MM916 Project 1: Exploring predictions of extreme heat under climate change

Background

Climate-change projections are often described in terms of changes in *average* temperatures: for example, the 2016 Paris Agreement is an international treaty to limit future warming to 2 °C (in the global average). But often the worst impacts of climate change are driven by changes in *extremes*, not the mean: heat waves and cold snaps, floods and droughts, which have only an indirect relationship with changes in the mean. This blog post explains this further from a statistical perspective.

Every 6-7 years, the UN Intergovernmental Panel on Climate Change (IPCC) issues a huge report reflecting the current consensus among the world's scientists on the science of climate change, the likely global impacts, and what we can do about it. Detailed 3D computer simulations of the atmosphere, land, and ocean are at the heart of this process. There are currently about 60 global earth-system *models* (ESMs) run by research groupd around the world (with names like NorESM, from Norway, MRI from Japan, GFDL from the US, UKESM from the UK): each one is a distinct piece of computer code, perhaps 10,000 lines long. For every IPCC report, these research groups perform the same set of virtual experiments across all the models: a set of what-if *scenarios* which ask, essentially, if humans emit the following amount of CO2 over the next century, how will the climate respond? The EU Copernicus project provides summary measures of extreme weather (hot/cold/dry/wet) from all the scenarios in all the climate models included in the IPCC process.

This project will take you through an exploration of a subset of this dataset.

About the data

Even the Copernicus extreme-weather summaries are huge datasets, each summary consisting of a grid of values covering the earth for every day in past and future centuries. For this project, we downloaded results from three *scenarios* for each of three *models* (GFDL, MRI, NorESM). The three scenarios are

- *RCP2.6*, an optimistic scenario, in which societies around the world immediately change course and get to work reducing emissions quickly: this is essentially the scenario in which the world actually meets the promises it made in the Paris Agreement.
- *RCP8.5*, a more pessimistic scenario (but by no means the worst-case scenario), in which the world carries on along its present course for several more decades.
- *historical*, an eapproximate reconstruction of the climate over the past 100+ years than can be used as a baseline.

For each scenario in each model, we have retained two extreme-weather *metrics*, averaged into yearly values:

- `wsdi` (warm spell duration index), which summarises how often per year a particular location experiences 6 days of conditions that, back in the period before global warming began, we would have considered "hot" (that is, above the 90th percentile of daily temperature in that location)
- `txx` (maximum temperature), the highest daily-high temperature reached in each year

We retained only the locations of cities with populations above one million. You can find the results in a data frame named `ex` saved inside a file named *climate_extremes.RData*: the command `load('climate_extremes.RData')` will load the data frame `ex` along with a subset of it used in Q1.

You will need one additional dataset: `world.cities` in the `maps` package gives information on lots of world cities including population (`pop`). You can access this data with the lines

```
library(maps)
data(world.cities)
```

This creates `world.cities` as a data frame in your Environment.

Instructions

Use these data to answer the questions below. Make a single Word, PDF, or RMarkdown file containing your answers to the questions below, including the R code you used to answer them, and the plots produced by your code. Submit this document via MyPlace.

It is not necessary to run any statistical tests of your interpretations of the data. This project is about *exploratory* data analysis and so guesses based on what you see by eye are sufficient (this is how statistical hypotheses are generated, not tested).

Please do not be discouraged or skip later questions if you have difficulty with early questions. We will give partial credit for partial solutions, including code that has the right idea but does not run, plots that include some but not all required elements, and so on.

You can make use of everything on MyPlace and also general web searches, but you may not work in groups or copy from a classmate: these are individual projects and everything below needs to be your own work.

