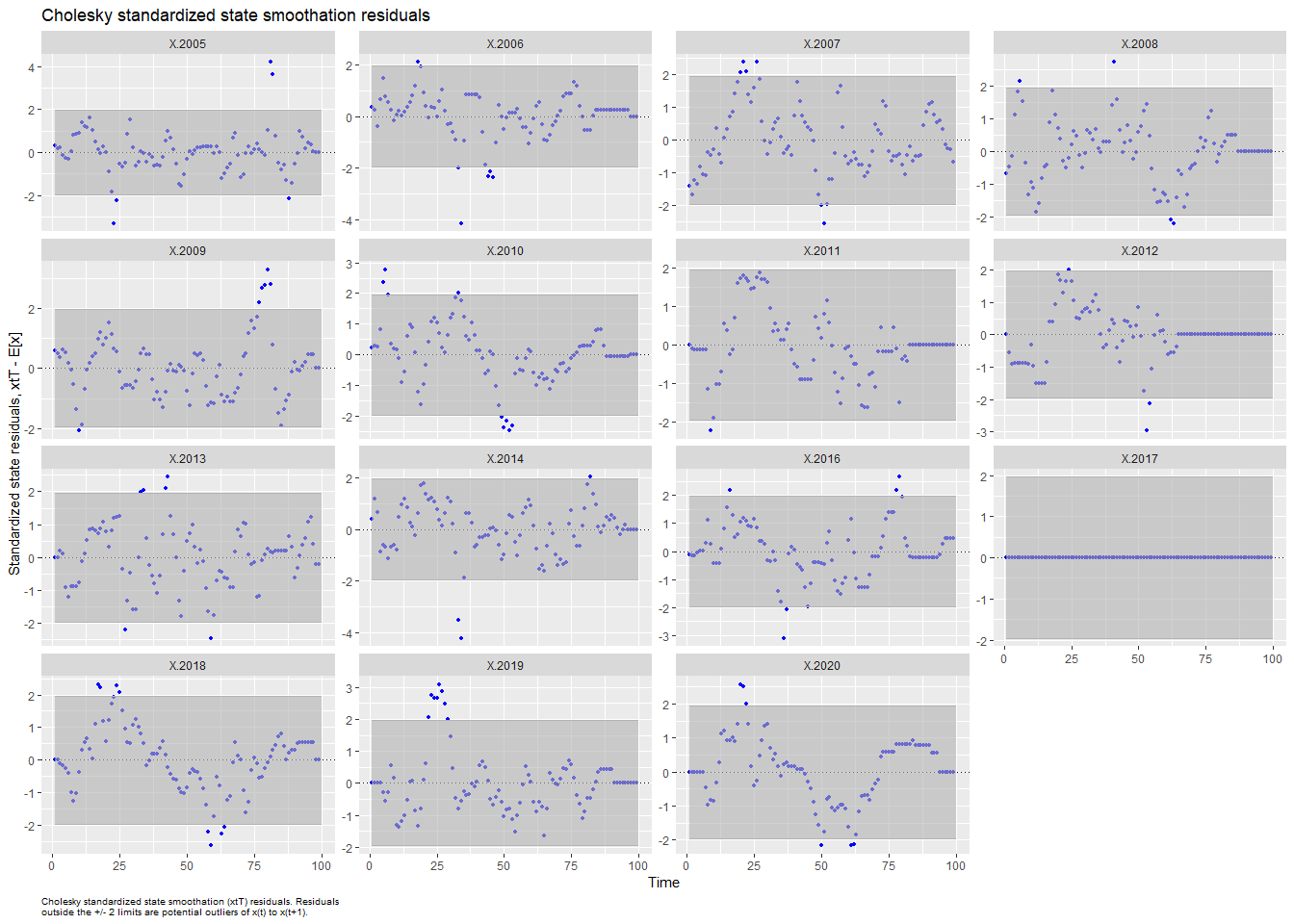
## Hit <Return> to see next plot (q to exit):



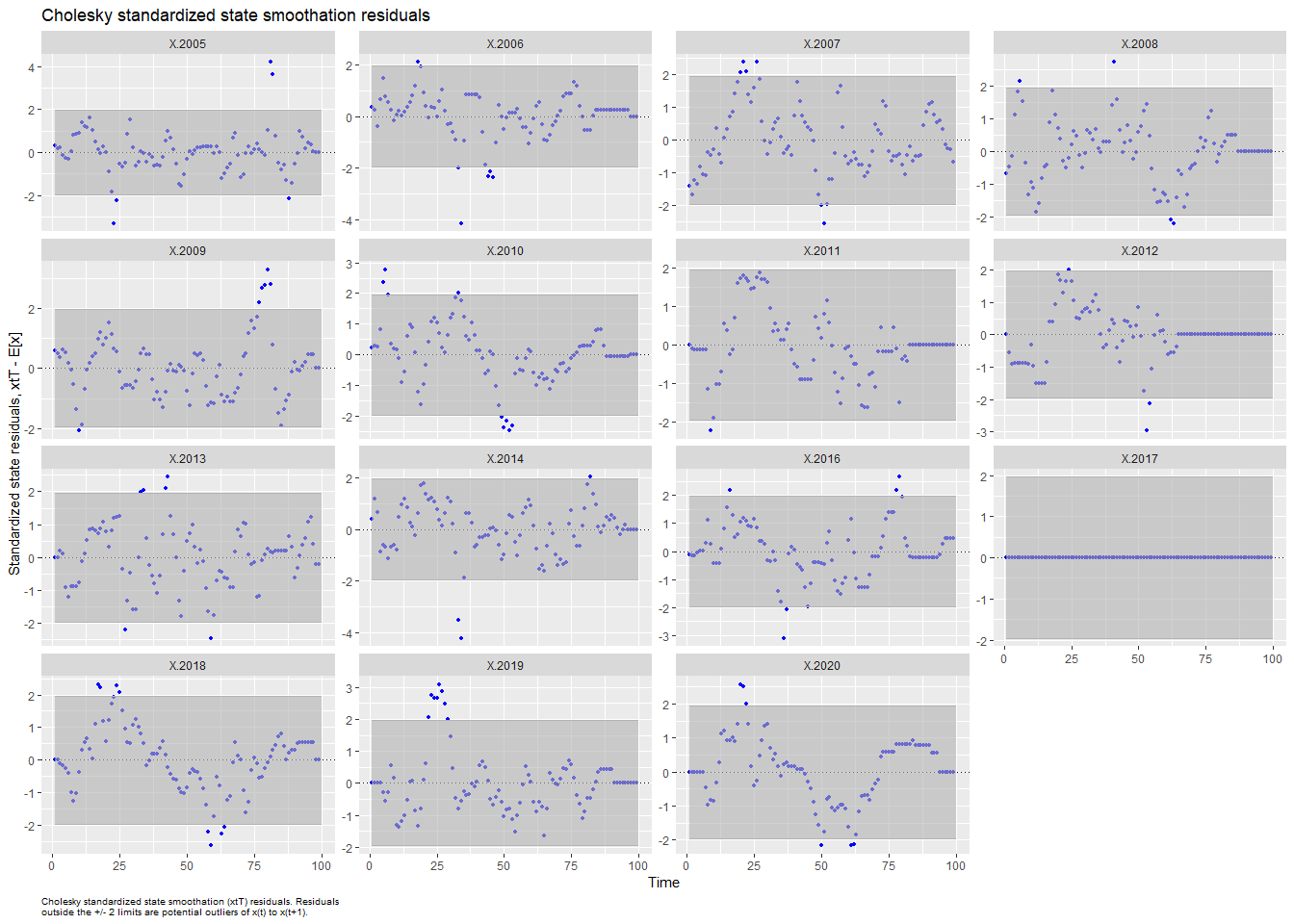
## plot.type = std.model.resids.ytT



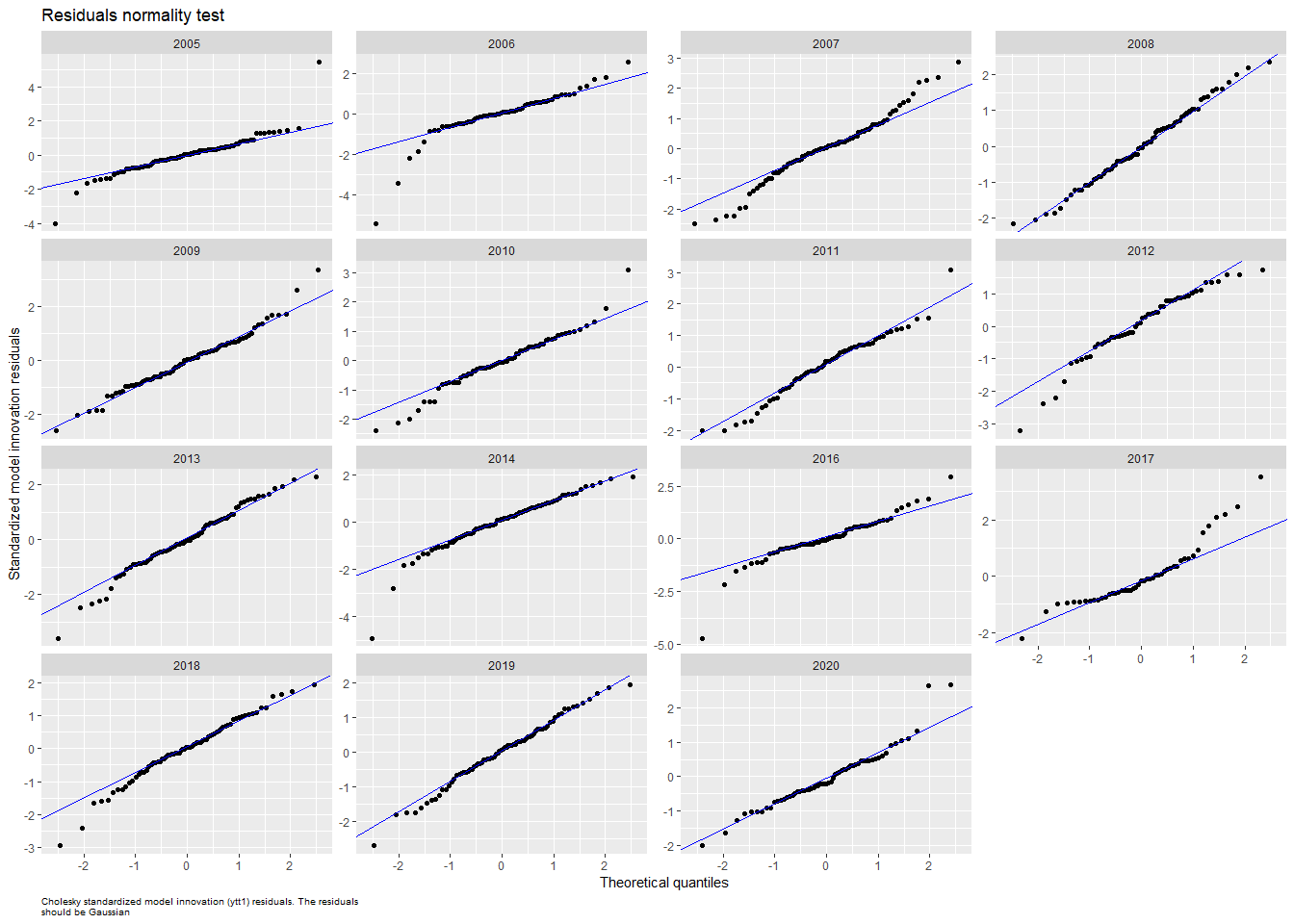
## Hit <Return> to see next plot (q to exit):



