21MAP500 Coursework

Errata:

1. **The number in Question 2b should read “150” rather than “160” [fixed on 2021-10-14].**
2. **The data set to be created in Question 4a should be called `stop_search_1` not `stop_search` [fixed on 2021-10-14].**
3. **In Question 2b, the column `co2` should contain yearly averages of CO2 levels for those years in which more than one data point is available (also fixed the “cleaned” data set `combined_co2`) [fixed 2021-10-15].**
4. **Fixed a slight error in the “cleaned” `sea_ice_2` data set [fixed 2021-10-15].**

General instructions

Files to submit

You will submit your coursework as a single R notebook (i.e. `.Rmd` file) which can be rendered (“knitted”) to an `.html` document. Specifically, **submit both**

- your R notebook (i.e. the `.Rmd` file),
- the rendered `.html` version of your notebook (in case there are any problems knitting your `.Rmd` during marking).

Do not include any identifying information such as your name or student ID in the submitted documents

Getting started

- Create a new RStudio project.
- Within your project folder, create a folder `data`.
- Download `` ` ``nasa_global_temperature.txt`, `nasa_arctic_sea_ice.csv`, `nasa_sea_level.csv`, `nasa_carbon_dioxide.txt`, `luthi_carbon_dioxide.txt`, `nsidc_sea_ice_daily_extent.xlsx` and `stop_and_search.csv`(contained in `data_raw.zip`) to the `data` folder of your RStudio project. These files were downloaded from the following websites, where you can find additional information (some of the files also contain headers describing the variables).
    - https://climate.nasa.gov/vital-signs/global-temperature/
    - https://climate.nasa.gov/vital-signs/arctic-sea-ice/
    - https://climate.nasa.gov/vital-signs/sea-level/
    - https://climate.nasa.gov/vital-signs/carbon-dioxide/
    - https://www.ncei.noaa.gov/pub/data/paleo/icecore/antarctica/epica_domec/edc-co2-2008.txt (based on https://doi.org/10.1038/nature06949)
    - https://nsidc.org/arcticseaicenews/sea-ice-tools/
    - https://www.ethnicity-facts-figures.service.gov.uk/crime-justice-and-the-law/policing/stop-and-search/latest
- Do not modify these files under any circumstances! All data wrangling must take place entirely within the R code inside your notebook.

Getting help

The questions below instruct you to construct tibbles `nasa_temp`, `nasa_sea`, `nasa_co2`, `nasa_ice`, `nasa`, `combinded_co2`, `sea_ice_1`, `sea_ice_2`, `stop_search`, `stop_search_1` and `stop_search_2`. If you have difficulty creating one (or more) of these tibbles as instructed, you may download the corresponding CSV file (contained in `data_cleaned.zip`) and use its contents in for all subsequent tasks. For instance, if you do not manage to create the tibble `nasa_sea` in Question 1(b), you may download `nasa_sea.csv` to your `data` folder and read its contents into R as the tibble `nasa_sea`. You may then use `nasa_sea` in subsequent tasks, e.g. to create the visualisation in Question 1(b) or as one ingredients in the combined data set `nasa`.

Questions and Marks

Question 1: [22]

a. Read `nasa_global_temperature.txt` into a tibble called `nasa_temp` containing only the variables `date` and `temp`. The former should have type `date` (you can assume that month and day are January, 1st). The latter is the column named `No_smoothing` in the original file. You may understand `temp` as the *average temperature across global land and ocean surfaces in °C*. Visualise the contents of `nasa_temp`. [3]

b. Read `nasa_arctic_sea_ice.csv` into a tibble called `nasa_ice` containing only the variables `date` and `ice`. The former should have type `date` (you can assume that month and day are January 1st). The latter is the column named `extent` in the original file. You may understand `ice` as the *minimum arctic sea ice extent in million square km*. Visualise the contents of `nasa_ice`. [3]

c. Read `nasa_sea_level.csv` into a tibble called `nasa_sea` containing only variables `date` and `sea`. The former should have type `date` (you can assume that fractional years have been calculated for time zone “UTC”). The latter should be the values from the twelfth column in the original file. You may understand `sea` as the *change in sea level compared to a reference year in mm*. Visualise the contents of `nasa_sea`. [3]

d. Read `nasa_carbon_dioxide.txt` into a tibble called `nasa_co2` containing only the variables `date` and `co2`. The former should have type `date` (you can assume that the day of the month is always the 1st).The latter should be the values from the column `monthly average`. You may understand `co2` as the *average global CO2 level in parts per million (ppm)*. Visualise the contents of `nasa_co2`. [3]

e. Combine `nasa_temp`, `nasa_ice`, `nasa_sea` and `nasa_co2` into a single tibble called `nasa` without loss of any data. [4]

