An Exploration of temperature and solar radiation using WorldClim 2 database

DATA 200 Empirical Project
Yingxin Lin
12/20/2021

Abstract

This project used the WorldClim 2 database to explore the relationship between solar radiation and temperature. The WorldClim 2 Historical climate data contains aggregated data for maximum temperature, minimum temperature, solar radiation, precipitation, and wind speed from all over the world for the period 1970-2000. This project mainly tested some theories, for example, verifying common knowledge that when solar radiation is concentrated in smaller surface areas, it causes temperatures to rise; verifying classical models that use temperature to predict solar radiation. I use ablation studies and linear regression to delve into the relationships between maximum temperature, minimum temperature, temperature range, solar radiation, latitude, and longitude.

Section 1: Introduction

In this project, I used the Historical Climate Data in the WorldClim version 2 database. This database includes spatially monthly climate data for global land areas with a very high spatial resolution(from approximately 1 km^2 to 340 km^2). The Historical Climate data includes monthly temperature (minimum, maximum, and average), precipitation, solar radiation, vapor pressure, and wind speed. The data was "aggregated across a target temporal range of 1970–2000, using data from between 9000 and 60 000 weather stations"[1]. However, there is some area with lower weather station density. Therefore, satellite-derived data and other covariables were used for interpolation. In the model Fick et al. [1] built, "Global cross-validation correlations were≥0.99 for temperature and humidity, 0.86 for precipitation and 0.76 for wind speed".

This project mainly tested three theories. Firstly, I tested the common sense that air temperature can be increased by solar radiation heating. I assumed that solar radiation had a significant and obvious effect on the extremes and ranges of temperature, even though latitude, terrain, and season also have an effect. In addition to the causal relationship that solar radiation will cause temperatures to rise, I was also curious about if the temperature could be used to predict the intensity of solar radiation.

Secondly, I tested the theory in Bristow's et al. [2] paper "On the relationship between incoming solar radiation and daily maximum and minimum temperature". They used data of solar irradiance and the range in daily temperature extremes, which were collected at Pullman, Great Falls, Montana Seattle/Tacoma, and Washington, on a daily basis between 1 June 1980 and 31 May 1981. And they finally developed a relationship between atmospheric transmittance and the daily range of air temperature, which is

$$T_t = A[1 - \exp(-B\Delta T^C)] \quad (1)$$

where "T_t is the daily total atmospheric transmittance, ΔT is the daily range of air temperature, and A, B, and C are empirical coefficients, determined for a particular location from measured solar radiation data"[2].

In this equation, it used the daily total transmission coefficient T_t instead of using daily solar irradiance. To clarify, solar irradiance is the ratio of the incident radiant power per unit area of the surface. While in the WorldClim 2 database, the variable solar radiation is the calculated extraterrestrial radiation (i.e. the solar radiation without atmospheric effects). The temperature is collected daily in Bristow's paper, while the historical climate data in the WorldClim database was aggregated from 1970 to 2000.

In this project, I reviewed some relevant literature about the relationship between solar radiation and temperature in Section 2. I introduced the data set and the preprocessing steps and showed some preliminary graphs to learn their relationships in Section 3. In Section 4, I performed linear regression in R with solar radiation, maximum and minimum temperatures, latitude, and longitude as different combinations of response and predictors. In Section 5 I compared and analyzed all the linear regression models in Section 4 with relevant literature reviewed. I also analyzed whether the model results in this project could match the assumptions I made.

Section 2: Literature Review

Although one has previously focused on modeling radiation-mediated processes such as photosynthesis and transpiration, Bristow et al. [2] were interested in the relationship between solar irradiance and the range in daily temperature extremes and developed a simple equation (1) to explain it.

In their experiment, the range in daily temperature extremes (T) was calculated as

$$\Delta T(J) = T_{max}(J) - (T_{min}(J) + T_{min}(J+1))/2 \qquad (2)$$

where T_{max} is the daily maximum temperature (°C), T_{min} is the daily minimum temperature (°C), and J is the day number. When warm air mass appears on day J could increase the $T_{max}(J)$ beyond the possible value of the radiation alone. So only using the extremes of temperature may not be adequate to develop an accurate model. So the range in daily temperature extremes in equation (2) was used in equation (1).

The daily total atmospheric transmittance T_t in equation (1) was computed as the fraction of daily measured irradiance and daily extraterrestrial insolation. With the data at Pullman, they developed the equation (1) and then applied it to data from Great Falls, Montana Seattle/Tacoma, Washington. Tests on these three data sets indicate that “70-90% of the daily solar radiation variation can be explained by this simple model”[2].

Thornton et al. [3] proposed an algorithm for estimating daily incident solar radiation with an improvement of Bristow’s model, using data from the Solar and Meteorological Surface Observation Network (SAMSON) database. Bristow’s model considered only temperature, while Thornton’s model considered temperature, humidity, and precipitation. They pursued a parametric solution that yields accurate results at sites with different climates without the need to re-match model parameters one by one. The model performs well in boreal climates, but because the SAMSON database lacks sites in tropical or boreal climates, their model has the largest mean absolute error at the only tropical site, Miami.

Bender et al. [4] aimed to evaluate the performance of temperature-based models for estimating global solar radiation to fill the gaps in historical weather series (1980–2009) in Brazil. The main difficulty to realize the prediction is that Brazil has a low density of weather stations. In this study, “seven global climate models from CMIP5 under intermediate (RCP4.5) and high (RCP8.5) emission scenarios, were performed”(Bender, 2018). And the models showed warmer conditions for all scenarios over Brazil.

Section 3: Data Description and Visualization

I downloaded the historical climate data for maximum temperature, minimum temperature, and solar radiation from http://worldclim.org/. The historical climate data includes the aggregated monthly data from 1970 to 2000. The subset of each category, such as maximum temperature, minimum temperature, consists of 12 GeoTiff (.tif) files, one for each month of the year (January is 1; December is 12).

After importing each GeoTiff file and converting it to a matrix, there are three columns and 808,053 entries in total. The first two columns are x and y, which are longitude and latitude, respectively. And the third column shows the data of maximum temperature, minimum temperature, and solar radiation, respectively.

The original data was worldwide, and I cropped the data to be able to reduce calculation costs and focus more on the characteristics of a particular region. I focused on areas with latitude between 0°N to 70°N and longitude between 150°W and 40°W. As a result, the sample size for each month was reduced to 277,200 entries.

To analyze the time trends of the monthly average maximum temperature and the monthly average solar radiation, I plotted two line graphs with their average value as y and the month as x. Comparing these two line charts (Fig 1, Fig 2 in Appendix), they show a similar trend, with growth starting in January, peaking around June or July, and then beginning to decline. As shown in Fig 1 in Appendix, the average solar radiation peaked in June, while the average maximum temperature peaked in July, as shown in Fig 2 in Appendix.

The two variables move in the same direction over time and the turning points are close to each other, so I consider that the two variables have a positive correlation. Then because they reach their peak at different times, the maximum temperature reaches its peak one month later than the solar radiation reaches its peak. I think this is because there is a time delay in the causality of the temperature rise due to the accumulation of solar radiation.

Section 4: Model

In Section 3, I mainly performed a time-period analysis. In Section 4, I discussed the relationships between the extremes and ranges of temperature and solar radiation. So I merged the data as needed, for example, I combined the solar radiation matrix for January and the maximum temperature matrix for January into a new matrix with 4 columns. For rows with the same latitude and longitude in both matrices, they are merged into the same row.

Then I combined all 12 months of data, but I did not add a column for the month because the purpose was not to analyze time but to increase the sample size. For data sets containing minimum temperature and solar radiation, maximum temperature and solar radiation, and temperature range and solar radiation, the sample size was 1,325,220 for all. Since the ranges of all these variables differed significantly from each other, I then performed normalization.

I ran linear regression models as shown in Table 1 in R. The variable before the "~" sign is the response and variables behind the "~" sign are predictors. As shown in Table 1, the p-values for each predictor in all models are smaller than 0.05 so the null hypothesis is rejected.

For model 3 and model 7 with the highest R squared, and the residual standard errors are the lowest. The equations can be written as,

tmax = (1.716e-15) + (-1.119e-01)*x + (-5.291e-01)*y + (5.657e-01)*sr (3)

tmin = (6.662e-15) + (-2.663e-02)*x + (-5.141e-01)*y + (5.218e-01)*sr (4)

Inspired by the Thornton's model, I then further added the precipitation data to train linear regression model to predict solar radiation (model 13 in Table 1).

Section 5: Empirical Analysis

From Table 1, we can see that the model with the highest R squared value, i.e., 0.8123, is the one whose response is maximum temperature and predictors are longitude, latitude, and solar radiation. The model with the second-highest R squared value, i.e., 0.7485, is the one whose response is minimum temperature and predictors are longitude, latitude, and solar radiation. When not considering longitude and latitude factors, as in models 4 and 8, the R squared values are 0.6024 and 0.5368, respectively. The estimated coefficient of the variable solar radiation was always positive when the temperature was the response and the solar radiation was the predictor (See model 3,4,7,8, in Table 1). This can indicate that solar radiation has a boosting effect on temperature elevation.

When I tried to predict solar radiation using the maximum or minimum temperature, the R squared value was maintained in the range of 0.53 to 0.66. From the summary of the model 1, 2, 5, and 6 in the Appendix, although there is no causal relationship between increased temperature leading to enhanced solar radiation, they are also positively correlated.

The model with solar radiation as response and temperature range as predictor or vice versa has the lowest R-squared value. As in models 10 and 12, the R squared values are still relatively low even with the addition of latitude and longitude. However, in the equation developed by Bristow, a daily temperature range was used instead of extremes of the temperature, in order to predict the atmospheric transmittance, which is closely related to solar radiation. This may be due to the different temporal dimensions of the two data set; Bristow's experiment used daily data, while the WorldClim 2 data set aggregates monthly maximum and minimum temperatures from 1970-2000. Secondly, the formulas are also different. The exponential was not involved in this experiment, and there is no way to know the empirical coefficients for this data set, such as the empirical coefficients A, B, and C in equation (1).

Inspired by Thornton's improved algorithm, I took the precipitation into account to predict solar radiation. The R squared results showed that the model that also considered precipitation (model 13 in Table 1) was slightly better than the one that did not (model 1, 5 in Table 1).

Conclusion

In this project, I used the WorldClim database to test three theories.

The first theory is common knowledge that the concentration of solar radiation causes the temperature to rise. From the comparison of the time when solar radiation peaked and the time when the maximum temperature peaked, and the results of models 3, 4, 7, and 8, I can conclude that solar radiation contributes to temperature rise. In particular, model 3 with solar radiation as the response and latitude, longitude, and maximum temperature as the predictors has the highest R squared value.

The second theory I tested is Bristow's model. The equation Bristow developed showed that solar radiation can be determined by a range of temperature extremes. Although Bristow's formula involved exponential and mine involved only linear regression models, the temperature range has less effect on the solar radiation than the temperature extremes in my experiments. In a comparison of all the linear regression models I developed, the models that related the temperature range to the solar radiation had the worst performance.

