



CLIMATE CHANGE WILL STRIKE BUDAPEST CRITICALLY

Posted by Gergely Lajtai-Szabó | Jul 9, 2017 | Society | 0

Summers are becoming hotter and hotter since the Industrial Revolution. If humanity cannot decrease the use of carbon-dioxide and other greenhouse gases, the temperature will keep on rising. This phenomenon seems to affect Budapest in particular, according to [24.hu](#).

Klimawandel

Die durch den Treibhauseffekt hervorgerufene Erderwärmung gilt als größte Herausforderung für die Menschheit. Allen ist klar, dass schnell etwas getan werden muss.

Datum 08.05.2018
Themensseiten Energie
Schicken Sie uns Ihre P
Drucken Seite drucker
Permalink http://p.dw.com



Die Prognose der Klimaexperten ist drastisch: Wenn der Treibhausgasausstoß nicht deutlich gesenkt wird, könnte die Durchschnittstemperatur auf der Erde bis zum Jahr 2100 um weitere fünf Grad steigen - mit verheerenden Folgen für das Überleben in vielen Regionen.

- Matern und eine gute Nachbarschaft schützen gegen die Fluten der Donau
- Westafrika: Fulani-Konflikt spitzt sich zu 05.05.2018
- Klimagesprache: Ein Gehör für die schwächsten Länder 05.05.2018
- Reden, um das Wetter zu retten 30.04.2018



New climate 'feedback loop' discovered in freshwater lakes

... over the next 50 years because of a novel "feedback loop" say scientists. Climate... greenhouse gases to be produced in freshwater lakes. They say the warming climate... different climate scenarios. "This forecasts a whole range of these different aquatic..."

News | Science & Environment



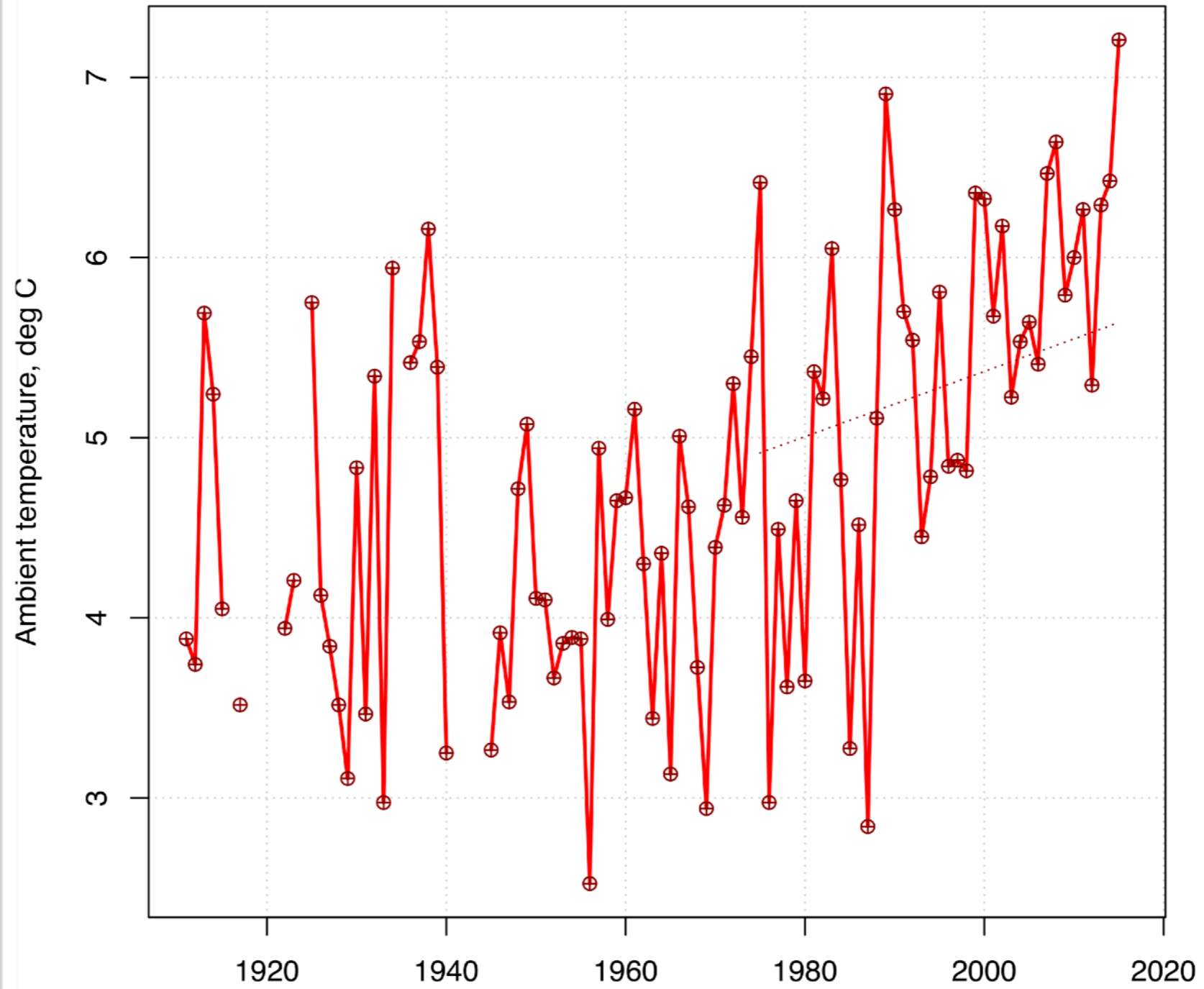
Rising levels of 'frustration' at UN climate stalemate