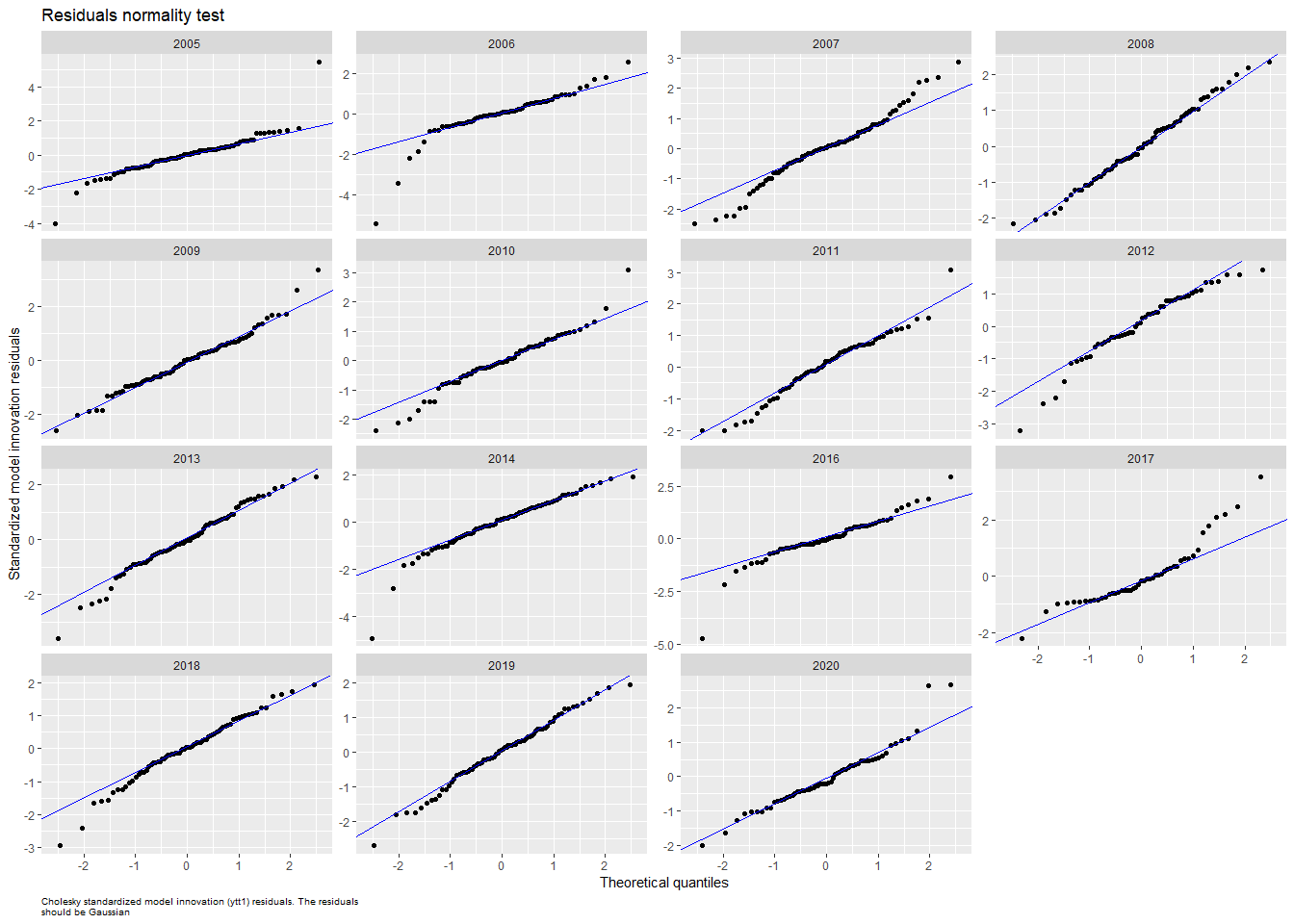
## plot.type = std.state.resids.xtT



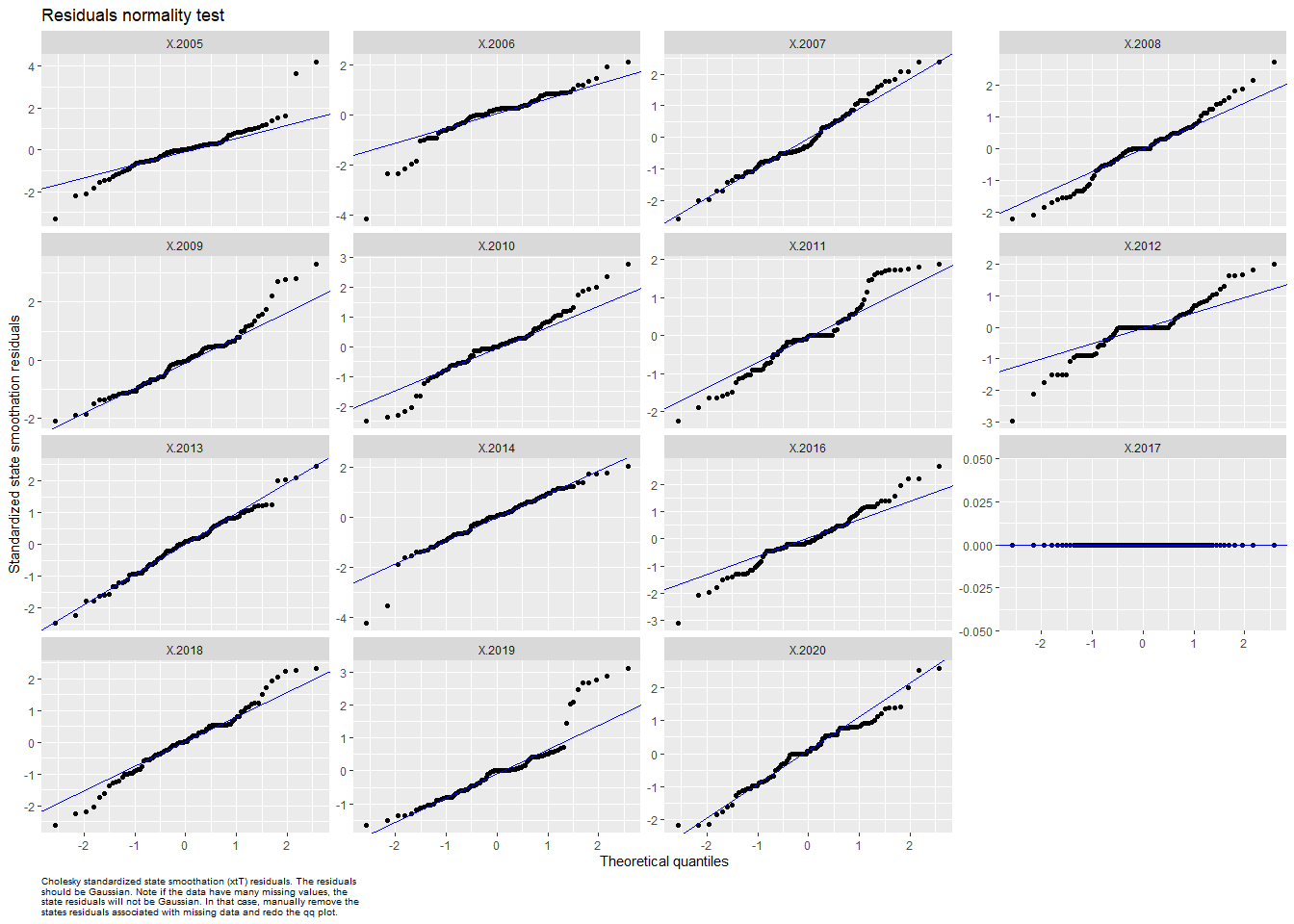
## Hit <Return> to see next plot (q to exit):



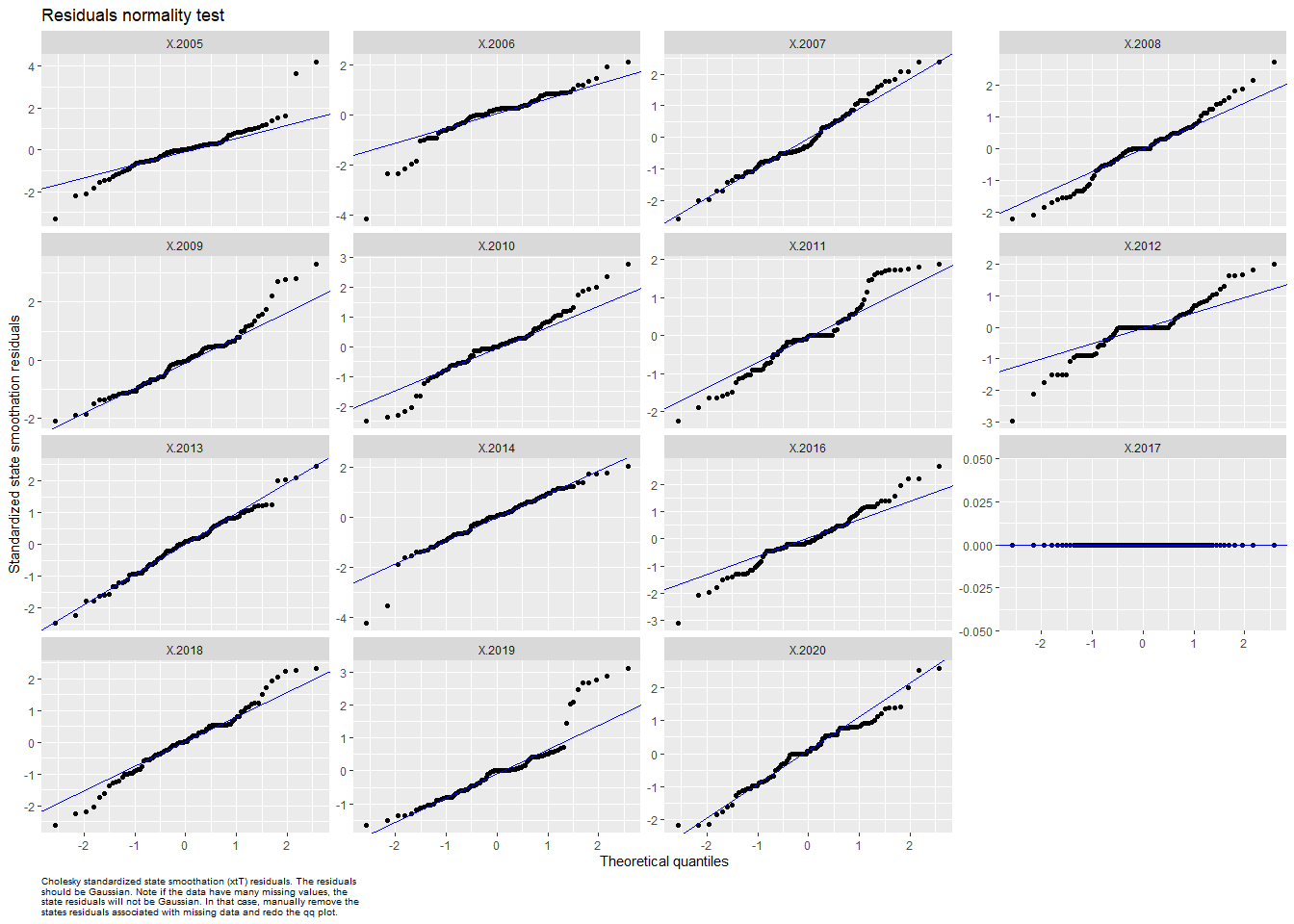
## plot.type = qqplot.std.model.resids.ytt1