Q1 What's in this dataset? (3, 3, 5, 3, 3 points)

(a) What years are covered by each of the three model scenarios? Make a table that gives the start and end year for each scenario.

```
ex %>% group_by(scenario) %>% summarise(start = min(year), end = max(year))
```

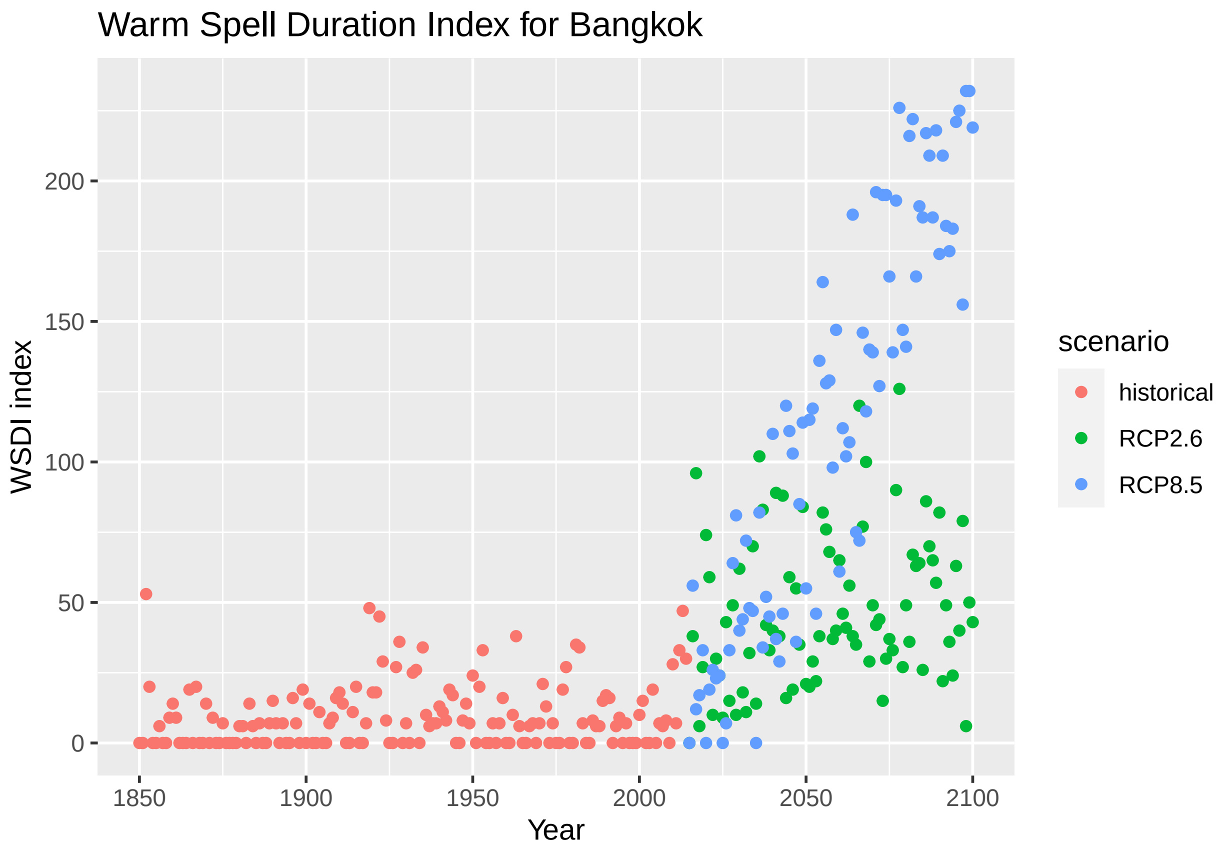
```
## # A tibble: 3 x 3
##   scenario   start   end
##   <chr>      <dbl> <dbl>
## 1 historical  1850  2014
## 2 RCP2.6      2015  2100
## 3 RCP8.5      2015  2100
```

(b) Let's look at the data from one example city. The dataframe `bangkokGFDL` contains the data for Bangkok, Thailand from the GFDL model. This is a subset of the full data frame `ex`. Give the tidyverse code that would produce `bangkokGFDL` from `ex`.

```
bangkokGFDL = ex %>% filter(city=="Bangkok" & model=="GFDL")
```

(c) Plot `wsdi` for Bangkok in all three scenarios against year on one set of axes, so that you have a timeline that shows the past together with two possible futures. Give each scenario its own colour or symbol type, and give the plot a descriptive title and axis labels. Describe the patterns you see in a 1-2 sentences.

