# 8: Lab - Time Series

Environmental Data Analytics | John Fay and Luana Lima

Fall 2024

## Objectives

1. Answer questions on M7
2. Explore the components of times series: trend, seasonal, random
3. Perform trend analysis on another dataset

## Set up

```r
library(tidyverse)
library(lubridate)
#install.packages("trend")
library(trend)
#install.packages("zoo")
library(zoo)
#install.packages("Kendall")
library(Kendall)
#install.packages("tseries")
library(tseries)
library(here)
here
```

```
## function (...)
## {
##     .root_env$root$f(...)
## }
## <bytecode: 0x12d530498>
## <environment: namespace:here>
```

```r
# Set theme
mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "top")
theme_set(mytheme)
```

## Import Datasets

Today we will work with data from the Climate Change Knowledge Portal from the World Bank Group. More specifically historical rainfall and temperature averages for Brazil. You will find two new data files on folder "/Data/Raw/". One with rainfall named "pr__1901__2016__BRA.csv" and another with temperature named "tas__1901__2016__BRA.csv". The data span the period from 1901 to 2016 in monthly steps. You can download the data [here][https://climateknowledgeportal.worldbank.org/download-data]

Research question: Can you see any changes on the rainfall regime or average temperature over time?

```
#Exercise 1: Import both datasets using the read.csv function.
Rainfall_BR <- read.csv(here("Data/Raw/pr_1901_2016_BRA.csv"), stringsAsFactors = TRUE)

Temp_BR <- read.csv(here("Data/Raw/tas_1901_2016_BRA.csv"), stringsAsFactors = TRUE)

#Exercise 2: Tidy the rainfall data sets.
#a Rename the column with Rainfall to get rid of the dots.
#b Note that on both data sets that is a column with the month name and average. Convert it to a Month
#c Now you should have a column with Month. Use the paste0() function to paste month and year together
#d Select only the columns of interest: Date and rainfall

Rainfall_BR_processed <-
  Rainfall_BR %>%
  rename( Rainfall = Rainfall....MM.) %>%
  separate(Statistics,c("Null","Month","Null2")," ") %>%
  mutate( Date = my(paste0(Month,"-",Year))) %>%
  select(Date,Rainfall)

#Exercise 3: Repeat exercise 2 to the temperature dataset.
Temp_BR_processed <-
  Temp_BR %>%
  rename( Temperature_C = Temperature....Celsius.) %>%
  separate(Statistics,c("Null","Month","Null2")," ") %>%
  mutate( Date = my(paste0(Month,"-",Year))) %>%
  select(Date,Temperature_C)


#Exercise 4: Join the temperature and rainfall into one tidy data frame with 3 columns: Date, Rainfall
BR_complete <- inner_join(Rainfall_BR_processed,Temp_BR_processed)
```

```
## Joining with `by = join_by(Date)`
```

## Initial plots

```
#Exercise 5: Check if there is any missing data for both temperature and rainfall series.
summary(BR_complete$Rainfall)
```

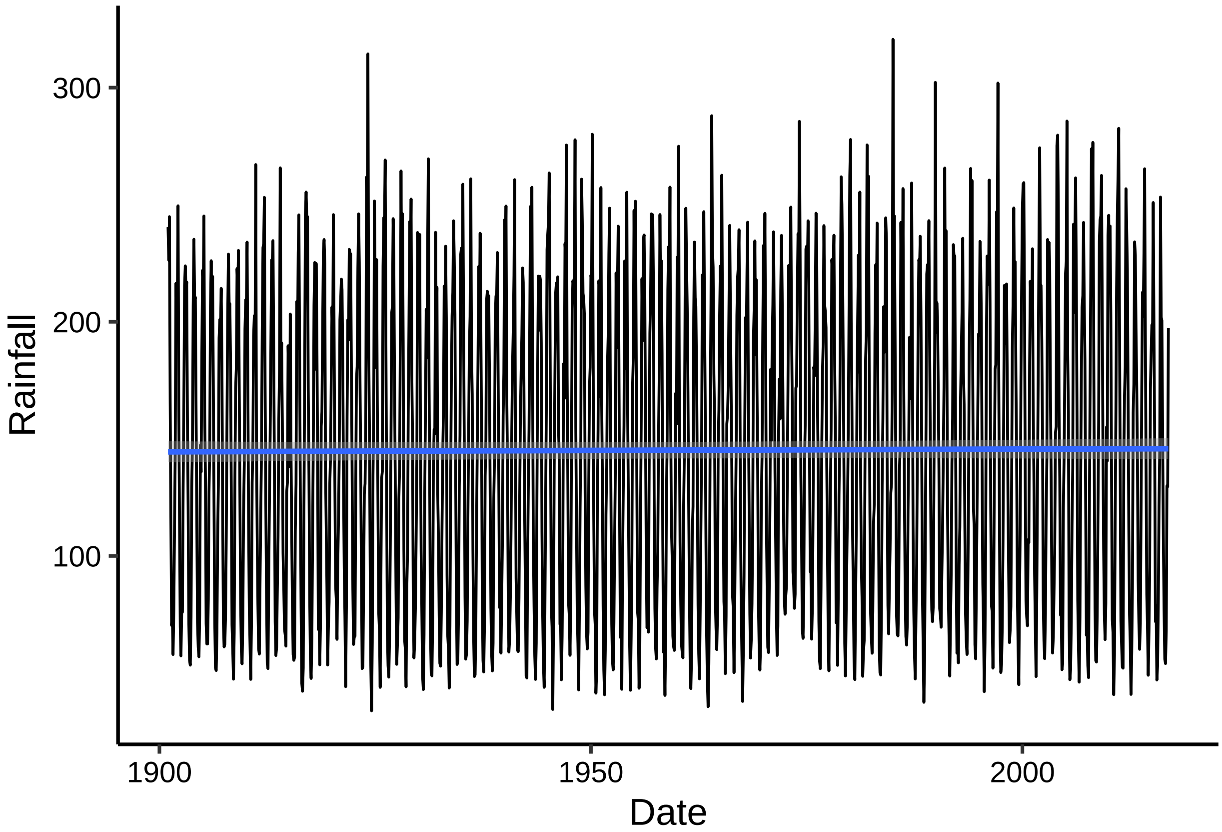
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   33.86   77.80  143.60  145.15  205.44  320.65
```

