logistic_regression

Sujan Bhattarai

Running Code

This is end to End project on classifying precipitation phase using Logistic regression, statistical analysis, and change point detection. The file tree is located in README.md. The orginal data will not be shared in the github, considering github is public platform and data used is highly sensitive. However, the format of data used, and its sample is included in README so that it can be reproduced with ease .

things before you begin

to install CRHMr package for updated R, you first need these packages install.packages(c("operators", "topmodel", "DEoptim", "XML")) the installation of CRHMr package requires download of Ecohydrology package install.packages("devtools") #library(devtools) #install_github("CentreForHydrology/CRHMr")

```
#load all required packages that will be used for analysis
library(CRHMr)
library(lubridate)
```

```
Attaching package: 'lubridate'

The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

```
library(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
```

```r
library(trend)
library(changepoint)
```

```
Loading required package: zoo


Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

Successfully loaded changepoint package version 2.2.4
 See NEWS for details of changes.
```

```r
library(patchwork)
library(caret)
```

```
Loading required package: ggplot2

Loading required package: lattice
```

```r
library(mblm)
library(zoo)
library(pracma)
library(car)
```

```
Loading required package: carData


Attaching package: 'car'

The following object is masked from 'package:pracma':

    logit

The following object is masked from 'package:dplyr':

    recode
```

```r
library(gridExtra)
```

```
Attaching package: 'gridExtra'

The following object is masked from 'package:dplyr':

    combine
```

```r
library(xts)
```

```
######################## Warning from 'xts' package ########################
#                                                                           #
# The dplyr lag() function breaks how base R's lag() function is supposed to  #
# work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
# source() into this session won't work correctly.                          #
#                                                                           #
# Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
# conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop          #
# dplyr from breaking base R's lag() function.                              #
#                                                                           #
# Code in packages is not affected. It's protected by R's namespace mechanism #
# Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
#                                                                           #
###############################################################################
```

```
Attaching package: 'xts'

The following objects are masked from 'package:dplyr':

    first, last
```

```r
library(here)
```

```
here() starts at /Users/sujanbhattarai/Desktop/paras-daiwork/PhaseChange
```

```r
#set the path for plots so that all plots are saved inside `plots` folder
plot_path = file.path(here('plots'))
```

```r
#constuct file path using here package
climate_data <- here("data", "climate_data.obs")
climate_data <- readObsFile(climate_data, timezone = 'MST', quiet = F)
```

```
                summary                          value
1   Time step (hours):                               1
2          Total rows:                          350638
3       Complete rows:                          350638
4                From:                      1979-01-01
5                  To:                      2019-01-01
6 First complete date:                      1979-01-01
7  Last complete date:                      2019-01-01
8 Number of variables:                               6
9      Variable names: t.1 ea.1 u.1 Qsi.1 Qli.1 p.1
```

```r
#multiply wind variable for air speed at height of 5m from the ground
climate_data$u.1 <- climate_data$u.1 * 5

#Obs_Data is specific to CRHM package, so the the code below automatically chose
```

```r
## Ea to change to Relative humidity, and select onlly the required columns
climate_data     <- changeEatoRH(climate_data)
climate_data     <- subset(climate_data, select = c(datetime, t.1, rh.1, u.1, p.1))
```

```r
#phaseCorrect function partitions based on the temp
climate_partitioned <- phaseCorrect(climate_data, Tcol = 1, RHcol = 2, Ucol = 3, Pcol = 4,
                            shield = 0, quiet = F)
```

```
Time interval: 1 hours
Model time step: 2
b: 2.50286
c: 0.125006
               summary
1   Time step (hours):
2          Total rows:
3       Complete rows:
4                From:
5                  To:
6 First complete date:
7  Last complete date:
8 Number of variables:
9      Variable names:

1
2
3
4                                                                                   1979
5                                                                                   2019
6                                                                                   1979
7                                                                                   2018
8
9 t.1 rh.1 u.1 p.1 phaseCorrectRain phaseCorrectSnow phaseCorrectTotalPrecip phaseCorrectRai
```

```r
#make new columns for months and days, add seasons to make it easy for  analysis and visuali
climate_cleaned <- climate_partitioned %>%
        mutate(phaseCorrectRainRatio= round(phaseCorrectRainRatio, digits=2),
               year= year(datetime),
               month= month(datetime, label = TRUE),
               day= day(datetime),
               period= if_else(hour(datetime)>5 & hour(datetime)< 19, "Day", "Night"),
               season = ifelse(
               month <= "Feb"  | month == "Dec", "winter",
               ifelse( month == "Mar" | month == "Apr" | month == "May", "pre_monsoon",
                       ifelse(month == "Oct" | month == "Nov", "post_monsoon", "monsoon")))
       filter(year != 2019 & p.1 != 0)
```

```r
##duplicate copy, the original data is often required in later stage
clone_copy <- climate_partitioned %>%
        mutate(phaseCorrectRainRatio= round(phaseCorrectRainRatio, digits=2),
               year= year(datetime),
               month= month(datetime, label = TRUE),
               day= day(datetime),
               period= if_else(hour(datetime)>5 & hour(datetime)< 19, "Day", "Night"),
               season = ifelse(
               month <= "Feb"  | month == "Dec", "winter",
               ifelse( month == "Mar" | month == "Apr" | month == "May", "pre_monsoon",
                       ifelse(month == "Oct" | month == "Nov", "post_monsoon", "monsoon"))))
```

```r
##### summarized yearly data for exponential moving average
moving_average <- climate_cleaned %>%
                  group_by(year) %>%
                  summarize(temp= mean(t.1), rh= mean(rh.1), wind=mean(u.1), precip=sum(p.1)
                        rain_ratio= mean(phaseCorrectRainRatio)) %>% ungroup()
```

```r
# for each variable run the function and store the variable
# type 'e' is used for exponential weighted average
moving_average$temp_ema        <- movavg(moving_average$temp,       n=10, type='e')
moving_average$rh_ema          <- movavg(moving_average$rh,         n=5, type='e')
moving_average$wind_ema        <- movavg(moving_average$wind,       n=5, type='e')
moving_average$precip_ema      <- movavg(moving_average$precip,     n=5, type='e')
moving_average$rain_ratio_ema  <- movavg(moving_average$rain_ratio, n=5, type='e')
```

```r
###smoother precipitation curve as it shows cyclicity
library(bayesforecast)
```

```
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

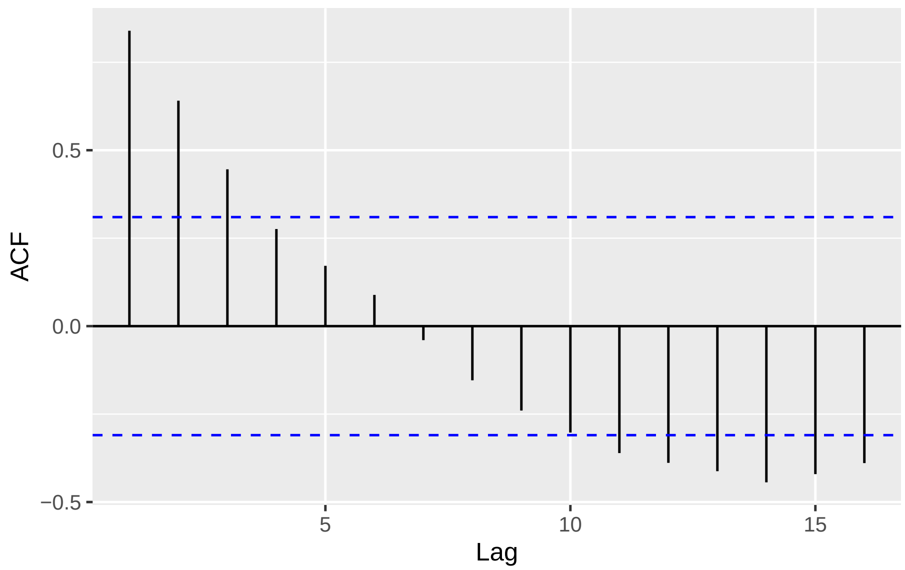
```
Registered S3 methods overwritten by 'bayesforecast':
  method      from
  autoplot.ts forecast
  forecast.ts forecast
  fortify.ts  forecast
  print.garch tseries
```

```
Attaching package: 'bayesforecast'
```