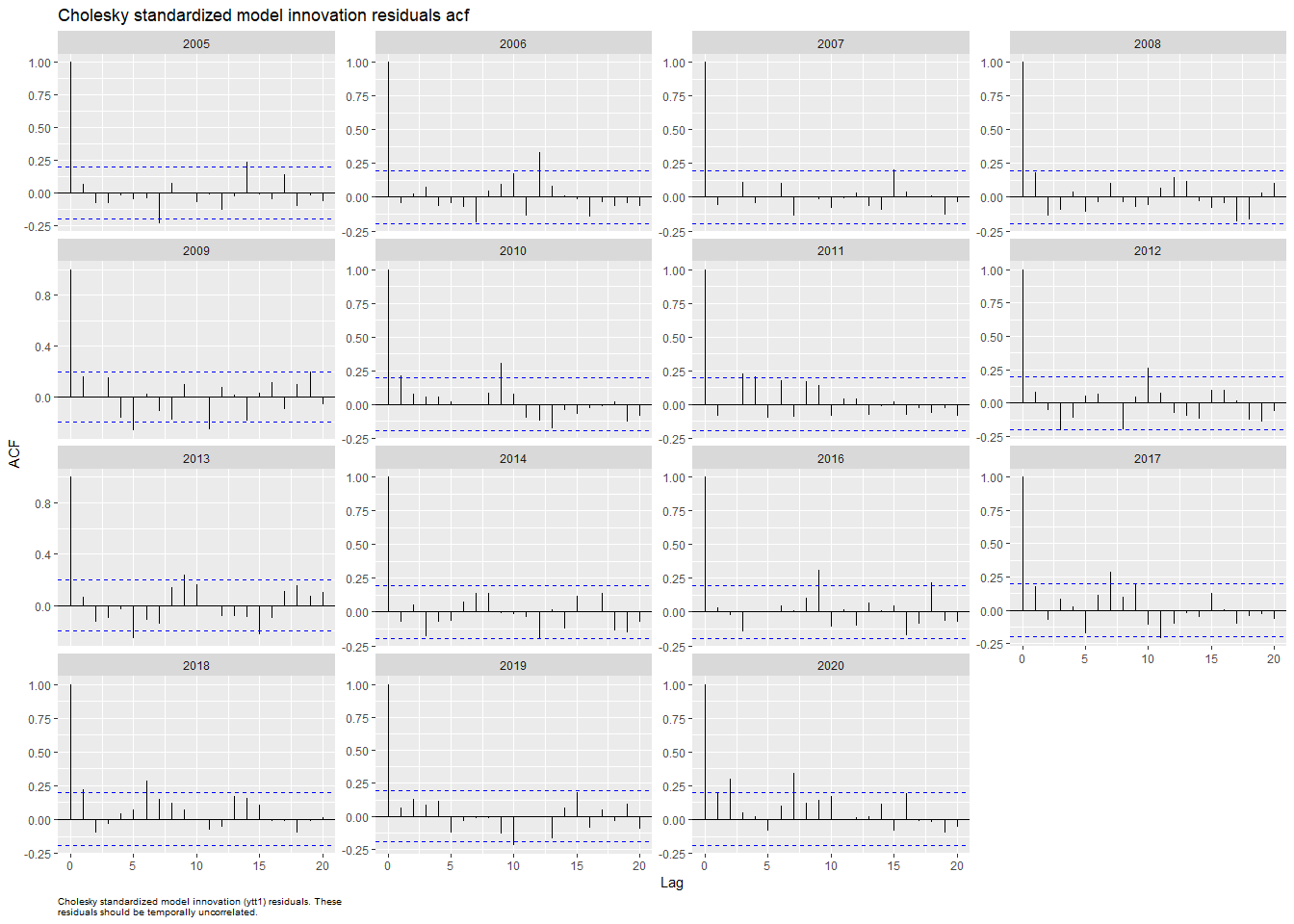
## Hit <Return> to see next plot (q to exit):



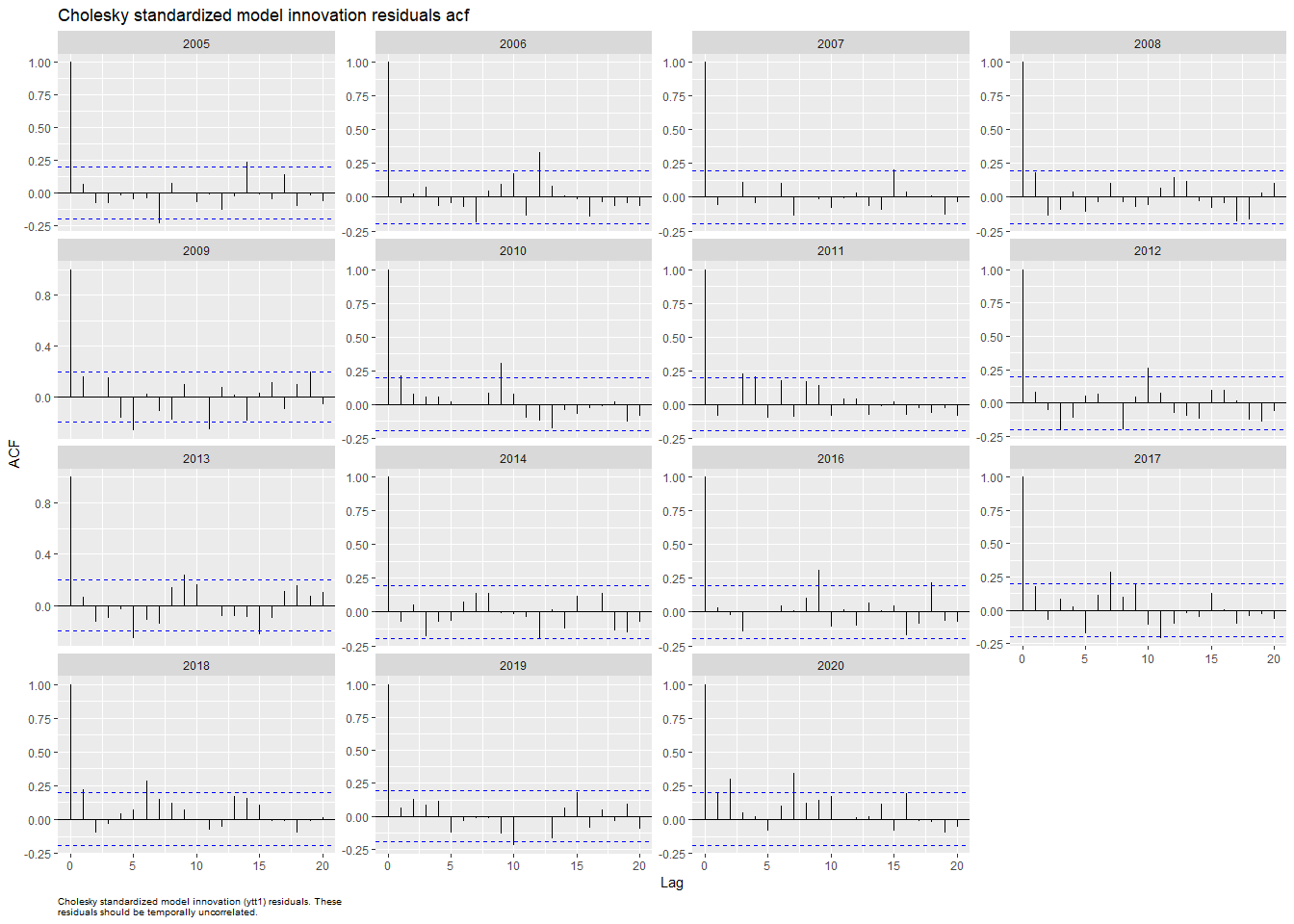
## plot.type = qqplot.std.state.resids.xtT



## Finished plots.



## plot.type = acf.std.model.resids.ytt1



## Hit <Return> to see next plot (q to exit):

Now, I will add flow as a covariate in the model.

covariates\_coho\_hatchery\_temp\_flow\_all\_years <- arrange(data,doy) %>%  
 filter(year != 2015 & doy > 100 & doy <=200) %>%  
 select(coho1\_hatchery\_num\_interpolate,year,doy, temp, flow) %>%  
 pivot\_wider(names\_from = year, values\_from = c(coho1\_hatchery\_num\_interpolate,temp, flow)) %>%  
 column\_to\_rownames(var = "doy") %>%  
 as.matrix() %>%  
 t()

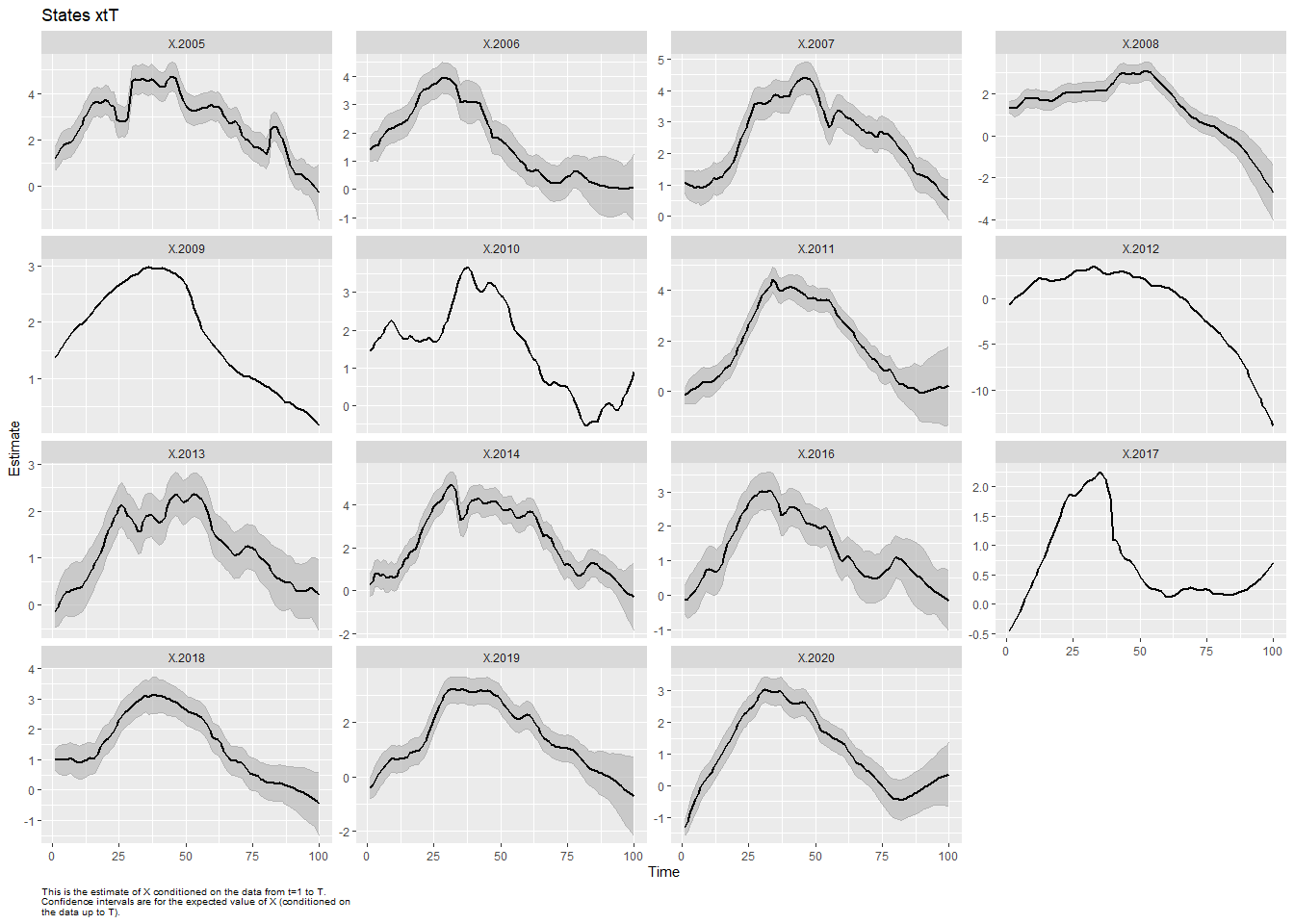
Specifying the model framework:

mod.list.hatchery\_temp\_flow\_all\_years <- list(  
 U = "unequal",  
 R = "diagonal and equal",  
 Q = "diagonal and unequal",  
 C = matrix(list(  
 "h2005",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2005",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"f2005",0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
 0,"h2006",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2006",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"f2006",0,0,0,0,0,0,0,0,0,0,0,0,0,  
 0,0,"h2007",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2007",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"f2007",0,0,0,0,0,0,0,0,0,0,0,0,  
 0,0,0,"h2008",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2008",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"f2008",0,0,0,0,0,0,0,0,0,0,0,  
 0,0,0,0,"h2009",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2009",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"f2009",0,0,0,0,0,0,0,0,0,0,  
 0,0,0,0,0,"h2010",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2010",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"f2010",0,0,0,0,0,0,0,0,0,  
 0,0,0,0,0,0,"h2011",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2011",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"f2011",0,0,0,0,0,0,0,0,  
 0,0,0,0,0,0,0,"h2012",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2012",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"f2012",0,0,0,0,0,0,0,  
 0,0,0,0,0,0,0,0,"h2013",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2013",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"f2013",0,0,0,0,0,0,  
 0,0,0,0,0,0,0,0,0,"h2014",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2014",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"f2014",0,0,0,0,0,  
 0,0,0,0,0,0,0,0,0,0,"h2016",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2016",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"f2016",0,0,0,0,  
 0,0,0,0,0,0,0,0,0,0,0,"h2017",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2017",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"f2017",0,0,0,  
 0,0,0,0,0,0,0,0,0,0,0,0,"h2018",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2018",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"f2018",0,0,  
 0,0,0,0,0,0,0,0,0,0,0,0,0,"h2019",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2019",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"f2019",0,  
 0,0,0,0,0,0,0,0,0,0,0,0,0,0,"h2020",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"t2020",0,0,0,0,0,0,0,0,0,0,0,0,0,0,"f2020"),  
 15,45, byrow = TRUE),  
 c = covariates\_coho\_hatchery\_temp\_flow\_all\_years  
)  
  
fit\_temp\_hatchery\_flow\_all\_years <- MARSS(coho, model = mod.list.hatchery\_temp\_flow\_all\_years, method = "BFGS")