```
#no NAs
summary(BR_complete$Temperature_C)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   21.73   24.27   25.20   24.96   25.67   27.35
```

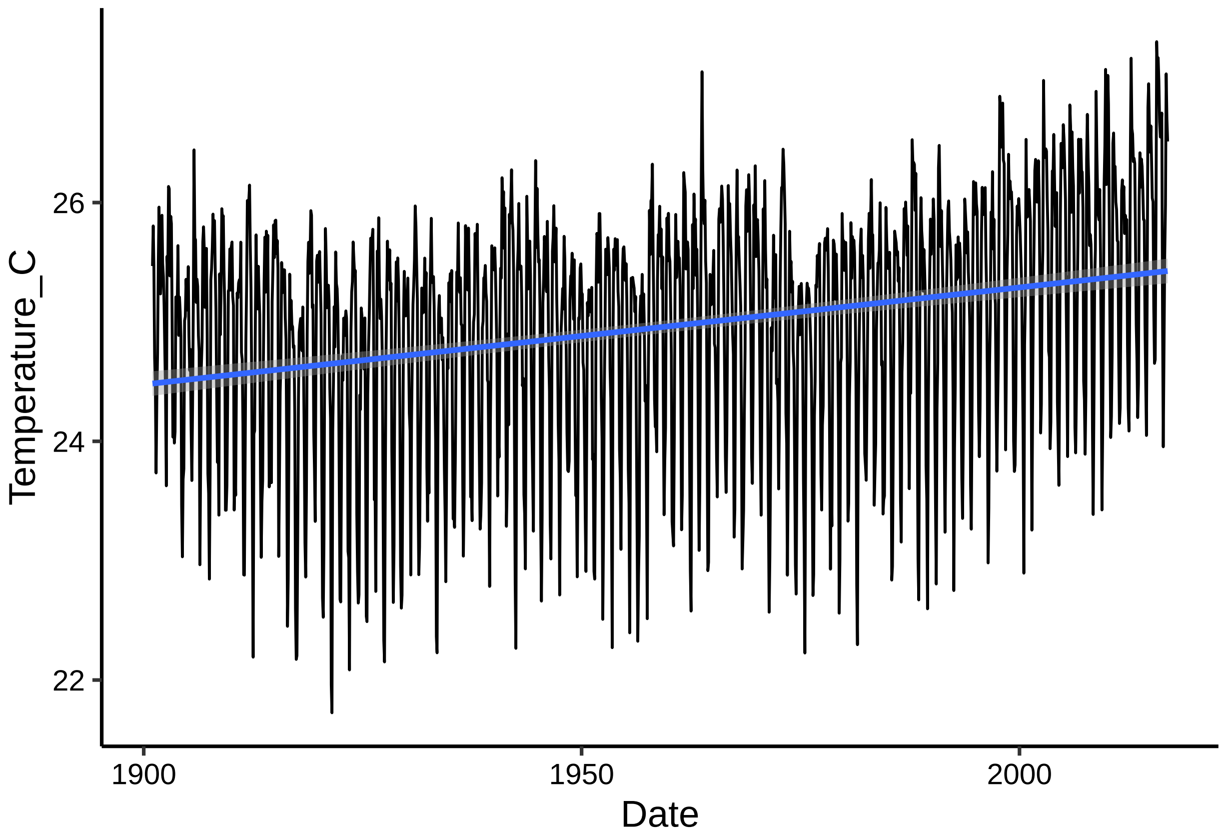
```
#no NAs

