R Notebook

```
## Loading required package: ggplot2

## Loading required package: ggpubr

## Loading required package: magrittr

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: janitor

## Loading required package: stringr

## Loading required package: tidyverse

## -- Attaching packages ---------------------------------------------------------------------------- t

## v tibble  1.4.2     v readr   1.3.1
## v tidyr   0.8.2     v purrr   0.2.5
## v tibble  1.4.2     v forcats 0.3.0

## -- Conflicts ------------------------------------------------------------------------------ tidyver
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x purrr::set_names() masks magrittr::set_names()
```

R

Allows you to analyze data easily

R has a huge community and many solutions are available online

Many libraries offer ease abstractions to analyze and plot data

About environmental awareness

R allows you to load data in many formats (e.g. csv files)

```
globalTemperatures <- read_csv("GlobalTemperatures.csv")
```

```
## Parsed with column specification:
## cols(
##   dt = col_date(format = ""),
##   LandAverageTemperature = col_double(),
##   LandAverageTemperatureUncertainty = col_double(),
##   LandMaxTemperature = col_logical(),
```

```
##   LandMaxTemperatureUncertainty = col_logical(),
##   LandMinTemperature = col_logical(),
##   LandMinTemperatureUncertainty = col_logical(),
##   LandAndOceanAverageTemperature = col_logical(),
##   LandAndOceanAverageTemperatureUncertainty = col_logical()
## )
```

```
## Warning: 11952 parsing failures.
##  row                           col           expected               actual                      file
## 1201 LandMaxTemperature             1/0/T/F/TRUE/FALSE 8.241999999999999   'GlobalTemperatures.csv'
## 1201 LandMaxTemperatureUncertainty  1/0/T/F/TRUE/FALSE 1.7380000000000002  'GlobalTemperatures.csv'
## 1201 LandMinTemperature             1/0/T/F/TRUE/FALSE -3.2060000000000004 'GlobalTemperatures.csv'
## 1201 LandMinTemperatureUncertainty  1/0/T/F/TRUE/FALSE 2.822               'GlobalTemperatures.csv'
## 1201 LandAndOceanAverageTemperature 1/0/T/F/TRUE/FALSE 12.832999999999998  'GlobalTemperatures.csv'
## .... ............................. .................. ................... ........................
## See problems(...) for more details.
```

```
head(globalTemperatures, n=5)
```

```
## # A tibble: 5 x 9
##   dt         LandAverageTemp~ LandAverageTemp~ LandMaxTemperat~
##   <date>                <dbl>            <dbl> <lgl>
## 1 1750-01-01             3.03             3.57 NA
## 2 1750-02-01             3.08             3.70 NA
## 3 1750-03-01             5.63             3.08 NA
## 4 1750-04-01             8.49             2.45 NA
## 5 1750-05-01            11.6              2.07 NA
## # ... with 5 more variables: LandMaxTemperatureUncertainty <lgl>,
## #   LandMinTemperature <lgl>, LandMinTemperatureUncertainty <lgl>,
## #   LandAndOceanAverageTemperature <lgl>,
## #   LandAndOceanAverageTemperatureUncertainty <lgl>
```

Data manipulation

You can select columns and filter data in an ease and readable syntax

```
select(filter(globalTemperatures, dt > as.Date("1800-01-01")), dt, LandAverageTemperature)
```

```
## # A tibble: 2,591 x 2
##    dt         LandAverageTemperature
##    <date>                      <dbl>
##  1 1800-02-01                   3.63
##  2 1800-03-01                   4.45
##  3 1800-04-01                   9.12
##  4 1800-05-01                  11.1
##  5 1800-06-01                  13.6
##  6 1800-07-01                  13.8
##  7 1800-08-01                  13.6
##  8 1800-09-01                  10.8
##  9 1800-10-01                   9.47
## 10 1800-11-01                   5.57
## # ... with 2,581 more rows
```

Data transformation

You can also transform your data applying operators (e.g. converting a `date` field to a `numeric` year field) as you seem fit

