Exercises for MedCof7 RTC Practical Training 2016 (Group B)

Edmondo Di Giuseppe

November 16, 2016

Contents

Prelude 1

In order to be ready for the practical session **"Hands-on session using R Forecast Verification Package: a) unconditional biases and hits; b) scoring probabilistic forecasts, c) reliability and resolution; d) ROC diagrams"**, leaded by *Dr Caio Augusto dos Santos Coelho* please install and test the package verification.

```
install.packages("verification")
library(verification)
```

Prelude 2

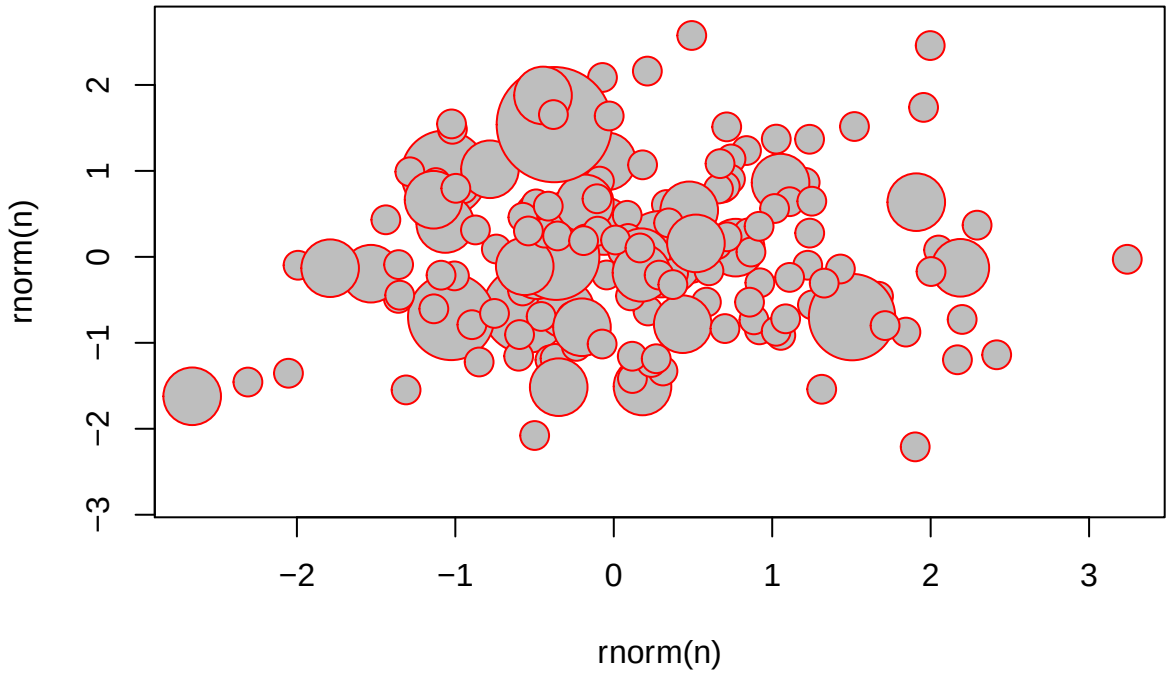
Create a directory anywhere named **R_practical_training**, then open RStudio and create a new project pointing at it. After that, create a new R file script named `R_course_Wed16Nov.R` and a new directory `/Data` in it (for those who don not use RStudio, a similar procedure can be followed except for the project creation phase). Finally, log in the Moodle platform and download the file `SeasFor_GCM_NCEP_datafile.xlsx` from **/Day two/R_lesson** directory of **Verification of Operational Seasonal Forecasts in the Mediterranean region** face-to-face course into `/R_practical_training/Data` directory. From now on, use `R_course_Wed16Nov.R` for copying and executing the chunks of this lesson one after the other.