#Exercise 6: Plot both series over time on separate plots. Add a trend line using geom_smooth(). Can yo
ggplot(BR_complete, aes( x = Date, y = Rainfall)) +
  geom_line() +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```
ggplot(BR_complete, aes( x = Date, y = Temperature_C)) +
  geom_line() +
  geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




### Creating time series objects

```
#Exercise 7: Create a time series objects for each series using ts(). Make sure you specify the startin

f_month <- month(first(BR_complete$Date))
f_year <- year(first(BR_complete$Date))

BR_Rain_ts <- ts(BR_complete$Rainfall,frequency = 12,start=c(1901,1))
BR_Temp_ts <- ts(BR_complete$Temperature_C,frequency = 12,start=c(f_year,f_month))
```

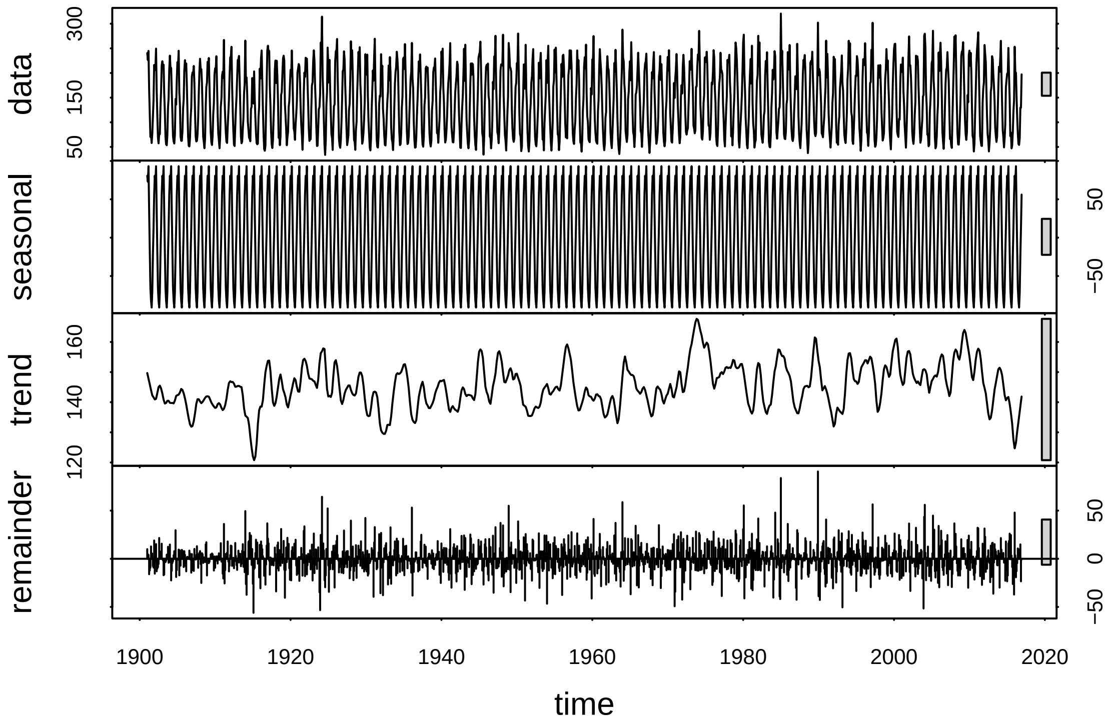
### Decomposing a time series dataset

A given time series can be made up of several component series:

