

NetCDF and timeseries - mapping, cropping, correlating

M Campbell

01/10/2020

Contents

1	NetCDF Files	1
2	Packages	1
3	Data	2
4	Mapping GISTEMP - the August 2020 temperature anomaly	2
5	Cropping down to Australia only	6
6	Correlating!	9

1 NetCDF Files

I have avoided NetCDF files like the plague - I always thought they were scary, and up until now I've been time-oriented, with little consideration for space (a poor trait in a geographer). I know everyone uses them, but I found it fairly hard to do some basic stuff (and find tutorials for basic stuff), so here's a collection of simple steps for using spatio-temporal data with `ncdf4` and `raster` (and some other packages).

Below includes scripts and results to: 1. Map a layer from the GISTEMP temperature anomaly 2. Crop the GISTEMP raster brick 3. Correlate the time series for each cell of the raster brick by a synthetic time series (e.g. representative of a single station's data, or a palaeoclimate proxy).

Nothing is neat or tidy or nice, this is literally the output of my little learning journey shoved into a .rmd file, but hopefully it might make someone's life easier.

2 Packages

```
library(rnaturalearth)
library(rnaturalearthdata)
library(ncdf4) # package for netcdf manipulation
library(raster) # package for raster manipulation
library(rgdal) # package for geospatial analysis
library(ggplot2)
library(rasterVis)
library(pals)
```

3 Data

I downloaded the 'Surface air temperature (no ocean data), 250km smoothing' compressed NetCDF file from the gistemp page.