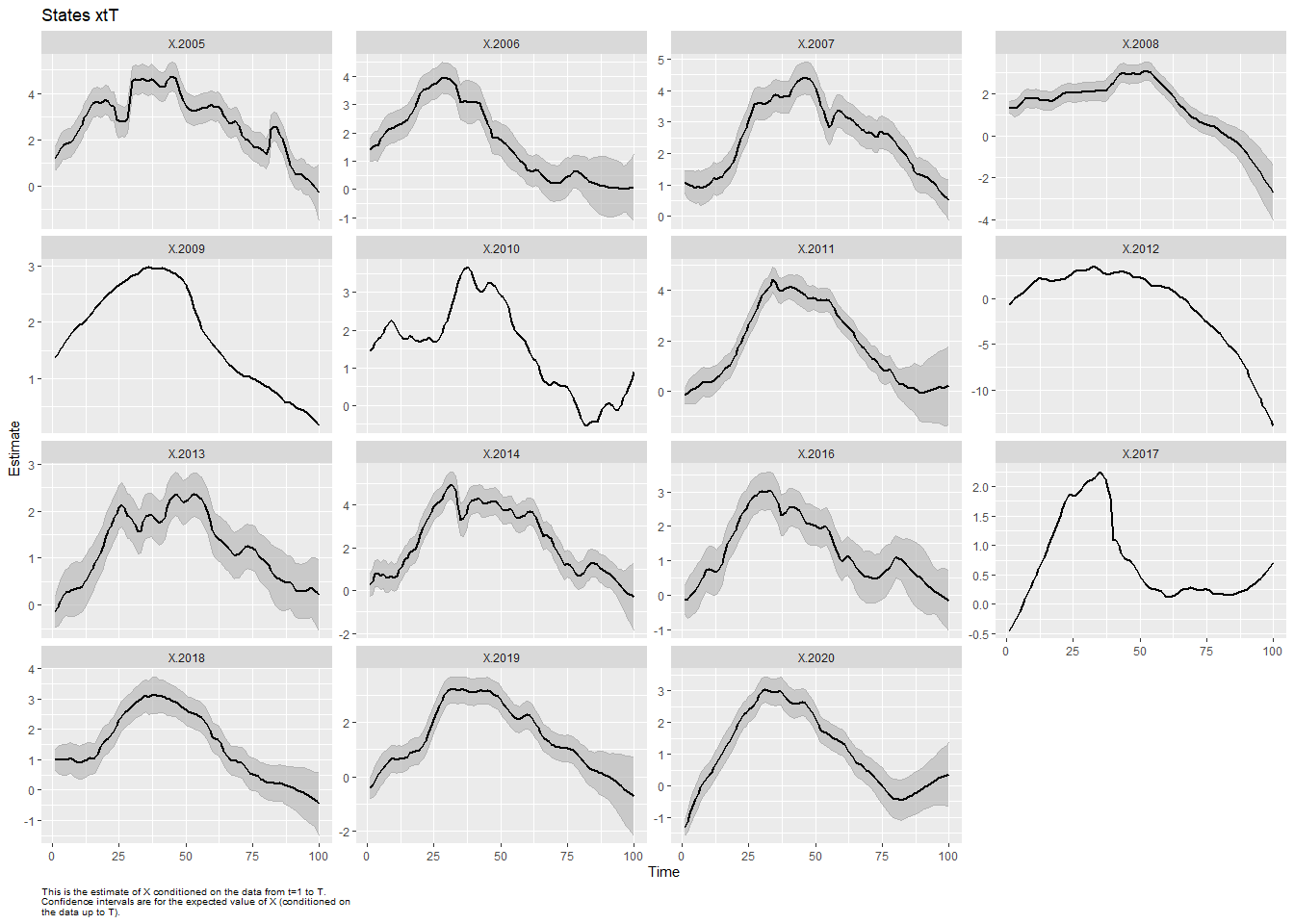
## Success! Converged in 1421 iterations.  
## Function MARSSkfas used for likelihood calculation.  
##   
## MARSS fit is  
## Estimation method: BFGS   
## Estimation converged in 1421 iterations.   
## Log-likelihood: -1200.322   
## AIC: 2582.644 AICc: 2599.421   
##   
## Estimate  
## R.diag 3.96e-01  
## U.X.2005 6.51e-01  
## U.X.2006 1.34e-01  
## U.X.2007 4.49e-01  
## U.X.2008 1.56e-01  
## U.X.2009 2.06e-01  
## U.X.2010 -4.24e-01  
## U.X.2011 -8.08e-02  
## U.X.2012 1.44e+00  
## U.X.2013 2.84e-01  
## U.X.2014 5.07e-01  
## U.X.2016 4.43e-01  
## U.X.2017 2.62e-01  
## U.X.2018 1.57e-02  
## U.X.2019 5.26e-01  
## U.X.2020 5.60e-01  
## Q.(X.2005,X.2005) 1.13e-01  
## Q.(X.2006,X.2006) 6.01e-02  
## Q.(X.2007,X.2007) 4.65e-02  
## Q.(X.2008,X.2008) 2.47e-02  
## Q.(X.2009,X.2009) 2.54e-10  
## Q.(X.2010,X.2010) 1.49e-11  
## Q.(X.2011,X.2011) 3.15e-02  
## Q.(X.2012,X.2012) 1.04e-11  
## Q.(X.2013,X.2013) 3.35e-02  
## Q.(X.2014,X.2014) 1.15e-01  
## Q.(X.2016,X.2016) 5.95e-02  
## Q.(X.2017,X.2017) 4.96e-12  
## Q.(X.2018,X.2018) 3.96e-02  
## Q.(X.2019,X.2019) 3.93e-02  
## Q.(X.2020,X.2020) 1.84e-02  
## x0.X.2005 9.78e-01  
## x0.X.2006 1.35e+00  
## x0.X.2007 1.00e+00  
## x0.X.2008 1.37e+00  
## x0.X.2009 1.31e+00  
## x0.X.2010 1.40e+00  
## x0.X.2011 -2.62e-01  
## x0.X.2012 -8.17e-01  
## x0.X.2013 -1.99e-01  
## x0.X.2014 1.12e-01  
## x0.X.2016 -2.16e-01  
## x0.X.2017 -5.42e-01  
## x0.X.2018 9.94e-01  
## x0.X.2019 -5.52e-01  
## x0.X.2020 -1.54e+00  
## C.h2005 3.95e-04  
## C.h2006 -3.18e-04  
## C.h2007 3.00e-04  
## C.h2008 -9.56e-05  
## C.h2009 -4.79e-05  
## C.h2010 -7.34e-05  
## C.h2011 -1.22e-04  
## C.h2012 2.61e-04  
## C.h2013 -1.89e-05  
## C.h2014 4.57e-04  
## C.h2016 4.55e-05  
## C.h2017 -5.08e-04  
## C.h2018 -2.20e-04  
## C.h2019 -1.15e-05  
## C.h2020 -8.50e-05  
## C.t2005 -4.73e-02  
## C.t2006 -6.17e-03  
## C.t2007 -2.28e-02  
## C.t2008 -3.21e-02  
## C.t2009 -1.61e-02  
## C.t2010 9.99e-02  
## C.t2011 5.19e-02  
## C.t2012 -1.55e-01  
## C.t2013 -1.99e-02  
## C.t2014 -3.12e-02  
## C.t2016 -2.78e-02  
## C.t2017 -1.01e-02  
## C.t2018 -9.30e-03  
## C.t2019 -3.99e-02  
## C.t2020 -2.97e-02  
## C.f2005 -7.21e-04  
## C.f2006 -1.34e-04  
## C.f2007 -5.29e-04  
## C.f2008 1.38e-04  
## C.f2009 -1.53e-04  
## C.f2010 -6.05e-04  
## C.f2011 -4.17e-04  
## C.f2012 -5.12e-04  
## C.f2013 -1.72e-04  
## C.f2014 -4.53e-04  
## C.f2016 -2.32e-04  
## C.f2017 -2.03e-04  
## C.f2018 1.27e-04  
## C.f2019 -2.54e-04  
## C.f2020 -4.48e-04  
## Initial states (x0) defined at t=0  
##   
## Standard errors have not been calculated.   
## Use MARSSparamCIs to compute CIs and bias estimates.

Plotting model diagnostics.

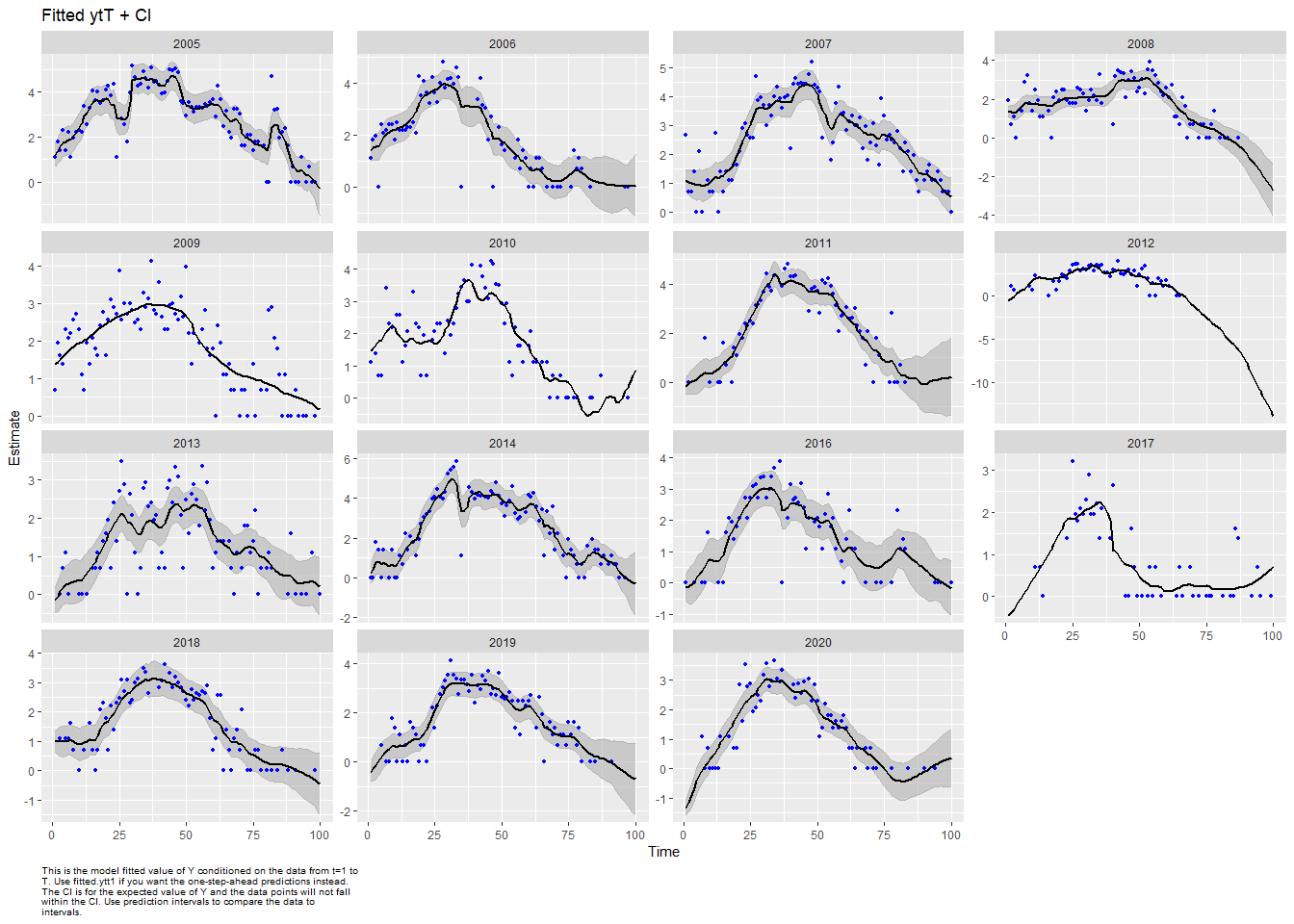
autoplot(fit\_temp\_hatchery\_flow\_all\_years)



