Methods

Maria Kuruvilla

2023-05-11

## 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 have taken the sum of the all the salmon smolts migrating on a given say 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. 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 for the years 2005-2020 from the Washington Department of Ecology.

#### Accumulated thermal units

We calculated accumulated thermal units as the cumulative sum of the temperature experienced since Winter Solstice.

## Analysis

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()

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.

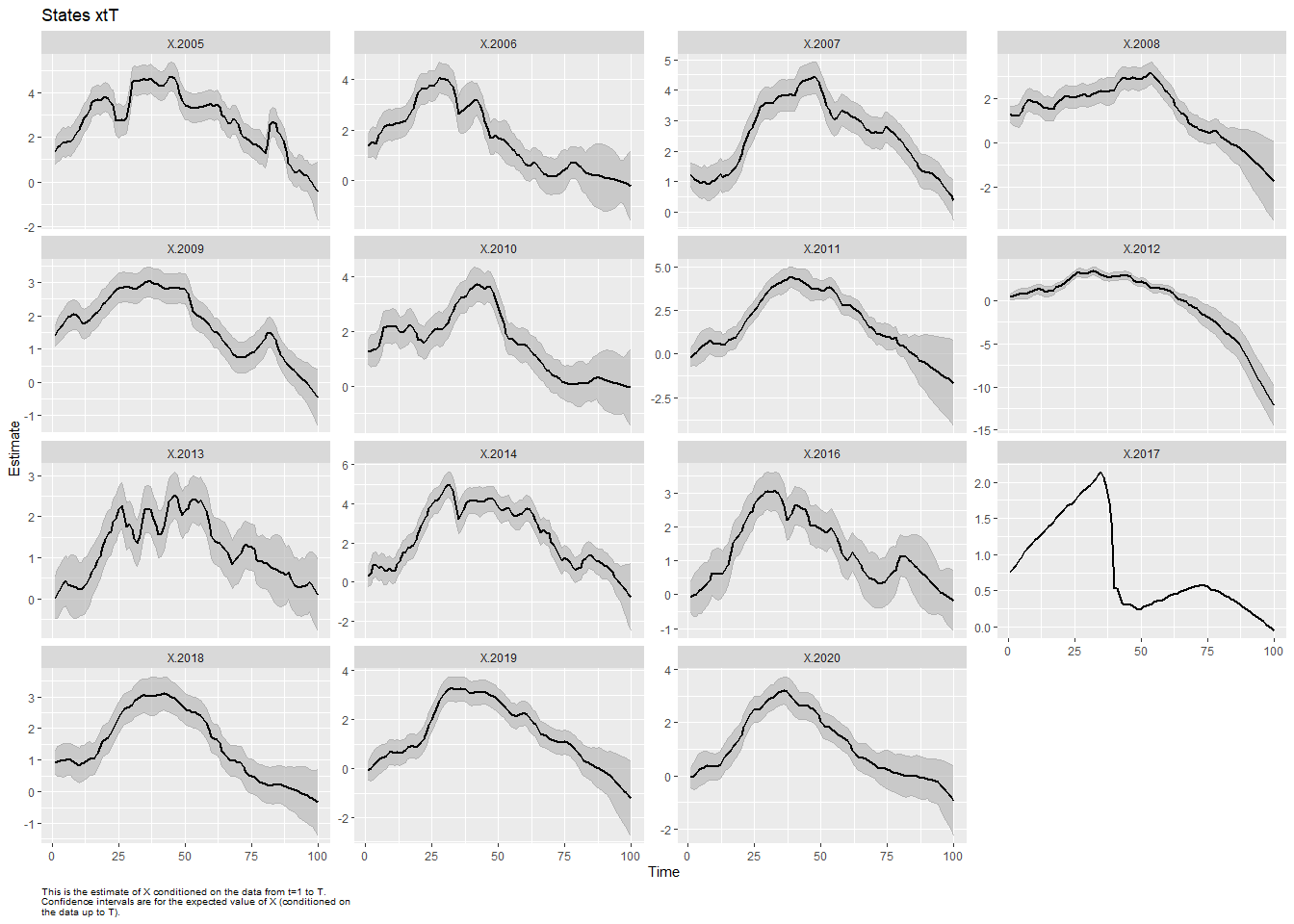
data$coho1\_hatchery\_num\_interpolate <- na.approx(data$coho1\_hatchery\_num)  
  
covariates\_coho\_hatchery\_temp\_all\_years <- arrange(data,doy) %>%  
 filter(year != 2015 & doy > 100 & doy <=200) %>%  
 select(coho1\_hatchery\_num\_interpolate,year,doy, temp) %>%  
 pivot\_wider(names\_from = year,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

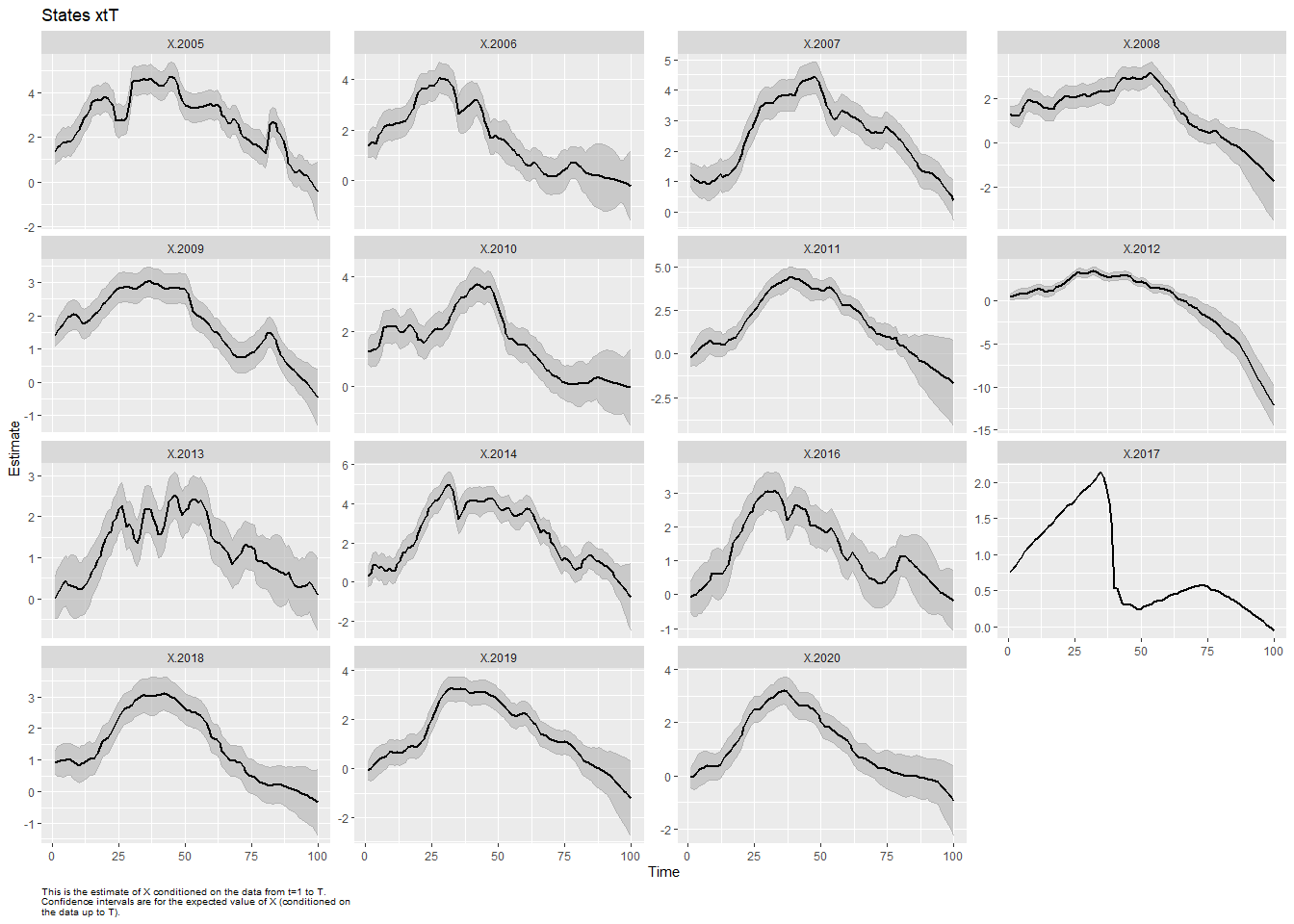
## Success! Converged in 955 iterations.  
## Function MARSSkfas used for likelihood calculation.  
##   
## MARSS fit is  
## Estimation method: BFGS   
## Estimation converged in 955 iterations.   
## Log-likelihood: -1235.898   
## AIC: 2623.796 AICc: 2635.35   
##   
## Estimate  
## R.diag 3.61e-01  
## 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.

