CVEN 6833 - Homework 3

Alex Belenguer

Contents

Topics

- Parametric/Nonparametric Time Series
- Hidden Markov Model
- Wavelet Spectral Analysis
- Extreme Value Time Series
- Copulas

Chapter 1

Seasonal AR(1) model

Fit a seasonal AR(1) model – i.e., nonstationary time series model to the monthly Colorado River Flow at Lees Ferry.

1.1 Generate 250 simulations each of same length as the historical data.

```
# Load libraries
libr=c("magrittr","sm","stats","moments")
options(warn=1)
suppressPackageStartupMessages(lapply(libr, require, character.only = TRUE))
```

```
# import and set up flow data
flow = read.table(
  "http://civil.colorado.edu/~balajir/CVEN6833/HWs/HW-3-2018/LeesFerry-monflows-1906-2016.txt")

flow = flow[,2:13] %>% `rownames<-`(flow[,1]) %>%
  setNames(.,c("jan","feb","mar","apr","may","jun",
                      "jul","aug","sep","oct","nov","dec")) %>%
                      {./10^6} # convert AF to MAF
head(flow,n=1L) # show values
```

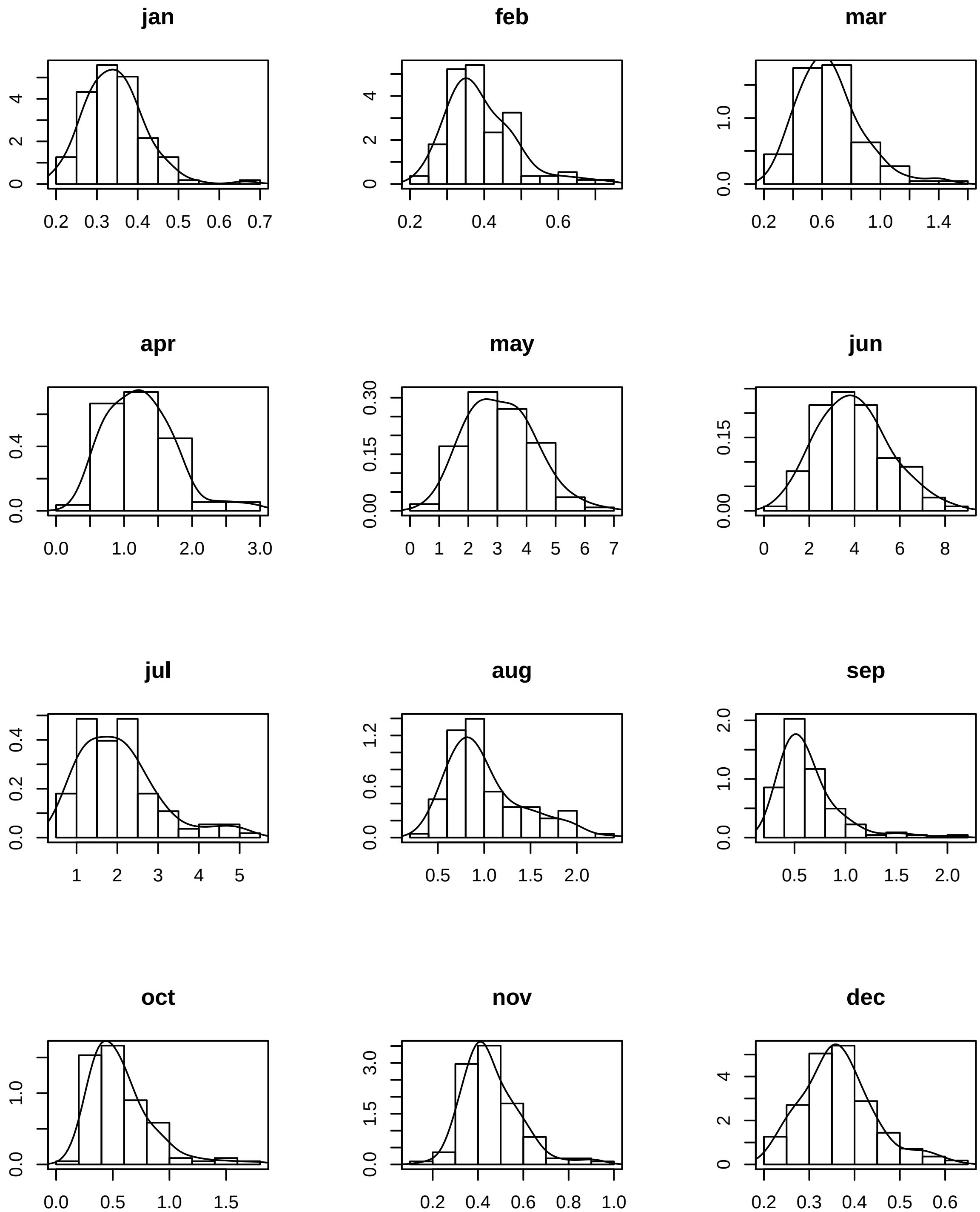
```
##           jan      feb      mar     apr      may      jun     jul      aug
## 1906 0.244314 0.292534 0.678174 1.20464 3.635101 5.014167 2.95046 1.605086
##           sep      oct      nov      dec
## 1906 1.503159 0.739807 0.503006 0.353312
```

```
tail(flow,n=1L)
```