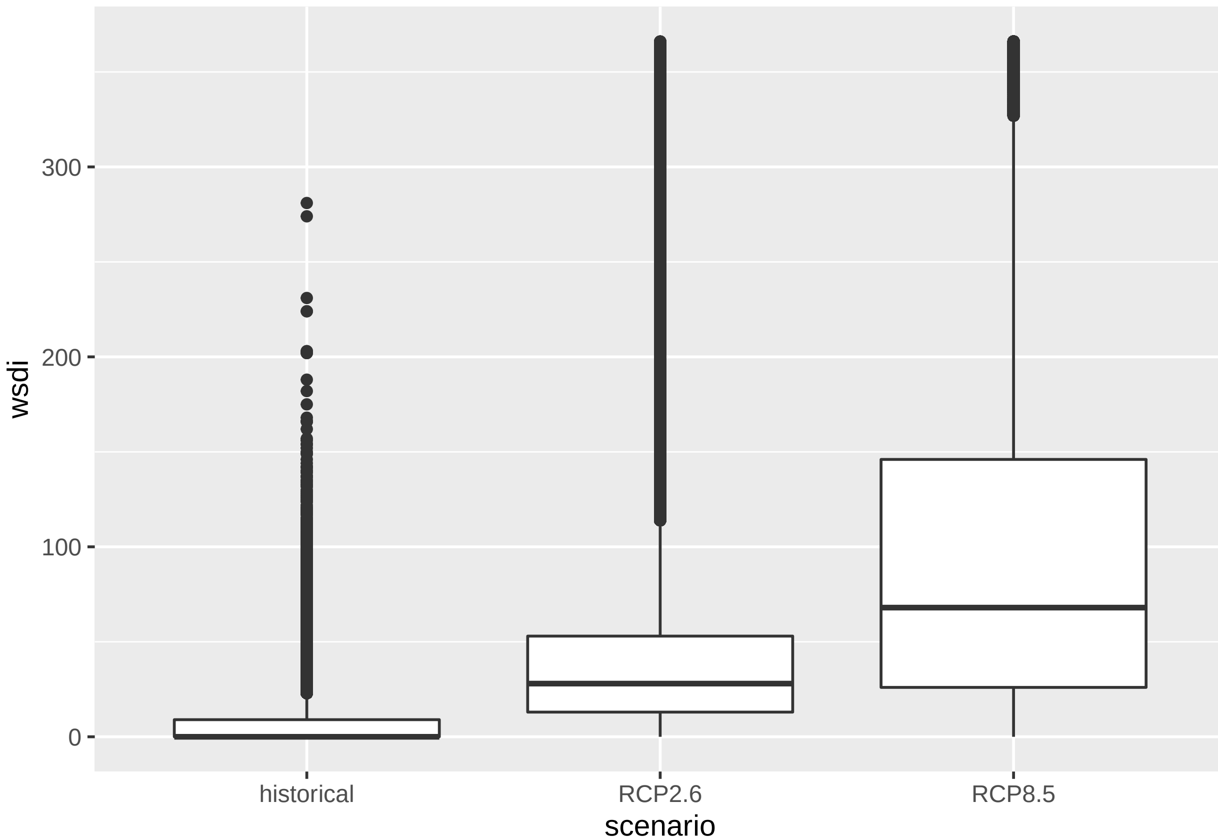
```
bangkokGFDL %>% ggplot() +
  geom_point(aes(year,wsdi,colour=scenario)) +
  labs(title="Warm Spell Duration Index for Bangkok",
       x="Year",y="WSDI index")
```




```
# WSDI in Bangkok was low throughout the historical period (almost by definition, since it's a measure
```

(d) Make a plot that summarises the overall distribution of values of `wsdi`, within each scenario. How do the results compare across scenarios, and does this make sense in terms of what you know about climate change?

