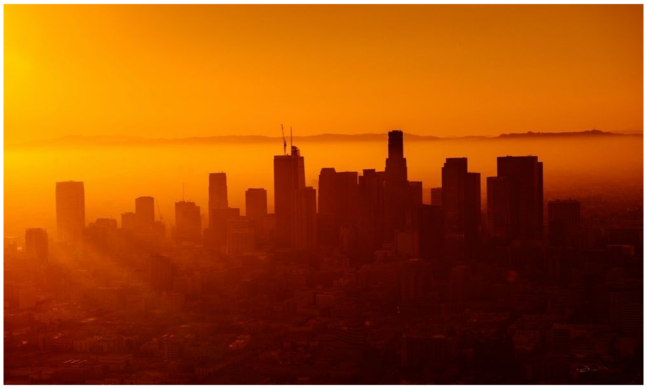
# BIOS0052: Human And Ecosystem Health In A Changing World

Week 2 practical session - Solutions

## Mapping temperature and heat exposure in the Southwestern USA



In today's practical we will explore, visualise and analyse temperature data from the Southwestern USA. We will learn how we can use climate datasets to understand geographic variation in health hazards - here, extreme heat exposure - both in the present day and under future scenarios. We will be using spatial data and GIS tools in R via the packages *"terra"* and *"sf"* - if you are taking *BIOS0050: Data Science for Ecology, Climate Change & Health*, this should be familiar from your practical sessions. If not, please see Moodle for a brief introduction to spatial data types.

Our aim for today's session is to understand the risks of extreme heat exposure in the Southwestern USA, with a focus on California and Nevada. Extreme heat and heatwaves significantly impact morbidity and mortality, both directly via heat stress, and indirectly by exacerbating other conditions (e.g. cardiovascular and respiratory disease). First, we will analyse trends in extreme heat in the present-day using daily temperature data from the ERA5-Land reanalysis dataset. Next, we will explore how these risks could change in future decades using future climate scenario data from several general circulation models (GCMs) and RCP-SSP scenarios. This week's lecture introduced these concepts in some depth, so check out the materials on Moodle if you need to catch up.

This workbook is structured as a series of code snippets with short exercises interspersed. The solutions for the short exercises are available in the Rmarkdown script in the GitHub folder, but *please avoid looking at these* until you have tried to solve them!

At the end, there are some **longer extension exercises** to allow you to apply your knowledge and prepare you for the assessments. The solutions for these will be uploaded to Moodle next week.

All the data you'll need for the workshop are in the GitHub, in the "Week2-Possible-Futures" folder. Please download the whole folder using download-directory.github.io. Set this folder as your working directory, then all the materials you will need are contained within the "data" subfolder.

```r
# package dependencies
# use the "install.packages()" command if not already installed
library(terra); library(dplyr); library(magrittr); library(ggplot2); library(sf)
library(rstudioapi); library(tidyr)

# automatically set working directory
# (or if this doesn't work,
# manually set your working directory to the folder "Week2-Possible-Futures")
PATH = dirname(rstudioapi::getSourceEditorContext()$path)
setwd(PATH)
```

## Our research questions

The Southwestern USA is a warm, ecologically diverse and highly populated region that encompasses some of the hottest places on Earth (including the Mojave Desert and Death Valley), and is suscepible to many climate hazards including heatwaves, droughts and wildfires. We can start our analyses with a broad question: **how frequently are populations in this region exposed to extreme temperatures?**, and **how might this change in a hotter future world?**

An important question we need to ask is, how will we define our exposure, “extreme temperature”? Many studies use daily mean temperature, as this captures the average conditions experienced over the course of the day (one example is this study). Another option, which we will use today, is to look at **daily minimum temperature** (**Tmin**, i.e. night-time temperature). Hot nights are strongly associated with increased morbidity and mortality, as intense heat during the night does not give the body the opportunity to cool down and recover from daytime thermal stress. So let’s examine the frequency of hot nights, how they vary between years, and what they could look like in the future.

## Reading and visualising temperature data from ERA5-Land

For the first part of today’s session we will use daily minimum temperature (*Tmin*) data derived from the ERA5-Land reanalysis dataset. Climate reanalysis is a cutting-edge approach to reconstructing the historical climate with high accuracy, by combining an ensemble of climate models (GCMs) with observational data from thousands of weather stations. The *“era5-land”* subfolder in the *“data”* folder contains data on Tmin for our study region. These are formatted as **rasters** - i.e. spatial grids with a coordinate reference system, where each grid cell has a temperature value. We can use these to visualise and map exposure to hot nights across the study area, as well as to look at the time-series of Tmin for specific locations.

First, we’ll start by looking at the mean nightly temperature across the region. The raster *“tmin_mean_cali.tif”* contains the mean Tmin in degrees Celcius, calculated across all days between 2000 and 2023. This will allow us to map the average conditions (the climatology) of different areas.

```r
# the "terra" package is the core package for working with raster data in R
# read in the raster using the "rast" function
tmin = terra::rast("data/era5-land/tmin_mean_cali.tif")

# look at its attributes - what do you see?
# how many grid cells are in this raster?
tmin
```

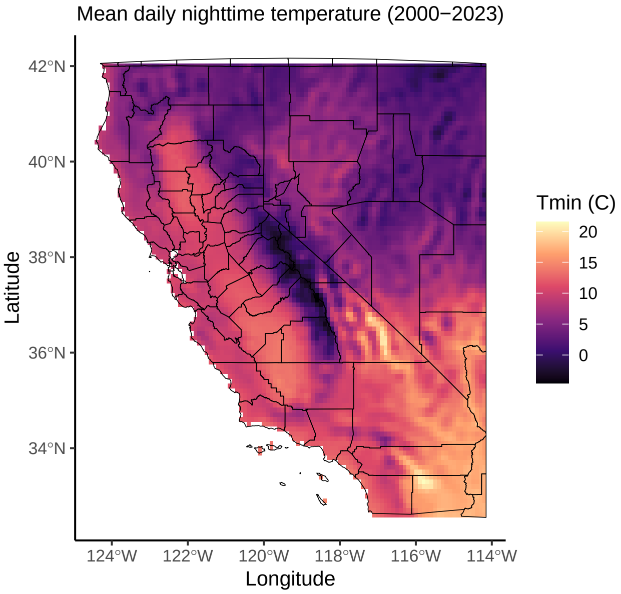
- **Q1.** You can quickly plot a raster using the “plot” function. Try doing this - what do you notice about the geographic distribution of Tmin?

An alternative and much more flexible way to visualise rasters is to use the ggplot package.

```r
# convert the raster to a dataframe with xy coordinates
# then plot a "geom_raster" in ggplot
tmin_df = as.data.frame(tmin, xy=TRUE)
plot1 = ggplot(tmin_df) +
  geom_raster(aes(x, y, fill=mean))

# you can add additional customisations to an existing ggplot object
# to improve the visualisation let's tweak the theme
# and add a better colour scale
plot1 = plot1 +
  scale_fill_viridis_c(option="magma", name="Tmin (C)") +
  theme_classic() +
  xlab("Longitude") + ylab("Latitude") +
  ggtitle("Mean daily nighttime temperature (2000-2023)") +
  theme(plot.title=element_text(size=11, hjust=0.5))

# take a look - what's changed?
```

When working with health datasets we often use administrative/political unit divisions, as many health, socioeconomic and census metrics are aggregated and reported by admin unit. So let's add administrative units to this map. These are stored as polygons in a shapefile, in the *shapefiles* subfolder; we need to read these in as a simple features object using the "sf" package.

```r
# read in USA counties data for the region
shp = sf::st_read("./data/shapefiles/usa_counties_sw.shp")

# take a look at "shp"
# what type of spatial data is this, and what information do we have?
# try plotting it - what do you see?
shp

# ggplot2 provides excellent functionality for working with sf objects
# let's add these admin boundaries to "plot1" using the geom_sf function
plot1 = plot1 +
  geom_sf(data=shp, fill=NA, color="black")

# take a look - are temperatures fairly similar geographically, or fairly variable?
# why might this be?
plot1
```



```r
# we can also easily save ggplot objects to our working directory using "ggsave"
ggsave(plot1, filename="california_Tmin.jpg", device="jpg", units="in", width=5, height=6, dpi=600)
```

**Do you think this map of the 24-year mean is useful for estimating extreme heat exposure risk? Why?**

Calculating the average climatology over a long period of time masks significant climate variability, and extreme conditions are - by their nature - transient events. What we really need is data at a much finer temporal granularity to estimate how often people might be exposed to extreme heat conditions.

