`rWBclimate`

Introduction

rWBclimate is an R interface for the World Bank climate data used in the World Bank climate knowledge portal.

Package description

`rWBclimate` provides access to three different classes of climate data at two different spatial scales. The three different classes of data are GCM model , ensemble and historical data. Each data class will let you download two different four different types for two different variables. The two variables are either precipitation expressed in millimeters or temperature as degrees celcius, and for each variable you can download your data in one of four types. The data is fully described below along with examples.

Data classes

Model data

Almost all model data in the Climate Data API are derived from 15 global circulation models (GCMs) used by the Intergovernmental Panel on Climate Change (IPCC) 4th Assessment Reports. The models simulate the response of the global climate system to increasing greenhouse gas concentrations. The data in the Climate Data API have been aggregated to both the country and basin levels, as explained below. Note these data are modeled estimates of temperature and precipitation changes in different time periods under different GCMs and scenarios. They include changes for future time periods and also as "backcasting" (model representations of the past) set for past time periods. The latter should not be confused with any instrumental or observed data. There is a specific dataset with historical measured climate data as well.

Data types

Type	Description
Monthly average	The monthly average for all 12 months for a given time period
Annual average	a single average for a given time period
Monthly anomaly	Average monthly change (anomaly). The control period is 1961-1999 for temperature and precipitation
Annual anomaly	Average annual change (anomaly). The control period is 1961-1999 for temperature and precipitation v

Data time scales

Climate model data is only available as averages for 20 year chunks. The package will automatically convert any start and end data into valid API calls and will return all data between the given start and end date. The following time periods are available

Past		Future	
start	*end*	*start*	*end*
1920	1939	2020	2039
1940	1959	2040	2059
1960	1979	2060	2079

Past		Future	
1980	1999	2080	2099

Data spatial scales Data is available at two spatial scales. The first is country level. You can download data for any country in the world using a valid ISO 3 letter country code. Alternatively you can download data at the basin network for a slightly higher resolution represented as a number 1-468. This is based on level 2 boundaries from waterbase.org, or you can view a full map of all the available basins.

Downloading GCM Model Data

Model data is downloaded for two different scenarios, the A2 and B1. Generally A2 scenarios are where there is little difference between the future and now and B1 is a more ecologically friendly world with greater decrease in emissions. Both the A2 and the B1 will be downloaded for 15 different GCM models listed in the table below:

Name in output	Model name
bccr_bcm2_0	BCM 2.0
csiro_mk3_5	CSIRO Mark 3.5
ingv_echam4	ECHAM 4.6
cccma_cgcm3_1	CGCM 3.1 (T47)
cnrm_cm3	CNRM CM3
gfdl_cm2_0	GFDL CM2.0
gfdl_cm2_1	GFDL CM2.1
ipsl_cm4	IPSL-CM4
microc3_2_medres	MIROC 3.2 (medres)
miub_echo_g	ECHO-G
mpi_echam5	ECHAM5/MPI-OM
mri_cgcm2_3_2a	MRI-CGCM2.3.2
inmcm3_0	INMCM3.0
ukmo_hadcm3	UKMO HadCM3
ukmo_hadgem1	UKMO HadGEM1

The model data can be downloaded with two main functions:

```
get_model_temp()  ## Get model temperature data
get_model_precip() ## Get model precipitation data
```

Example 1: Plotting monthly data from different GCM's Say you want to compare temperature from two different models in the USA to see how they vary. You can download data for the USA and then subset it to the specific models you're interested in and then plot them.

```
usa.dat <- get_model_temp("USA", "mavg", 2080, 2100)
usa.dat.bcc <- usa.dat[usa.dat$gcm == "bccr_bcm2_0", ]
usa.dat.had <- usa.dat[usa.dat$gcm == "ukmo_hadcm3", ]
```

```
## Add a unique ID to each for easier plotting
usa.dat.bcc$ID <- paste(usa.dat.bcc$scenario, usa.dat.bcc$gcm, sep = "-")
usa.dat.had$ID <- paste(usa.dat.had$scenario, usa.dat.had$gcm, sep = "-")
plot.df <- rbind(usa.dat.bcc, usa.dat.had)
ggplot(plot.df, aes(x = as.factor(month), y = data, group = ID, colour = gcm,
    linetype = scenario)) + geom_point() + geom_path() + ylab("Average temperature in degrees C \n betw
    xlab("Month") + theme_bw()
```