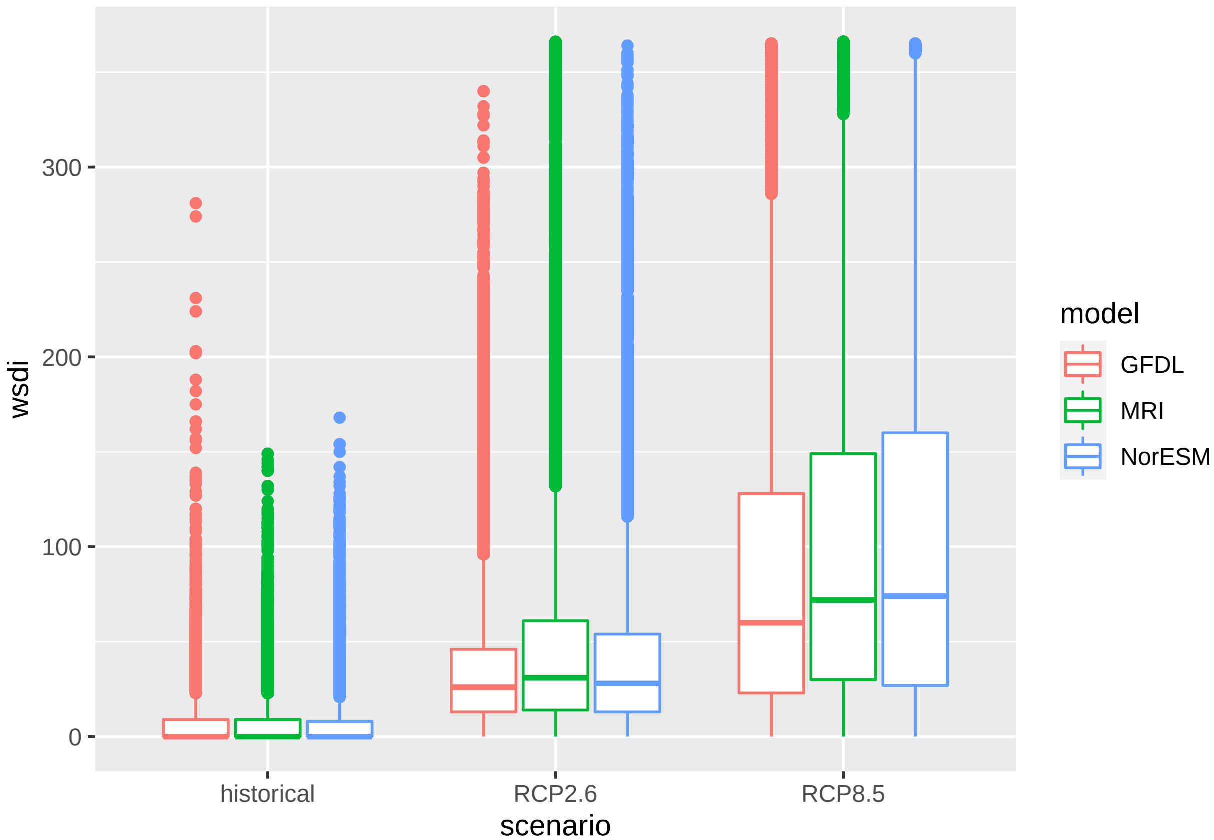
```
ggplot(ex) + geom_boxplot(aes(scenario,wsdi))
```




```
# As expected given global warming, warm spell duration appears to be higher in both
# future scenarios than in the historical period, and highest for scenario RCP8.5
# in which the world does not take rapid action to slow down climate change.
```

(e) Modify this plot to show results for all three models for each of the three scenarios.

```
ggplot(ex) + geom_boxplot(aes(scenario,wsdi,color=model))
```



Q2 High temperatures at the end of the century (5, 3, 5, 5 points)