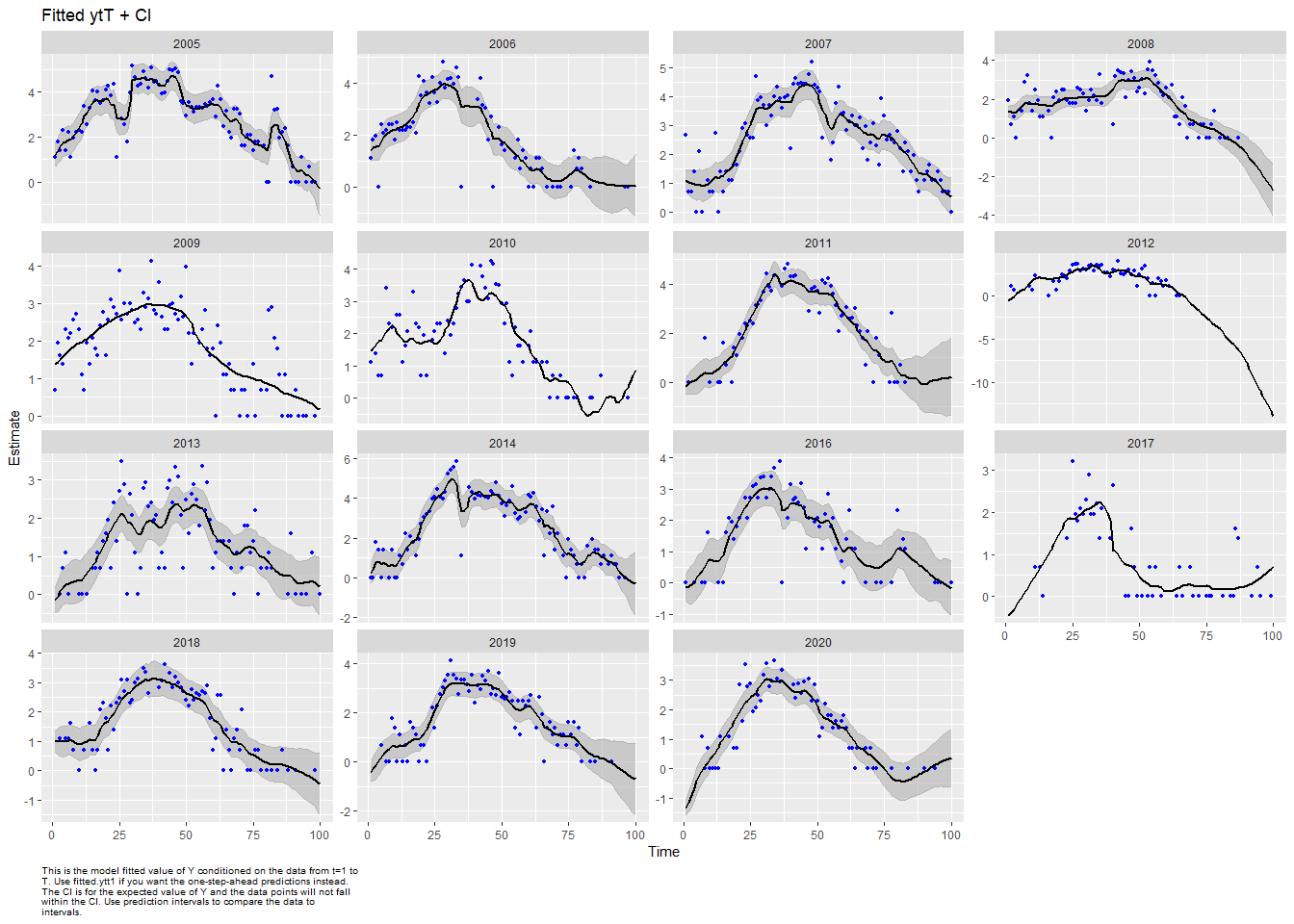
## plot.type = xtT



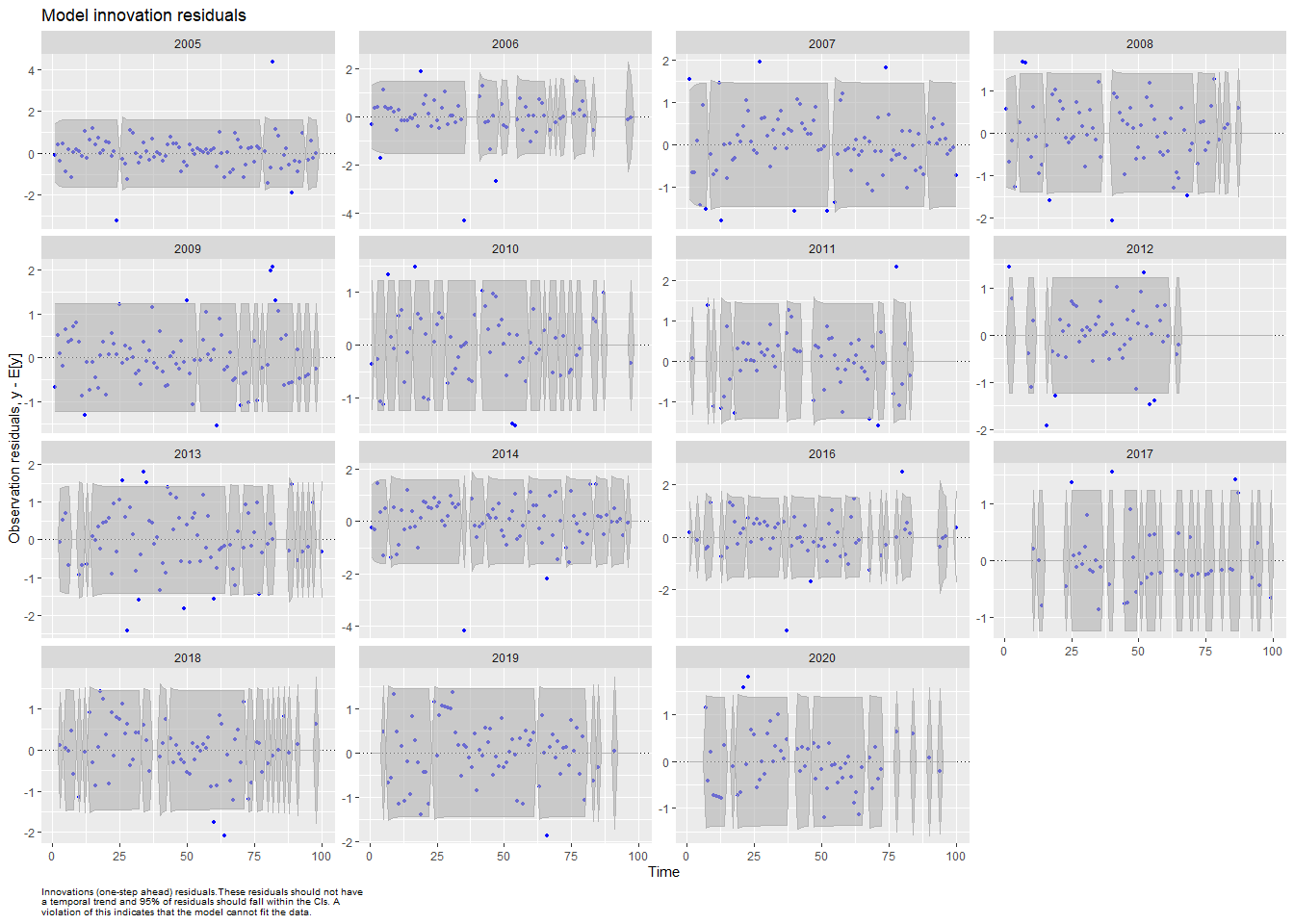
## Hit <Return> to see next plot (q to exit):



