If you go down to the woods today. . . .

Matt Grainger

4 6 2019

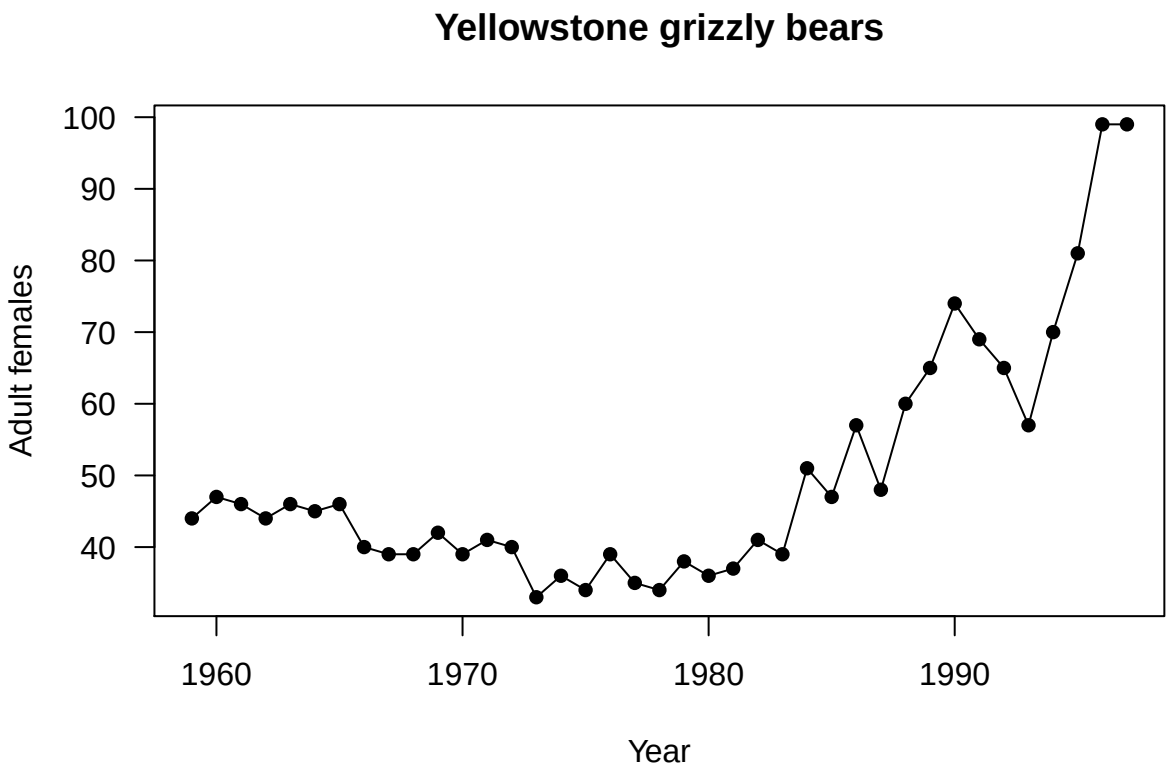
Bear population challenge

Apply your knowledge about population models to this grizzly bear dataset. Answer the questions and show your code (either in a new script file or in a new Rmarkdown file - you can adapt the original). Save the file and send it to me.

```
# a # sign means that R does not read this as code.
#install.packages("popbio") #remove the first hash if you do not have popbio installed as a package
library(popbio)
```

Upload and explore the bears dataset

```
data(grizzly)
View(grizzly)
#?grizzly# what is the grizzly data about?
attach(grizzly) ## You need to attach the data so that
#you can use it in the functions that follow
#(otherwise you need to signal to R which data you are using)
plot(year, N, type='o', pch=16, las=1, xlab="Year",
     ylab="Adult females", main="Yellowstone grizzly bears")
```



What is the grizzly data set all about

Look up the grizzly dataset from the R help (see above ?grizzly will search for the dataset in the helpfiles).

Provide a summary of the grizzly dataset

Calculate the rate of population change

Their are only female bears in this count. What could you do to make this data more representative of the population?

Do we need more data than just female bears?

Use the population models to assess the population change in bears

Make use of only the first 10 years of the bears dataset to predict the future 17 years. Plot this on the same graph (hint in base R you can add points to a plot using the "points()" fuction)

#Extra credit:: Make the plots look pretty (look at ?ggplot2 for example and search on Google for ggplot2)

```
grizzly[1:10,]
```

```
##    year  N
## 1  1959 44
## 2  1960 47
## 3  1961 46
## 4  1962 44
## 5  1963 46
```

```
## 6  1964 45
## 7  1965 46
## 8  1966 40
## 9  1967 39
## 10 1968 39
```

```
lam<-c(rep("NA",9))

for (i in 2:10){
  lam[i]<-grizzly$N[i+1]/grizzly$N[i]
}
lam
```

```
##  [1] "NA"                "0.978723404255319" "0.956521739130435"
##  [4] "1.04545454545455"  "0.978260869565217" "1.02222222222222"
##  [7] "0.869565217391304" "0.975"             "1"
## [10] "1.07692307692308"
```

```
mean(as.numeric(lam), na.rm=TRUE)
```

```
## Warning in mean(as.numeric(lam), na.rm = TRUE): NAs introduced by coercion
```

```
## [1] 0.9891857
```

```
sd(as.numeric(lam), na.rm=TRUE)
```

```
## Warning in is.data.frame(x): NAs introduced by coercion
```

```
## [1] 0.05903223
```

```
# 0.9891
#0.05903223
```