(a) Write a function called `futureHighs` that takes a city name and the `ex` data frame as input and returns two values in a named list: the annual high temperature (`txx`) averaged over the period 2080-2100 for that city in the RCP2.6 scenario, and the same quantity in the RCP8.5 scenario. You can take an average across all three climate models or just rely on one, as you prefer. Test your function using the city of your choice.

```
futureHighs <- function(ex,cityName) {
  ex1 = ex %>% filter(city==cityName & year>=2080 & year <= 2100)
  txxRCP2.6 = mean(ex1$txx[ex1$scenario=="RCP2.6"],na.rm=TRUE)
  txxRCP8.5 = mean(ex1$txx[ex1$scenario=="RCP8.5"],na.rm=TRUE)
  list(txxRCP2.6 = txxRCP2.6, txxRCP8.5 = txxRCP8.5)
}

futureHighs(ex,"Bangkok")
```

```
## $txxRCP2.6
## [1] 40.98097
##
## $txxRCP8.5
## [1] 44.03761
```

(b) Modify your function so that it returns a useful error message if the city is not in the dataset. Test the modification.

```
futureHighs <- function(ex,cityName) {
  if (cityName %in% ex$city) {
```

```
    ex1 = ex %>% filter(city==cityName & year>=2080 & year <= 2100)
    txxRCP2.6 = mean(ex1$txx[ex1$scenario=="RCP2.6"],na.rm=TRUE)
    txxRCP8.5 = mean(ex1$txx[ex1$scenario=="RCP8.5"],na.rm=TRUE)
    list(txxRCP2.6 = txxRCP2.6, txxRCP8.5 = txxRCP8.5)
  } else {
    stop('City name not recognised. Maybe its population is below 1 million?')
  }
}

# futureHighs(ex,"Mars")
```

(c) 'Write another function that takes a country name and a number N as input, and returns the names of the N highest-population cities in that country, in an appropriate type of variable.

```
  largestCities <- function(country,N) {
    data(world.cities)
    oneCountry <- world.cities %>% filter(country.etc==country) %>%
      arrange(desc(pop))
    largest = oneCountry$name[1:N]
    return(largest)
  }
```

(d) Make a data frame showing the projected annual high temperature 2080-2100 (as in the `futureHighs` function) for RCP8.5, for the ten largest cities in India, in descending order of population. Give the columns in the data frame appropriate names.

```
topten <- largestCities("India",10)
txx = c()
for (i in 1:10) {
  res = futureHighs(ex, topten[i])
  txx[i] = res$txxRCP8.5
}
data.frame(city=topten,futureHighTemp=txx)
```

```
##         city futureHighTemp
## 1     Bombay       40.15505
## 2      Delhi       53.20673
## 3  Bangalore       44.13045
## 4   Calcutta       50.98192
## 5    Chennai       41.62851
## 6  Ahmadabad       49.36978
## 7  Hyderabad       52.04458
## 8       Pune       45.94894
## 9      Surat       46.23534
## 10    Kanpur       53.24059
```