- **Q2**: In the data folder is a raster called *"tmin_daily_cali_20152023.tif"*, which contains daily Tmin for the study area over a longer period of time. Read this file into R using the "rast" function, and store it as a new object called *"tmin_daily"*. Examine this object - what is different about this, compared to the first raster?

```r
# read
tmin_daily = terra::rast("data/era5-land/tmin_daily_cali_20152023.tif")

# examine - this contains multiple layers (one per day)
tmin_daily
```

- **Q3**: Use the *"names()"* and *"nlyr()"* functions to examine the temporal scope of the data. How many days of Tmin data do we have in total? What range of dates do they cover?

```r
# 3287 layers! (1 per day)
terra::nlyr(tmin_daily)

# dates run from 1st Jan 2015 until 31 Dec 2023
names(tmin_daily)[1] # start date
names(tmin_daily)[nlyr(tmin_daily)] # end date
```

Hopefully you will have noticed that "tmin_daily" contains multiple raster layers! These are all stacked up like a sandwich on top of each other. This is a critical innovation of geographic innovation systems (GIS) -

multiple sets of data can be overlaid, provided they share the same coordinate reference system (CRS). By looking at the same grid cell in each layer, we can access the same variable at different time points - this gives us a time series of daily data to examine.

```
# working with stacked rasters
# we can use subset functions to extract and look at individual layers or sets of layers
tmin_daily[[ 1 ]]
tmin_daily[[10:20]]

# we can plot a subset to see more clearly what we have
plot(tmin_daily[[5:10]])

# the "names" function gives the name of each layer
# what information does the name contain that we might need?
names(tmin_daily)
```

We have over 3,200 layers, which is far too many to just plot everything. Instead we can use these data to extract a time-series of daily Tmin values for one specific location, then analyse what is happening at that location. (This is also useful as a demonstration for the huge quantity of information we can derive from a raster dataset like this). Let's focus on **San Diego** - the second most populous city in California.

To extract information for San Diego we need its geographic location - its central coordinates are *-117.126 (longitude), 32.728 (latitude)*. To use this coordinate within R's GIS functionality, we need to convert this to a simple features point object.

```
# create a data frame of containing the XY coordinates
sd_loc = data.frame(Location = "San Diego", x = -117.126, y = 32.728)

# convert into an sf object and look at it - now it has a geometry field
sd_loc = sf::st_as_sf(sd_loc, coords = c("x", "y"))
sd_loc

# ensure the CRS is harmonised with the raster data
sf::st_crs(sd_loc) = terra::crs(tmin_daily)
```

- **Q4**: Add a point location for San Diego to your map ('plot1' of the study area). Where is it, and what is the average climate like? *(Hint: you can use geom_sf() to plot any sf objects in ggplot)*

```
# adding a point for San Diego to your existing plot
# use geom_sf to add a point
plot1 = plot1 +
  geom_sf(data=sd_loc, color="blue", fill="yellow", pch=21, size=3)
plot1
```

- **Q5**: Now that we have San Diego's location, we can use terra's "extract" function to extract the values of a raster at that location. What was the Tmin in San Diego on 5th January 2015? *(Hint: R's help() command will provide information about how to use specific function. Remember, you can use "[[]]" to select a specific layer of the raster stack)*

```
# examine the names
# we can see that the fifth layer is the 5th Jan 2015
names(tmin_daily)[1:10]
```

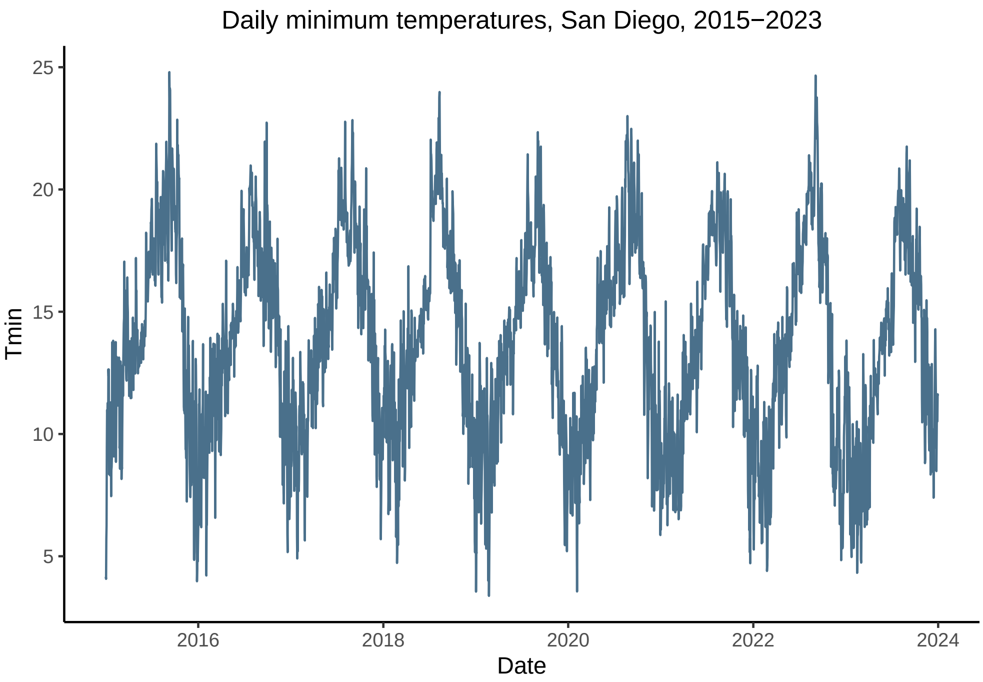
```
# extract from this layer
# minimum temperature was 9.43 degrees celcius
tmin_daily_5jan = tmin_daily[[5]]
terra::extract(tmin_daily_5jan, sd_loc, ID=FALSE)
```

For this approach to be useful, we need to extract the Tmin for all days within the time series, and then convert this into a data frame we can use for visualisation and analysis. Work through the following code block to do this, ensuring you understand each line (if you don't, please ask!).

```
# run terra::extract on the entire raster stack
# outputs a dataframe with columns as layers, and rows as locations
# a bit difficult to see what's going on as it's a very wide 1-row dataframe!
sd_daily = terra::extract(tmin_daily, sd_loc, ID=FALSE)

# convert to a longform dataframe using pivot_longer
sd_daily = sd_daily %>%
  tidyr::pivot_longer(cols = everything(), names_to = "date", values_to = "tmin")

# what is in this dataframe?
head(sd_daily)

# to work effectively with dates/times in R, we need to ensure they are encoded in the correct format
# check the current class of the "date" column - stored as a character
class(sd_daily$date)

# convert to a date format
# remove the "X" from the start of the string, then format as a Date
# now R can recognise the sequential nature of the dates for plotting and analysis
sd_daily$date = substr(sd_daily$date, 2, 15)
sd_daily$date = as.Date(sd_daily$date, format="%Y_%m_%d")
class(sd_daily$date)
```