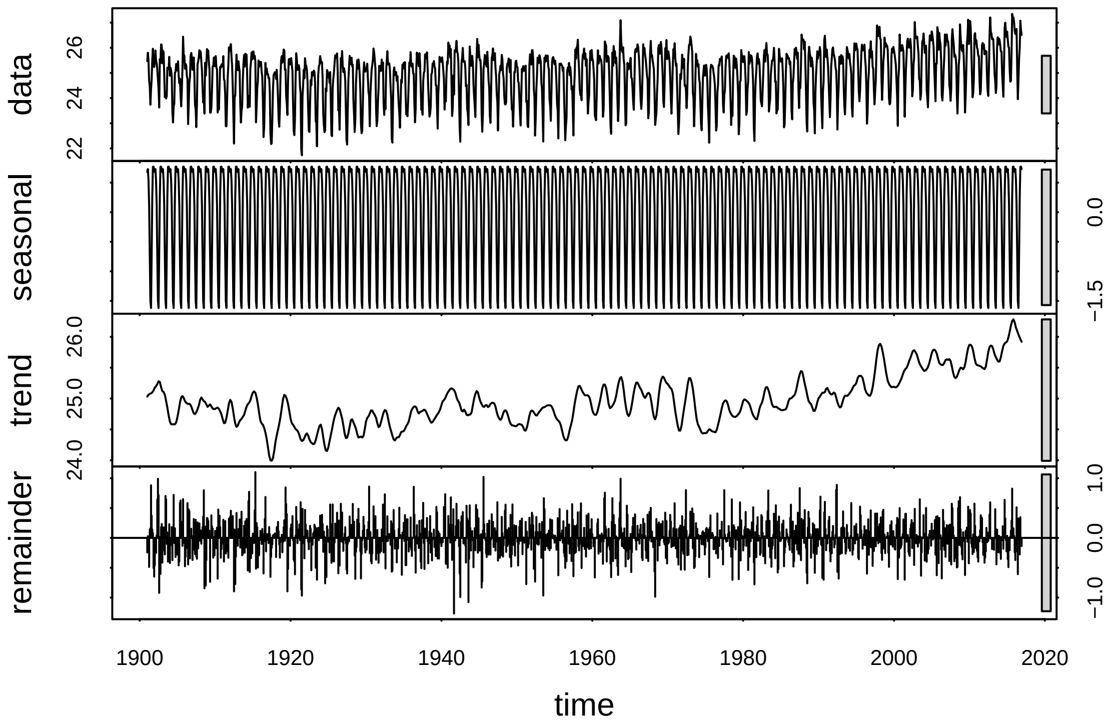
1. A **seasonal** component, which repeats over a fixed known period (e.g., seasons of the year, months, days of the week, hour of the day)
2. A **trend** component, which quantifies the upward or downward progression over time. The trend component of a time series does not have to be monotonic.
3. An **error** or **random** component, which makes up the remainder of the time series after other components have been accounted for. This component reflects the noise in the dataset.
4. (optional) A **cyclical** component, which repeats over periods greater than the seasonal component. A good example of this is El Niño Southern Oscillation (ENSO) cycles, which occur over a period of 2-8 years.

The `stl` function decomposes the time series object into its component parts. We must specify that the window for seasonal extraction is either "periodic" or a specific number of at least 7. The decomposition proceeds through a loess (locally estimated scatterplot smoothing) function.

```
# Exercise 8: Use the stl() function to decompose both series.
# Visualize the decomposed series using plot(). What components can you identify?

BR_Rain_Decomposed <- stl(BR_Rain_ts, s.window = "periodic")
plot(BR_Rain_Decomposed)
```

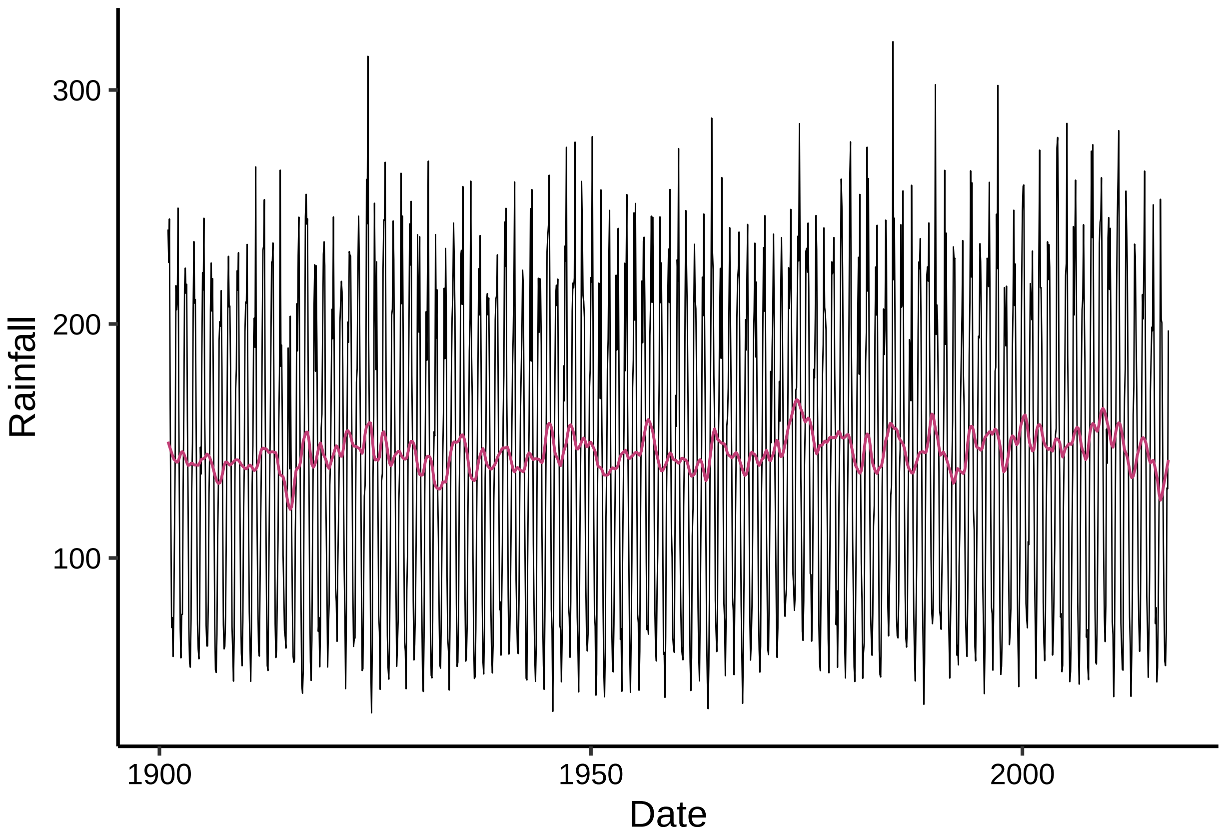



```
BR_Temp_Decomposed <- stl(BR_Temp_ts, s.window = "periodic")
plot(BR_Temp_Decomposed)
```