Thirdly, although I didn't have access to humidity data, I trained a linear regression model to predict solar radiation using temperature extremes and precipitation, inspired by Thornton's model. When comparing models predicting solar radiation (model 1, 2, 5, 6, 9, 10, and 13), the R squared results show that the model including precipitation is slightly better than models with only temperature extremes.

References

[1] Fick, S. E., & Hijmans, R. J. (2017). WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas. International journal of climatology, 37(12), 4302-4315.

[2] Bristow, Keith L., and Gaylon S. Campbell. "On the relationship between incoming solar radiation and daily maximum and minimum temperature." Agricultural and forest meteorology 31.2 (1984): 159-166.

[3] Thornton, Peter E., and Steven W. Running. "An improved algorithm for estimating incident daily solar radiation from measurements of temperature, humidity, and precipitation." Agricultural and Forest Meteorology 93.4 (1999): 211-228.

[4] Bender, F. D., & Sentelhas, P. C. (2018). Solar radiation models and gridded databases to fill gaps in weather series and to project climate change in Brazil. Advances in Meteorology, 2018.
[5] Bastin, J. F., Finegold, Y., Garcia, C., Mollicone, D., Rezende, M., Routh, D., ... & Crowther, T. W. (2019). The global tree restoration potential. Science, 365(6448), 76-79.
[6] Intergovernmental Panel on Climate Change (IPCC), An IPCC Special Report on the Impacts of Global Warming of 1.5 °C Above Pre-Industrial Levels and Related Global Greenhouse Gas Emission Pathways (IPCC, 2018).
[7] Flato, G. M. (2011). Earth system models: an overview. Wiley Interdisciplinary Reviews: Climate Change, 2(6), 783-800.
[8] Kleidon, A., & Renner, M. (2013). A simple explanation for the sensitivity of the hydrologic cycle to surface temperature and solar radiation and its implications for global climate change. Earth System Dynamics, 4(2), 455-465.

Appendix

Table 1. R squared and p-value for all models, where “sr” refers to solar radiation, “x” refers to longitude, “y” refers to latitude, “tmin” refers to minimum temperature, “tmax” refers to maximum temperature, “trange” refers to the absolute difference between maximum and minimum temperature, and “prec” refers to precipitation. The variable before the “~” is the response and variables behind “~” are predictors.

Model Number	Model	R^2	p-value	Residual standard error
1	sr ~ x, y, tmax	0.6573	all are <2e-16	0.5854
2	sr ~ tmax	0.6024	<2e-16	0.6305
3	tmax ~ x, y, sr	0.8123	all are <2e-16	0.4332
4	tmax ~ sr	0.6024	<2e-16	0.6305
5	sr ~ x, y, tmin	0.5642	all are <2e-16	0.6601
6	sr ~ tmin	0.5368	<2e-16	0.6806
7	tmin ~ x, r, sr	0.7485	all are <2e-16	0.5015
8	tmin ~ sr	0.5368	<2e-16	0.6806
9	sr ~ trange	0.223	<2e-16	0.8815
10	sr ~ x,y, trange	0.3203	all are <2e-16	0.8244
11	trange ~ sr	0.223	<2e-16	0.8815
12	trange ~ x, y, sr	0.3884	all are <2e-16	0.7821
13	sr ~ x, y, tmax, tmin, prec	0.6878	all are <2e-16	0.5588

Fig 1. Average Solar Radiation from January to December

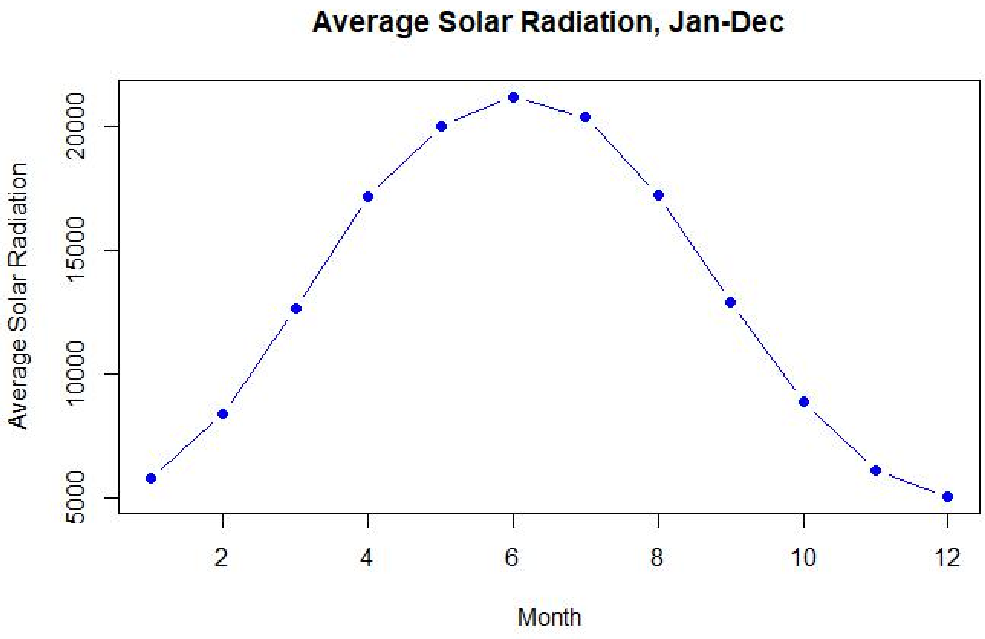
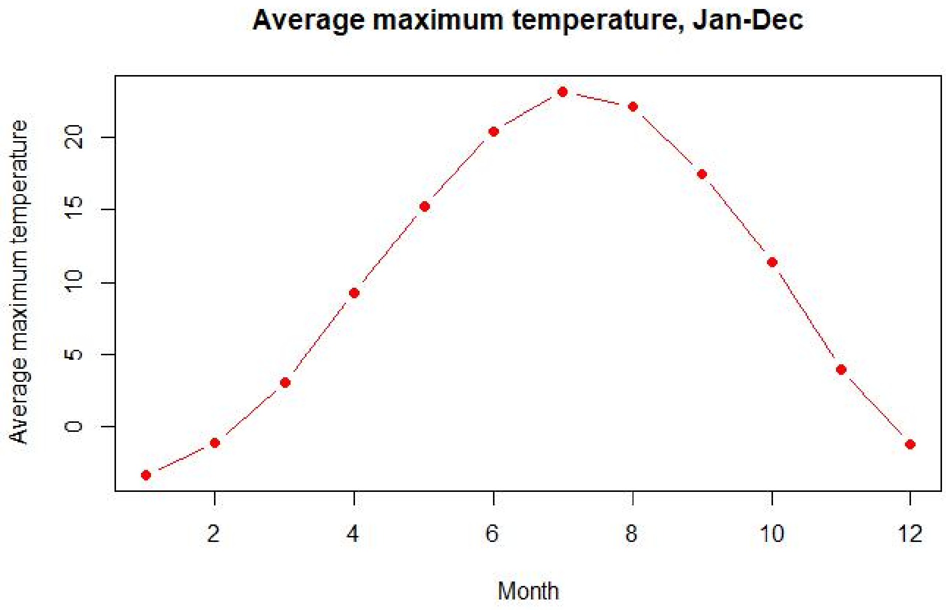


Fig 2. Average Solar Radiation from January to December



An Exploration of temperature and solar radiation using WorldClim 2 database

Yingxin Lin

Importing Data and Preprocessing

Importing Library and clear the environment

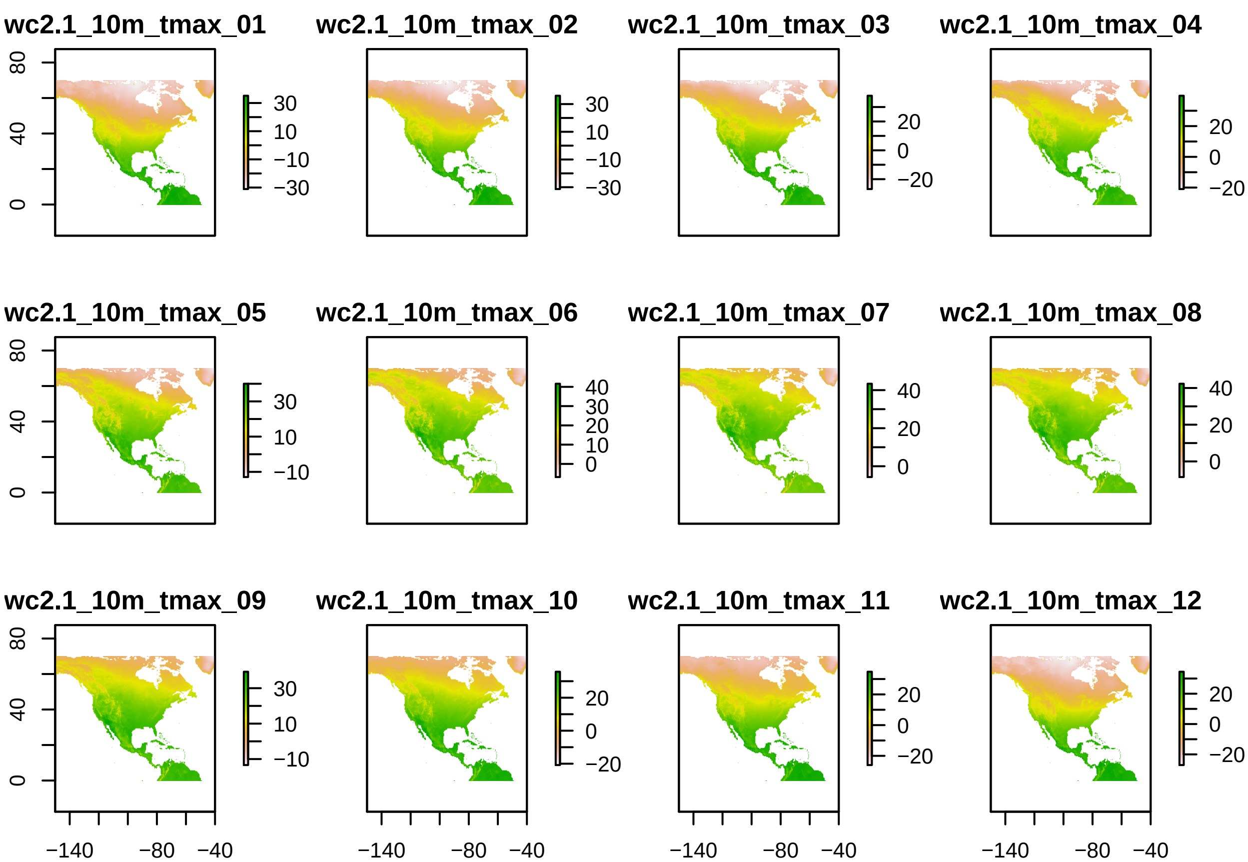
```r
# List of all packages
load.lib<-c("raster", "rgdal","caret","sp","hrbrthemes")

install.lib<-load.lib[!load.lib %in% installed.packages()]
for(lib in install.lib) install.packages(lib,dependencies=TRUE)
sapply(load.lib,require,character=TRUE)
```

```
## Loading required package: raster

## Loading required package: sp

## Loading required package: rgdal

## rgdal: version: 1.5-12, (SVN revision 1018)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 3.0.4, released 2020/01/28
## Path to GDAL shared files: C:/Program Files/Microsoft/R Open/R-4.0.2/library/rgdal/gdal
## GDAL binary built with GEOS: TRUE
## Loaded PROJ runtime: Rel. 6.3.1, February 10th, 2020, [PJ_VERSION: 631]
## Path to PROJ shared files: C:/Program Files/Microsoft/R Open/R-4.0.2/library/rgdal/proj
## Linking to sp version:1.4-2
## To mute warnings of possible GDAL/OSR exportToProj4() degradation,
## use options("rgdal_show_exportToProj4_warnings"="none") before loading rgdal.

## Loading required package: caret

## Loading required package: lattice

## Loading required package: ggplot2

## Loading required package: hrbrthemes

## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.

##       Please use hrbrthemes::import_roboto_condensed() to install Roboto Condensed and

##       if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow

##     raster      rgdal      caret         sp hrbrthemes
##       TRUE       TRUE       TRUE       TRUE       TRUE
```

```r
# Clear the environment
#rm(list = ls(all.names = TRUE)) # will clear all objects, including hidden objects
#gc() # free up memory and report memory usage
```