- **Q6**: Use "sd_daily" to plot a graph of daily Tmin for San Diego over the study period. What patterns do you notice in the data? How much variability do you see within and between years?

```
# line graph
sd_plot = ggplot() +
  geom_line(data=sd_daily, aes(date, tmin), color="skyblue4") +
  theme_classic() +
  xlab("Date") + ylab("Tmin") +
  ggtitle("Daily minimum temperatures, San Diego, 2015-2023") +
  theme(plot.title=element_text(size=12, hjust=0.5))
sd_plot
```



## Estimating the historical frequency of night-time extreme heat

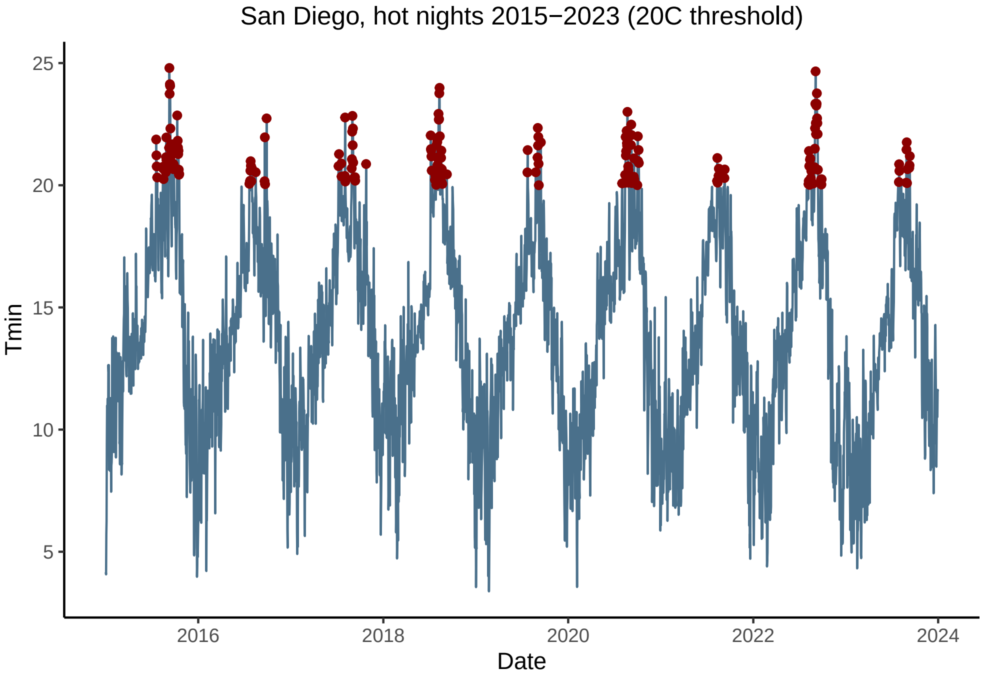
This daily time-series of night-time temperatures provides the basis to do some simple climate epidemiology. We can use it to ask, how many hot nights is the population of San Diego exposed to? To do this, we first have to quantitatively define a **"hot night"**. Let's take a simple definition that is widely used in European studies, which defines a "hot night" as a night when the temperature does not fall below 20°C.

- **Q7**: Use R's subset functionality to create a new dataframe called "hot_nights1" that contains only the dates where Tmin was greater than 20°C. How many nights met this criteria between 2015 and 2023?

```r
# subset to identify dates meeting this criteria
hot_nights1 = sd_daily[ sd_daily$tmin > 20, ]

# how many? 183 over 9 years
nrow(hot_nights1)
```

Now we can **summarise and visualise** these data to understand the frequency of hot nights during the last decade.

```r
# add points per hot night to our time series graph
# is there a pattern to when they occur?
sd_plot = sd_daily %>%
  ggplot() +
  geom_line(aes(date, tmin), color="skyblue4") +
  theme_classic() +
  xlab("Date") + ylab("Tmin") +
  ggtitle("San Diego, hot nights 2015-2023 (20C threshold)") +
  theme(plot.title=element_text(size=12, hjust=0.5)) +
  geom_point(data = hot_nights1, aes(date, tmin), color="darkred")
sd_plot
```



```
# what about differences between years?
# use lubridate's "year" function to create an additional column for year
hot_nights1$year = lubridate::year(hot_nights1$date)

# calculate the number of hot nights per year
# what do you noice?
hotnights_annual1 = hot_nights1 %>%
  dplyr::group_by(year) %>%
  dplyr::summarise(n_hot_nights = length(date))
```

## Defining a locally-specific extreme heat threshold

We now have an estimate of the frequency of hot nights, which we can use to start understanding risks. However, a large body of evidence shows that **heat stress thresholds are context-dependent**! The temperature at which the body begins to experience morbidity and mortality from thermal stress depends, in part, on acclimation to local climatic conditions. In hotter regions, people are used to hotter temperatures, and their behaviour and design of housing, infrastructure and built environments reflect this.

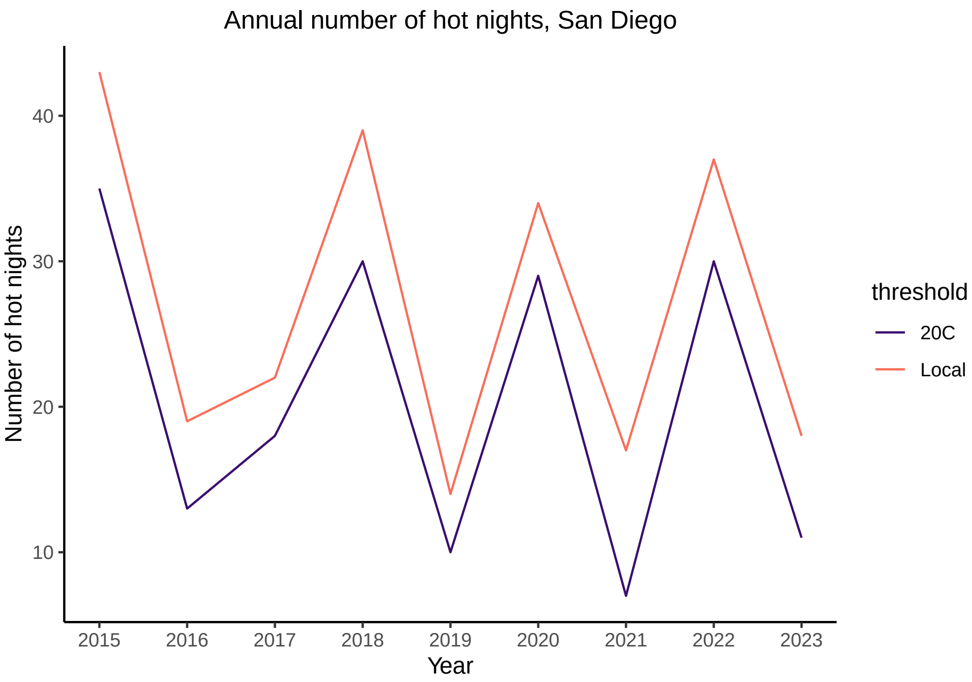
So it may be more appropriate to **instead define our threshold based on the normal historical temperature range** for San Diego. One common definition is that a "hot night" is a night whose temperature exceeds the 95th percentile of night-time temperatures across a historical reference period (see this paper for an example). The *"data/era-land"* folder already has a raster of the 95th percentile of Tmin - called *"tmin_upper95_cali.tif"* - pre-calculated across the period 2000-2023.