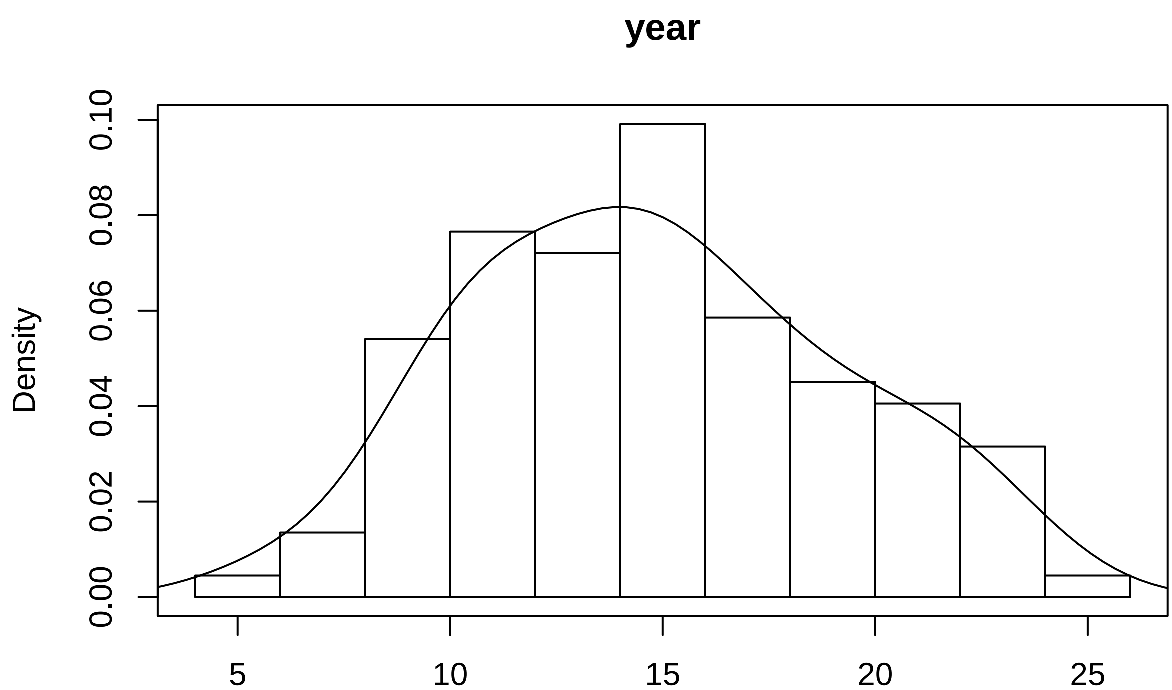
```
##           jan      feb     mar      apr      may      jun      jul
## 2016 0.360703 0.448837 0.67914 1.099567 2.967581 3.910287 1.342044
##           aug      sep      oct      nov      dec
## 2016 0.609946 0.485507 0.546633 0.426289 0.345163
```

```
flow$year = rowSums(flow) # add year in 13th column

par(mfrow=c(4,3)) # plot histogram and density of monthly flow
for(i in 1:12){
  hist(flow[,i], freq=FALSE,
       main = colnames(flow)[i],xlab = "",ylab = "")
  sm.density(flow[,i], add=TRUE)
  }
```




```r
par(mfrow=c(1,1)) # plot histogram and density of anual flow
  hist(flow[,13], freq=FALSE,
             main = colnames(flow)[13],xlab = "")
  sm.density(flow[,13], add=TRUE)
```



The seasonal AR model is fitted using the Thomas Fiering coefficients. The anual flow AR model is fitted using *stats::arima*.

```
# Get the parameters of the Thomas Fiering Model (12 models, 1 for each transition)
coef1 = coef2 = rep(0,length.out = 12)

coef1[1] = cor(flow[-1,"jan"],flow[-111,"dec"]) # jan - dec
coef2[1] = sqrt((var(flow[,1])) * (1. - coef1[1]*coef1[1]))

for(i in 2:12){ # remaining month pairs
    coef1[i] = cor(flow[,i],flow[,i-1])
    coef2[i] = sqrt((var(flow[,i])) * (1. - coef1[i]*coef1[i]))
}

# The anual flow is modeled using single AR(1) model
ar.year=ar(flow$year,order.max = 1) #AR order 1, MA
```

The 12 pairs of the TF coefficients are used to run 250 simulations (synthetic values) that will populate the statistics for the models. For the anual flow model, the simulations are obtained via *stats::arima.sim*.