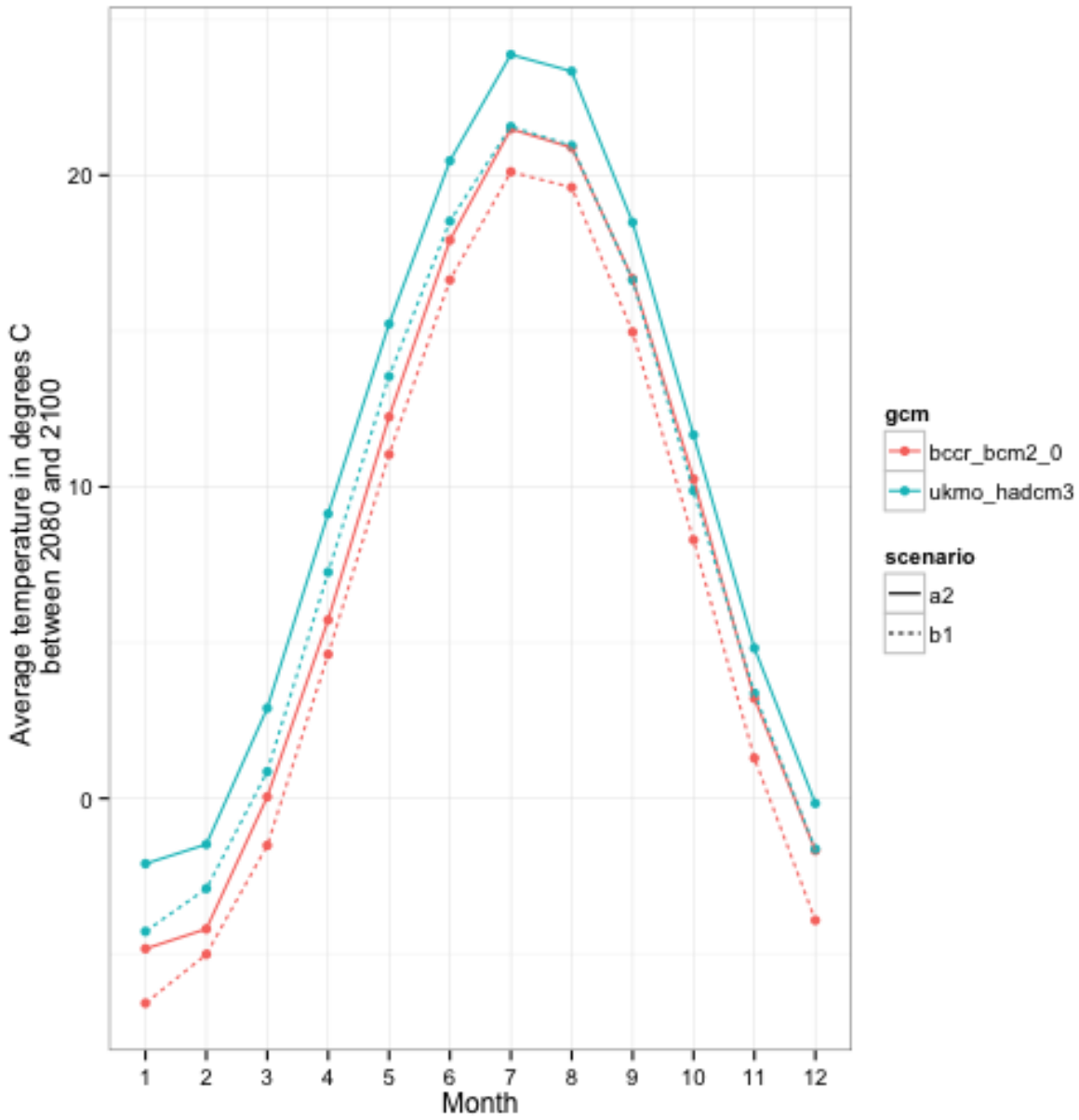

Figure 1: plot of chunk getmodeldata

Subsetting all the data can be a bit tedious. You could also compare all the models but just for one scenario, the A2.

```
ggplot(usa.dat[usa.dat$scenario == "a2", ], aes(x = month, y = data, group = gcm,
    colour = gcm)) + geom_point() + geom_path() + ylab("Average temperature in degrees C \n between 208
    xlab("Month") + theme_bw()
```