- **Q8:** Read in this raster and extract the 95th percentile value for San Diego. Re-run your analyses from above, using this value instead of the 20°C threshold for defining a hot night. Do you estimate more or fewer hot nights using the locally-defined threshold? Why?

```
# read in raster of 95 percentile
tmin_upper = terra::rast("./data/era5-land/tmin_upper95_cali.tif")

# extract at san diego location
thresh = terra::extract(tmin_upper, sd_loc, ID=FALSE)

# apply the same threshold, visualise and calculate
hot_nights2 = sd_daily[ sd_daily$tmin > as.numeric(thresh), ]
nrow(hot_nights2) # 243 compared to 183 - over 30% increase!

# plot
sd_plot = ggplot() +
  geom_line(data=sd_daily, aes(date, tmin), color="skyblue4") +
  theme_classic() +
  xlab("Date") + ylab("Tmin") +
  ggtitle("San Diego, hot nights 2015-2023 (local 95th percentile threshold)") +
  theme(plot.title=element_text(size=12, hjust=0.5)) +
  geom_point(data = hot_nights2, aes(date, tmin), color="darkred")
```

Now we can visualise the interannual trend in hot nights for both thresholds in parallel:

```
# calculate annual number of hot nights
hot_nights2$year = lubridate::year(hot_nights2$date)
hotnights_annual2 = hot_nights2 %>%
  dplyr::group_by(year) %>%
  dplyr::summarise(n_hot_nights = length(date))

# combine both thresholds data into the same dataframe
hotnights_annual1 = hotnights_annual1 %>% dplyr::mutate(threshold="20C")
hotnights_annual2 = hotnights_annual2 %>% dplyr::mutate(threshold="Local")
annual_combined = rbind(hotnights_annual1, hotnights_annual2)

# plot both on the same graph
annual_plot = ggplot() +
  geom_line(data = annual_combined, aes(year, n_hot_nights, color=threshold)) +
  theme_classic() +
  xlab("Year") + ylab("Number of hot nights") +
  ggtitle("Annual number of hot nights, San Diego") +
  theme(plot.title=element_text(size=12, hjust=0.5)) +
  scale_color_viridis_d(option="magma", begin=0.2, end=0.7) +
  scale_x_continuous(breaks=2015:2023, labels=2015:2023)
annual_plot
```



- **Q9**: Compare and contrast the results you observe using different thresholds. Which threshold do you think is more accurate or appropriate, and why? Can you think of some applications in which either threshold method might be most useful?

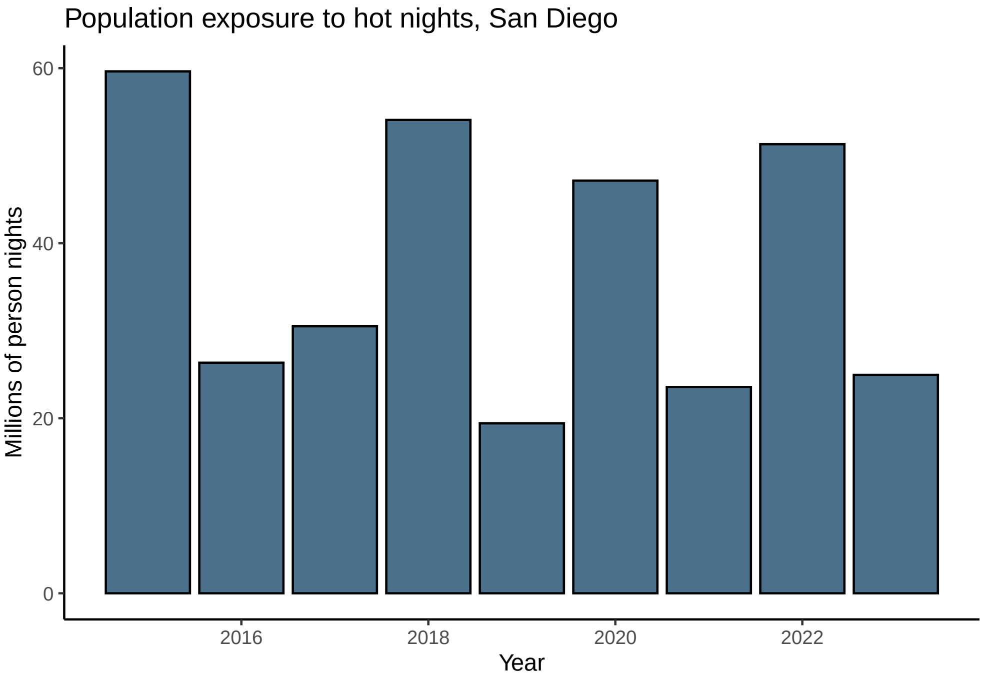
## Combining hazard and exposure to estimate population risk

So far, we have focused on estimating the annual climate **hazard**, i.e. the frequency of night-time temperatures exceeding some threshold. However, recall that **hazards only become realised risks if people are exposed (and vulnerable) to them**. We can define exposure as the *size of the population that experiences the hazard*, and use this to develop a crude estimate for the population at risk. The population of San Diego, as of the 2020 census, was 1,386,932 people. We can use this to calculate a better risk metric than simply the number of hot nights - we can calculate the **number of person-nights** of extreme heat per year, based on the local threshold.

```
# estimate person nights by multiplying the number of hot nights by population
hotnights_annual2$person_nights = hotnights_annual2$n_hot_nights * 1386932

# transform to millions of person nights for easier visualisation
hotnights_annual2$person_nights_mill = hotnights_annual2$person_nights/10^6
```

- **Q10**: Plot a graph visualising these risk estimates. What does this tell you about the magnitude of this potential health risk, and how consistent it is year-on-year? What additional data or information do you think we would need to improve our risk estimate?

```
# plot hazard * exposure annually
exposure_plot = ggplot() +
  geom_bar(data=hotnights_annual2, aes(year, person_nights_mill), stat="identity", fill="skyblue4", col
  xlab("Year") + ylab("Millions of person nights") +
  ggtitle("Population exposure to hot nights, San Diego") +
  theme(plot.title=element_text()) +
  theme_classic()
exposure_plot
```



## Understanding the future potential for extreme heat exposure under climate change

Climate change is already having significant health and ecological impacts in the Southwestern USA, including droughts, heatwaves, and increasingly severe wildfires. The recent climate data we have analysed - covering 2015 to 2023 - likely already contains the fingerprint of anthropogenic climate change! But if we were a decision-maker, for example working on urban planning, land use, or health systems preparedness, it could be very useful to understand the potential magnitude of extreme heat risk in the next few decades. This is where **scenario data from climate models (GCMs)** can be very useful. In this section, we will explore how we can use future climate model outputs (from the Coupled Model Intercomparison Project; CMIP6) to estimate changes to the climate and their potential consequences for health.

The following table provides data on the locations and populations of several cities in California and Nevada. We can use these to explore future changes in heat in these different places.

```
# cities data for later analyses
cities = data.frame(
  City = c("San Diego", "San Francisco", "Las Vegas", "Palm Springs"),
  Longitude = c(-117.126, -122.422, -115.147, -116.527),
  Latitude = c(32.728, 37.768, 36.166, 33.836),
  Population_2020 = c(1386932, 873965, 641903, 44575)
)
```

| City | Longitude | Latitude | Population_2020 |
|---|---|---|---|
| San Diego | -117.126 | 32.728 | 1386932 |
| San Francisco | -122.422 | 37.768 | 873965 |
| Las Vegas | -115.147 | 36.166 | 641903 |
| Palm Springs | -116.527 | 33.836 | 44575 |

In the last section we used daily temperature data to calculate the frequency of hot nights, across different thresholds. Future projections using climate models are usually focused on estimating changes in the *climatology* - average conditions over time - since precisely predicting daily weather conditions so far in the future is (currently) not possible. So rather than daily temperature, we will instead use data on the **average Tmin in the month of July (hottest summer month)**, calculated over a multi-year reference period. This is a useful measure of average summer night-time temperature, and therefore a reasonable indicator for how hot night frequency may change over time.