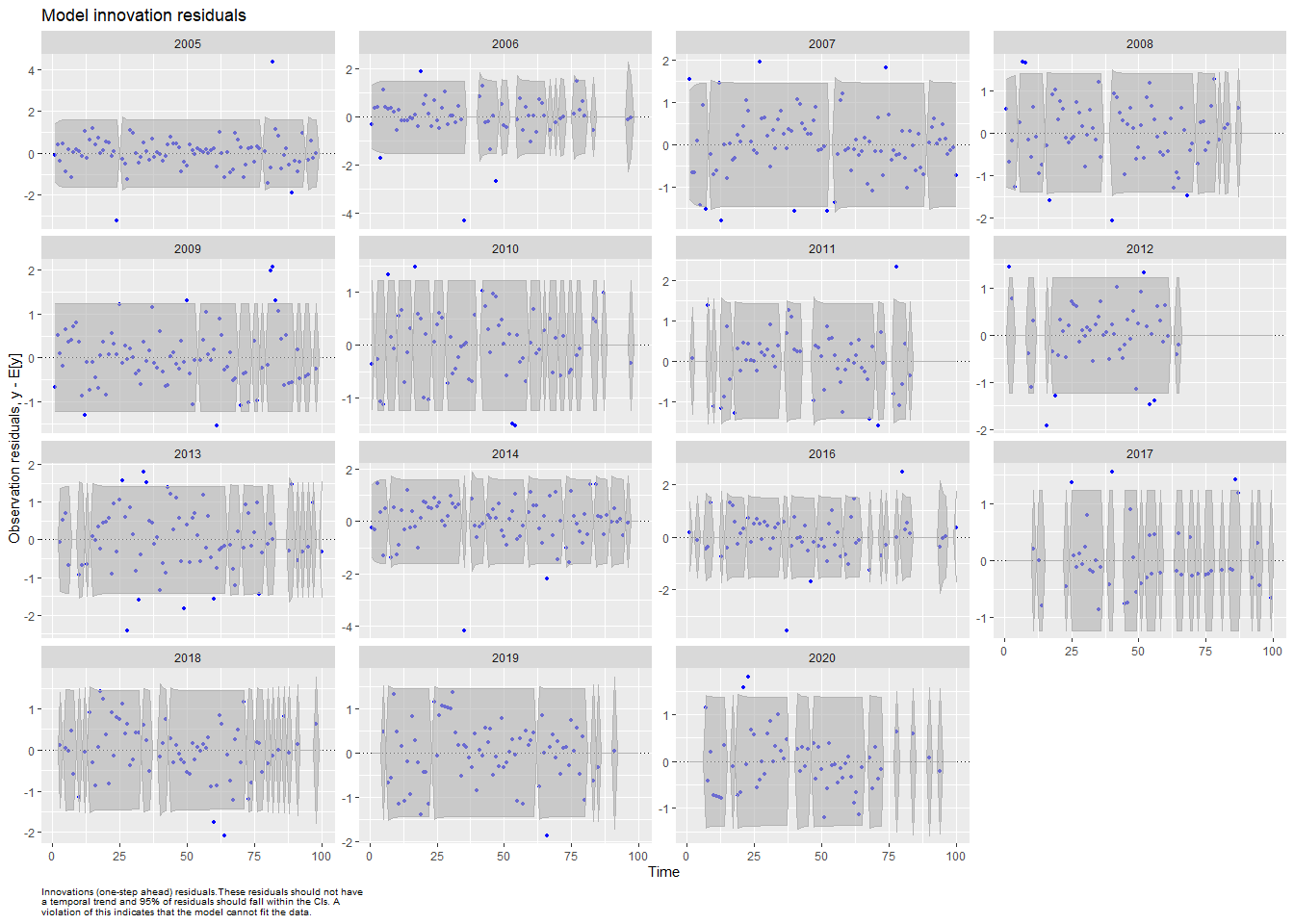
## plot.type = fitted.ytT



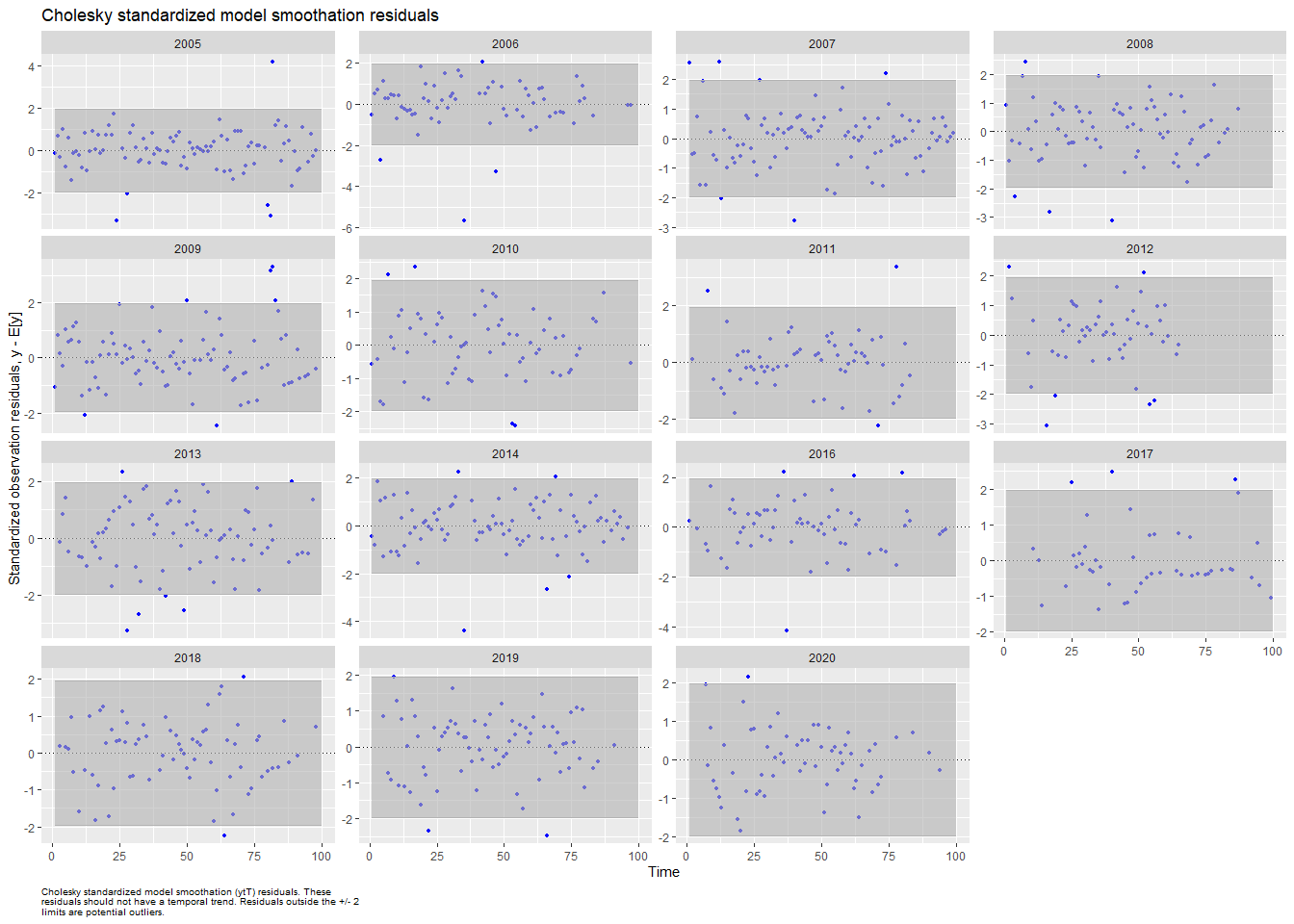
## Hit <Return> to see next plot (q to exit):



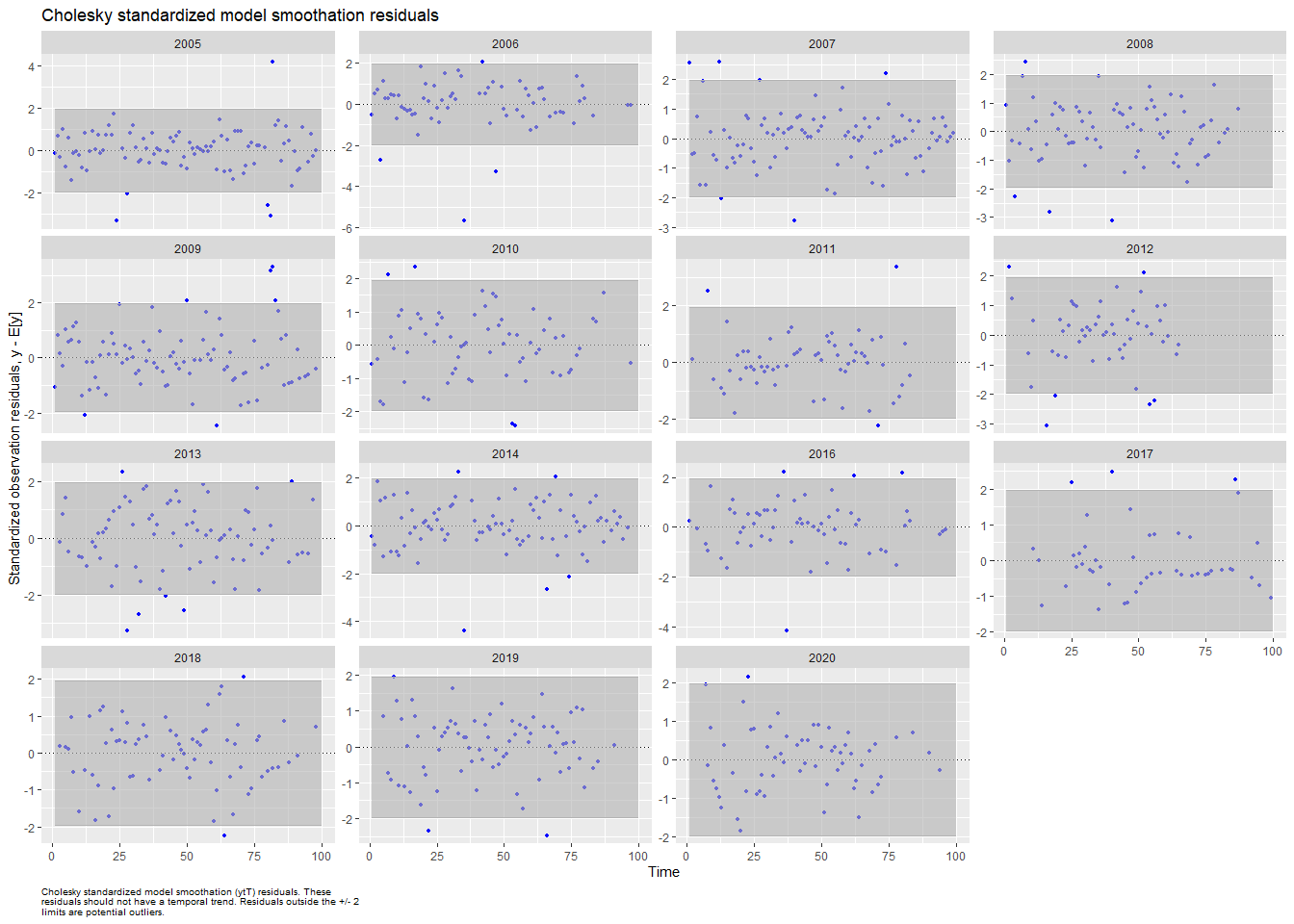
## plot.type = model.resids.ytt1



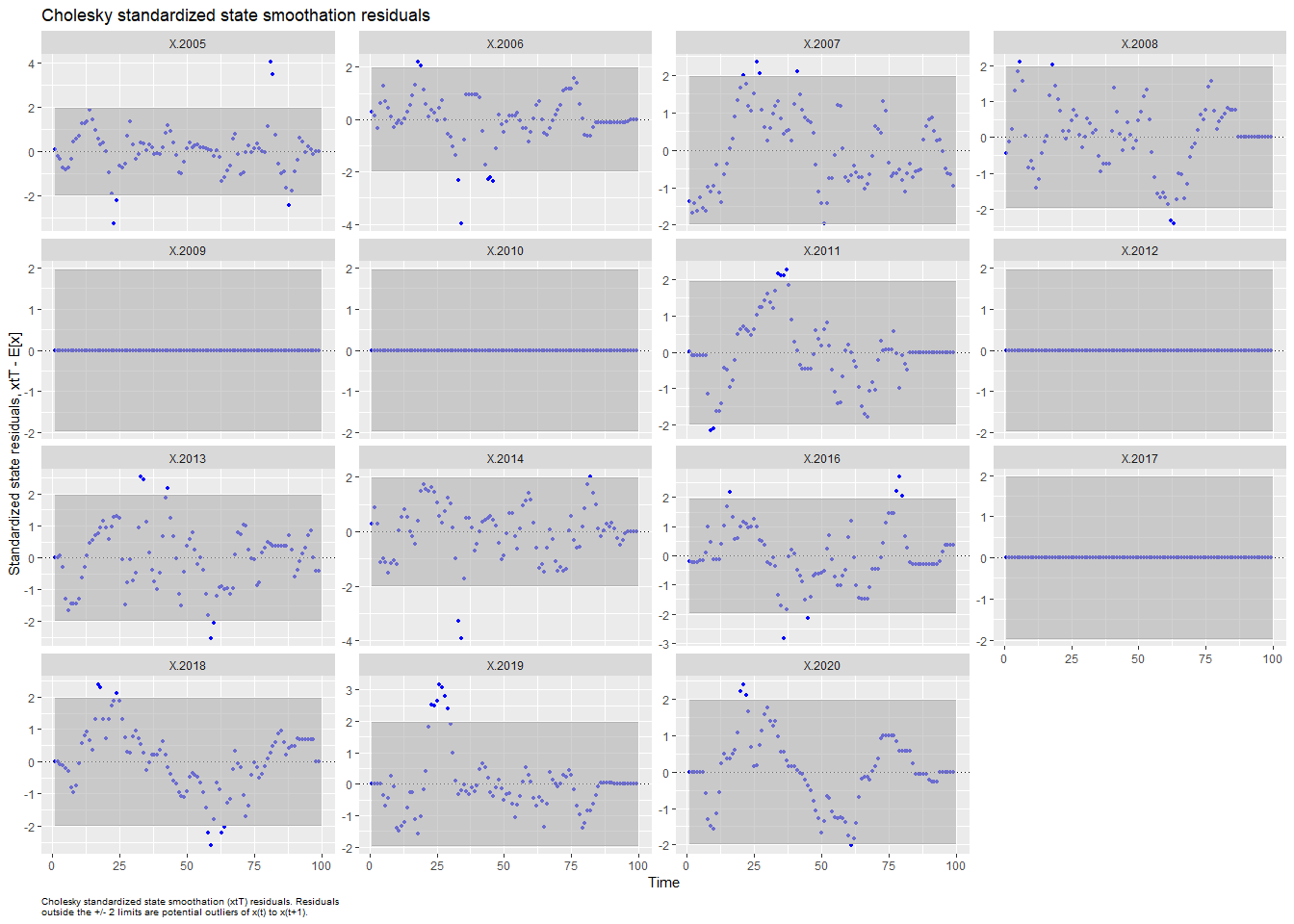
## Hit <Return> to see next plot (q to exit):



