Session 2 Code

Ralph Trancoso, Sarah Chapman, Rohan Eccles

2025-06-11

Table of contents

Install packages

```
#install.packages("terra")
#install.packages("dplyr")
#install.packages("sf")
#install.packages("ggplot2")
#install.packages("dismo")
#install.packages("rasterVis")
#install.packages("reshape")

## Import packages
library(terra)
library(dplyr)
library(sf)
library(ggplot2)
library(dismo)
library(rasterVis)
library(reshape)
library(RColorBrewer)
```

Part 1: Temperature Data (one model)

Set working directory

```
#setwd("C:/R Code/Training/ICCB/")
getwd()                    # get work directory
```

```
[1] "/Users/scottforrest/Library/CloudStorage/OneDrive-QueenslandUniversityofTechnology/PhD
```

```
dir()                      # list folders in the work directory
```

```
[1] "data"
[2] "ICCB2025_Session2_ClimateProjections.pdf"
[3] "README.md"
[4] "scripts"
[5] "session_2_code_files"
[6] "session_2_code.html"
[7] "session_2_code.qmd"
[8] "session_2_code.rmarkdown"
[9] "session_2_home.qmd"
```

```
dir("data/annual/")        # List files in subdirectory
```

```
 [1] "pr_ACCESS-ESM1-5_ssp126_r6i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
 [2] "pr_ACCESS-ESM1-5_ssp245_r6i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
 [3] "pr_ACCESS-ESM1-5_ssp370_r6i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
 [4] "pr_EC-Earth3_ssp126_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
 [5] "pr_EC-Earth3_ssp245_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
 [6] "pr_EC-Earth3_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
 [7] "pr_GFDL-ESM4_ssp126_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
 [8] "pr_GFDL-ESM4_ssp245_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
 [9] "pr_GFDL-ESM4_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
[10] "tas_ACCESS-ESM1-5_ssp126_r6i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
[11] "tas_ACCESS-ESM1-5_ssp245_r6i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
[12] "tas_ACCESS-ESM1-5_ssp370_r6i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
[13] "tas_EC-Earth3_ssp126_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
[14] "tas_EC-Earth3_ssp245_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
[15] "tas_EC-Earth3_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
[16] "tas_GFDL-ESM4_ssp126_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
[17] "tas_GFDL-ESM4_ssp245_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
[18] "tas_GFDL-ESM4_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
```

Retrieve file names in the directory

```
files=dir("data/annual/")
files[1]
```

```
[1] "pr_ACCESS-ESM1-5_ssp126_r6i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc"
```

Load file and query data (working with one model)

```
tas = rast("data/annual/tas_GFDL-ESM4_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc")
tas
```

```
class       : SpatRaster
dimensions  : 205, 176, 120  (nrow, ncol, nlyr)
resolution  : 0.1, 0.1  (x, y)
extent      : 137.45, 155.05, -29.45, -8.95  (xmin, xmax, ymin, ymax)
coord. ref. : lon/lat WGS 84 (CRS84) (OGC:CRS84)
source      : tas_GFDL-ESM4_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc
varname     : tas_annual (Seasonal average of Near-Surface Air Temperature (Annual))
names       : tas_annual_1, tas_annual_2, tas_annual_3, tas_annual_4, tas_annual_5, tas_annu
unit        :         degC,         degC,         degC,         degC,         degC,
time (days) : 1981-01-01 to 2100-01-01 (2 steps)
```

Check your lat and lon coords

```
xFromCol(tas)
```

```
  [1] 137.5 137.6 137.7 137.8 137.9 138.0 138.1 138.2 138.3 138.4 138.5 138.6
 [13] 138.7 138.8 138.9 139.0 139.1 139.2 139.3 139.4 139.5 139.6 139.7 139.8
 [25] 139.9 140.0 140.1 140.2 140.3 140.4 140.5 140.6 140.7 140.8 140.9 141.0
 [37] 141.1 141.2 141.3 141.4 141.5 141.6 141.7 141.8 141.9 142.0 142.1 142.2
 [49] 142.3 142.4 142.5 142.6 142.7 142.8 142.9 143.0 143.1 143.2 143.3 143.4
 [61] 143.5 143.6 143.7 143.8 143.9 144.0 144.1 144.2 144.3 144.4 144.5 144.6
 [73] 144.7 144.8 144.9 145.0 145.1 145.2 145.3 145.4 145.5 145.6 145.7 145.8
 [85] 145.9 146.0 146.1 146.2 146.3 146.4 146.5 146.6 146.7 146.8 146.9 147.0
 [97] 147.1 147.2 147.3 147.4 147.5 147.6 147.7 147.8 147.9 148.0 148.1 148.2
[109] 148.3 148.4 148.5 148.6 148.7 148.8 148.9 149.0 149.1 149.2 149.3 149.4
[121] 149.5 149.6 149.7 149.8 149.9 150.0 150.1 150.2 150.3 150.4 150.5 150.6
[133] 150.7 150.8 150.9 151.0 151.1 151.2 151.3 151.4 151.5 151.6 151.7 151.8
[145] 151.9 152.0 152.1 152.2 152.3 152.4 152.5 152.6 152.7 152.8 152.9 153.0
[157] 153.1 153.2 153.3 153.4 153.5 153.6 153.7 153.8 153.9 154.0 154.1 154.2
[169] 154.3 154.4 154.5 154.6 154.7 154.8 154.9 155.0
```

```
yFromRow(tas)
```

```
  [1]  -9.0  -9.1  -9.2  -9.3  -9.4  -9.5  -9.6  -9.7  -9.8  -9.9 -10.0 -10.1
 [13] -10.2 -10.3 -10.4 -10.5 -10.6 -10.7 -10.8 -10.9 -11.0 -11.1 -11.2 -11.3
 [25] -11.4 -11.5 -11.6 -11.7 -11.8 -11.9 -12.0 -12.1 -12.2 -12.3 -12.4 -12.5
 [37] -12.6 -12.7 -12.8 -12.9 -13.0 -13.1 -13.2 -13.3 -13.4 -13.5 -13.6 -13.7
 [49] -13.8 -13.9 -14.0 -14.1 -14.2 -14.3 -14.4 -14.5 -14.6 -14.7 -14.8 -14.9
 [61] -15.0 -15.1 -15.2 -15.3 -15.4 -15.5 -15.6 -15.7 -15.8 -15.9 -16.0 -16.1
 [73] -16.2 -16.3 -16.4 -16.5 -16.6 -16.7 -16.8 -16.9 -17.0 -17.1 -17.2 -17.3
 [85] -17.4 -17.5 -17.6 -17.7 -17.8 -17.9 -18.0 -18.1 -18.2 -18.3 -18.4 -18.5
 [97] -18.6 -18.7 -18.8 -18.9 -19.0 -19.1 -19.2 -19.3 -19.4 -19.5 -19.6 -19.7
[109] -19.8 -19.9 -20.0 -20.1 -20.2 -20.3 -20.4 -20.5 -20.6 -20.7 -20.8 -20.9
[121] -21.0 -21.1 -21.2 -21.3 -21.4 -21.5 -21.6 -21.7 -21.8 -21.9 -22.0 -22.1
[133] -22.2 -22.3 -22.4 -22.5 -22.6 -22.7 -22.8 -22.9 -23.0 -23.1 -23.2 -23.3
[145] -23.4 -23.5 -23.6 -23.7 -23.8 -23.9 -24.0 -24.1 -24.2 -24.3 -24.4 -24.5
[157] -24.6 -24.7 -24.8 -24.9 -25.0 -25.1 -25.2 -25.3 -25.4 -25.5 -25.6 -25.7
[169] -25.8 -25.9 -26.0 -26.1 -26.2 -26.3 -26.4 -26.5 -26.6 -26.7 -26.8 -26.9
[181] -27.0 -27.1 -27.2 -27.3 -27.4 -27.5 -27.6 -27.7 -27.8 -27.9 -28.0 -28.1
[193] -28.2 -28.3 -28.4 -28.5 -28.6 -28.7 -28.8 -28.9 -29.0 -29.1 -29.2 -29.3
[205] -29.4
```