The folder *“data/cmip6-chelsa”* contains three raster files, containing climatology data on July average Tmin for a **present day reference period** (1981-2010), and two future epochs, **2041-2070** (medium-term) and **2071-2100** (long-term). These data were accessed from the CHELSA (Climatologies at High Resolution for the Earth’s Land Surface Area) database, which has developed statistically-downscaled present-day and future climate data products for use in impact research.

- **Q11**: Use terra’s “rast” function to read in the **present day** raster of July average Tmin. Visualise the raster, and add points for the locations of the cities, if you like. How does this compare to the mean Tmin map that you developed from the ERA5-Land dataset?

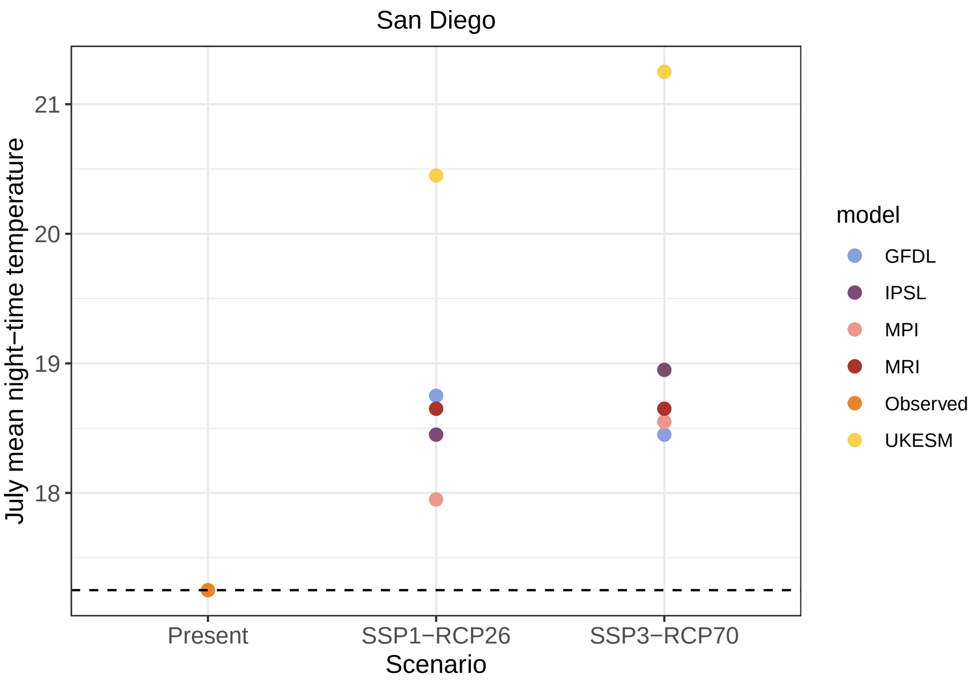
```r
# mean daily nighttime temperature in the present-day for July (1981-2010)
gc = terra::rast("data/cmip6_chelsa/chelsa_tmin_july_present_cali.tif")

# make data frame for plotting and change column title
gc_df = as.data.frame(gc, xy=TRUE) %>%
  dplyr::rename("tmin_july"=`tmin_present_1981-2020`)

map_present = ggplot() +
  geom_raster(data=gc_df, aes(x, y, fill=tmin_july)) +
  scale_fill_viridis_c(option="magma", name="Tmin (C)") +
  theme_classic() +
  xlab("Longitude") + ylab("Latitude") +
  ggtitle("Mean daily minimum temperature\n(July, 1981-2010, CHELSA dataset)") +
  theme(plot.title=element_text(size=11, hjust=0.5)) +
  geom_point(data=cities, aes(Longitude, Latitude), color="blue", fill="yellow", pch=21, size=3)
```

Now let’s compare the future scenario predictions to the present-day data. For trend analyses, future changes are always defined relative to some baseline period. In this example we will compare the July average nighttime temperature in the period 1981-2010, to the same metric in the future, predicted using climate models. Using a consistent data source for both present-day and future data is important: it ensures that the data have all undergone the same pre-processing steps and bias correction process (i.e. rescaling the GCM data to ensure they are directly comparable to the present day data). *Schoeman et al. 2023* is a very useful reference to understand this in more depth.

The raster files for the future each contain 10 layers: these correspond to predictions from **five different climate GCMs** (GFDL; IPSL; MPI; MRI; UKESM) under **two future socioeconomic and emissions scenarios** (SSP1-RCP2.6 and SSP3-RCP7.0).

Work through the following code block, which extracts and compares future and present-day climate data for San Diego. Check the outputs line-by-line to ensure you understand what it is doing, and ask if you have any questions!

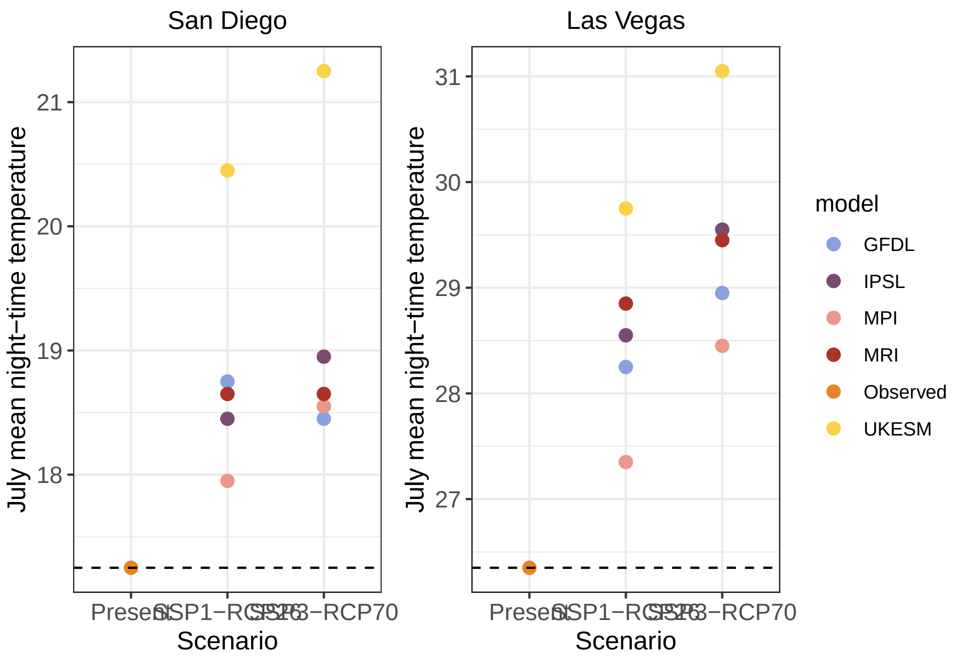
```r
# read in rasters
gc_2070 = terra::rast("data/cmip6_chelsa/chelsa_tmin_july_2070_cali.tif")

# names of each layer contain information on:
# climate metric (tmin)
# rcp-ssp scenario (emissions-socioeconomic)
# gcm (model used to project future climate)
# final year of climatology (here, 2070)
names(gc_2070)

# create a coordinates object for a specific city
sd = cities %>%
  dplyr::filter(City == "San Diego") %>%
  sf::st_as_sf(coords = c("Longitude", "Latitude"))

# extract july tmin for the present day reference period
# creating columns for "year", "scenario" and model
tmin_pres = terra::extract(gc, sd, ID = FALSE) %>%
  tidyr::pivot_longer(cols = everything(), names_to = "layer", values_to = "tmin") %>%
  dplyr::mutate(year = "2010", scenario = "Present", model = "Observed")

# extract the temperatures in the future
tmin_2070 = terra::extract(gc_2070, sd, ID = FALSE) %>%
  tidyr::pivot_longer(cols = everything(), names_to = "layer", values_to = "tmin") %>%
  dplyr::mutate(year = "2070")

# extract the names of the scenario and gcm from the raster layer name
# this code splits the names by underscores
# then subsets to the specific sections we want
tmin_2070$scenario = unlist(lapply(strsplit(tmin_2070$layer, "_"), "[", 2))
tmin_2070$model = unlist(lapply(strsplit(tmin_2070$layer, "_"), "[", 3))

# look at future data
head(tmin_2070, 10)

# combine present and future data in one data frame
tmin_change = rbind(tmin_pres, tmin_2070)

# plot the changes, colouring points by different GCMs
# includes a dashed horizontal line to reflect the present day baseline
futures_sd = tmin_change %>%
  ggplot() +
  geom_point(aes(scenario, tmin, color=model), size=2.5) +
  geom_hline(yintercept = tmin_change$tmin[ tmin_change$scenario == "Present"], lty=2) +
  theme_bw() +
  ggtitle(sd$City) +
  MetBrewer::scale_colour_met_d(name="Archambault") +
  ylab("July mean night-time temperature") +
  theme(plot.title = element_text(size=12, hjust=0.5),
        axis.text = element_text(size=11),
        axis.title = element_text(size=11.5))+
  xlab("Scenario")
futures_sd
```