The random gamma values are related to the normal following the relationship explained in the thread below: https://stats.stackexchange.com/questions/37461/the-relationship-between-the-gamma-distribution-and-the-normal-distrib

```
innovation = "Normal" # defines nature of innovations: "Normal" or "Gamma"

# parameters for equivalent gamma distribution
```

```
# N(x;0,s) ~ lim(a -> +inf) G((a-1)*sqrt(1/a)*s;a,sqrt(1/a)*s)
a=5
sm=1 #sd for monthly innovations
sy=sd(flow$year) #sd for monthly innovations

# peak density for each month
peak=rep(NA,12)
for(i in 1:12){
  aux=sm.density(flow[,i],display="none")
  peak[i]=aux$eval.points[which.max(aux$estimate)]*1.5
}
```

```
# Simulations (innovation defines st. distribution of error)
nsim=250    # number of simulations
nyrs=length(flow[,1]) # years

armean=matrix(0,nsim,12)    #matrices that store the statistics
arstdev=matrix(0,nsim,12)
arcor=matrix(0,nsim,12)
arskw=matrix(0,nsim,12)
armax=matrix(0,nsim,12)
armin=matrix(0,nsim,12)

ar.year.stat=matrix(NA,ncol = 6,nrow = nsim) # year statistics
colnames(ar.year.stat) = c("mean","stdev","min","max","skew","cor")

# Points where May PDF is evaluated
xeval=seq(min(flow$may)-0.25*sd(flow$may),
          max(flow$may)+0.25*sd(flow$may),length=100)
simpdf=matrix(0,nrow=nsim,ncol=100) # Array to store May simulated PDF

# Points where anual PDF is evaluated
yeval=seq(min(flow$year)-0.25*sd(flow$year),
          max(flow$year)+0.25*sd(flow$year),length=100)
year.pdf=matrix(0,nrow=nsim,ncol=100) # Array to store anual simulated PDF

for(k in 1:nsim){
  nmons=nyrs*12 #number of values to be generated
  xsim=1:nmons
  r=sample(1:nyrs,1)

  xsim[1]=flow[r,1] # Starting point for sim
  xprev=xsim[1]

  for(i in 2:nmons){
    j=i %% 12
    if(j == 0) j=12
    j1=j-1

    if(j == 1) j1=12

    x1=xprev-ifelse(innovation=="Normal",mean(flow[,j1]),peak[j1])
    x2=coef2[j]*ifelse(innovation=="Normal",rnorm(1,0,1),
```

```
      rgamma(1,shape=a,scale=sqrt(1/a)*sm)-(a-1)*sqrt(1/a)*sm)

    xsim[i]=mean(flow[,j]) + x1*coef1[j] + x2
    xprev=xsim[i]
  }

  #Store simulated values in matrix form, get May values and PDF
  simdismon=matrix(xsim,ncol = 12, byrow = TRUE) # filled by row
  maysim = simdismon[,5] # Synthetic values for May
  simpdf[k,]=sm.density(maysim,eval.points=xeval,display="none")$estimate

  # Fill statistics for each month
  for(j in 1:12){
    armean[k,j]=mean(simdismon[,j])
    armax[k,j]=max(simdismon[,j])
    armin[k,j]=min(simdismon[,j])
    arstdev[k,j]=sd(simdismon[,j])
    arskw[k,j]=skewness(simdismon[,j])
  }
  arcor[k,1]=cor(simdismon[-nyrs,12],simdismon[2:nyrs,1]) #cor dec-jan
  for(j in 2:12){ # rest of pairs
    j1=j-1
    arcor[k,j]=cor(simdismon[,j],simdismon[,j1])
  }

  # anual flow simulations
  if(innovation=="Normal"){
    ar.year.sim = arima.sim(n = nyrs, list(ar = ar.year$ar),
                          sd = sqrt(ar.year$var.pred)) +
      mean(flow$year)
  }else{
    ar.year.sim = arima.sim(n = nyrs, list(ar = ar.year$ar),
      rand.gen = function(n, ...) rgamma(n,shape=a,
      scale=sqrt(1/a)*sy)-(a-1)*sqrt(1/a)*sy) +
      yeval[which.max(year.density)]*0.85

  }
  # Get anual PDF
  year.pdf[k,]=sm.density(ar.year.sim,eval.points=
                          yeval,display="none")$estimate

  # Calculate statistics
  ar.year.stat[k,"mean"]=mean(ar.year.sim)
  ar.year.stat[k,"max"]=max(ar.year.sim)
  ar.year.stat[k,"min"]=min(ar.year.sim)
  ar.year.stat[k,"stdev"]=sd(ar.year.sim)
  ar.year.stat[k,"skew"]=skewness(ar.year.sim)
  ar.year.stat[k,"cor"]=cor(ar.year.sim[-nyrs],ar.year.sim[2:nyrs])
}
```

The statistics from the synthetic values and the historical data are bound in the same matrix.

```r
# Compute statistics from the historical data.
obsmean=1:12
obsstdev=1:12
obscor=1:12
obsskw=1:12
obsmax=1:12
obsmin=1:12

for(i in 1:12){
  obsmax[i]=max(flow[,i])
  obsmin[i]=min(flow[,i])
  obsmean[i]=mean(flow[,i])
  obsstdev[i]=sd(flow[,i])
    obsskw[i]=skewness(flow[,i])
}

obscor[1]= cor(flow[-nyrs,12], flow[2:nyrs,1])
for(i in 2:12){
  i1=i-1
  obscor[i]=cor(flow[,i], flow[,i1])
}

# bind the stats of the historic data at the top..
armean=rbind(obsmean,armean)
arstdev=rbind(obsstdev,arstdev)
arskw=rbind(obsskw,arskw)
arcor=rbind(obscor,arcor)
armax=rbind(obsmax,armax)
armin=rbind(obsmin,armin)

# anual flow binding
year.stat=c(mean(flow$year),sd(flow$year),min(flow$year),
            max(flow$year),skewness(flow$year),
            cor(flow$year[-nyrs],flow$year[2:nyrs]))

ar.year.stat = rbind(year.stat,ar.year.stat)
```