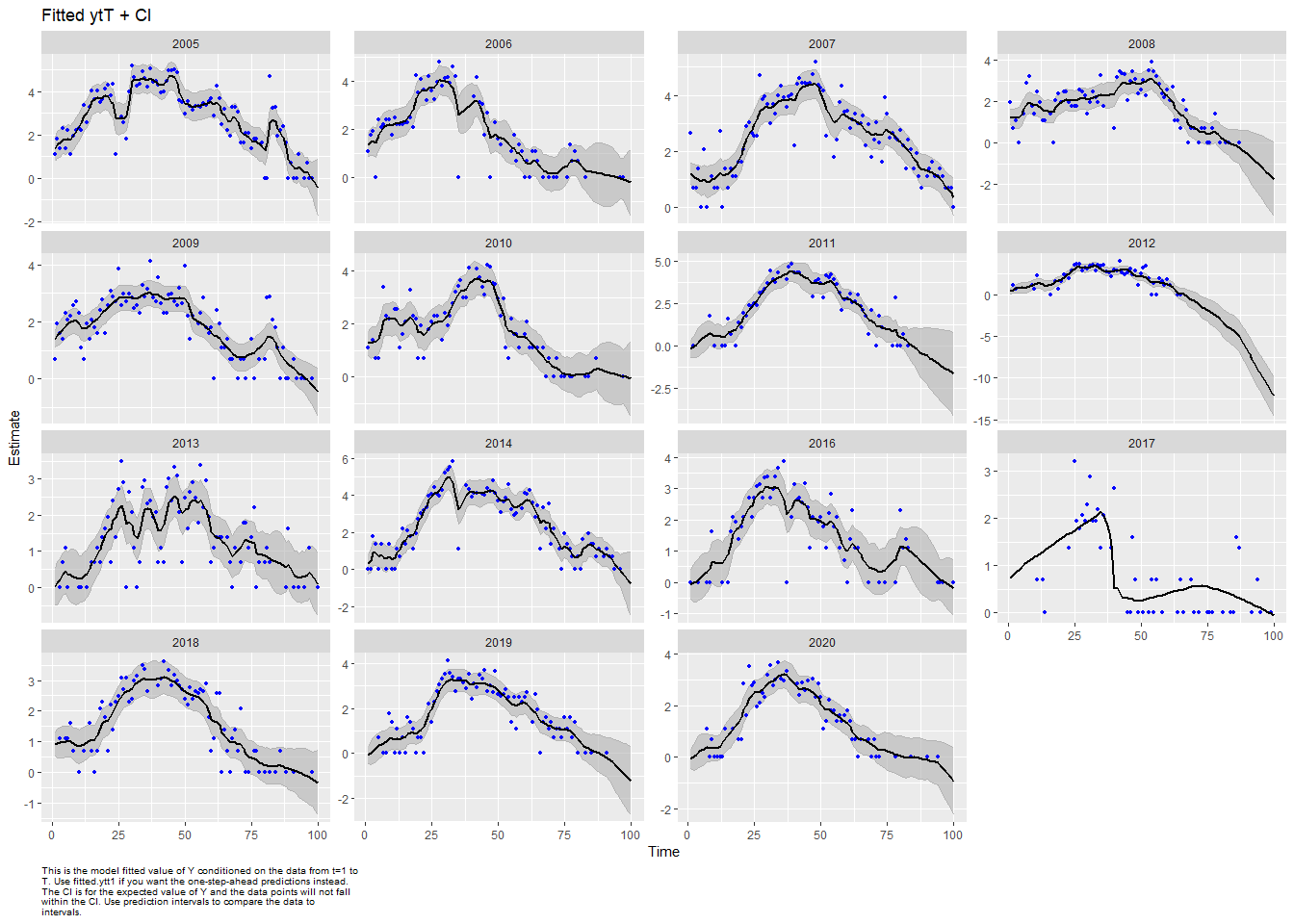
autoplot(fit\_temp\_hatchery\_all\_years)