- **Q12**: Examine and interpret the results of this graph. What is our overall finding about how July night-time temperatures are expected to change in the future in San Diego? Does one future emissions scenario look riskier than the other? Think about and describe the uncertainty in these predictions. How much is due to structural uncertainty (i.e. between models)? What about uncertainty between emissions scenarios?

- **Q13** Modify the code block above to examine the future heat prospects for Las Vegas, and compare these to your results for San Diego. How do the two cities compare for present-day and future risks? Does the uncertainty differ between the two cities, and if so, how? *(Hint: remember, you can save your graphs using ggsave() to compare them, or you can create a multipanel plot using R's 'patchwork' or 'gridExtra' packages)*

```
# create coordinates object for Las Vegas
# note that we then use this "lv" object in subsequent code
lv = cities %>%
  dplyr::filter(City == "Las Vegas") %>%
  sf::st_as_sf(coords = c("Longitude", "Latitude"))

# extract july tmin for the present day reference period
# creating columns for "year", "scenario" and model
tmin_pres = terra::extract(gc, lv, ID = FALSE) %>%
  tidyr::pivot_longer(cols = everything(), names_to = "layer", values_to = "tmin") %>%
  dplyr::mutate(year = "2010", scenario = "Present", model = "Observed")

# extract the temperatures in the future
tmin_2070 = terra::extract(gc_2070, lv, ID = FALSE) %>%
  tidyr::pivot_longer(cols = everything(), names_to = "layer", values_to = "tmin") %>%
  dplyr::mutate(year = "2070")

# extract the names of the scenario and gcm from the raster layer name
# this code splits the names by underscores
# then subsets to the specific sections we want
tmin_2070$scenario = unlist(lapply(strsplit(tmin_2070$layer, "_"), "[", 2))
tmin_2070$model = unlist(lapply(strsplit(tmin_2070$layer, "_"), "[", 3))

# combine present and future data in one data frame
tmin_change = rbind(tmin_pres, tmin_2070)

# plot the changes, colouring points by different GCMs
# includes a dashed horizontal line to reflect the present day baseline
futures_lv = tmin_change %>%
  ggplot() +
  geom_point(aes(scenario, tmin, color=model), size=2.5) +
  geom_hline(yintercept = tmin_change$tmin[ tmin_change$scenario == "Present"], lty=2) +
  theme_bw() +
  ggtitle(lv$City) +
  MetBrewer::scale_colour_met_d(name="Archambault") +
  ylab("July mean night-time temperature") +
  theme(plot.title = element_text(size=12, hjust=0.5),
        axis.text = element_text(size=11),
        axis.title = element_text(size=11.5))+
  xlab("Scenario")

# combined plot
library(patchwork)
futures_sd + futures_lv + patchwork::plot_layout(guides = "collect")
```



# Extension exercises

The following extension exercises provide an opportunity further expand your analyses of both the daily ERA5-Land data, and the future scenarios data. **I strongly recommend working through them**, either during the practical session if there is time, or during your own time. The solutions will be uploaded to Moodle next week, with an opportunity to discuss them in class.

## ERA5-Land daily temperature data

Use the "cities" dataframe provided above, which contains data on locations and populations for San Diego, Las Vegas, Palm Springs and San Francisco. Analyse historical hot night exposure across these four cities, using the ERA5-Land 2015-2023 data on daily Tmin. Consider the following questions.

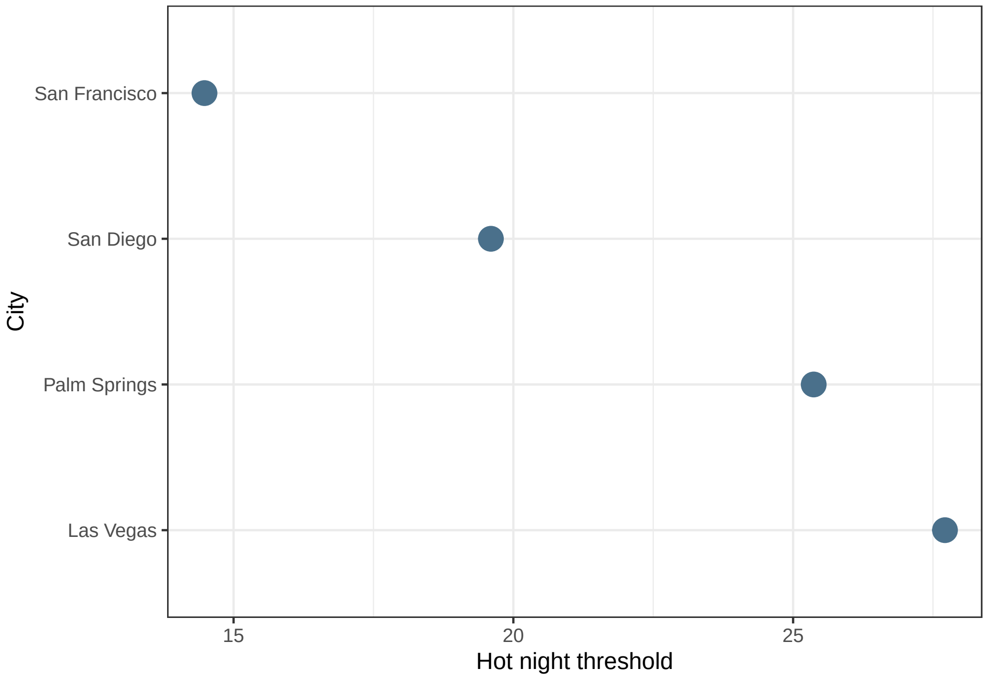
- Map the locations of all cities in relation to average Tmin. What do you notice about their different climatologies?

```r
plot1 = tmin %>%
  as.data.frame(xy=TRUE) %>%
  ggplot() +
  geom_raster(aes(x, y, fill=mean)) +
  scale_fill_viridis_c(option="magma", name="Tmin (C)") +
  theme_classic() +
  xlab("Longitude") + ylab("Latitude") +
  ggtitle("Mean daily nighttime temperature (2000-2023)") +
  theme(plot.title=element_text(size=11, hjust=0.5)) +
  geom_point(data=cities, aes(Longitude, Latitude), color="blue", fill="yellow", pch=21, size=3)
```

- Use the 95th percentile raster to estimate hot night temperature thresholds for each city. Compare them between cities. How different are they? Why?