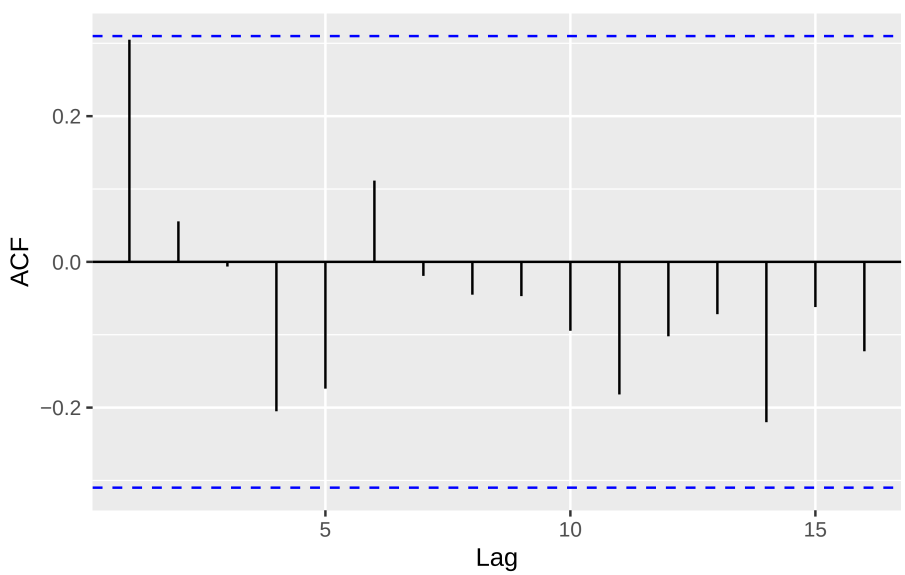
```
The following objects are masked from 'package:base':

    beta, gamma
```

```r
ggacf(moving_average$precip_ema, title = NULL)
```




```r
# cyclicity is observed in the data, so remove that by subtracting
moving_average$smooth_precip <- moving_average$precip - moving_average$precip_ema
ggacf(moving_average$smooth_precip, title = NULL)
```




```r
###trend test for all variables
mk.test(moving_average$temp_ema)
```

```
    Mann-Kendall trend test

data:  moving_average$temp_ema
z = 6.9324, n = 40, p-value = 4.139e-12
alternative hypothesis: true S is not equal to 0
sample estimates:
          S        varS         tau
 596.0000000 7366.6666667    0.7641026
```

```r
sens.slope(moving_average$temp_ema)
```

```
    Sen's slope

data:  moving_average$temp_ema
z = 6.9324, n = 40, p-value = 4.139e-12
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.01790674 0.02428181
sample estimates:
Sen's slope
 0.02155175
```

```r
pettitt.test(moving_average$temp_ema)
```

```
    Pettitt's test for single change-point detection

data:  moving_average$temp_ema
U* = 386, p-value = 2.413e-06
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             20
```

```r
mk.test(moving_average$rh_ema)
```

```
    Mann-Kendall trend test

data:  moving_average$rh_ema
z = 2.505, n = 40, p-value = 0.01225
alternative hypothesis: true S is not equal to 0
sample estimates:
          S        varS         tau
 216.0000000 7366.6666667    0.2769231
```

```
sens.slope(moving_average$rh_ema)
```

```
    Sen's slope

data:  moving_average$rh_ema
z = 2.505, n = 40, p-value = 0.01225
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.01023207 0.05947579
sample estimates:
Sen's slope
 0.02721846
```

```
pettitt.test(moving_average$rh_ema)
```

```
    Pettitt's test for single change-point detection

data:  moving_average$rh_ema
U* = 348, p-value = 3.094e-05
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             14
```

```
mk.test(moving_average$wind_ema)
```

```
    Mann-Kendall trend test

data:  moving_average$wind_ema
z = -4.6022, n = 40, p-value = 4.181e-06
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
-396.0000000 7366.6666667   -0.5076923
```

```
sens.slope(moving_average$wind_ema)
```

```
    Sen's slope

data:  moving_average$wind_ema
z = -4.6022, n = 40, p-value = 4.181e-06
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
```

```
 -0.007121856 -0.003655108
sample estimates:
 Sen's slope
-0.005428641
```

```r
pettitt.test(moving_average$wind_ema)
```

```

    Pettitt's test for single change-point detection

data:  moving_average$wind_ema
U* = 317, p-value = 0.0002039
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             19
```

```r
mk.test(moving_average$smooth_precip)
```

```

    Mann-Kendall trend test

data:  moving_average$smooth_precip
z = -0.43109, n = 40, p-value = 0.6664
alternative hypothesis: true S is not equal to 0
sample estimates:
            S          varS           tau
 -38.00000000 7366.66666667   -0.04871795
```

```r
sens.slope(moving_average$smooth_precip)
```

```

    Sen's slope

data:  moving_average$smooth_precip
z = -0.43109, n = 40, p-value = 0.6664
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 -3.822617  2.640772
sample estimates:
Sen's slope
  -0.959712
```

```r
pettitt.test(moving_average$smooth_precip)
```

```

    Pettitt's test for single change-point detection
```

```
data:  moving_average$smooth_precip
U* = 117, p-value = 0.5718
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                              5
```

```
mk.test(moving_average$rain_ratio_ema)
```

```

    Mann-Kendall trend test

data:  moving_average$rain_ratio_ema
z = 4.9284, n = 40, p-value = 8.291e-07
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 424.0000000 7366.6666667    0.5435897
```

```
sens.slope(moving_average$rain_ratio_ema)
```

```

    Sen's slope

data:  moving_average$rain_ratio_ema
z = 4.9284, n = 40, p-value = 8.291e-07
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.0009512447 0.0018916136
sample estimates:
Sen's slope
0.001390375
```

```
pettitt.test(moving_average$rain_ratio_ema)
```

```

    Pettitt's test for single change-point detection

data:  moving_average$rain_ratio_ema
U* = 368, p-value = 8.35e-06
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             16
```

```
#create slope values for sen's slopem
# this value will be used in next plot
mkmodel_temp     <- mblm(temp_ema ~ year, data=moving_average)
```

```
mkmodel_wind     <- mblm(wind_ema ~ year, data=moving_average)
mkmodel_rh       <- mblm(rh_ema ~ year, data=moving_average)
mkmodel_precip   <- mblm(precip_ema ~ year, data=moving_average)
mkmodel_smoother <- mblm(smooth_precip ~ year, data=moving_average)
```

```
###legend color, manual color setup, often this can be used upon
  #multiples graphs
colors <- c("variable_color"= "blue", "weight_color"="lightblue", 'abline_color'= 'red')

#prepare graphs
make_graph <- function(type, variable, weight, xlab, ylab, mkmodel, legend.position="FALSE")
  plot <- moving_average %>%
    ggplot(aes(year, {{variable}})) +
    xlab(xlab) +
    ylab(ylab) +
    geom_line(data = moving_average, aes(year, {{weight}}, color = 'col2')) +
    geom_point(data= moving_average, aes(year, {{weight}}, color= "col2"))+
    geom_abline(intercept = coef(mkmodel)[1], slope = coef(mkmodel)[2] ) +
    theme_bw() +
    scale_x_continuous(breaks = seq(1979, 2019, by = 10)) +
    theme(axis.title = element_text(size = 12)) +
    theme(legend.position = legend.position) +
    scale_color_manual(values = c('col2' = '#E69F00'))

  if (identical(type, geom_bar)) {
    plot <- plot + geom_bar(stat = "identity", fill = "#0072B2", alpha=0.5)
  } else {
    plot <- plot + type(stat = "identity", aes(color = "col1")) +
      scale_color_manual(values = c('col1' = '#0072B2', 'col2' = '#E69F00'))
  }
  plot <- plot +
    geom_abline(intercept = coef(mkmodel)[1], slope = coef(mkmodel)[2], color = "red")
  return(plot)
}
```

```
png("plots/trend_test.tiff",  height = 5, width = 7, units = "in", res = 300)
grid.arrange(make_graph( type = geom_line, variable= temp,    weight = temp_ema,   xlab="",
             make_graph( type = geom_line, variable= wind,    weight = wind_ema,   xlab="",
             make_graph( type = geom_step, variable= rh,     weight = rh_ema,     xlab="", y
             make_graph( type = geom_bar,  variable =smooth_precip, weight = precip_ema, xla
```

```
Scale for colour is already present.
Adding another scale for colour, which will replace the existing scale.
Scale for colour is already present.
Adding another scale for colour, which will replace the existing scale.
Scale for colour is already present.
```

```
Adding another scale for colour, which will replace the existing scale.
```

```r
dev.off()
```

```
pdf
  2
```

```r
png("plots/temp.tiff",  height = 5, width = 7, units = "in", res = 300)
make_graph( type = geom_line, variable= temp,    weight = temp_ema,   xlab="", ylab = "Tempe
```

```
Scale for colour is already present.
Adding another scale for colour, which will replace the existing scale.
```

```r
dev.off()
```

```
pdf
  2
```

```r
### Do MK trend test, Sen's slope in all ema columns
calculate_mk_test <- function(data) {
  start <- which(colnames(data) == "temp_ema")
  end   <- which(colnames(data) == "rain_ratio_ema")
  mk_sen_results <- list()

  for (i in start:end) {
    column <- data[[i]]
    mk <- mk.test(column)
    sen <- sens.slope(column)
    petite <- pettitt.test(column)
    mk_sen_results[[colnames(data)[i]]] <- list(mk, sen, petite )
  }
  return(mk_sen_results)
}
calculate_mk_test(moving_average)
```

```
$temp_ema
$temp_ema[[1]]

    Mann-Kendall trend test

data:  column
z = 6.9324, n = 40, p-value = 4.139e-12
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 596.0000000 7366.6666667    0.7641026


$temp_ema[[2]]
```

```
    Sen's slope

data:  column
z = 6.9324, n = 40, p-value = 4.139e-12
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.01790674 0.02428181
sample estimates:
Sen's slope
 0.02155175


$temp_ema[[3]]

    Pettitt's test for single change-point detection

data:  column
U* = 386, p-value = 2.413e-06
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             20



$rh_ema
$rh_ema[[1]]

    Mann-Kendall trend test

data:  column
z = 2.505, n = 40, p-value = 0.01225
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 216.0000000 7366.6666667    0.2769231


$rh_ema[[2]]

    Sen's slope

data:  column
z = 2.505, n = 40, p-value = 0.01225
alternative hypothesis: true z is not equal to 0
```

```
95 percent confidence interval:
 0.01023207 0.05947579
sample estimates:
Sen's slope
 0.02721846


$rh_ema[[3]]

    Pettitt's test for single change-point detection

data:  column
U* = 348, p-value = 3.094e-05
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             14



$wind_ema
$wind_ema[[1]]

    Mann-Kendall trend test

data:  column
z = -4.6022, n = 40, p-value = 4.181e-06
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
-396.0000000 7366.6666667   -0.5076923


$wind_ema[[2]]

    Sen's slope

data:  column
z = -4.6022, n = 40, p-value = 4.181e-06
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 -0.007121856 -0.003655108
sample estimates:
 Sen's slope
-0.005428641
```

```
$wind_ema[[3]]

    Pettitt's test for single change-point detection

data:  column
U* = 317, p-value = 0.0002039
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             19



$precip_ema
$precip_ema[[1]]

    Mann-Kendall trend test

data:  column
z = 0.43109, n = 40, p-value = 0.6664
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
3.800000e+01 7.366667e+03 4.871795e-02


$precip_ema[[2]]

    Sen's slope

data:  column
z = 0.43109, n = 40, p-value = 0.6664
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 -2.314116  5.041670
sample estimates:
Sen's slope
  0.8953558


$precip_ema[[3]]

    Pettitt's test for single change-point detection

data:  column
```

```
U* = 186, p-value = 0.08449
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             12



$rain_ratio_ema
$rain_ratio_ema[[1]]

    Mann-Kendall trend test

data:  column
z = 4.9284, n = 40, p-value = 8.291e-07
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 424.0000000 7366.6666667    0.5435897


$rain_ratio_ema[[2]]

    Sen's slope

data:  column
z = 4.9284, n = 40, p-value = 8.291e-07
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.0009512447 0.0018916136
sample estimates:
Sen's slope
0.001390375


$rain_ratio_ema[[3]]

    Pettitt's test for single change-point detection

data:  column
U* = 368, p-value = 8.35e-06
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             16
```

```r
# par(mar = c(1, 1, 1, 1)) # this is used to set the graphical plot window: not a part of cl
# cpt.meanvar(annual_mean$annual_average, method="AMOC", minseglen = 10) # this test identi
# the code below this point test the same thing for annual daytime and nighttime
# filtering data only for days(6am and 6pm) requred in model fitting in next step

annual_Day <- climate_cleaned %>%
  group_by(year) %>%
  filter( period=="Day") %>%
  summarize(rain_ratio= mean(phaseCorrectRainRatio)) %>%
  ungroup() %>%
  mutate(day_rain_ratio_ema = movavg( rain_ratio, n=5, type='e'))

mk.test(annual_Day$day_rain_ratio_ema)
```

```
    Mann-Kendall trend test

data:  annual_Day$day_rain_ratio_ema
z = 5.4177, n = 40, p-value = 6.036e-08
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 466.0000000 7366.6666667    0.5974359
```

```r
sens.slope(annual_Day$day_rain_ratio_ema)
```

```
    Sen's slope

data:  annual_Day$day_rain_ratio_ema
z = 5.4177, n = 40, p-value = 6.036e-08
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.001253790 0.002368411
sample estimates:
Sen's slope
  0.0017924
```

```r
pettitt.test(annual_Day$day_rain_ratio_ema)
```

```
    Pettitt's test for single change-point detection

data:  annual_Day$day_rain_ratio_ema
U* = 368, p-value = 8.35e-06
alternative hypothesis: two.sided
sample estimates:
```

```
probable change point at time K
                             16
```

```
x <- cpt.meanvar(annual_Day$day_rain_ratio_ema, method="AMOC", minseglen = 10)# identifies c
param.est(x)
```

```
$mean
[1] 0.6809435 0.7250185

$variance
[1] 0.0004474687 0.0003826883
```

```
#### for night period
annual_Night <- climate_cleaned %>%
  group_by(year) %>%
  filter( period=="Night" ) %>%
  summarize(rain_ratio= mean(phaseCorrectRainRatio)) %>%
  ungroup() %>%
  mutate(night_rain_ratio_ema = movavg(rain_ratio, n=5, type='e'))

mk.test(annual_Night$night_rain_ratio_ema)
```

```

    Mann-Kendall trend test

data:  annual_Night$night_rain_ratio_ema
z = 3.5303, n = 40, p-value = 0.0004151
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 304.0000000 7366.6666667    0.3897436
```

```
sens.slope(annual_Night$night_rain_ratio_ema)
```

```

    Sen's slope

data:  annual_Night$night_rain_ratio_ema
z = 3.5303, n = 40, p-value = 0.0004151
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.0005336645 0.0015619942
sample estimates:
Sen's slope
0.001046205
```

```
pettitt.test(annual_Night$night_rain_ratio_ema)
```

```
	Pettitt's test for single change-point detection

data:  annual_Night$night_rain_ratio_ema
U* = 343, p-value = 4.244e-05
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             15
```

```
x <- cpt.meanvar(annual_Night$night_rain_ratio_ema, method="AMOC", minseglen = 10)# identifi
param.est(x)
```

```
$mean
[1] 0.5408739 0.5805139

$variance
[1] 0.0007946858 0.0001063264
```

```
# ggplot does not have readymade function for sens slope.. so manual model is created for da
# so that it can be used in visualization part
mkmodel1 <- mblm(day_rain_ratio_ema ~ year, data=annual_Day) #for day
mkmodel2 <- mblm(night_rain_ratio_ema ~ year, data=annual_Night) #for night

# this data is required for plotting for day and night averages
annual_mean <- climate_cleaned %>%
  group_by(year, period) %>%
  filter(p.1 != 0 ) %>%
  summarize(annual_average= mean(phaseCorrectRainRatio))
```