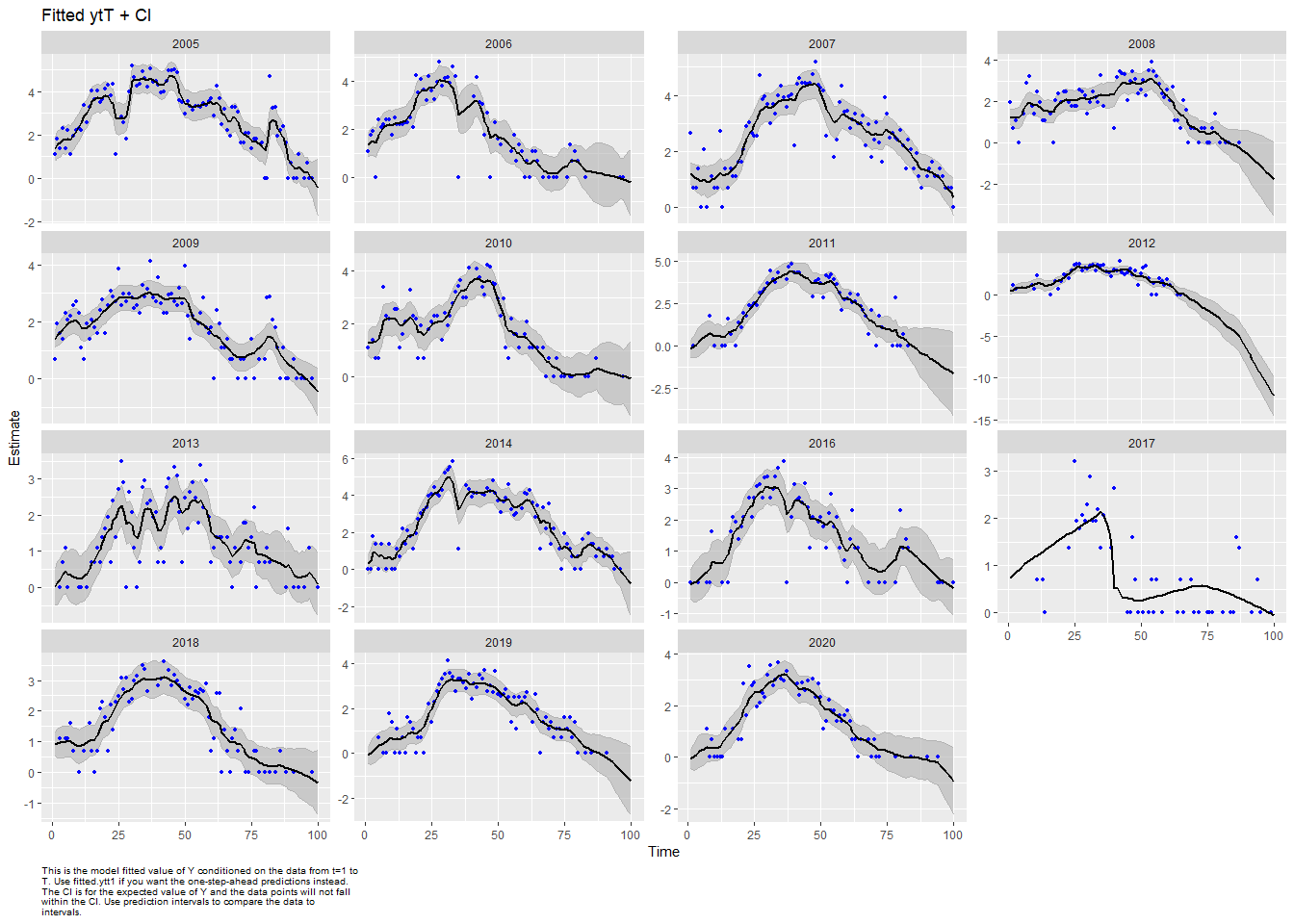
## plot.type = xtT



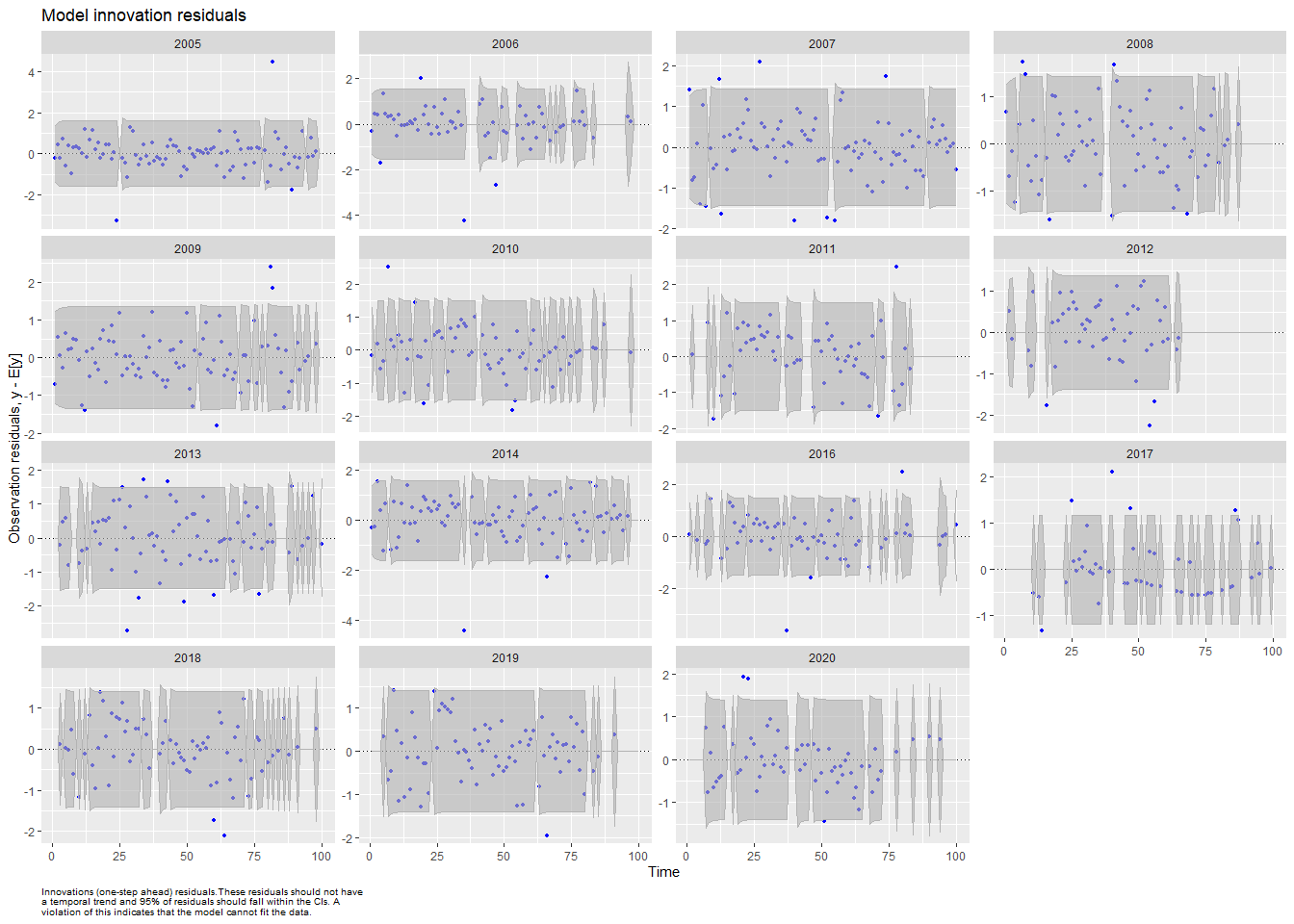
## Hit <Return> to see next plot (q to exit):