Adding missing year values to the data

```
dates = seq(as.Date("1981-01-01"), as.Date("2100-12-01"), by="year")
names(tas) = dates  # fixing the time data in the NetCDF
tas
```

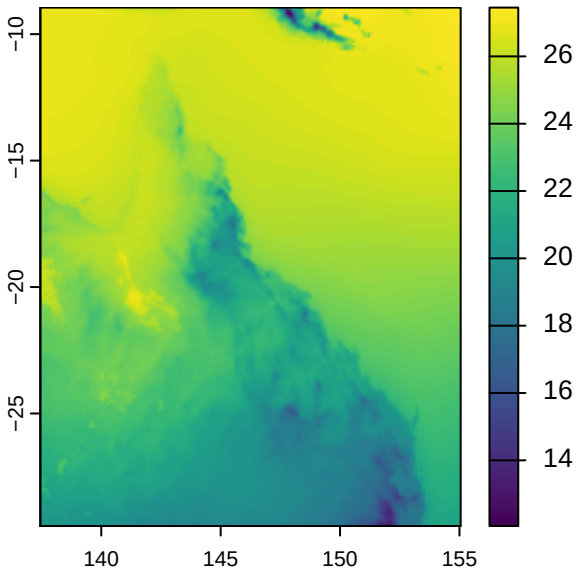
```
class       : SpatRaster
dimensions  : 205, 176, 120  (nrow, ncol, nlyr)
resolution  : 0.1, 0.1  (x, y)
extent      : 137.45, 155.05, -29.45, -8.95  (xmin, xmax, ymin, ymax)
coord. ref. : lon/lat WGS 84 (CRS84) (OGC:CRS84)
source      : tas_GFDL-ESM4_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc
varname     : tas_annual (Seasonal average of Near-Surface Air Temperature (Annual))
names       : 1981-01-01, 1982-01-01, 1983-01-01, 1984-01-01, 1985-01-01, 1986-01-01, ...
unit        :       degC,       degC,       degC,       degC,       degC,       degC, ...
time (days) : 1981-01-01 to 2100-01-01 (2 steps)
```

Sub-setting and plotting the data

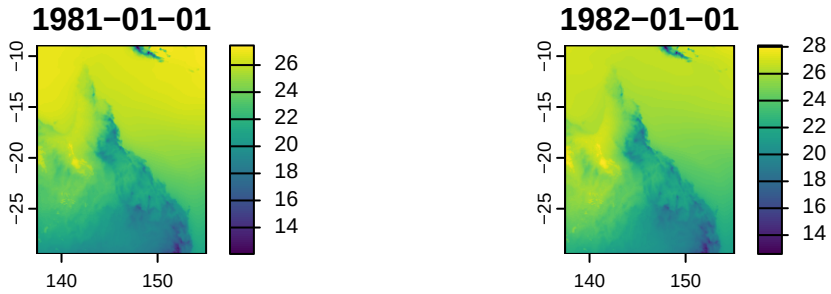
```
tas[[1]]
```

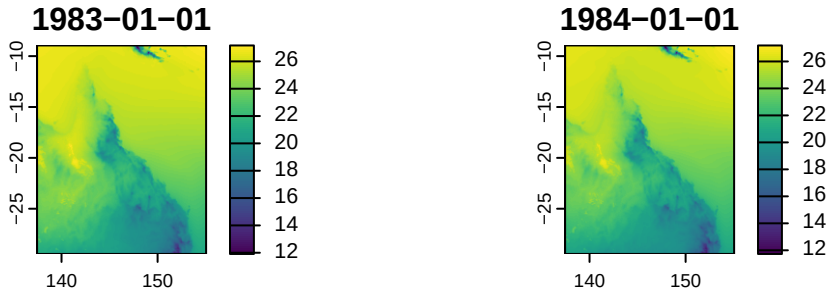
```
class       : SpatRaster
dimensions  : 205, 176, 1  (nrow, ncol, nlyr)
resolution  : 0.1, 0.1  (x, y)
extent      : 137.45, 155.05, -29.45, -8.95  (xmin, xmax, ymin, ymax)
coord. ref. : lon/lat WGS 84 (CRS84) (OGC:CRS84)
source      : tas_GFDL-ESM4_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_sem_1981-2100.nc
varname     : tas_annual (Seasonal average of Near-Surface Air Temperature (Annual))
name        : 1981-01-01
unit        :       degC
time (days) : 1981-01-01
```

```
plot(tas[[1]])
```