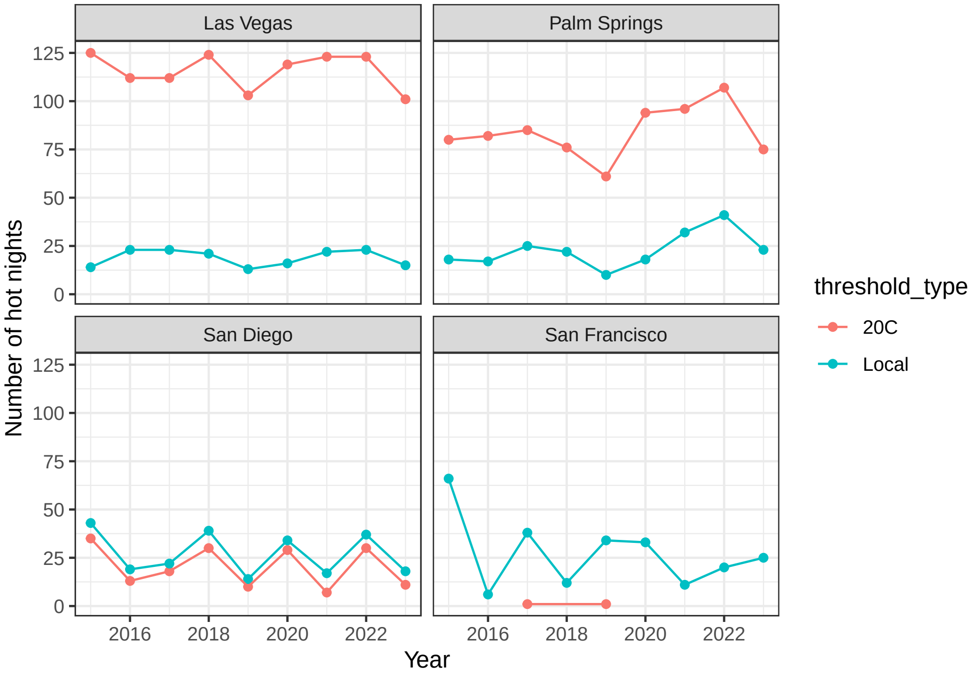
```r
# this is a loop to extract the same information for every city and combine it
# you probably approached this by copying the same code block several times and that is fine
# the below is just more effficient for coding space

# for saving everything
thresholds = data.frame()
dailies = data.frame()
hotnights1 = data.frame()
hotnights2 = data.frame()

# for each city (1 in 1:4)
for(i in 1:nrow(cities)){

  # get the info for city i and create sf object
  ct = cities[i, ]
  ct = ct %>% sf::st_as_sf(coords = c("Longitude", "Latitude"))

  # extract daily time series, format date
  sf::st_crs(ct) = terra::crs(tmin_daily)
  dailies_i = terra::extract(tmin_daily, ct, ID=FALSE) %>%
    tidyr::pivot_longer(cols = everything(), names_to = "date", values_to = "tmin")
  dailies_i$date = substr(dailies_i$date, 2, 15)
  dailies_i$date = as.Date(dailies_i$date, format="%Y_%m_%d")
  dailies = rbind(dailies, dailies_i %>% dplyr::mutate(City = ct$City))

  # calculate local hot nights threshold
  tmin_upper = terra::rast("./data/era5-land/tmin_upper95_cali.tif")
  thresh_i = terra::extract(tmin_upper, ct, ID=FALSE)$`q0.95`
  thresh_i = data.frame(City = ct$City, threshold = thresh_i)

  # add to data frame of thresholds
  thresholds = rbind(thresholds, thresh_i)

  # create dataframe and summarise hot nights based on 20C threshold
  hn1 = dailies_i %>%
    dplyr::filter(tmin > 20) %>%
    dplyr::mutate(year = lubridate::year(date))
  hn1 = as.data.frame(table(hn1$year)) %>%
    dplyr::mutate(City = ct$City)
  hn1$exposure = (hn1$Freq * ct$Population_2020) / 10^6
  hn1$threshold_type = "20C"

  # add to combined data frame
  hotnights1 = rbind(hotnights1, hn1)

  # create dataframe and summarise hot nights based on local threshold
  hn2 = dailies_i %>% dplyr::filter(tmin > thresh_i$threshold) %>%
    dplyr::mutate(year = lubridate::year(date))
  hn2 = as.data.frame(table(hn2$year)) %>%
    dplyr::mutate(City = ct$City)
  hn2$exposure = (hn2$Freq * ct$Population_2020) / 10^6
  hn2$threshold_type = "Local"

  # add to combined data frame
  hotnights2 = rbind(hotnights2, hn2)
}

# look at outputs
head(thresholds)
head(hotnights1)
head(hotnights2)
```

```r
# compare thresholds between cities
# very different depending on local climatology!
thresholds %>%
  ggplot() +
  geom_point(aes(City, threshold), size=5, color="skyblue4") +
  ylab("Hot night threshold") +
  coord_flip() +
  theme_bw()
```



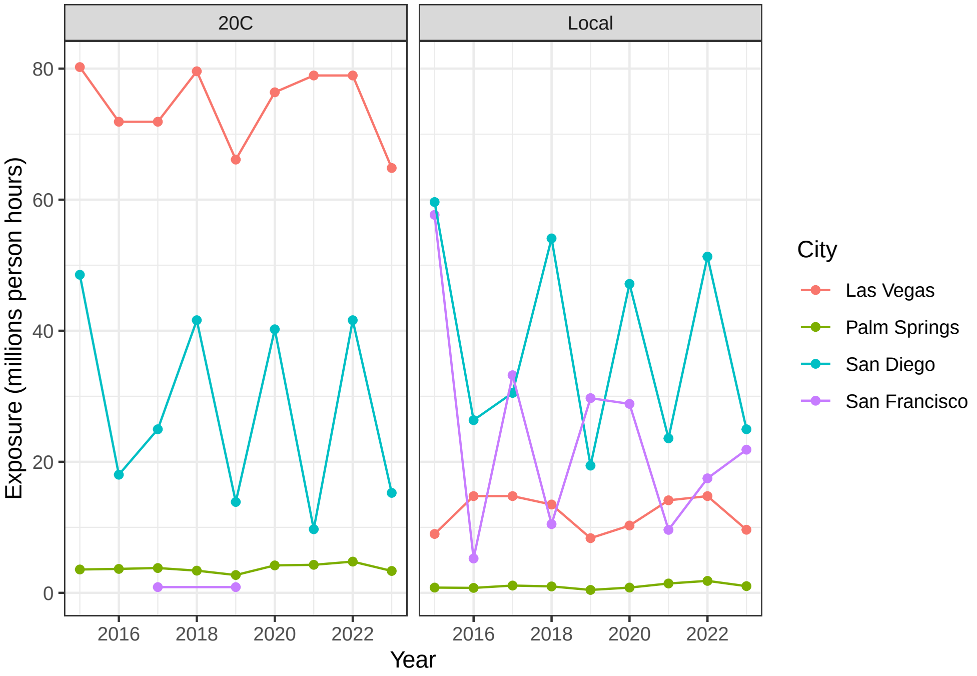
- Calculate the number of hot nights per year for each city, using both the 20°C threshold, and the locally-defined threshold. Compare the results between the different thresholds. Are the findings similar between all four cities? If not, why not?

```
# compare
# quite similar using local thresholds
# but very different using 20C thresholds
# Las Vegas and Palm Springs are much hotter on average
hotnights1 %>%
  rbind(hotnights2) %>%
  ggplot() +
  geom_point(aes(as.numeric(as.vector(Var1)), Freq, color=threshold_type)) +
  geom_line(aes(as.numeric(as.vector(Var1)), Freq, color=threshold_type)) +
  theme_bw() +
  xlab("Year") + ylab("Number of hot nights") +
  facet_wrap(~City)
```