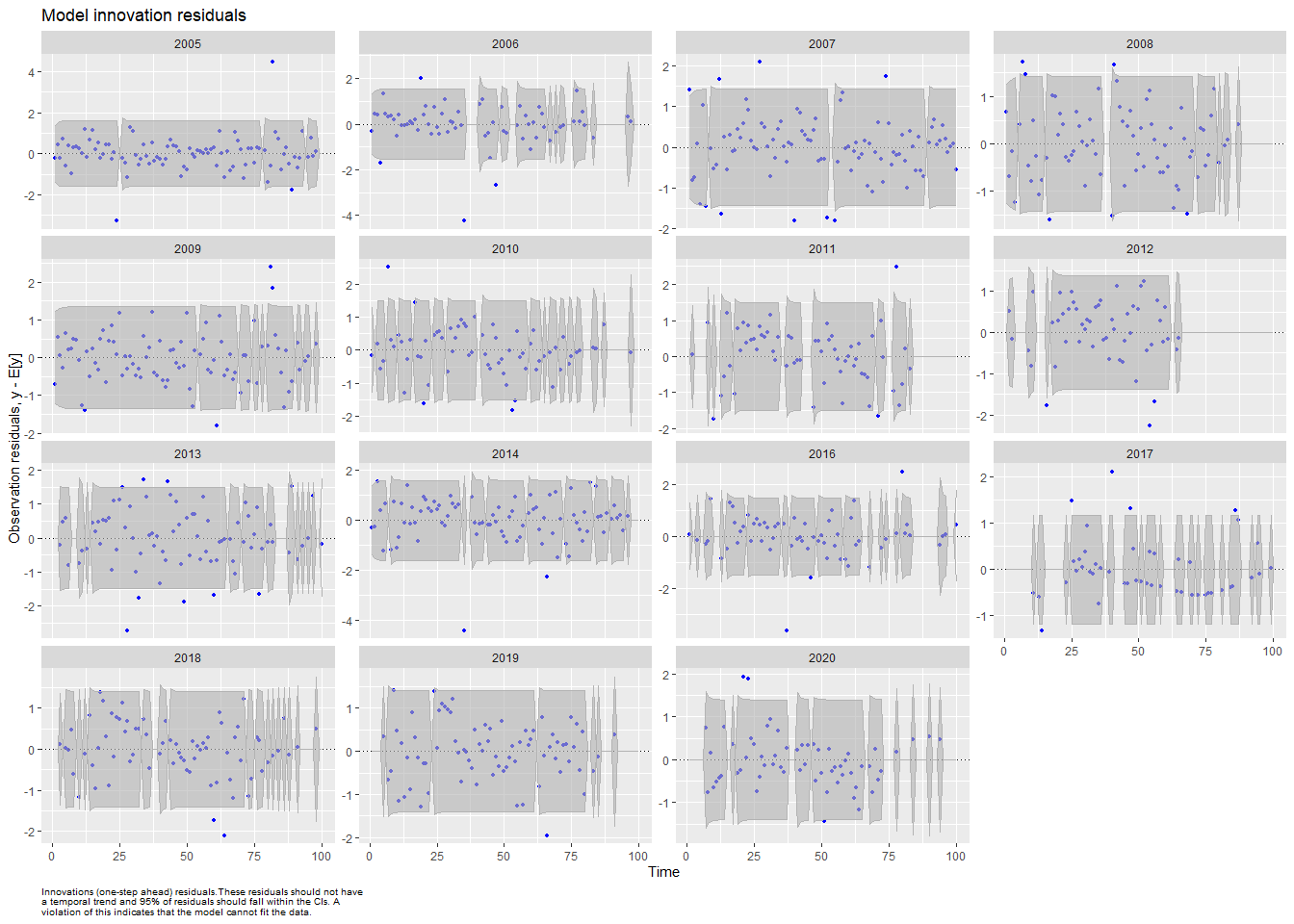
## plot.type = fitted.ytT



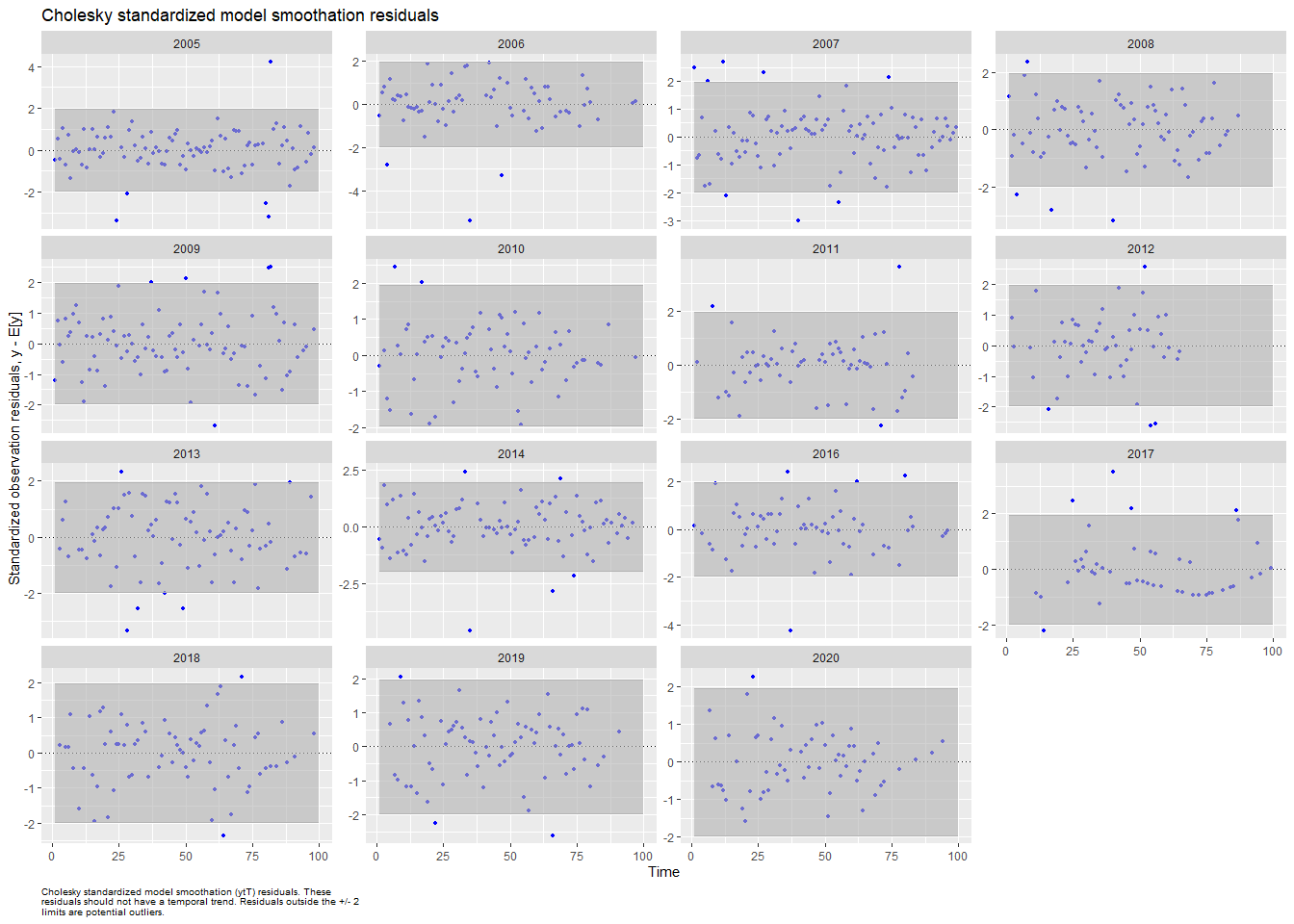
## Hit <Return> to see next plot (q to exit):