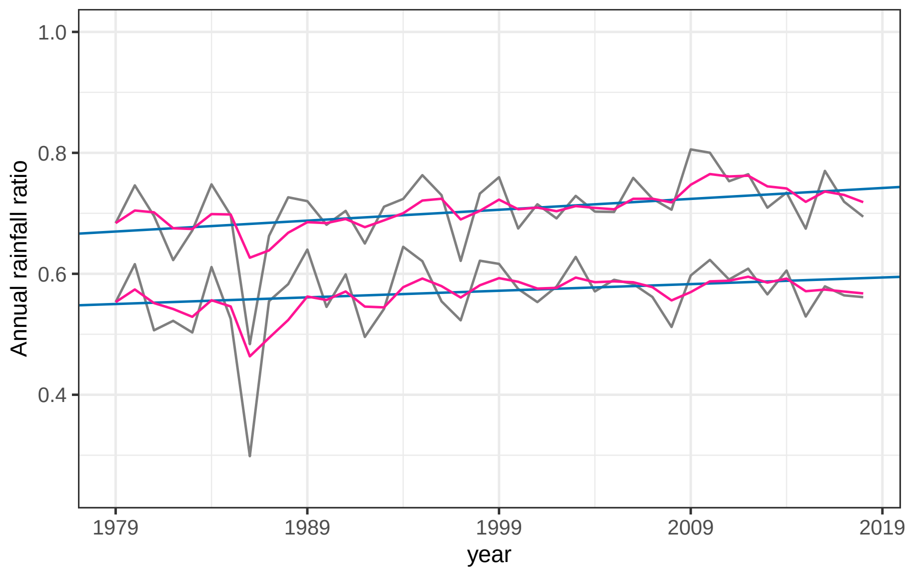
```
`summarise()` has grouped output by 'year'. You can override using the
`.groups` argument.
```

```
# plotting trend#for annual day and night
ggplot(annual_mean, aes(year, annual_average, color= period))+
  geom_line()+
  theme_bw()+
  geom_abline(intercept = coef(mkmodel1)[1], slope = coef(mkmodel1)[2], color = '#0072B2')+
  geom_abline(intercept = coef(mkmodel2)[1], slope = coef(mkmodel2)[2], color = '#0072B2')+
  ylab("Annual rainfall ratio")+
  ylim(0.25, 1)+
  scale_x_continuous(breaks = seq(1979, 2019, by = 10))+
  theme(axis.title = element_text(size = 10))+
  geom_line(data=annual_Day, aes(year, day_rain_ratio_ema), color="deeppink")+
  #annotate("text", x = 1999, y = 0.25, label = "sen's slope for diurnal period= 0.0014, p v
  geom_line(data =annual_Night, aes(year, night_rain_ratio_ema), color="deeppink")+  #454545
  #annotate("text", x = 1999, y = 0.3, label = "sen's slope for night period= 0.0008, p valu
  scale_colour_manual(name= "", values=colors)+
```

```
  theme(legend.position="top")+
  scale_color_manual(values = c('col1' = '#0072B2', 'col2' = '#E69F00'))
```

```
Scale for colour is already present.
Adding another scale for colour, which will replace the existing scale.
```

```
Warning: No shared levels found between `names(values)` of the manual scale and the
data's colour values.
No shared levels found between `names(values)` of the manual scale and the
data's colour values.
```




```
ggsave("plots/day-and-night.tiff", height=5, width=4) # saves the previously created ggplot
```

```
Warning: No shared levels found between `names(values)` of the manual scale and the
data's colour values.
No shared levels found between `names(values)` of the manual scale and the
data's colour values.
```

```
# for august and for all other months
#caluculating rainfall ratio trend for aug
# for other months, need to change month value to different one

monthwise_test <- function(x, y) {
  filtered_data <- climate_cleaned %>%
    filter(month == x &  period==y) %>%
    group_by(year, month, period) %>%
    summarize(mean = mean(phaseCorrectRainRatio)) %>%
    ungroup() %>%
```

```r
    mutate(mean_ema = movavg(mean, n=5, type='e'))
  return(list(filtered_data, mk.test(filtered_data$mean_ema), sens.slope(filtered_data$mean_
              pettitt.test(filtered_data$mean_ema), param.est(cpt.meanvar(filtered_data$mean
}

monthwise_test("Jan", "Day")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.

[[1]]
# A tibble: 37 x 5
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Jan   Day        0        0
 2  1980 Jan   Day        0        0
 3  1981 Jan   Day        0        0
 4  1982 Jan   Day        0        0
 5  1983 Jan   Day        0        0
 6  1984 Jan   Day        0        0
 7  1985 Jan   Day        0        0
 8  1986 Jan   Day        0        0
 9  1987 Jan   Day        0        0
10  1988 Jan   Day        0        0
# i 27 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = -0.47763, n = 37, p-value = 0.6329
alternative hypothesis: true S is not equal to 0
sample estimates:
            S          varS           tau
 -37.00000000 5681.00000000   -0.05800214


[[3]]

    Sen's slope

data:  filtered_data$mean_ema
z = -0.47763, n = 37, p-value = 0.6329
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
```

```
 -4.329419e-07  2.259132e-09
sample estimates:
Sen's slope
          0


[[4]]

    Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 286, p-value = 0.0001599
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             11


[[5]]
[[5]]$mean
[1] 5.175692e-05 6.032546e-07

[[5]]$variance
[1] 7.125528e-09 5.230504e-13
```

```r
monthwise_test("Feb", "Day")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.

[[1]]
# A tibble: 38 x 5
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Feb   Day        0        0
 2  1980 Feb   Day        0        0
 3  1981 Feb   Day        0        0
 4  1982 Feb   Day        0        0
 5  1983 Feb   Day        0        0
 6  1984 Feb   Day        0        0
 7  1985 Feb   Day        0        0
 8  1986 Feb   Day        0        0
 9  1987 Feb   Day        0        0
10  1988 Feb   Day        0        0
# i 28 more rows

[[2]]
```

```
	Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 2.5188, n = 38, p-value = 0.01177
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 196.0000000 5993.3333333    0.2988152


[[3]]

	Sen's slope

data:  filtered_data$mean_ema
z = 2.5188, n = 38, p-value = 0.01177
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.000000e+00 7.827879e-06
sample estimates:
 Sen's slope
3.147718e-06


[[4]]

	Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 336, p-value = 1.195e-05
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             14


[[5]]
[[5]]$mean
[1] 0.0000000000 0.0002767242

[[5]]$variance
[1] 0.000000e+00 3.949226e-08
```

```
monthwise_test("Mar", "Day")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```
[[1]]
# A tibble: 40 x 5
    year month period     mean mean_ema
   <dbl> <ord> <chr>     <dbl>    <dbl>
 1  1979 Mar   Day    0.000159 0.000159
 2  1980 Mar   Day    0        0.000106
 3  1981 Mar   Day    0.00314  0.00112
 4  1982 Mar   Day    0.00778  0.00334
 5  1983 Mar   Day    0.000435 0.00237
 6  1984 Mar   Day    0.118    0.0411
 7  1985 Mar   Day    0.0138   0.0320
 8  1986 Mar   Day    0        0.0213
 9  1987 Mar   Day    0.0369   0.0265
10  1988 Mar   Day    0        0.0177
# i 30 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 2.2253, n = 40, p-value = 0.02606
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 192.0000000 7366.6666667    0.2461538


[[3]]

    Sen's slope

data:  filtered_data$mean_ema
z = 2.2253, n = 40, p-value = 0.02606
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.0001151312 0.0008043902
sample estimates:
 Sen's slope
0.0004462775


[[4]]
```

```
    Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 225, p-value = 0.0195
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             15


[[5]]
[[5]]$mean
[1] 0.02419907

[[5]]$variance
[1] 0.0003142598
```

```
monthwise_test("Apr", "Day")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.

[[1]]
# A tibble: 40 x 5
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Apr   Day    0.184    0.184
 2  1980 Apr   Day    0.268    0.212
 3  1981 Apr   Day    0.355    0.260
 4  1982 Apr   Day    0.140    0.220
 5  1983 Apr   Day    0.151    0.197
 6  1984 Apr   Day    0.192    0.195
 7  1985 Apr   Day    0.338    0.243
 8  1986 Apr   Day    0.118    0.201
 9  1987 Apr   Day    0.138    0.180
10  1988 Apr   Day    0.176    0.179
# i 30 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 2.3419, n = 40, p-value = 0.01919
alternative hypothesis: true S is not equal to 0
sample estimates:
```

```
          S         varS          tau
 202.0000000 7366.6666667    0.2589744


[[3]]

	Sen's slope

data:  filtered_data$mean_ema
z = 2.3419, n = 40, p-value = 0.01919
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.000309825 0.003162061
sample estimates:
Sen's slope
0.001769415


[[4]]

	Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 295, p-value = 0.0006986
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             21


[[5]]
[[5]]$mean
[1] 0.1873154 0.2547889

[[5]]$variance
[1] 0.001046581 0.002431502
```

```
monthwise_test("May", "Day")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```
[[1]]
# A tibble: 40 x 5
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 May   Day    0.806    0.806
```

```
 2  1980 May   Day    0.696    0.770
 3  1981 May   Day    0.716    0.752
 4  1982 May   Day    0.641    0.715
 5  1983 May   Day    0.625    0.685
 6  1984 May   Day    0.782    0.717
 7  1985 May   Day    0.737    0.724
 8  1986 May   Day    0.561    0.670
 9  1987 May   Day    0.504    0.614
10  1988 May   Day    0.789    0.673
# i 30 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 0.99034, n = 40, p-value = 0.322
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
  86.0000000 7366.6666667    0.1102564


[[3]]

    Sen's slope

data:  filtered_data$mean_ema
z = 0.99034, n = 40, p-value = 0.322
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 -0.0005919162  0.0016300767
sample estimates:
Sen's slope
0.000534731


[[4]]

    Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 208, p-value = 0.03824
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
```

```
[[5]]
[[5]]$mean
[1] 0.7221465

[[5]]$variance
[1] 0.001301978
```

```
monthwise_test("Jun", "Day")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```
[[1]]
# A tibble: 40 x 5
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Jun   Day    0.925    0.925
 2  1980 Jun   Day    0.924    0.924
 3  1981 Jun   Day    0.926    0.925
 4  1982 Jun   Day    0.895    0.915
 5  1983 Jun   Day    0.843    0.891
 6  1984 Jun   Day    0.960    0.914
 7  1985 Jun   Day    0.889    0.905
 8  1986 Jun   Day    0.358    0.723
 9  1987 Jun   Day    0.878    0.775
10  1988 Jun   Day    0.878    0.809
# i 30 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 3.0642, n = 40, p-value = 0.002182
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 264.0000000 7366.6666667    0.3384615


[[3]]

    Sen's slope
```

```
data:  filtered_data$mean_ema
z = 3.0642, n = 40, p-value = 0.002182
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.0002388902 0.0015994686
sample estimates:
 Sen's slope
0.0008399705


[[4]]

    Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 308, p-value = 0.0003411
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             20


[[5]]
[[5]]$mean
[1] 0.8768385 0.9212176

[[5]]$variance
[1] 2.780797e-03 2.985449e-05
```

```
monthwise_test("Jul", "Day")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```
[[1]]
# A tibble: 40 x 5
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Jul   Day    0.972    0.972
 2  1980 Jul   Day    0.987    0.977
 3  1981 Jul   Day    0.987    0.980
 4  1982 Jul   Day    0.980    0.980
 5  1983 Jul   Day    0.944    0.968
 6  1984 Jul   Day    0.980    0.972
 7  1985 Jul   Day    0.950    0.965
 8  1986 Jul   Day    0.334    0.754
 9  1987 Jul   Day    0.964    0.824
```

```
10  1988 Jul   Day    0.986    0.878
# i 30 more rows

[[2]]

	Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 1.8758, n = 40, p-value = 0.06068
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 162.0000000 7366.6666667    0.2076923


[[3]]

	Sen's slope

data:  filtered_data$mean_ema
z = 1.8758, n = 40, p-value = 0.06068
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 -0.0000102805  0.0007787939
sample estimates:
 Sen's slope
0.0002676427


[[4]]

	Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 264, p-value = 0.003408
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             18


[[5]]
[[5]]$mean
[1] 0.9293460 0.9752049

[[5]]$variance
```

```
[1] 4.244757e-03 3.879712e-05
```

```r
monthwise_test("Aug", "Day")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.

[[1]]
# A tibble: 40 x 5
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Aug   Day    0.941    0.941
 2  1980 Aug   Day    0.975    0.952
 3  1981 Aug   Day    0.968    0.957
 4  1982 Aug   Day    0.976    0.964
 5  1983 Aug   Day    0.950    0.959
 6  1984 Aug   Day    0.923    0.947
 7  1985 Aug   Day    0.982    0.959
 8  1986 Aug   Day    0.958    0.959
 9  1987 Aug   Day    0.919    0.945
10  1988 Aug   Day    0.976    0.956
# i 30 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 5.9304, n = 40, p-value = 3.022e-09
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 510.0000000 7366.6666667    0.6538462


[[3]]

    Sen's slope

data:  filtered_data$mean_ema
z = 5.9304, n = 40, p-value = 3.022e-09
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.0007104811 0.0011050136
sample estimates:
 Sen's slope
0.0009288176
```

```
[[4]]

    Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 379, p-value = 3.938e-06
alternative hypothesis: two.sided
sample estimates:
probable change point at time K                                <NA>
                             17                                  19


[[5]]
[[5]]$mean
[1] 0.9556842 0.9750562

[[5]]$variance
[1] 4.765227e-05 4.031308e-05
```

```
monthwise_test("Sep", "Day")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```
[[1]]
# A tibble: 40 x 5
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Sep   Day    0.892    0.892
 2  1980 Sep   Day    0.939    0.908
 3  1981 Sep   Day    0.918    0.911
 4  1982 Sep   Day    0.891    0.905
 5  1983 Sep   Day    0.930    0.913
 6  1984 Sep   Day    0.930    0.919
 7  1985 Sep   Day    0.908    0.915
 8  1986 Sep   Day    0.878    0.903
 9  1987 Sep   Day    0.901    0.902
10  1988 Sep   Day    0.907    0.904
# i 30 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
```

```
z = 2.3419, n = 40, p-value = 0.01919
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 202.0000000 7366.6666667    0.2589744


[[3]]

    Sen's slope

data:  filtered_data$mean_ema
z = 2.3419, n = 40, p-value = 0.01919
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 4.676159e-05 5.957547e-04
sample estimates:
 Sen's slope
0.0003367713


[[4]]

    Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 263, p-value = 0.003577
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             15


[[5]]
[[5]]$mean
[1] 0.9159204

[[5]]$variance
[1] 9.816127e-05
```

```r
monthwise_test("Oct", "Day")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```
[[1]]
# A tibble: 39 x 5
```