Practical training

Exercise 1 (warm-up)

Make a bubble plot (i.e. a scatter plot that represents its points as circles) with 500 data extracted from `rnorm()` function, where the bubble size is determined by a poisson process with $\lambda = 0.4$.

```
n = 500
set.seed(123)      # fix the seed for number random generation
#par(mar = c(4, 4, 0.1, 0.1))
plot(rnorm(n), rnorm(n), pch = 21, cex = 2 * rpois(n,lambda = 0.4), col = "red", bg = "gray")
```



Exercise 2 (ggplot2)

1. Install `{gdata}`, `{Hmisc}`, `{datasets}`, `{maps}` and `{ggplot2}` packages
2. Import and print data from file `Data/SeasFor_GCM_NCEP_datafile.xlsx`

```
library(gdata)
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
##
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
##
## Attaching package: 'gdata'
## The following object is masked from 'package:stats':
##
##     nobs
## The following object is masked from 'package:utils':
```

```
##
##     object.size

## The following object is masked from 'package:base':
##
##     startsWith
```

```
Ncep <- read.xls("Data/SeasFor_GCM_Ncep_datafile.xlsx")

str(Ncep)
```

```
## 'data.frame':    497 obs. of  3 variables:
##  $ L                 : num  0.5 1.5 2.5 3.5 4.5 5.5 6.5 0.5 1.5 2.5 ...
##  $ Forecast.Started  : Factor w/ 71 levels "1998-07-01T00:00",..: 1 1 1 1 1 1 1 2 2 2 ...
##  $ precipitation.rate: num  0.82 0.511 1.343 2.074 3.142 ...
```

```
summary(Ncep)
```

```
##        L               Forecast.Started precipitation.rate
##  Min.   :0.5   1998-07-01T00:00:  7     Min.   :0.511
##  1st Qu.:1.5   1998-08-01T00:00:  7     1st Qu.:1.238
##  Median :3.5   1998-09-01T00:00:  7     Median :2.074
##  Mean   :3.5   1998-10-01T00:00:  7     Mean   :2.077
##  3rd Qu.:5.5   1998-11-01T00:00:  7     3rd Qu.:2.826
##  Max.   :6.5   1998-12-01T00:00:  7     Max.   :4.487
##                (Other)         :455     NA's   :50
```

3. Transform data of `$Forecast.Started` column in *date format* and extract the number of days from each forecasted month

```
Ncep$Forecast.Started<-
  as.Date(Ncep$Forecast.Started,format="%Y-%m-%dT%H:%M")
head(Ncep)
```

```
##     L Forecast.Started precipitation.rate
## 1 0.5       1998-07-01          0.8201456
## 2 1.5       1998-07-01          0.5109688
## 3 2.5       1998-07-01          1.3429490
## 4 3.5       1998-07-01          2.0742040
## 5 4.5       1998-07-01          3.1423840
## 6 5.5       1998-07-01          3.3625960
```

```
library(Hmisc)   # for monthDays()
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following object is masked from 'package:gdata':
##
##     combine

## The following objects are masked from 'package:base':
##
```

```
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
Forecast.Days0<-monthDays(Ncep$Forecast.Started)
```

4. Calculate the cumulated precipitation from each season starting from data in `$precipitation.rate` column (precipitation rate unity measure is *mm/day*)

```
Ncep$Forecast.Days<-Forecast.Days0*Ncep$L
Ncep$Forecast.Date<-Ncep$Forecast.Started + Ncep$Forecast.Days

Ncep$Mon.Cum <- round(Ncep$precipitation.rate * (Ncep$Forecast.Days+15),1)
range(Ncep$Mon.Cum,na.rm=T)     # a quick check
```