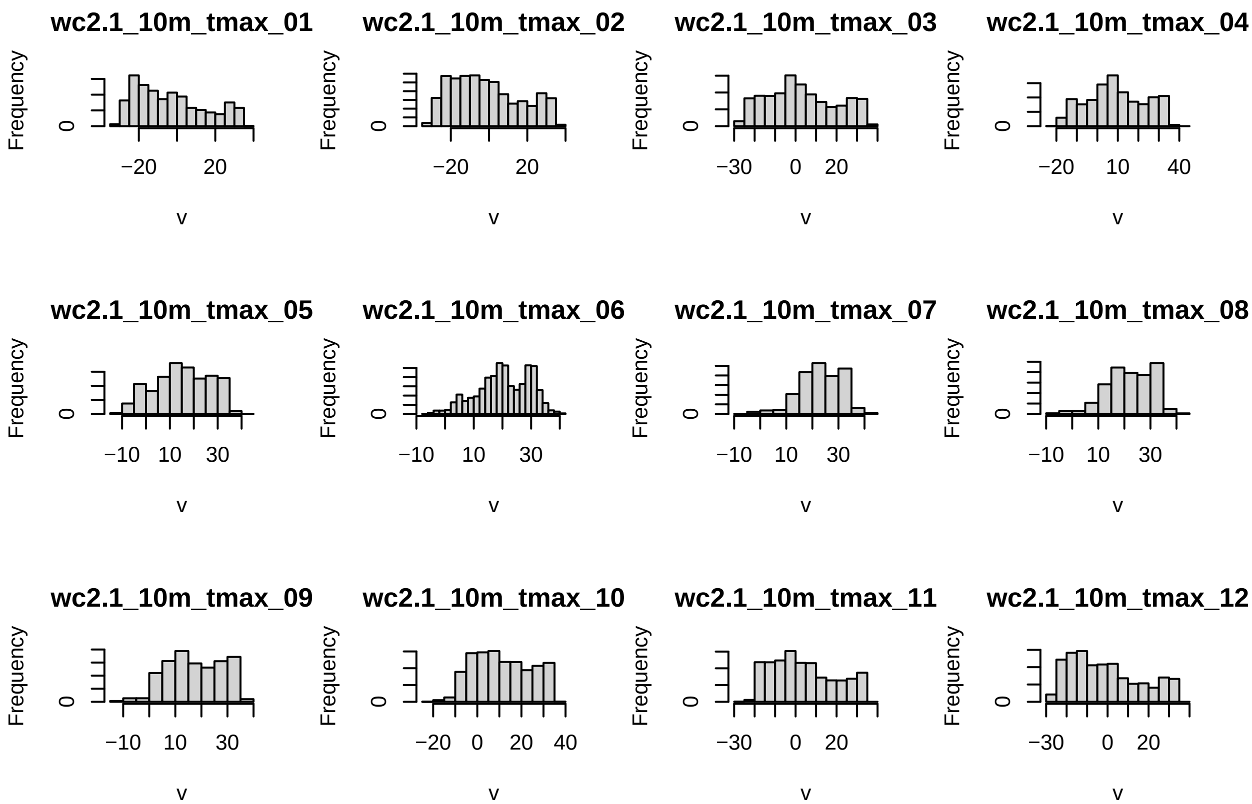
Importing the tmax(maximum temperature) (1970-2000)

```
###############################################################
### Importing the tmax(maximum temperature) (1970-2000)
###############################################################
tmax_1<- raster("maximum_temperature/wc2.1_10m_tmax_01.tif")
tmax_2<- raster("maximum_temperature/wc2.1_10m_tmax_02.tif")
tmax_3<- raster("maximum_temperature/wc2.1_10m_tmax_03.tif")
tmax_4<- raster("maximum_temperature/wc2.1_10m_tmax_04.tif")
tmax_5<- raster("maximum_temperature/wc2.1_10m_tmax_05.tif")
tmax_6<- raster("maximum_temperature/wc2.1_10m_tmax_06.tif")
tmax_7<- raster("maximum_temperature/wc2.1_10m_tmax_07.tif")
tmax_8<- raster("maximum_temperature/wc2.1_10m_tmax_08.tif")
tmax_9<- raster("maximum_temperature/wc2.1_10m_tmax_09.tif")
tmax_10<- raster("maximum_temperature/wc2.1_10m_tmax_10.tif")
tmax_11<- raster("maximum_temperature/wc2.1_10m_tmax_11.tif")
tmax_12<- raster("maximum_temperature/wc2.1_10m_tmax_12.tif")

tmax_stack <- stack(tmax_1,tmax_2,tmax_3,tmax_4,tmax_5,
                    tmax_6,tmax_7,tmax_8,tmax_9,tmax_10,
                    tmax_11,tmax_12)
Crop <- c(-150,-40, 0,70)
tmax_crop <- crop(tmax_stack, Crop)
# Plot cropped area
plot(tmax_crop)
```



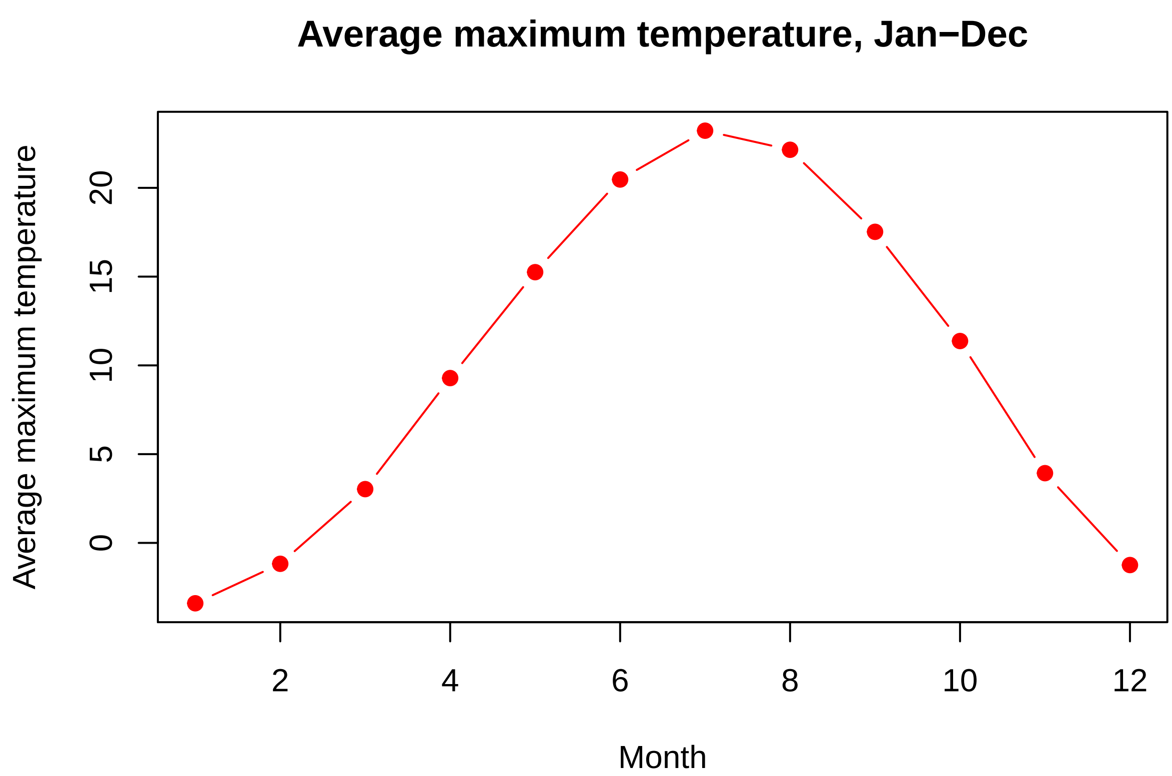
Visualization

```
hist(tmax_crop)
```




```
############################################################
### Convert the RasterLayer to Matrix for tmax
############################################################
tmax_points_1 <- rasterToPoints(tmax_1)
tmax_points_2 <- rasterToPoints(tmax_2)
tmax_points_3 <- rasterToPoints(tmax_3)
tmax_points_4 <- rasterToPoints(tmax_4)
tmax_points_5 <- rasterToPoints(tmax_5)
tmax_points_6 <- rasterToPoints(tmax_6)
tmax_points_7 <- rasterToPoints(tmax_7)
tmax_points_8 <- rasterToPoints(tmax_8)
tmax_points_9 <- rasterToPoints(tmax_9)
tmax_points_10 <- rasterToPoints(tmax_10)
tmax_points_11 <- rasterToPoints(tmax_11)
tmax_points_12 <- rasterToPoints(tmax_12)

# Stack the 12 month data
tmax_stack_points <- rasterToPoints(tmax_stack)
# Convert the matrix data to dataframe format
tmax_stack_df <- as.data.frame(tmax_stack_points)
# Cropping data within the area
new_subset <- subset(tmax_stack_df, tmax_stack_df$x >=-150 & tmax_stack_df$x <=-40
                     & tmax_stack_df$y >= 0 & tmax_stack_df$y <=70)
```

