Communicating Climate Trends

Marc Los Huertos

February 6, 2024 (ver. 0.08)

1 Introduction

Developing the skills to communicate climate trends is an important skill for scientists and policy makers. This exercise will focus on developing the skills to communicate climate trends using R and RMarkdown. We'll produce graphics that can be used in the video presentations.

1.1 Goals

Create a compelling story...

1.2 Approach

1.2.1 R Code with Custom Functions

From the Canvas page, go to the Guide2functions.R file and download the file to your computer. Then upload the file to Rstudio directory you are using for the project.

Open the file in Rstudio and run the code, using the "source". button near the top of the editor window.

Run the Guide4functions.R code and the functions will be loaded into your environment automatically.

As we get further into the project, these code chunks may require tweaking because of the questions you are interested for your location falls outside the design of the custom functions. Please let Marc or the mentors know to get assistance tweaking the code!

2 Mapping (GIS) and Weather Stations

2.1 Simple Mapping of Stations

2.2 More Complex Mapping of Stations

```r
library(geodata)
d <- worldclim_country(country = "USA", var = "tmin",
                       path = tempdir())
terra::plot(mean(d), plg = list(title = "Min. temperature (C)"))
```

```r
library(here)
library(xtable)

stations.active.oldest = read.csv(
  here("04_Regional_Climate_Trends", "stations.active.oldest.csv"))

# OR
# use file.choose() to select the file
# filename = "MY.PATH/04_Regional_Climate_Trends/stations.active.oldest.csv"
# stations.active.oldest = read.csv(filename)
```

2.3 Map US Weather Stations

Here's map that has been transformed using a bunch of libraries that I don't really know how to use, but I found webpage that told me how to make this!

It would be nice to add Canada and Mexico so the USA is not floating in space. I'll work on that later.

```r
library(usmap)
library(ggplot2)
library(sf)

## Linking to GEOS 3.7.2, GDAL 3.0.4, PROJ 6.3.2; sf_use_s2() is TRUE

library(tidyverse)

## -- Attaching core tidyverse packages ------------------------ tidyverse
2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts()
--
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to
force all conflicts to become errors
```

```
library("rnaturalearth")
library("rnaturalearthdata")

##
## Attaching package:  'rnaturalearthdata'
##
## The following object is masked from 'package:rnaturalearth':
##
##     countries110

library(usmap)

world <- ne_countries(scale = "medium", returnclass = "sf")
class(world)

## [1] "sf"         "data.frame"

usa <- subset(world, admin == "United States of America")

(mainland <- ggplot(data = usa) +
     geom_sf(fill = "cornsilk") +
     #geom_sf(data = subset(sites), size = 3, shape = 21, fill = "gray70") +
     geom_sf(data = subset(sites.2163), size = 1, shape = 21, fill = "red") +
    coord_sf(crs = st_crs(2163), xlim = c(-2500000, 2500000), ylim = c(-2300000,
         730000)))

## Error in subset(sites.2163):  object 'sites.2163' not found

(alaska <- ggplot(data = usa) +
     geom_sf(fill = "cornsilk") +
    geom_sf(data = subset(sites.2163), size = 1, shape = 21, fill = "red") +
     coord_sf(crs = st_crs(3467), xlim = c(-2400000, 1600000), ylim = c(200000,
         2500000), expand = FALSE, datum = NA))

## Error in subset(sites.2163):  object 'sites.2163' not found

(hawaii  <- ggplot(data = usa) +
     geom_sf(fill = "cornsilk") +
     geom_sf(data = subset(sites.2163), size = 1, shape = 21, fill = "red") +
     coord_sf(crs = st_crs(4135), xlim = c(-161, -154), ylim = c(18,
         23), expand = FALSE, datum = NA))

## Error in subset(sites.2163):  object 'sites.2163' not found

(world.sf <- ggplot(data =world) +
     geom_sf() +
     #geom_sf(data = subset(sites), size = 3, shape = 21, fill = "gray70") +
     geom_sf(data = subset(sites.2163), size = 1, shape = 21, fill = "red") +
    coord_sf(crs = st_crs(2163), xlim = c(-2500000, 2500000), ylim = c(-2300000,
         730000)))
```

```
## Error in subset(sites.2163):  object 'sites.2163' not found
```

```
mainland +
 annotation_custom(
      grob = ggplotGrob(alaska),
      xmin = -2750000,
      xmax = -2750000 + (1600000 - (-2400000))/2.5,
      ymin = -2450000,
      ymax = -2450000 + (2500000 - 200000)/2.5
  ) +
  annotation_custom(
      grob = ggplotGrob(hawaii),
      xmin = -1250000,
      xmax = -1250000 + (-154 - (-161))*120000,
      ymin = -2450000,
      ymax = -2450000 + (23 - 18)*120000
  )
```

```
## Error in eval(expr, envir, enclos):  object 'mainland' not found
```