... threatening to limit progress in UN climate negotiations. Discussions between negotiators... on the Paris climate agreement. But developing countries say they are "frustrated" with the lack... inadequate" they said. 2018 marks a critical stage in global climate negotiations...

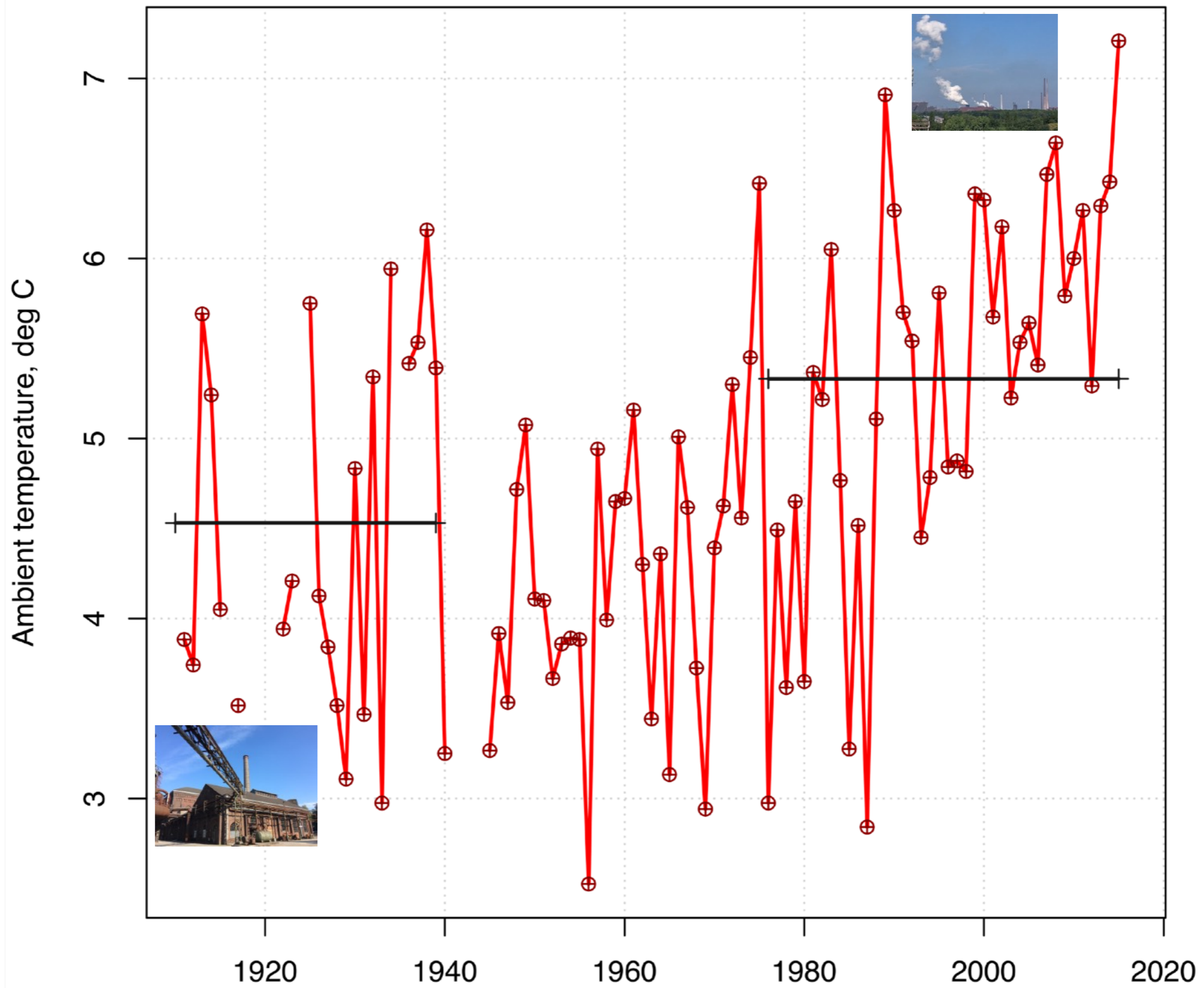
News | Science & Environment



Energy under climate change



Meteorological observations



DATASET

CMIP5 simulation data

- netCDF format
- spatial: global area
- temporary resolution?