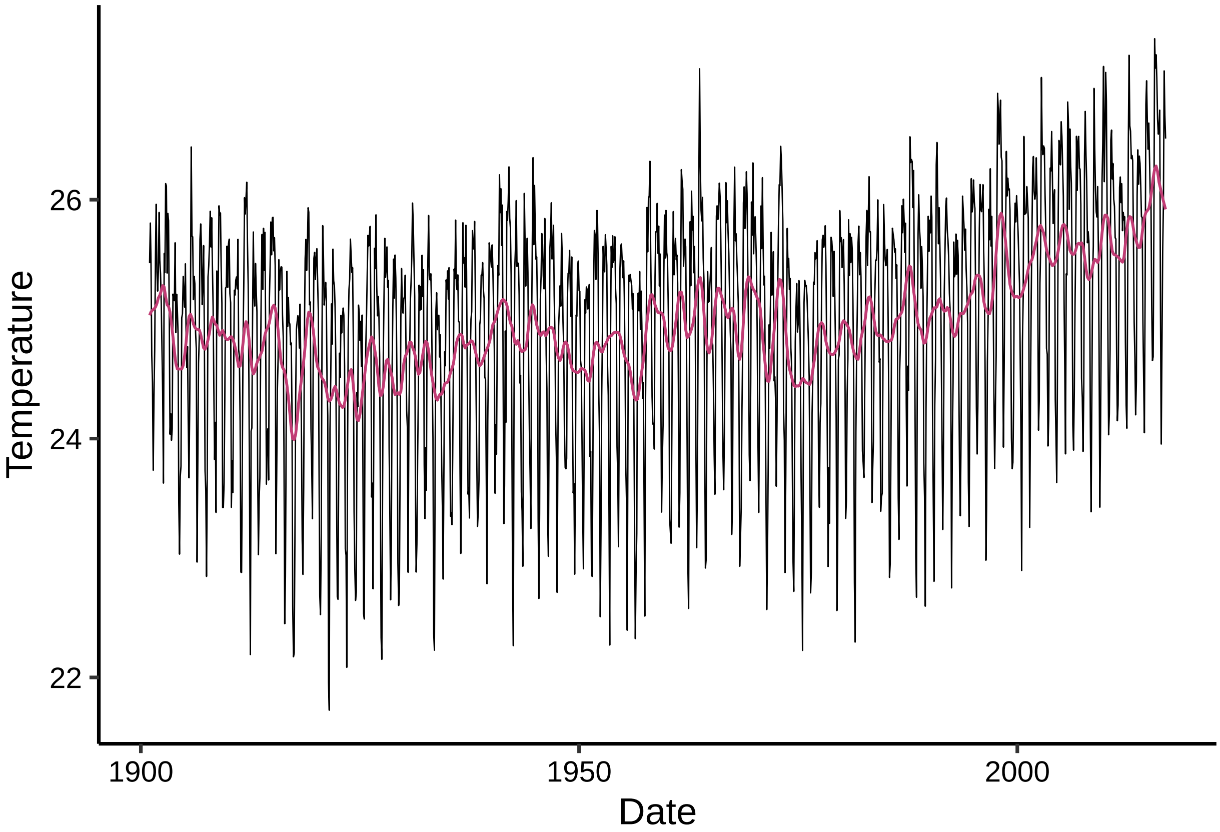



```r
# Exercise 9: Use the times series object generated with the stl() to extract the
# components of each series and turn them into data frames.

BR_Rain_Components <- as.data.frame(BR_Rain_Decomposed$time.series[,1:3])
BR_Temp_Components <- as.data.frame(BR_Temp_Decomposed$time.series[,1:3])

BR_Rain_Components <- mutate(BR_Rain_Components,
        Observed = BR_complete$Rainfall,
        Date = BR_complete$Date)

BR_Temp_Components <- mutate(BR_Temp_Components,
        Observed = BR_complete$Temperature_C,
        Date = BR_complete$Date)


# Exercise 10: Visualize how the trend maps onto the data for both series
ggplot(BR_Rain_Components) +
  geom_line(aes(y = Observed, x = Date),  size = 0.25) +
  geom_line(aes(y = trend, x = Date), color = "#c13d75ff") +
  #geom_line(aes(y = seasonal, x = Date), color = "blue") +
  #geom_hline(yintercept = 0, lty = 2) +
  ylab("Rainfall")
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