4 Mapping GISTEMP - the August 2020 temperature anomaly

```
# data -----

nc_data <- nc_open("D:/CampbellM1/R_learning_space/gisstemp_2020/gistemp250_GHCNv4.nc")

# data cleaning -----

{ # for looking at the data
  sink("gisstemp.txt")
  print(nc_data)
  sink()
}

lon <- ncvar_get(nc_data, "lon")
lat <- ncvar_get(nc_data, "lat")
t <- ncvar_get(nc_data, "time")
tdate <- as.Date(t, origin=c('1800-01-01')) ## convert time (days since 18000101 to date)

temp.array <- ncvar_get(nc_data, "tempanomaly") #extract temp anomaly data
fillvalue <- ncatt_get(nc_data, "tempanomaly", "_FillValue")
nc_close(nc_data)

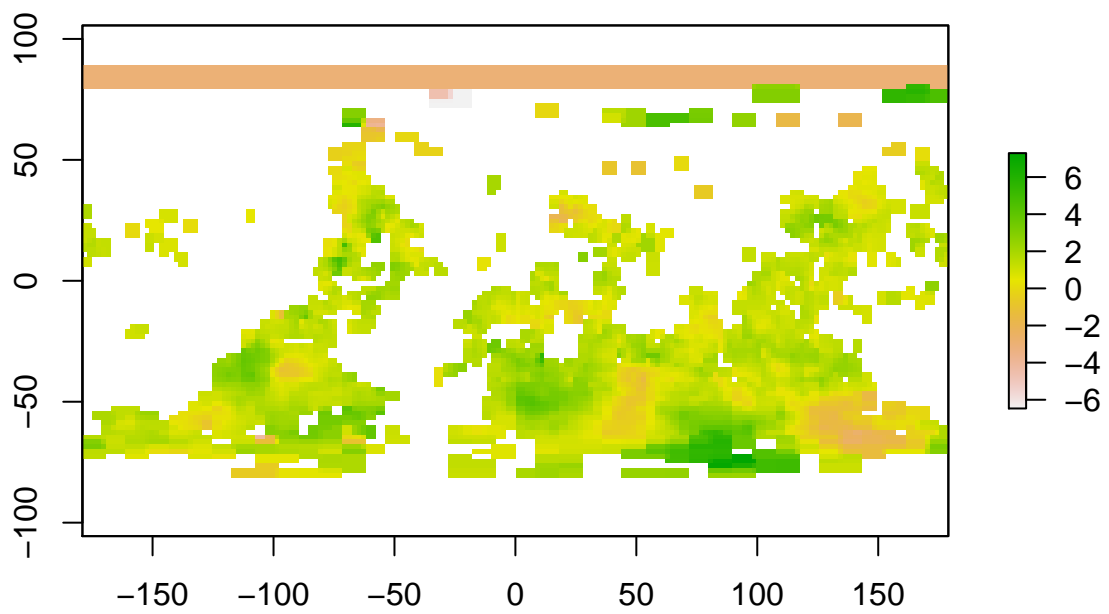