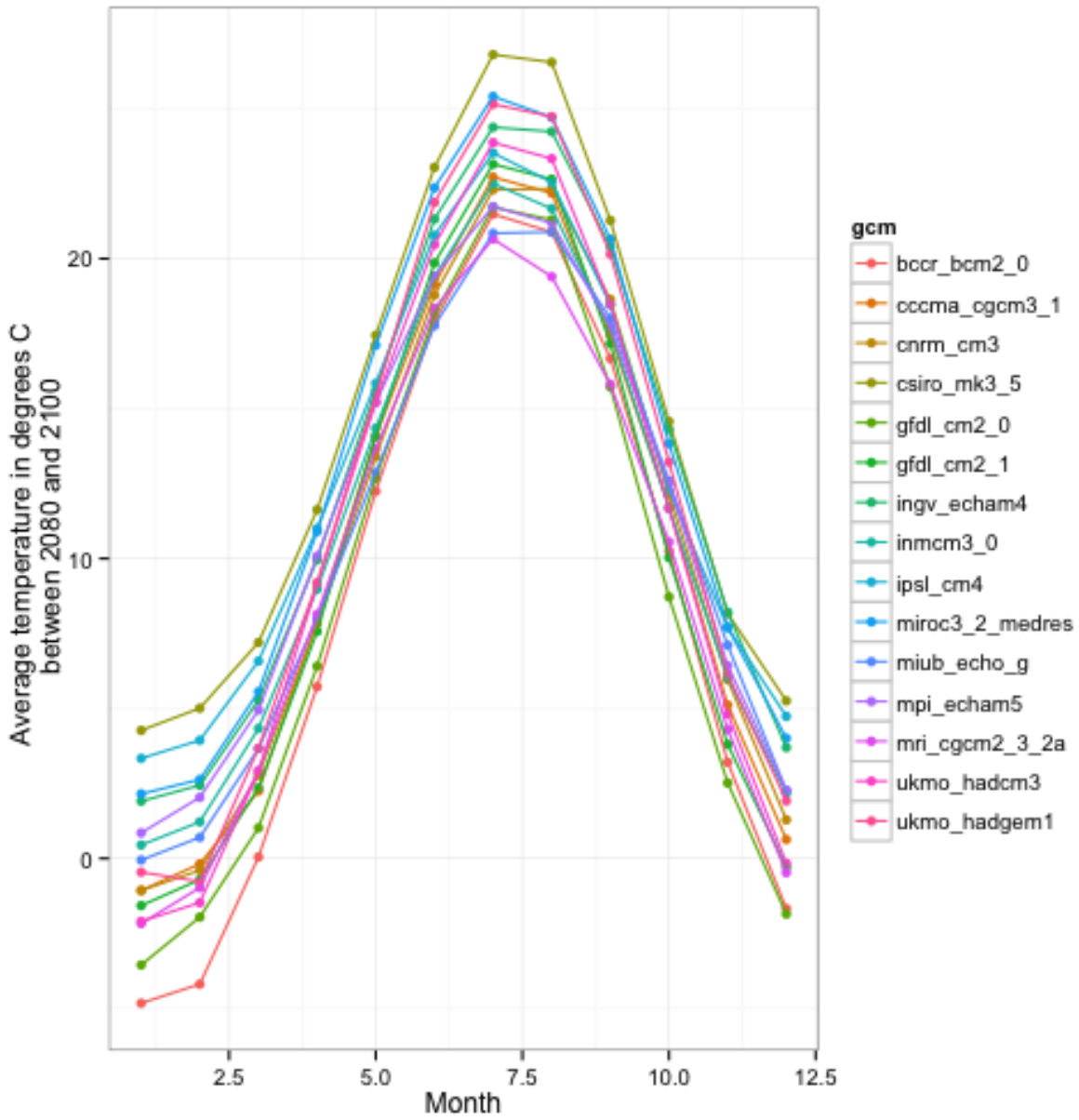

Figure 2: plot of chunk plotallmodeldata

Example 2: Plotting annual data for different countries

Data can be extracted from countries or basins submitted as vectors. Here we will plot the expected temperature anomaly for each 20 year period over a baseline control period of 1961-2000. These countries chosen span the north to south pole. It's clear from the plot that the northern most countries (US and Canada) have the biggest anomaly, and Belize, the most equatorial country, has the smallest anomaly.

```
country.list <- c("CAN", "USA", "MEX", "BLZ", "COL", "PER", "BOL", "ARG")
country.dat <- get_model_temp(country.list, "annualanom", 2020, 2100)
# Subset data
country.dat.bcc <- country.dat[country.dat$gcm == "bccr_bcm2_0", ]
## Exclude A2 scenario
country.dat.bcc <- subset(country.dat.bcc, country.dat.bcc$scenario != "a2")
ggplot(country.dat.bcc, aes(x = fromYear, y = data, group = locator, colour = locator)) +
    geom_point() + geom_path() + ylab("Temperature anomaly over baseline") +
    theme_bw()
```

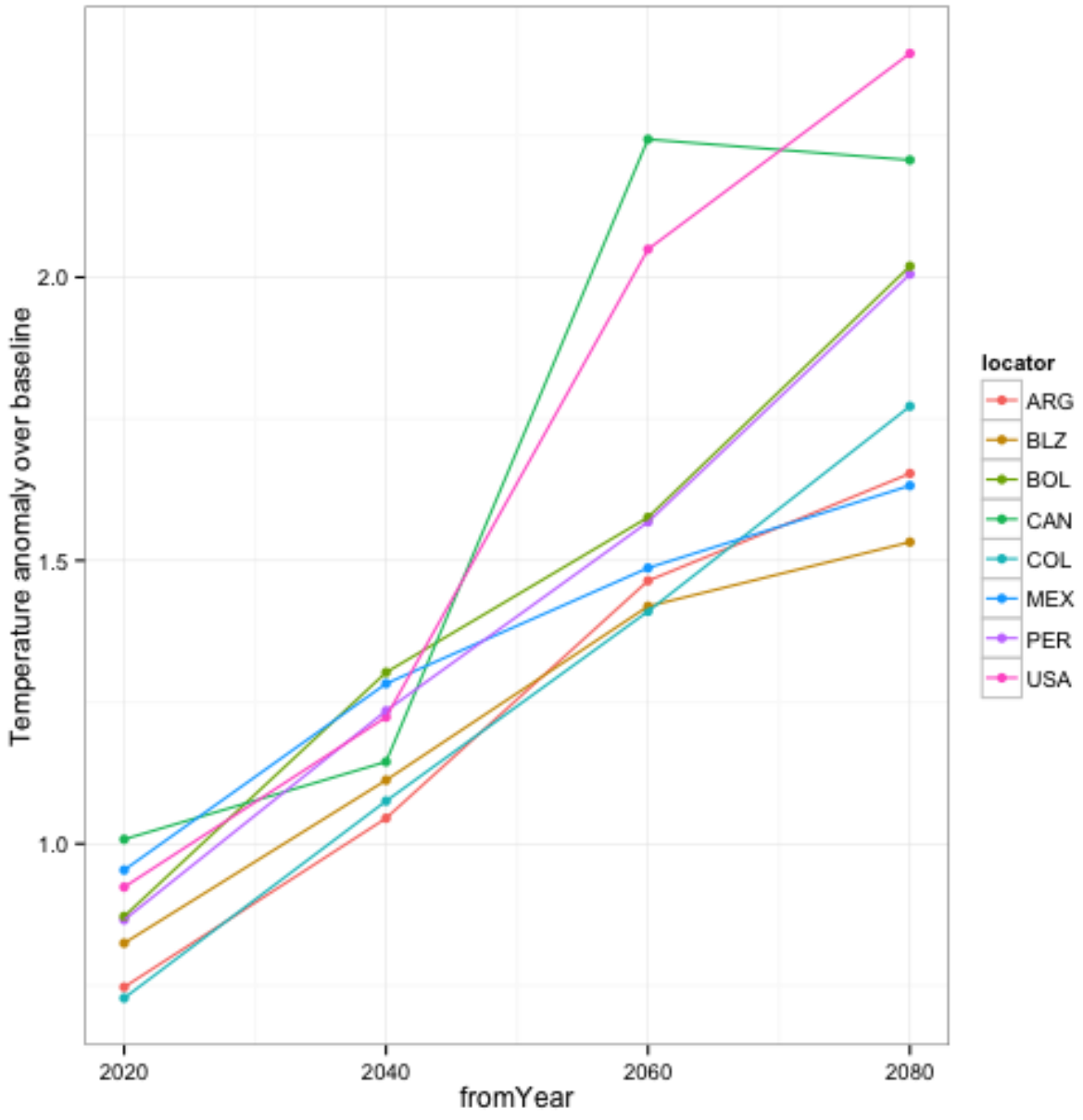

Figure 3: plot of chunk annualdata

Downloading Ensemble Model Data

Getting raw model data is really useful for comparing scenarios or different GCM's. However many users might be more interested in aggregated measurements. If so you can download ensemble climate data. This will give access to all 15 GCM's combined and queries will return the 10th, 50th and 90th quantiles for the ensemble of all GCM's. All queries are constructed the same as raw model data above with the same data types, spatial scales and time scales.