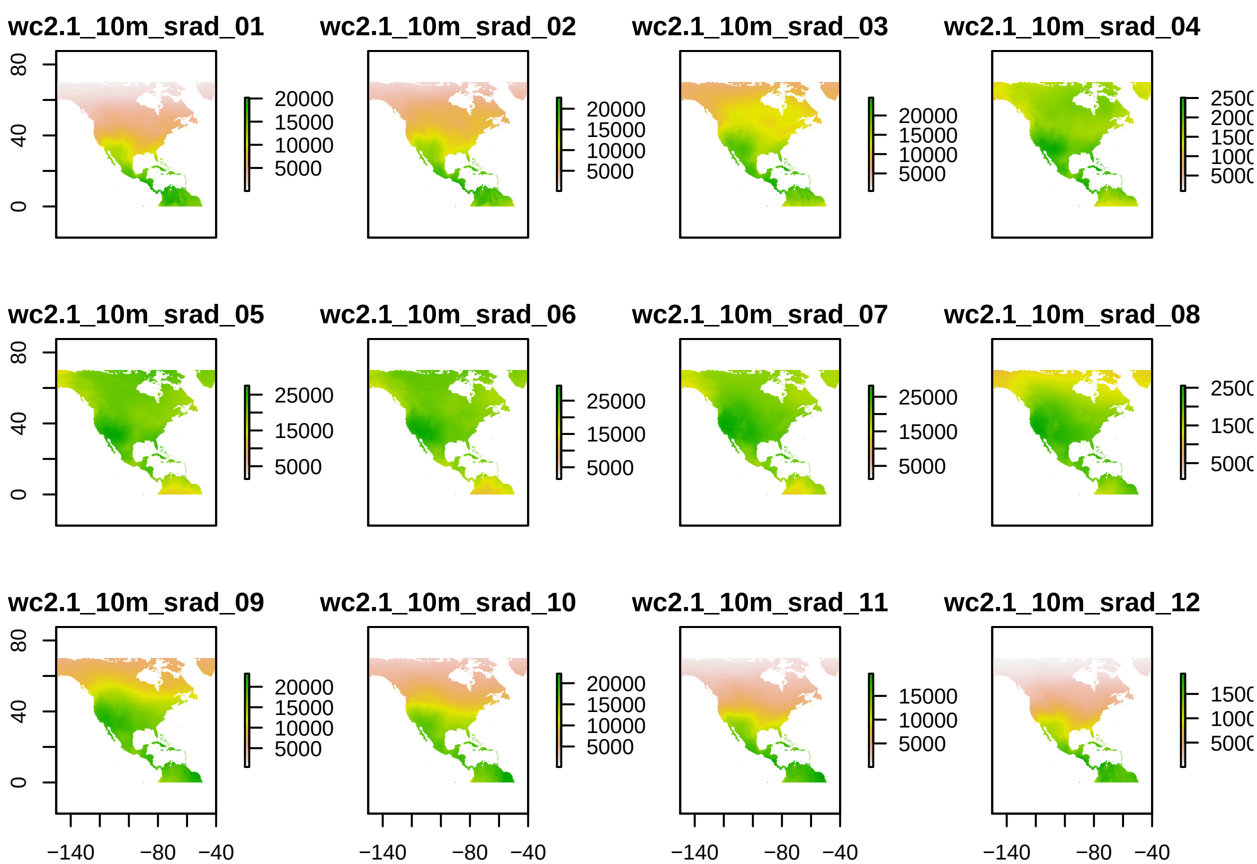
```r
# Prepare the Average tmax data
months <- 1:12
mean_tmax<-c(mean(new_subset$wc2.1_10m_tmax_01),mean(new_subset$wc2.1_10m_tmax_02),
             mean(new_subset$wc2.1_10m_tmax_03),mean(new_subset$wc2.1_10m_tmax_04),
             mean(new_subset$wc2.1_10m_tmax_05),mean(new_subset$wc2.1_10m_tmax_06),
             mean(new_subset$wc2.1_10m_tmax_07),mean(new_subset$wc2.1_10m_tmax_08),
             mean(new_subset$wc2.1_10m_tmax_09),mean(new_subset$wc2.1_10m_tmax_10),
             mean(new_subset$wc2.1_10m_tmax_11),mean(new_subset$wc2.1_10m_tmax_12)
)
################### Plotting a line chart #################################
# Plot the average maximum temperature from Jan to Dec in 1970-2000
plot(months,mean_tmax,type = "b", pch = 19,
     col = "red", xlab = "Month", ylab = "Average maximum temperature",
     main='Average maximum temperature, Jan-Dec')
```



Importing the solar radiation (1970-2000)

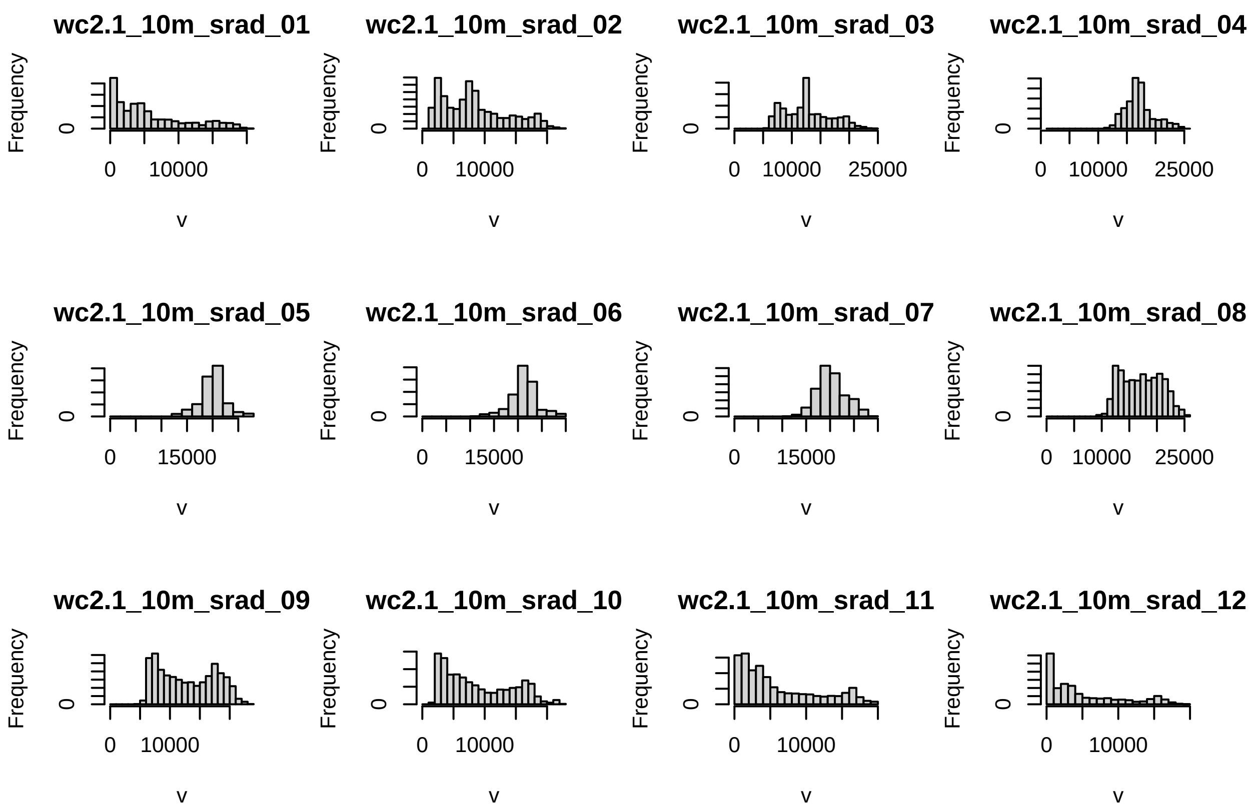
```r
############################################################################
##################### Importing the solar radiation (1970-2000)
############################################################################
sr_1<- raster("solar_radiation/wc2.1_10m_srad_01.tif")
sr_2<- raster("solar_radiation/wc2.1_10m_srad_02.tif")
sr_3<- raster("solar_radiation/wc2.1_10m_srad_03.tif")
sr_4<- raster("solar_radiation/wc2.1_10m_srad_04.tif")
sr_5<- raster("solar_radiation/wc2.1_10m_srad_05.tif")
sr_6<- raster("solar_radiation/wc2.1_10m_srad_06.tif")
sr_7<- raster("solar_radiation/wc2.1_10m_srad_07.tif")
```

```r
sr_8<- raster("solar_radiation/wc2.1_10m_srad_08.tif")
sr_9<- raster("solar_radiation/wc2.1_10m_srad_09.tif")
sr_10<- raster("solar_radiation/wc2.1_10m_srad_10.tif")
sr_11<- raster("solar_radiation/wc2.1_10m_srad_11.tif")
sr_12<- raster("solar_radiation/wc2.1_10m_srad_12.tif")

sr_list <- c(sr_1,sr_2,sr_3,sr_4,sr_5,sr_6,sr_7,sr_8,sr_9,sr_10,sr_11,sr_12)

sr_stack <- stack(sr_list)
sr_crop <- crop(sr_stack, Crop)
plot(sr_crop)
```



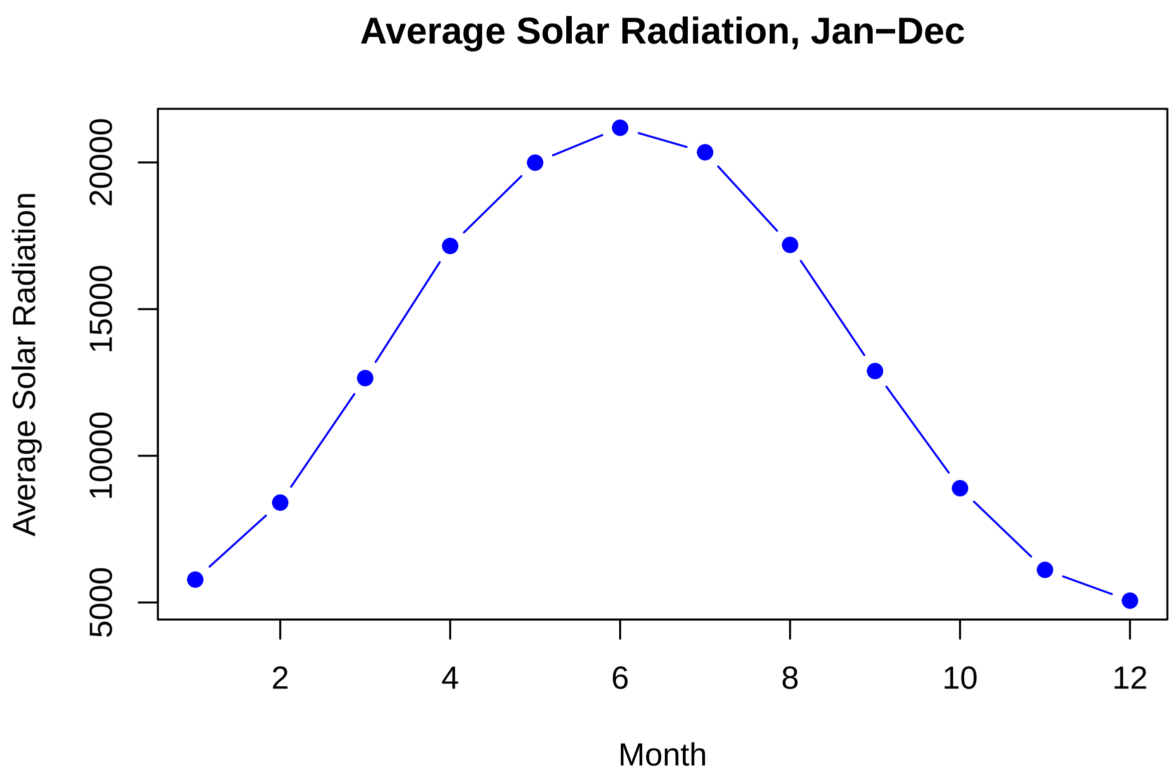
Visualization

```r
hist(sr_crop)
```




```
##############################################################
### Convert the RasterLayer to Matrix for solar radiation
##############################################################
sr_points_1 <- rasterToPoints(sr_1)
sr_points_2 <- rasterToPoints(sr_2)
sr_points_3 <- rasterToPoints(sr_3)
sr_points_4 <- rasterToPoints(sr_4)
sr_points_5 <- rasterToPoints(sr_5)
sr_points_6 <- rasterToPoints(sr_6)
sr_points_7 <- rasterToPoints(sr_7)
sr_points_8 <- rasterToPoints(sr_8)
sr_points_9 <- rasterToPoints(sr_9)
sr_points_10 <- rasterToPoints(sr_10)
sr_points_11 <- rasterToPoints(sr_11)
sr_points_12 <- rasterToPoints(sr_12)

# Stack the 12 months data
sr_stack <- stack(sr_1,sr_2,sr_3,sr_4,sr_5,
                              sr_6,sr_7,sr_8,sr_9,sr_10,
                              sr_11,sr_12)

sr_stack_points <- rasterToPoints(sr_stack)
sr_stack_df <- as.data.frame(sr_stack_points)
# Crop the data within the area
new_subset <- subset(sr_stack_df, sr_stack_df$x >=-150 & sr_stack_df$x <=-40
                     & sr_stack_df$y >= 0 & sr_stack_df$y <=70)
```