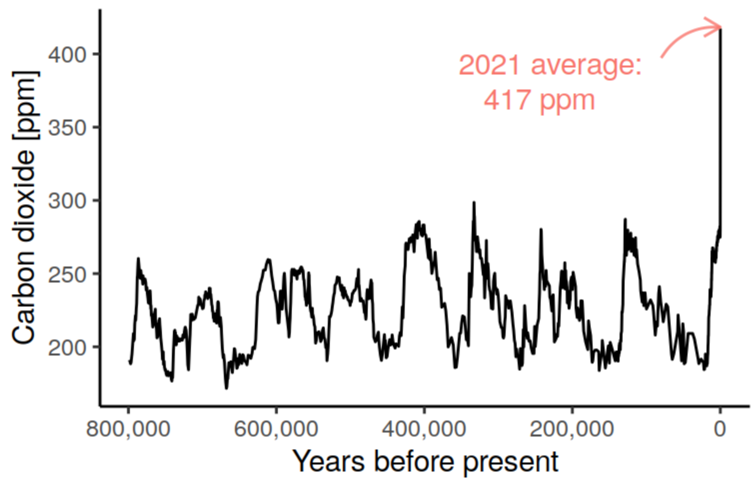
f. Visualise the correlation of the variables `co2` and `temp` in `nasa` for the years 1960–2020 in a scatterplot whose points are *sequentially* coloured by `year` such that the points associated with each decade are shaded differently. Ensure that a meaningful sequential colour scheme is used and that all axes and the legend are labeled appropriately. [6]

Question 2 [16]

a. Read the table found under under “3. Composite CO2 record (0-800 kyr BP)” in `luthi_carbon_dioxide.txt` (i.e. starting from Line 774) into a tibble called `historic_co2` containing the variables `yrbp` (“years before present”), the first column from the original file, and `co2`, the second column in the original file. You may again interpret `co2` as the *average global CO2 level in parts per million (ppm)*. [2]

b. Assume that the reference year in the original file is 2008, i.e. that `yrbp` counts the years before 2008. Change the reference year to 2021 so that, e.g., the value 137 [years before 2008] of `yrbp` should now be **150** [years before 2021]. Likewise, add a column `yrbp` to `nasa_co2` which similarly counts the years before 2021 for each measurement. Finally, combine `historic_co2` and the modified version of `nasa_co2` into a single tibble called `combined_co2` which contains only the variables `co2` **(as yearly averages where needed)** and `yrbp`. [6]

c. Recreate the following figure based on the data set `combined_co2` as accurately as possible (the placement and colour of the annotation need not match exactly). [8]

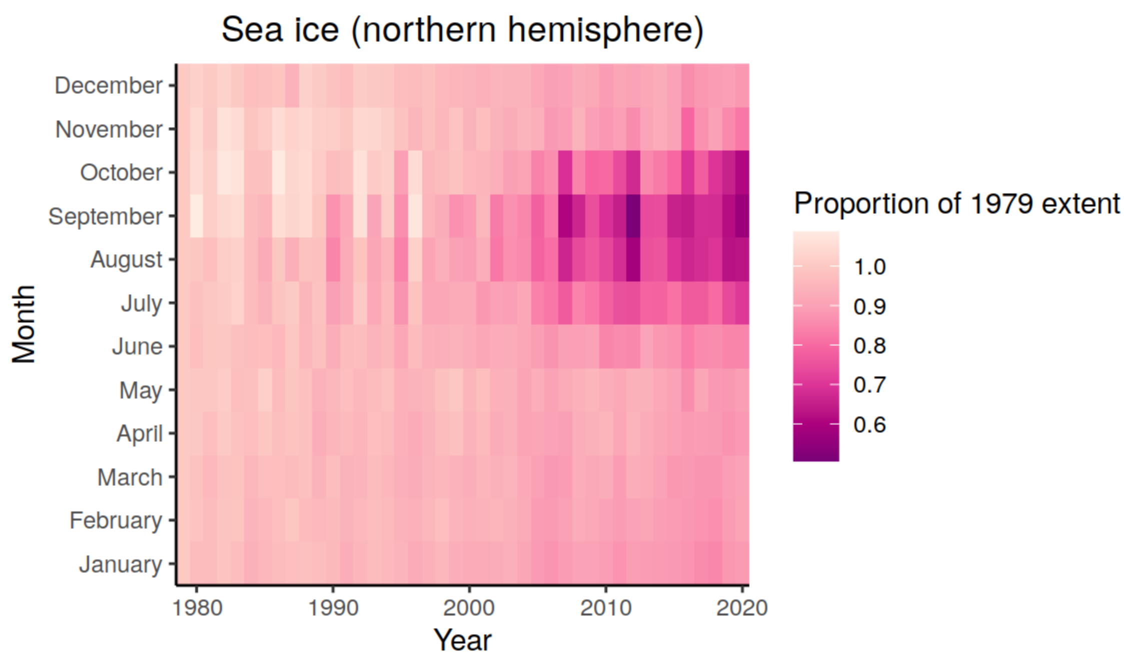


Question 3: [30]