Multi-annual averages
are of interest

Project 1 ▲

type to filter... Sort by ▾

Show all

IPCC-AR5_CMIP5 (IPCC As... 12155

Topic Name 1 ▲

type to filter... Sort by ▾

Show all

air_temperature 12155

Keywords 133 ▾

Aggregation 4 ▲

type to filter... Sort by ▾

Show all

<input checked="" type="checkbox"/> mon	6358
<input type="checkbox"/> day	4468
<input type="checkbox"/> 6hr	1044
<input type="checkbox"/> 3hr	285

R-MINDED APPROACH

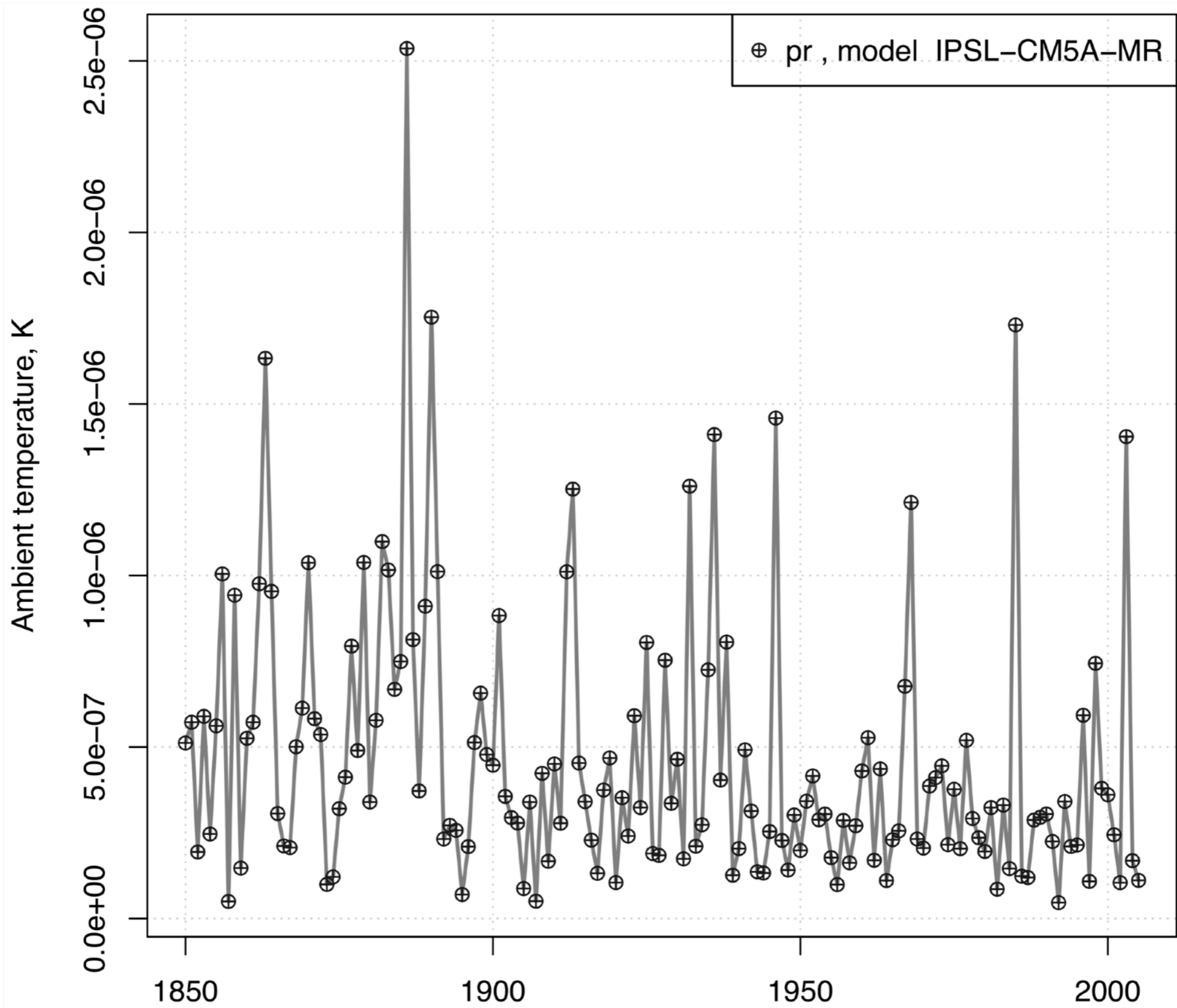
Classic R, ::raster, ::RCMIP5

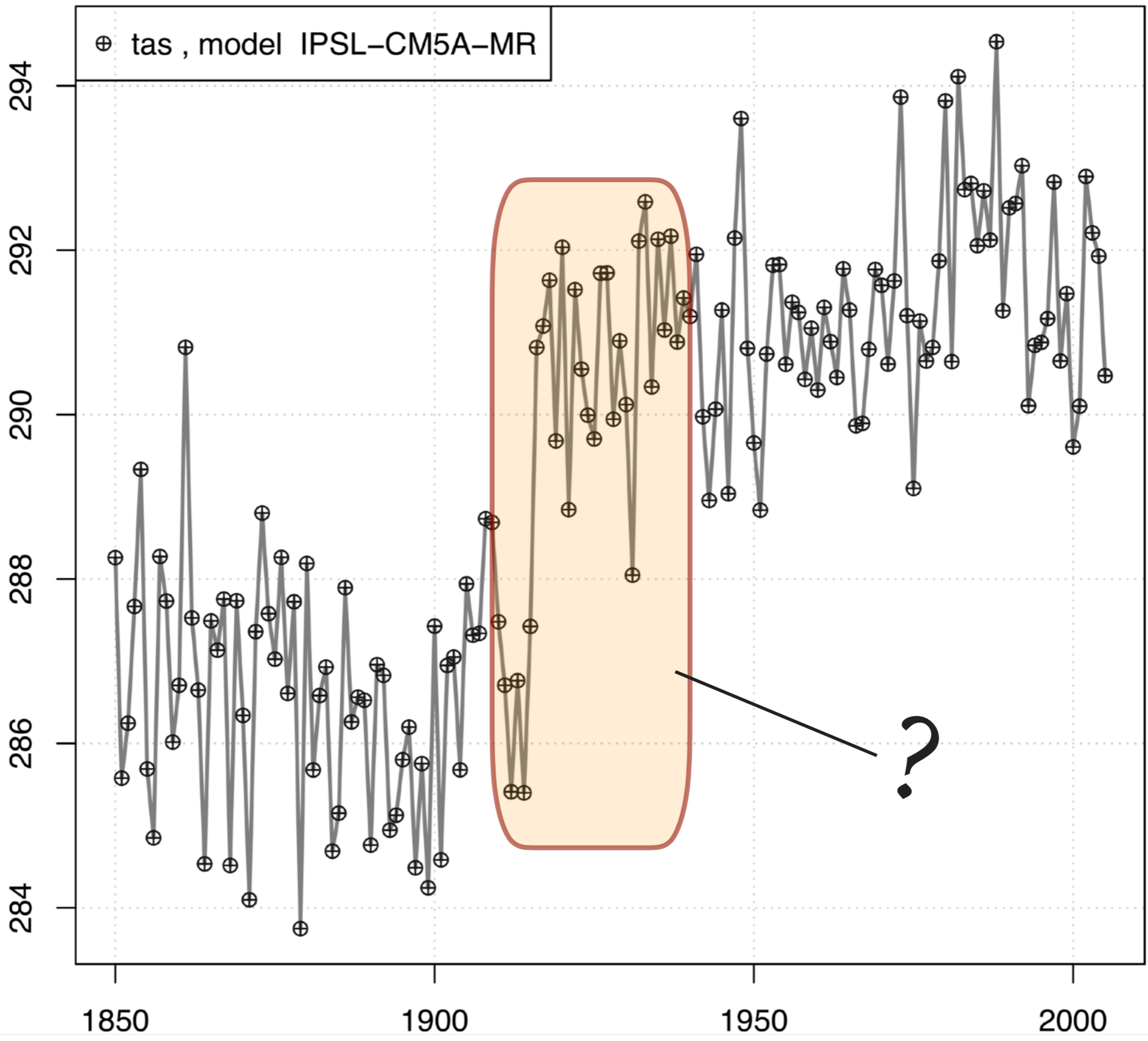
- 1) *set the items' ids to select*
- 2) *subset*
- 3) *apply a function (spacial averaging)*