3 Communicating Long-term Weather Records

3.1 Complete Records vs. Post 1975 Trends

Communicating climate change based on station records is tricky. The long-term record would on the surface to be the most robust, but several issues arise with a naive analytical approach – my favorite!

```
GSOM_1975.png = paste0(fips$State2, "_", stid, "_GSOM_1975.png")

png(paste0(png_private, GSOM_1975.png), width = 480, height = 320, units = "px", pointsize =
par(las=1, mfrow=c(1,1))
plot(TMAX~Date, GSOM, pch=20, cex=.5, col="grey", ylab="F", main=paste0(fips$State2, "-", st
GSOM.lm = lm(TMAX~Date, GSOM)
pred_dates <-data.frame(Date = GSOM$Date);
nrow(pred_dates);# pred_dates
#Predits the values with confidence interval
ci <- predict(GSOM.lm, newdata = pred_dates,
              interval = 'confidence')
lines(pred_dates$Date, as.numeric(ci[,1]), col="gray50")

# Post 1975
GSOM.lm = lm(TMAX~Date, GSOM[GSOM$Year>1975,])
pred_dates <-data.frame(Date = GSOM$Date[GSOM$Year>1975]);
nrow(pred_dates); #pred_dates
```

```
#Predits the values with confidence interval
ci <- predict(GSOM.lm, newdata = pred_dates,
              interval = 'confidence')
lines(pred_dates$Date, as.numeric(ci[,1]), col="red")
lines(pred_dates$Date, as.numeric(ci[,2]), col="darkorange")
lines(pred_dates$Date, ci[,3], col="darkorange")
location_index = round(length(GSOM[GSOM$Year>1975,]$Date) * 0.99,0)
text(pred_dates$Date[location_index], ci[location_index,3],
     paste(report_prob3(GSOM.lm))[2], pos=2, cex=1.0, col="red")
#abline(coef(lm(TMAX~Date, GSOM)), col="black")
#abline(coef(lm(TMAX~Date, GSOM[GSOM£Year>1975,])), col="red")
dev.off()
```

The noise in the data may suggests that no trend is present

4 Extreme Events–Using Daily Records

4.1 Complicated Nature of Rainfall Patterns

Rainfall trends are tough. Exteme events can occur in 24 hours or over long periods that might result in floods or droughts. Each region might have different patterns, so developing a consistent approach is tough.

We can look for trends in monthly averages, number of days without rain (important in tropics), and/or extreme events based on daily or hourly data.

I don't know of a robust way to look at this for the entire globe.

Rainfall totals by season might be a useful way to think about changes, because the rainfall is often seasonal, I wonder if we can see pattners by season.

```
ggplot( ) +
   geom_bar(data = PRCP.Season.Total,
      aes(x=Year, y=PRCP, fill=Season), stat="identity") +
         xlim(min(CHCND$Year), max(CHCND$Year)-1) +
   #ylab("Number of Extreme Temps") + # for the y axis label
   geom_smooth(data = PRCP.Total,
      aes(y=PRCP, x=Year), method = "lm",
      se = T, color= "black")

# + geom_smooth(data= PRCP.Season.Total, aes(x=Year, y = PRCP, color = Season, group=Season,
```

4.2 Drought

Days without rain...within a calendar year... bleed over between years isn't captured.. This is screwed up, Drought.run needs work.

4.3 Rainfall Probability Distributions by decade... to be developed.

4.4 Record Setting Temperature Records

In many cases, people seem to "feel" how temperature has been changing over time, and new records seem to capture the attention in the media. So, we'll create a updated record of maximum temperatures and display them.

This is a common way to communicate temperatures changes. I suspect we have a better sense of change when we notice "extreme" events...

I tried to use a for loop and in then statements and it was painfully slow, so I converted the data to a matrix that can be used by barplots with much more effeciency!

Create the matrix

@

The patterns of record temperatures often shows increasing number of new high temperature records and fewer record low temperatures more recently, but as usual, it depends on the location (Figure **??**).

4.5 Iterate TMAX vs. Month Boxplots

Evaluating the changes in TMAX and Monthly temperatures might be useful, but for now, I think it's hard to see the patterns.

4.6 Four Plots Compelling Figures

To test the code, I have created graphics that can then be used in the animation process, i.e. try to create code that doesn't get too complicated and then fail!

4.7 KISS

Keeping it simple is critical in communicating scientific information. In this section, I try to come up with a consistent message for every state and a simple graphic.

4.7.1 Change Point Analysis

First, TMIN and TMAX and change point analysis...

https://cran.r-project.org/web/packages/mcp/readme/README.html

4.8 Temp & Precipitation Probability

To highlight the patterns of change, it might be useful to analyze how the probability ditributuion might change – we can use a normal probability distribion as a theoretical distribution (and we can check if this distribuion is appropriate with a Chi-Square test), or we can use the data to create a emperical distribution, which is my favored approach.