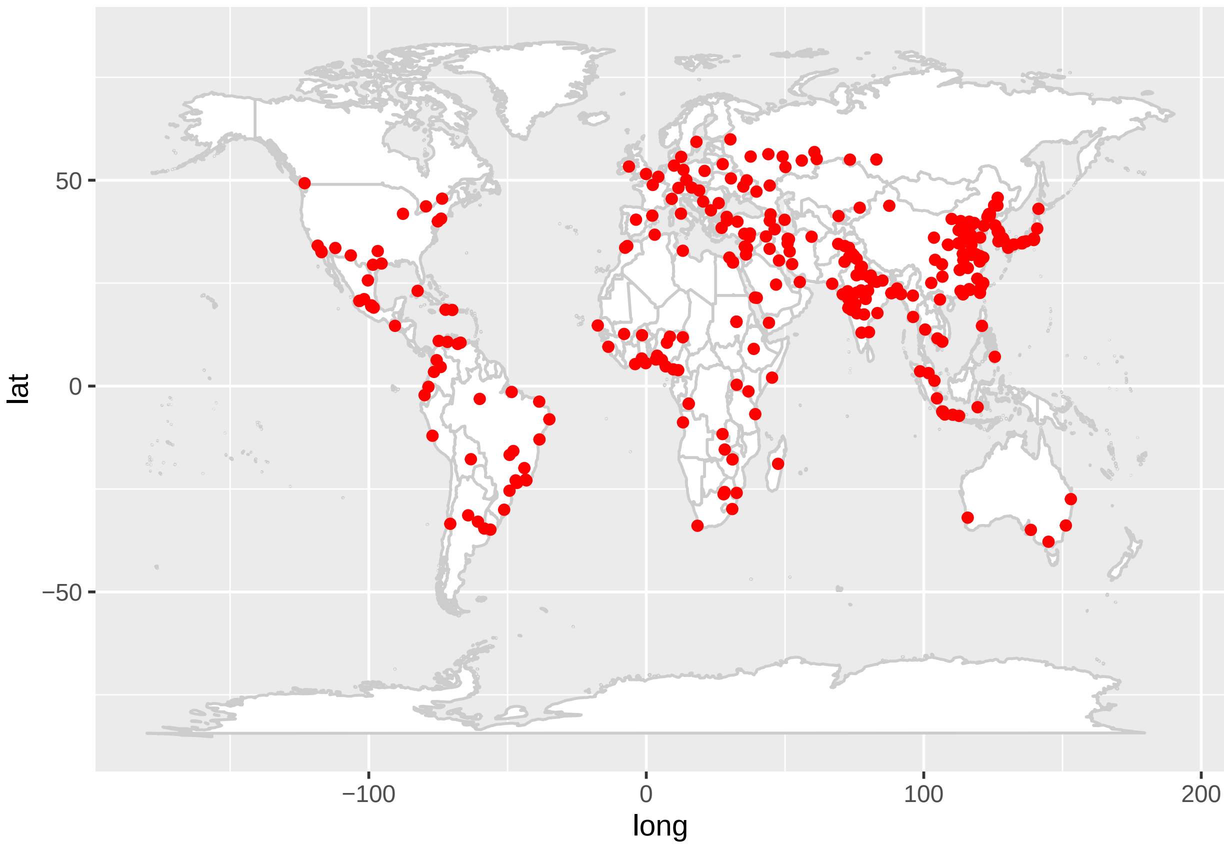
Q3 Mapping future climate (5, 5, 5 points)

(a) Plot all the cities in the dataset on a world map: that is, make a scatter plot of latitude (`lat` for short, the north-south coordinate) vs. longitude (`lon` for short, the east-west coordinate). Note that a simple way to make a blank world map in ggplot is

```
ggplot() + borders("world",fill="white",colour="gray80")
```