- Use the population data to calculate the annual person-hours exposed to hot nights, using both thresholds, and visualise these for each city. What does this tell you about which city might be at greatest risk from extreme heat?

```
# very different conclusions based on both
# population exposure (exposure matters!) and threshold choice (model matters!)
# San Diego and San Francisco both have very large populations so high risk due to high exposure
# but if using a 20C threshold, potentially overpredicts risk in
# naturally hotter areas, and vice versa for cooler areas
hotnights1 %>%
  rbind(hotnights2) %>%
  ggplot() +
  geom_point(aes(as.numeric(as.vector(Var1)), exposure, color=City)) +
  geom_line(aes(as.numeric(as.vector(Var1)), exposure, color=City)) +
  theme_bw() +
  xlab("Year") + ylab("Exposure (millions person hours)") +
  facet_wrap(~threshold_type)
```



- What are the relative strengths and weaknesses of using a fixed 20°C threshold, versus a locally-defined threshold, for comparing extreme heat risks between different locations?

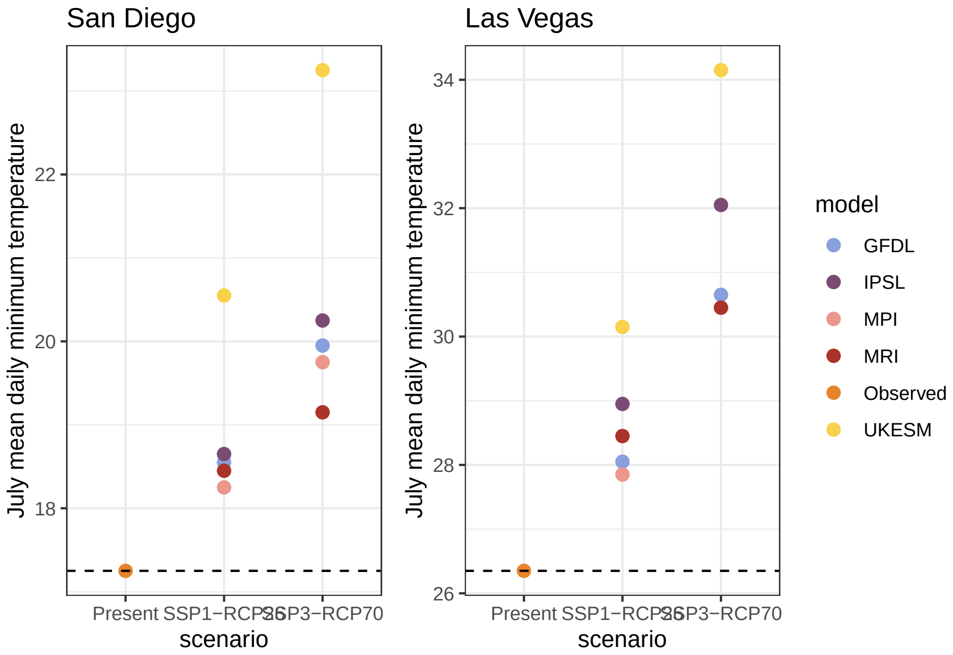
## Summer night-time temperature change by the end of this century

In the practical so far, we examined future changes in July night-time temperatures in the 2041-2070 period. However, we are only 16 years (!) away from the beginning of that period, and it's possible that the situation may change even more in the longer-term. The folder *"data/cmip6-chelsa"* also contains a raster stack of projected July night-time temperatures for the period 2071-2100, which includes the same scenarios, GCMs and naming conventions.

- Adapt the code from the practical to examine changes in July night-time temperatures between the present-day and the 2071-2100 period. Again, compare your results between San Diego and Las Vegas.
- What is the overall result, and how does this compare to the 2041-2070 period? Does one emissions scenario look clearly riskier than the other?
- Is the uncertainty in these projections similar, or higher, to the uncertainty in our medium-term analyses? How much is due to structural uncertainty? What about emissions scenarios uncertainty?

```
# present-day and future data
gc = terra::rast("data/cmip6_chelsa/chelsa_tmin_july_present_cali.tif")
gc_2100 = terra::rast("data/cmip6_chelsa/chelsa_tmin_july_2100_cali.tif")

# san diego
sd = cities %>% dplyr::filter(City == "San Diego") %>%
  sf::st_as_sf(coords = c("Longitude", "Latitude"))
tmin_pres = terra::extract(gc, sd, ID = FALSE) %>%
  tidyr::pivot_longer(cols = everything(), names_to = "layer", values_to = "tmin") %>%
  dplyr::mutate(year = "2010", scenario = "Present", model = "Observed")
```

```r
tmin_2100 = terra::extract(gc_2100, sd, ID = FALSE) %>%
  tidyr::pivot_longer(cols = everything(), names_to = "layer", values_to = "tmin") %>%
  dplyr::mutate(year = "2100") # adding columns for "year" and "scenario"
tmin_2100$scenario = unlist(lapply(strsplit(tmin_2100$layer, "_"), "[", 2))
tmin_2100$model = unlist(lapply(strsplit(tmin_2100$layer, "_"), "[", 3))
tmin_change = rbind(tmin_pres, tmin_2100)

plot_sandiego = tmin_change %>%
  ggplot() +
  geom_point(aes(scenario, tmin, color=model), size=2.5) +
  geom_hline(yintercept = tmin_change$tmin[ tmin_change$scenario == "Present"], lty=2) +
  theme_bw() +
  ggtitle(sd$City) +
  MetBrewer::scale_colour_met_d(name="Archambault") +
  ylab("July mean daily minimum temperature")

# las vegas
sd = cities %>% dplyr::filter(City == "Las Vegas") %>%
  sf::st_as_sf(coords = c("Longitude", "Latitude"))
tmin_pres = terra::extract(gc, sd, ID = FALSE) %>%
  tidyr::pivot_longer(cols = everything(), names_to = "layer", values_to = "tmin") %>%
  dplyr::mutate(year = "2010", scenario = "Present", model = "Observed")

tmin_2100 = terra::extract(gc_2100, sd, ID = FALSE) %>%
  tidyr::pivot_longer(cols = everything(), names_to = "layer", values_to = "tmin") %>%
  dplyr::mutate(year = "2100") # adding columns for "year" and "scenario"
tmin_2100$scenario = unlist(lapply(strsplit(tmin_2100$layer, "_"), "[", 2))
tmin_2100$model = unlist(lapply(strsplit(tmin_2100$layer, "_"), "[", 3))
tmin_change = rbind(tmin_pres, tmin_2100)

plot_lasvegas = tmin_change %>%
  ggplot() +
  geom_point(aes(scenario, tmin, color=model), size=2.5) +
  geom_hline(yintercept = tmin_change$tmin[ tmin_change$scenario == "Present"], lty=2) +
  theme_bw() +
  ggtitle(sd$City) +
  MetBrewer::scale_colour_met_d(name="Archambault") +
  ylab("July mean daily minimum temperature")

# combine and compare
library(patchwork)
plot_sandiego + plot_lasvegas + patchwork::plot_layout(guides = "collect")
```



```
# key findings
# much greater difference between SSP1-2.6 and SSP3-7.0 by 2070-2100
# scenario pathways have clearly diverged
# structural uncertainty (among models) is mostly lower than scenario uncertainty
# more confidence in how these pathways would diverge
# SSP3-7.0 clearly a more dangerous future in terms of extreme heat
# especially in Las Vegas
```