```r
ggplot(BR_Temp_Components) +
  geom_line(aes(y = Observed, x = Date),  size = 0.25) +
  geom_line(aes(y = trend, x = Date), color = "#c13d75ff") +
  #geom_line(aes(y = seasonal, x = Date), color = "blue") +
  #geom_hline(yintercept = 0, lty = 2) +
  ylab("Temperature")
```




### Trend analysis

Monotonic trends are a gradual shift over time that is consistent in direction. Specific tests for monotonic trend analysis are listed below, with assumptions and tips:

- **linear regression**: no seasonality, fits the assumptions of a parametric test. Function: `lm` Regress time series over time. If significant relationship, then there is a trend.
- **Mann-Kendall**: no seasonality, non-parametric, missing data allowed. Function: `Kendall::MannKendall()` or `trend::mk.test()`

H0: S = 0, i.e., no trend H1: S != 0, i.e., follow a trend

- **Seasonal Mann-Kendall**: seasonality, non-parametric `Kendall::SeasonalMannKendall` or `trend::smk.test()`
- **Spearman Rho**: no seasonality, non-parametric, missing data allowed. Function: `stats::cor.test(method="spearma` Similar to `lm` but allows for non-linear trend.

```
# Exercise 9: Apply one of the trend detection test to the original data set. Remember that the data ha

Rain_trend <- trend::smk.test(BR_Rain_ts)
# Inspect results
Rain_trend
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  BR_Rain_ts
## z = 4.8698, p-value = 1.117e-06
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##       S    varS
##    7071 2107719
```

```
summary(Rain_trend)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: BR_Rain_ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                         S     varS    tau      z  Pr(>|z|)
## Season 1:   S = 0   662 175643.3  0.099  1.577 0.1147505
## Season 2:   S = 0   828 175643.3  0.124  1.973 0.0484632  *
## Season 3:   S = 0   486 175643.3  0.073  1.157 0.2471716
## Season 4:   S = 0  1230 175643.3  0.184  2.932 0.0033626 **
## Season 5:   S = 0  1148 175643.3  0.172  2.737 0.0062035 **
## Season 6:   S = 0   756 175643.3  0.113  1.801 0.0716262  .
## Season 7:   S = 0  1290 175643.3  0.193  3.076 0.0021004 **
## Season 8:   S = 0    28 175643.3  0.004  0.064 0.9486326
## Season 9:   S = 0  -110 175643.3 -0.016 -0.260 0.7948004
## Season 10:   S = 0 -594 175643.3 -0.089 -1.415 0.1570853
## Season 11:   S = 0  988 175643.3  0.148  2.355 0.0185199  *
## Season 12:   S = 0  359 175642.3  0.054  0.854 0.3929845
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