I started with decade bins, but used 20 years bins (scores) to simplify the graphics while keeping a pretty good temporal resolution.

This figure is pretty effective, but still needs work.

4.9 Using library densEstBayes

Now, I used a screen split to look at the distribution of the temperate anomolies. First, we look at a simple histogram of the entire dataset.

The data center around zero, as expected, but are these normally distributed?

Next we use a function to estimate the probability distribution using a markof chain the creates an estimated probability distribution. This doesn't always work when the distribution is not even and their only 10 years of data per slot. I suspect, I should make this by every 20 years. Plus that will go way faster and I think the data visualization will be more robust.

```
## Error in fips$State2:  object of type 'closure' is not subsettable
## Error in paste0(png_private, GSOM_estPDF.png):  object 'png_private'
not found
```

The process to create these figures is very time consuming, so in general, I need to come up with an if then statement to avoid creating these everytime!

5 Animated GIFs

So far, this creates a gif file, but I haven't been able get the gif in the pdf directly yet. I will need an additional package or create separate png that are combined. For now, we'll create a gif file to be used in separate documents.

5.1 Probability Distributions

```
## Error in fips$State2:  object of type 'closure' is not subsettable
## Error in paste0(gif_private, GSOM_dnorm.gif):  object 'gif_private'
not found
```

5.2 4 Weather Trend Plots

```
panel4.gif = paste0(fips$State2, "_", stid, "_4panel.gif")

## Error in fips$State2:  object of type 'closure' is not subsettable

if(!file.exists(paste0(gif_private, panel4.gif))){
   print("Creating animated 4panel.gif")
```

```
img <- image_graph(600, 480, res = 96)
# START ----
ylim_new=NA
for(i in seq(min(GSOM$Year), max(GSOM$Year), by=2))
   {
par(las=1, mfrow=c(4,1), mar= c(2, 4, 2, 1) + 0.1)
# TMINmonthMax
   GSOMsub <- GSOM[GSOM$Month==TMINmonthMax & GSOM$Year<=i,]
   if(nrow(GSOMsub)<10) next
plot(TMIN~Date, GSOMsub[GSOMsub$Month==TMINmonthMax,],
   col='gray70', pch=20, xlab="",
   main=paste("Mean", format(GSOMsub$Date,"%B")[1],
              "Min. Temp", GSOM_Longest$name))
GSOM.lm = lm(TMIN~Date, GSOMsub)
pred_dates <-data.frame(Date = GSOMsub$Date);
nrow(pred_dates); pred_dates
#Predits the values with confidence interval
ci <- predict(GSOM.lm, newdata = pred_dates,
              interval = 'confidence')
lines(pred_dates$Date, as.numeric(ci[,1]), col="darkred")
lines(pred_dates$Date, as.numeric(ci[,2]), col="darkorange")
lines(pred_dates$Date, ci[,3], col="darkorange")
location_index = round(length(GSOMsub$Date) * 0.99,0)
text(pred_dates$Date[location_index], ci[location_index,3],
     paste(report_prob2(GSOM.lm)), pos=2, cex=1.5)

# Box Plot of TMAX by Month
CHCNDsub = subset(CHCND, CHCND$Year<=i,
      select=c(Month, Month.name, TMAX, TMIN))
boxplot(TMAX ~ Month.name, data=CHCNDsub, xlab="", main="")
symbol.y = (par()$yaxp[2])-diff(par()$yaxp[1:2])*.99
#symbol.y = (par()£yaxp[2])
text(sumstats$Month, symbol.y, sumstats$TMAX_Symbol,
     col="red", cex=2)
mtext(paste("Maximum Daily Temperatures", min(CHCND$Year),
      "-", i, GSOM_Longest$name), line=1)
mtext("(NOTE: Red astrisks correspond to signficant changes)",
      line=0, cex=.7)

# TMAXmonthMax
GSOMsub <- GSOM[GSOM$Month==TMAXmonthMax & GSOM$Year<=i,]
ylim = range(GSOMsub$TMAX)
#if(!is.na(ylim_new)) ylim[2]=ylim_new
plot(TMAX~Date, GSOMsub, col='gray70', pch=20,
     ylim=ylim, xlab="",
```

```
     main=paste("Mean", format(GSOMsub$Date,"%B")[1],
                "Max. Temp", GSOM_Longest$name))
GSOM.lm = lm(TMAX~Date, GSOMsub)

ci <- predict(GSOM.lm, newdata = pred_dates,
              interval = 'confidence')
lines(pred_dates$Date, as.numeric(ci[,1]), col="darkred")
lines(pred_dates$Date, as.numeric(ci[,2]), col="darkorange")
lines(pred_dates$Date, ci[,3], col="darkorange")

text(pred_dates$Date[location_index], ci[location_index,3],
     paste(report_prob2(GSOM.lm)), pos=2, cex=1.5)

# Record High Temperatures
# START
   j = which(years %in% i)
   if(sum(is.na(TMAX.mat.noleap[,j]))==366) next
TMAX1 = apply(TMAX.mat.noleap[,1:j], 1, function (x) which.max(x));
is.na(TMAX1) <- lengths(TMAX1) == 0
TMAX1 <- unlist(TMAX1)
TMAX1 <- count(TMAX1)
#str(TMAX1)
names(TMAX1)=c("Year", "TMAX")
TMAX_na = data.frame(Year=1:j)
TMAX <- merge(TMAX_na, TMAX1, all.x=TRUE, by="Year")

if(sum(is.na(TMIN.mat[,j]))==366) next
   # Select Minimum and Change to Negative Value
TMIN1 = apply(TMIN.mat[,1:j], 1, function (x) which.min(x));
is.na(TMIN1) <- lengths(TMIN1) == 0
TMIN1 <- unlist(TMIN1)
TMIN1 <- count(TMIN1) # Max Counts Negagive
#str(TMIN1)
names(TMIN1)=c("Year", "TMIN")
TMIN_na <- data.frame(Year=1:j)
TMIN <- merge(TMIN_na, TMIN1, all.x=TRUE, by="Year")

R1 <- merge(TMAX, TMIN, by="Year")
R1$Index = rep(j, nrow(R1))
#results = rbind(results, R)
R1$TMIN = -R1$TMIN
## Sorting out X Axis
tic.no <- 4
rowskip = round(nrow(R1)/tic.no, 0)
row_numb <- seq_len(nrow(R1)) %% rowskip
```

```
row.sel = which(row_numb %in% c(1))
index.year <- years[row.sel]
# switch to decades?

xtics = row.sel
xlabs = index.year

yrange = range(c(R1$TMIN, R1$TMAX), na.rm=T)
ytics = floor(seq(yrange[1], yrange[2], length.out=tic.no))
ylabs = as.character(abs(ytics))

par(las=1, xpd=TRUE)
plot(c(1,nrow(R1)), c(yrange[1], yrange[2]), ty="n", xaxt='n', yaxt='n', ylab="No. of Record
axis(2,at=ytics, labels=ylabs)
axis(1,at=xtics, labels=xlabs)
barplot(height = R1$TMAX, space=0, add = TRUE, axes = FALSE, col="red")
barplot(height = R1$TMIN, space=0, add = TRUE, axes = FALSE, col="blue")
# END
}

# STOP ----
dev.off()

GSOM_animation <- image_animate(img, fps = 1, loop=2, optimize = TRUE)
image_write(GSOM_animation, paste0(gif_private, panel4.gif))

} else {
   print("Skipping animated GSOM_4plots chunk")}

## Error in paste0(gif_private, panel4.gif):  object 'gif_private' not
found
```