temp.array[temp.array == fillvalue$value] <- NA ## replace fill value with NA
temp.slice <- temp.array[, , 1688] ## pulls one month(?) of data out.
dim(temp.slice)

## [1] 180 90

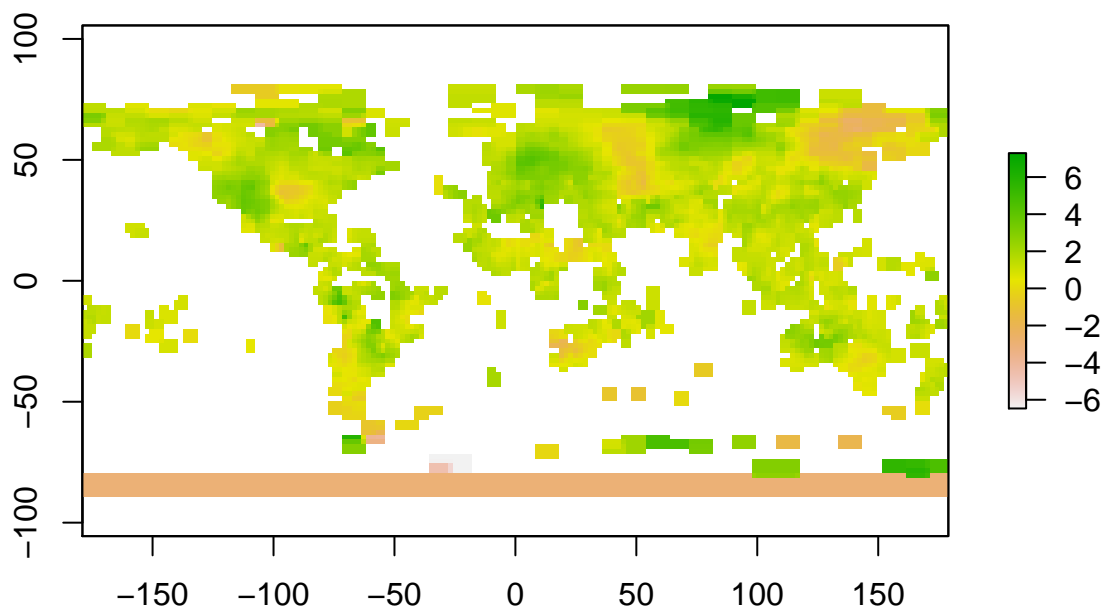
dim(temp.array)

## [1] 180 90 1688

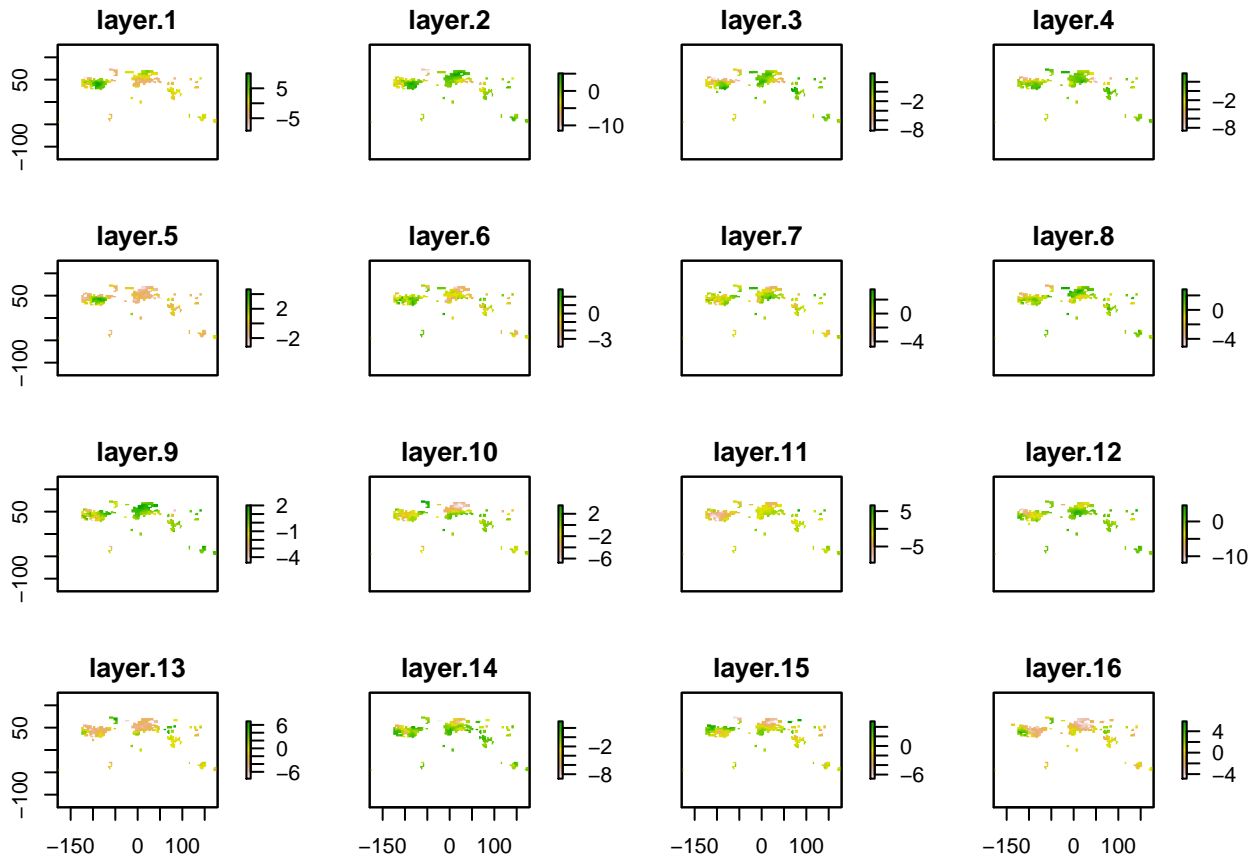
r <- raster(t(temp.slice), xmn = min(lon), xmx = max(lon), ymn = min(lat), ymx = max(lat), crs = CRS("+proj=stere",
plot(r)
```



```
r <- flip(r, direction = 'y')  
plot(r)
```



```
### bricking collapses all layers into one object.
r_brick <- brick(temp.array, xmn=min(lat), xmx=max(lat), ymn=min(lon), ymx=max(lon), crs=CRS("+proj=long",
r_brick <- t(r_brick)
r_brick <- flip(r_brick, direction = "y")
plot(r_brick)
```