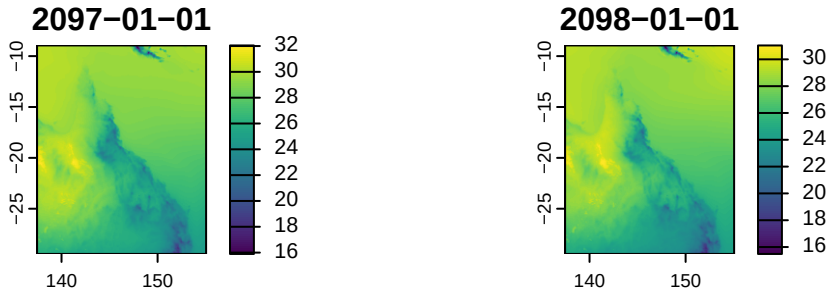


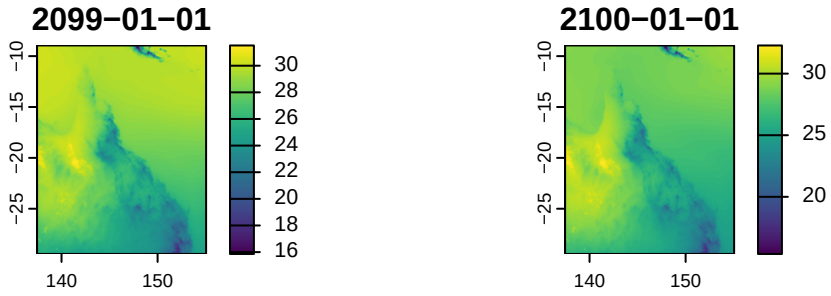

```
plot(tas[[1:4]])
```





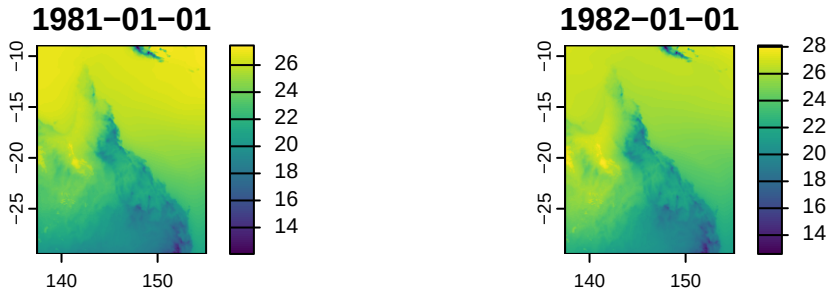

```
plot(tas[[117:120]])
```

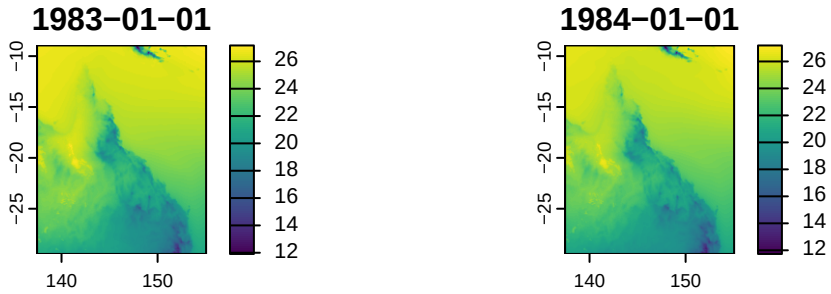




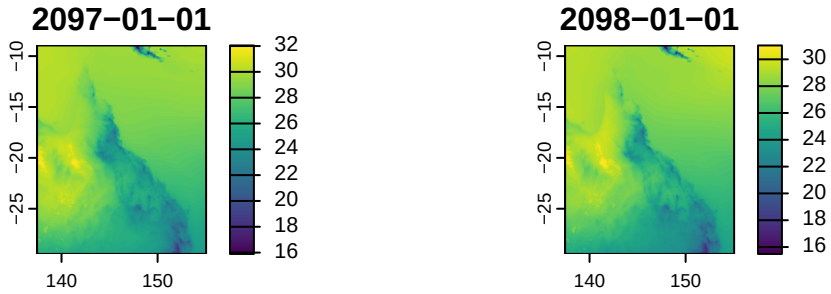
Can also sub-set the data according to the date array

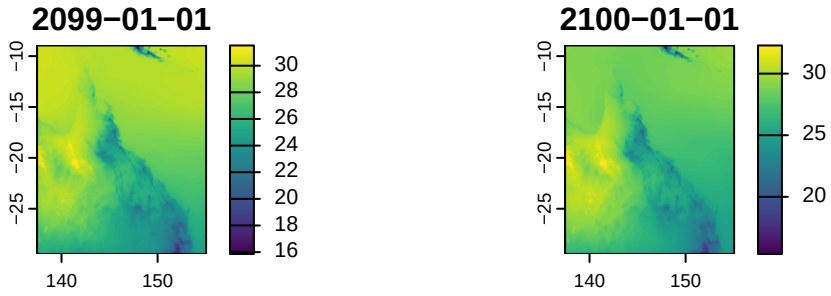
```
plot(tas[[dates >= as.Date("1981-01-01") & dates <= as.Date("1984-01-01")]])
```






```
plot(tas[[dates >= as.Date("2097-01-01") & dates <= as.Date("2100-01-01")]])
```

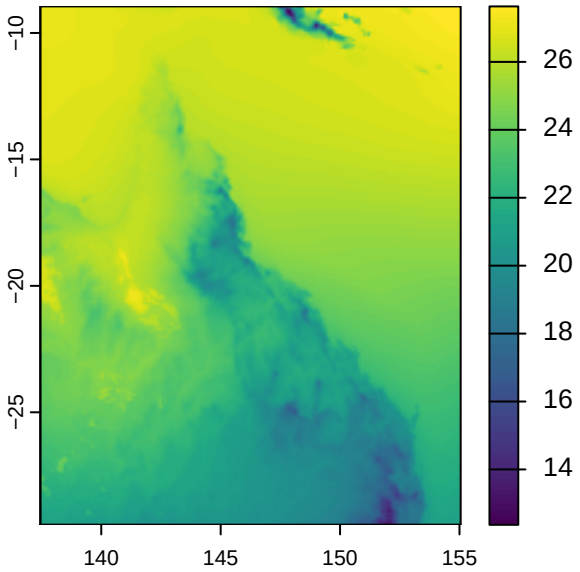




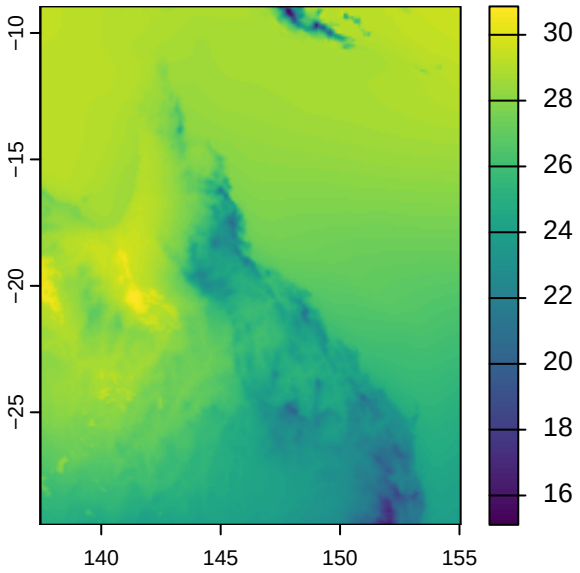

```
tas_base = mean(tas[[1:30]])      # Calculating climatology for baseline (1981-2010)
tas_fut = mean(tas[[91:120]])     # Calculating climatology for future (2071-2100)
```

Q: Can you cut the historical data and future based on the dates?

```
plot(tas_base)
```

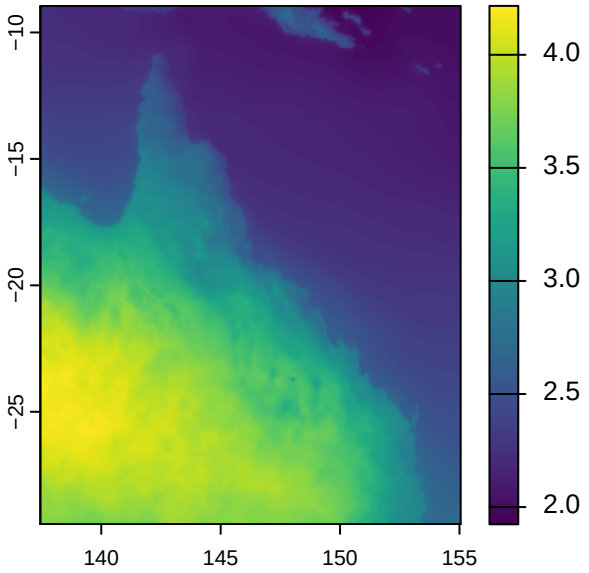



```
plot(tas_fut)
```



Change in future temperature (future - base)

```
tas_dif = tas_fut - tas_base
plot(tas_dif)
```



Q. Can you make this plot nicer? Add a title and change the colours

Hint: You can set plot titles using 'main'

Control the colours using col = brewer.pal(11, 'PaletteName') (see https://colorbrewer2.org/ for colour options)

You can plot multiple figures in one plot using par (mfrow = c(nrows, ncols))

Also check the instructions from Session 1!