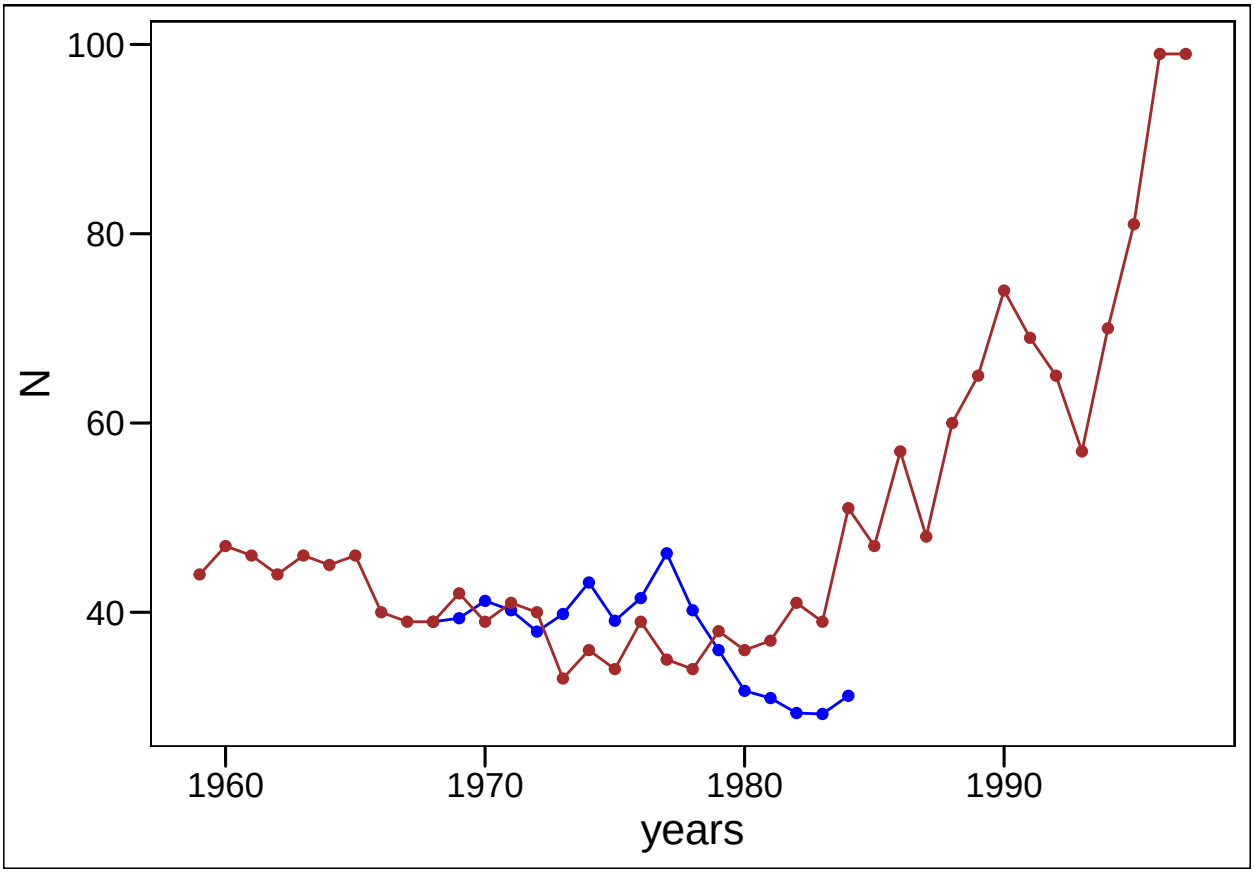
```
dim(grizzly)
```

```
## [1] 39  2
```

```
N=NULL

N[1]=grizzly$N[10]

for (i in 1:16){
  N[i+1]<-rnorm(1,0.9891, 0.05903223)*N[i]
}

N
```

```
##  [1] 39.00000 39.37163 41.20649 40.21433 37.95809 39.81900 43.14806 39.10567
##  [9] 41.50056 46.23020 40.21380 36.00012 31.69849 30.93530 29.35013 29.25494
## [17] 31.17771
```

```
library(tidyverse)
bears<-data.frame(N=N, years=grizzly$year[10:26])
bears %>%
  ggplot(aes(years,N))+
  geom_point(col="blue")+
  geom_line(col="blue")+
  geom_point(data=grizzly, aes(year, N), colour="brown")+
  geom_line(data=grizzly, aes(year, N), colour="brown")+
  ggthemes::theme_base()
```




```
detach(grizzly)# remember to detach the data from the session
rm(list=ls())# this cleans your environment - is useful but
#you might want to keep it sometimes - use caution
```