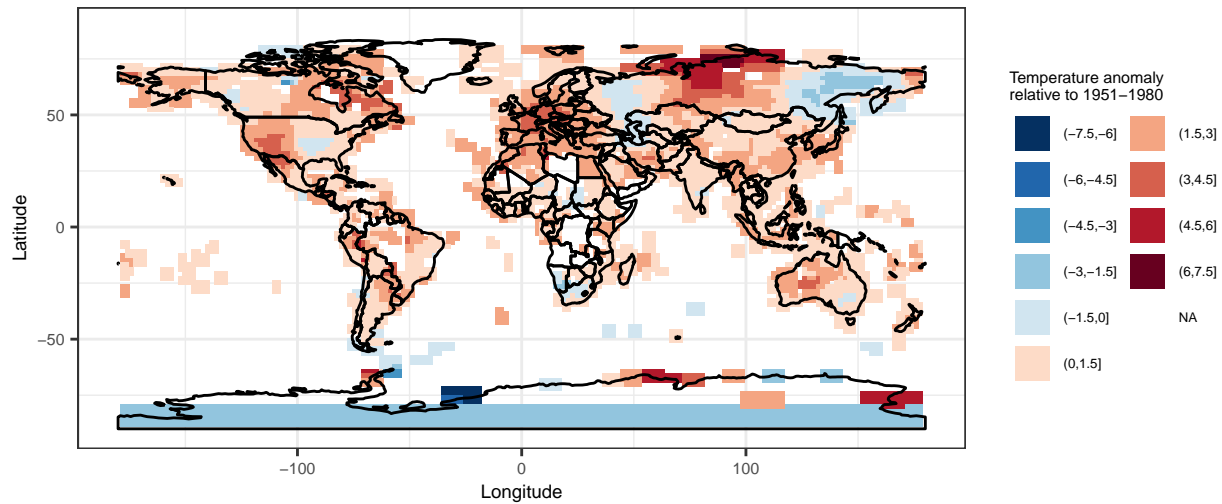
```
## Plot august 2020 anomaly
```

```
tempCuts <- seq(-7.5, 7.5, length.out = 11)
```

```
world <- ne_countries()
rasterVis::gplot(r) +
  geom_tile(aes(fill = cut(value, tempCuts))) +
  geom_path(data = world, aes(x = long, y = lat, group = group)) +
  scale_fill_brewer(palette = "RdBu",
                    drop = FALSE,
                    direction = -1) +

  coord_equal() +
  theme_bw() +
  labs(x = "Longitude",
       y = "Latitude",
       fill = "Temperature anomaly \nrelative to 1951-1980",
       title = "August 2020 GISTEMP V4 \nSurface Air Temperature Anomaly",
       subtitle = "Surface Air Temperature (no ocean data): 250 km smoothing") +
  guides(fill = guide_legend(override.aes = list(size = 2), ncol = 2)) +
  theme(plot.title = element_text(size = 8),
        plot.subtitle = element_text(size = 6),
        legend.title = element_text(size = 6),
        legend.text = element_text(size = 5),
        axis.title = element_text(size = 7),
        axis.text = element_text(size = 6))
```