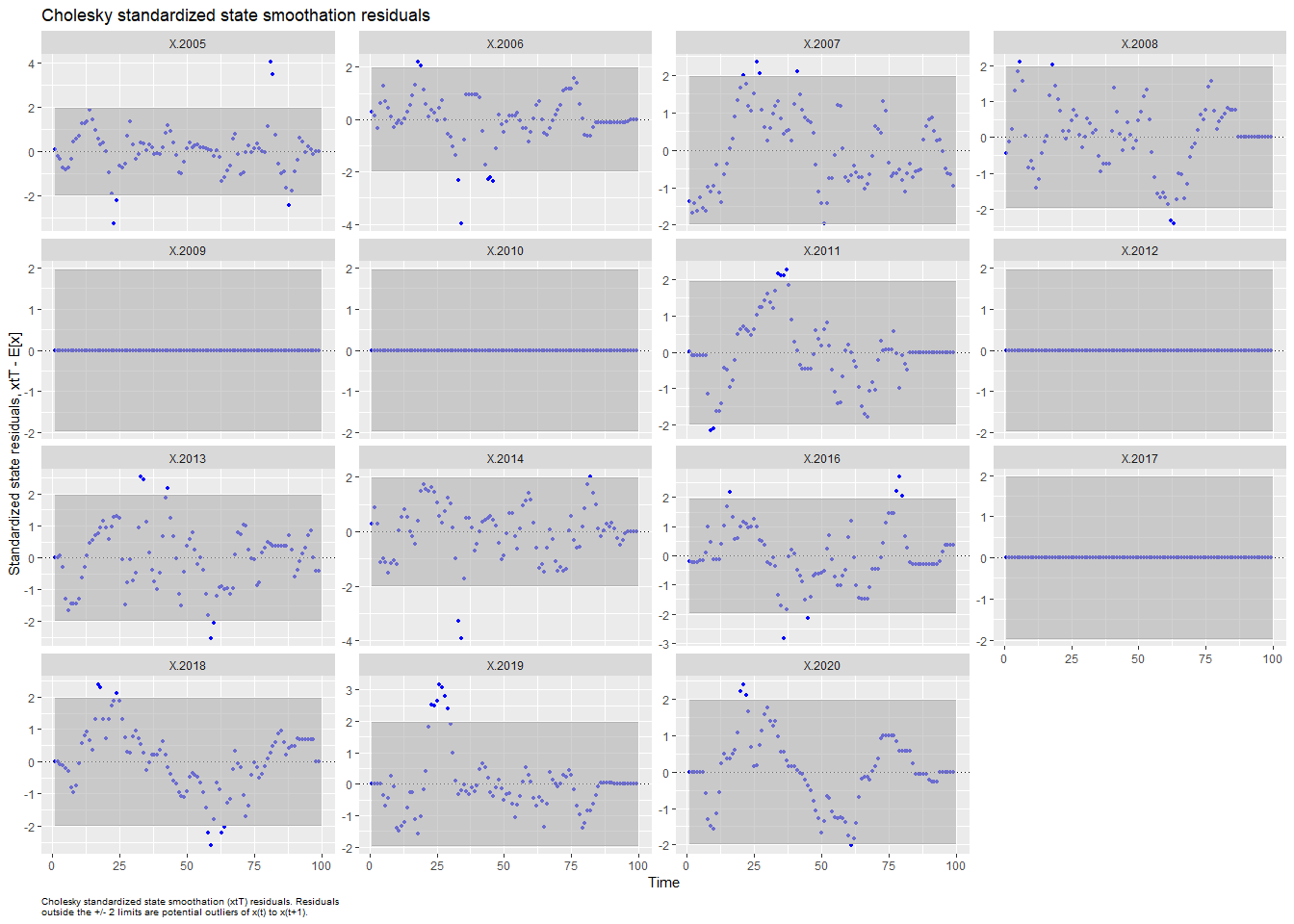
## plot.type = std.model.resids.ytT



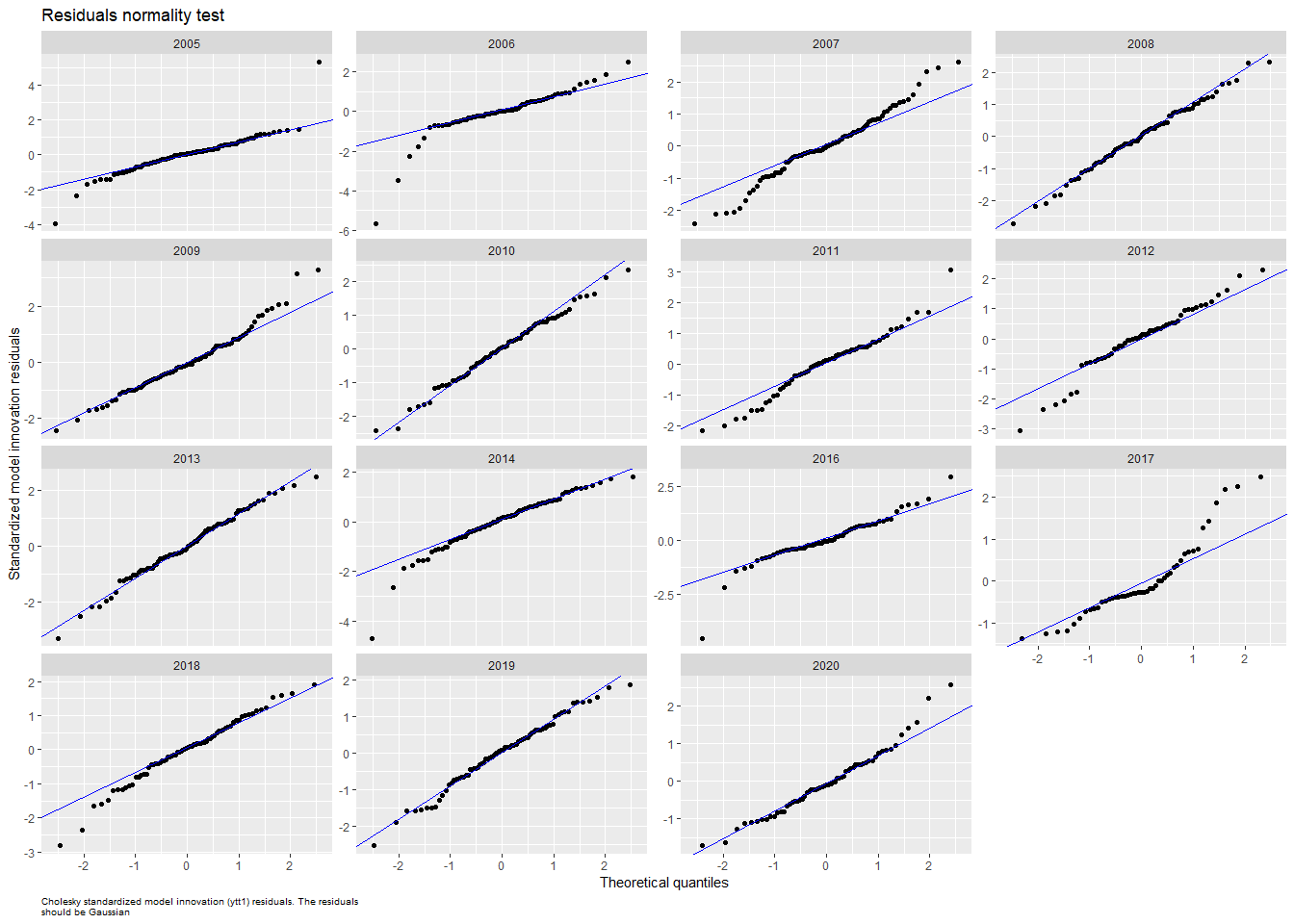
## Hit <Return> to see next plot (q to exit):



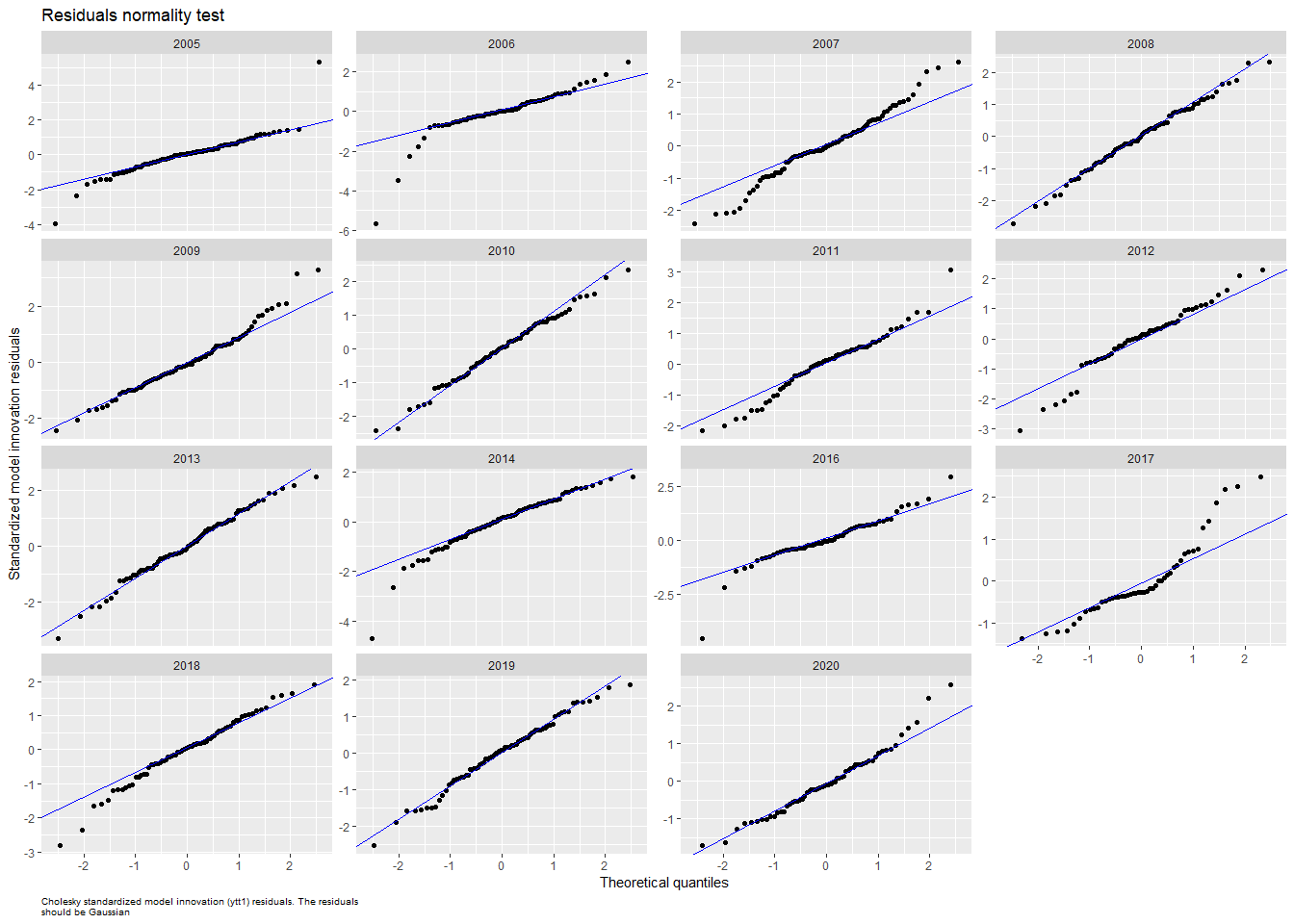
## plot.type = std.state.resids.xtT



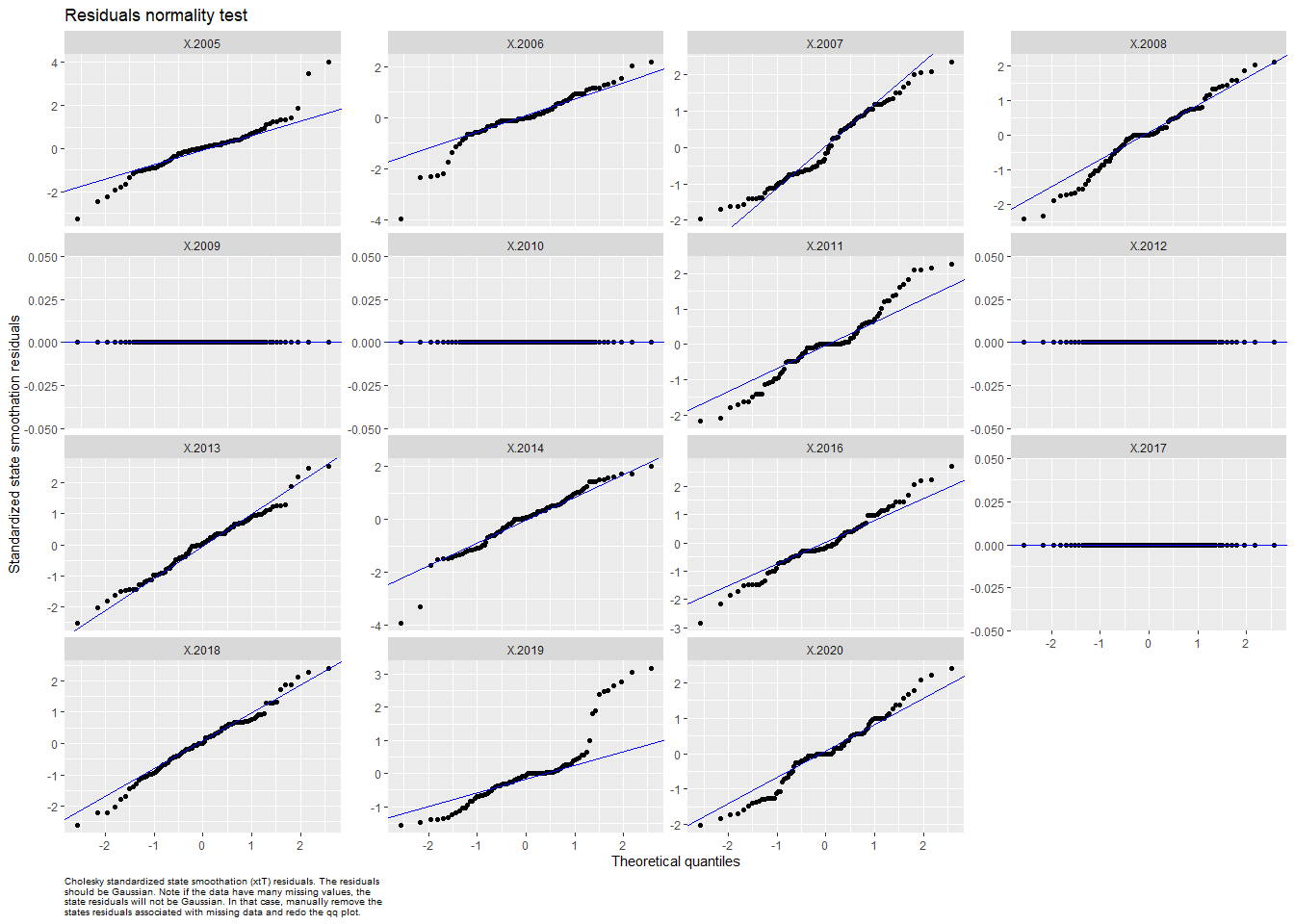
## Hit <Return> to see next plot (q to exit):