1.2 Plot statistics from simulations

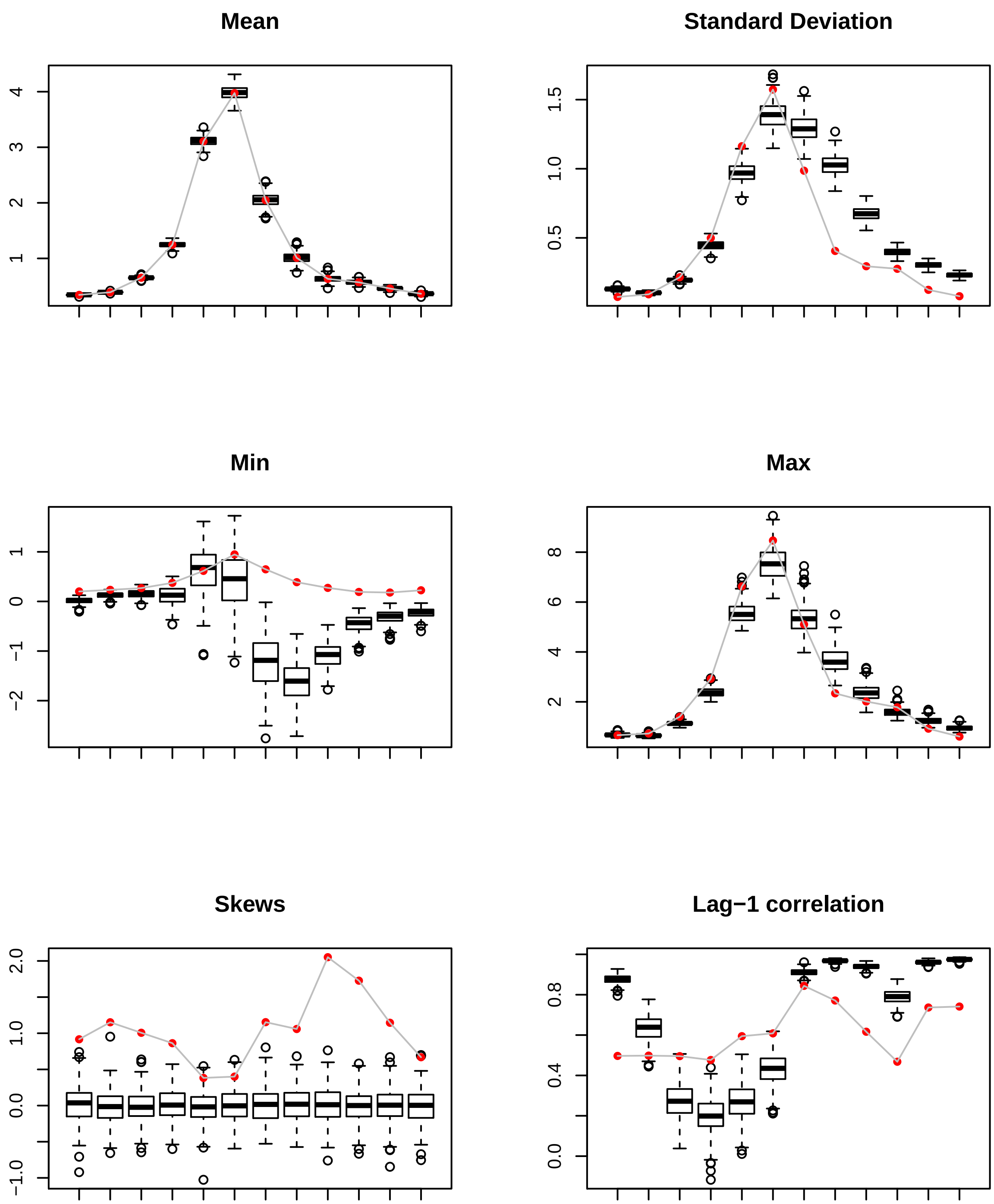
- **Create boxplots of annual and monthly, mean, variance, skew, lag-1 correlation, minimum, maximum and PDFs of May and annual flows. Comment on what you observe and also on why some of the monthly statistics are not captured.**

```r
# function to plot boxplots with the structure: hist. in first row
plot.bp = function(matrix,name){
  xmeans=as.matrix(matrix)
  n=length(xmeans[,1])
  xmeans1=as.matrix(xmeans[2:n,]) #the first row is the original data
  xs=1:12
  zz=boxplot(split(xmeans1,col(xmeans1)), plot=F, cex=1.0)
  zz$names=rep("",length(zz$names))
  z1=bxp(zz,ylim=range(xmeans),xlab="",ylab="",cex=1.00)
```

```
  points(z1,xmeans[1,],pch=16, col="red")
  lines(z1,xmeans[1,],pch=16, col="gray")
  title(main=name)
}
```

The plots for the statistics of the simulated time series (shown as boxplots) vs. the historical data (shown as points and lines) are reproduced below:

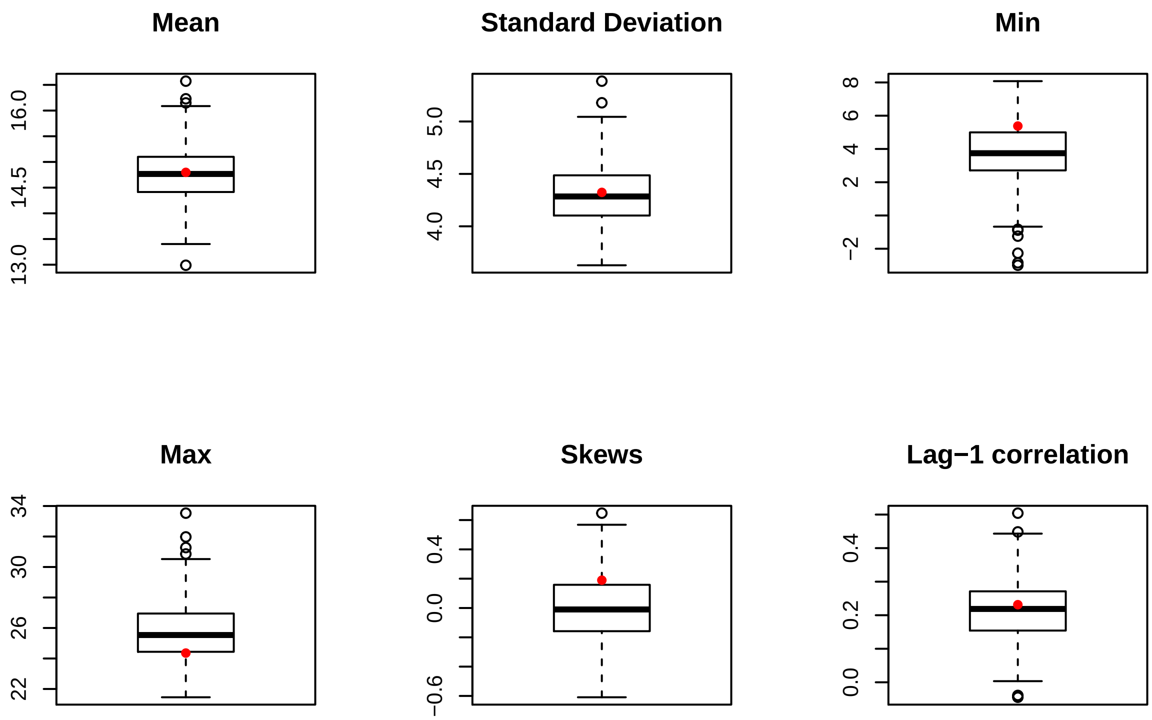
```
par(mfrow=c(3,2))
plot.bp(armean,"Mean")
plot.bp(arstdev,"Standard Deviation")
plot.bp(armin,"Min")
plot.bp(armax,"Max")
plot.bp(arskw,"Skews")
plot.bp(arcor,"Lag-1 correlation")
```



The model proficiently captures the mean and max values. A fair fit is obtained with the standard deviation. However, the normality of the innovations results in a poor fit of minimum values and skews.

The anual statistics are similarly represented below:

```
par(mfrow=c(2,3))
plot.bp(ar.year.stat[,"mean"],"Mean")
plot.bp(ar.year.stat[,"stdev"],"Standard Deviation")
plot.bp(ar.year.stat[,"min"],"Min")
plot.bp(ar.year.stat[,"max"],"Max")
plot.bp(ar.year.stat[,"skew"],"Skews")
plot.bp(ar.year.stat[,"cor"],"Lag-1 correlation")
```



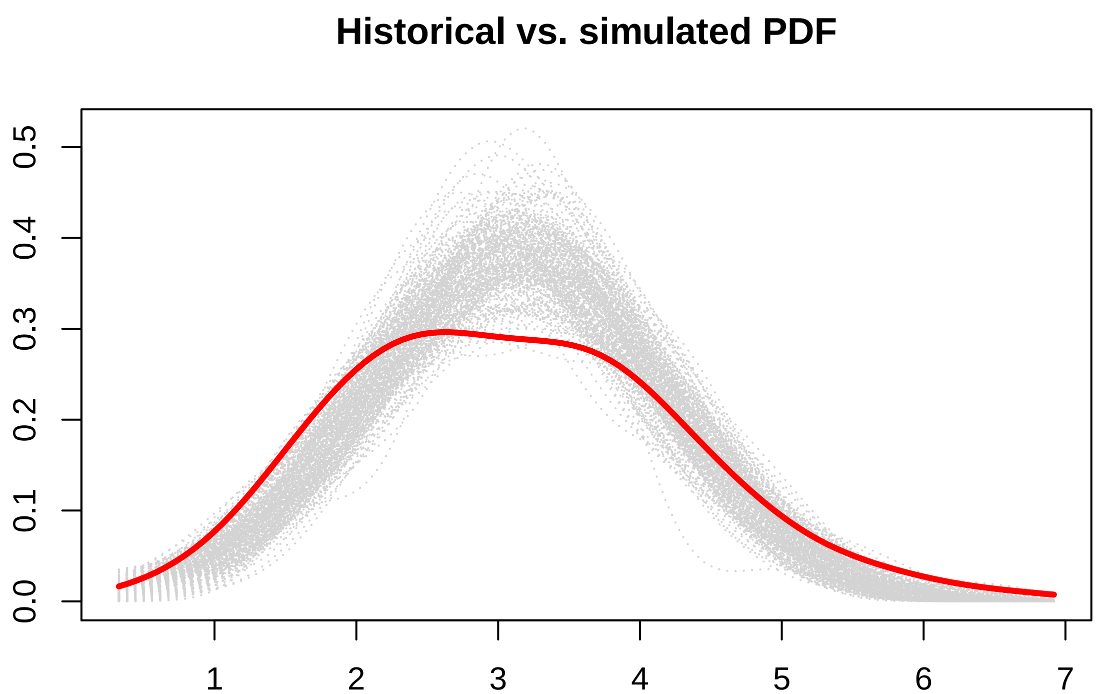
The best fitting occurs for mean, sd, and correlation. Min, max and skew hardly contain historical values within the 25th/75th percentile limits.