August 2020 GISTEMP V4
 Surface Air Temperature Anomaly
 Surface Air Temperature (no ocean data): 250 km smoothing



5 Cropping down to Australia only

```
# colorpalette -----

## want to go from -4 to 4 at 0.5 degrees = 17 colours
palcoolwarm <- coolwarm(n = 17)

# data -----

nc_data <- nc_open("D:/CampbellM1/R_learning_space/gisstemp_2020/gistemp250_GHCnv4.nc")

lon <- ncvar_get(nc_data, "lon")
lat <- ncvar_get(nc_data, "lat")
t <- ncvar_get(nc_data, "time")
tdate <- as.Date(t, origin=c('1800-01-01')) ## convert time (days since 18000101 to date)

temp.array <- ncvar_get(nc_data, "tempanomaly") #extract temp anomaly data
fillvalue <- ncatt_get(nc_data, "tempanomaly", "_FillValue")
nc_close(nc_data)

temp.array[temp.array == fillvalue$value] <- NA ## replace fill value with NA
```

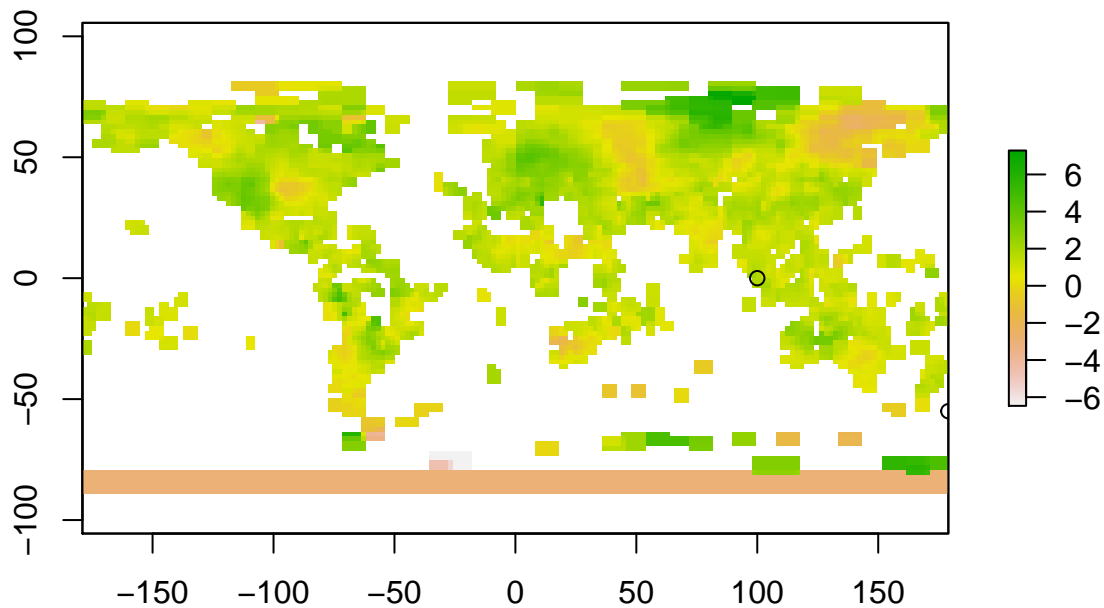
```

r_brick <- brick(temp.array, xmn=min(lat), xmx=max(lat), ymn=min(lon), ymx=max(lon), crs=CRS("+proj=longlat"))
r_brickcorrect <- t(r_brick)
r_brickcorrect <- flip(r_brickcorrect, direction = "y")

print(r_brick)

## class      : RasterBrick
## dimensions : 180, 90, 16200, 1688 (nrow, ncol, ncell, nlayers)
## resolution : 1.977778, 1.988889 (x, y)
## extent     : -89, 89, -179, 179 (xmin, xmax, ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs
## source     : memory
## names      : layer.1, layer.2, layer.3, layer.4, layer.5, layer.6, layer.7, layer.8, layer.9, layer.10, layer.11, layer.12, layer.13, layer.14, layer.15, layer.16, layer.17, layer.18, layer.19, layer.20, layer.21, layer.22, layer.23, layer.24, layer.25, layer.26, layer.27, layer.28, layer.29, layer.30, layer.31, layer.32, layer.33, layer.34, layer.35, layer.36, layer.37, layer.38, layer.39, layer.40, layer.41, layer.42, layer.43, layer.44, layer.45, layer.46, layer.47, layer.48, layer.49, layer.50, layer.51, layer.52, layer.53, layer.54, layer.55, layer.56, layer.57, layer.58, layer.59, layer.60, layer.61, layer.62, layer.63, layer.64, layer.65, layer.66, layer.67, layer.68, layer.69, layer.70, layer.71, layer.72, layer.73, layer.74, layer.75, layer.76, layer.77, layer.78, layer.79, layer.80, layer.81, layer.82, layer.83, layer.84, layer.85, layer.86, layer.87, layer.88, layer.89, layer.90
## min values :  -9.15,  -11.53,  -8.17,  -8.62,  -3.17,  -3.92,  -4.74,  -5.08,  -4.64,  -6.15,  -5.15,  -4.15,  -3.15,  -2.15,  -1.15,  -0.15,  0.85,  1.85,  2.85,  3.85,  4.85,  5.85,  6.85,  7.85,  8.85,  9.85, 10.85, 11.85, 12.85, 13.85, 14.85, 15.85, 16.85, 17.85, 18.85, 19.85, 20.85, 21.85, 22.85, 23.85, 24.85, 25.85, 26.85, 27.85, 28.85, 29.85, 30.85, 31.85, 32.85, 33.85, 34.85, 35.85, 36.85, 37.85, 38.85, 39.85, 40.85, 41.85, 42.85, 43.85, 44.85, 45.85, 46.85, 47.85, 48.85, 49.85, 50.85, 51.85, 52.85, 53.85, 54.85, 55.85, 56.85, 57.85, 58.85, 59.85, 60.85, 61.85, 62.85, 63.85, 64.85, 65.85, 66.85, 67.85, 68.85, 69.85, 70.85, 71.85, 72.85, 73.85, 74.85, 75.85, 76.85, 77.85, 78.85, 79.85, 80.85, 81.85, 82.85, 83.85, 84.85, 85.85, 86.85, 87.85, 88.85, 89.85, 90.85
## max values :  10.00,   5.10,   3.84,   3.83,   4.65,   2.87,   3.47,   2.84,   2.01,   3.15,   4.15,   5.15,   6.15,   7.15,   8.15,   9.15,  10.15,  11.15,  12.15,  13.15,  14.15,  15.15,  16.15,  17.15,  18.15,  19.15,  20.15,  21.15,  22.15,  23.15,  24.15,  25.15,  26.15,  27.15,  28.15,  29.15,  30.15,  31.15,  32.15,  33.15,  34.15,  35.15,  36.15,  37.15,  38.15,  39.15,  40.15,  41.15,  42.15,  43.15,  44.15,  45.15,  46.15,  47.15,  48.15,  49.15,  50.15,  51.15,  52.15,  53.15,  54.15,  55.15,  56.15,  57.15,  58.15,  59.15,  60.15,  61.15,  62.15,  63.15,  64.15,  65.15,  66.15,  67.15,  68.15,  69.15,  70.15,  71.15,  72.15,  73.15,  74.15,  75.15,  76.15,  77.15,  78.15,  79.15,  80.15,  81.15,  82.15,  83.15,  84.15,  85.15,  86.15,  87.15,  88.15,  89.15,  90.15
plot(r_brickcorrect$layer.1688)
pr <- r_brickcorrect
pt <- cbind(c(100, 179), c(0, -55))
plot(pr[[1688]])
points(pt)

```

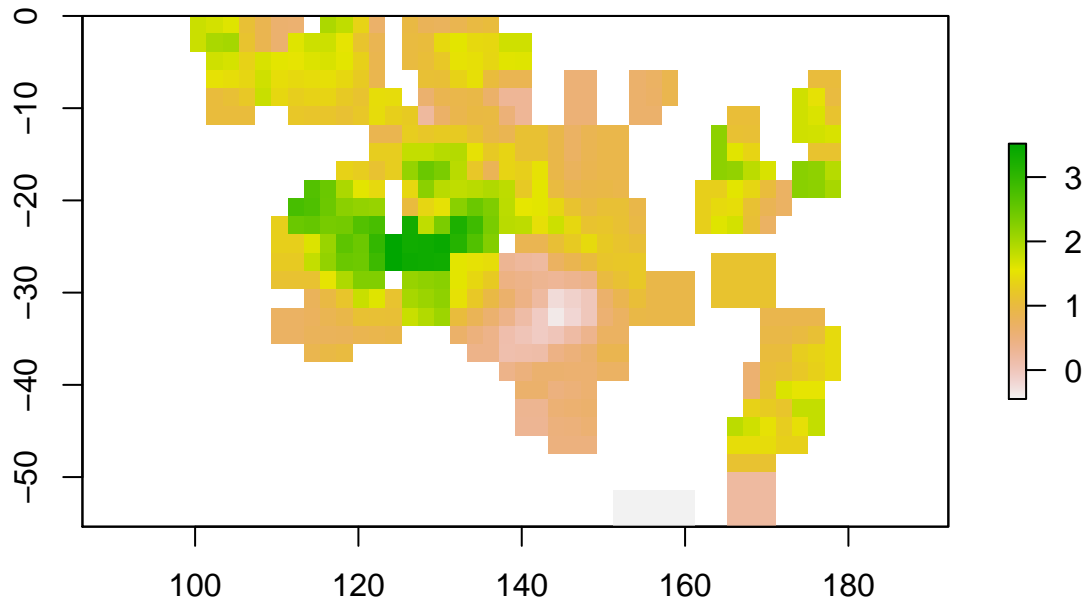


```

croptentent <- extent(c(100, 179, -55, 0))
cropped_gistemp <- crop(r_brickcorrect, croptentent)

```

```
plot(cropped_gistemp[[1688]])
```



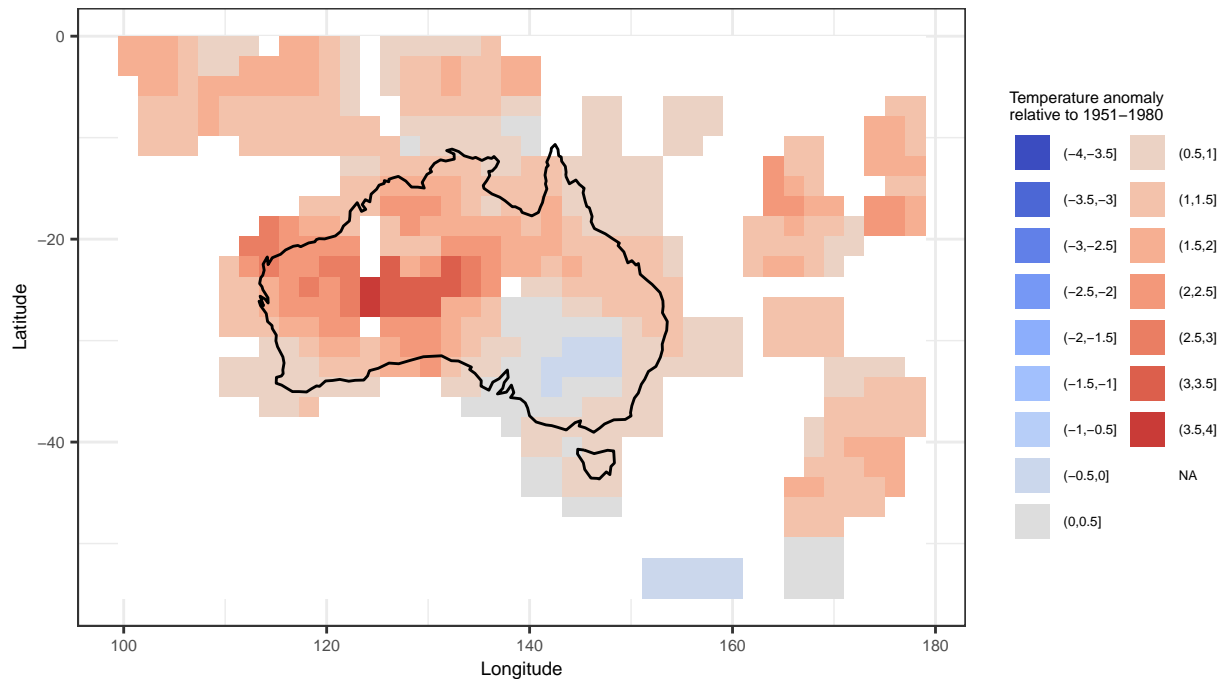
```
tempCuts <- seq(-4, 4, length.out = 17)

aus <- ne_countries(country = "australia")
rasterVis::gplot(cropped_gistemp[[1688]]) +
  geom_tile(aes(fill = cut(value, tempCuts))) +
  geom_path(data = aus, aes(x = long, y = lat, group = group)) +
  scale_fill_manual(values = as.vector(palcoolwarm),
                    drop = FALSE,
                    na.value = "white") +

  coord_equal() +
  theme_bw() +
  labs(x = "Longitude",
       y = "Latitude",
       fill = "Temperature anomaly \nrelative to 1951-1980",
       title = "August 2020 GISTEMP V4 \nSurface Air Temperature Anomaly",
       subtitle = "Surface Air Temperature (no ocean data): 250 km smoothing") +
  guides(fill = guide_legend(override.aes = list(size = 2), ncol = 2)) +
  theme(plot.title = element_text(size = 8),
        plot.subtitle = element_text(size = 6),
        legend.title = element_text(size = 6),
        legend.text = element_text(size = 5),
        axis.title = element_text(size = 7),
        axis.text = element_text(size = 6))
```


August 2020 GISTEMP V4 Surface Air Temperature Anomaly

Surface Air Temperature (no ocean data): 250 km smoothing



6 Correlating!

Each brick has $x * y$ cells by z monthly layers. To correlate each cell against a one-dimensional timeseries (representative of, for e.g. a single station's data), we can use a tool from **raster**. Note that I just made up some data with trend, so the correlation is rubbish!

This uses the same `cropped_gistemp` data from the previous section.

```
fakedata <- data.frame(t = seq(1:1688), fakey = rnorm(1688, mean = 20, sd = 1) * t^(0.2)) # make some sy

fun <- function(x) { # function to apply in raster::calc
  cor(x, fakedata$fakey, method = "pearson") ## pearson's correlation of fake data and raster vals
}

x2 <- raster::calc(cropped_gistemp, fun) # correlation!

cropped_gistemp_e <- extract(cropped_gistemp, matrix(c(145, -30), ncol = 2)) ## check correlation by ext
cropped_gistemp_e <- t(data.frame(y = cropped_gistemp_e)) # make useable

test <- cor(cropped_gistemp_e, fakedata$fakey) # correlation for extracted point timeseries and fake da
test

##           [,1]
## [1,] 0.03598129

x2cor <- extract(x2, matrix(c(145, -30), ncol = 2)) # extract correlation
x2cor
```

```
##
## 0.03598129
tempCuts <- seq(-1, 1, length.out = 7)
aus <- ne_countries(country = "australia")
rasterVis::gplot(x2) +
  geom_tile(aes(fill = cut(value, tempCuts))) +
  geom_path(data = aus, aes(x = long, y = lat, group = group)) +
  geom_point(aes(x = 145, y = -30)) +
  scale_fill_brewer(palette = "RdBu",
                    drop = FALSE,
                    direction = -1) +
  coord_equal() +
  theme_bw() +
  labs(x = "Longitude",
       y = "Latitude",
       fill = "Correlation GISTEMP v4 against synthetic timseries",
       title = "Correlation check - Gistemp v4",
       subtitle = "Surface Air Temperature (no ocean data): 250 km smoothing") +
  guides(fill = guide_legend(override.aes = list(size = 2), ncol = 2)) +
  theme(plot.title = element_text(size = 8),
        plot.subtitle = element_text(size = 6),
        legend.title = element_text(size = 6),
        legend.text = element_text(size = 5),
        axis.title = element_text(size = 7),
        axis.text = element_text(size = 6))
```