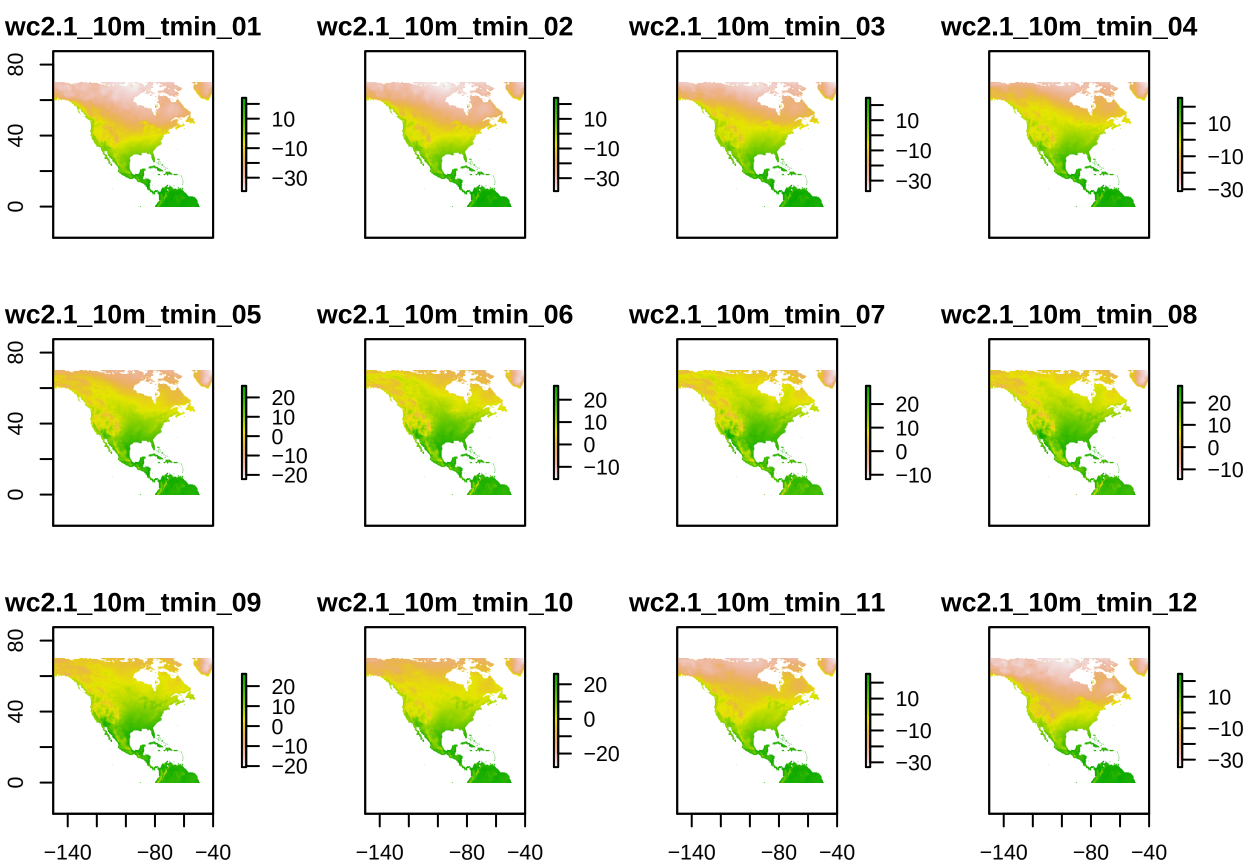
```
# Prepare the average solar radiation data
months <- 1:12
mean_sr<-c(mean(new_subset$wc2.1_10m_srad_01),mean(new_subset$wc2.1_10m_srad_02),
             mean(new_subset$wc2.1_10m_srad_03),mean(new_subset$wc2.1_10m_srad_04),
             mean(new_subset$wc2.1_10m_srad_05),mean(new_subset$wc2.1_10m_srad_06),
             mean(new_subset$wc2.1_10m_srad_07),mean(new_subset$wc2.1_10m_srad_08),
             mean(new_subset$wc2.1_10m_srad_09),mean(new_subset$wc2.1_10m_srad_10),
             mean(new_subset$wc2.1_10m_srad_11),mean(new_subset$wc2.1_10m_srad_12)
)
################### Plotting a line chart #################################
# Plot the average solar radiation from Jan to Dec in 1970-2000
plot(months,mean_sr,type = "b", pch = 19,
     col = "blue", xlab = "Month", ylab = "Average Solar Radiation",
     main='Average Solar Radiation, Jan-Dec')
```



Importing the tmin(minimum temperature) (1970-2000)

```
############################################################
# Importing the tmin(minimum temperature) (1970-2000)
###########################################################
tmin_1<- raster("minimum_temperature/wc2.1_10m_tmin_01.tif")
tmin_2<- raster("minimum_temperature/wc2.1_10m_tmin_02.tif")
tmin_3<- raster("minimum_temperature/wc2.1_10m_tmin_03.tif")
tmin_4<- raster("minimum_temperature/wc2.1_10m_tmin_04.tif")
tmin_5<- raster("minimum_temperature/wc2.1_10m_tmin_05.tif")
tmin_6<- raster("minimum_temperature/wc2.1_10m_tmin_06.tif")
```

```
tmin_7<- raster("minimum_temperature/wc2.1_10m_tmin_07.tif")
tmin_8<- raster("minimum_temperature/wc2.1_10m_tmin_08.tif")
tmin_9<- raster("minimum_temperature/wc2.1_10m_tmin_09.tif")
tmin_10<- raster("minimum_temperature/wc2.1_10m_tmin_10.tif")
tmin_11<- raster("minimum_temperature/wc2.1_10m_tmin_11.tif")
tmin_12<- raster("minimum_temperature/wc2.1_10m_tmin_12.tif")

# Stack the data from Jan to Dec
tmin_stack <- stack(tmin_1,tmin_2,tmin_3,tmin_4,tmin_5,
                    tmin_6,tmin_7,tmin_8,tmin_9,tmin_10,
                    tmin_11,tmin_12)
Crop <- c(-150,-40, 0,70)
tmin_crop <- crop(tmin_stack, Crop)
plot(tmin_crop)
```