The simulated May PDF vs. the historical May PDF is plotted at 100 points.

```
xdensityorig = flow$may %>% sm.density(.,eval.points=xeval,display="none") %>%
  .$estimate

plot.pdf = function(eval,histPDF,simPDF){
  xeval = eval
  plot(xeval,histPDF,pch=".",col="red",ylim=range(simPDF,histPDF),
       xlab="",ylab = "")
  for(i in 1:nsim)lines(xeval,simPDF[i,],col='lightgrey',lty=3)
  lines(xeval,histPDF,lwd=3,col="red")
  title(main="Historical vs. simulated PDF")
}
```

```
plot.pdf(xeval,xdensityorig,simpdf)
```

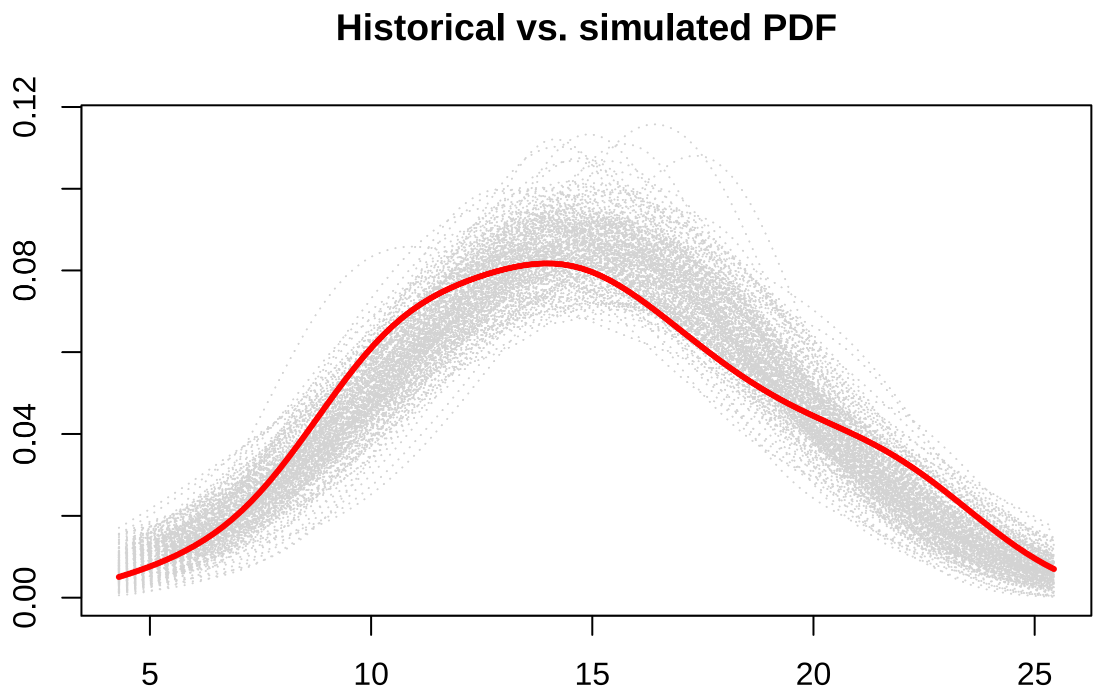


The bimodal historical May PDF is not captured by the simulations due to the Normal nature of the innovations.

The simulated vs. historical anual flow PDF is similarly compared.

```
year.density = flow$year %>% sm.density(.,eval.points=yeval,
               display="none") %>% .$estimate
```

```
plot.pdf(yeval,year.density,year.pdf)
```