Example 1: Comparing model quantiles

Let's look at monthly precipitation predictions for Indonesia for the period of 2080-2100. We'll create a plot of the precipitation expected in the future for the two different scenarios and plot them with different colours and dashed lines to indicate quantiles.

```
idn.dat <- get_ensemble_precip("IDN", "mavg", 2080, 2100)
# Set line types
ltype <- rep(1, dim(idn.dat)[1])
ltype[idn.dat$percentile != 50] <- 2
idn.dat$ltype <- ltype
# Create uniqueIDs
idn.dat$uid <- paste(idn.dat$scenario, idn.dat$percentile, sep = "-")
ggplot(idn.dat, aes(x = as.factor(month), y = data, group = uid, colour = scenario,
    linetype = as.factor(ltype))) + geom_point() + geom_path() + xlab("Month") +
    ylab("Rain in mm") + theme_bw()
```

Example 2: Ensemble statistics

You can also download 13 different ensemble statistics about basins or countries aside from the raw data as above. These derived statistics are often given relative to a control period, 1961-2000. The time periods however are only for 2046-2065, or 2081-2100, not the standard 20 year intervals available for raw model data. Below is a full table of available statistics.

Parameter name	Description
tmin_means	Average daily minimum temperature
tmax_means	Average daily maximum temperature
tmax_days90th	Number of days with maximum temperature above the control period's 90th percentile (hot days)
tmin_days90th	Number of days with minimum temperature above the control period's 90th percentile (warm nights)
tmax_days10th	Number of days with maximum temperature below the control period's 10th percentile (cool days)
tmin_days10th	Number of days with minimum temperature below the control period's 10th percentile (cold nights)
tmin_days0	Number of days with minimum temperature below 0 degrees Celsius
ppt_days	Number of days with precipitation greater than 0.2 mm
ppt_days2	Number of days with precipitation greater than 2 mm
ppt_days10	Number of days with precipitation greater than 10 mm
ppt_days90th	Number of days with precipitation greater than the control period's 90th percentile
ppt_dryspell	Average number of days between precipitation events
ppt_means	Average daily precipitation

Similar to our previous example where we looked at temperature anomaly along a latitudinal gradient, we can examine more complex ensemble statistics. Here we examine the average minimum daily temperature in Scavavadian countries. Also given that only two time periods are available, there is no need to enter in a

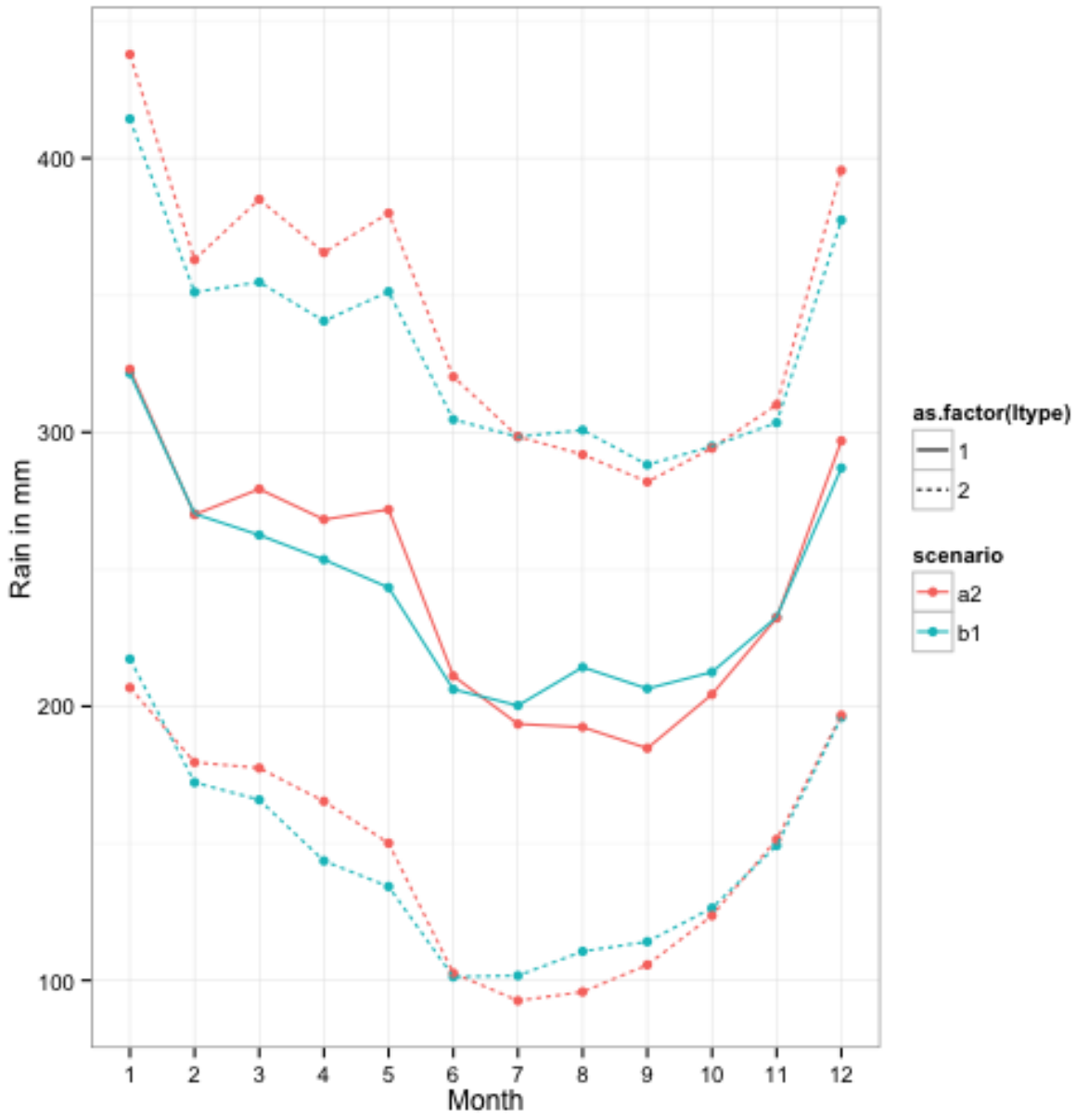

Figure 4: plot of chunk comparing quantiles

specific year range

```
country.list <- c("ISL", "FIN", "NOR", "SWE")
country.dat <- get_ensemble_stats(country.list, "mavg", "tmin_means")
####### Subset data Exclude A2 scenario
country.dat.b1 <- subset(country.dat, country.dat$scenario == "b1")
# choose just one percentile
country.dat.b1 <- subset(country.dat.b1, country.dat.b1$percentile == 50)
# get just one year period
country.dat.b1 <- subset(country.dat.b1, country.dat.b1$fromYear == 2081)


ggplot(country.dat.b1, aes(x = month, y = data, group = locator, colour = locator)) +
    geom_point() + geom_path() + ylab("Average daily minimum temperature") +
    theme_bw() + xlab("Month")
```