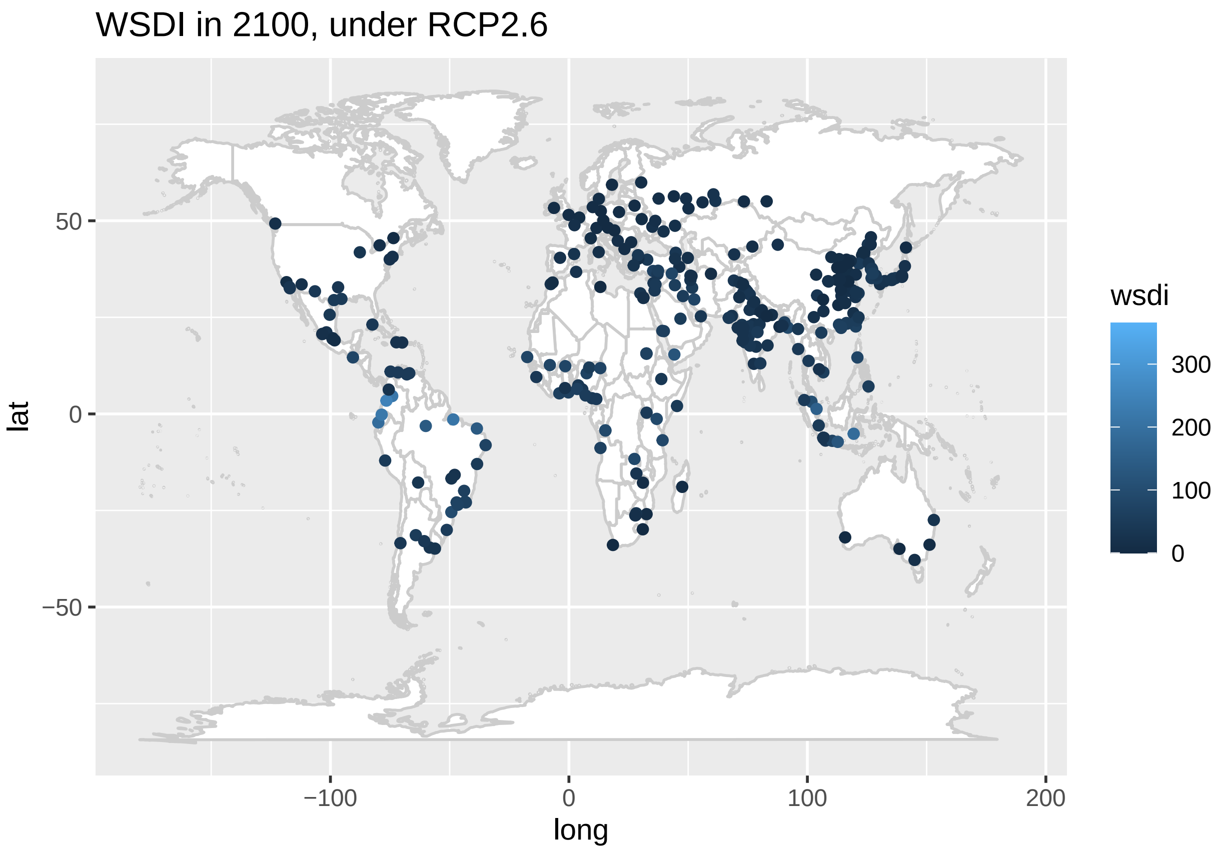
and you can add further plot layers from there.

```r
ggplot(r20mm_change) + borders("world",fill="white",colour="gray80") +
  geom_point(aes(lon,lat),color="red")
```