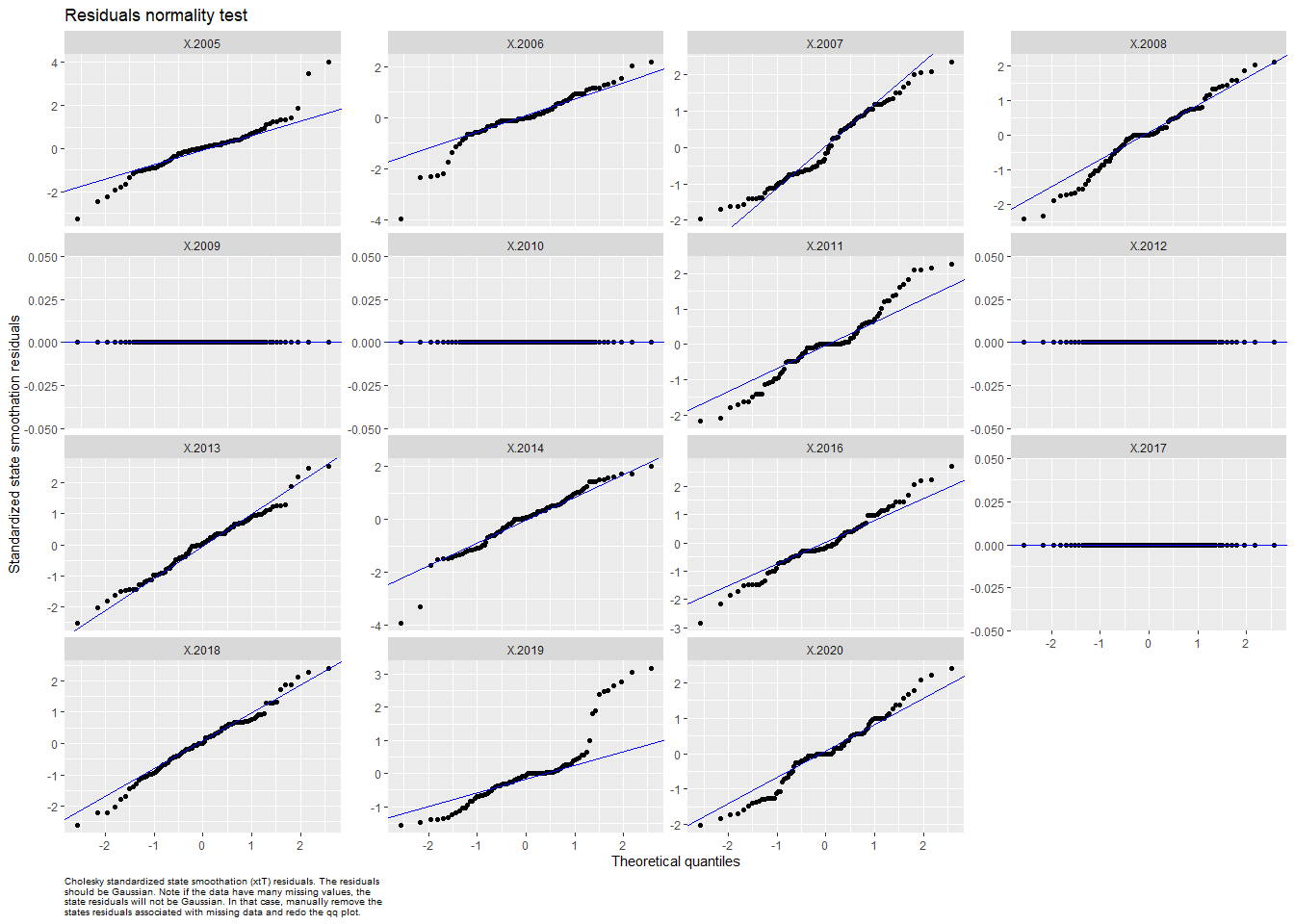
## plot.type = qqplot.std.model.resids.ytt1



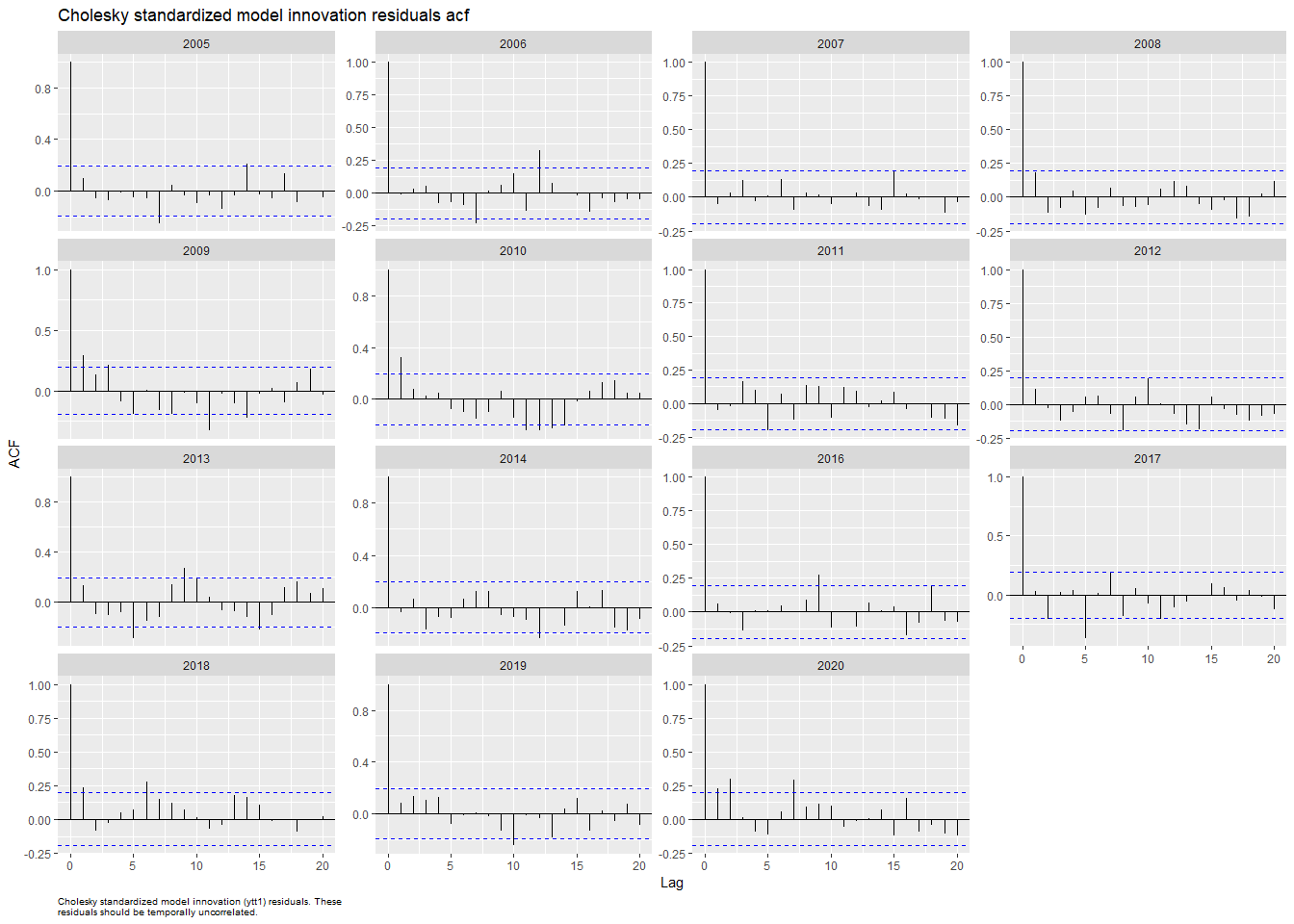
## Hit <Return> to see next plot (q to exit):



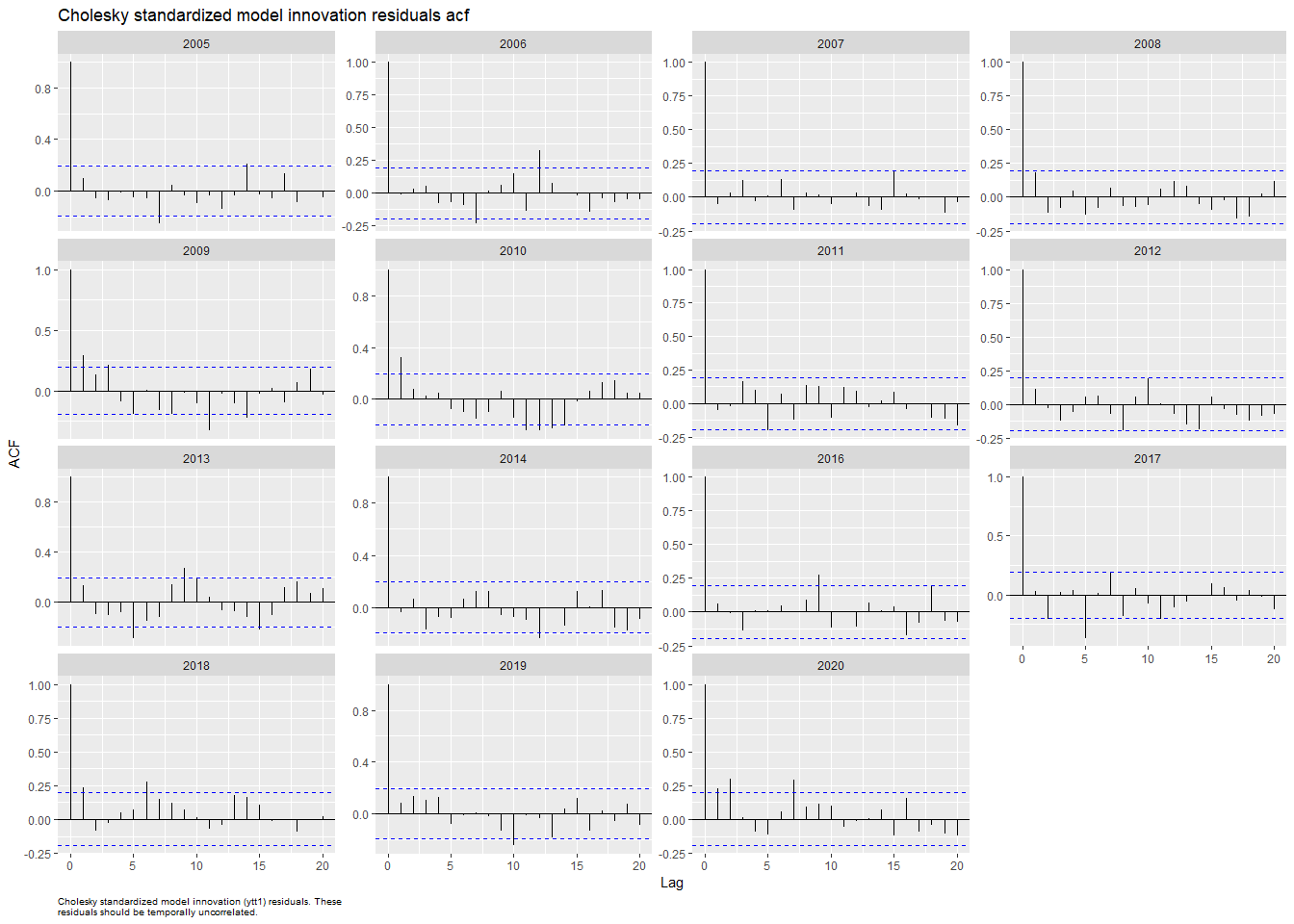
## plot.type = qqplot.std.state.resids.xtT



## Finished plots.



## plot.type = acf.std.model.resids.ytt1



## Hit <Return> to see next plot (q to exit):

Hinrichsen, Richard, and Elizabeth Holmes. 2009. “Using Multivariate State-Space Models to Study Spatial Structure and Dynamics,” August. <https://doi.org/10.1201/9781420059861.ch8>.