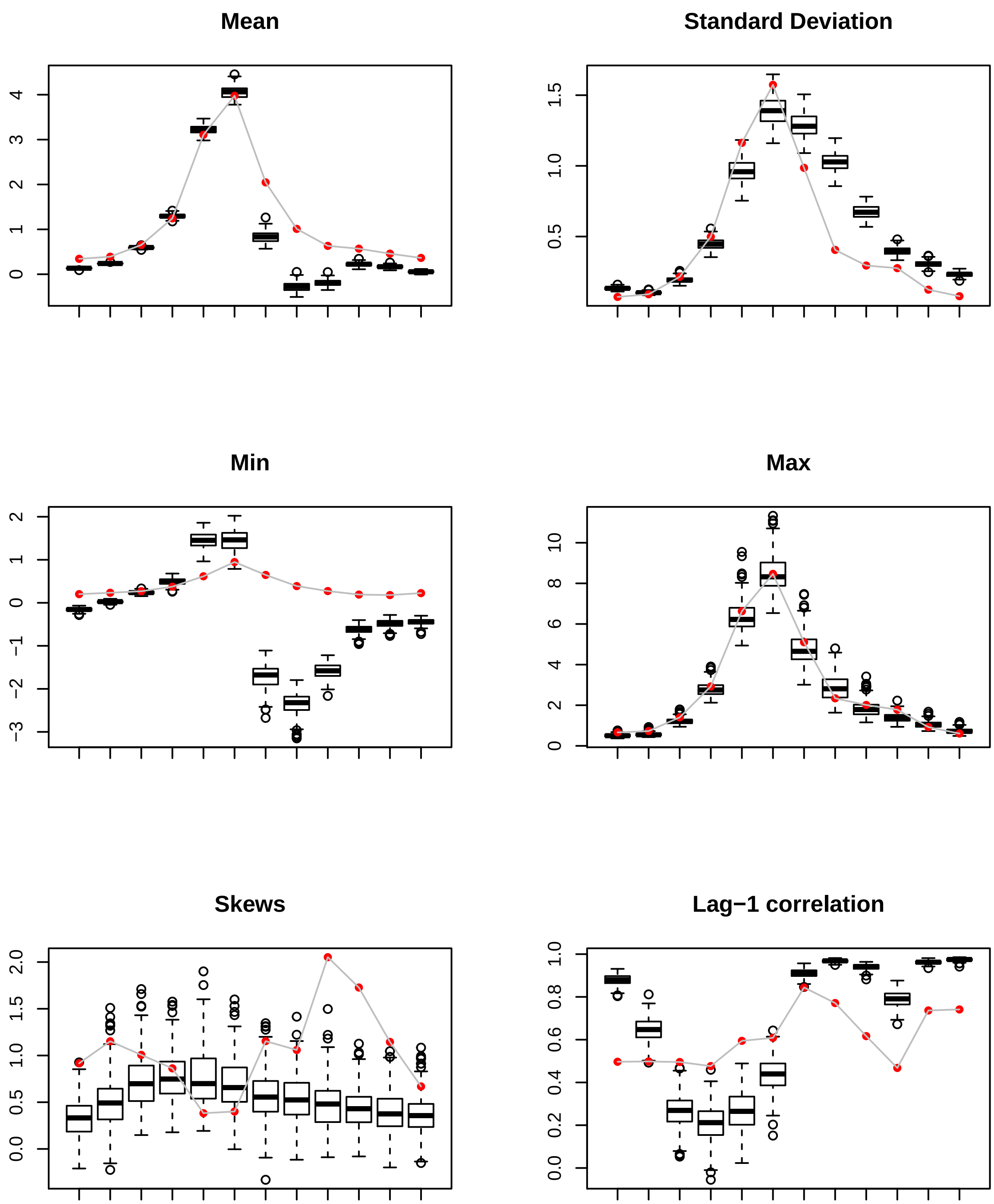


1.3 Replace the simulation of the errors (or innovations) from Normal to Gamma

The simulation code chunks are rerun via r markdown code with innovation = "Gamma".

```
innovation="Gamma"
```

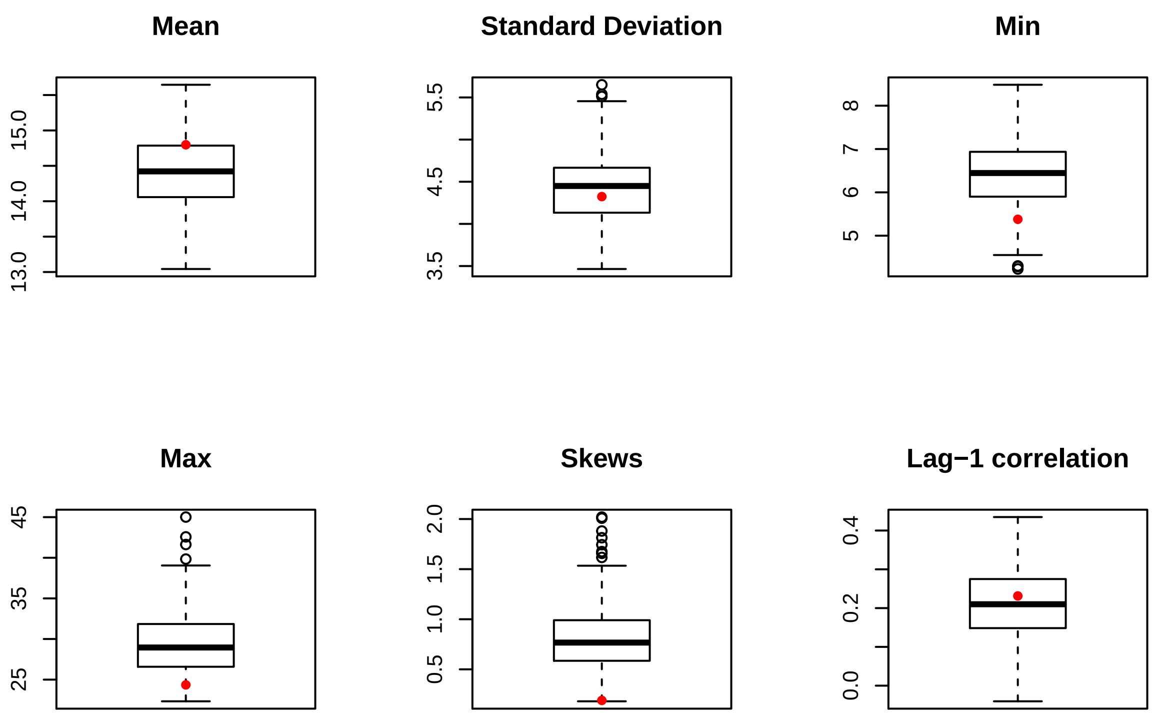
```
par(mfrow=c(3,2))
plot.bp(armean,"Mean")
plot.bp(arstdev,"Standard Deviation")
plot.bp(armin,"Min")
plot.bp(armax,"Max")
plot.bp(arskw,"Skews")
plot.bp(arcor,"Lag-1 correlation")
```