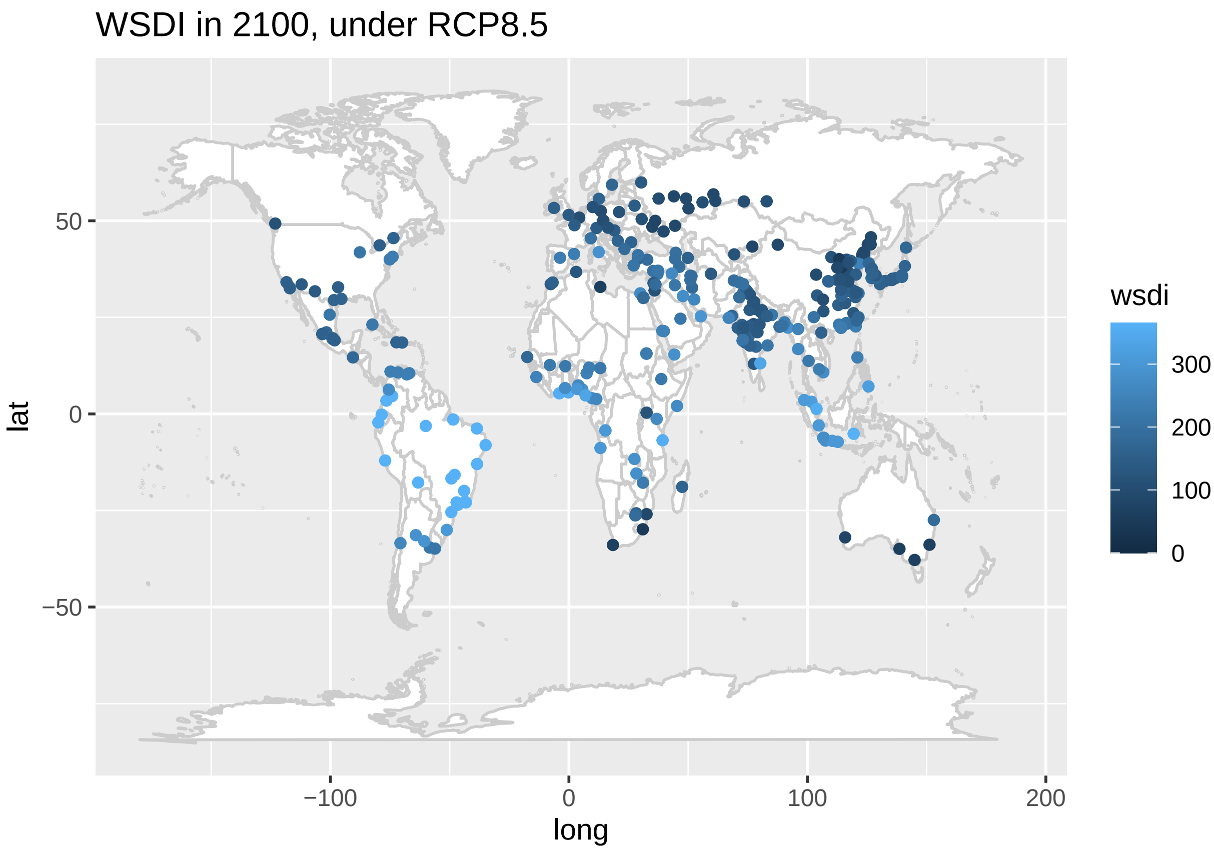


(b) Make two maps similar to the one in part (a), but in which each city's dot is coloured by the `wsdi` predicted for 2100 in RCP2.6 (first map) and in RCP8.5 (second map). Give each map an appropriate title. Set the color scales on the two plots to the same range of `wsdi` values, to make them easier to compare. In one or two sentences, explain what preliminary conclusions you draw.

```r
ex %>% filter(year==2100 & model=="GFDL" & scenario=="RCP2.6") %>%
  ggplot() +
  borders("world",fill="white",colour="gray80") +
  geom_point(aes(lon,lat,colour=wsdi)) +
  labs(title="WSDI in 2100, under RCP2.6") +
  scale_colour_continuous(limits=c(0,365))
```




```
ex %>% filter(year==2100 & model=="GFDL" & scenario=="RCP8.5") %>%
  ggplot() +
  borders("world",fill="white",colour="gray80") +
  geom_point(aes(lon,lat,colour=wsdi)) +
  labs(title="WSDI in 2100, under RCP8.5") +
  scale_colour_continuous(limits=c(0,365))
```




```
# The GFDL model predicts that WSDI will remain fairly low in most of the world in
# the optimistic RCP2.6 scenario, with the exception of northern South America. But
# in RCP8.5, WSDI is predicted to increase to hundreds of days per year.
```

(c) Because of how ‘wsdi’ is defined it is always low in the historical scenario: it’s a measure of future change. In contrast, ‘txx’ is simply a temperature, with widely varying values even in the historical period. Make a map similar to those in (b) showing the *change* in `txx` between 2000 (historical scenario) and 2100 (RCP8.5), for the GFDL model.

Here is a recommendation of how to do it with tidyverse (although you could also do it with a loop and no tidyverse):

- Extract only the rows you need into a simplified data frame
- Remove extraneous columns, and reshape to put the 2000 and 2100 values for each city on the same row (this is basically a kind of untidying!)
- Add a new column in which you calculate the temperature change
- Make the map as in part (b).

```
ex %>% filter(model=="GFDL" &
                scenario %in% c("historical","RCP8.5") &
                year %in% c(2000,2100)) %>%
  select(c("year","city","txx","lon","lat")) %>%
  pivot_wider(names_from="year", values_from="txx", names_prefix="txx_") %>%
  mutate(diff_txx= txx_2100-txx_2000) %>%
  ggplot() +
  borders("world",fill="white",colour="gray80") +
  geom_point(aes(lon,lat,colour=diff_txx)) +
```

```
scale_colour_gradient(low="yellow",high="red") +
labs(title="Change in annual high temp. 2000-2100 (RCP8.5)",
     color="Change (degC)",x="",y="")
```