Extracting out point data (timeseries)

```
tas[50,50]                  # extracts data from the 50th lat and 50th lon position
```

```
  1981-01-01 1982-01-01 1983-01-01 1984-01-01 1985-01-01 1986-01-01 1987-01-01
1   25.24902   25.26156   24.88015   24.41082   24.94491   24.89218   25.25115
  1988-01-01 1989-01-01 1990-01-01 1991-01-01 1992-01-01 1993-01-01 1994-01-01
1   25.23535   25.03982   25.42452   25.37658    25.3677   25.10143    25.3218
  1995-01-01 1996-01-01 1997-01-01 1998-01-01 1999-01-01 2000-01-01 2001-01-01
1   24.87841   25.11816   25.46691   25.25924   25.05966   25.53942   25.29299
  2002-01-01 2003-01-01 2004-01-01 2005-01-01 2006-01-01 2007-01-01 2008-01-01
1   25.82775   25.32467   25.54989   25.65566   25.43701   25.56466   25.35082
  2009-01-01 2010-01-01 2011-01-01 2012-01-01 2013-01-01 2014-01-01 2015-01-01
1   25.64773   25.44879   25.77899   25.52069   26.40573   26.00399   26.02764
  2016-01-01 2017-01-01 2018-01-01 2019-01-01 2020-01-01 2021-01-01 2022-01-01
1   25.50436   25.88986   26.02621   25.68951    25.9519   25.43524   26.02264
  2023-01-01 2024-01-01 2025-01-01 2026-01-01 2027-01-01 2028-01-01 2029-01-01
1   26.07147   25.98385    26.4558   25.86419   26.31649   25.50891   26.05761
  2030-01-01 2031-01-01 2032-01-01 2033-01-01 2034-01-01 2035-01-01 2036-01-01
1   26.26635   26.16805   25.56466    25.6921    26.2561   26.34539    25.9425
```

```
  2037-01-01 2038-01-01 2039-01-01 2040-01-01 2041-01-01 2042-01-01 2043-01-01
1   26.57571   26.29916   26.85446   26.48287   26.22824   26.75714   26.34863
  2044-01-01 2045-01-01 2046-01-01 2047-01-01 2048-01-01 2049-01-01 2050-01-01
1   26.01452   26.37426   26.30599   26.89645   26.49154   26.71673   26.64852
  2051-01-01 2052-01-01 2053-01-01 2054-01-01 2055-01-01 2056-01-01 2057-01-01
1   26.51864   26.47985   26.61602    26.9179   26.97494   26.73034   27.19863
  2058-01-01 2059-01-01 2060-01-01 2061-01-01 2062-01-01 2063-01-01 2064-01-01
1   27.05032   26.64703   27.30004   26.55456   27.39593   27.40051   27.23416
  2065-01-01 2066-01-01 2067-01-01 2068-01-01 2069-01-01 2070-01-01 2071-01-01
1    27.5408   27.24737   27.33636   27.09204   27.59817   26.69656   27.19824
  2072-01-01 2073-01-01 2074-01-01 2075-01-01 2076-01-01 2077-01-01 2078-01-01
1   26.80004   27.00533    28.2991   27.67822    27.7326   27.16381   28.68814
  2079-01-01 2080-01-01 2081-01-01 2082-01-01 2083-01-01 2084-01-01 2085-01-01
1   27.56314    27.8674   27.92907   27.94192    27.6759   27.62152   27.59582
  2086-01-01 2087-01-01 2088-01-01 2089-01-01 2090-01-01 2091-01-01 2092-01-01
1   28.27426     27.941   28.31573   27.85235   27.81686   27.98327   28.39962
  2093-01-01 2094-01-01 2095-01-01 2096-01-01 2097-01-01 2098-01-01 2099-01-01
1   28.12872   28.79971   28.32763   27.89684   28.33975   27.95684   28.80871
  2100-01-01
1    28.1459
```

```
cells <- cellFromRowCol(tas[[1]], 50, 50)
xyFromCell(tas,cells)
```

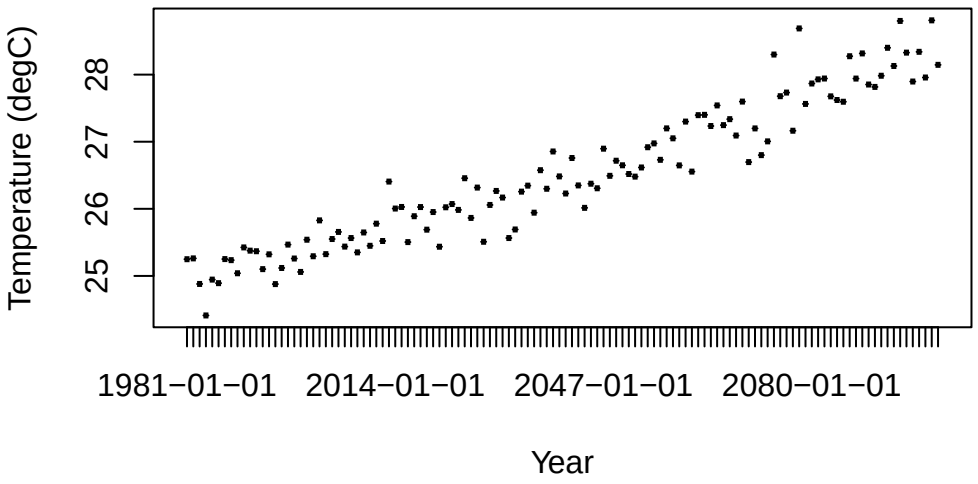
```
         x     y
[1,] 142.4 -13.9
```

```
df = melt(tas[50,50])
```

```
Using  as id variables
```