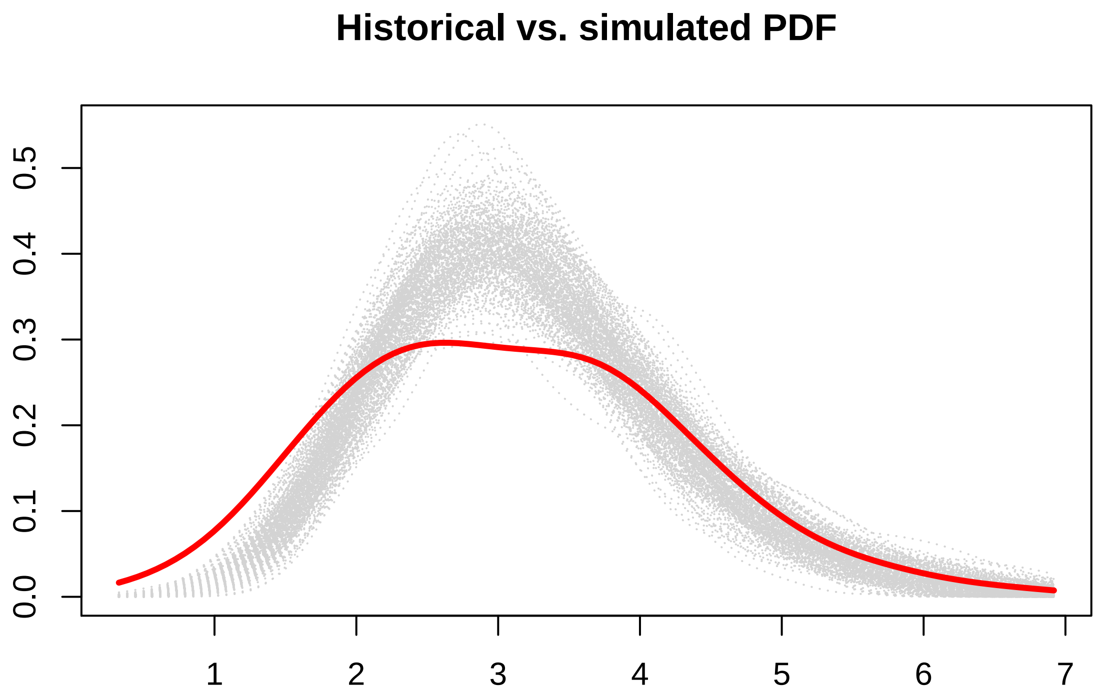
The graphs show a differentiated skew performance, although the fit seems to be equivalent.

```
par(mfrow=c(2,3))
plot.bp(ar.year.stat[,"mean"],"Mean")
plot.bp(ar.year.stat[,"stdev"],"Standard Deviation")
```

```
plot.bp(ar.year.stat[,"min"],"Min")
plot.bp(ar.year.stat[,"max"],"Max")
plot.bp(ar.year.stat[,"skew"],"Skews")
plot.bp(ar.year.stat[,"cor"],"Lag-1 correlation")
```

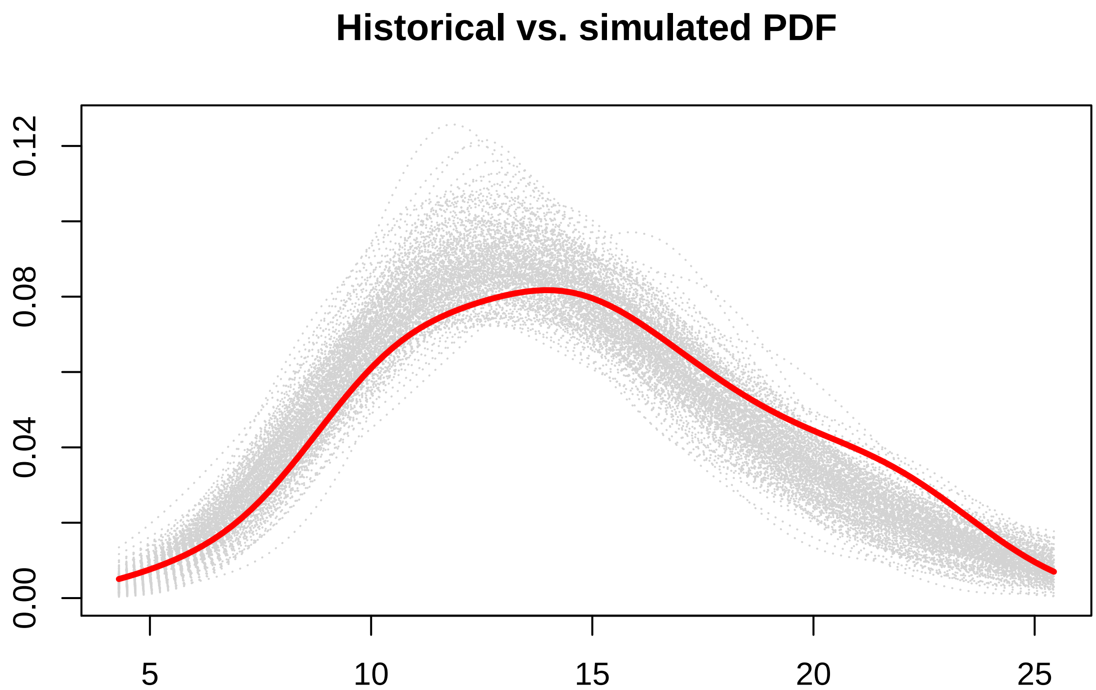



```
plot.pdf(xeval,xdensityorig,simpdf)
```



The simulated May PDF is no longer symmetric, as it would be expected from a Gamma Distribution

```
plot.pdf(yeval,year.density,year.pdf)
```



The same effect is depicted in the anual PDF.