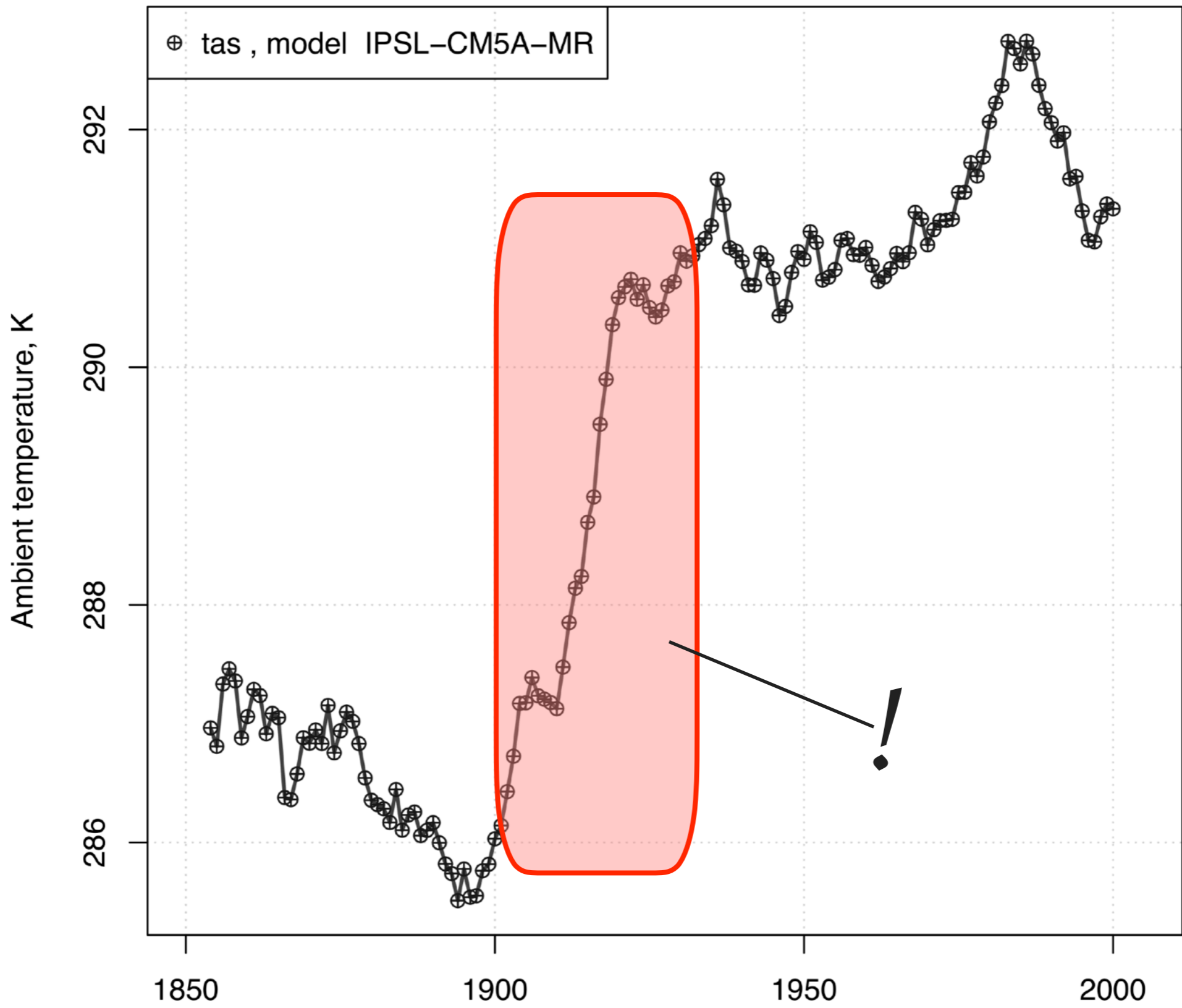
```
# 1) set a processed month
month_nL <- 3L

# 2) subset the data
months_nc <- as.numeric(format(time_nc_date, "%m"))
nc_dates_slctd <- time_nc_date[which(months_nc %in% month_nL)]
T_3D_array_slctd <- T_3D_array_to_process[, , which(months_nc %in% month_nL)]

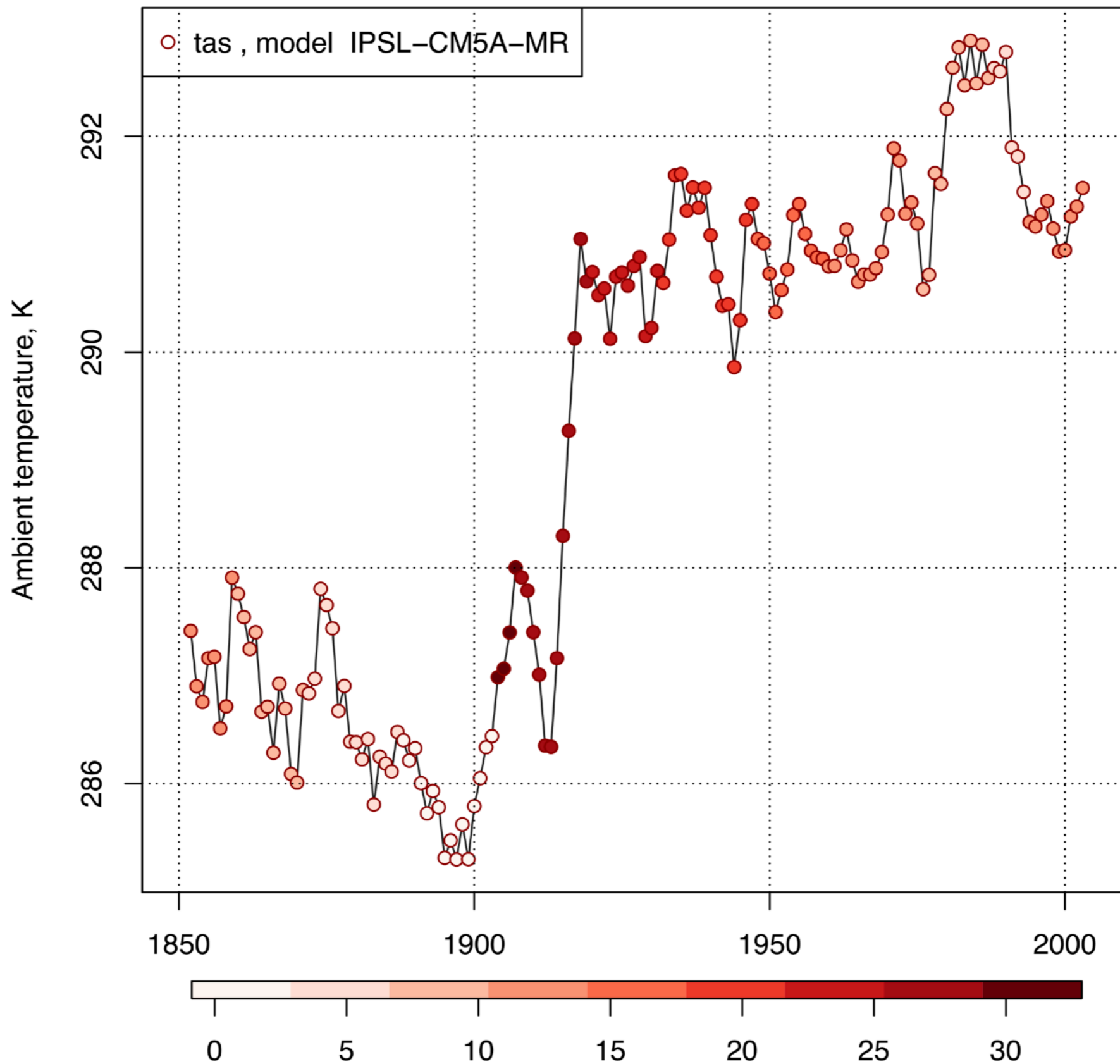
# 3) make calculations
T_time_series_slctd <- sapply(function(i) mean(T_3D_array_slctd[, , i]),
  X = seq(along.with = T_3D_array_slctd[1, 1, ]))
```