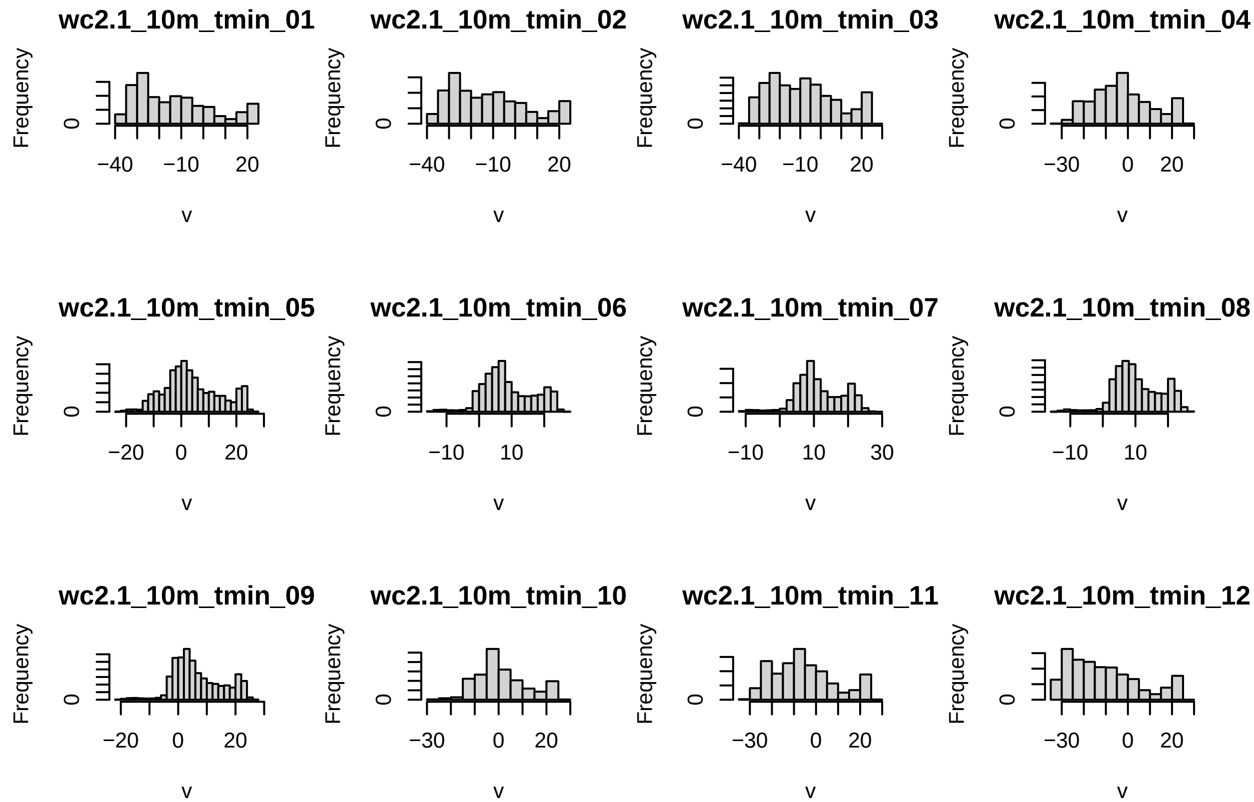


Visualization

```
hist(tmin_crop)
```

```
## Warning in .hist1(raster(x, y[i]), maxpixels = maxpixels, main = main[y[i]], :
## 36% of the raster cells were used. 100000 values used.

## Warning in .hist1(raster(x, y[i]), maxpixels = maxpixels, main = main[y[i]], :
## 36% of the raster cells were used. 100000 values used.

## Warning in .hist1(raster(x, y[i]), maxpixels = maxpixels, main = main[y[i]], :
## 36% of the raster cells were used. 100000 values used.
```

```
## Warning in .hist1(raster(x, y[i]), maxpixels = maxpixels, main = main[y[i]], :
## 36% of the raster cells were used. 100000 values used.

## Warning in .hist1(raster(x, y[i]), maxpixels = maxpixels, main = main[y[i]], :
## 36% of the raster cells were used. 100000 values used.

## Warning in .hist1(raster(x, y[i]), maxpixels = maxpixels, main = main[y[i]], :
## 36% of the raster cells were used. 100000 values used.

## Warning in .hist1(raster(x, y[i]), maxpixels = maxpixels, main = main[y[i]], :
## 36% of the raster cells were used. 100000 values used.

## Warning in .hist1(raster(x, y[i]), maxpixels = maxpixels, main = main[y[i]], :
## 36% of the raster cells were used. 100000 values used.

## Warning in .hist1(raster(x, y[i]), maxpixels = maxpixels, main = main[y[i]], :
## 36% of the raster cells were used. 100000 values used.

## Warning in .hist1(raster(x, y[i]), maxpixels = maxpixels, main = main[y[i]], :
## 36% of the raster cells were used. 100000 values used.

## Warning in .hist1(raster(x, y[i]), maxpixels = maxpixels, main = main[y[i]], :
## 36% of the raster cells were used. 100000 values used.

## Warning in .hist1(raster(x, y[i]), maxpixels = maxpixels, main = main[y[i]], :
## 36% of the raster cells were used. 100000 values used.
```




```r
##############################################################
# Convert the RasterLayer to Matrix
#############################################################
tmin_points_1 <- rasterToPoints(tmin_1)
tmin_points_2 <- rasterToPoints(tmin_2)
tmin_points_3 <- rasterToPoints(tmin_3)
tmin_points_4 <- rasterToPoints(tmin_4)
tmin_points_5 <- rasterToPoints(tmin_5)
tmin_points_6 <- rasterToPoints(tmin_6)
tmin_points_7 <- rasterToPoints(tmin_7)
tmin_points_8 <- rasterToPoints(tmin_8)
tmin_points_9 <- rasterToPoints(tmin_9)
tmin_points_10 <- rasterToPoints(tmin_10)
tmin_points_11 <- rasterToPoints(tmin_11)
tmin_points_12 <- rasterToPoints(tmin_12)
```

Prepare Training Data

Prepare training data for tmax and solar radiation

```r
##############################################################
####### Prepare training data for tmax and solar radiation
##############################################################
tmax_sr_1 <- cbind(tmax_points_1[,1:2],tmax=tmax_points_1[,3],sr=sr_points_1[,3])
tmax_sr_2 <- cbind(tmax_points_2[,1:2],tmax=tmax_points_2[,3],sr=sr_points_2[,3])
```

```r
tmax_sr_3 <- cbind(tmax_points_3[,1:2],tmax=tmax_points_3[,3],sr=sr_points_3[,3])
tmax_sr_4 <- cbind(tmax_points_4[,1:2],tmax=tmax_points_4[,3],sr=sr_points_4[,3])
tmax_sr_5 <- cbind(tmax_points_5[,1:2],tmax=tmax_points_5[,3],sr=sr_points_5[,3])
tmax_sr_6 <- cbind(tmax_points_6[,1:2],tmax=tmax_points_6[,3],sr=sr_points_6[,3])
tmax_sr_7 <- cbind(tmax_points_7[,1:2],tmax=tmax_points_7[,3],sr=sr_points_7[,3])
tmax_sr_8 <- cbind(tmax_points_8[,1:2],tmax=tmax_points_8[,3],sr=sr_points_8[,3])
tmax_sr_9 <- cbind(tmax_points_9[,1:2],tmax=tmax_points_9[,3],sr=sr_points_9[,3])
tmax_sr_10 <- cbind(tmax_points_10[,1:2],tmax=tmax_points_10[,3],sr=sr_points_10[,3])
tmax_sr_11 <- cbind(tmax_points_11[,1:2],tmax=tmax_points_11[,3],sr=sr_points_11[,3])
tmax_sr_12 <- cbind(tmax_points_12[,1:2],tmax=tmax_points_12[,3],sr=sr_points_12[,3])

# Merge the data from Jan to Dec
total_tmax_sr <- rbind(tmax_sr_1, tmax_sr_2,tmax_sr_3,tmax_sr_4,tmax_sr_5,tmax_sr_6,
               tmax_sr_7,tmax_sr_8,tmax_sr_9,tmax_sr_10,tmax_sr_11,tmax_sr_12)
total_tmax_sr <- as.data.frame(total_tmax_sr)
# Crop the data within the area
total_tmax_sr <- subset(total_tmax_sr, total_tmax_sr$x >=-150 & total_tmax_sr$x <=-40
                     & total_tmax_sr$y >= 0 & total_tmax_sr$y <=70)

# Normalization
# Summarize before normalization
summary(total_tmax_sr)
```

```
##        x                  y                 tmax              sr
##  Min.   :-149.92   Min.   : 0.08333   Min.   :-31.340   Min.   :    1
##  1st Qu.:-112.75   1st Qu.:35.91667   1st Qu.: -2.202   1st Qu.: 6690
##  Median : -96.92   Median :49.58333   Median : 12.290   Median :14385
##  Mean   : -95.53   Mean   :45.82135   Mean   : 10.033   Mean   :12971
##  3rd Qu.: -76.75   3rd Qu.:60.91667   3rd Qu.: 24.022   3rd Qu.:19034
##  Max.   : -40.08   Max.   :69.91667   Max.   : 44.687   Max.   :29527
```

```r
preproc1 <- preProcess(total_tmax_sr, method=c("center", "scale"))
norm_tmax_sr <- predict(preproc1, total_tmax_sr)
# Summarize after normalization
summary(norm_tmax_sr)
```

```
##        x                   y                 tmax              sr
##  Min.   :-2.27918   Min.   :-2.4862   Min.   :-2.4931   Min.   :-1.8001
##  1st Qu.:-0.72165   1st Qu.:-0.5384   1st Qu.:-0.7373   1st Qu.:-0.8717
##  Median :-0.05812   Median : 0.2045   Median : 0.1359   Median : 0.1962
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.78700   3rd Qu.: 0.8205   3rd Qu.: 0.8429   3rd Qu.: 0.8415
##  Max.   : 2.32358   Max.   : 1.3098   Max.   : 2.0881   Max.   : 2.2978
```

Prepare training data for tmin and solar radiation

```r
#####################################################
# Prepare training data for tmin and solar radiation
#####################################################
tmin_sr_1 <- cbind(tmin_points_1[,1:2],tmin=tmin_points_1[,3],sr=sr_points_1[,3])
tmin_sr_2 <- cbind(tmin_points_2[,1:2],tmin=tmin_points_2[,3],sr=sr_points_2[,3])
tmin_sr_3 <- cbind(tmin_points_3[,1:2],tmin=tmin_points_3[,3],sr=sr_points_3[,3])
tmin_sr_4 <- cbind(tmin_points_4[,1:2],tmin=tmin_points_4[,3],sr=sr_points_4[,3])
tmin_sr_5 <- cbind(tmin_points_5[,1:2],tmin=tmin_points_5[,3],sr=sr_points_5[,3])
```

```r
tmin_sr_6 <- cbind(tmin_points_6[,1:2],tmin=tmin_points_6[,3],sr=sr_points_6[,3])
tmin_sr_7 <- cbind(tmin_points_7[,1:2],tmin=tmin_points_7[,3],sr=sr_points_7[,3])
tmin_sr_8 <- cbind(tmin_points_8[,1:2],tmin=tmin_points_8[,3],sr=sr_points_8[,3])
tmin_sr_9 <- cbind(tmin_points_9[,1:2],tmin=tmin_points_9[,3],sr=sr_points_9[,3])
tmin_sr_10 <- cbind(tmin_points_10[,1:2],tmin=tmin_points_10[,3],sr=sr_points_10[,3])
tmin_sr_11 <- cbind(tmin_points_11[,1:2],tmin=tmin_points_11[,3],sr=sr_points_11[,3])
tmin_sr_12 <- cbind(tmin_points_12[,1:2],tmin=tmin_points_12[,3],sr=sr_points_12[,3])

# Merge the data from Jan to Dec
total_tmin_sr <- rbind(tmin_sr_1, tmin_sr_2,tmin_sr_3,tmin_sr_4,tmin_sr_5,tmin_sr_6,
                       tmin_sr_7,tmin_sr_8,tmin_sr_9,tmin_sr_10,tmin_sr_11,tmin_sr_12)
total_tmin_sr <- as.data.frame(total_tmin_sr)
# Crop the data within the area
total_tmin_sr <- subset(total_tmin_sr, total_tmin_sr$x >=-150 & total_tmin_sr$x <=-40
                        & total_tmin_sr$y >= 0 & total_tmin_sr$y <=70)

# Normalization
summary(total_tmin_sr)
```

```
##        x                 y                 tmin                 sr
##  Min.   :-149.92   Min.   : 0.08333   Min.   :-39.3305   Min.   :    1
##  1st Qu.:-112.75   1st Qu.:35.91667   1st Qu.:-11.2216   1st Qu.: 6690
##  Median : -96.92   Median :49.58333   Median :  1.1467   Median :14385
##  Mean   : -95.53   Mean   :45.82135   Mean   : -0.8469   Mean   :12971
##  3rd Qu.: -76.75   3rd Qu.:60.91667   3rd Qu.:  9.9763   3rd Qu.:19034
##  Max.   : -40.08   Max.   :69.91667   Max.   : 28.6363   Max.   :29527
```

```r
preproc1 <- preProcess(total_tmin_sr, method=c("center", "scale"))
norm_tmin_sr <- predict(preproc1, total_tmin_sr)
summary(norm_tmin_sr)
```

```
##        x                  y                tmin               sr
##  Min.   :-2.27918   Min.   :-2.4862   Min.   :-2.5080   Min.   :-1.8001
##  1st Qu.:-0.72165   1st Qu.:-0.5384   1st Qu.:-0.6761   1st Qu.:-0.8717
##  Median :-0.05812   Median : 0.2045   Median : 0.1299   Median : 0.1962
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.78700   3rd Qu.: 0.8205   3rd Qu.: 0.7054   3rd Qu.: 0.8415
##  Max.   : 2.32358   Max.   : 1.3098   Max.   : 1.9214   Max.   : 2.2978
```