Downloading Historical Data

It is possible to extract historical data from GCM queries, but this is acutally model backcasting output, not true historical records. The climate data api can download data at the same spatial scales as all other requests (country or basin), but the time scale differs. You can download data at monthly, yearly or decadanal time scales. Monthly data is actually the mean monthly temperature or precipitation value for each month from 1901-2009 for countries, or 1960-2009 for basins. You can indicate the type of data you want in the call by setting the `time_scale` parameter to *month*,*year* or *decade*.

Example 1: Downloading monthly data

You can download historical precipitation for Belize, Colombia, Peru and Bolivia.

```
country.list <- c("BLZ", "COL", "PER", "BOL")
country.dat <- get_historical_precip(country.list, "month")

ggplot(country.dat, aes(x = month, y = data, group = locator, colour = locator)) +
    geom_point() + geom_path() + ylab("Average historical precipitation (mm)") +
    theme_bw() + xlab("Month")
```

Example 2: Downloading annual data

Another use of historical data is to look at increases in temperature over time.

```
country.list <- c("USA", "MEX", "CAN", "BLZ")
country.dat <- get_historical_temp(country.list, "year")

ggplot(country.dat, aes(x = year, y = data, group = locator)) + geom_point() +
    geom_path() + ylab("Average annual temperature of Canada") + theme_bw() +
    xlab("Year") + stat_smooth(se = F, colour = "black") + facet_wrap(~locator,
    scale = "free")
```

Mapping climate Data

Data can be mapped in ggplot2 by downloading KML files from the climate database, creating dataframes and plotting using ggplot2. Maps are available at the basin or country spatial scale. Because KML files can be large files are downloaded and locally cached in a directory you must set using the kmlpath option. You can set to any directory as follows: `options(kmlpath = <yourpath>)`. KML files will be stored there and only downloaded again if they aren't found locally. If you wish to get rid of them, you will need to manually delete them.