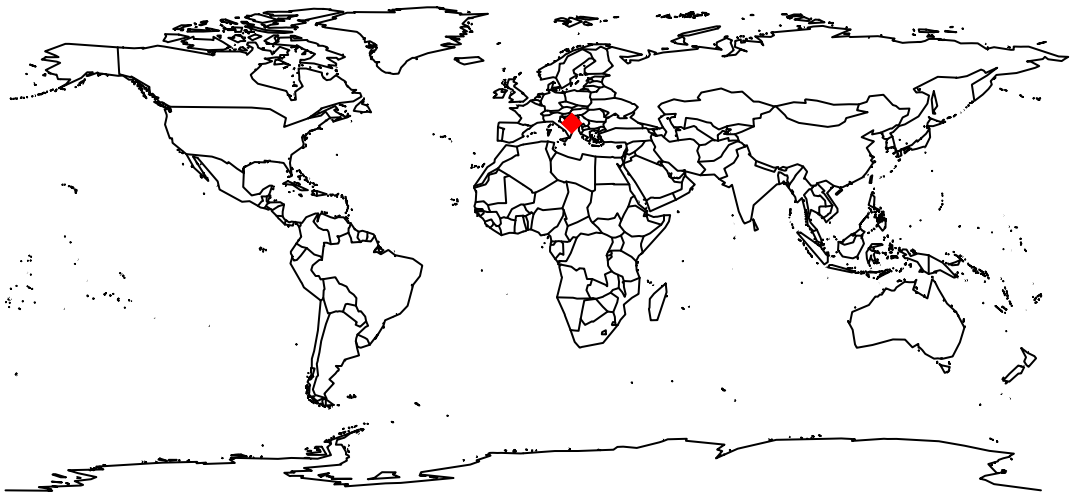
```
## [1]  18.4 916.9
```

5. Locate the cell over the world map

```
require(maps)
```

```
## Loading required package: maps
```

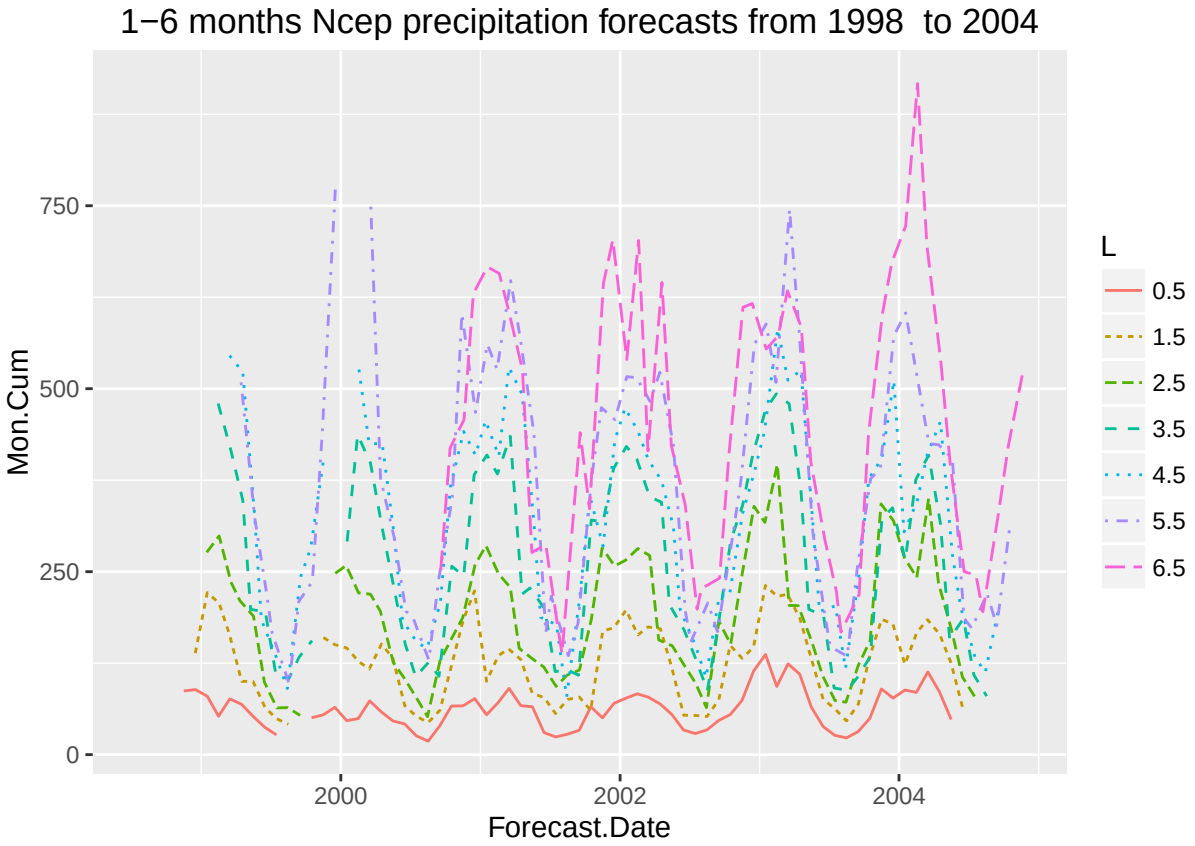
```
map("world")
points(16.875, 43.2542, col="red", pch=18,cex=1.5)
```