Merging the tmin, tmax, sr, trange

```r
################################################################
# Merging the tmin, tmax, sr, trange
################################################################
total_tmin_tmax_sr <- cbind(total_tmax_sr,tmin=total_tmin_sr[,3])
# Create new column, which is the absolute difference between tmin and tmax
total_tmin_tmax_sr$trange <- abs(total_tmin_tmax_sr[,3]-total_tmin_tmax_sr[,5])

# Normalization
summary(total_tmin_tmax_sr)
```

```
##        x                 y                 tmax               sr
##  Min.   :-149.92   Min.   : 0.08333   Min.   :-31.340   Min.   :    1
##  1st Qu.:-112.75   1st Qu.:35.91667   1st Qu.: -2.202   1st Qu.: 6690
```

```
##  Median : -96.92   Median :49.58333   Median : 12.290   Median :14385
##  Mean   : -95.53   Mean   :45.82135   Mean   : 10.033   Mean   :12971
##  3rd Qu.: -76.75   3rd Qu.:60.91667   3rd Qu.: 24.022   3rd Qu.:19034
##  Max.   : -40.08   Max.   :69.91667   Max.   : 44.687   Max.   :29527
##       tmin              trange
##  Min.   :-39.3305   Min.   : 1.000
##  1st Qu.:-11.2216   1st Qu.: 8.247
##  Median :  1.1467   Median :10.661
##  Mean   : -0.8469   Mean   :10.880
##  3rd Qu.:  9.9763   3rd Qu.:13.074
##  Max.   : 28.6363   Max.   :24.773
```

```
preproc1 <- preProcess(total_tmin_tmax_sr, method=c("center", "scale"))
norm_tmin_tmax_sr <- predict(preproc1, total_tmin_tmax_sr)
summary(norm_tmin_tmax_sr)
```

```
##        x                  y                tmax               sr
##  Min.   :-2.27918   Min.   :-2.4862   Min.   :-2.4931   Min.   :-1.8001
##  1st Qu.:-0.72165   1st Qu.:-0.5384   1st Qu.:-0.7373   1st Qu.:-0.8717
##  Median :-0.05812   Median : 0.2045   Median : 0.1359   Median : 0.1962
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.78700   3rd Qu.: 0.8205   3rd Qu.: 0.8429   3rd Qu.: 0.8415
##  Max.   : 2.32358   Max.   : 1.3098   Max.   : 2.0881   Max.   : 2.2978
##       tmin             trange
##  Min.   :-2.5080   Min.   :-2.84643
##  1st Qu.:-0.6761   1st Qu.:-0.75855
##  Median : 0.1299   Median :-0.06324
##  Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 0.7054   3rd Qu.: 0.63193
##  Max.   : 1.9214   Max.   : 4.00248
```

Linear Regression Model

```
# Create linear regression-1
# response: sr
# predictors: x,y,tmax
lm_sr_tmax_all = lm(sr~., data = norm_tmax_sr)
summary(lm_sr_tmax_all)
```

```
##
## Call:
## lm(formula = sr ~ ., data = norm_tmax_sr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.14609 -0.43773 -0.00651  0.38434  2.26375
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.800e-15  5.085e-04     0.0        1
## x            8.943e-02  5.489e-04   162.9   <2e-16 ***
## y            3.662e-01  7.953e-04   460.4   <2e-16 ***
```

```
## tmax         1.033e+00  7.567e-04  1365.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5854 on 1325216 degrees of freedom
## Multiple R-squared:  0.6573, Adjusted R-squared:  0.6573
## F-statistic: 8.471e+05 on 3 and 1325216 DF,  p-value: < 2.2e-16
```

```r
# Create linear regression-2
# response: sr
# predictors: tmax
lm_sr_tmax = lm(sr~tmax, data = norm_tmax_sr)
summary(lm_sr_tmax)
```

```
##
## Call:
## lm(formula = sr ~ tmax, data = norm_tmax_sr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6337 -0.5156 -0.0805  0.4408  2.1538
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.073e-15  5.477e-04       0        1
## tmax         7.762e-01  5.477e-04    1417   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6305 on 1325218 degrees of freedom
## Multiple R-squared:  0.6024, Adjusted R-squared:  0.6024
## F-statistic: 2.008e+06 on 1 and 1325218 DF,  p-value: < 2.2e-16
```

```r
# Create linear regression-3
# response: tmax
# predictors: x,y,sr
lm_tmax_all = lm(tmax~., data = norm_tmax_sr)
summary(lm_tmax_all)
```

```
##
## Call:
## lm(formula = tmax ~ ., data = norm_tmax_sr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.59051 -0.28957  0.00127  0.34072  1.65880
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.716e-15  3.763e-04     0.0        1
## x           -1.119e-01  3.985e-04  -280.8   <2e-16 ***
## y           -5.291e-01  4.365e-04 -1212.1   <2e-16 ***
## sr           5.657e-01  4.143e-04  1365.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.4332 on 1325216 degrees of freedom
## Multiple R-squared:  0.8123, Adjusted R-squared:  0.8123
## F-statistic: 1.912e+06 on 3 and 1325216 DF,  p-value: < 2.2e-16
```

```r
# Create linear regression-4
# response: tmax
# predictors: sr
lm_tmax_sr = lm(tmax~sr, data = norm_tmax_sr)
summary(lm_tmax_sr)
```

```
##
## Call:
## lm(formula = tmax ~ sr, data = norm_tmax_sr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0973 -0.4084  0.1038  0.4752  1.3685
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 9.390e-16  5.477e-04       0        1
## sr          7.762e-01  5.477e-04    1417   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6305 on 1325218 degrees of freedom
## Multiple R-squared:  0.6024, Adjusted R-squared:  0.6024
## F-statistic: 2.008e+06 on 1 and 1325218 DF,  p-value: < 2.2e-16
```

```r
# Create linear regression-5
# response: sr
# predictors: x,y,tmin
lm_sr_tmin_all = lm(sr~., data = norm_tmin_sr)
summary(lm_sr_tmin_all)
```

```
##
## Call:
## lm(formula = sr ~ ., data = norm_tmin_sr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.18151 -0.48842 -0.01294  0.46928  2.20672
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.020e-15  5.734e-04    0.00        1
## x           -9.227e-03  6.080e-04  -15.18   <2e-16 ***
## y            2.353e-01  8.638e-04  272.35   <2e-16 ***
## tmin         9.041e-01  8.310e-04 1087.93   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6601 on 1325216 degrees of freedom
## Multiple R-squared:  0.5642, Adjusted R-squared:  0.5642
## F-statistic: 5.719e+05 on 3 and 1325216 DF,  p-value: < 2.2e-16
```

```r
# Create linear regression-6
# response: sr
# predictors: tmin
lm_sr_tmin = lm(sr~tmin, data = norm_tmin_sr)
summary(lm_sr_tmin)
```

```
##
## Call:
## lm(formula = sr ~ tmin, data = norm_tmin_sr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.86750 -0.54822 -0.07385  0.50963  2.08267
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.673e-15  5.912e-04       0        1
## tmin         7.326e-01  5.912e-04    1239   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6806 on 1325218 degrees of freedom
## Multiple R-squared:  0.5368, Adjusted R-squared:  0.5368
## F-statistic: 1.536e+06 on 1 and 1325218 DF,  p-value: < 2.2e-16
```

```r
# Create linear regression-7
# response: tmin
# predictors: x,y,sr
lm_tmin_all = lm(tmin~., data = norm_tmin_sr)
summary(lm_tmin_all)
```

```
##
## Call:
## lm(formula = tmin ~ ., data = norm_tmin_sr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.51429 -0.35099  0.01998  0.36349  1.80524
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  6.622e-15  4.357e-04     0.00        1
## x           -2.663e-02  4.614e-04   -57.71   <2e-16 ***
## y           -5.141e-01  5.053e-04 -1017.27   <2e-16 ***
## sr           5.218e-01  4.797e-04  1087.93   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5015 on 1325216 degrees of freedom
## Multiple R-squared:  0.7485, Adjusted R-squared:  0.7485
## F-statistic: 1.314e+06 on 3 and 1325216 DF,  p-value: < 2.2e-16
```

```r
# Create linear regression-8
# response: tmin
# predictors: sr
```

```r
lm_tmin_sr = lm(tmin~sr, data = norm_tmin_sr)
summary(lm_tmin_sr)
```

```
##
## Call:
## lm(formula = tmin ~ sr, data = norm_tmin_sr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.06222 -0.48515  0.04881  0.52496  1.77747
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.832e-15  5.912e-04       0        1
## sr          7.326e-01  5.912e-04    1239   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6806 on 1325218 degrees of freedom
## Multiple R-squared:  0.5368, Adjusted R-squared:  0.5368
## F-statistic: 1.536e+06 on 1 and 1325218 DF,  p-value: < 2.2e-16
```

```r
# Create linear regression-9
# response: sr
# predictors: trange
lm_sr_trange = lm(sr~trange, data = norm_tmin_tmax_sr)
summary(lm_sr_trange)
```

```
##
## Call:
## lm(formula = sr ~ trange, data = norm_tmin_tmax_sr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.92193 -0.79497  0.06516  0.72917  2.79161
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -9.254e-16  7.657e-04     0.0        1
## trange       4.722e-01  7.657e-04   616.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8815 on 1325218 degrees of freedom
## Multiple R-squared:  0.223,  Adjusted R-squared:  0.223
## F-statistic: 3.803e+05 on 1 and 1325218 DF,  p-value: < 2.2e-16
```

```r
# Create linear regression-10
# response: sr
# predictors: x,y,trange
lm_sr_trange_all = lm(sr~x+y+trange, data = norm_tmin_tmax_sr)
summary(lm_sr_trange_all)
```

```
##
## Call:
```

```
## lm(formula = sr ~ x + y + trange, data = norm_tmin_tmax_sr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.86704 -0.74041 -0.04158  0.65201  3.05105
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.882e-16  7.162e-04     0.0        1
## x            1.327e-01  8.414e-04   157.7   <2e-16 ***
## y           -2.442e-01  8.369e-04  -291.8   <2e-16 ***
## trange       4.424e-01  8.312e-04   532.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8244 on 1325216 degrees of freedom
## Multiple R-squared:  0.3203, Adjusted R-squared:  0.3203
## F-statistic: 2.082e+05 on 3 and 1325216 DF,  p-value: < 2.2e-16
```

```r
# Create linear regression-11
# response: trange
# predictors: sr
lm_trange = lm(trange~sr, data = norm_tmin_tmax_sr)
summary(lm_trange)
```

```
##
## Call:
## lm(formula = trange ~ sr, data = norm_tmin_tmax_sr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5300 -0.5886 -0.0229  0.5635  3.5246
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.366e-16  7.657e-04     0.0        1
## sr          4.722e-01  7.657e-04   616.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8815 on 1325218 degrees of freedom
## Multiple R-squared:  0.223,  Adjusted R-squared:  0.223
## F-statistic: 3.803e+05 on 1 and 1325218 DF,  p-value: < 2.2e-16
```