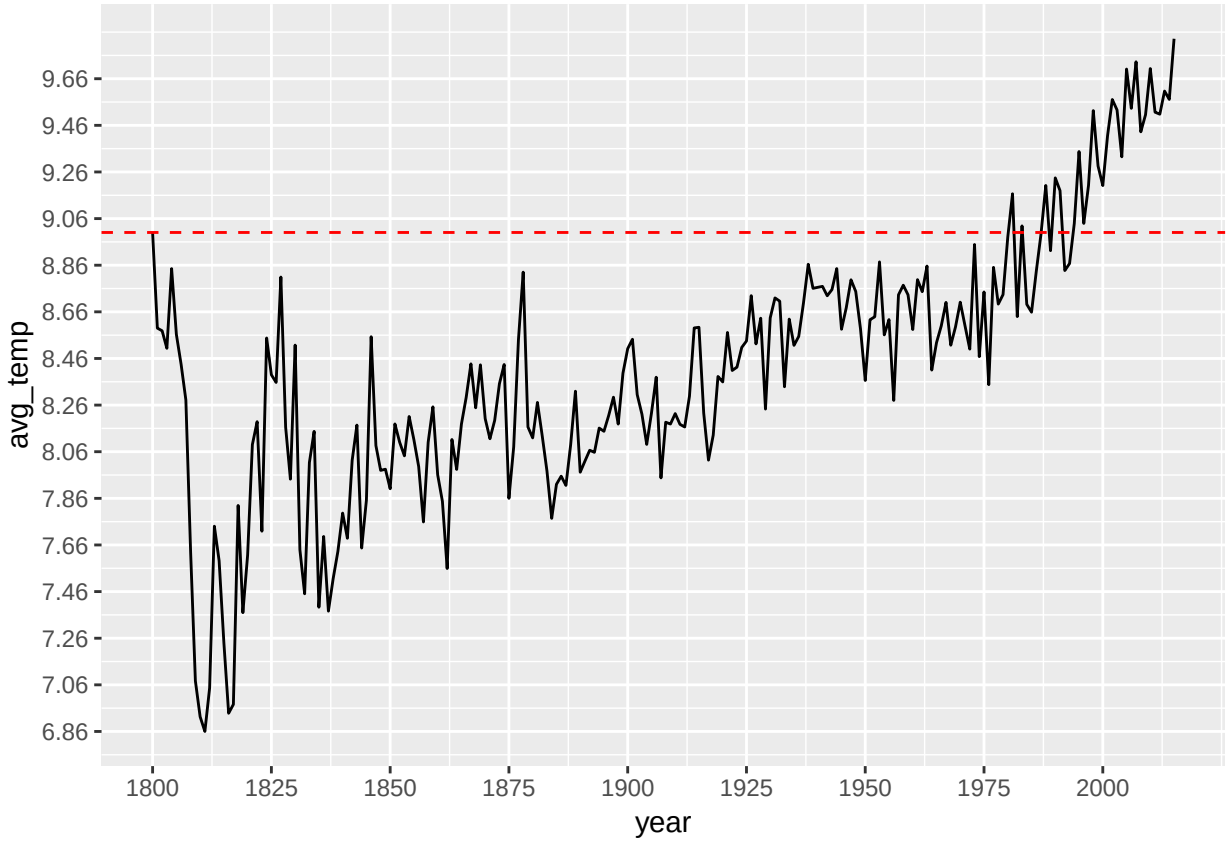
```
dt_temperature <- globalTemperatures %>%
  filter(dt > as.Date("1800-01-01")) %>%
  select(dt, LandAverageTemperature) %>%
  mutate(year = as.numeric(format(dt, "%Y")))

head(dt_temperature, n=5)
```

```
## # A tibble: 5 x 3
##   dt         LandAverageTemperature  year
##   <date>                      <dbl> <dbl>
## 1 1800-02-01                   3.63  1800
## 2 1800-03-01                   4.45  1800
## 3 1800-04-01                   9.12  1800
## 4 1800-05-01                  11.1   1800
## 5 1800-06-01                  13.6   1800
```

Data visualization

```
p
```



Hypothesis Testing

How is the temperature around our neighbours?

```
globalTemperaturesByCountry <- read_csv("GlobalLandTemperaturesByCountry.csv")
```

```
## Parsed with column specification:
## cols(
##   dt = col_date(format = ""),
##   AverageTemperature = col_double(),
##   AverageTemperatureUncertainty = col_double(),
##   Country = col_character()
## )
```

```
unique(globalTemperaturesByCountry %>% select(Country))
```

```
## # A tibble: 243 x 1
##    Country
##    <chr>
##  1 Åland
##  2 Afghanistan
##  3 Africa
##  4 Albania
##  5 Algeria
##  6 American Samoa
##  7 Andorra
##  8 Angola
##  9 Anguilla
## 10 Antarctica
## # ... with 233 more rows
```