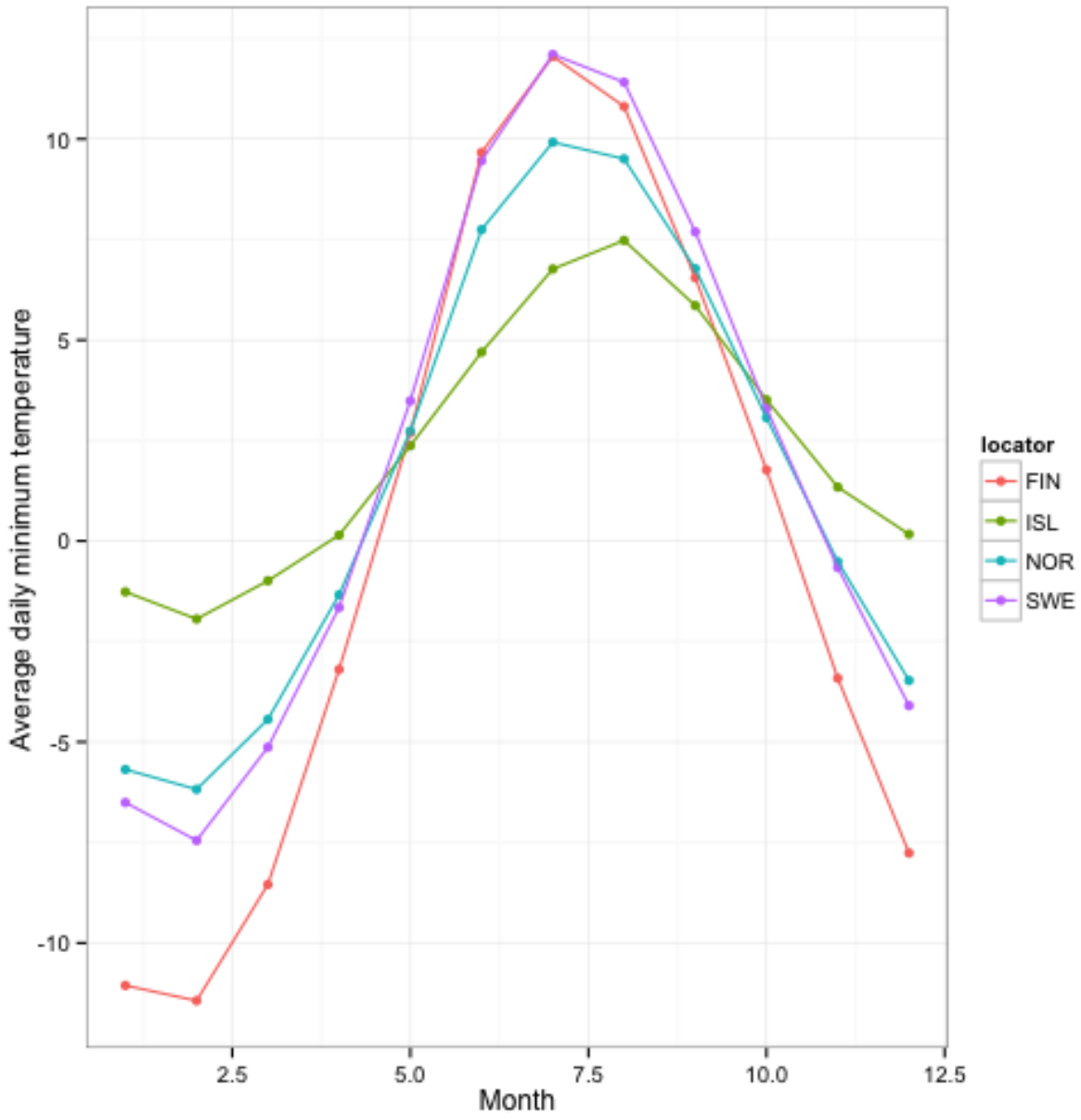

Figure 5: plot of chunk enesmble data

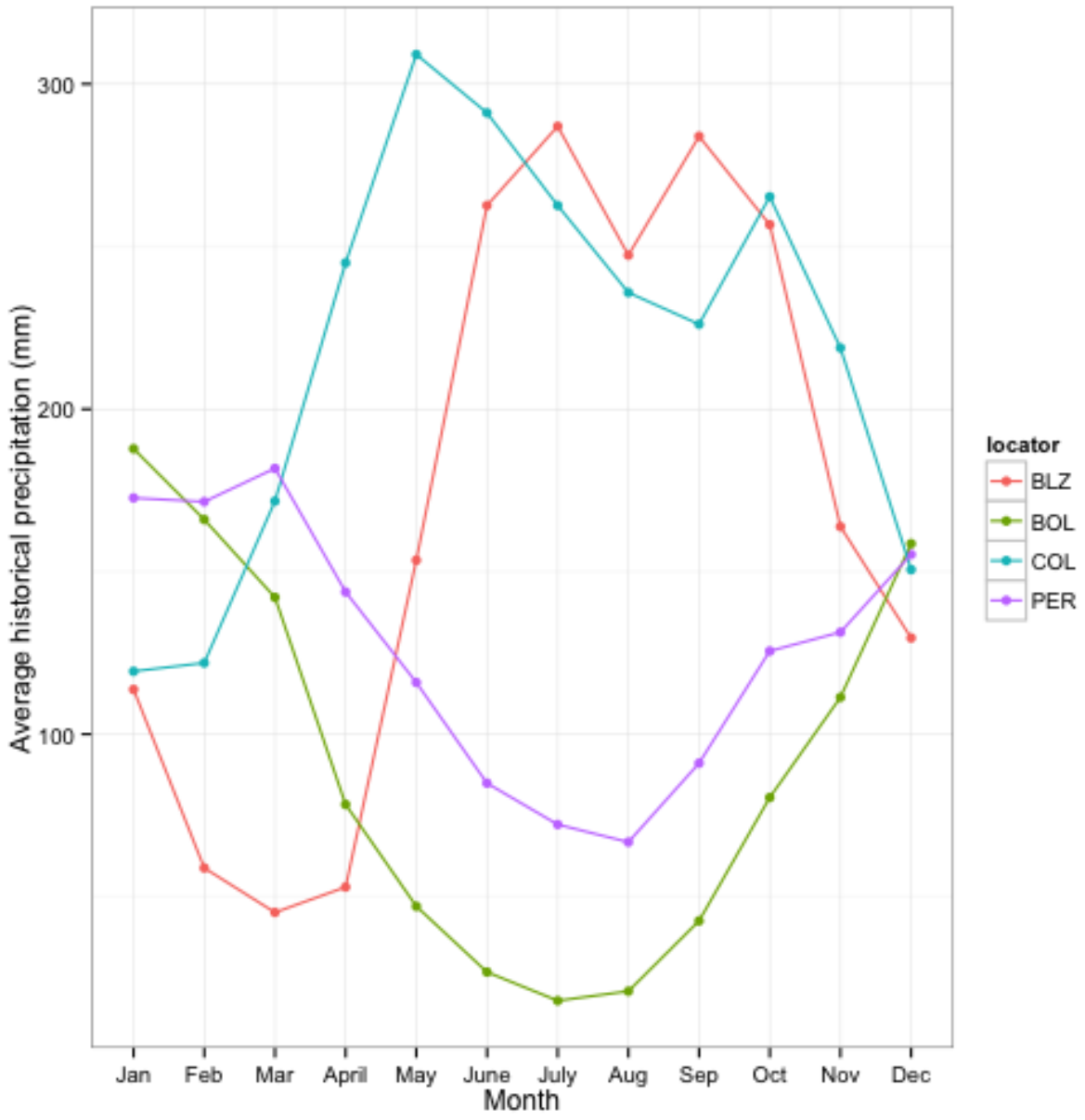

Figure 6: plot of chunk historicalmonth

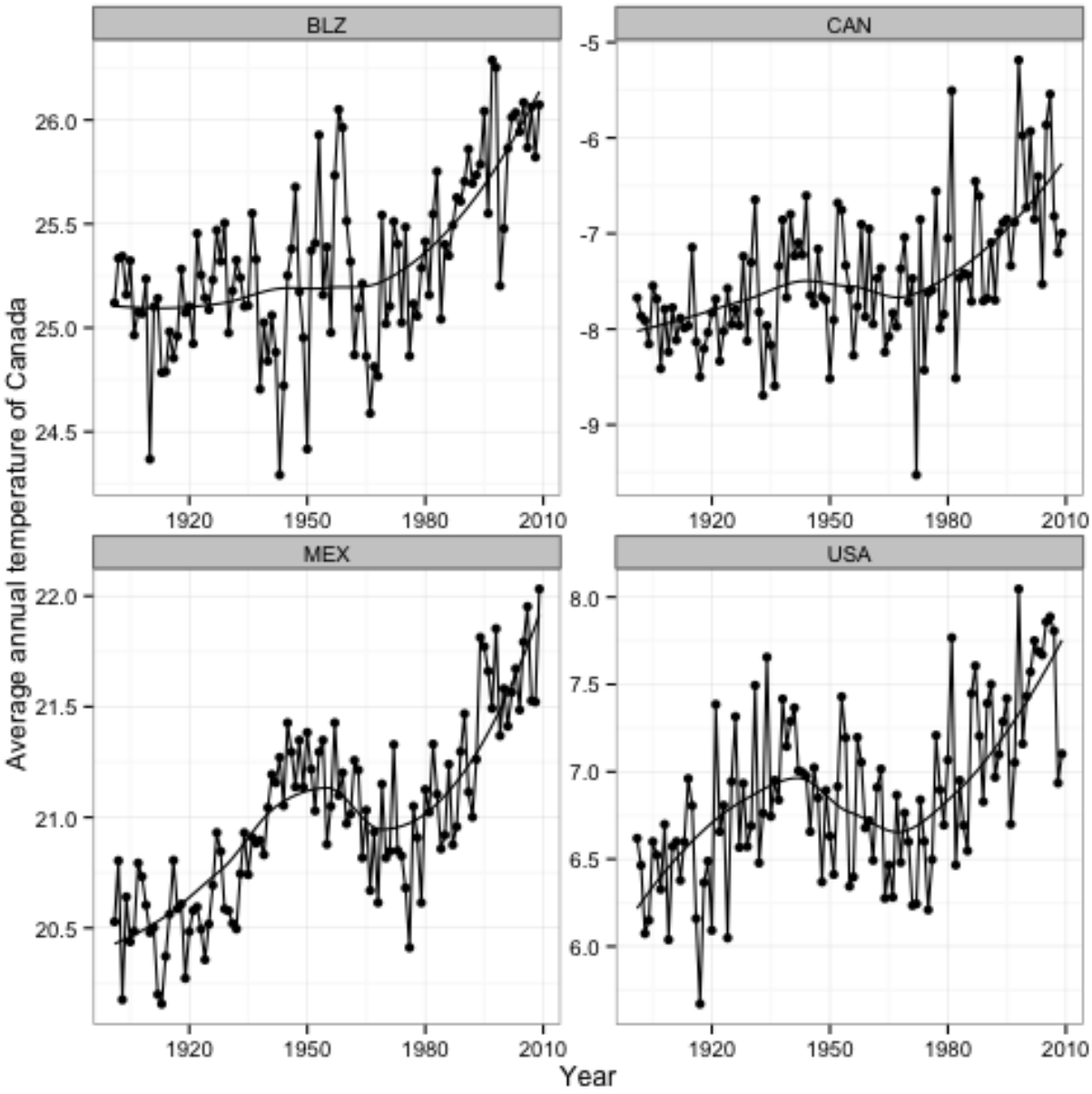

Figure 7: plot of chunk historicalyear

`rWBclimate` allows you to download maps and plot them as ggplot maps, or it can automatically create maps of climate data. For your convenience we've created basin and country vectors for all six continents *(Antartica has no data in the database)*. These data files are automatically loaded with the package and can be accessed for basins as *NoAm_basin*, *SoAm_basin*,*Asia_basin*,*Eur_basin*,*Africa_basin*,and *Oceana_basin*. Countries can be similarly accessed as *NoAm_country*, *SoAm_country*,*Asia_country*,*Eur_country*,*Africa_country*,and *Oceana_country*. If you are plotting by continent, it can take some time to first download all the KML files.

Example 1: Creating map dataframe and plotting

Creating map data frames is straightforward, simply provide a list of valid country codes to the function `create_map_df`

```
# Set the kmlpath option
options(kmlpath = "~/kmltemp")
## Here we use a list basins for Africa
af_basin <- create_map_df(Africa_basin)
```

```
ggplot(af_basin, aes(x = long, y = lat, group = group)) + geom_polygon() + theme_bw()
```

Example 2: Mapping climate data

In order to map climate data you need to have a single point of data for each spatial polygon(country or basin) in your map dataframe. We've already created the Africa basin map dataframe, so now you need to get some data for each basin.

```
af_basin_dat <- get_ensemble_temp(Africa_basin, "annualanom", 2080, 2100)
## Subset data to just one scenario, and one percentile
af_basin_dat <- subset(af_basin_dat, af_basin_dat$scenario == "a2")
af_basin_dat <- subset(af_basin_dat, af_basin_dat$percentile == 50)
```

Now that we have both the map dataframe and the data we can bind them together with the function `climate_map()`. The function has two options, it can return a `ggplot2` map or it can return a dataframe that you can plot easily with `ggplot2`. This is useful because it means that you could bind together multiple dataframes for bigger plots, or perhaps add new data yourself

```
af_map <- climate_map(af_basin, af_basin_dat, return_map = TRUE)
af_map + scale_fill_continuous("Temperature \n anomaly", low = "yellow", high = "red") +
    theme_bw()
```