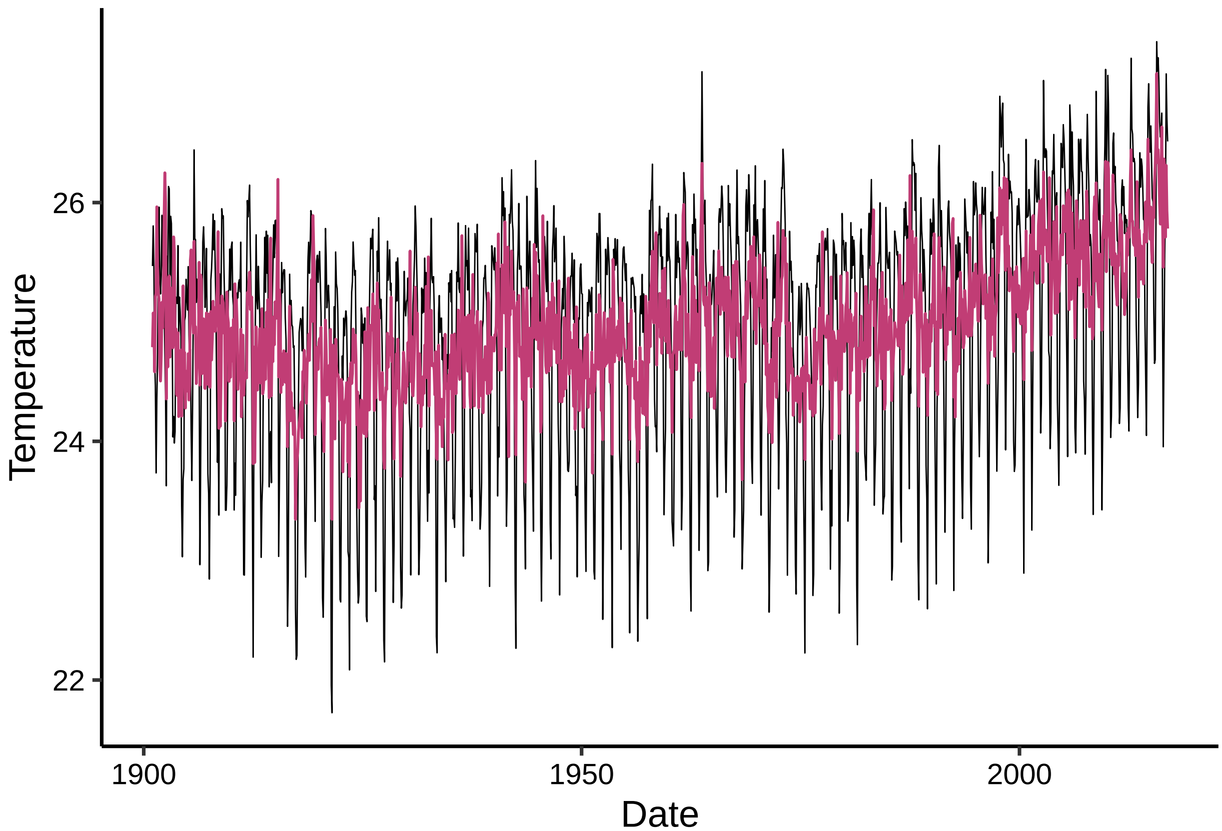
```
Temp_trend <- trend::smk.test(BR_Temp_ts)
# Inspect results
Temp_trend
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  BR_Temp_ts
## z = 20.178, p-value < 2.2e-16
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##       S    varS
##   29295 2107713
```

```r
summary(Temp_trend)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: BR_Temp_ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                         S      varS   tau      z   Pr(>|z|)
## Season 1:   S = 0  2740 175641.3 0.411 6.535 6.3398e-11 ***
## Season 2:   S = 0  2251 175642.3 0.338 5.369 7.9313e-08 ***
## Season 3:   S = 0  2581 175642.3 0.387 6.156 7.4562e-10 ***
## Season 4:   S = 0  2375 175642.3 0.356 5.665 1.4740e-08 ***
## Season 5:   S = 0  2093 175642.3 0.314 4.992 5.9855e-07 ***
## Season 6:   S = 0  1692 175643.3 0.254 4.035 5.4636e-05 ***
## Season 7:   S = 0  1706 175643.3 0.256 4.068 4.7366e-05 ***
## Season 8:   S = 0  2606 175643.3 0.391 6.216 5.1088e-10 ***
## Season 9:   S = 0  2053 175642.3 0.308 4.896 9.7687e-07 ***
## Season 10:   S = 0 3286 175643.3 0.493 7.838 4.5684e-15 ***
## Season 11:   S = 0 2822 175643.3 0.423 6.731 1.6836e-11 ***
## Season 12:   S = 0 3090 175643.3 0.463 7.371 1.6988e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Exercise 10: Now remove the seasonality and apply another test. Compare the results with what you obt

#removing seasonality for temperature
BR_Temp_Nonseas <- BR_Temp_ts - BR_Temp_Decomposed$time.series[,1]

BR_Temp_Components <- mutate(BR_Temp_Components,
                             Nonseasonal = BR_Temp_Nonseas)
#plotting resulting series
ggplot(BR_Temp_Components) +
  geom_line(aes(y = Observed, x = Date),  size = 0.25) +
  geom_line(aes(y = Nonseasonal, x = Date), color = "#c13d75ff") +
  ylab("Temperature")
```