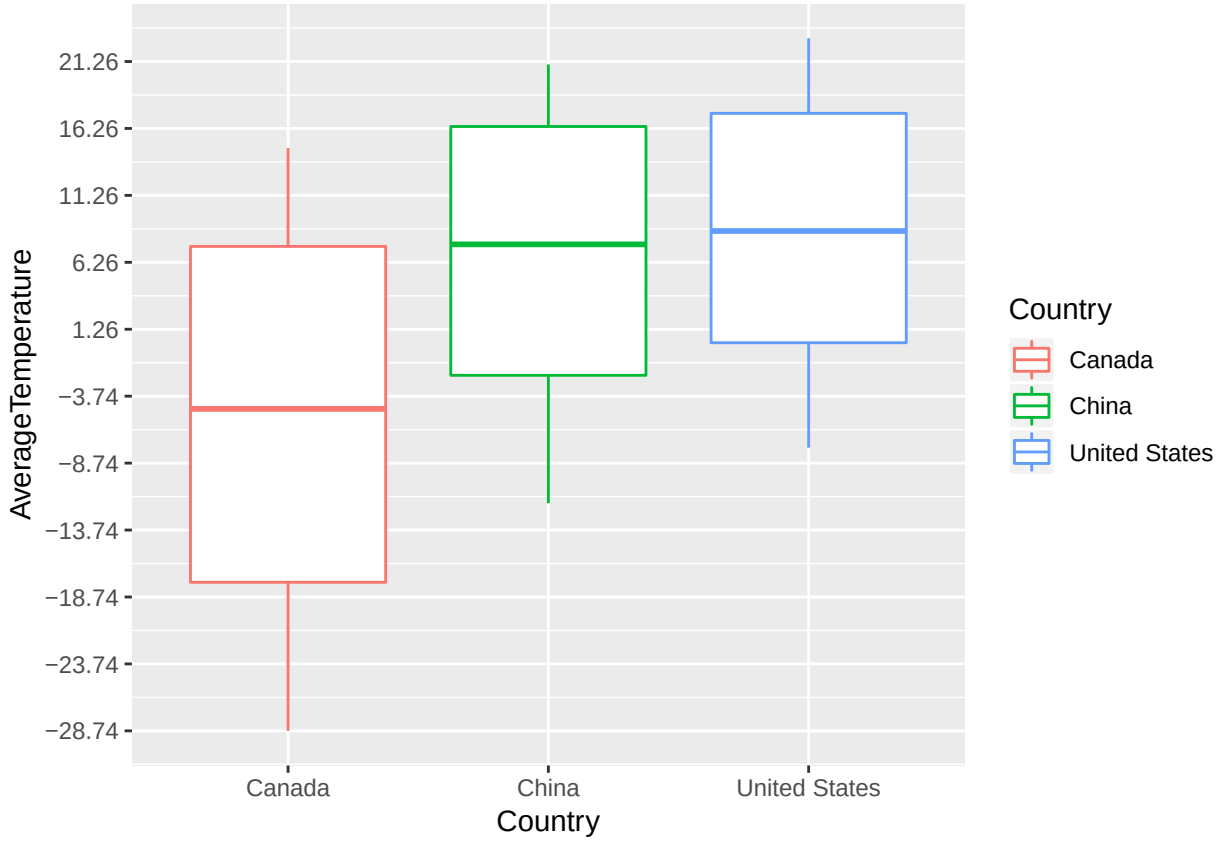
Data transformation

Libraries such as `dplyr` and `tidyverse` allow cleaning our data and removing non-valid entries

```
dt_temperature <- globalTemperaturesByCountry %>%
  na.omit() %>%
  filter(dt > as.Date("1800-01-01")) %>%
  filter(Country == 'Canada' | Country == 'China' | Country == 'United States') %>%
  select(dt, AverageTemperature, Country) %>%
  mutate(year = as.numeric(format(dt, "%Y")))
```