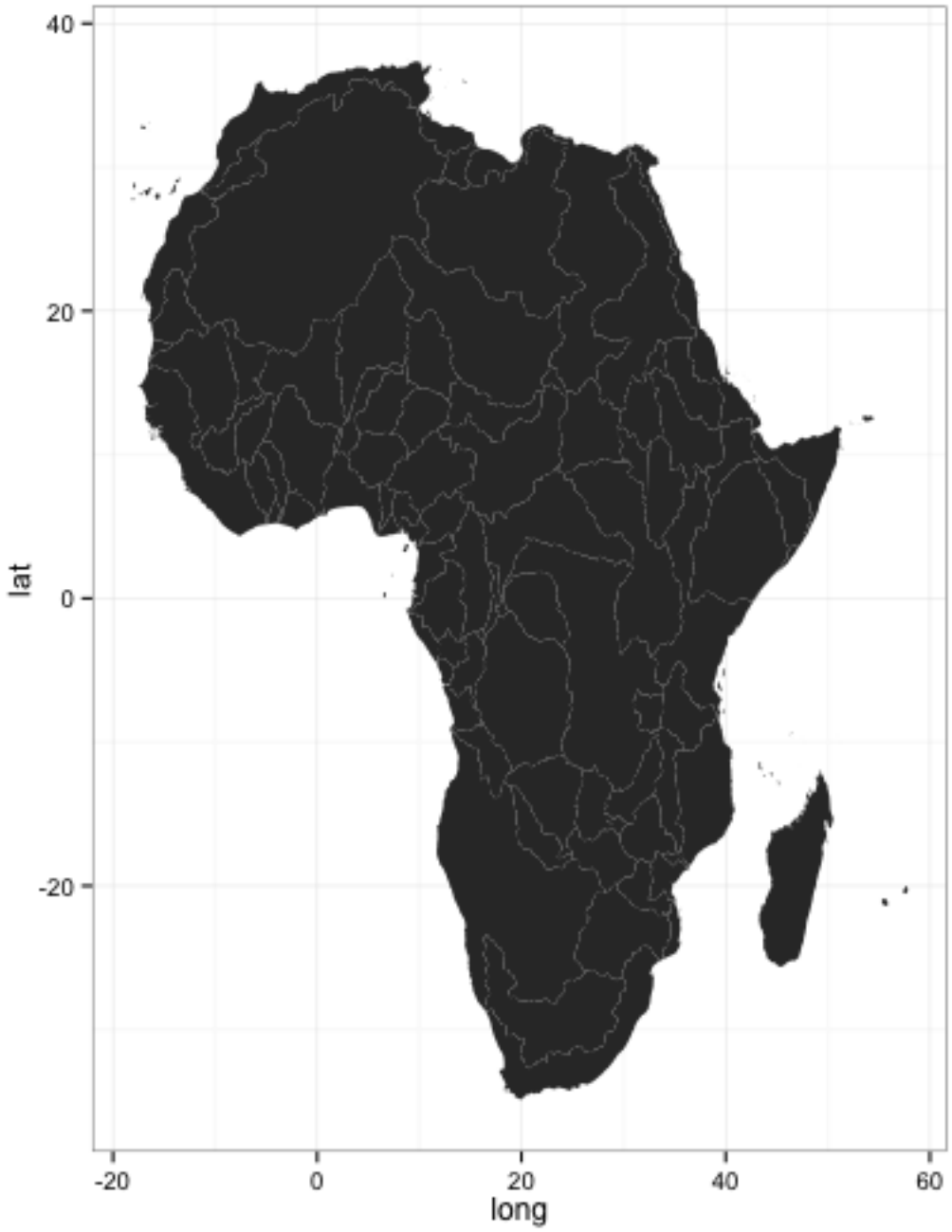

Figure 8: plot of chunk map_plot

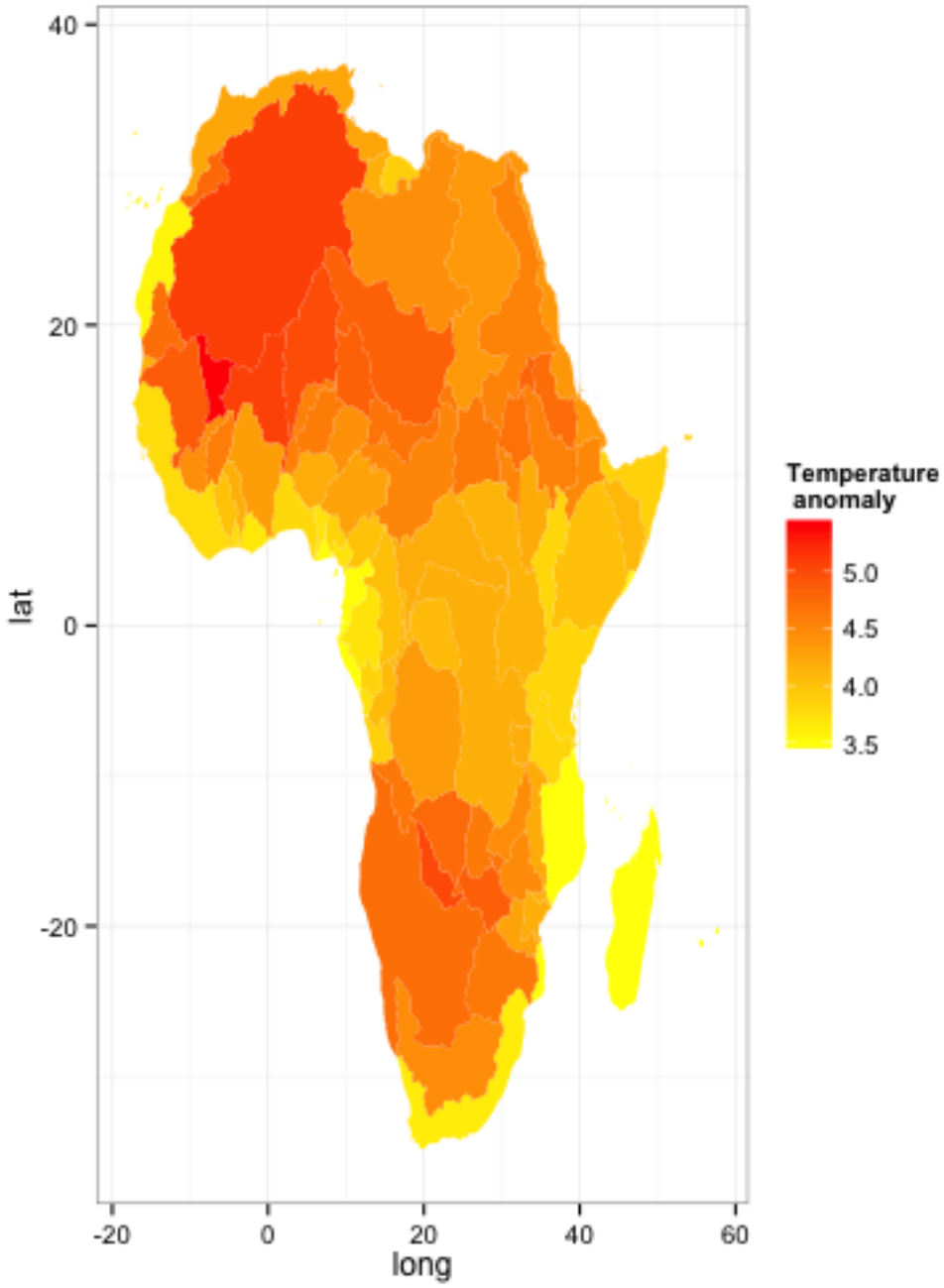

Figure 9: plot of chunk climatemap