```
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Oct   Day    0.433    0.433
 2  1980 Oct   Day    0.320    0.395
 3  1981 Oct   Day    0.606    0.466
 4  1982 Oct   Day    0.187    0.373
 5  1983 Oct   Day    0.577    0.441
 6  1984 Oct   Day    0.692    0.525
 7  1985 Oct   Day    0.448    0.499
 8  1986 Oct   Day    0.197    0.398
 9  1987 Oct   Day    0.341    0.379
10  1988 Oct   Day    0.443    0.400
# i 29 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 3, n = 39, p-value = 0.0027
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 249.0000000 6833.6666667    0.3360324


[[3]]

    Sen's slope

data:  filtered_data$mean_ema
z = 3, n = 39, p-value = 0.0027
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.001280663 0.005457376
sample estimates:
Sen's slope
0.003448929


[[4]]

    Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 270, p-value = 0.001509
```

```
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             13


[[5]]
[[5]]$mean
[1] 0.4277332 0.5304438

[[5]]$variance
[1] 0.002212934 0.003532080
```

```
monthwise_test("Nov", "Day")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.

[[1]]
# A tibble: 33 x 5
    year month period     mean mean_ema
   <dbl> <ord> <chr>     <dbl>    <dbl>
 1  1979 Nov   Day    0.0268    0.0268
 2  1980 Nov   Day    0.0120    0.0218
 3  1981 Nov   Day    0.0193    0.0210
 4  1982 Nov   Day    0.00637   0.0161
 5  1983 Nov   Day    0.000333  0.0109
 6  1984 Nov   Day    0         0.00724
 7  1985 Nov   Day    0.0210    0.0118
 8  1986 Nov   Day    0.0129    0.0122
 9  1987 Nov   Day    0         0.00812
10  1988 Nov   Day    0         0.00541
# i 23 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 3.1454, n = 33, p-value = 0.001659
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS           tau
 204.0000000 4165.3333333     0.3863636


[[3]]
```

```
    Sen's slope

data:  filtered_data$mean_ema
z = 3.1454, n = 33, p-value = 0.001659
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.001147359 0.004193464
sample estimates:
Sen's slope
0.002700871


[[4]]

    Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 272, p-value = 1.242e-05
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             17


[[5]]
[[5]]$mean
[1] 0.01422831 0.09318854

[[5]]$variance
[1] 7.172363e-05 8.417790e-04
```

```
monthwise_test("Dec", "Day")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```
[[1]]
# A tibble: 36 x 5
    year month period     mean  mean_ema
   <dbl> <ord> <chr>     <dbl>     <dbl>
 1  1979 Dec   Day    0        0
 2  1980 Dec   Day    0        0
 3  1981 Dec   Day    0        0
 4  1982 Dec   Day    0        0
 5  1983 Dec   Day    0        0
 6  1984 Dec   Day    0.000638 0.000213
```

```
 7  1985 Dec   Day    0        0.000142
 8  1986 Dec   Day    0        0.0000946
 9  1987 Dec   Day    0        0.0000630
10  1988 Dec   Day    0        0.0000420
# i 26 more rows

[[2]]

	Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 1.9781, n = 36, p-value = 0.04792
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 146.0000000 5373.3333333    0.2336075


[[3]]

	Sen's slope

data:  filtered_data$mean_ema
z = 1.9781, n = 36, p-value = 0.04792
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.00000e+00 5.85531e-06
sample estimates:
 Sen's slope
1.105771e-06


[[4]]

	Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 267, p-value = 0.0002674
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             19


[[5]]
[[5]]$mean
```

```
[1] 3.347955e-05 5.321244e-04

[[5]]$variance
[1] 3.167745e-09 4.906163e-07
```

```
monthwise_test("Jan", "Night")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.

[[1]]
# A tibble: 37 x 5
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Jan   Night      0        0
 2  1980 Jan   Night      0        0
 3  1981 Jan   Night      0        0
 4  1982 Jan   Night      0        0
 5  1983 Jan   Night      0        0
 6  1984 Jan   Night      0        0
 7  1985 Jan   Night      0        0
 8  1986 Jan   Night      0        0
 9  1987 Jan   Night      0        0
10  1988 Jan   Night      0        0
# i 27 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = NaN, n = 37, p-value = NA
alternative hypothesis: true S is not equal to 0
sample estimates:
   S varS  tau
   0    0  NaN


[[3]]

    Sen's slope

data:  filtered_data$mean_ema
z = NaN, n = 37, p-value = NA
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0 0
```

```
sample estimates:
Sen's slope
          0


[[4]]

    Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 0, p-value = 1
alternative hypothesis: two.sided
sample estimates:
probable change point at time K                                <NA>
                              1                                   2
                           <NA>                                <NA>
                              3                                   4
                           <NA>                                <NA>
                              5                                   6
                           <NA>                                <NA>
                              7                                   8
                           <NA>                                <NA>
                              9                                  10
                           <NA>                                <NA>
                             11                                  12
                           <NA>                                <NA>
                             13                                  14
                           <NA>                                <NA>
                             15                                  16
                           <NA>                                <NA>
                             17                                  18
                           <NA>                                <NA>
                             19                                  20
                           <NA>                                <NA>
                             21                                  22
                           <NA>                                <NA>
                             23                                  24
                           <NA>                                <NA>
                             25                                  26
                           <NA>                                <NA>
                             27                                  28
                           <NA>                                <NA>
                             29                                  30
                           <NA>                                <NA>
                             31                                  32
                           <NA>                                <NA>
```

```
                                 33                               34
                               <NA>                             <NA>
                                 35                               36
                               <NA>
                                 37


[[5]]
[[5]]$mean
[1] 0

[[5]]$variance
[1] 0
```

```r
monthwise_test("Feb", "Night")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.

[[1]]
# A tibble: 38 x 5
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Feb   Night      0        0
 2  1980 Feb   Night      0        0
 3  1981 Feb   Night      0        0
 4  1982 Feb   Night      0        0
 5  1983 Feb   Night      0        0
 6  1984 Feb   Night      0        0
 7  1985 Feb   Night      0        0
 8  1986 Feb   Night      0        0
 9  1987 Feb   Night      0        0
10  1988 Feb   Night      0        0
# i 28 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = NaN, n = 38, p-value = NA
alternative hypothesis: true S is not equal to 0
sample estimates:
   S varS  tau
   0    0  NaN
```

```
[[3]]

	Sen's slope

data:  filtered_data$mean_ema
z = NaN, n = 38, p-value = NA
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0 0
sample estimates:
Sen's slope
          0


[[4]]

	Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 0, p-value = 1
alternative hypothesis: two.sided
sample estimates:
probable change point at time K                             <NA>
                              1                                2
                           <NA>                             <NA>
                              3                                4
                           <NA>                             <NA>
                              5                                6
                           <NA>                             <NA>
                              7                                8
                           <NA>                             <NA>
                              9                               10
                           <NA>                             <NA>
                             11                               12
                           <NA>                             <NA>
                             13                               14
                           <NA>                             <NA>
                             15                               16
                           <NA>                             <NA>
                             17                               18
                           <NA>                             <NA>
                             19                               20
                           <NA>                             <NA>
                             21                               22
                           <NA>                             <NA>
                             23                               24
```

```
                        <NA>                        <NA>
                          25                          26
                        <NA>                        <NA>
                          27                          28
                        <NA>                        <NA>
                          29                          30
                        <NA>                        <NA>
                          31                          32
                        <NA>                        <NA>
                          33                          34
                        <NA>                        <NA>
                          35                          36
                        <NA>                        <NA>
                          37                          38


[[5]]
[[5]]$mean
[1] 0

[[5]]$variance
[1] 0
```

```
monthwise_test("Mar", "Night")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```
[[1]]
# A tibble: 40 x 5
    year month period     mean  mean_ema
   <dbl> <ord> <chr>     <dbl>     <dbl>
 1  1979 Mar   Night  0        0
 2  1980 Mar   Night  0        0
 3  1981 Mar   Night  0        0
 4  1982 Mar   Night  0        0
 5  1983 Mar   Night  0        0
 6  1984 Mar   Night  0.000263 0.0000877
 7  1985 Mar   Night  0        0.0000585
 8  1986 Mar   Night  0        0.0000390
 9  1987 Mar   Night  0        0.0000260
10  1988 Mar   Night  0        0.0000173
# i 30 more rows

[[2]]

    Mann-Kendall trend test
```

```
data:  filtered_data$mean_ema
z = 3.4176, n = 40, p-value = 0.0006317
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 294.0000000 7350.0000000    0.3793627


[[3]]

    Sen's slope

data:  filtered_data$mean_ema
z = 3.4176, n = 40, p-value = 0.0006317
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 1.369705e-06 6.441232e-06
sample estimates:
 Sen's slope
2.854479e-06


[[4]]

    Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 325, p-value = 0.0001275
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             15


[[5]]
[[5]]$mean
[1] 2.592732e-05 2.678801e-04

[[5]]$variance
[1] 7.373442e-10 9.801105e-08
```

```r
monthwise_test("Apr", "Night")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```
[[1]]
# A tibble: 40 x 5
    year month period      mean mean_ema
   <dbl> <ord> <chr>      <dbl>    <dbl>
 1  1979 Apr   Night  0.00506    0.00506
 2  1980 Apr   Night  0.00328    0.00447
 3  1981 Apr   Night  0.00351    0.00415
 4  1982 Apr   Night  0          0.00277
 5  1983 Apr   Night  0.00108    0.00220
 6  1984 Apr   Night  0.000882   0.00176
 7  1985 Apr   Night  0.00909    0.00421
 8  1986 Apr   Night  0.000167   0.00286
 9  1987 Apr   Night  0.0000662  0.00193
10  1988 Apr   Night  0.00112    0.00166
# i 30 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 2.0622, n = 40, p-value = 0.03919
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 178.0000000 7366.6666667    0.2282051


[[3]]

    Sen's slope

data:  filtered_data$mean_ema
z = 2.0622, n = 40, p-value = 0.03919
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 2.329045e-06 1.179127e-04
sample estimates:
 Sen's slope
5.679043e-05


[[4]]

    Pettitt's test for single change-point detection
```

```
data:  filtered_data$mean_ema
U* = 278, p-value = 0.001703
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             16


[[5]]
[[5]]$mean
[1] 0.002774721 0.005194864

[[5]]$variance
[1] 1.276670e-06 3.917176e-06
```

```
monthwise_test("May", "Night")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.

[[1]]
# A tibble: 40 x 5
    year month period   mean mean_ema
   <dbl> <ord> <chr>   <dbl>    <dbl>
 1  1979 May   Night  0.222     0.222
 2  1980 May   Night  0.0973    0.181
 3  1981 May   Night  0.147     0.169
 4  1982 May   Night  0.0366    0.125
 5  1983 May   Night  0.0676    0.106
 6  1984 May   Night  0.311     0.174
 7  1985 May   Night  0.139     0.163
 8  1986 May   Night  0.0160    0.114
 9  1987 May   Night  0.0996    0.109
10  1988 May   Night  0.222     0.147
# i 30 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 2.5516, n = 40, p-value = 0.01072
alternative hypothesis: true S is not equal to 0
sample estimates:
          S         varS          tau
 220.0000000 7366.6666667    0.2820513
```

```
[[3]]

    Sen's slope

data:  filtered_data$mean_ema
z = 2.5516, n = 40, p-value = 0.01072
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.000418345 0.002771350
sample estimates:
Sen's slope
0.001552632


[[4]]

    Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 294, p-value = 0.0007372
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             16


[[5]]
[[5]]$mean
[1] 0.1611015 0.2188621

[[5]]$variance
[1] 0.001101112 0.001231344
```

```r
monthwise_test("Jun", "Night")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.

[[1]]
# A tibble: 40 x 5
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Jun   Night  0.778    0.778
 2  1980 Jun   Night  0.835    0.797
 3  1981 Jun   Night  0.771    0.788
 4  1982 Jun   Night  0.698    0.758
```

```
 5  1983 Jun   Night  0.536    0.684
 6  1984 Jun   Night  0.877    0.749
 7  1985 Jun   Night  0.660    0.719
 8  1986 Jun   Night  0.172    0.537
 9  1987 Jun   Night  0.648    0.574
10  1988 Jun   Night  0.631    0.593
# i 30 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 1.6894, n = 40, p-value = 0.09114
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 146.0000000 7366.6666667    0.1871795


[[3]]

    Sen's slope

data:  filtered_data$mean_ema
z = 1.6894, n = 40, p-value = 0.09114
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 -0.0001296932  0.0031017261
sample estimates:
Sen's slope
0.001397693


[[4]]

    Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 243, p-value = 0.009026
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             19
```

```
[[5]]
[[5]]$mean
[1] 0.6907840 0.7651868

[[5]]$variance
[1] 0.0059388573 0.0006421996
```

```
monthwise_test("Jul", "Night")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```
[[1]]
# A tibble: 40 x 5
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Jul   Night  0.952    0.952
 2  1980 Jul   Night  0.978    0.961
 3  1981 Jul   Night  0.981    0.968
 4  1982 Jul   Night  0.968    0.968
 5  1983 Jul   Night  0.890    0.942
 6  1984 Jul   Night  0.961    0.948
 7  1985 Jul   Night  0.887    0.928
 8  1986 Jul   Night  0.207    0.688
 9  1987 Jul   Night  0.944    0.773
10  1988 Jul   Night  0.980    0.842
# i 30 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 2.202, n = 40, p-value = 0.02766
alternative hypothesis: true S is not equal to 0
sample estimates:
          S         varS          tau
 190.0000000 7366.6666667    0.2435897


[[3]]

    Sen's slope

data:  filtered_data$mean_ema
z = 2.202, n = 40, p-value = 0.02766
alternative hypothesis: true z is not equal to 0
```

```
95 percent confidence interval:
 8.030745e-05 1.524302e-03
sample estimates:
 Sen's slope
0.0006727866


[[4]]

	Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 276, p-value = 0.001884
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             18


[[5]]
[[5]]$mean
[1] 0.8986298 0.9571095

[[5]]$variance
[1] 0.006193057 0.000177692
```

```r
monthwise_test("Aug", "Night")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.

[[1]]
# A tibble: 40 x 5
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Aug   Night  0.858    0.858
 2  1980 Aug   Night  0.943    0.887
 3  1981 Aug   Night  0.945    0.906
 4  1982 Aug   Night  0.956    0.923
 5  1983 Aug   Night  0.894    0.913
 6  1984 Aug   Night  0.780    0.869
 7  1985 Aug   Night  0.957    0.898
 8  1986 Aug   Night  0.892    0.896
 9  1987 Aug   Night  0.826    0.873
10  1988 Aug   Night  0.950    0.898
# i 30 more rows
```

```
[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 5.3711, n = 40, p-value = 7.825e-08
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
 462.0000000 7366.6666667    0.5923077


[[3]]

    Sen's slope

data:  filtered_data$mean_ema
z = 5.3711, n = 40, p-value = 7.825e-08
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.001543072 0.002727062
sample estimates:
Sen's slope
0.002047942


[[4]]

    Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 369, p-value = 7.806e-06
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             21


[[5]]
[[5]]$mean
[1] 0.9079254 0.9527947

[[5]]$variance
[1] 0.0004710678 0.0001967934
```

```r
monthwise_test("Sep", "Night")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.

[[1]]
# A tibble: 40 x 5
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Sep   Night  0.600    0.600
 2  1980 Sep   Night  0.708    0.636
 3  1981 Sep   Night  0.564    0.612
 4  1982 Sep   Night  0.514    0.579
 5  1983 Sep   Night  0.799    0.652
 6  1984 Sep   Night  0.703    0.669
 7  1985 Sep   Night  0.624    0.654
 8  1986 Sep   Night  0.491    0.600
 9  1987 Sep   Night  0.665    0.622
10  1988 Sep   Night  0.681    0.641
# i 30 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 0.68741, n = 40, p-value = 0.4918
alternative hypothesis: true S is not equal to 0
sample estimates:
           S         varS          tau
6.000000e+01 7.366667e+03 7.692308e-02


[[3]]

    Sen's slope

data:  filtered_data$mean_ema
z = 0.68741, n = 40, p-value = 0.4918
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 -0.0007224173  0.0014701522
sample estimates:
 Sen's slope
0.0003867111
```

```
[[4]]

	Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 202, p-value = 0.04789
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             10


[[5]]
[[5]]$mean
[1] 0.6587073

[[5]]$variance
[1] 0.00133305
```

```
monthwise_test("Oct", "Night")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.

[[1]]
# A tibble: 40 x 5
    year month period     mean mean_ema
   <dbl> <ord> <chr>     <dbl>    <dbl>
 1  1979 Oct   Night  0.0770     0.0770
 2  1980 Oct   Night  0.000563   0.0515
 3  1981 Oct   Night  0.0318     0.0450
 4  1982 Oct   Night  0.000154   0.0300
 5  1983 Oct   Night  0.0668     0.0423
 6  1984 Oct   Night  0.0483     0.0443
 7  1985 Oct   Night  0.0418     0.0435
 8  1986 Oct   Night  0.00423    0.0304
 9  1987 Oct   Night  0.0177     0.0262
10  1988 Oct   Night  0.0429     0.0318
# i 30 more rows

[[2]]

	Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 4.2759, n = 40, p-value = 1.903e-05
```

```
alternative hypothesis: true S is not equal to 0
sample estimates:
          S          varS           tau
 368.0000000 7366.6666667    0.4717949


[[3]]

	Sen's slope

data:  filtered_data$mean_ema
z = 4.2759, n = 40, p-value = 1.903e-05
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.001142253 0.002745751
sample estimates:
Sen's slope
0.001969209


[[4]]

	Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 375, p-value = 5.189e-06
alternative hypothesis: two.sided
sample estimates:
probable change point at time K
                             19


[[5]]
[[5]]$mean
[1] 0.0437565 0.1021342

[[5]]$variance
[1] 0.0001488208 0.0021234646
```

```r
monthwise_test("Nov", "Night")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```
[[1]]
# A tibble: 31 x 5
    year month period  mean mean_ema
```

```
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Nov   Night      0        0
 2  1980 Nov   Night      0        0
 3  1981 Nov   Night      0        0
 4  1982 Nov   Night      0        0
 5  1983 Nov   Night      0        0
 6  1984 Nov   Night      0        0
 7  1985 Nov   Night      0        0
 8  1986 Nov   Night      0        0
 9  1988 Nov   Night      0        0
10  1989 Nov   Night      0        0
# i 21 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = 3.9863, n = 31, p-value = 6.712e-05
alternative hypothesis: true S is not equal to 0
sample estimates:
          S         varS          tau
 206.0000000 2644.6666667    0.5571435


[[3]]

    Sen's slope

data:  filtered_data$mean_ema
z = 3.9863, n = 31, p-value = 6.712e-05
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0.00000e+00 1.65529e-06
sample estimates:
 Sen's slope
7.765558e-07


[[4]]

    Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 228, p-value = 7.874e-05
alternative hypothesis: two.sided
```

```
sample estimates:
probable change point at time K
                             19


[[5]]
[[5]]$mean
[1] 0.000000e+00 4.501882e-05

[[5]]$variance
[1] 0.000000e+00 7.645748e-10
```

```
monthwise_test("Dec", "Night")
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.

[[1]]
# A tibble: 33 x 5
    year month period  mean mean_ema
   <dbl> <ord> <chr>  <dbl>    <dbl>
 1  1979 Dec   Night      0        0
 2  1980 Dec   Night      0        0
 3  1981 Dec   Night      0        0
 4  1982 Dec   Night      0        0
 5  1983 Dec   Night      0        0
 6  1984 Dec   Night      0        0
 7  1985 Dec   Night      0        0
 8  1986 Dec   Night      0        0
 9  1987 Dec   Night      0        0
10  1988 Dec   Night      0        0
# i 23 more rows

[[2]]

    Mann-Kendall trend test

data:  filtered_data$mean_ema
z = NaN, n = 33, p-value = NA
alternative hypothesis: true S is not equal to 0
sample estimates:
   S varS  tau
   0    0  NaN


[[3]]
```

```
    Sen's slope

data:  filtered_data$mean_ema
z = NaN, n = 33, p-value = NA
alternative hypothesis: true z is not equal to 0
95 percent confidence interval:
 0 0
sample estimates:
Sen's slope
          0


[[4]]

    Pettitt's test for single change-point detection

data:  filtered_data$mean_ema
U* = 0, p-value = 1
alternative hypothesis: two.sided
sample estimates:
probable change point at time K                           <NA>
                              1                              2
                           <NA>                           <NA>
                              3                              4
                           <NA>                           <NA>
                              5                              6
                           <NA>                           <NA>
                              7                              8
                           <NA>                           <NA>
                              9                             10
                           <NA>                           <NA>
                             11                             12
                           <NA>                           <NA>
                             13                             14
                           <NA>                           <NA>
                             15                             16
                           <NA>                           <NA>
                             17                             18
                           <NA>                           <NA>
                             19                             20
                           <NA>                           <NA>
                             21                             22
                           <NA>                           <NA>
                             23                             24
                           <NA>                           <NA>
                             25                             26
```

```
                                    <NA>                              <NA>
                                      27                                28
                                    <NA>                              <NA>
                                      29                                30
                                    <NA>                              <NA>
                                      31                                32
                                    <NA>
                                      33


[[5]]
[[5]]$mean
[1] 0

[[5]]$variance
[1] 0
```

```r
# par(mar = c(1, 1, 1, 1))
# logistic regression for temperature boundary point
# phase is estimated based on rain ratio.. with greater than 0.5 as 1, and less than 0.5 as
#then creating phase column as factor, which is enforced to logistic model
```

```r
#### data for logistic maping
summary_log <- climate_cleaned %>%
  group_by(year, month, day) %>%
  summarize(temp= mean(t.1),
            rh= mean(rh.1),
            wind=mean(u.1),
            pp=mean(p.1),
            rain_ratio= mean(phaseCorrectRainRatio),
            season= unique(season)) %>%
  mutate(phase=as.factor(ifelse(rain_ratio > 0.5, 1, 0)))
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```r
#summary_log$temp_1 <-  rnorm(9546, 0, 0.01) + summary_log$temp
##define the model

log_regression <- glm(phase ~  temp + rh + wind, family = "binomial", data= summary_log)
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(log_regression)
```

```
Call:
glm(formula = phase ~ temp + rh + wind, family = "binomial",
```

```
    data = summary_log)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -19.766061   1.114538 -17.735   <2e-16 ***
temp          2.786350   0.119064  23.402   <2e-16 ***
rh            0.183061   0.009604  19.061   <2e-16 ***
wind         -0.018703   0.046164  -0.405    0.685
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 13106.1  on 9545  degrees of freedom
Residual deviance:  1087.4  on 9542  degrees of freedom
AIC: 1095.4

Number of Fisher Scoring iterations: 10
```

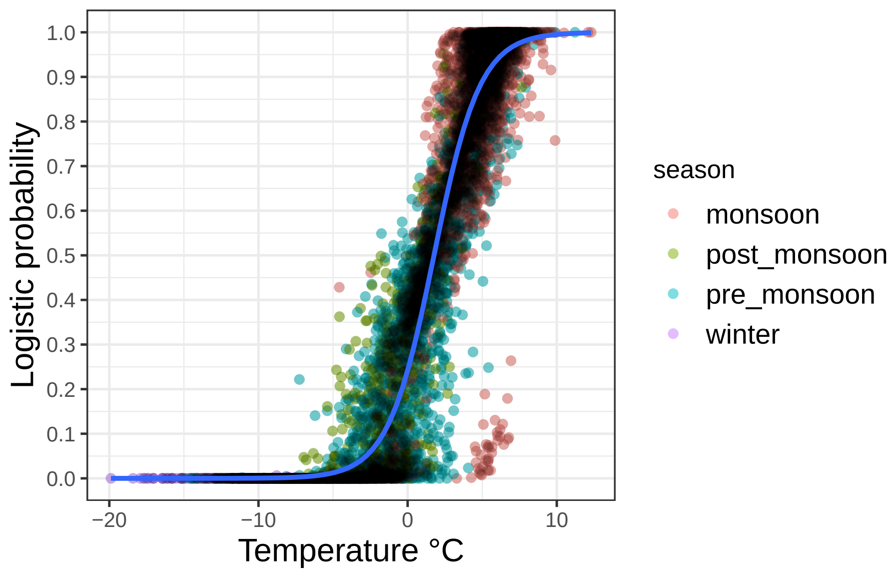
```
vif(log_regression)
```

```
    temp       rh     wind
2.824468 3.186100 1.373769
```

```
# plotting logistic graph between temp and rainfall fraction
ggplot(summary_log, aes(temp, rain_ratio))+
  geom_point(aes(color= season), alpha=0.5)+
  theme_bw()+ ylab("rainfall ratio")+
  scale_y_continuous(breaks = seq(0.00, 1.00, by = 0.10))+
  geom_jitter(width = 0.01, alpha=0.1)+
  geom_smooth(method= 'glm',  method.args=list(family="binomial"))+
  theme(axis.title = element_text(size = 14))+ xlab("Temperature °C")+ ylab("Logistic probab
  theme(legend.text=element_text(size= 12))
```