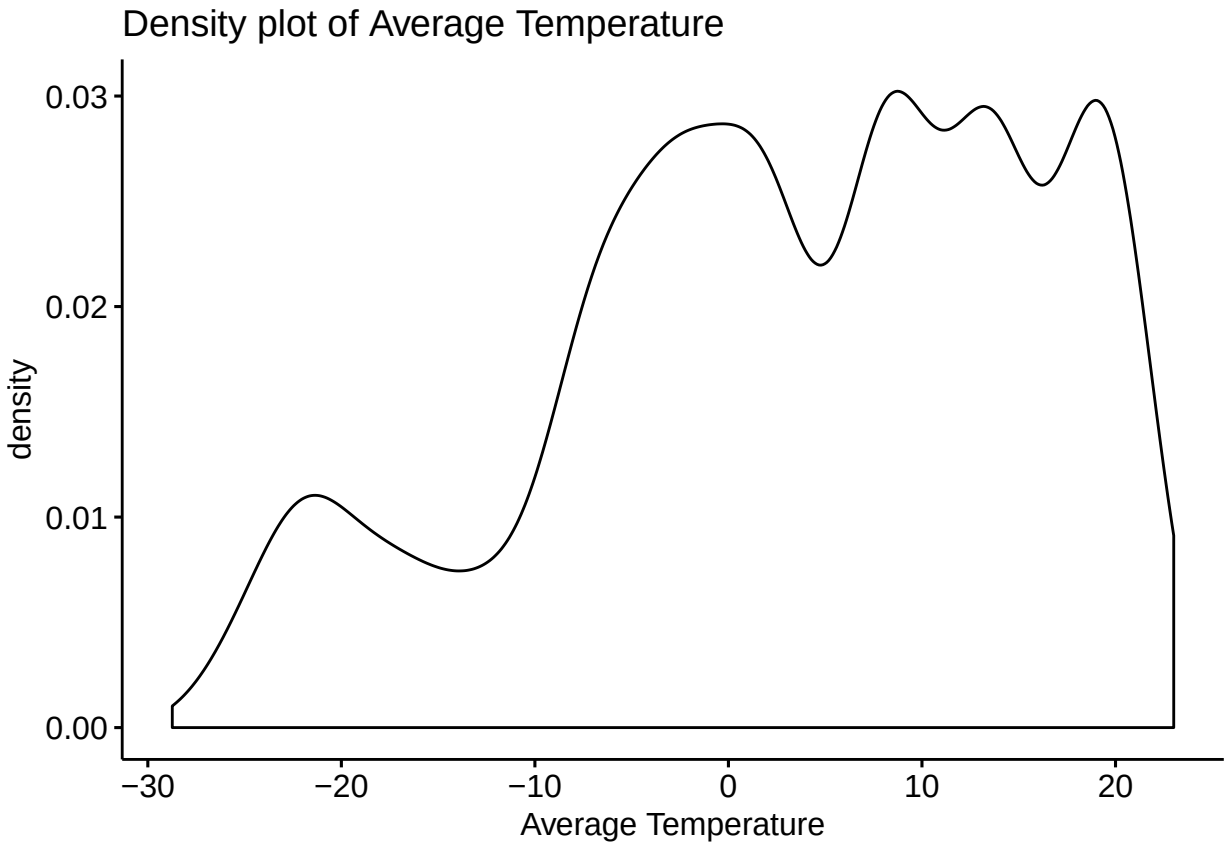
Comparing data visually

```
p
```



Testing Normal Distribution

```
ggdensity(dt_temperature$AverageTemperature,
          main = "Density plot of Average Temperature",
          xlab = "Average Temperature")
```



Checking if our data has an equal number of entries for comparison

```
us_temp <- dt_temperature %>% filter(Country == 'United States') %>% select(AverageTemperature)
cn_temp <- dt_temperature %>% filter(Country == 'China') %>% select(AverageTemperature)
ca_temp <- dt_temperature %>% filter(Country == 'Canada') %>% select(AverageTemperature)

length(us_temp$AverageTemperature)
```

```
## [1] 2399
```

```
length(cn_temp$AverageTemperature)
```

```
## [1] 2201
```

```
length(ca_temp$AverageTemperature)
```

```
## [1] 2399
```

Creating a data frame for comparison

```
head(df, n=5)
```

```
##   AverageTemperature       Country
## 1             19.157 United States
## 2              9.148 United States
## 3             14.862 United States
```

```
## 4             -3.116 United States
## 5              1.602 United States
```

Hypothesis testing

```
kruskal.test(Country ~ AverageTemperature, data = df)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Country by AverageTemperature
## Kruskal-Wallis chi-squared = 6243.1, df = 6047, p-value = 0.03832
```

RStudio is awesome!

Did I mention that this notebook was created on RStudio?