6. Group and plot by outlooks (to do step by step with teacher)

```
library(ggplot2)

# get years  from date and select the first and the last:
Ybegin<-as.numeric(getYear(Ncep$Forecast.Date)[1])
Yend<-as.numeric(getYear(Ncep$Forecast.Date)[nrow(Ncep)] )

#transform Ncep data frame:
Ncep2<-Ncep
Ncep2$L<-as.factor(Ncep$L)     # L as factor

#Plot
p<-ggplot(Ncep2, aes(x=Forecast.Date, y=Mon.Cum,  colour=L,linetype=L)) + geom_line()
p+ ggtitle(paste("1-6 months Ncep precipitation forecasts from",Ybegin," to", Yend))
```

```
## Warning: Removed 20 rows containing missing values (geom_path).
```




```
ggsave(paste(getwd(),"/Plots/Ncep_forecasts.pdf",sep=""),
       width = 19, height = 14, units = "cm")
```

```
## Warning: Removed 20 rows containing missing values (geom_path).
```

Exercise 3 (linear regression model)

The goal of this exercise is mainly to become familiar with `list` objects, **loops**, **linear regression model**.

Part A

1. Load package *datasets* (install it, if necessary).
2. Find `nhtemp` and `nottem` time series. The former is the **mean annual temperature** in degrees Fahrenheit in New Haven, Connecticut, from 1912 to 1971. The latter time series object containing **average air temperatures** at Nottingham Castle in degrees Fahrenheit for 20 years.
3. Set a *list object* that contains `nhtemp` and `nottem` and set also names for the two dataset (HINT 1: subsetting of list is done by double square brackets; HINT 2: set an object that contains the name of the two dataset)

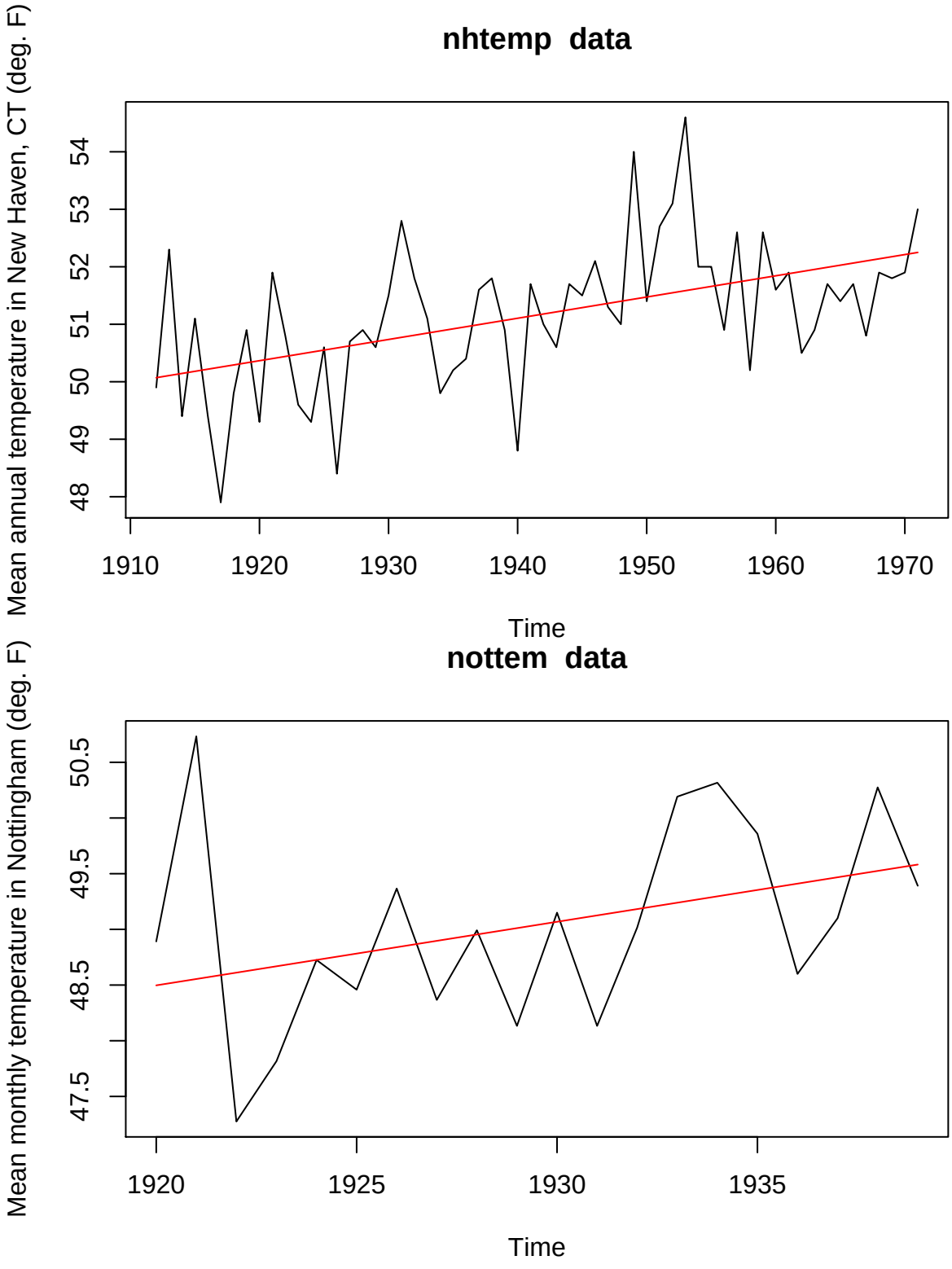
Part B

4. Iterate the following actions:
   - yearly aggregation of `nottem` time series, exclusively;

- compute the *coefficients* of **linear regression model**
- extract the *P value* of β_1 (HINT 3: use `str(summary(model))` to find out the name of P value
- if `Pvalue<0.05`, then compute fitted values of linear model for time series
- save the plot of `original time series + fitted` (HINT 4: use conditional statement inside `ylab` argument of plot)
- write a function to transform from Farhenait to Celsius and plot again.

Solution

```
datax<-list(nhtemp,nottem)
names(datax)<-c("nhtemp","nottem")
for (i in 1:2) {
  dataXX<-datax[[i]]     #HINT1:subsetting of list is done by double square brackets
  #HINT2: set an object that contains the name of the two dataset
  NameDataXX<-names(datax[i])

  # yearly aggregation of nottem time series:
  if(NameDataXX=="nottem"){dataXX<-aggregate.ts(nottem,FUN=mean)}

  # 3) compute the coefficients of linear regression model:
  model<-lm(dataXX~time(dataXX),data=dataXX)

  # 4) extract Pvalue of Beta_1:
  #str(summary(model))     #HINT3:use str(summary(model)) to find out the name of Pvalue
  p.value<-summary(model)$coefficients[2,4]

  # 5) if Pvalue<0.05, then compute fitted values of linear model for time series
  ##model$fitted.values   # or
  Beta_0<-model$coefficients[1]
  Beta_1<-model$coefficients[2]

  fitted<-Beta_0 + Beta_1*time(dataXX)

  # 6) Save plot original time series + fitted:
  ##HINT4: use conditional statement inside ylab argument of plot

  # print on screen
  plot(dataXX, main = paste(NameDataXX," data"), cex.lab=0.7,
       ylab = ifelse(NameDataXX=="nhtemp","Mean annual temperature in New Haven, CT (deg. F)","Mean mon
  )
  lines(fitted,col="red")

  # print on device
  png(filename = paste("Plots/Graphic_Trend_",NameDataXX,sep=""))
  plot(dataXX, main = paste(NameDataXX," data"), cex.lab=0.7,
       ylab = ifelse(NameDataXX=="nhtemp","Mean annual temperature in New Haven, CT (deg. F)","Mean mon
  )
  lines(fitted,col="red")
  dev.off()


  # 7) write a function to transform from Farhenait to Celsius and plot again

}
```


nhtemp data
Mean annual temperature in New Haven, CT (deg. F)
48
49
50
51
52
53
54
1910
1920
1930
1940
1950
1960
1970
Time
nottem data
Mean monthly temperature in Nottingham (deg. F)
47.5
48.5
49.5
50.5
1920
1925
1930
1935
Time

Recommended book

1. install package `{gcookbook}` and load it, then have a look at climate object, which contains **Global climate temperature anomaly data from 1800 to 2011**;
2. follow the example below to plot data (this example is taken from [Cookbook for R] (http://www.cookbook-r.com), where `Anomaly10y` is a 10-year running average of the deviation (in Celsius) from the average 1950–1980 temperature, and `Unc10y` is the 95% confidence interval.

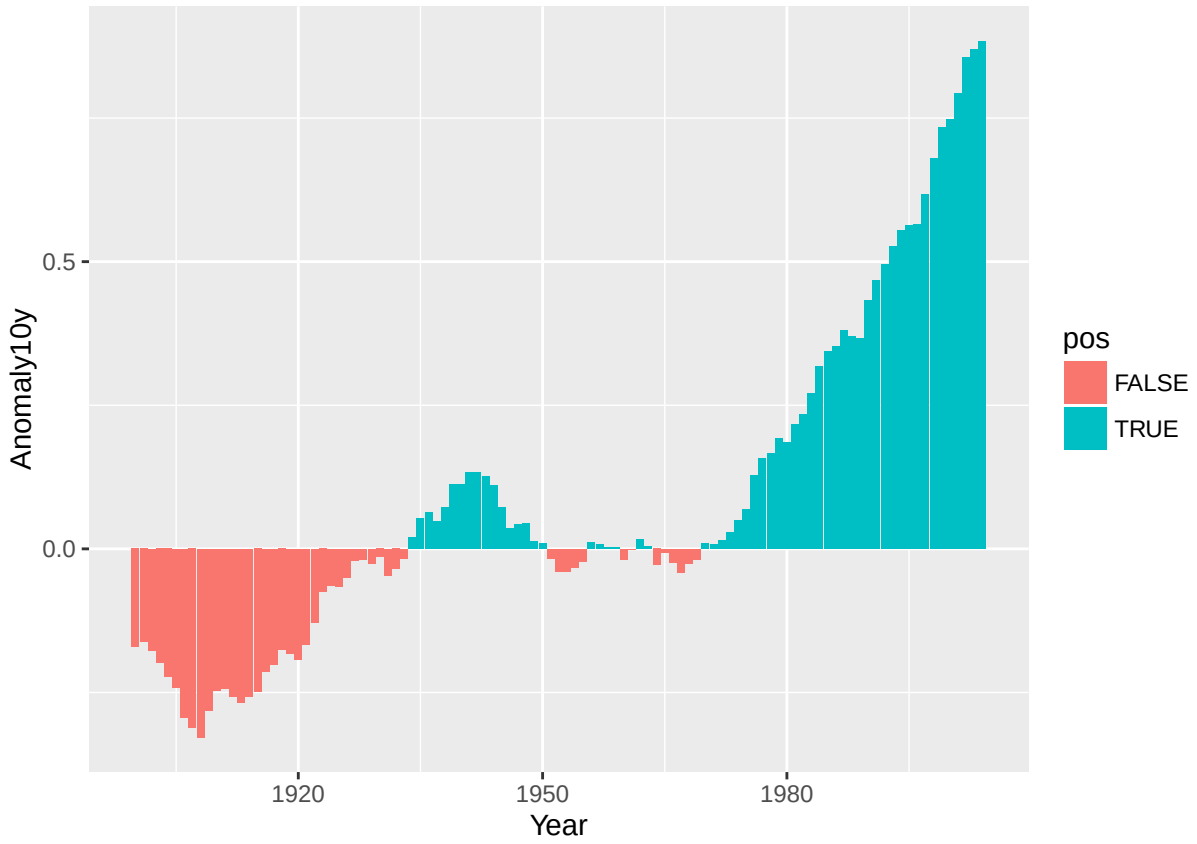
First plot

You want to use different colors for negative and positive-valued bars.

```
library(ggplot2)
library(gcookbook)
library(help="gcookbook")
str(climate)
```

```
## 'data.frame':    499 obs. of  6 variables:
##  $ Source    : chr  "Berkeley" "Berkeley" "Berkeley" "Berkeley" ...
##  $ Year      : num  1800 1801 1802 1803 1804 ...
##  $ Anomaly1y : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Anomaly5y : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Anomaly10y: num  -0.435 -0.453 -0.46 -0.493 -0.536 -0.541 -0.59 -0.695 -0.763 -0.818 ...
##  $ Unc10y    : num  0.505 0.493 0.486 0.489 0.483 0.475 0.468 0.461 0.453 0.451 ...
```

```
# Grab a subset of the climate data:
csub <- subset(climate, Source=="Berkeley" & Year >= 1900)
csub$pos <- csub$Anomaly10y >= 0

ggplot(csub, aes(x=Year, y=Anomaly10y, fill=pos)) +
      geom_bar(stat="identity", position="identity")
```



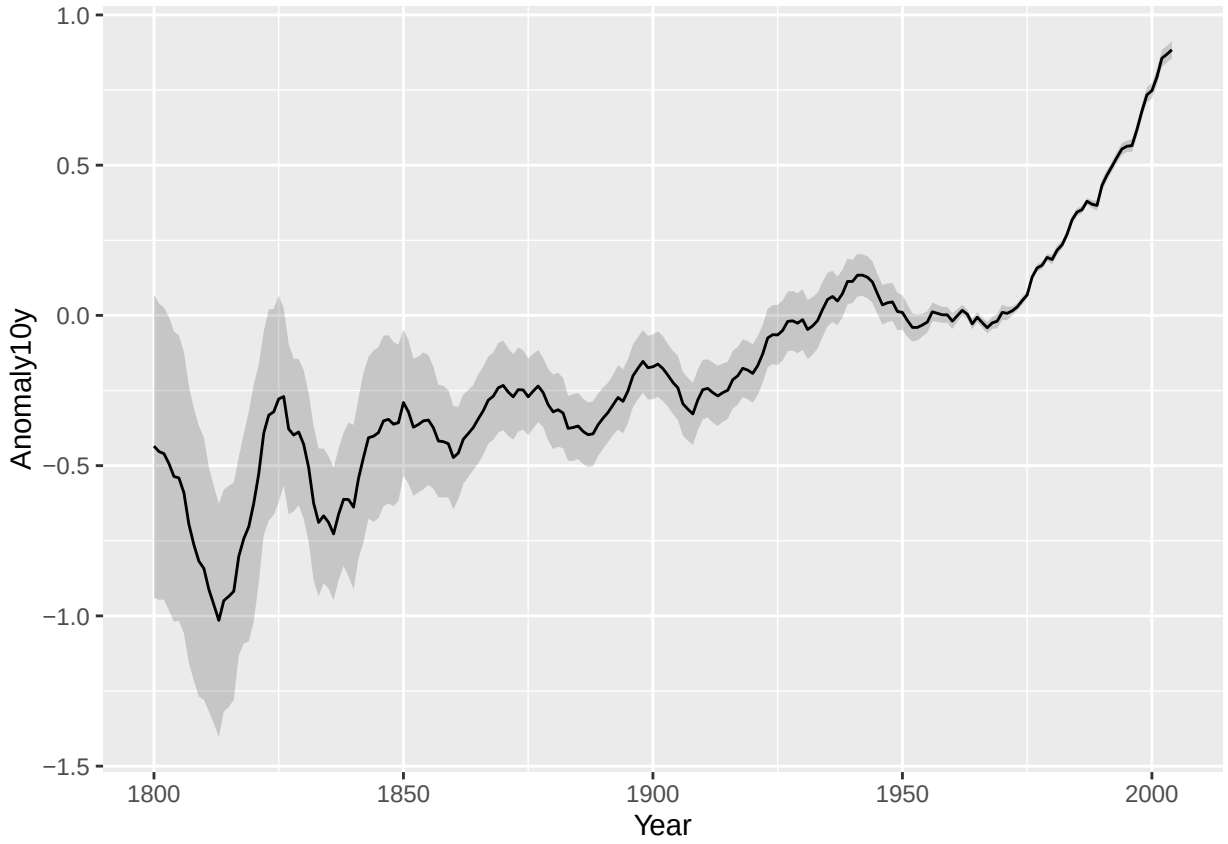
Second plot

You want to add a confidence region to a graph.

```
# Grab a subset of the climate data
clim <- subset(climate, Source == "Berkeley",
               select=c("Year", "Anomaly10y", "Unc10y"))
head(clim)
```

```
##   Year Anomaly10y Unc10y
## 1 1800     -0.435  0.505
## 2 1801     -0.453  0.493
## 3 1802     -0.460  0.486
## 4 1803     -0.493  0.489
## 5 1804     -0.536  0.483
## 6 1805     -0.541  0.475
```

```
# Shaded region
ggplot(clim, aes(x=Year, y=Anomaly10y)) +
    geom_ribbon(aes(ymin=Anomaly10y-Unc10y, ymax=Anomaly10y+Unc10y),alpha=0.2) +
geom_line()
```


1.0
0.5
0.0
−0.5
−1.0
−1.5
Anomaly10y
1800
1850
1900
1950
2000
Year