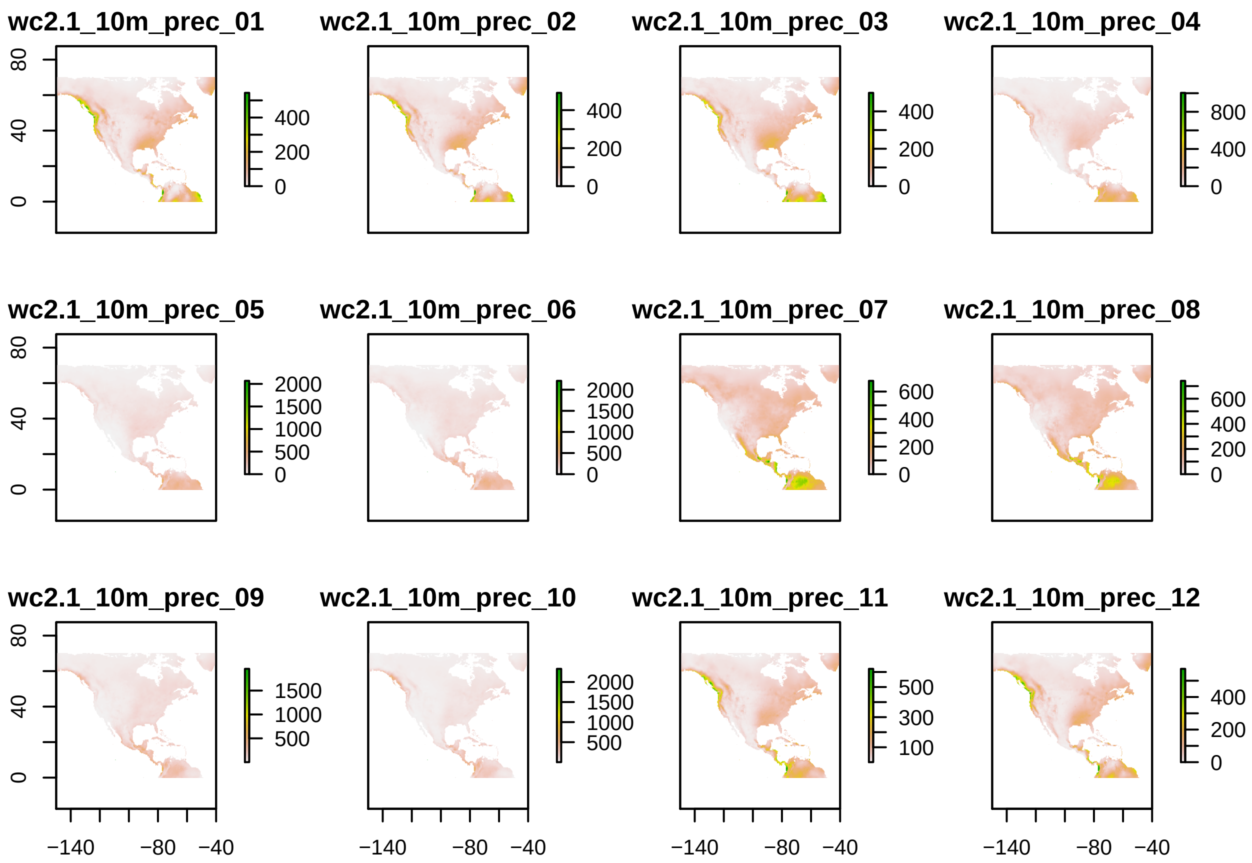
```r
# Create linear regression-12
# response: trange
# predictors: x,y,sr
lm_xy_trange = lm(trange~x+y+sr, data = norm_tmin_tmax_sr)
summary(lm_xy_trange)
```

```
##
## Call:
## lm(formula = trange ~ x + y + sr, data = norm_tmin_tmax_sr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -3.6116 -0.5318  0.0196  0.5110  3.0953
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.604e-16  6.793e-04     0.0        1
## x           -4.173e-01  7.194e-04  -580.1   <2e-16 ***
## y           -2.571e-01  7.880e-04  -326.2   <2e-16 ***
## sr           3.981e-01  7.479e-04   532.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7821 on 1325216 degrees of freedom
## Multiple R-squared:  0.3884, Adjusted R-squared:  0.3884
## F-statistic: 2.805e+05 on 3 and 1325216 DF,  p-value: < 2.2e-16
```

Importing the precipitation((1970-2000)

```
############################################################
### Importing the tmax(maximum temperature) (1970-2000)
############################################################
prec_1<- raster("precipitation/wc2.1_10m_prec_01.tif")
prec_2<- raster("precipitation/wc2.1_10m_prec_02.tif")
prec_3<- raster("precipitation/wc2.1_10m_prec_03.tif")
prec_4<- raster("precipitation/wc2.1_10m_prec_04.tif")
prec_5<- raster("precipitation/wc2.1_10m_prec_05.tif")
prec_6<- raster("precipitation/wc2.1_10m_prec_06.tif")
prec_7<- raster("precipitation/wc2.1_10m_prec_07.tif")
prec_8<- raster("precipitation/wc2.1_10m_prec_08.tif")
prec_9<- raster("precipitation/wc2.1_10m_prec_09.tif")
prec_10<- raster("precipitation/wc2.1_10m_prec_10.tif")
prec_11<- raster("precipitation/wc2.1_10m_prec_11.tif")
prec_12<- raster("precipitation/wc2.1_10m_prec_12.tif")


prec_stack <- stack(prec_1,prec_2,prec_3,prec_4,prec_5,
                    prec_6,prec_7,prec_8,prec_9,prec_10,
                    prec_11,prec_12)
Crop <- c(-150,-40, 0,70)
prec_crop <- crop(prec_stack, Crop)
# Plot cropped area
plot(prec_crop)
```




```
############################################################
### Convert the RasterLayer to Matrix for tmax
############################################################
prec_points_1 <- rasterToPoints(prec_1)
prec_points_2 <- rasterToPoints(prec_2)
prec_points_3 <- rasterToPoints(prec_3)
prec_points_4 <- rasterToPoints(prec_4)
prec_points_5 <- rasterToPoints(prec_5)
prec_points_6 <- rasterToPoints(prec_6)
prec_points_7 <- rasterToPoints(prec_7)
prec_points_8 <- rasterToPoints(prec_8)
prec_points_9 <- rasterToPoints(prec_9)
prec_points_10 <- rasterToPoints(prec_10)
prec_points_11 <- rasterToPoints(prec_11)
prec_points_12 <- rasterToPoints(prec_12)


# Merge the data from Jan to Dec
total_prec <- rbind(prec_points_1, prec_points_2,prec_points_3,prec_points_4,prec_points_5,prec_points_(
                       prec_points_7,prec_points_8,prec_points_9,prec_points_10,prec_points_11,prec_poi
total_prec <- as.data.frame(total_prec)
# Crop the data within the area
total_prec <- subset(total_prec, total_prec$x >=-150 & total_prec$x <=-40
                        & total_prec$y >= 0 & total_prec$y <=70)
```

```r
##################################################################
# Merging the tmin, tmax, sr, precipitation
##################################################################
total_tmin_tmax_prec_sr <- cbind(total_tmax_sr,tmin=total_tmin_sr[,3],prec=total_prec[,3])

# Normalization
summary(total_tmin_tmax_prec_sr)
```

```
##        x                  y                tmax              sr
##  Min.   :-149.92   Min.   : 0.08333   Min.   :-31.340   Min.   :    1
##  1st Qu.:-112.75   1st Qu.:35.91667   1st Qu.: -2.202   1st Qu.: 6690
##  Median : -96.92   Median :49.58333   Median : 12.290   Median :14385
##  Mean   : -95.53   Mean   :45.82135   Mean   : 10.033   Mean   :12971
##  3rd Qu.: -76.75   3rd Qu.:60.91667   3rd Qu.: 24.022   3rd Qu.:19034
##  Max.   : -40.08   Max.   :69.91667   Max.   : 44.687   Max.   :29527
##       tmin              prec
##  Min.   :-39.3305   Min.   :   0.00
##  1st Qu.:-11.2216   1st Qu.:  22.00
##  Median :  1.1467   Median :  47.00
##  Mean   : -0.8469   Mean   :  68.13
##  3rd Qu.:  9.9763   3rd Qu.:  88.00
##  Max.   : 28.6363   Max.   :2328.00
```

```r
preproc1 <- preProcess(total_tmin_tmax_prec_sr, method=c("center", "scale"))
norm_tmin_tmax_prec_sr <- predict(preproc1, total_tmin_tmax_prec_sr)
summary(norm_tmin_tmax_prec_sr)
```

```
##        x                   y                tmax              sr
##  Min.   :-2.27918   Min.   :-2.4862   Min.   :-2.4931   Min.   :-1.8001
##  1st Qu.:-0.72165   1st Qu.:-0.5384   1st Qu.:-0.7373   1st Qu.:-0.8717
##  Median :-0.05812   Median : 0.2045   Median : 0.1359   Median : 0.1962
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.78700   3rd Qu.: 0.8205   3rd Qu.: 0.8429   3rd Qu.: 0.8415
##  Max.   : 2.32358   Max.   : 1.3098   Max.   : 2.0881   Max.   : 2.2978
##       tmin             prec
##  Min.   :-2.5080   Min.   :-0.9579
##  1st Qu.:-0.6761   1st Qu.:-0.6486
##  Median : 0.1299   Median :-0.2971
##  Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.7054   3rd Qu.: 0.2794
##  Max.   : 1.9214   Max.   :31.7732
```

Linear Regression Model to predict solar radiation

```r
# Create linear regression-13
# response: sr
# predictors: x,y,tmin,tmax,prec
lm_tmin_tmax_prec_sr = lm(sr~., data = norm_tmin_tmax_prec_sr)
summary(lm_tmin_tmax_prec_sr)
```

```
##
## Call:
## lm(formula = sr ~ ., data = norm_tmin_tmax_prec_sr)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -2.01399 -0.43332 -0.01122  0.35092  2.20595
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.795e-15  4.854e-04    0.00        1
## x            1.690e-01  5.710e-04  296.00   <2e-16 ***
## y            3.625e-01  8.183e-04  442.99   <2e-16 ***
## tmax         1.822e+00  2.978e-03  611.91   <2e-16 ***
## tmin        -7.907e-01  2.955e-03 -267.60   <2e-16 ***
## prec        -6.054e-02  6.633e-04  -91.27   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5588 on 1325214 degrees of freedom
## Multiple R-squared:  0.6878, Adjusted R-squared:  0.6878
## F-statistic: 5.838e+05 on 5 and 1325214 DF,  p-value: < 2.2e-16
```

```r
# Create linear regression-14
# response: sr
# predictors: x,y,tmin,tmax
lm_sr_tmin_tmax = lm(sr~x+y+tmin+tmax, data = norm_tmin_tmax_sr)
summary(lm_sr_tmin_tmax)
```

```
##
## Call:
## lm(formula = sr ~ x + y + tmin + tmax, data = norm_tmin_tmax_sr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.95996 -0.43558 -0.01501  0.34717  2.24736
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.703e-15  4.869e-04     0.0        1
## x            1.685e-01  5.728e-04   294.1   <2e-16 ***
## y            3.897e-01  7.645e-04   509.7   <2e-16 ***
## tmin        -9.141e-01  2.635e-03  -346.9   <2e-16 ***
## tmax         1.938e+00  2.706e-03   716.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5605 on 1325215 degrees of freedom
## Multiple R-squared:  0.6858, Adjusted R-squared:  0.6858
## F-statistic: 7.231e+05 on 4 and 1325215 DF,  p-value: < 2.2e-16
```