a. Read the first spreadsheet from the file `nsidc_sea_ice_daily_extent.xlsx` into a tidy tibble called `sea_ice_1` containing the column `extent` as well as integer columns `year`, `month` and `day`. [12]

b. Transform `sea_ice_1` so that you are left with a tibble with only three variables: `year`, `month` and `proportion_baseline_extent`. The latter should be the monthly averages of the original extent divided by a month-specific baseline extent. As a baseline, take the monthly averages from the year 1979. Store the output in a tibble called `sea_ice_2`. [10]

c. Recreate the following figure based on the data set `sea_ice_2` as accurately as possible (the colour scheme is `RdPu` from the **RColorBrewer** package). Note that years with incomplete records (i.e. 1978 and 2021) are not shown. [8]



Question 4: [32]

a. Load the data set `stop_and_search.csv` into a tibble called `stop_search_1`. Ensure that all variables have a sensible data type and that long variable/column names are avoided by renaming `Number of stops...` to `stops` `Population by...` to `population` and `Rate of...` to `rate`. Focus only on cases in which `ethnicity` is one of “All”, “Asian”, “Black”, “White”, “Other”. You may discard all other cases and any redundant variables. [2]

b. Add a column `relative_disparity` to `stop_search_1` which, for each ethnicity, gives the stop-and-search rate divided by the stop-and-search rate for “White”. Store the output in `stop_search_2`. [6]

c. State three interesting and specific questions that can be answered using the data. For each question, also mention how it is operationalised. The questions must be qualitatively different. [6 points – 2 per operationalised question]

d. For each of the three operationalised questions from (c), provide an answer in the form of one or more suitable visualisations along with a brief text (only one or two but full sentences) explaining how the figure provides the answer to the question. [18 points – 6 per question]

Assessment criteria

To obtain full marks in each question your submitted R notebook must satisfy the following conditions.

1. Reproducibility

- Your notebook must be able to be “knit” on another computer which is running the latest versions of R, RStudio and all relevant packages and has access to the raw data organised in the same folder structure as mentioned under “General instructions”. In particular, this means that
  - your project folder must contain a folder `data` which holds all your data files as instructed above,
  - your notebook must specify the paths to the data using relative – not absolute – paths,
  - any data wrangling/data cleaning must be done via the R code inside your notebook.

2. Tidyverse syntax

- All data importation and data manipulation must be achieved using **tidyverse** commands and syntax, in particular the “pipe” operator `%>%`; data sets must be stored in `tibble` objects rather than base R `data.frame` objects.
- All visualisations must be created using **ggplot2** commands and syntax.

3. Figure formatting

- You must use meaningful plot types.
- If you use colours in your figures, these must not be redundant and the colour scheme must be appropriate.
- All axes and legends must be appropriately labeled using words that are understandable to someone who has not seen your code. In particular, avoid all but common abbreviations in figures unless absolutely necessary.
- The figures should have a consistent look (e.g. when using colour, the same variables should be represented by the same colours throughout the report and, e.g., you should avoid having `theme_dark()` in one figure and `theme_grey()` in another).

4. Data

- For each question, you must ensure that variables in the cleaned data frames are as instructed and that numbers are stored in numeric (i.e. `integer` or `double`) columns and dates in `date` columns.
- Check each data set for obvious data-entry errors (e.g. if a measurement is indicated as having been taken in the year 2022); exclude such points during data cleaning and make a brief note of this in accompanying text.

5. Coding style (excluding comments)

- Code layout and naming conventions for variables and functions must follow the style guidelines from Section 2.1–2.4 and 2.6 of Chapter 2 and Section 1.4.6 of Chapter 4 of the lecture notes.
- You must use “snake_case” for naming objects and files and avoid spaces in file names.
- Lines of code (excluding comments) should not be longer than 80 characters.
- Additional discussion of results, such as in Question 4(d) should be written in a text block (i.e. not as a comment inside a code chunk).

6. Comments

- Add a brief comment (one sentence) **after each line of R code**.
- This comment should explain the purpose of the line of code.
- Note that this contradicts the best practices for commenting taught in Chapters 1 and 2 of the lecture notes.
- These comments will be marked selectively (i.e. only in certain pre-determined questions).