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

Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```




```
ggsave("plots/logistic.png", width = 7, height= 7, type="cairo", dpi= 300)
```

```
Warning: Using ragg device as default. Ignoring `type` and `antialias`
arguments
```

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

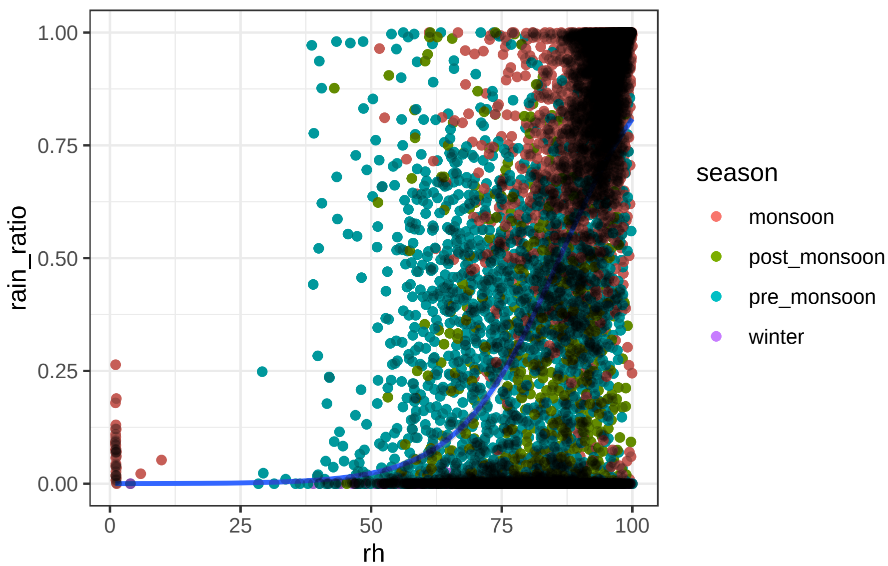
```
Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## visual representation of rain-ratio with rall variables
visual_relation <- function(variable){
  summary_log %>%
    ggplot(aes({{variable}}, rain_ratio))+
    geom_point(aes(color= season))+
    geom_smooth(method= "glm",  method.args=list(family="binomial"))+
    theme_bw()+
    geom_jitter( alpha=0.2)
}

visual_relation(rh)
```

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

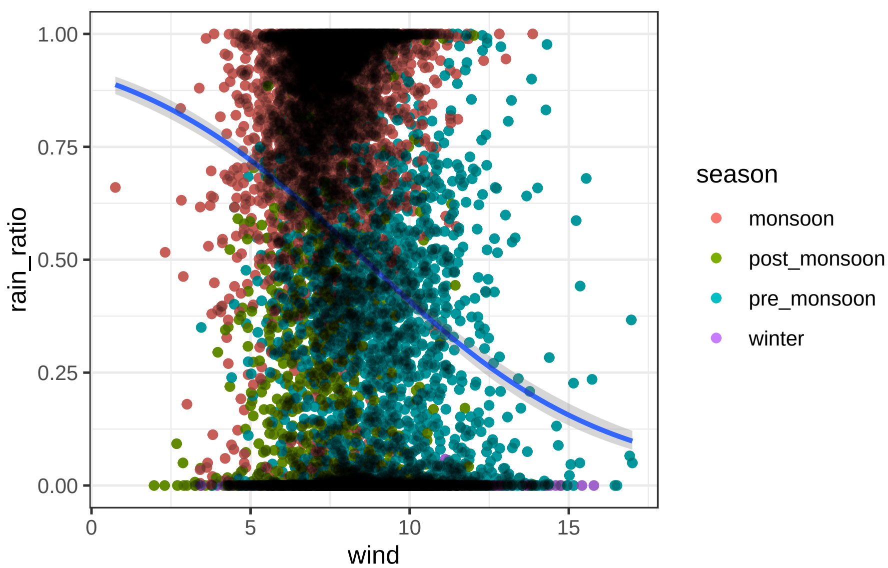
```
Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```