The file is saved in the main directory.

5.3 Evaluating Records

TBD

5.4 Export Options

TBD

6 Sea Surface Temperature Data – SURP PROJECT WAITING TO HAPPEN

In contrast to terrestrial data, sea surface temperature (SST) is quite difficult to obtain and process. There are numerous tools to access the data, but they often require knowledge of complex software tools that are not easy to set up or programming experience with python or others.

https://climexp.knmi.nl/select.cgi?id=someone@somewhere&field=ersstv5

There are, however, a few tools build for R users that seem to accomplish all that we need.

https://rda.ucar.edu/index.html?hash=data_user&action=register

https://rda.ucar.edu/datasets/ds277.9/

Alternatively, we can download flat ascII tables of gridded data:

https://www1.ncdc.noaa.gov/pub/data/cmb/ersst/v5/ascii/

7 Satellite Data

TBD

8 Ice-Core Data

TBD

9 Conclusions

Developing a robust method to analyze weather stations is both time consuming and difficult to justify the outcome. In part because the data suggest that each station (region) requires different types of analysis, based on the expected patterns of temperature and rainfall. As climate scientists have known for decades, the terminology of global warming is not very useful. Not because scientists are trying to hide something or promote some biased agenda, but that even as warming of the global average is well documented, the impacts of climate change on each region is highly specific, requiring specificity in the analysis.

Hopefully, this little analysis has created some mechanism for others to appreciate this compexity.

```
## Error in paste0(GSOM_Longest$name, " (ID: ", GSOM_Longest$id, ")"):
object 'GSOM_Longest' not found
## Error in fips$State:  object of type 'closure' is not subsettable
## Error in is.data.frame(x):  object 'dbase' not found
## Error in eval(expr, envir, enclos):  object 'start_time' not found
```