Points colors according to the day of the month



SOLUTION

Use interpolation to regular seasonally time

instead of

```
04" "1901-03-04" "1902-03-04" "1903-03-04" "1904-03-03" "1905-03-03" "1906-03-03" "1907-03-03" "1908-03-02" "1909-03-02"
02" "1911-03-02" "1912-03-01" "1913-03-01" "1914-03-01" "1915-03-01" "1916-03-31" "1917-03-31" "1918-03-31" "1919-03-31"
30" "1921-03-30" "1922-03-30" "1923-03-30" "1924-03-29" "1925-03-29" "1926-03-29" "1927-03-29" "1928-03-28" "1929-03-28"
```

work with

```
[3,] 1892-04-03 1893-04-03 1894-04-03 1895-04-03 1896-04-04 1897-04-04 1898-04-04
      [,50]      [,51]      [,52]      [,53]      [,54]      [,55]      [,56]
[1,] "1899-02-03" "1900-02-03" "1901-02-03" "1902-02-03" "1903-02-03" "1904-02-03" "1905-02-02"
[2,] "1899-03-04" "1900-03-04" "1901-03-04" "1902-03-04" "1903-03-04" "1904-03-03" "1905-03-03"
[3,] "1899-04-04" "1900-04-04" "1901-04-04" "1902-04-04" "1903-04-04" "1904-04-03" "1905-04-03"
      [,57]      [,58]      [,59]      [,60]      [,61]      [,62]      [,63]
[1,] "1906-02-02" "1907-02-02" "1908-02-02" "1909-02-01" "1910-02-01" "1911-02-01" "1912-02-01"
[2,] "1906-03-03" "1907-03-03" "1908-03-02" "1909-03-02" "1910-03-02" "1911-03-02" "1912-03-01"
[3,] "1906-04-03" "1907-04-03" "1908-04-02" "1909-04-02" "1910-04-02" "1911-04-02" "1912-04-01"
      [,64]      [,65]      [,66]      [,67]      [,68]      [,69]      [,70]
[1,] "1913-01-31" "1914-01-31" "1915-01-31" "1916-02-29" "1917-02-28" "1918-02-28" "1919-02-28"
[2,] "1913-03-01" "1914-03-01" "1915-03-01" "1916-03-31" "1917-03-31" "1918-03-31" "1919-03-31"
[3,] "1913-04-01" "1914-04-01" "1915-04-01" "1916-04-30" "1917-04-30" "1918-04-30" "1919-04-30"
      [,71]      [,72]      [,73]      [,74]      [,75]      [,76]      [,77]
[1,] "1920-02-28" "1921-02-27" "1922-02-27" "1923-02-27" "1924-02-27" "1925-02-26" "1926-02-26"
[2,] "1920-03-30" "1921-03-30" "1922-03-30" "1923-03-30" "1924-03-29" "1925-03-29" "1926-03-29"
[3,] "1920-04-29" "1921-04-29" "1922-04-29" "1923-04-29" "1924-04-28" "1925-04-28" "1926-04-28"
      [,78]      [,79]      [,80]      [,81]      [,82]      [,83]      [,84]
[1,] "1927-02-26" "1928-02-26" "1929-02-25" "1930-02-25" "1931-02-25" "1932-02-25" "1933-02-24"
```