```r
visual_relation(wind)
```

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

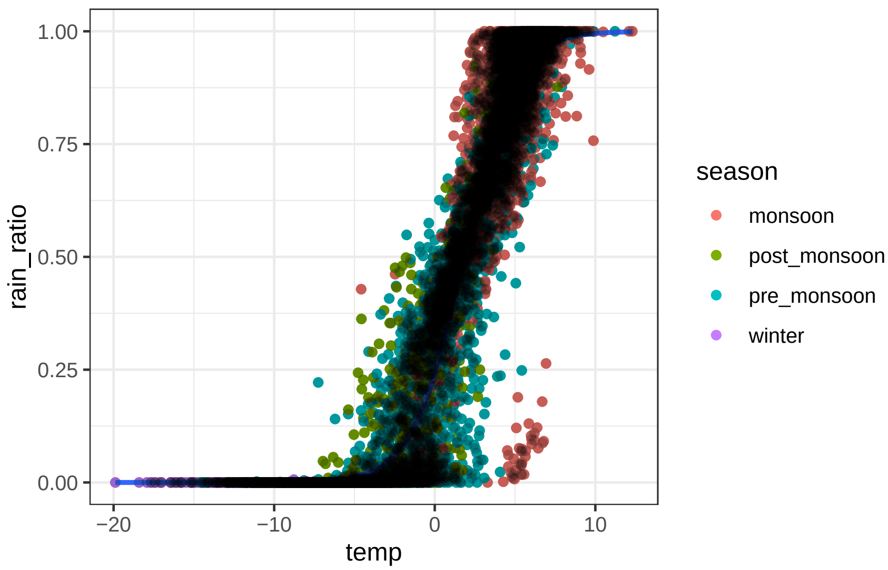
```
Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```




```r
visual_relation(temp)
```

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

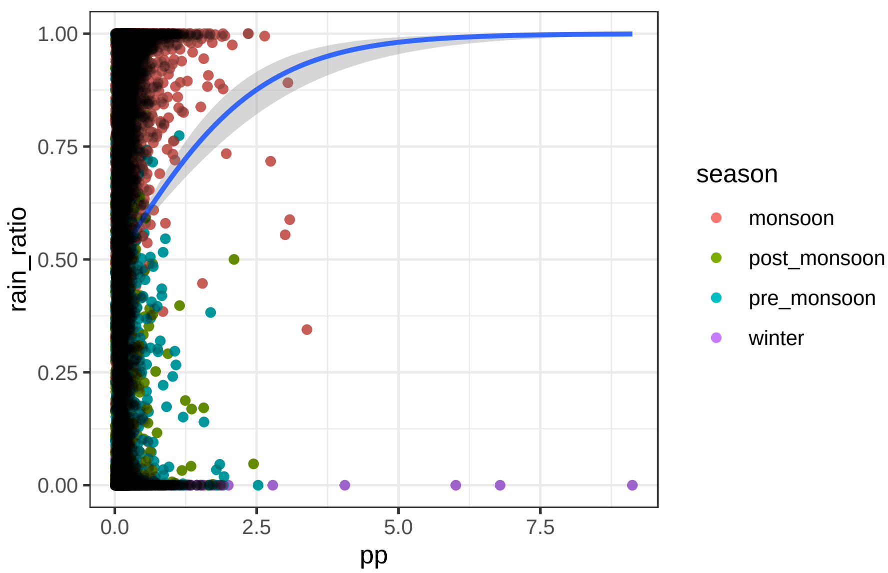
```
Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```




```r
visual_relation(pp)
```

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

```
Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```



```r
# creating graph for sample year 2000
datetime <- seq(
  from = as.Date("1979/01/01"),
  to = as.Date("2018/12/31"),
  by = "days")

data <- clone_copy %>%
  mutate(phaseCorrectRainRatio= round(phaseCorrectRainRatio, digits=2),
         year= year(datetime),
         month= month(datetime, label = TRUE),
         day= day(datetime),
         period= if_else(hour(datetime)>5 & hour(datetime)< 19, "Day", "Night"),
         season = ifelse( month <= "Feb"  | month == "Dec", "winter",
                          ifelse( month == "Mar" | month == "Apr" | month == "May", "pre_mon
                                  ifelse(month == "Oct" | month == "Nov", "post_monsoon", "m
filter(year != 2019)
```

```r
time_series <-  data %>%
                group_by(year, month, day) %>%
                summarize(rain=sum(phaseCorrectRain), snow= sum(phaseCorrectSnow), temp= mea
```

```
`summarise()` has grouped output by 'year', 'month'. You can override using the
`.groups` argument.
```

```r
ts2 <- xts(time_series[4:5], order.by = datetime)
ts2_2000 <- ts2["2000"]

dev.new()
png( "plots/rain and snow.tiff", height = 4, width = 10, units = "in", res = 300)
plot( ts2$rain, lwd = 0.5, xlim=c(1979, 2018), col = "#454545", main = NA,
      yaxis.right = F, ylab = "Precipitation(mm/day)", grid.col=NA, cex=0.7)
lines(ts2$snow, col = 4, lwd = 0.5)
addLegend(legend.loc = "topright", legend.names = c("rainfall", "snowfall"), col = c("#45454
          lty = 1, lwd = 1.5)

axis(side = 4, at = pretty(ts2_2000$phaseCorrectSnow.sum), col = 4, lwd = 1.6)
dev.off()
```

```
pdf
  2
```

```r
############# plot for the sample year
dev.new()
png("plots/rain and snow 2000.tiff", height = 4, width = 10, units = "in", res = 300)

plot(ts2_2000$rain, lwd = 1, xlim(1979, 2018), col = "#454545",
     main = NA, yaxis.right = F, ylab = "Precipitation(mm/day)", cex=0.7,
```

```
     grid.col=NA)
lines(ts2_2000$snow, col = 4,lwd = 1)
addLegend(
  legend.loc = "topright",
  legend.names = c("rainfall", "snowfall"),
  col = c("#454545", 4),
  lty = 1,
  lwd = 1.5)
dev.off()
```

```
pdf
  2
```

```
# month_wise grid plot with 40 lines
lines_graph_maker <- function(calculate, column, ylabel, legend){
  monthly_data <- data %>%
    group_by(year, month) %>%
    summarize(mean = calculate({{column}}, na.rm = TRUE))

  monthly_data$year <- factor(monthly_data$year)
  color_palette <- colorRampPalette(c("#0165fc", "yellow", "red"))(length(unique(monthly_dat

  ggplot(monthly_data, aes(month, mean, group = year, color = year)) +
    geom_line(size = ifelse(between(as.integer(monthly_data$year), 1, 1) |
                            between(as.integer(monthly_data$year), 40, 40), 0.7, 0.2),
              alpha = 1) +
    scale_color_manual(values = color_palette) +
    # geom_line(data = monthly_data[1, ], aes(group = year, color = year), size = 2) +
    # geom_line(data = monthly_data[length(monthly_data$year), ], aes(group = year, color =
    theme_bw() +
    theme(axis.title = element_text(size = 10)) +
    ylab(ylabel) +
    xlab("") +
    theme(legend.position = legend, legend.text = element_text(size = 12))
}

plot1  <- lines_graph_maker(max,  t.1,  "Maximum Daily Temperature °C", legend="none")
```

```
`summarise()` has grouped output by 'year'. You can override using the
`.groups` argument.

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

```
plot2  <- lines_graph_maker(min,  t.1,  "Minimum Daily Temperature °C", legend="none")
```

```
`summarise()` has grouped output by 'year'. You can override using the
```

```
`.groups` argument.
```

```
plot3  <- lines_graph_maker(mean, rh.1,  "Relative Humidity (%)", legend="none")
```

```
`summarise()` has grouped output by 'year'. You can override using the
`.groups` argument.
```

```
plot4  <- lines_graph_maker(mean, u.1,  expression("Wind (ms"^-1*")"), legend="none")
```

```
`summarise()` has grouped output by 'year'. You can override using the
`.groups` argument.
```

```
plot5  <- lines_graph_maker(sum,  p.1,  "Precipitation (mm)", legend="none")
```

```
`summarise()` has grouped output by 'year'. You can override using the
`.groups` argument.
```

```
png("plots/4plots_combined.tiff", height = 8, width = 8, units = "in", res = 300)
grid.arrange(plot1, plot2, plot3, plot4, plot5)
dev.off()
```

```
pdf
  2
```

```
png("plots/oneplotwithlegend.tiff", height = 7, width = 7, units = "in", res = 300)
lines_graph_maker(sum,  p.1,  "Precipitation (mm/s)", legend="right")
```

```
`summarise()` has grouped output by 'year'. You can override using the
`.groups` argument.
```

```
dev.off()
```

```
pdf
  2
```

```
png("plots/phase_trend_line_graph.tiff", height = 5, width = 8, units = "in", res = 300)
lines_graph_maker(sum,  phaseCorrectRainRatio,  "Daily Additive Rainfall ratio", legend="rig
```

```
`summarise()` has grouped output by 'year'. You can override using the
`.groups` argument.
```

```
dev.off()
```

```
pdf
  2
```

```
#
# ####seasonal mathematical measure
#
# seasonal <- function(x, y) {
#   filtered_data <- clone_copy  %>%
#     filter(season == x &  period==y) %>%
```

```
#     group_by(year, season, period) %>%
#     summarize(mean = mean(phaseCorrectRainRatio)) %>%
#     ungroup() %>%
#     mutate(mean_ema = movavg(mean, n=5, type='e'))
#   return(list(filtered_data, mk.test(filtered_data$mean_ema), sens.slope(filtered_data$mea
#               pettitt.test(filtered_data$mean_ema), cpt.meanvar(filtered_data$mean_ema, me
# }
#
# seasonal("monsoon", "Day")
# seasonal("monsoon", "Night")
# seasonal('pre_monsoon', "Day")
# seasonal('pre_monsoon', "Night")
# seasonal('post_monsoon', "Day")
# seasonal('post_monsoon', "Night")
# seasonal('winter', "Day")
# seasonal('winter', "Night")
#
# ######rain to snow precipitation ration
# volume <- data %>%
#   group_by(year) %>%
#   summarize(rain = mean(phaseCorrectRain),
#             snow = mean(phaseCorrectSnow)) %>%
#   mutate(ratio=rain/snow) %>%
#   summarize(mean = mean(ratio),
#             lower_ci = quantile(ratio, probs = 0.025),
#             upper_ci = quantile(ratio, probs = 0.975))
```