Basic plot

```
plot(df, xlab = "Year", ylab = "Temperature (degC)")
```



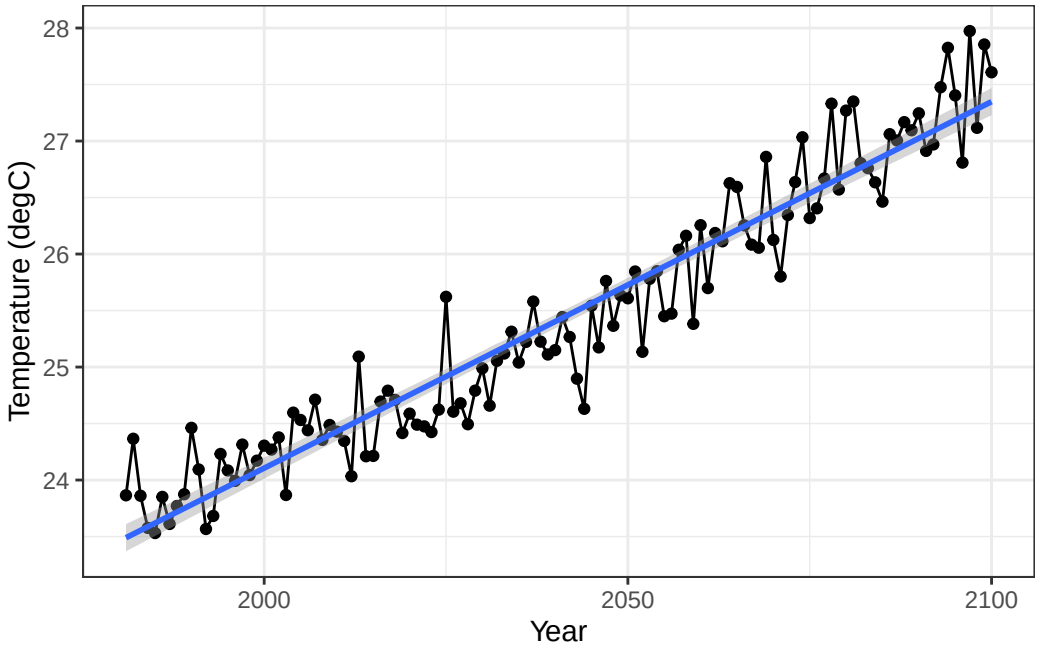
Calculating spatial average of all data

```
spat_ave = global(tas, fun=mean, na.rm=TRUE)
spat_ave$date = dates
```

ggplot

```
ggplot(data = spat_ave, aes(y=mean, x=date))+
  ylab('Temperature (degC)') + xlab('Year') +
  geom_point() +
  geom_line() +
  geom_smooth(method = "lm") +
  theme_bw()
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Q. Can you add another model to this plot and compare the two?

Hint: You'll need to prepare a dataframe with data for all models in it. One of the columns will need to be the values, and the other the model name.

Part 2: Rainfall Data (multiple models)

Working with multiple Models

```
pr_files <- list.files(path = "data/annual/", pattern = "pr", full.names = TRUE)
pr_files <- list.files(path = "data/annual/", pattern = "pr.*ssp370", full.names = TRUE)
```

```
pr_data = rast(pr_files)*365  # daily mean to annual total. CCAM has a 365 day calendar.
pr_data
```

```
class       : SpatRaster
dimensions  : 205, 176, 360  (nrow, ncol, nlyr)
resolution  : 0.1, 0.1  (x, y)
extent      : 137.45, 155.05, -29.45, -8.95  (xmin, xmax, ymin, ymax)
coord. ref. : lon/lat WGS 84 (CRS84) (OGC:CRS84)
source(s)   : memory
varname     : pr_annual (Seasonal average of Precipitation (Annual))
names       : pr_annual_1, pr_annual_2, pr_annual_3, pr_annual_4, pr_annual_5, pr_annual_6,
min values  :      52.196,    48.72456,    85.55378,    54.58001,    150.5908,    117.2378,
```

```
max values  :    6151.675,  7233.45194,  5595.35947,  6152.59279,   5482.5640,   5449.5375,
time (days) : 1981-01-01 to 2100-01-01 (2 steps)
```

Repeating the year names multiple times to correspond with multiple models

```
years = seq(1981,2100)
years_rep = rep(years, times =3)
names(pr_data) = years_rep
```

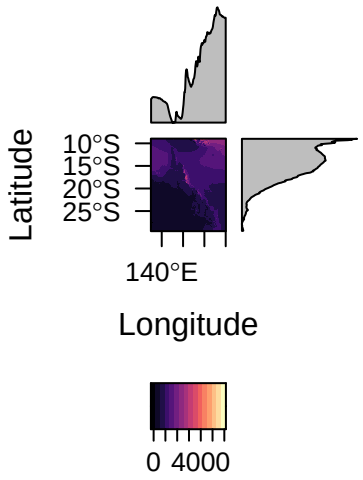
Calculating the model average

```
pr_modavg = tapp(pr_data, years, fun = mean)

# Calculating climatology for baseline (1981-2010)
pr_base = mean(pr_modavg[[1:30]])  # Converting from daily mean to annual mean

# Calculating climatology for future (2071-2100)
pr_fut = mean(pr_modavg[[91:120]])  # Converting from daily mean to annual mean
```

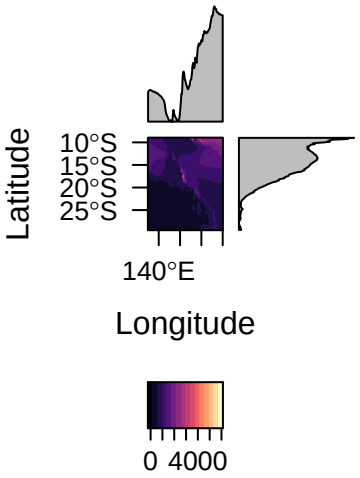
Q: Can you select the data based on the years instead?

Plot historic and future rainfall

```
levelplot(pr_base)
```

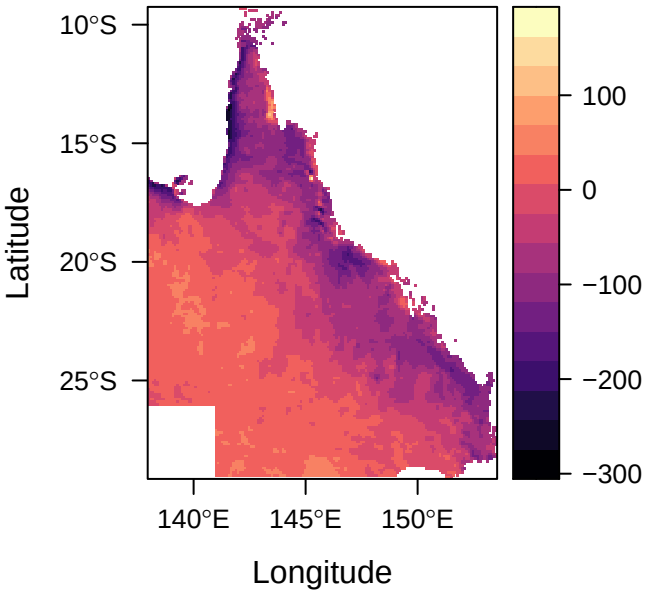



```
levelplot(pr_fut)
```



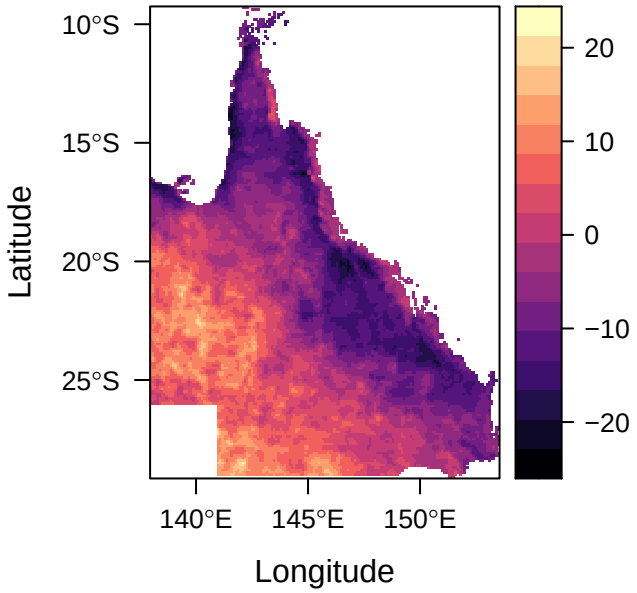
Cutting data to Queensland