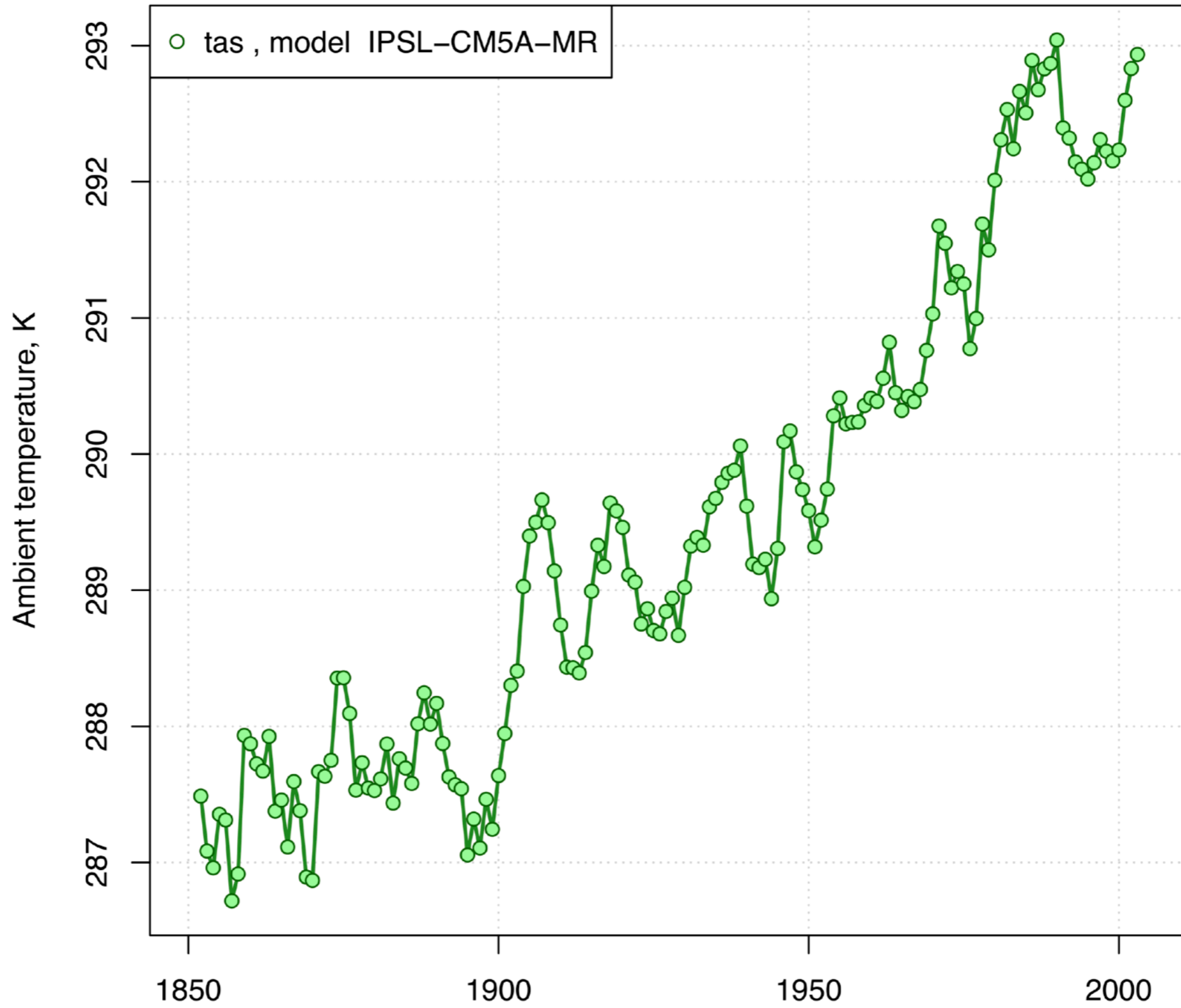
SOLUTION

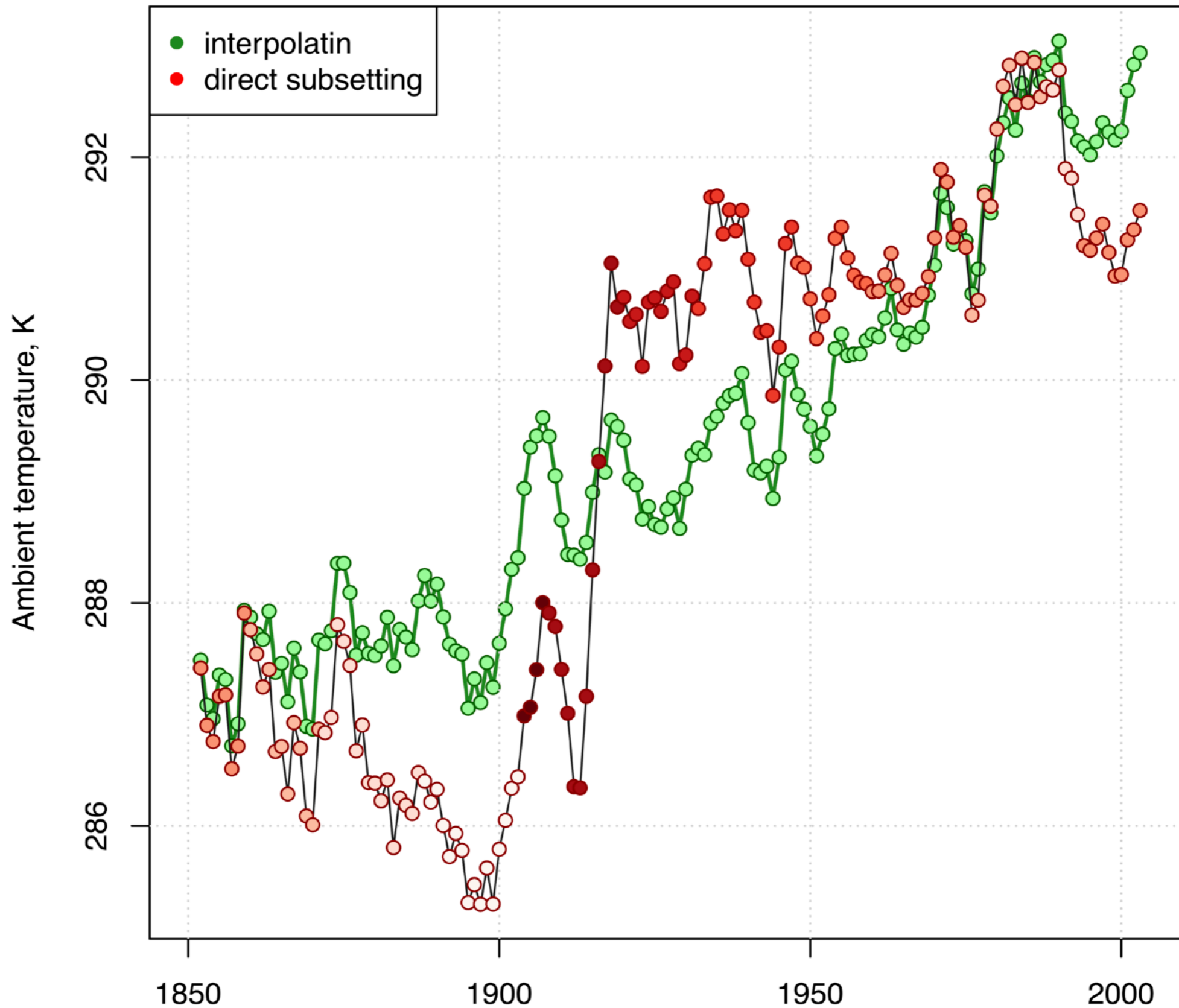
```
# 1) construct an interpolating function
# @ind_to_interp is a three-column matrix of indices to interpolate from
# @regular_date is a list data sequences to interpolate to
# @param_3D is a 3D array to process
Month_Approx <- function(i, indices_to_int, regular_date, param_3D) {
  tau_intp_from <- time_nc_date[indices_to_int[i, ]]
  dates_seq_into <- regular_date[[i]]
  T_3D_array_intp_from <- param_3D[ , , indices_to_int[i, ]]
  T_area_avr_intp_from <- lapply(function(j) mean(param_3D[ , , indices_to_int[i, j]]),
    X = seq(along.with = indices_to_int[i, ]))
  appr_res <- approx(x = tau_intp_from, y = T_area_avr_intp_from, xout = regular_date[[i]])
  T_res <- mean(appr_res$y)
  return(T_res)
}

# 2) set a data sequence
first_month_day <- ymd(paste(as.numeric(format(nc_dates_slctd, "%Y")), month_nL, 1, sep = "-"))
last_month_day <- ymd(paste(as.numeric(format(nc_dates_slctd, "%Y")),
  (month_nL + 1), 1, sep = "-")) - 1
dates_seq_intp_to <- lapply(function(k) seq.Date(from = first_month_day[k],
  to = last_month_day[k], by = 1), X = 1:length(nc_dates_slctd))

# 3) interpolate the monthly-aggregated value for all selected years
T_monthly_intpd <- sapply(function(i) Month_Approx(i, indices_to_int = ind_to_interp,
  regular_date = dates_seq_intp_to, param_3D = T_3D_array_to_process),
  X = 1:length(nc_dates_slctd))
```

After accurate interpolation





SUMMARY

Modellers have right to their own way of thinking

*Practical points to be kept in mind while work with **netCDF***

- 1) quick-and-easy solutions may be dangerous*
- 2) careful **examination of netCDF structure** is essential*
- 3) implementation to netCDF/CMIP5 packages?*

*Many thanks for your
attention!*

We highly acknowledge CMIP5 modelling groups and CERA-team for sharing CMIP5 simulation data. We are also very grateful to the Russian Weather Office for supporting meteorological observation data

The work was supported with a President Grant MK- 1494.2017.8