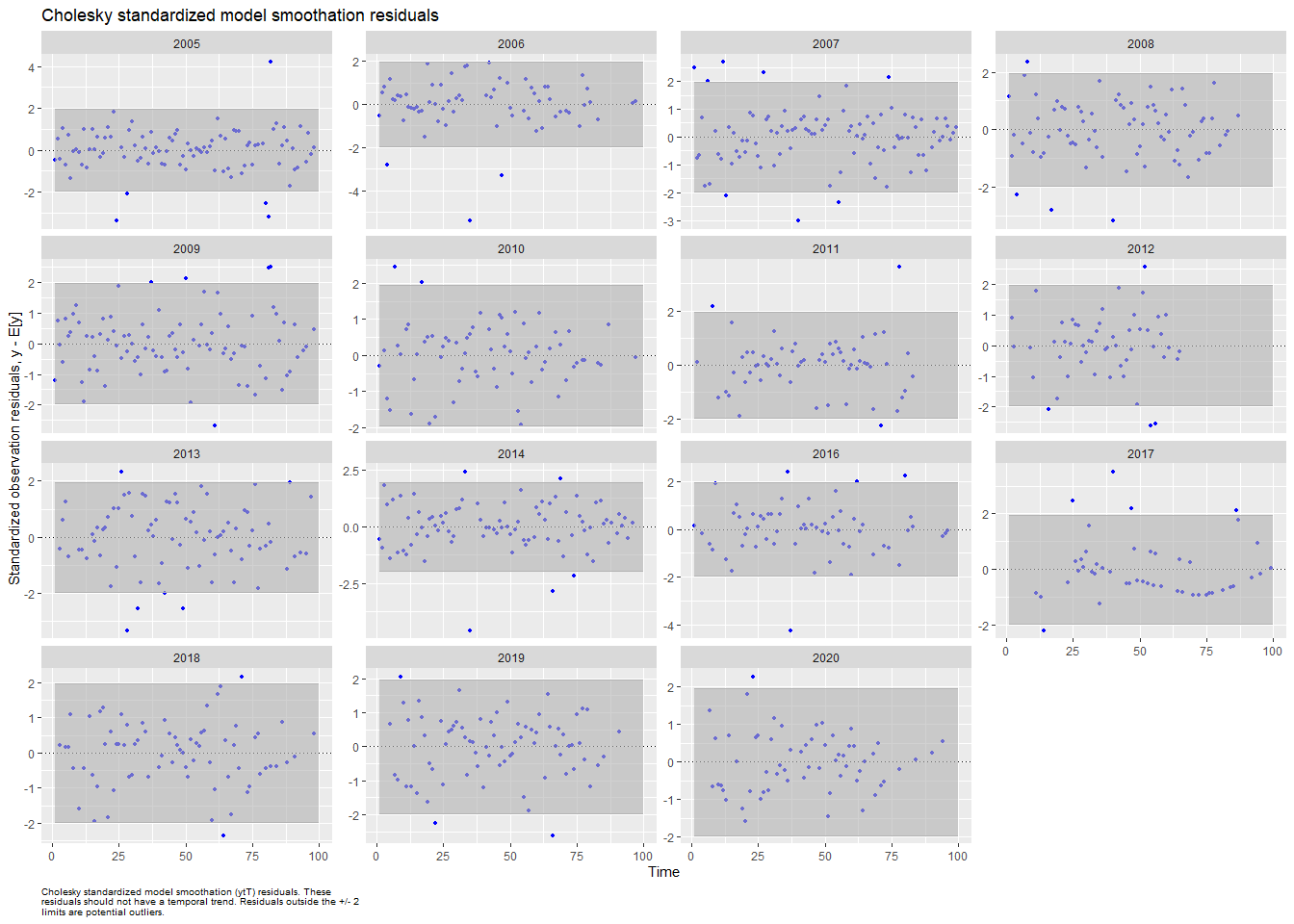
## plot.type = model.resids.ytt1



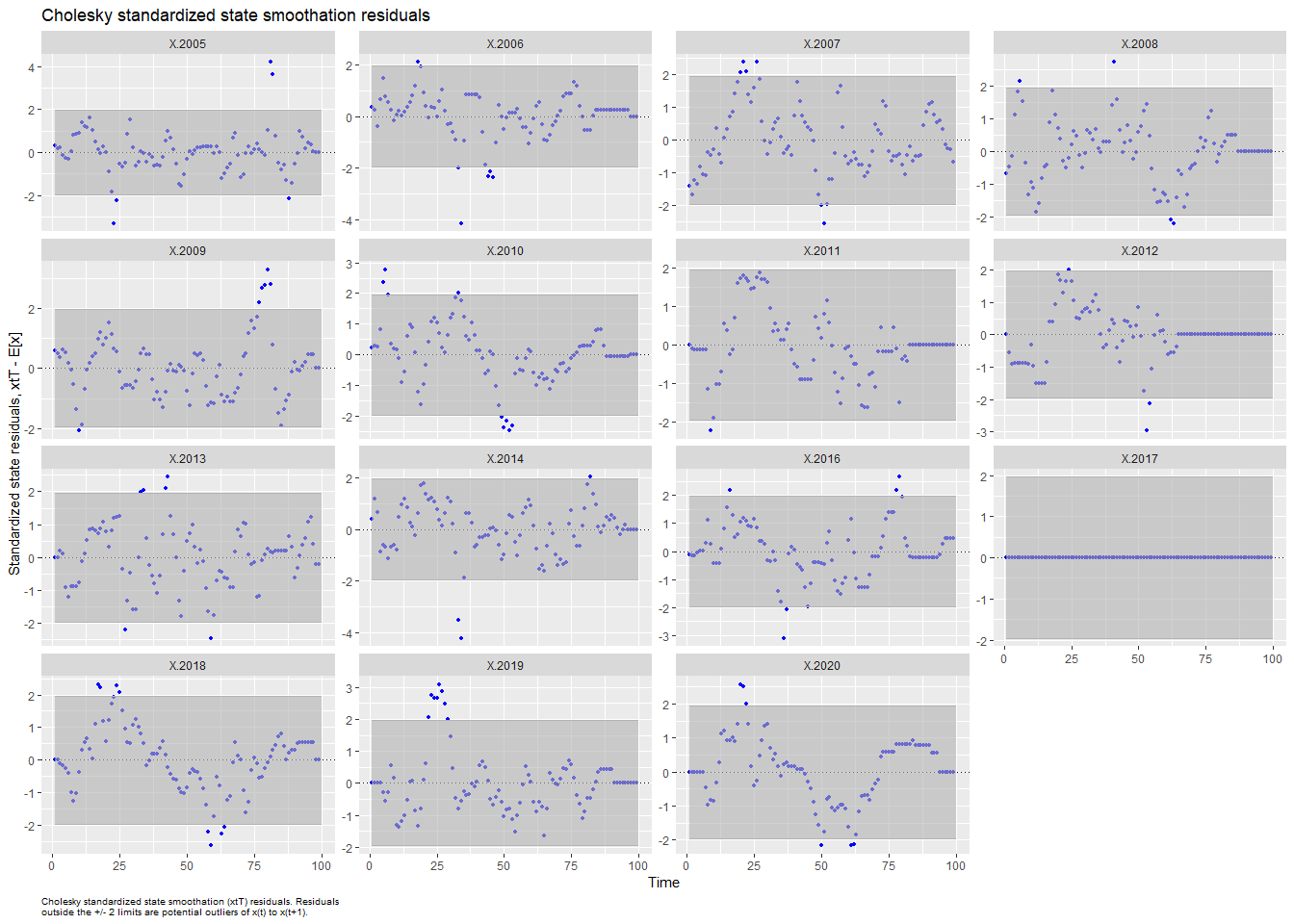
## Hit <Return> to see next plot (q to exit):



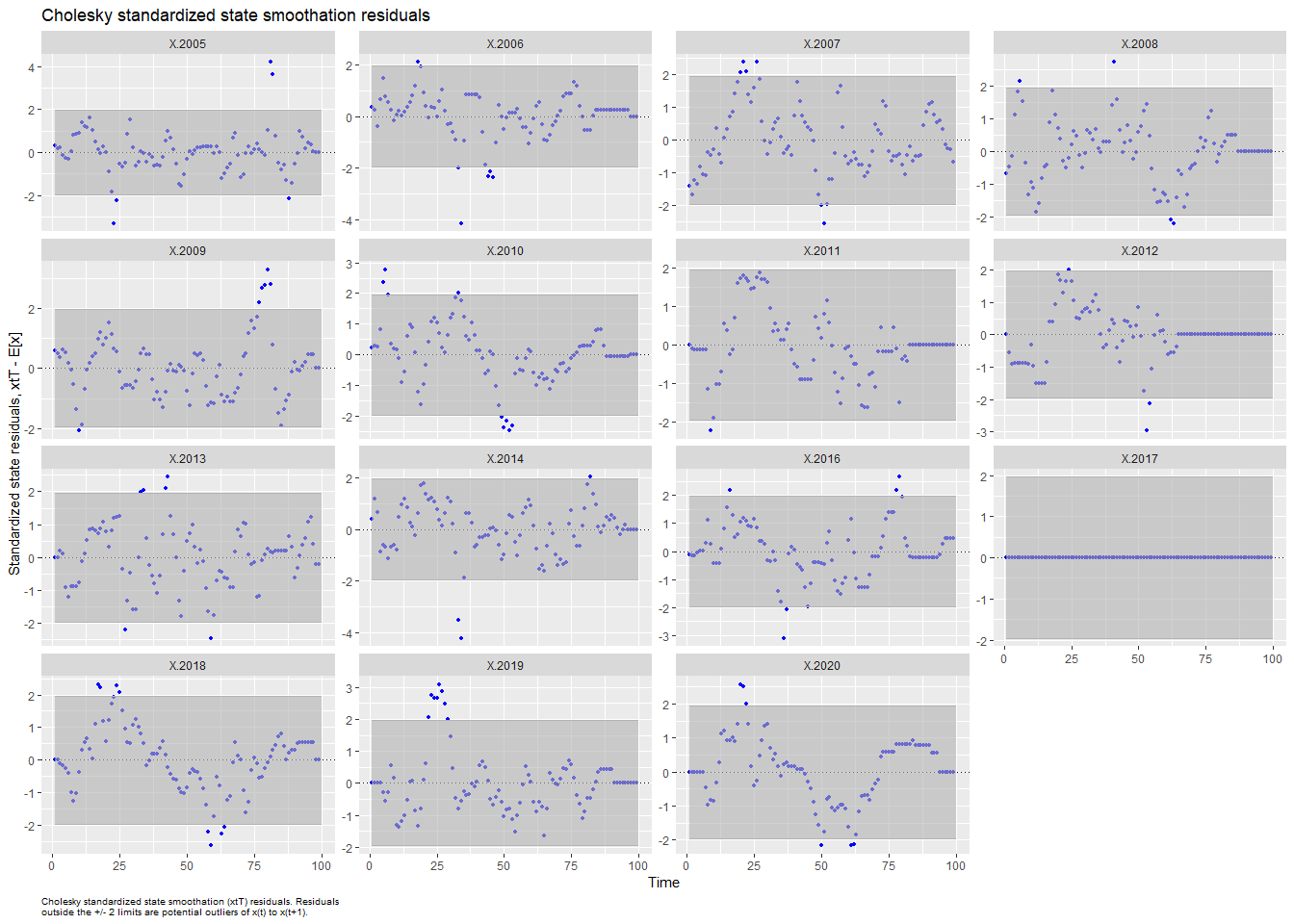
## plot.type = std.model.resids.ytT



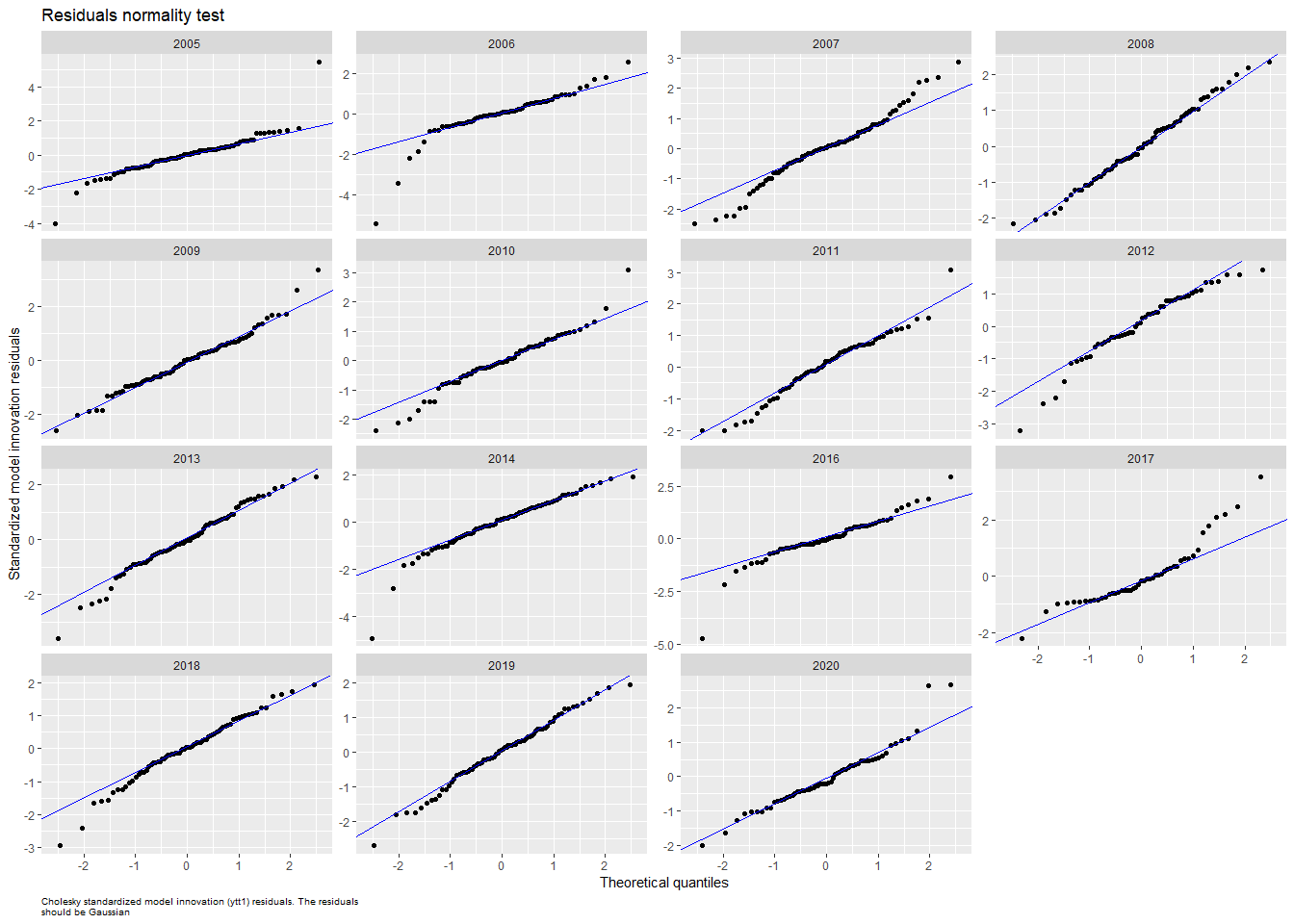
## Hit <Return> to see next plot (q to exit):



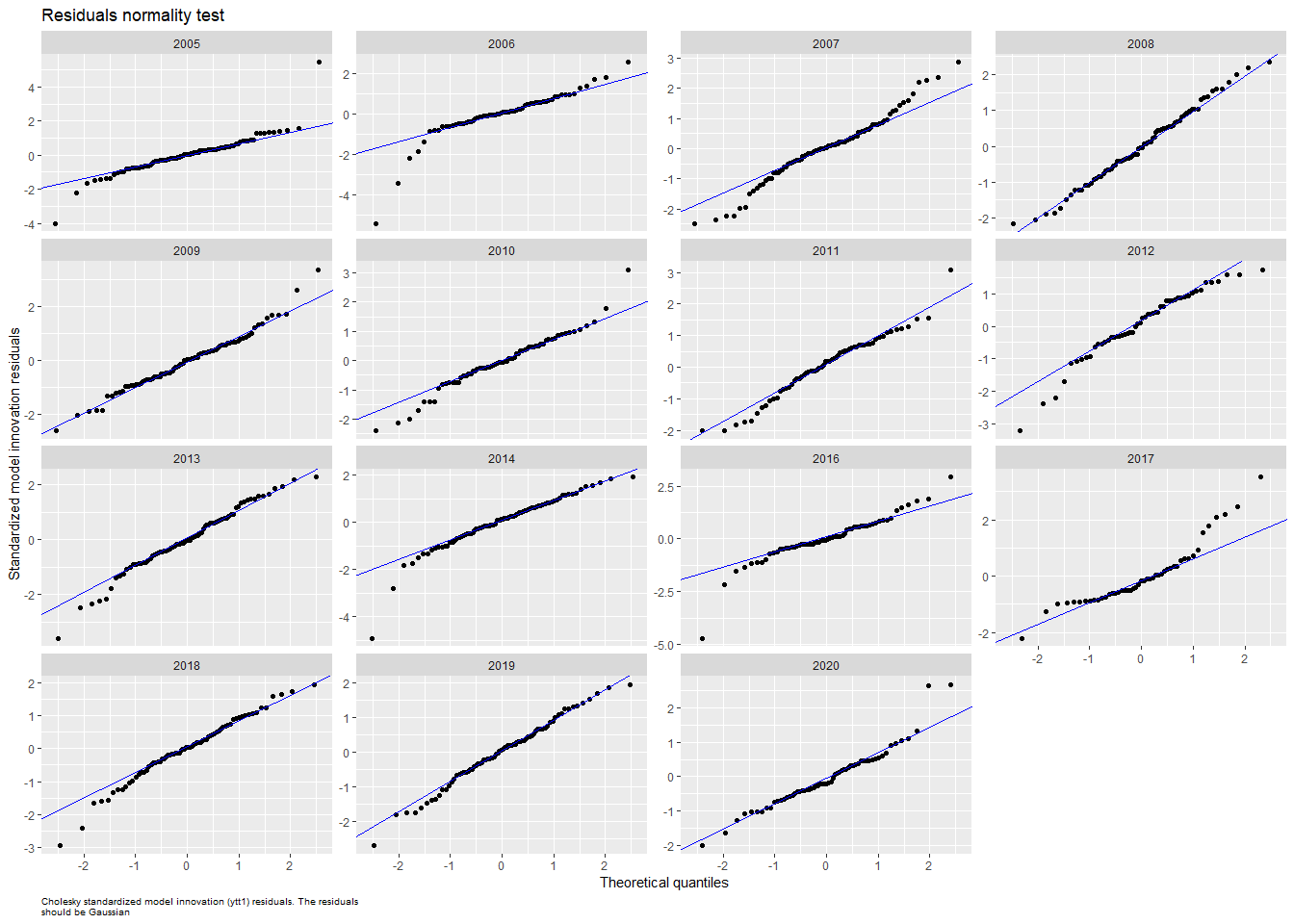
## plot.type = std.state.resids.xtT



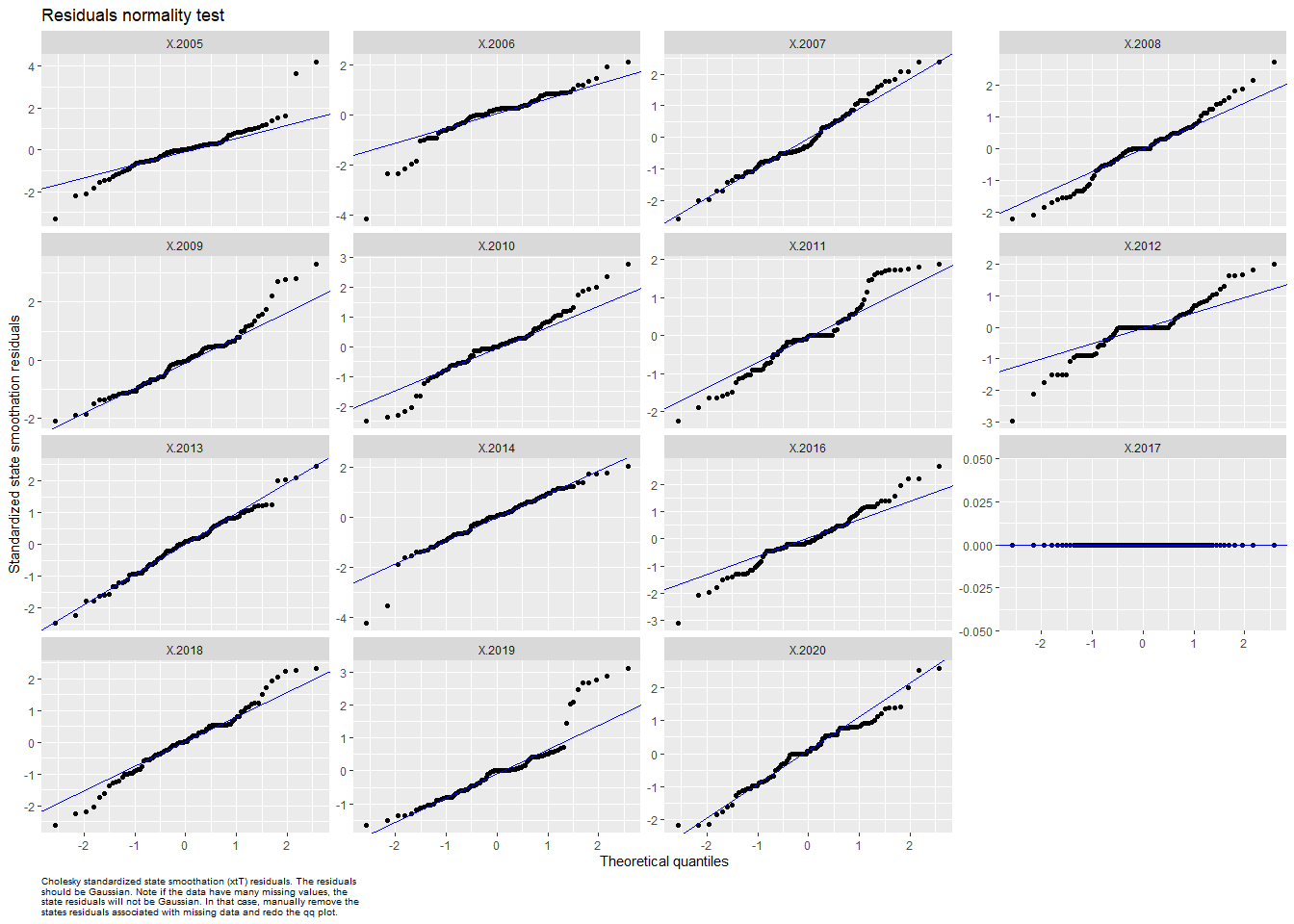
## Hit <Return> to see next plot (q to exit):



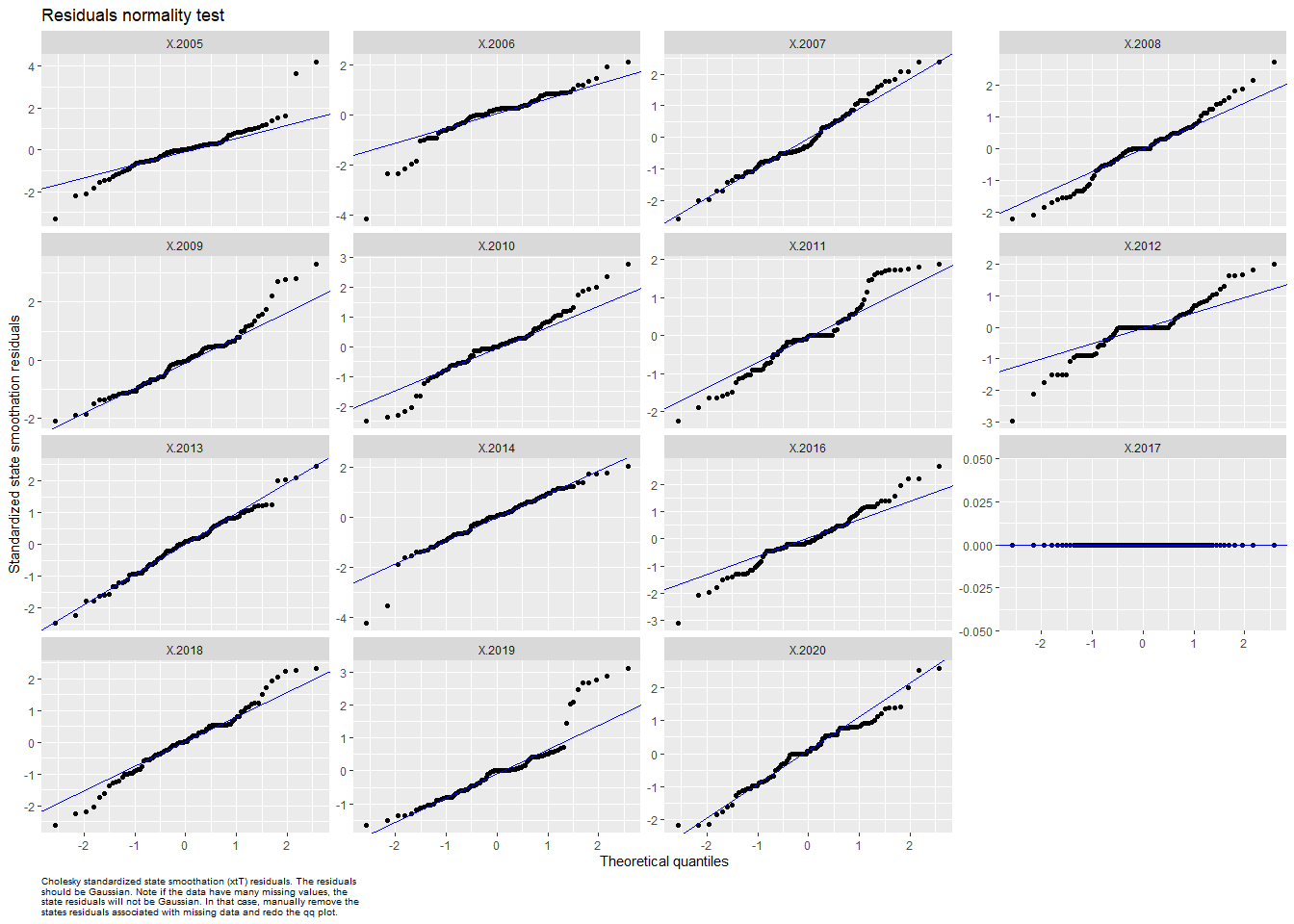
## plot.type = qqplot.std.model.resids.ytt1



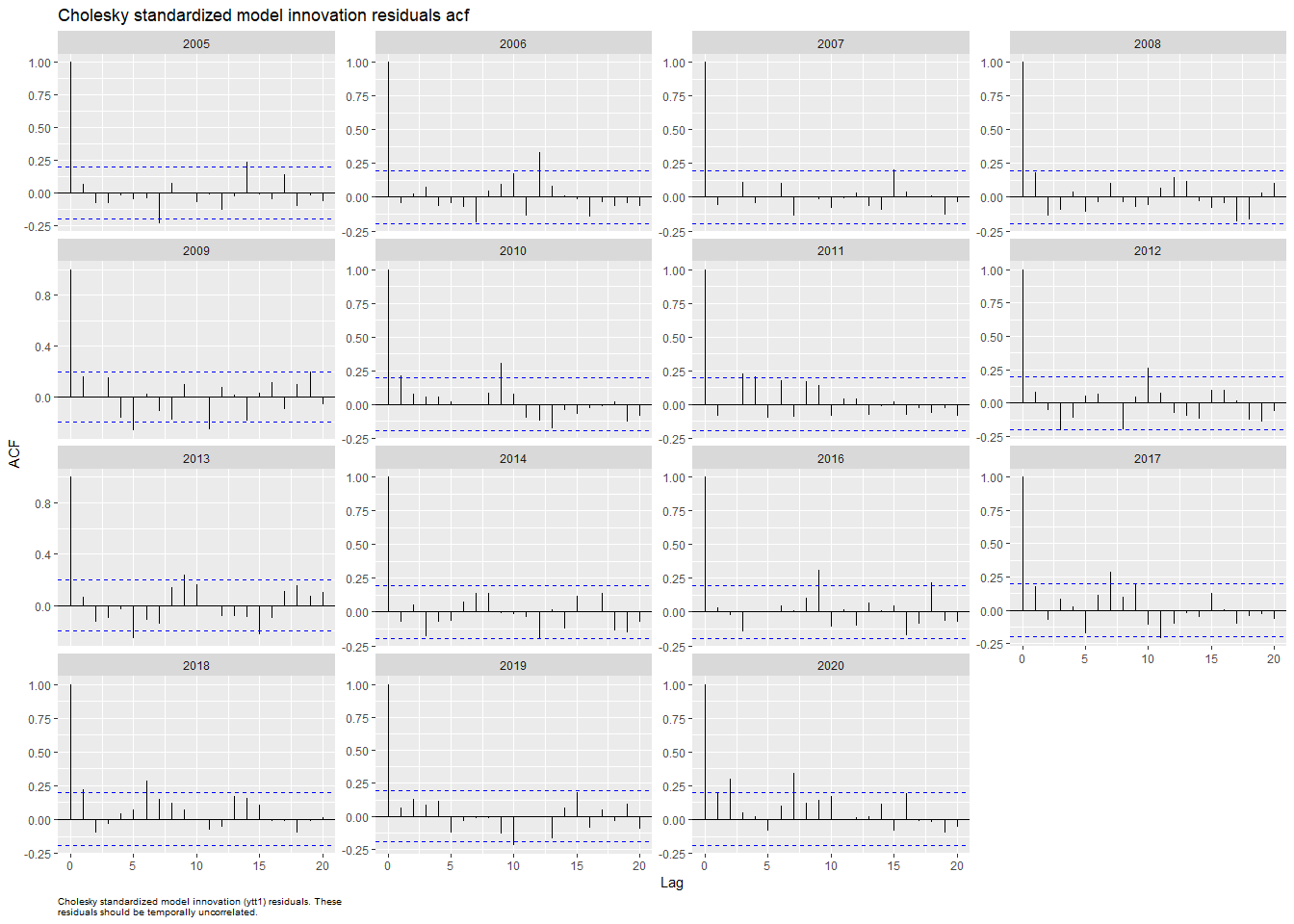
## Hit <Return> to see next plot (q to exit):



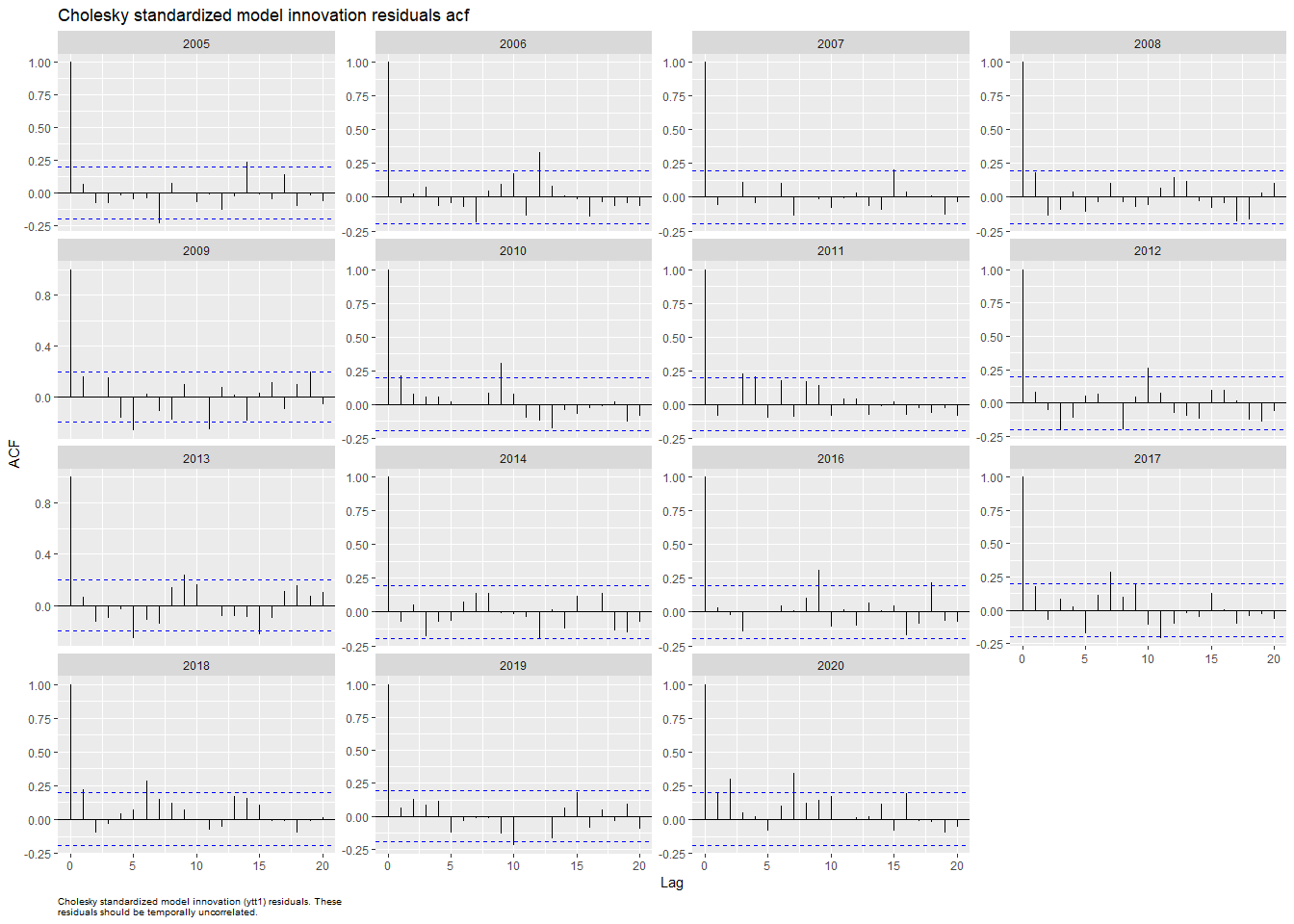
## plot.type = qqplot.std.state.resids.xtT



## Finished plots.



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



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