```
qld_shp = vect('data/shp/QLD_State_Mask.shp')
pr_dif = pr_fut - pr_base
pr_dif_masked <- crop(pr_dif, qld_shp, mask = TRUE)
levelplot(pr_dif_masked, margin = FALSE)
```



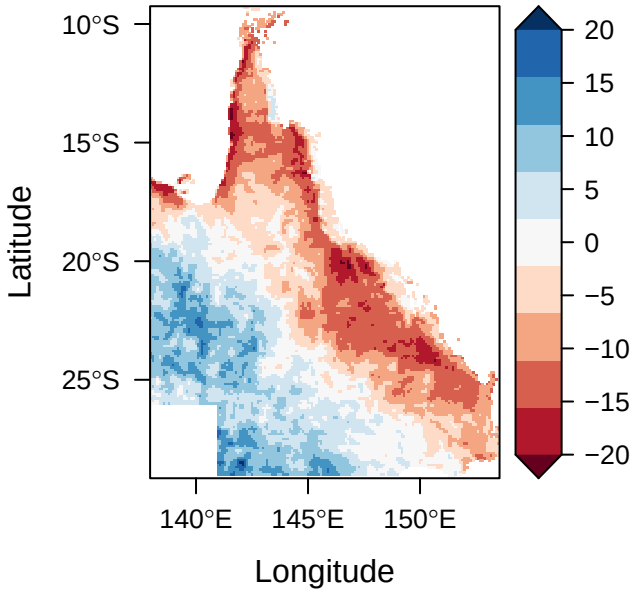
Plotting the percent change

```
pr_pdif = (pr_fut - pr_base ) / pr_base *100  #Percent difference
pr_pdif_masked <- crop(pr_pdif, qld_shp, mask = TRUE)
levelplot(pr_pdif_masked, margin = FALSE)
```



Specifying plotting bins and colours

```
my.at <- seq(-20, 20, length.out = 10)
my.at = c(-Inf, my.at, Inf)
levelplot(pr_pdif_masked, margin = FALSE, at = my.at, cuts=11, pretty=T,
                col.regions=((brewer.pal(11,"RdBu"))))
```



Q. Can you modify this plot to show more infomation? Can you add a title and change the colours?

Would showing multiple models on this plot help?

Q. Can we compare the results from SSP370 to another Scenario?

Part 3: Validating your data Calculating BioClim Indices

List monthly climate files

```
dir("data/monthly/")
```

```
 [1] "pr_ACCESS-CM2_ssp126_r2i1p1f1_CCAM10oc_aus-10i_10km_mon_1981-2100.nc"
 [2] "pr_ACCESS-CM2_ssp245_r2i1p1f1_CCAM10oc_aus-10i_10km_mon_1981-2100.nc"
 [3] "pr_ACCESS-CM2_ssp370_r2i1p1f1_CCAM10oc_aus-10i_10km_mon_1981-2100.nc"
 [4] "pr_EC-Earth3_ssp126_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
 [5] "pr_EC-Earth3_ssp245_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
 [6] "pr_EC-Earth3_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
 [7] "pr_GFDL-ESM4_ssp126_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
 [8] "pr_GFDL-ESM4_ssp245_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
 [9] "pr_GFDL-ESM4_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
[10] "tasmax_ACCESS-CM2_ssp126_r2i1p1f1_CCAM10oc_aus-10i_10km_mon_1981-2100.nc"
[11] "tasmax_ACCESS-CM2_ssp245_r2i1p1f1_CCAM10oc_aus-10i_10km_mon_1981-2100.nc"
[12] "tasmax_ACCESS-CM2_ssp370_r2i1p1f1_CCAM10oc_aus-10i_10km_mon_1981-2100.nc"
[13] "tasmax_EC-Earth3_ssp126_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
[14] "tasmax_EC-Earth3_ssp245_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
[15] "tasmax_EC-Earth3_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
[16] "tasmax_GFDL-ESM4_ssp126_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
[17] "tasmax_GFDL-ESM4_ssp245_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
[18] "tasmax_GFDL-ESM4_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
[19] "tasmin_ACCESS-CM2_ssp126_r2i1p1f1_CCAM10oc_aus-10i_10km_mon_1981-2100.nc"
[20] "tasmin_ACCESS-CM2_ssp245_r2i1p1f1_CCAM10oc_aus-10i_10km_mon_1981-2100.nc"
[21] "tasmin_ACCESS-CM2_ssp370_r2i1p1f1_CCAM10oc_aus-10i_10km_mon_1981-2100.nc"
[22] "tasmin_EC-Earth3_ssp126_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
[23] "tasmin_EC-Earth3_ssp245_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
[24] "tasmin_EC-Earth3_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
[25] "tasmin_GFDL-ESM4_ssp126_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
[26] "tasmin_GFDL-ESM4_ssp245_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
[27] "tasmin_GFDL-ESM4_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc"
```

Load shapefile for Sunshine Coast

```
lga_shp = vect('data/shp/SunshineCoast.shp')
```

Load monthly climate data

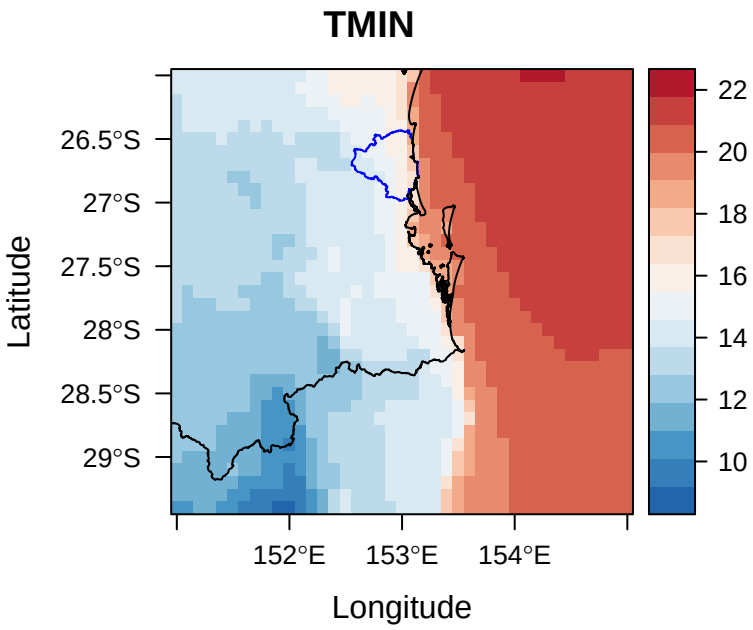
```
tmax = rast("data/monthly/tasmax_GFDL-ESM4_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100
tmin = rast("data/monthly/tasmin_GFDL-ESM4_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100
pr = rast("data/monthly/pr_GFDL-ESM4_ssp370_r1i1p1f1_CCAM10_aus-10i_10km_mon_1981-2100.nc")
```

Assign time labels

```
dates <- seq(as.Date("1981-01-01"), as.Date("2100-12-01"), by = "month")
names(tmax) = dates
names(tmin) = dates
names(pr) = dates
```

Plot temperature minimum

```
temp_stack <- mean(tmin[[1:360]])
lga_shp2 <- as(lga_shp, "Spatial")
qld2 <- as(qld_shp, "Spatial")
levelplot(temp_stack, margin = FALSE, par.settings = BuRdTheme, main = 'TMIN') +
  latticeExtra::layer(sp.polygons(lga_shp2, col = 'blue')) +
  latticeExtra::layer(sp.polygons(qld2))
```



Load observational data

```
dir("data/obs/")
```

```
[1] "agcd_v1_precip_r005_daily_1981_2020.nc"
[2] "agcd_v1_tmax_r005_daily_1981_2020.nc"
[3] "agcd_v1_tmin_r005_daily_1981_2020.nc"
```

```
obs_tmax = rast(list.files(path = "data/obs/", pattern = "tmax", full.names = TRUE))
obs_tmin = rast(list.files(path = "data/obs/", pattern = "tmin", full.names = TRUE))
obs_pr = rast(list.files(path = "data/obs/", pattern = "precip", full.names = TRUE))
```

Resample observational data

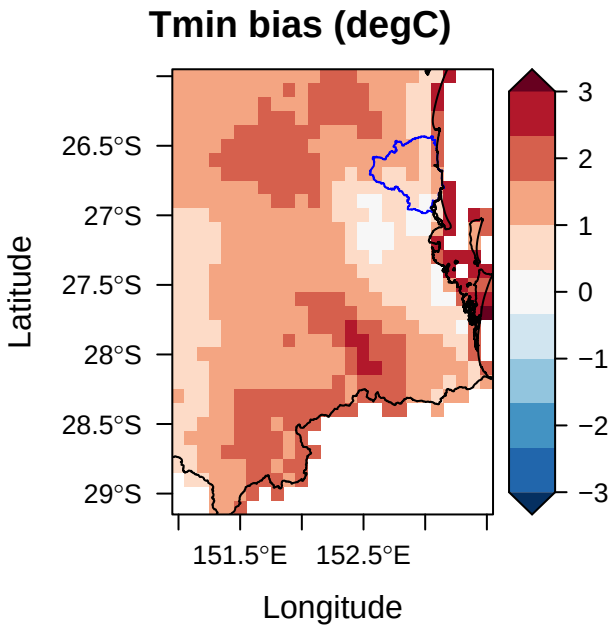
```
obs_tmax_regridded <- resample(obs_tmax, tmax, method = "bilinear")
obs_tmin_regridded <- resample(obs_tmin, tmin, method = "bilinear")
obs_pr_regridded <- resample(obs_pr, pr, method = "bilinear")
```

Extract historical slices

```
tmax_his = tmax[[dates >= as.Date("1981-01-01") & dates < as.Date("2021-01-01")]]
tmin_his = tmin[[dates >= as.Date("1981-01-01") & dates < as.Date("2021-01-01")]]
pr_his = pr[[dates >= as.Date("1981-01-01") & dates < as.Date("2021-01-01")]]
```

Evaluate tmin bias

```
tmin_bias = mean(tmin_his) - mean(obs_tmin_regridded)
tmin_bias_masked <- crop(tmin_bias, qld_shp, mask = TRUE)

my.at <- c(-Inf, seq(-3, 3, length.out = 10), Inf)
levelplot(tmin_bias_masked, at = my.at, margin = FALSE, cuts = 11, pretty = TRUE,
          main = 'Tmin bias (degC)',
          col.regions = rev(brewer.pal(11, "RdBu"))) +
  latticeExtra::layer(sp.polygons(lga_shp2, col = 'blue')) +
  latticeExtra::layer(sp.polygons(qld2))
```



Bias metrics

```
rmse = global((mean(tmin_his) - mean(obs_tmin_regridded))^2, fun = "mean", na.rm = TRUE)[1]
mape = global(abs((mean(tmin_his) - mean(obs_tmin_regridded)) / mean(obs_tmin_regridded)) *
print(paste("RMSE:", rmse))
```

```
[1] "RMSE: 3.48752996185524"
```

```
print(paste("MAPE:", mape))
```

```
[1] "MAPE: 11.8785189666455"
```

Mask climate data to Sunshine Coast

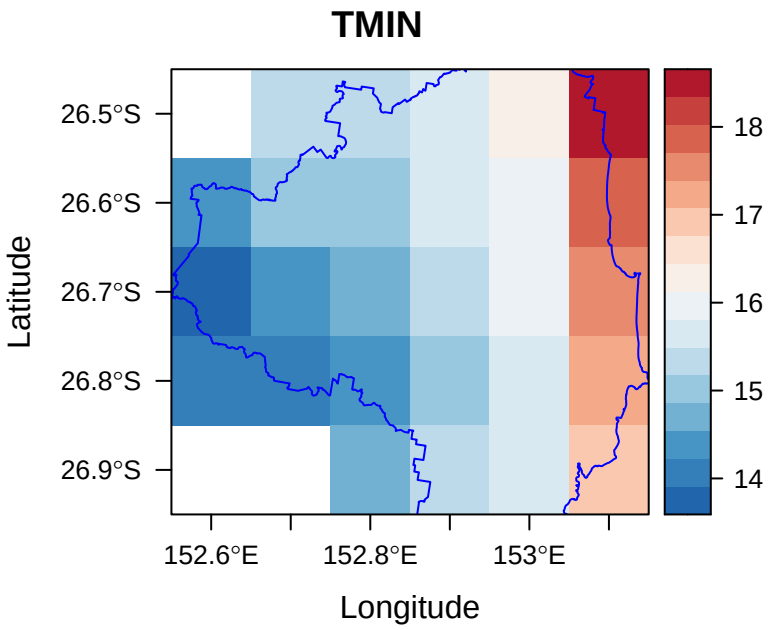
```
pr_masked <- crop(pr, lga_shp, mask = TRUE)
tmin_masked <- crop(tmin, lga_shp, mask = TRUE)
tmax_masked <- crop(tmax, lga_shp, mask = TRUE)
```

Spatial averages

```
pr_ave_coarse = global(pr_masked, fun = mean, na.rm = TRUE)
tmin_ave_coarse = global(tmin_masked, fun = mean, na.rm = TRUE)
tmax_ave_coarse = global(tmax_masked, fun = mean, na.rm = TRUE)
```

Plot masked tmin

```
tmin_masked_mean <- mean(tmin_masked[[1:360]])
levelplot(tmin_masked_mean, margin = FALSE, par.settings = BuRdTheme, main = 'TMIN') +
  latticeExtra::layer(sp.polygons(lga_shp2, col = 'blue'))
```

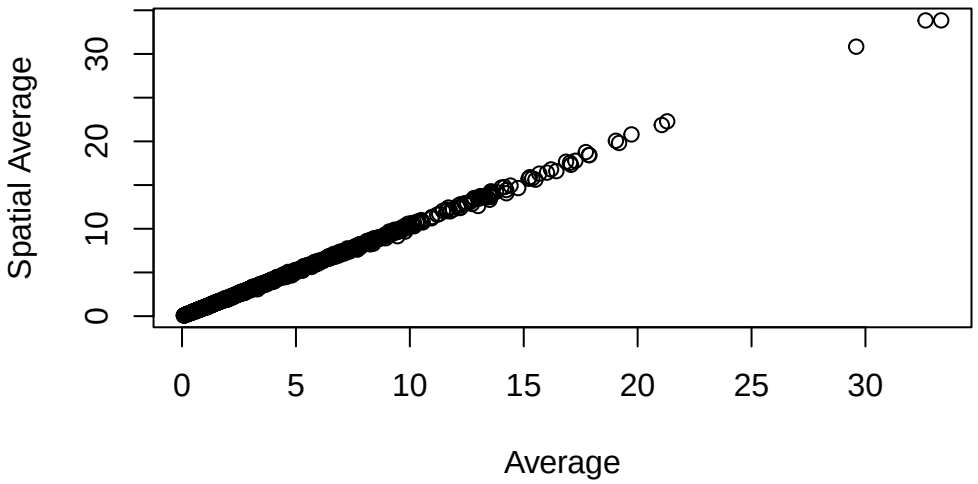


Weighted average

```
pr_ave = as.data.frame(t(terra::extract(pr, lga_shp, weights = TRUE, fun = mean, na.rm = TRU
tmin_ave = as.data.frame(t(terra::extract(tmin, lga_shp, weights = TRUE, fun = mean, na.rm =
tmax_ave = as.data.frame(t(terra::extract(tmax, lga_shp, weights = TRUE, fun = mean, na.rm =
```

Compare averages

```
par(mfrow = c(1, 1))
plot(pr_ave_coarse$mean, pr_ave$V1, xlab = "Average", ylab = "Spatial Average")
```



Prepare dataframe for biovars

```
colnames(pr_ave)[1] <- "pr"
colnames(tmin_ave)[1] <- "tmin"
colnames(tmax_ave)[1] <- "tmax"
pr_ave$date <- rownames(pr_ave)
tmin_ave$date <- rownames(tmin_ave)
tmax_ave$date <- rownames(tmax_ave)
df <- merge(pr_ave, tmin_ave, by = "date", all = TRUE)
df <- merge(df, tmax_ave, by = "date", all = TRUE)
```

Subset to baseline and future

```
df$date <- as.Date(df$date)
df$year <- as.numeric(format(df$date, "%Y"))
df_base = subset(df, year >= 1981 & year <= 2010)
df_fut = subset(df, year >= 2071 & year <= 2100)
```

Calculate biovars

```
bio_base = biovars(df_base$pr, df_base$tmin, df_base$tmax)
bio_fut = biovars(df_fut$pr, df_fut$tmin, df_fut$tmax)
print(bio_base[, c("bio5", "bio6", "bio12", "bio15")])
```

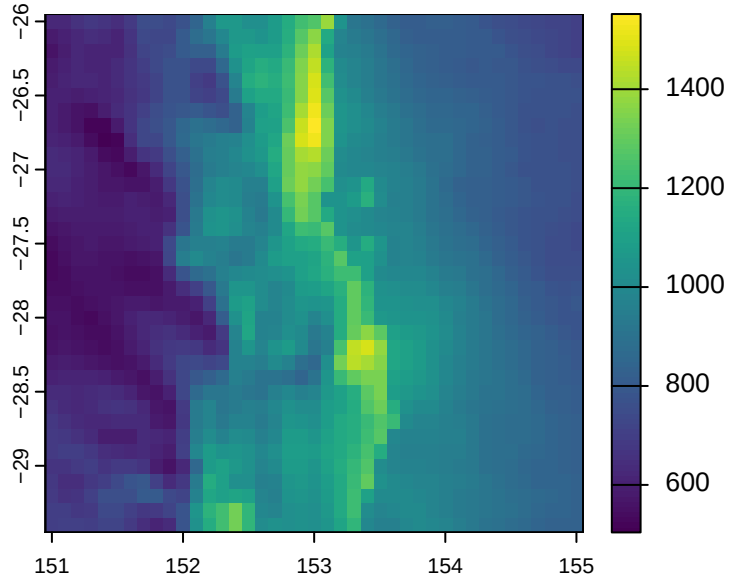
```
      bio5       bio6      bio12      bio15
 31.247510   8.574035 1343.710569  88.641010
```

```
print(bio_fut[, c("bio5", "bio6", "bio12", "bio15")])
```

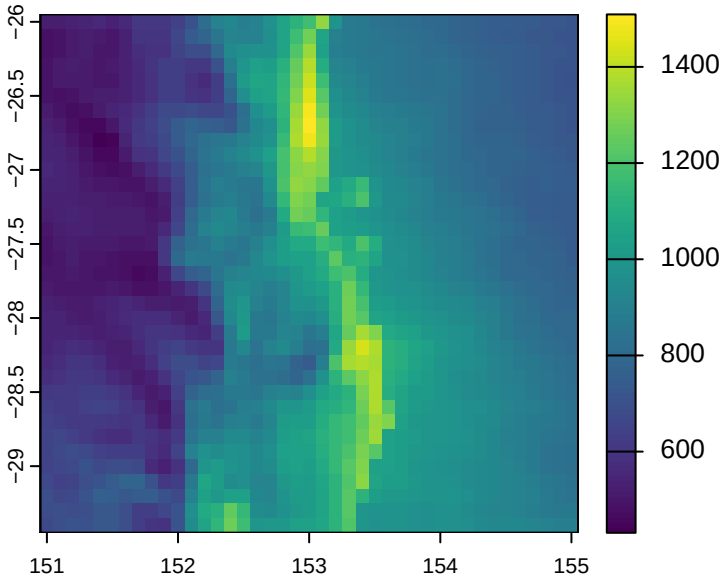
```
     bio5       bio6      bio12      bio15
 36.10810   11.37925 1266.43503   74.49997
```

Full raster bioclim calculation

```
tmax_base = tmax[[dates >= as.Date("1981-01-01") & dates < as.Date("2011-01-01")]]
tmin_base = tmin[[dates >= as.Date("1981-01-01") & dates < as.Date("2011-01-01")]]
pr_base = pr[[dates >= as.Date("1981-01-01") & dates < as.Date("2011-01-01")]]
tmax_fut = tmax[[dates >= as.Date("2071-01-01") & dates < as.Date("2101-01-01")]]
tmin_fut = tmin[[dates >= as.Date("2071-01-01") & dates < as.Date("2101-01-01")]]
pr_fut = pr[[dates >= as.Date("2071-01-01") & dates < as.Date("2101-01-01")]]

bioclim_input_base = c(pr_base, tmin_base, tmax_base)
bioclim_input_fut = c(pr_fut, tmin_fut, tmax_fut)

fun_bio_calc <- function(x) {
  n <- length(x) / 3
  pr <- x[1:n]
  tmin <- x[(n + 1):(2 * n)]
  tmax <- x[(2 * n + 1):(3 * n)]
  bio <- biovars(pr, tmin, tmax)
  return(bio)
}

bio_base <- app(bioclim_input_base, fun = fun_bio_calc)
bio_fut <- app(bioclim_input_fut, fun = fun_bio_calc)

plot(bio_base[[12]])
```



```
plot(bio_fut[[12]])
```



Discussion Questions

Q: Can you plot the bioclimatic indices in the past and present and compare the changes?

Q: What does the bioclimatic indicators look like for a lower emissions scenario?