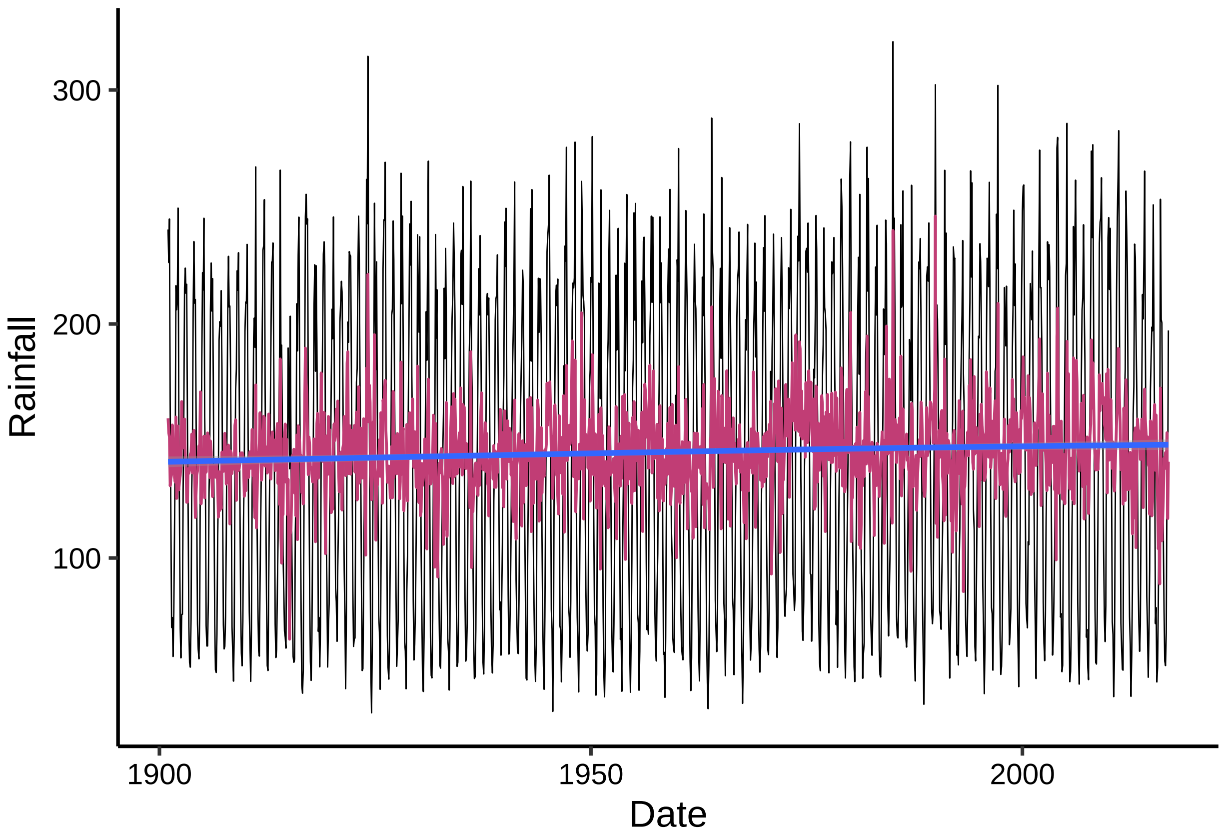



```r
Temp_Nonseas_trend <- trend::mk.test(BR_Temp_Nonseas)
# Inspect results
Temp_Nonseas_trend
```

```
##
##  Mann-Kendall trend test
##
## data:  BR_Temp_Nonseas
## z = 19.941, n = 1392, p-value < 2.2e-16
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## 3.453990e+05 3.000146e+08 3.567683e-01
```

```r
#removing seasonality for rainfall
BR_Rain_Nonseas <- BR_Rain_ts - BR_Rain_Decomposed$time.series[,1]

BR_Rain_Components <- mutate(BR_Rain_Components,
                             Nonseasonal = BR_Rain_Nonseas)
#plotting resulting series
ggplot(BR_Rain_Components) +
  geom_line(aes(y = Observed, x = Date),  size = 0.25) +
  geom_line(aes(y = Nonseasonal, x = Date), color = "#c13d75ff") +
  geom_smooth(aes(y = Nonseasonal, x = Date)) +
  ylab("Rainfall")
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```r
Rain_Nonseas_trend <- trend::mk.test(BR_Rain_Nonseas)
# Inspect results
Rain_Nonseas_trend
```

```
##
##  Mann-Kendall trend test
##
## data:  BR_Rain_Nonseas
## z = 4.9012, n = 1392, p-value = 9.523e-07
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##            S         varS          tau
## 8.489500e+04 3.000146e+08 8.768917e-02
```

What would we conclude based on these findings?

> Answer: