Time Series Model for School Appropriation

Eugene Quinn Ph.D.

May 6, 2024

Overview

This code fits a time series (ARIMA) model to the school appropriation using 28 years of data covering 1998 through 2025.

Model

The model is a variation on a first order autoregressive model. The feature that is different is that the coefficient of the lag 1 term is fixed at one.

Suppose we observe n appropriation values $y_1, y_2, \ldots, y_n$. The model is:

$$y_i = y_{i-1} + \beta_j + e_i, \quad i = 2, 3, \ldots, n \quad j = 1, 2, 3, 4, 5$$

The five β parameters $\beta_1, \beta_2, \ldots, \beta_5$ represent the expected increase in the school appropriation for the following periods:

- β_1 is the expected appropriation increase for normal years from 1998 to 2009
- β_2 is the expected appropriation increase for the anomalous year 2008 (large increase outlier)
- β_3 is the expected appropriation increase for the anomalous year 2010 (large decrease outlier)
- β_4 is the expected appropriation increase for the normal years from 2011 to 2024
- β_5 is the expected increase for the anomalous year 2018 (level funded)
- The e_i is the residual error or 'noise' terms, assumed to be independent with an expected value of zero and a common standard deviation of σ_e

Since the e_i term has an expected value of zero, the expected appropriation in year i is the previous year's appropriation plus the expected value of the β term.

$$E(y_i) = y_{i-1} + \beta_j$$

Parameters are estimated using a hierarchical Bayesian model with partial pooling.

Get data input

Load the inoput data and list it.

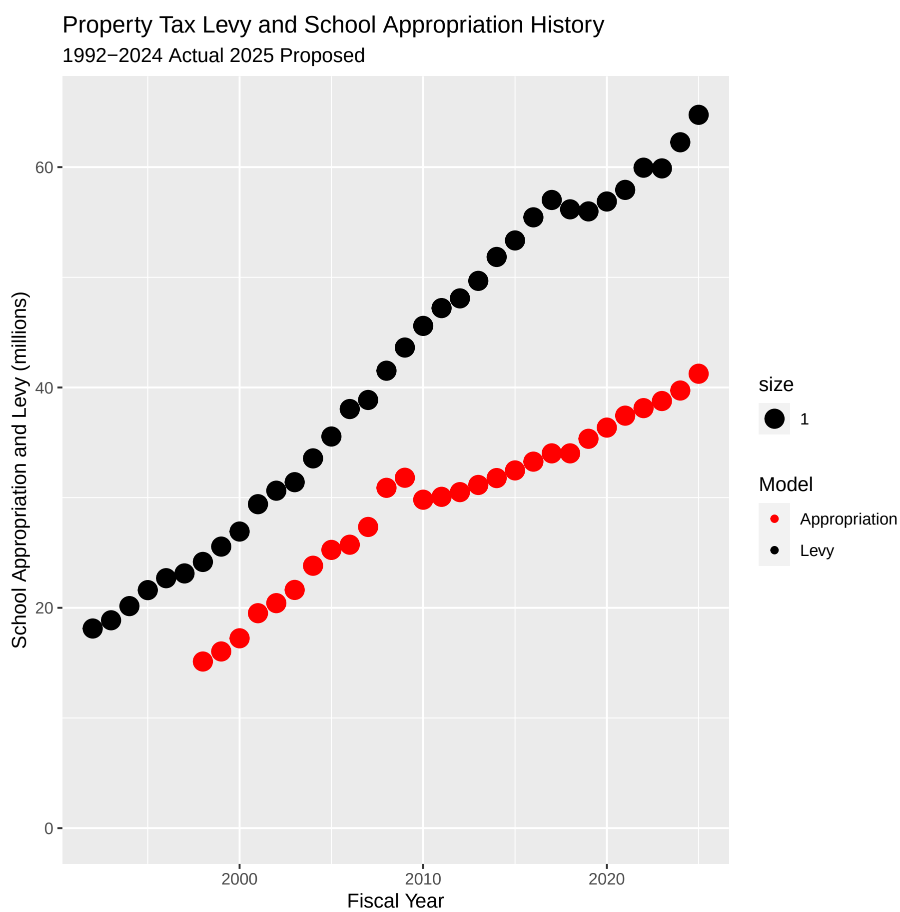
```
library(ggplot2)
library(rstan)

## Loading required package:  StanHeaders
##
## rstan version 2.32.5 (Stan version 2.32.2)
## For execution on a local, multicore CPU with excess RAM we recommend
calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend
calling
## rstan_options(auto_write = TRUE)
## For within-chain threading using `reduce_sum()` or `map_rect()` Stan
functions,
## change `threads_per_chain` option:
## rstan_options(threads_per_chain = 1)

library(shinystan)

## Loading required package:  shiny
##
## This is shinystan version 2.6.0

library(RMariaDB)
library(rlist)

rm(list=ls())

options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)

con <- dbConnect(RMariaDB::MariaDB(), group = "r-mdb")
```

```
res <- dbSendQuery(con, "SELECT * FROM CCD_Revenue WHERE `DOE Agency ID`=4400300;")
egdf = dbFetch(res)
dbClearResult(res)

res <- dbSendQuery(con, "SELECT * FROM Revenue WHERE `Dist No`=90 AND `Obj` in (41210,41101,
eguc = dbFetch(res)
dbClearResult(res)


dbDisconnect(con)

fa = eguc[eguc$Obj==43101,]                          #state aid
la = eguc[eguc$Obj!=43101,]                          #local appropriation

y1 = as.numeric(substr(egdf$`School Year`,6,10))    #build year, appropriation, state aid
apr = egdf$`Lcl Parent Govt`/1000000
sta = egdf$`St Gen Formula`/1000000

y2 = as.numeric(substr(la$`School Year`,6,10))
apru = la$Actual/1000000
stau = fa$Actual/1000000

years = c(y1[1:12],y2,2023,2024,2025)
#app = c(apr[1:12],apru,38.770,39.723,41.864) #ask
app = c(apr[1:12],apru,38.770,39.723,41.246) #tm proposed
sa  = c(sta[1:12],stau,4.702,5.969,6.731)

adf = data.frame(years,app,sa)                       #apptopriation and state aid dataframe

ll = list.load('Levy_additive_AR1.RDS')   #load total levy data
ptaxdf = ll$ptaxdf                        #extract dataframe
levy = c(ptaxdf$levy,64.747)
year = c(ptaxdf$year,2025)
aix1 = c(ptaxdf$aix1,1)
aix2 = c(ptaxdf$aix2,1)
ptaxdf2=data.frame(year,levy,aix1,aix2)
```

Plot School Appropriation and Tax Levy History

```
grp = c(rep('levy',nrow(ptaxdf2)),rep('appropriation',nrow(adf)))
y = c(ptaxdf2$levy,adf$app[1:(nrow(adf))])
year=c(ptaxdf2$year,adf$years[1:(nrow(adf))])
```

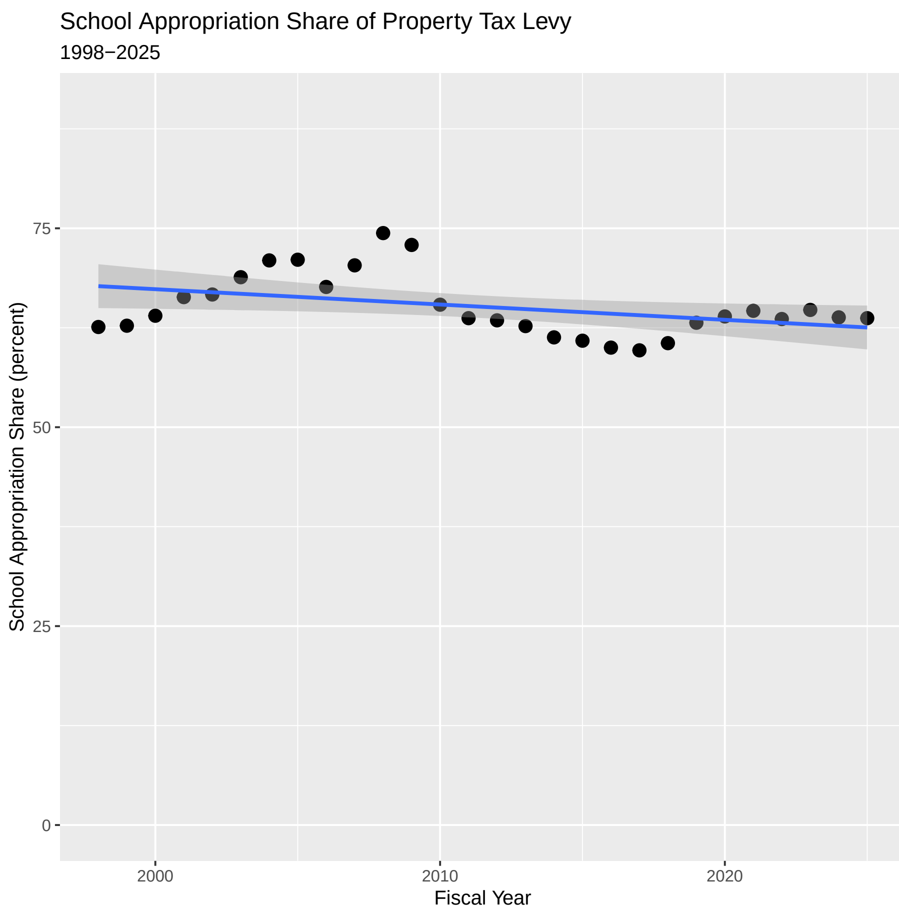
```r
ggdf=data.frame(y,year,grp)

ggplot(ggdf, aes(x=year, y=y)) +
  geom_point(aes(color=grp,size=1)) + ylim(0,65) +
  scale_color_manual(name='Model', labels=c('Appropriation', 'Levy'),
                     values=c('red', 'black')) +
  labs(x = "Fiscal Year", y="School Appropriation and Levy (millions)",
       title = "Property Tax Levy and School Appropriation History",
       subtitle = "1992-2024 Actual 2025 Proposed")
```




```r
fname = paste("images/School_Appropriation_and_Levy_History_Through_2025_",Sys.Date(),".png'
fname

## [1] "images/School_Appropriation_and_Levy_History_Through_2025_2024-05-06.png"

ggsave(fname, width = 8, height = 5, units = "in")
```

Plot School Appropriation Percentage of the Tax Levy

```
ptx2 = ptaxdf2[ptaxdf2$year > 1997,]
adf2 = adf[adf$years < 2025,]
yrs = ptx2$year
levy = ptx2$levy
#app = adf2£app
app = adf$app
ptax0 = data.frame(yrs,levy,app)
ptax0$pcts = 100*((ptax0$app/ptax0$levy))
ggplot(ptax0, aes(x=yrs, y=pcts)) +
  geom_point(size=3) + ylim(0,90) +
  geom_smooth(method='lm', formula= y~x) +
  labs(x = "Fiscal Year", y="School Appropriation Share (percent)",
       title = "School Appropriation Share of Property Tax Levy",
       subtitle="1998-2025")
```




```
fname = paste("images/School_Appropriation_Share_of_Property_Tax_Levy_",Sys.Date(),".png",se
fname

## [1] "images/School_Appropriation_Share_of_Property_Tax_Levy_2024-05-06.png"

ggsave(fname, width = 8, height = 5, units = "in")
```

/section*Set up and call STAN Define the STAN input parameters and call STAN

```
                           #type of year codes
bix1 = rep(1,length(app))  #type of year 1=normal 1992-2007   #levels for type of year
bix1[11]    = 2      #2008 sharp increase
bix1[13]    = 3      #2010 sharp decrease
bix1[14:28] = 4      #normal 2011-2025 except 2018
bix1[21]    = 5      #level-fund 2018
```

```r
                                              #STAN setup
appin = app[1:length(app)]                    #  appropriation actual data (drop 2025)
Ndata = length(appin)                         #  num
years_ahead = 13                              #  number of years ahead for posterior predictive
future_beta = 4                               #  beta level to use for future years
y = appin                                     #  model file dependent variable is y
bixin = bix1[1:Ndata]                         #  type of year vector
n_params = length(table(bixin[1:Ndata]))      #  number of year type codes
chains <- 4                                   #  four chains - one for each cpu
                                              #  call STAN
gp <- stan("Appropriation_time_series.stan",
           data=list(N=Ndata,y=y,n_params=n_params,bix=bixin,
                     years_ahead=years_ahead,future_beta=future_beta),
           control=list(max_treedepth=18),
           chains=chains,cores=4,iter=8000)

#launch_shinystan(gp)                          #launch shinystan to check results
pd = rstan::extract(gp)                       #extract posterior draw from stanfit object
```

1 Quantiles for 2025 Appropriation

```r
mean(pd$yp[,28])

## [1] 40.53302

sum(pd$yp[,28]<=41.246)/length(pd$yp[,28])

## [1] 0.9244375

quantile(pd$yp[,28],c(.25,.5,.75,.9,.95,.99))

##      25%      50%      75%      90%      95%      99%
## 40.20433 40.53711 40.86133 41.17287 41.36183 41.71318

for (i in 1:length(pd$beta[1,])){
  print(i)
  print(quantile(pd$beta[,i],c(.25,.5,.75)))
  print(mean(pd$beta[,i]))
  print(sd(pd$beta[,i]))
}

## [1] 1
##      25%      50%      75%
## 1.201567 1.301842 1.403104
```

```
## [1] 1.301691
## [1] 0.1546063
## [1] 2
##      25%      50%      75%
## 2.943340 3.273170 3.591852
## [1] 3.255664
## [1] 0.5027058
## [1] 3
##       25%       50%       75%
## -2.071925 -1.755919 -1.427784
## [1] -1.741897
## [1] 0.4965344
## [1] 4
##       25%       50%       75%
## 0.7314791 0.8171015 0.9030466
## [1] 0.8166072
## [1] 0.1322093
## [1] 5
##         25%         50%         75%
## -0.25208987  0.05645997  0.36640163
## [1] 0.05314215
## [1] 0.4706364
```

```
mean(pd$sigma_e)
```

```
## [1] 0.4850982
```

```
sd(pd$sigma_e)
```

```
## [1] 0.08047298
```

Computed Fitted and Residual Values and Mean Square Error

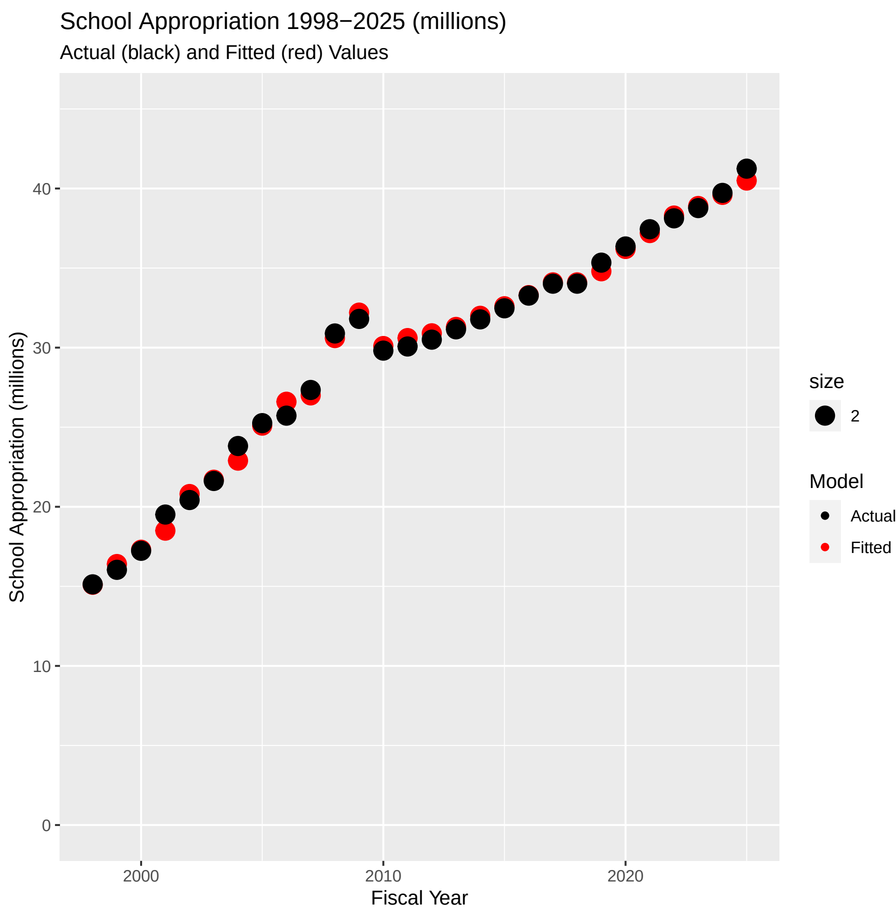
```
fitted = c()                               #use means of posterior draw as fitted values
for (i in 1:Ndata){
  fitted = c(fitted,mean(pd$yp[,i]))
}
fitted = round(fitted,1)
residual = appin - fitted;                 #calculate residuals
mse = sum(residual^2)/length(residual)   #calculate Mean Square Error
adf$fitted = fitted
adf$residual = residual
adf
```

```
##    years      app       sa fitted  residual
## 1   1998 15.12500 1.259000   15.1  0.025000
```

```
## 2   1999 16.03800 1.404000   16.4 -0.362000
## 3   2000 17.23300 1.460000   17.3 -0.067000
## 4   2001 19.51100 1.533000   18.5  1.011000
## 5   2002 20.42400 1.642000   20.8 -0.376000
## 6   2003 21.62200 1.796000   21.7 -0.078000
## 7   2004 23.81800 1.796000   22.9  0.918000
## 8   2005 25.25600 1.810000   25.1  0.156000
## 9   2006 25.72700 1.860000   26.6 -0.873000
## 10  2007 27.33900 1.950000   27.0  0.339000
## 11  2008 30.89000 1.950000   30.6  0.290000
## 12  2009 31.80500 1.184000   32.2 -0.395000
## 13  2010 29.81234 1.321450   30.1 -0.287656
## 14  2011 30.07252 1.296770   30.6 -0.527484
## 15  2012 30.50108 1.443106   30.9 -0.398924
## 16  2013 31.15012 1.944662   31.3 -0.149880
## 17  2014 31.77620 2.297840   32.0 -0.223800
## 18  2015 32.47210 2.437112   32.6 -0.127900
## 19  2016 33.26756 2.858605   33.3 -0.032444
## 20  2017 34.01890 2.775359   34.1 -0.081096
## 21  2018 34.01890 2.671596   34.1 -0.081096
## 22  2019 35.34081 3.091316   34.8  0.540812
## 23  2020 36.35756 2.589117   36.2  0.157564
## 24  2021 37.44126 3.415359   37.2  0.241264
## 25  2022 38.12580 4.450983   38.3 -0.174200
## 26  2023 38.77000 4.702000   38.9 -0.130000
## 27  2024 39.72300 5.969000   39.6  0.123000
## 28  2025 41.24600 6.731000   40.5  0.746000
```

Plot Actual and Fitted

```
yr = rep(adf$year,2)
grp=c(rep('fitted',nrow(adf)),rep('actual',nrow(adf)))
approp = c(adf$fitted,adf$app)

gdf = data.frame(yr,grp,approp)

ggplot(gdf, aes(x=yr, y=approp)) +
  geom_point(aes(color=grp,size=2)) + ylim(0,45) +
  scale_color_manual(name='Model', labels=c('Actual', 'Fitted'),
                     values=c('black', 'red')) +
  labs(x = "Fiscal Year", y="School Appropriation (millions)",
       title = "School Appropriation 1998-2025 (millions)",
       subtitle = "Actual (black) and Fitted (red) Values")
```




```
fname = paste("images/School_Appropriation_History_Through_2025_actual_and_fitted_",Sys.Date
fname

## [1] "images/School_Appropriation_History_Through_2025_actual_and_fitted_2024-05-06.png"

ggsave(fname, width = 8, height = 5, units = "in")
```

Forecast

Run forecast

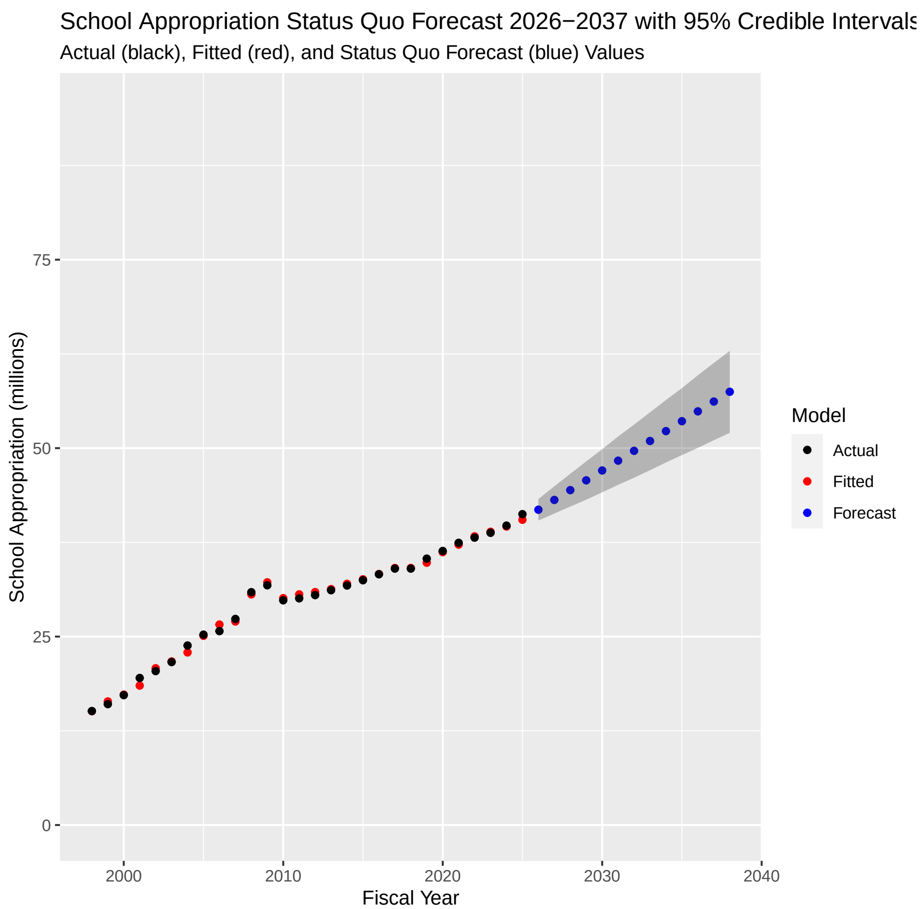
```
yrsp = length(pd$yp[1,]) - Ndata             #years ahead
last_data_year = adf$year[nrow(adf)]
last_data_year + yrsp
```

```
## [1] 2038
```

```
yr = seq(last_data_year+1,last_data_year+yrsp) #forecast years
grp = rep('Forecast',yrsp)
approp = c()
app01 = c()
app025= c()
app05 = c()
app25 = c()
app75 = c()
app95 = c()
app975= c()
app99 = c()
for (i in (1+nrow(adf)):(nrow(adf)+yrsp)){
  approp = c(approp,mean(pd$yp[,i]))
  app01 = c(app01,quantile(pd$yp[,i],0.01))
  app025 = c(app025,quantile(pd$yp[,i],0.025))
  app05 = c(app05,quantile(pd$yp[,i],0.05))
  app25 = c(app25,quantile(pd$yp[,i],0.25))
  app75 = c(app75,quantile(pd$yp[,i],0.75))
  app95 = c(app95,quantile(pd$yp[,i],0.95))
  app975 = c(app975,quantile(pd$yp[,i],0.975))
  app99 = c(app99,quantile(pd$yp[,i],0.99))
}

gdff = data.frame(yr,grp,approp)
gdfpred = data.frame(yr,approp,app01,app025,app05,app25,app75,app95,app975,app99)
gdfpred
```

```
##      yr   approp    app01   app025    app05    app25    app75    app95   app975
## 1  2026 41.83191 40.07473 40.39809 40.64411 41.35968 42.30816 43.00367 43.25471
## 2  2027 43.13001 40.91227 41.32312 41.62353 42.52857 43.73637 44.61284 44.93598
## 3  2028 44.43079 41.77382 42.25087 42.62568 43.71244 45.16310 46.23556 46.60071
## 4  2029 45.72766 42.68633 43.15668 43.64713 44.89822 46.55836 47.81373 48.23781
## 5  2030 47.03609 43.55800 44.13957 44.66053 46.08890 47.97623 49.38690 49.86457
## 6  2031 48.33766 44.42801 45.13149 45.70237 47.28194 49.39621 50.99809 51.54901
## 7  2032 49.64464 45.41283 46.07707 46.73375 48.47521 50.79170 52.55657 53.12509
## 8  2033 50.95173 46.31857 47.07534 47.78759 49.69068 52.19521 54.12981 54.74788
## 9  2034 52.26011 47.24340 48.10216 48.82445 50.88017 53.63542 55.70003 56.39442
## 10 2035 53.57410 48.15338 49.08061 49.89922 52.08526 55.04574 57.29683 57.96098
## 11 2036 54.88070 49.14150 50.03700 50.94139 53.30654 56.45633 58.83278 59.67239
## 12 2037 56.18253 49.99726 51.04930 51.93964 54.49686 57.87491 60.43081 61.29948
## 13 2038 57.47965 50.93481 52.02749 52.97823 55.68711 59.27064 61.97393 62.90647
##       app99
## 1  43.56637
## 2  45.34823
```

```
## 3  47.05119
## 4  48.74573
## 5  50.53997
## 6  52.19245
## 7  53.88833
## 8  55.55719
## 9  57.23976
## 10 58.92435
## 11 60.54869
## 12 62.27526
## 13 63.96076

gdf2 = rbind(gdf,gdff)
p = ggplot(gdf2, aes(x=yr, y=approp)) +
  geom_point(aes(color=grp)) + ylim(0,95) +
  scale_color_manual(name='Model', labels=c('Actual', 'Fitted','Forecast'),
                     values=c('black', 'red','blue')) +
  labs(x = "Fiscal Year", y="School Appropriation (millions)",
       title = "School Appropriation Status Quo Forecast 2026-2037 with 95% Credible Interva
       subtitle = "Actual (black), Fitted (red), and Status Quo Forecast (blue) Values")

p + geom_ribbon(data=gdfpred,aes(ymin=app025,ymax=app975),alpha=0.3)
```

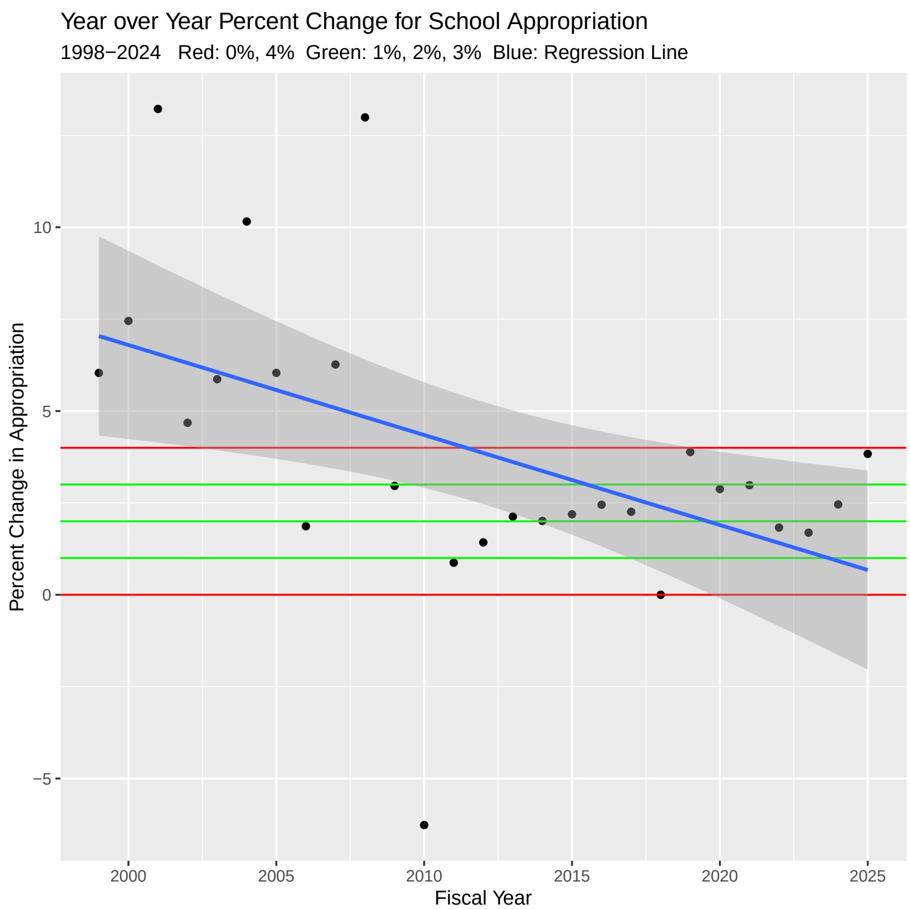



```
fname = paste("images/School_Appropriation_History_Through_2025_13_year_status_quo_forecast_
fname

## [1] "images/School_Appropriation_History_Through_2025_13_year_status_quo_forecast_2024-05

ggsave(fname, width = 8, height = 5, units = "in")
```

Compute Linear Properties

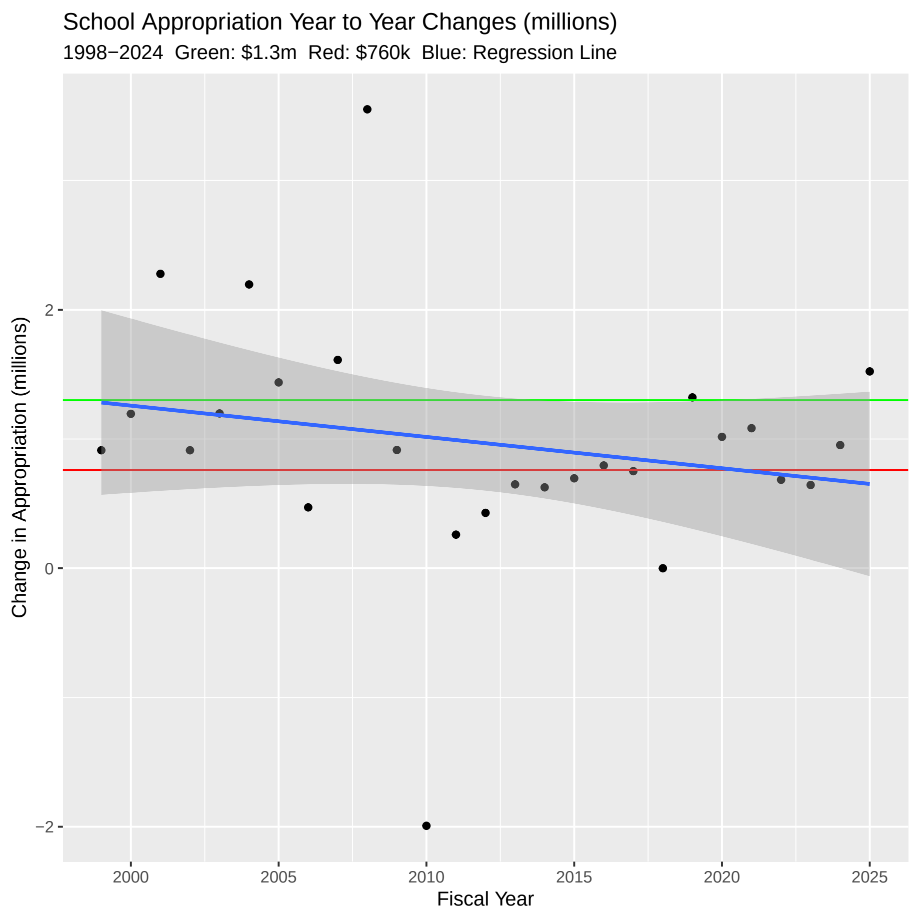
```
yrsin = adf$years[1:length(appin)]          #build dataframe for fitted and residual data
pdf = data.frame(yrsin,appin,fitted,residual)

pctchg = 100*(appin[2:length(appin)]/appin[1:(length(appin)-1)]-1) #year over year % change
```

```r
pyr = yrsin[2:length(yrsin)]                                  #year vector
pcdf = data.frame(pyr,pctchg)                                   #data frame for ggplot2
ggplot(pcdf, aes(x=pyr, y=pctchg)) +
  geom_point() + geom_hline(yintercept=3,linewidth=0.5,color='green') +
  geom_hline(yintercept=4, linewidth=0.5, color='red') +
  geom_hline(yintercept=2, linewidth=0.5, color='green') +
  geom_hline(yintercept=1, linewidth=0.5, color='green') +
  geom_hline(yintercept=0, linewidth=0.5, color='red') +
  geom_smooth(method='lm', formula= y~x) +
  labs(x = "Fiscal Year", y="Percent Change in Appropriation",
       title = "Year over Year Percent Change for School Appropriation",
       subtitle="1998-2024   Red: 0%, 4%  Green: 1%, 2%, 3%  Blue: Regression Line")
```




```r
delta = appin[2:length(appin)] - appin[1:(length(appin)-1)]   #year to year delta
dyr = yrsin[2:length(yrsin)]                                  #year vector
ddf = data.frame(dyr,delta)                                   #data frame for ggplot2
```

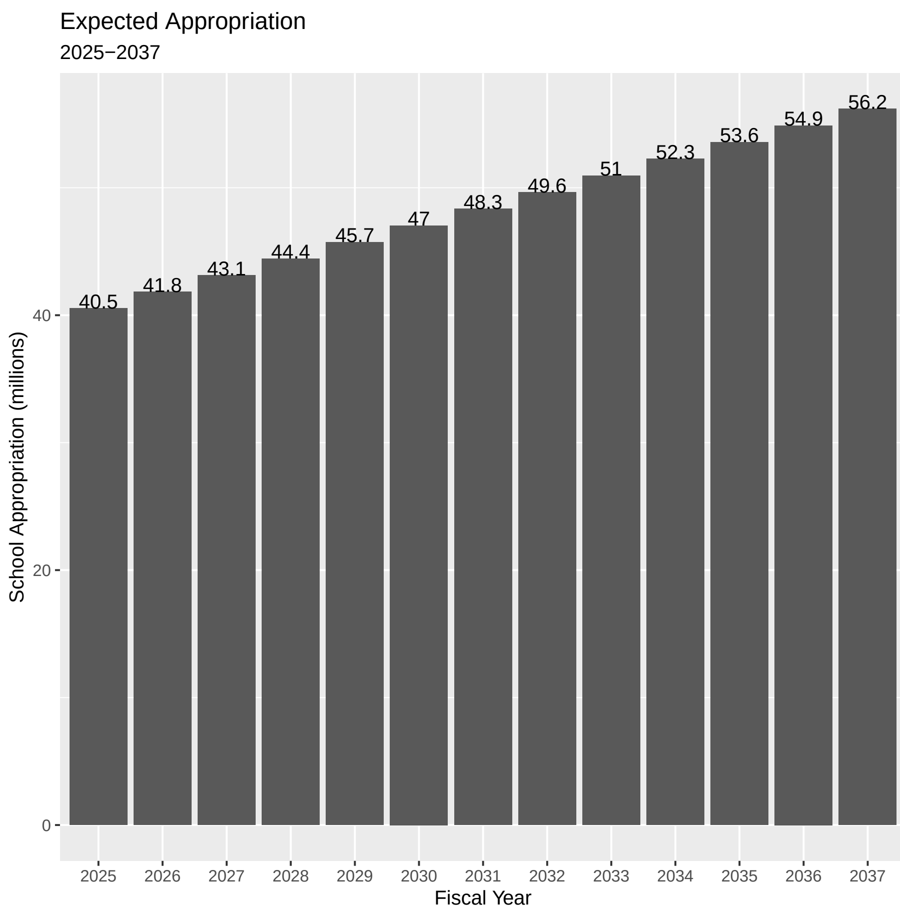
```r
ggplot(ddf, aes(x=dyr, y=delta)) +
  geom_point() + geom_hline(yintercept=1.3,linewidth=0.5,color='green') +
  geom_hline(yintercept=0.76, linewidth=0.5, color='red') +
  geom_smooth(method='lm', formula= y~x) +
  labs(x = "Fiscal Year", y="Change in Appropriation (millions)",
       title = "School Appropriation Year to Year Changes (millions)",
       subtitle = "1998-2024  Green: $1.3m  Red: $760k  Blue: Regression Line")
```



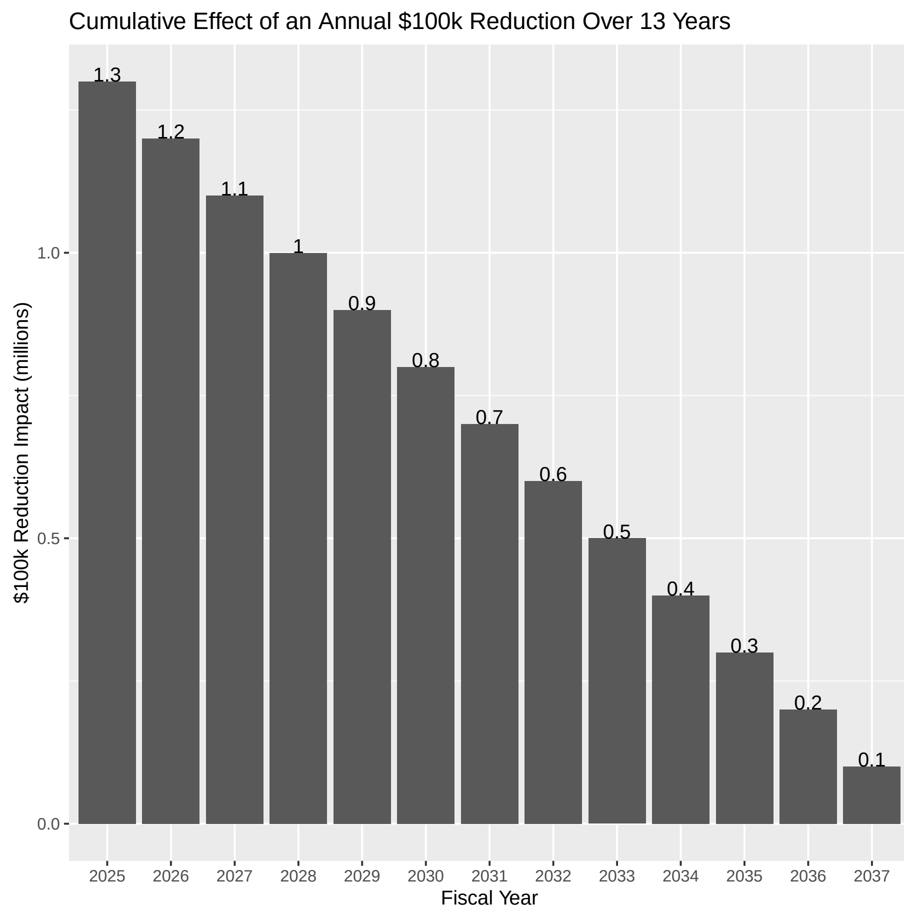
Barcharts for Forecast Values

```r
apt = rep(0,length(pd$yp[,1]))
for (i in 28:40){
  for (j in 1:length(pd$yp[,1])){
```

```
    apt[j] = apt[j] + pd$yp[j,i]
  }
}
predicted = c()
for (i in 28:40){
  predicted = c(predicted,mean(pd$yp[,i]))
}
pyears = as.factor(seq(2025,2037))
deltas = 0.100*(1+length(pyears) - seq(1:length(pyears)))

bcdf = data.frame(pyears,predicted,deltas)

ggplot(bcdf, aes(x = pyears, y = predicted)) +  # Plot with values on top
  geom_bar(stat = "identity") +
  geom_text(aes(label = round(predicted,1)), vjust = 0) +
  labs(x = "Fiscal Year", y="School Appropriation (millions)",
       title="Expected Appropriation",
       subtitle="2025-2037")
```




```
ggplot(bcdf, aes(x = pyears, y = deltas)) +  # Plot with values on top
  geom_bar(stat = "identity") +
  geom_text(aes(label = round(deltas,1)), vjust = 0) +
  labs(x = "Fiscal Year", y="$100k Reduction Impact (millions)",
       title="Cumulative Effect of an Annual $100k Reduction Over 13 Years")
```




```
savelist = list(egdf=egdf,eguc=eguc,adf=adf,ptaxdf2=ptaxdf2,gp=gp,mse=mse)
fname = paste("Appropriation_time_series_",Sys.Date(),".RDS",sep='')
fname

## [1] "Appropriation_time_series_2024-05-06.RDS"

list.save(savelist,fname)
```

STAN Model File

This is the STAN model file:

```r
get_stanmodel(gp)
```

```
## S4 class stanmodel 'anon_model' coded as follows:
## // time series model for school appropriation
## data {
##   int<lower=2> N;                              // observed data points
##   vector[N] y;                                 //levy
##   int<lower=1> n_params;                       //number of beta parameters
##   int<lower=1> years_ahead;                    //number of years for posterior predictives
##   int<lower=1,upper=n_params> future_beta;     //index of beta for future years
##   int bix[N];                                  //beta parameter index
## }
## parameters {
##   real a;                              //hyperparameter for beta
##   real<lower=0> b;                     //hyperparameter for sigma_beta
##   vector[n_params] beta;
##   vector<lower=0>[n_params] sigma_beta;  //vector of sigma_beta parameters
##   real<lower=0> sigma_e;               //residual standard error
## }
## model {
##   a ~ normal(0,1);                     //normal prior for hyperparameter a
##   b ~ normal(0,1);                     //normal prior for hyperparameter b
##   beta ~ normal(a,b);                  //hormal prior for betas
##   sigma_beta ~ normal(0,1);            //half-normal prior for sigma_betas
##   sigma_e ~ normal(0,1);               //half-normal prior for sigma_e
##   for (i in 2:N){
##     y[i] ~ normal(y[i-1] + beta[bix[i]],sigma_e);  //model for y
##   }
## }
## generated quantities{
##   vector[N+years_ahead] yp;             //fitted and posterior predicted
##   yp[1] = normal_rng(y[1],sigma_e);     //initial value for autoregressive
##   for (i in 2:N){
##     yp[i] = normal_rng(y[i-1] + beta[bix[i]],sigma_e);  //fitted
##   }
##   for (i in 1:years_ahead){
##     yp[i+N] = normal_rng(yp[i+N-1] + beta[bix[future_beta]],sigma_e);
##   }
## }
```

Appendix 1: Posterior Draw Summary

```r
summary(gp)
```

```
## $summary
##                       mean      se_mean         sd        2.5%        25%
## a               0.49170846 0.0048135831 0.59684534  -0.71430337  0.1098012
## b               1.62812460 0.0036561142 0.42290670   0.94473415  1.3257662
## beta[1]         1.30169069 0.0012261323 0.15460635   0.99619392  1.2015670
## beta[2]         3.25566439 0.0044325393 0.50270579   2.20219531  2.9433401
## beta[3]        -1.74189680 0.0043738173 0.49653437  -2.68951148 -2.0719253
## beta[4]         0.81660723 0.0010400936 0.13220928   0.55656469  0.7314791
## beta[5]         0.05314215 0.0035793749 0.47063644  -0.88630133 -0.2520899
## sigma_beta[1]   0.79510207 0.0046646470 0.59636397   0.03114861  0.3173036
## sigma_beta[2]   0.79764993 0.0047156471 0.60930159   0.02794342  0.3197044
## sigma_beta[3]   0.80651009 0.0046242519 0.60901852   0.03022234  0.3191419
## sigma_beta[4]   0.79858046 0.0046319938 0.60042389   0.03358620  0.3227472
## sigma_beta[5]   0.80844184 0.0046197451 0.60557245   0.03097736  0.3278168
## sigma_e         0.48509818 0.0008205194 0.08047298   0.35741564  0.4279397
## yp[1]          15.12594234 0.0039090238 0.49371443  14.16306427 14.8001835
## yp[2]          16.42313057 0.0040640306 0.51171467  15.41624045 16.0916506
## yp[3]          17.34552611 0.0040176070 0.51530824  16.32328297 17.0094835
## yp[4]          18.53875072 0.0041087570 0.51553181  17.52235919 18.1983626
## yp[5]          20.81058330 0.0040354798 0.51775450  19.77741699 20.4706213
## yp[6]          21.72432911 0.0040266753 0.51447516  20.69804553 21.3874482
## yp[7]          22.92246559 0.0042038303 0.52014809  21.89283553 22.5798500
## yp[8]          25.12358879 0.0039628418 0.50927286  24.11629871 24.7873801
## yp[9]          26.56171053 0.0040641562 0.51787838  25.52270533 26.2293341
## yp[10]         27.03431946 0.0041366915 0.51276150  26.01805883 26.7040617
## yp[11]         30.59546022 0.0060842482 0.70108335  29.18225945 30.1459245
## yp[12]         32.19454479 0.0040196518 0.51017659  31.18635231 31.8630525
## yp[13]         30.06332051 0.0058560831 0.69272549  28.73900268 29.6021833
## yp[14]         30.62867199 0.0039869919 0.50478241  29.63887818 30.2959071
## yp[15]         30.88855023 0.0040384807 0.50940842  29.86374502 30.5589472
## yp[16]         31.31827256 0.0040208228 0.50769415  30.31556597 30.9915536
## yp[17]         31.96605565 0.0039854940 0.50825271  30.98245919 31.6291164
## yp[18]         32.59113255 0.0040480254 0.51477416  31.57120739 32.2563852
## yp[19]         33.28880834 0.0039905249 0.50892460  32.26504649 32.9593424
## yp[20]         34.08533325 0.0040727256 0.51029760  33.06195727 33.7525380
## yp[21]         34.06986525 0.0052147696 0.67868275  32.72608598 33.6247998
## yp[22]         34.83951278 0.0039829825 0.50520341  33.83592054 34.5035896
## yp[23]         36.15892864 0.0040684578 0.51110134  35.16169049 35.8178062
## yp[24]         37.17259965 0.0040281897 0.50910967  36.17469480 36.8400445
## yp[25]         38.25903330 0.0040031233 0.50815084  37.25520600 37.9294025
## yp[26]         38.94895582 0.0040617622 0.51240041  37.96470098 38.6130572
## yp[27]         39.58564138 0.0040523355 0.51085844  38.56806950 39.2523531
## yp[28]         40.53302000 0.0040565326 0.50331462  39.53871488 40.2043275
## yp[29]         41.83190919 0.0057520426 0.72581759  40.39808941 41.3596791
```

```
## yp[30]         43.13000753 0.0071961433 0.91970080  41.32311623 42.5285686
## yp[31]         44.43079158 0.0085295717 1.10226740  42.25086903 43.7124369
## yp[32]         45.72765722 0.0097909301 1.27156486  43.15668060 44.8982208
## yp[33]         47.03609089 0.0111643668 1.44307253  44.13957349 46.0889032
## yp[34]         48.33766181 0.0125313916 1.61288678  45.13148810 47.2819356
## yp[35]         49.64463718 0.0138209731 1.77629357  46.07707312 48.4752125
## yp[36]         50.95172554 0.0152543867 1.93821632  47.07533914 49.6906798
## yp[37]         52.26011057 0.0164301665 2.09762183  48.10215816 50.8801722
## yp[38]         53.57410095 0.0177147486 2.26137677  49.08060734 52.0852585
## yp[39]         54.88070167 0.0190118491 2.42146161  50.03699663 53.3065353
## yp[40]         56.18252927 0.0202693590 2.58128202  51.04930006 54.4968637
## yp[41]         57.47965193 0.0215755087 2.74037538  52.02748600 55.6871087
## lp__           -6.69494350 0.0406174515 2.92016651 -13.35025799 -8.4598275
##                        50%         75%      97.5%     n_eff      Rhat
## a               0.50549146  0.8892012  1.6226353 15373.993 0.9997925
## b               1.58010181  1.8773642  2.5909691 13379.800 1.0000587
## beta[1]         1.30184155  1.4031037  1.6063274 15899.373 0.9998482
## beta[2]         3.27316960  3.5918516  4.1972335 12862.417 0.9999471
## beta[3]        -1.75591898 -1.4277842 -0.7214677 12887.757 1.0000802
## beta[4]         0.81710146  0.9030466  1.0782986 16157.681 0.9998717
## beta[5]         0.05645997  0.3664016  0.9736303 17288.477 1.0001990
## sigma_beta[1]   0.67655478  1.1482805  2.2104021 16345.011 1.0000624
## sigma_beta[2]   0.67007095  1.1569235  2.2470986 16694.831 0.9998581
## sigma_beta[3]   0.69016388  1.1648315  2.2807207 17345.149 1.0000720
## sigma_beta[4]   0.67668639  1.1475628  2.2310647 16802.736 1.0000224
## sigma_beta[5]   0.68318130  1.1641137  2.2652691 17182.889 0.9998525
## sigma_e         0.47562393  0.5301856  0.6674807  9618.835 1.0001317
## yp[1]          15.12812356 15.4476921 16.0966955 15951.995 1.0000644
## yp[2]          16.41560190 16.7611813 17.4293113 15854.107 1.0001191
## yp[3]          17.34379825 17.6778621 18.3621656 16451.264 1.0001673
## yp[4]          18.53608927 18.8809033 19.5598621 15743.091 0.9999282
## yp[5]          20.81413243 21.1525880 21.8241704 16461.045 0.9999841
## yp[6]          21.72604358 22.0625216 22.7229172 16324.339 1.0001576
## yp[7]          22.92540572 23.2665229 23.9523654 15309.593 0.9999486
## yp[8]          25.12583035 25.4570323 26.1379074 16515.343 0.9999650
## yp[9]          26.56627007 26.9023934 27.5667737 16237.336 0.9999869
## yp[10]         27.03367432 27.3650136 28.0578052 15364.717 1.0002047
## yp[11]         30.61705830 31.0651801 31.9286349 13277.780 0.9998687
## yp[12]         32.19758317 32.5275126 33.2056082 16108.837 0.9998267
## yp[13]         30.04586932 30.5101385 31.4601004 13992.904 1.0001958
## yp[14]         30.62605882 30.9600254 31.6248167 16029.417 1.0001596
## yp[15]         30.88571745 31.2218456 31.8924318 15910.954 1.0001862
## yp[16]         31.31081047 31.6495778 32.3303357 15943.162 0.9998501
## yp[17]         31.96402980 32.2951503 32.9969180 16262.791 0.9998893
## yp[18]         32.59291061 32.9286927 33.5895370 16171.378 0.9998473
```

```
## yp[19]        33.29363062 33.6190910 34.3056526 16264.729 1.0000012
## yp[20]        34.08219752 34.4223258 35.0880652 15699.172 0.9999231
## yp[21]        34.07759585 34.5131182 35.3986194 16938.048 0.9998514
## yp[22]        34.83726981 35.1692135 35.8435159 16088.508 1.0002052
## yp[23]        36.16311327 36.4880193 37.1768345 15781.723 0.9998935
## yp[24]        37.17587098 37.5051556 38.1808736 15973.602 1.0001116
## yp[25]        38.26170072 38.5895429 39.2628145 16113.406 0.9999004
## yp[26]        38.94636807 39.2835814 39.9683240 15914.388 0.9999404
## yp[27]        39.58670518 39.9202228 40.5934596 15892.432 0.9998532
## yp[28]        40.53711455 40.8613287 41.5193961 15394.626 0.9999312
## yp[29]        41.83282405 42.3081582 43.2547060 15922.483 0.9999837
## yp[30]        43.13780576 43.7363662 44.9359752 16334.037 0.9999350
## yp[31]        44.42498061 45.1631005 46.6007111 16700.115 1.0000888
## yp[32]        45.73607553 46.5583614 48.2378096 16866.662 1.0001925
## yp[33]        47.04381325 47.9762288 49.8645739 16707.371 1.0002763
## yp[34]        48.34335746 49.3962083 51.5490111 16565.676 1.0002221
## yp[35]        49.64886298 50.7917017 53.1250929 16517.802 1.0001302
## yp[36]        50.95503902 52.1952079 54.7478817 16144.143 1.0000998
## yp[37]        52.26695557 53.6354238 56.3944164 16299.357 1.0000575
## yp[38]        53.58569942 55.0457397 57.9609776 16295.807 0.9999713
## yp[39]        54.89511173 56.4563308 59.6723872 16222.077 0.9999947
## yp[40]        56.21443573 57.8749071 61.2994771 16217.760 0.9999228
## yp[41]        57.48866993 59.2706428 62.9064701 16132.364 0.9999289
## lp__          -6.31898063 -4.5726085 -2.0583424  5168.802 1.0003180
##
## $c_summary
## , , chains = chain:1
##
##                stats
## parameter             mean        sd        2.5%        25%         50%
##   a             0.49660288 0.5956873  -0.70950211  0.1113500  0.51551742
##   b             1.63247815 0.4286152   0.94170348  1.3265975  1.57894392
##   beta[1]       1.30185008 0.1548648   0.99995635  1.2007943  1.30015825
##   beta[2]       3.25022602 0.5029199   2.21816447  2.9247513  3.27181027
##   beta[3]      -1.75179418 0.5041263  -2.73685956 -2.0731184 -1.76435126
##   beta[4]       0.81563014 0.1346713   0.55220181  0.7245409  0.81585619
##   beta[5]       0.04526962 0.4725043  -0.89166418 -0.2693865  0.05319918
##   sigma_beta[1]  0.79521366 0.6109043   0.02395971  0.3019462  0.67298480
##   sigma_beta[2]  0.78870135 0.5978443   0.02624146  0.3260100  0.66235957
##   sigma_beta[3]  0.81124977 0.6062170   0.03141376  0.3348008  0.69907955
##   sigma_beta[4]  0.79158222 0.6057213   0.02926074  0.3118374  0.66545687
##   sigma_beta[5]  0.80365229 0.6012938   0.02692402  0.3214739  0.68392198
##   sigma_e       0.48559889 0.0812688   0.35708617  0.4277249  0.47638447
##   yp[1]        15.12950783 0.4943725  14.14158789 14.7990472 15.13925920
##   yp[2]        16.43216138 0.5197916  15.42025420 16.1005150 16.42461163
```

```
##   yp[3]          17.33839268 0.5116624  16.29748811 17.0139711 17.34582996
##   yp[4]          18.53711364 0.5285899  17.50094715 18.1912913 18.52641261
##   yp[5]          20.80659618 0.5196684  19.77723020 20.4705096 20.81252831
##   yp[6]          21.73573113 0.5157876  20.70602685 21.3903220 21.73870179
##   yp[7]          22.92220412 0.5137312  21.88914875 22.5778655 22.92724973
##   yp[8]          25.11100601 0.5094499  24.08095123 24.7748879 25.11556432
##   yp[9]          26.56508223 0.5061232  25.55848062 26.2421552 26.56809386
##   yp[10]         27.03042020 0.5172149  26.00267736 26.7074477 27.02968460
##   yp[11]         30.59762419 0.7098523  29.21500069 30.1411945 30.60487046
##   yp[12]         32.18739438 0.5114308  31.18737798 31.8445355 32.19454177
##   yp[13]         30.05237849 0.6951497  28.68774340 29.6050350 30.03932976
##   yp[14]         30.63403034 0.5134008  29.62975754 30.3000385 30.62838440
##   yp[15]         30.89625942 0.5085441  29.87441613 30.5590288 30.89262374
##   yp[16]         31.32182532 0.5129915  30.32589975 30.9904893 31.31185901
##   yp[17]         31.97311757 0.5061506  31.01602179 31.6310186 31.98114755
##   yp[18]         32.58443124 0.5185388  31.55466729 32.2449672 32.59158596
##   yp[19]         33.28411848 0.5134556  32.24217435 32.9557437 33.29394540
##   yp[20]         34.08829946 0.5054761  33.04332577 33.7689067 34.08562364
##   yp[21]         34.06970307 0.6782415  32.71746192 33.6244794 34.07931038
##   yp[22]         34.83622346 0.5072586  33.80938799 34.5100330 34.83403419
##   yp[23]         36.15746587 0.5185999  35.16681390 35.8064109 36.15837180
##   yp[24]         37.16217885 0.5054169  36.14115795 36.8382423 37.17911858
##   yp[25]         38.25998772 0.5048358  37.25654766 37.9308672 38.26529479
##   yp[26]         38.93624991 0.5181836  37.90982129 38.6049773 38.94092741
##   yp[27]         39.57731398 0.5173238  38.56086338 39.2338926 39.57171821
##   yp[28]         40.52158024 0.5060934  39.51037822 40.1996028 40.52121703
##   yp[29]         41.81630089 0.7320540  40.37364392 41.3415739 41.81855428
##   yp[30]         43.11181430 0.9199240  41.33826728 42.4951388 43.13833290
##   yp[31]         44.40696711 1.1097170  42.25070886 43.6552796 44.38512623
##   yp[32]         45.69950155 1.2819502  43.14786063 44.8373092 45.68457027
##   yp[33]         47.00979174 1.4479069  44.16817108 46.0394034 46.99144484
##   yp[34]         48.31340340 1.6158141  45.21201375 47.2184808 48.29148754
##   yp[35]         49.62270021 1.7805786  46.15747845 48.4216961 49.62560614
##   yp[36]         50.92600103 1.9470632  47.07453710 49.6148232 50.90928545
##   yp[37]         52.23898979 2.1086805  48.12136955 50.8176972 52.26164134
##   yp[38]         53.55680810 2.2790801  49.15832578 52.0171936 53.56773839
##   yp[39]         54.86470417 2.4282285  50.12495931 53.2218834 54.86658075
##   yp[40]         56.15944730 2.5868048  51.20470053 54.4012770 56.22737663
##   yp[41]         57.46829800 2.7438926  52.04407930 55.6080487 57.49741079
##   lp__           -6.83368331 2.9088221 -13.28082980 -8.6184484 -6.46098310
##                 stats
## parameter              75%      97.5%
##   a              0.8972109  1.5946325
##   b              1.8876346  2.6125028
##   beta[1]        1.4054169  1.6095794
```

```
##   beta[2]         3.5960610  4.1861233
##   beta[3]        -1.4345474 -0.7247567
##   beta[4]         0.9065058  1.0790232
##   beta[5]         0.3562019  0.9764585
##   sigma_beta[1]   1.1393496  2.2543096
##   sigma_beta[2]   1.1303400  2.2597748
##   sigma_beta[3]   1.1612732  2.2759694
##   sigma_beta[4]   1.1611242  2.2411220
##   sigma_beta[5]   1.1553385  2.2456760
##   sigma_e         0.5315373  0.6729631
##   yp[1]          15.4519162 16.0958153
##   yp[2]          16.7651980 17.4721538
##   yp[3]          17.6669059 18.3242937
##   yp[4]          18.8809541 19.5924397
##   yp[5]          21.1474425 21.8307368
##   yp[6]          22.0797883 22.7351160
##   yp[7]          23.2654287 23.9162930
##   yp[8]          25.4524570 26.1333060
##   yp[9]          26.8954294 27.5752681
##   yp[10]         27.3592443 28.0757396
##   yp[11]         31.0861327 31.9670864
##   yp[12]         32.5195078 33.1921610
##   yp[13]         30.4978854 31.4665292
##   yp[14]         30.9619971 31.6694178
##   yp[15]         31.2197491 31.8933389
##   yp[16]         31.6504678 32.3468438
##   yp[17]         32.3084090 33.0060167
##   yp[18]         32.9315953 33.5707148
##   yp[19]         33.6185905 34.2907652
##   yp[20]         34.4171555 35.0610244
##   yp[21]         34.5149748 35.3982679
##   yp[22]         35.1732556 35.8418874
##   yp[23]         36.5054142 37.2076569
##   yp[24]         37.5025589 38.1298582
##   yp[25]         38.5896748 39.2531878
##   yp[26]         39.2696178 39.9633190
##   yp[27]         39.9162459 40.6109385
##   yp[28]         40.8546554 41.5240187
##   yp[29]         42.3095368 43.2672471
##   yp[30]         43.7188476 44.8883098
##   yp[31]         45.1543939 46.6384232
##   yp[32]         46.5316471 48.2260798
##   yp[33]         47.9638660 49.8563878
##   yp[34]         49.3738979 51.5535714
##   yp[35]         50.7565201 53.1842657
```

```
##   yp[36]        52.1805361 54.7960748
##   yp[37]        53.6175321 56.5080037
##   yp[38]        55.0073638 58.1608365
##   yp[39]        56.4417681 59.6815197
##   yp[40]        57.8511137 61.3027516
##   yp[41]        59.2542316 62.9743964
##   lp__          -4.7382678 -2.1573668
##
## , , chains = chain:2
##
##                stats
## parameter              mean         sd        2.5%        25%         50%
##   a              0.48956693 0.60796360  -0.76117445  0.1037093  0.51009597
##   b              1.62939071 0.42167693   0.94069135  1.3330926  1.58378348
##   beta[1]        1.30249475 0.15942307   0.99218115  1.2003422  1.30456488
##   beta[2]        3.25643347 0.50989139   2.15428127  2.9547024  3.27833592
##   beta[3]       -1.73999778 0.47915245  -2.65822589 -2.0636191 -1.75235535
##   beta[4]        0.81527307 0.13111181   0.56096326  0.7311397  0.81586978
##   beta[5]        0.04794926 0.47531065  -0.90315321 -0.2520611  0.05274298
##   sigma_beta[1]  0.80662787 0.58555064   0.04185630  0.3406976  0.68889946
##   sigma_beta[2]  0.79893065 0.61673306   0.03023396  0.3107236  0.67201789
##   sigma_beta[3]  0.80648578 0.61846924   0.03651002  0.3069680  0.68237145
##   sigma_beta[4]  0.80572364 0.59888027   0.03911706  0.3250748  0.68304773
##   sigma_beta[5]  0.80405602 0.62042414   0.02902485  0.3148756  0.67054019
##   sigma_e        0.48525868 0.07958348   0.35460339  0.4287527  0.47633517
##   yp[1]         15.12564769 0.48331165  14.20015637 14.8100579 15.12661301
##   yp[2]         16.41156706 0.50687742  15.39142844 16.0791720 16.39854882
##   yp[3]         17.36816122 0.52205613  16.33813866 17.0193629 17.36481122
##   yp[4]         18.54881126 0.51528727  17.52859971 18.2053923 18.56517206
##   yp[5]         20.80183049 0.51445534  19.77137501 20.4586584 20.80688196
##   yp[6]         21.71625913 0.51324784  20.71776817 21.3748013 21.71291237
##   yp[7]         22.92568555 0.52294458  21.89364483 22.5842972 22.92791756
##   yp[8]         25.12831838 0.51403053  24.09995138 24.7922770 25.12320715
##   yp[9]         26.55850353 0.52855736  25.50044090 26.2150591 26.56791245
##   yp[10]        27.04463570 0.51331649  26.02232721 26.7113235 27.04398208
##   yp[11]        30.60733758 0.70989996  29.13821001 30.1630920 30.64325669
##   yp[12]        32.19801458 0.51082232  31.18635231 31.8653971 32.20018828
##   yp[13]        30.05953990 0.68539162  28.75258840 29.5949590 30.05516749
##   yp[14]        30.61055384 0.50095385  29.60432497 30.2899558 30.60945118
##   yp[15]        30.88828751 0.51559309  29.84810032 30.5609236 30.88466222
##   yp[16]        31.30739152 0.49815806  30.31180185 30.9920413 31.29729146
##   yp[17]        31.96436263 0.51467165  30.96042125 31.6232186 31.96633539
##   yp[18]        32.59355870 0.51843975  31.57641822 32.2584513 32.59217882
##   yp[19]        33.28054810 0.50704443  32.24023959 32.9609455 33.28860566
##   yp[20]        34.07950977 0.51296361  33.07814391 33.7415041 34.07396695
```

```
##   yp[21]        34.06423178 0.68652661  32.70159812 33.6160662 34.06802886
##   yp[22]        34.83040476 0.50263882  33.82248275 34.4960009 34.82954275
##   yp[23]        36.16493848 0.49794630  35.17697517 35.8371035 36.17111893
##   yp[24]        37.17203028 0.50327811  36.16308550 36.8406285 37.17509771
##   yp[25]        38.26105252 0.50947934  37.23525996 37.9400663 38.26439338
##   yp[26]        38.95907400 0.51893899  37.97409299 38.6237413 38.95515396
##   yp[27]        39.58769141 0.50695237  38.57859638 39.2504223 39.58615737
##   yp[28]        40.53780422 0.50464853  39.55026245 40.2150648 40.54396795
##   yp[29]        41.83925419 0.74363185  40.31712728 41.3655121 41.83575670
##   yp[30]        43.13607743 0.93058678  41.29566850 42.5352664 43.15251213
##   yp[31]        44.43082564 1.11714241  42.19448934 43.7203576 44.42614863
##   yp[32]        45.73059502 1.28243199  43.12163626 44.9301630 45.74414655
##   yp[33]        47.04885181 1.44846317  44.09001343 46.1117196 47.06801118
##   yp[34]        48.34540190 1.62778701  45.04377462 47.2820436 48.37327816
##   yp[35]        49.65475504 1.79473627  46.04565404 48.4887138 49.68350901
##   yp[36]        50.95096470 1.95432723  47.00183132 49.6628989 50.95978326
##   yp[37]        52.24630423 2.11101752  48.04933541 50.8711980 52.25325386
##   yp[38]        53.55968555 2.27693544  49.02592189 52.0727898 53.58153218
##   yp[39]        54.87512362 2.44282557  49.99502328 53.3116533 54.86521683
##   yp[40]        56.17476348 2.60343190  50.94284470 54.4721774 56.16742667
##   yp[41]        57.47408383 2.77392774  51.98084273 55.6857634 57.44091646
##   lp__          -6.69530374 2.90459339 -13.12212904 -8.4325349 -6.37036548
##               stats
## parameter              75%      97.5%
##   a              0.8923096  1.6520132
##   b              1.8788938  2.6041779
##   beta[1]        1.4065282  1.6094889
##   beta[2]        3.5827783  4.2357550
##   beta[3]       -1.4304667 -0.7730103
##   beta[4]        0.9011262  1.0684183
##   beta[5]        0.3691000  0.9637374
##   sigma_beta[1]  1.1565006  2.2078309
##   sigma_beta[2]  1.1587279  2.3079431
##   sigma_beta[3]  1.1667759  2.3145648
##   sigma_beta[4]  1.1493987  2.2557734
##   sigma_beta[5]  1.1598853  2.3156262
##   sigma_e        0.5293667  0.6663751
##   yp[1]         15.4405051 16.0689253
##   yp[2]         16.7595575 17.3789454
##   yp[3]         17.7061927 18.3957997
##   yp[4]         18.8960717 19.5492280
##   yp[5]         21.1443891 21.7868487
##   yp[6]         22.0569155 22.7141488
##   yp[7]         23.2723589 23.9709869
##   yp[8]         25.4638363 26.1252055
```

```
##   yp[9]          26.9163098 27.5529391
##   yp[10]         27.3781383 28.0562924
##   yp[11]         31.0871453 31.9116620
##   yp[12]         32.5251503 33.2068149
##   yp[13]         30.5052058 31.4294821
##   yp[14]         30.9365233 31.5889397
##   yp[15]         31.2233372 31.8909877
##   yp[16]         31.6341449 32.2634714
##   yp[17]         32.2906747 33.0170785
##   yp[18]         32.9274818 33.6050304
##   yp[19]         33.6074498 34.2820509
##   yp[20]         34.4180478 35.1240763
##   yp[21]         34.5210215 35.4074894
##   yp[22]         35.1517343 35.8256415
##   yp[23]         36.4875122 37.1371136
##   yp[24]         37.5136388 38.1498707
##   yp[25]         38.5902319 39.2654175
##   yp[26]         39.2916782 39.9714914
##   yp[27]         39.9207896 40.5836018
##   yp[28]         40.8538184 41.5232548
##   yp[29]         42.3143463 43.2872279
##   yp[30]         43.7437858 44.9491897
##   yp[31]         45.1664219 46.6333187
##   yp[32]         46.5597645 48.2520506
##   yp[33]         48.0025926 49.8453632
##   yp[34]         49.4240282 51.6034335
##   yp[35]         50.8346024 53.1873600
##   yp[36]         52.1965769 54.7985068
##   yp[37]         53.6209367 56.3783981
##   yp[38]         55.0591145 57.9435738
##   yp[39]         56.4678205 59.7151543
##   yp[40]         57.9200550 61.2627426
##   yp[41]         59.3097791 62.9553711
##   lp__           -4.5840898 -2.0588655
##
## , , chains = chain:3
##
##               stats
## parameter            mean         sd        2.5%        25%        50%
##   a             0.4851602 0.57510930 -0.68035897  0.1128775  0.50116852
##   b             1.6175076 0.42237777  0.93911510  1.3118329  1.56834096
##   beta[1]       1.3022003 0.15190550  1.00246896  1.2028127  1.30350408
##   beta[2]       3.2534519 0.50166207  2.19730381  2.9393650  3.27488232
##   beta[3]      -1.7366281 0.50707642 -2.68085050 -2.0843123 -1.75134603
##   beta[4]       0.8180366 0.13237624  0.55613309  0.7340170  0.81882002
```

```
## beta[5]          0.0522622 0.46754344  -0.88314789 -0.2503963  0.05148591
## sigma_beta[1]   0.7818616 0.59439086   0.02835225  0.3068867  0.66121771
## sigma_beta[2]   0.7972130 0.60369513   0.03026089  0.3262950  0.68060566
## sigma_beta[3]   0.8044288 0.60625874   0.03202215  0.3209992  0.68618978
## sigma_beta[4]   0.7970580 0.59549317   0.03628308  0.3322042  0.67373408
## sigma_beta[5]   0.8215609 0.60855422   0.03632502  0.3352392  0.69687115
## sigma_e         0.4855697 0.08006281   0.35670786  0.4290461  0.47624725
## yp[1]          15.1130471 0.50720559  14.12585087 14.7764093 15.10532815
## yp[2]          16.4229369 0.52024130  15.41623278 16.0763119 16.41177568
## yp[3]          17.3425064 0.50910306  16.32863946 17.0161112 17.33758342
## yp[4]          18.5337727 0.50938616  17.53047642 18.2032826 18.52675042
## yp[5]          20.8261958 0.52178073  19.79601113 20.4819384 20.82378189
## yp[6]          21.7126673 0.52516553  20.66243656 21.3689373 21.70582969
## yp[7]          22.9134736 0.52489283  21.88337342 22.5655696 22.91697074
## yp[8]          25.1239003 0.50999926  24.13206165 24.7887424 25.13113009
## yp[9]          26.5635927 0.51933739  25.48380640 26.2419828 26.57166389
## yp[10]         27.0316381 0.50908702  26.03492554 26.6983440 27.03045074
## yp[11]         30.5916943 0.68797726  29.20193487 30.1533753 30.61772008
## yp[12]         32.1980101 0.50562193  31.20582543 31.8712000 32.19163510
## yp[13]         30.0768213 0.70912775  28.73663865 29.6049998 30.04644408
## yp[14]         30.6352630 0.50453653  29.63309091 30.3034498 30.64118813
## yp[15]         30.8797429 0.50312014  29.88883181 30.5516652 30.86798658
## yp[16]         31.3238107 0.50862875  30.31361444 30.9950604 31.31865688
## yp[17]         31.9628312 0.50679834  30.96662965 31.6283826 31.95023011
## yp[18]         32.5972385 0.51724976  31.58450142 32.2643529 32.59871976
## yp[19]         33.3000843 0.51670439  32.25955876 32.9612637 33.30171020
## yp[20]         34.0818264 0.50607500  33.08283336 33.7440159 34.07867475
## yp[21]         34.0662770 0.67712101  32.69599660 33.6188784 34.07827861
## yp[22]         34.8544197 0.50411575  33.86150030 34.5120200 34.85816993
## yp[23]         36.1617068 0.51499434  35.18364126 35.8130132 36.16936862
## yp[24]         37.1774497 0.51176278  36.17967641 36.8446942 37.17575548
## yp[25]         38.2624067 0.51221666  37.26578916 37.9347646 38.26221312
## yp[26]         38.9514340 0.50494853  37.98241966 38.6157294 38.94210789
## yp[27]         39.5935503 0.50351074  38.58703026 39.2643457 39.60172561
## yp[28]         40.5337946 0.49885652  39.54532874 40.1938752 40.54388564
## yp[29]         41.8330715 0.71040505  40.43223959 41.3615325 41.83851660
## yp[30]         43.1372802 0.91406902  41.34381634 42.5420555 43.14045705
## yp[31]         44.4485154 1.09225184  42.27840836 43.7410373 44.46334670
## yp[32]         45.7520752 1.26274134  43.22352537 44.9131502 45.77294195
## yp[33]         47.0575720 1.43319363  44.17327279 46.1233737 47.06771360
## yp[34]         48.3722261 1.59759890  45.19310644 47.3152671 48.37063288
## yp[35]         49.6756919 1.76081401  46.12400556 48.5270759 49.65361033
## yp[36]         50.9932836 1.91061239  47.20914890 49.7605810 50.99487102
## yp[37]         52.2928154 2.06847845  48.12798499 50.9655311 52.28901906
## yp[38]         53.6045856 2.23220271  49.16880445 52.1732757 53.59701160
```

```
##   yp[39]        54.9211469 2.39329317  50.07881767 53.3902529 54.95494764
##   yp[40]        56.2228167 2.56320636  51.14412387 54.5850712 56.25046404
##   yp[41]        57.5224037 2.71468723  52.20299582 55.7663703 57.54942604
##   lp__          -6.5992123 2.88377939 -13.33917856 -8.2838523 -6.20084581
##                stats
## parameter              75%      97.5%
##   a              0.8679995  1.5765884
##   b              1.8650733  2.5513523
##   beta[1]        1.4015921  1.6006066
##   beta[2]        3.5982486  4.1848177
##   beta[3]       -1.4267632 -0.6649285
##   beta[4]        0.9015504  1.0875938
##   beta[5]        0.3674529  0.9784665
##   sigma_beta[1]  1.1193752  2.1993642
##   sigma_beta[2]  1.1653014  2.2003833
##   sigma_beta[3]  1.1542171  2.2682018
##   sigma_beta[4]  1.1262268  2.1987531
##   sigma_beta[5]  1.1819176  2.2708090
##   sigma_e        0.5310520  0.6621654
##   yp[1]         15.4440479 16.1329979
##   yp[2]         16.7684397 17.4594948
##   yp[3]         17.6746212 18.3345435
##   yp[4]         18.8681929 19.5690274
##   yp[5]         21.1707827 21.8411952
##   yp[6]         22.0474539 22.7583296
##   yp[7]         23.2619285 23.9393520
##   yp[8]         25.4467823 26.1592963
##   yp[9]         26.9011024 27.5795516
##   yp[10]        27.3489829 28.0559046
##   yp[11]        31.0398102 31.9365441
##   yp[12]        32.5276780 33.2339131
##   yp[13]        30.5420287 31.4898682
##   yp[14]        30.9646800 31.6186045
##   yp[15]        31.2108080 31.8711926
##   yp[16]        31.6575345 32.3373004
##   yp[17]        32.2899321 33.0049024
##   yp[18]        32.9333858 33.6307988
##   yp[19]        33.6304761 34.3532058
##   yp[20]        34.4159213 35.0738691
##   yp[21]        34.5058340 35.3719488
##   yp[22]        35.1852586 35.8750680
##   yp[23]        36.4876248 37.1994143
##   yp[24]        37.5032160 38.1982609
##   yp[25]        38.5897776 39.3003940
##   yp[26]        39.2844661 39.9784991
```

```
##   yp[27]        39.9282482 40.5631608
##   yp[28]        40.8670939 41.4957020
##   yp[29]        42.3005565 43.2160459
##   yp[30]        43.7365967 44.9661772
##   yp[31]        45.1644867 46.5307920
##   yp[32]        46.5806275 48.2395454
##   yp[33]        47.9819826 49.8396090
##   yp[34]        49.4243149 51.5238875
##   yp[35]        50.8173947 53.1223344
##   yp[36]        52.2367346 54.7207395
##   yp[37]        53.6406338 56.2791750
##   yp[38]        55.0715040 57.8759724
##   yp[39]        56.4912793 59.6053482
##   yp[40]        57.8836991 61.2326485
##   yp[41]        59.2776332 62.8728614
##   lp__          -4.4386019 -2.1196360
##
## , , chains = chain:4
##
##                stats
## parameter              mean         sd        2.5%        25%         50%
##   a              0.49550388 0.60816528 -0.70831156  0.1112731  0.49716768
##   b              1.63312194 0.41886847  0.96345761  1.3291115  1.58535328
##   beta[1]        1.30021761 0.15216185  0.99649261  1.2032544  1.29965342
##   beta[2]        3.26254612 0.49636388  2.25007387  2.9556333  3.26958034
##   beta[3]       -1.73916719 0.49535522 -2.67716852 -2.0718263 -1.75761011
##   beta[4]        0.81748907 0.13066977  0.55601600  0.7339253  0.81676351
##   beta[5]        0.06708753 0.46701083 -0.85620770 -0.2410429  0.06905701
##   sigma_beta[1]  0.79670517 0.59429105  0.03053214  0.3181044  0.68185128
##   sigma_beta[2]  0.80575470 0.61878858  0.02651457  0.3181332  0.66787272
##   sigma_beta[3]  0.80387605 0.60523201  0.02162891  0.3168688  0.69162145
##   sigma_beta[4]  0.79995801 0.60169232  0.03143551  0.3238429  0.68642466
##   sigma_beta[5]  0.80449811 0.59169128  0.03895073  0.3396540  0.68439507
##   sigma_e        0.48396549 0.08098443  0.36148425  0.4261563  0.47297995
##   yp[1]         15.13556678 0.48956620 14.18226555 14.8166027 15.14134253
##   yp[2]         16.42585692 0.49962212 15.43492326 16.1103314 16.42752195
##   yp[3]         17.33304411 0.51779754 16.29874053 16.9961670 17.32730754
##   yp[4]         18.53530528 0.50867319 17.52921612 18.1982740 18.52699567
##   yp[5]         20.80771076 0.51493761 19.77274209 20.4784908 20.81000559
##   yp[6]         21.73265887 0.50326700 20.71652710 21.4125187 21.74405788
##   yp[7]         22.92849909 0.51902613 21.91389610 22.5895500 22.92920549
##   yp[8]         25.13113048 0.50351399 24.14817535 24.7915911 25.13746780
##   yp[9]         26.55966368 0.51741564 25.54442889 26.2243508 26.55682727
##   yp[10]        27.03058382 0.51144619 26.01783582 26.6989512 27.03223093
##   yp[11]        30.58518477 0.69642922 29.17286579 30.1343273 30.60168169
```

```
##   yp[12]        32.19476006 0.51291932  31.17053459 31.8675320 32.20325616
##   yp[13]        30.06454236 0.68092551  28.79572910 29.6019990 30.04239547
##   yp[14]        30.63484078 0.49988145  29.69470231 30.2924917 30.62484170
##   yp[15]        30.88991109 0.51035310  29.83144292 30.5640989 30.89020119
##   yp[16]        31.32006273 0.51089809  30.30634800 30.9868827 31.31302858
##   yp[17]        31.96391121 0.50545958  30.98693161 31.6358622 31.95959901
##   yp[18]        32.58930173 0.50484400  31.59662482 32.2615453 32.58883441
##   yp[19]        33.29048250 0.49827393  32.31420279 32.9595381 33.29040711
##   yp[20]        34.09169741 0.51668628  33.06046153 33.7555517 34.09268267
##   yp[21]        34.07924915 0.67292701  32.79667384 33.6350126 34.08385363
##   yp[22]        34.83700322 0.50665701  33.86597474 34.4963009 34.83315702
##   yp[23]        36.15160342 0.51271732  35.12219977 35.8170019 36.15530170
##   yp[24]        37.17873977 0.51590566  36.19429111 36.8376956 37.17085396
##   yp[25]        38.25268632 0.50617374  37.25813388 37.9163668 38.25026635
##   yp[26]        38.94906543 0.50730530  37.99586352 38.6092574 38.94632312
##   yp[27]        39.58400986 0.51557216  38.55625820 39.2601166 39.58702658
##   yp[28]        40.53890097 0.50363174  39.54713039 40.2120965 40.54370494
##   yp[29]        41.83901018 0.71674527  40.44245291 41.3727760 41.83641099
##   yp[30]        43.13485817 0.91422883  41.33567549 42.5425350 43.12130966
##   yp[31]        44.43685818 1.08971243  42.30465403 43.7350106 44.43235944
##   yp[32]        45.72845717 1.25888020  43.17049904 44.9162837 45.74018794
##   yp[33]        47.02814797 1.44273778  44.11167616 46.0858902 47.05059988
##   yp[34]        48.31961579 1.61013336  45.08990380 47.3123715 48.34530082
##   yp[35]        49.62540153 1.76898822  46.05952008 48.4901239 49.62576064
##   yp[36]        50.93665283 1.94062910  47.02088883 49.7143445 50.94229274
##   yp[37]        52.26233280 2.10240983  48.08285148 50.8906614 52.28484959
##   yp[38]        53.57532451 2.25750484  49.00179751 52.1252565 53.59923608
##   yp[39]        54.86183199 2.42166918  49.93080174 53.2998627 54.88043713
##   yp[40]        56.17308958 2.57202580  50.96444153 54.5236912 56.17954764
##   yp[41]        57.45382224 2.72918648  51.93604355 55.6862193 57.45687069
##   lp__          -6.65157460 2.97850514 -13.63674476 -8.4261583 -6.26835120
##                stats
## parameter             75%      97.5%
##   a             0.8980643  1.6631155
##   b             1.8821675  2.5733262
##   beta[1]       1.3993566  1.6025547
##   beta[2]       3.5923421  4.1990906
##   beta[3]      -1.4163788 -0.7018594
##   beta[4]       0.9020889  1.0747862
##   beta[5]       0.3754797  0.9569276
##   sigma_beta[1]  1.1625516  2.1787979
##   sigma_beta[2]  1.1702019  2.2670906
##   sigma_beta[3]  1.1761428  2.2548206
##   sigma_beta[4]  1.1487584  2.2362762
##   sigma_beta[5]  1.1583092  2.2051803
```

```
##   sigma_e        0.5289604  0.6736146
##   yp[1]         15.4533491 16.0839332
##   yp[2]         16.7516965 17.4129005
##   yp[3]         17.6604515 18.3696461
##   yp[4]         18.8821926 19.5243298
##   yp[5]         21.1490420 21.8195355
##   yp[6]         22.0707534 22.6784181
##   yp[7]         23.2666675 23.9799838
##   yp[8]         25.4645336 26.1300804
##   yp[9]         26.8972107 27.5597054
##   yp[10]        27.3768684 28.0278977
##   yp[11]        31.0489541 31.8952737
##   yp[12]        32.5393268 33.1834318
##   yp[13]        30.5016433 31.4593471
##   yp[14]        30.9709373 31.6240231
##   yp[15]        31.2284184 31.9144163
##   yp[16]        31.6494314 32.3721265
##   yp[17]        32.2924355 32.9627481
##   yp[18]        32.9248834 33.5654413
##   yp[19]        33.6199590 34.2649360
##   yp[20]        34.4356961 35.1103694
##   yp[21]        34.5117405 35.4429675
##   yp[22]        35.1676004 35.8432610
##   yp[23]        36.4719576 37.1725825
##   yp[24]        37.5030132 38.2349579
##   yp[25]        38.5877028 39.2415995
##   yp[26]        39.2862757 39.9519435
##   yp[27]        39.9150073 40.6016440
##   yp[28]        40.8689486 41.5246017
##   yp[29]        42.3102801 43.2415264
##   yp[30]        43.7432164 44.9219334
##   yp[31]        45.1705474 46.5993552
##   yp[32]        46.5394257 48.2329790
##   yp[33]        47.9450948 49.8848690
##   yp[34]        49.3537973 51.5102000
##   yp[35]        50.7811087 53.0644131
##   yp[36]        52.1789632 54.6381074
##   yp[37]        53.6606067 56.4809869
##   yp[38]        55.0554158 57.9426807
##   yp[39]        56.4144059 59.6773691
##   yp[40]        57.8514650 61.3345376
##   yp[41]        59.2363811 62.8958456
##   lp__          -4.5180228 -1.9481994
```