```
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
#                                                                              #
#                           Shortlist stations from 1960-2020                  #
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
df = read.csv("D:/Anisha/UCD/Sem 3/Thesis/stations/StationDetails.csv" ,
              header = T)
colnames(df)
df = df[!(df$Open.Year>1960),] # remove stations with open year after 1960
df[df==""] <- NA #Replace empty values with NA
df = df[!(rowSums(is.na(df))==0),] # Remove non NA terms from close year
df <- subset(df,select=-c(Close.Year,Open.Year))

write.csv(df,"D:/Anisha/UCD/Sem 3/Thesis/stations/stations.csv",
          row.names = FALSE)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
#                                                                              #
#                           Plot stations on map                               #
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
library(tibble)
library(sf)
library(mapview)
# Plot 28 selected stations
data = read.csv("D:/Anisha/UCD/Sem 3/Thesis/stations/for plot.csv")
data = as_tibble(data)
locations <- st_as_sf(data, coords = c("Longitude", "Latitude"), crs = 4326)
mapview(locations)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
#                                                                              #
#                          Combining separate files to one file                #
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
# Identify file names
data_files <- list.files("D:/Anisha/UCD/Sem 3/Thesis/data sets/excel")
data_files  # Print file names
library(plyr)
library(readr)
setwd("D:/Anisha/UCD/Sem 3/Thesis/data sets/excel")
dat_csv = ldply(data_files, read_csv) #Compile data for all the stations
dat_csv = subset(dat_csv[,1:4])
# dat_csv$rain[1] represents amt rain for station 1 in Jan 1960
# dat_csv$rain[2] represents amt rain for station 1 in feb 1960
# dat_csv$rain[732] represents amt rain for station 1 in Dec 2020
# dat_csv$rain[733] represents amt rain for station 2 in Jan 1960
# dat_csv$rain[1464] represents amt rain for station 2 in Dec 2020
# dat_csv$rain[19765] represents amt rain for station 28 in Jan 1960
# dat_csv$rain[20496] represents amt rain for station 28 in Dec 2020
write.csv(dat_csv,"D:/Anisha/UCD/Sem 3/Thesis/stations/Final data.csv",
          row.names = FALSE)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
#                                                                              #
#                           Summary statistics and EDA                         #
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
data = read.csv("D:/Anisha/UCD/Sem 3/Thesis/Final data.csv")
# Convert distance to m from km
```

```r
data$distance<- data$distance*1000
# Summary statistics for rainfall
summary(data$rainmm)
mx = which.max(data$rainmm)
data[mx,]
mn = which.min(data$rainmm)
data[mn,]

# Correlation Plot
library("PerformanceAnalytics")
chart.Correlation(scale(data[c(4:7,10)]), histogram=TRUE, pch=19,
                  method = "spearman")

# avg rain plot by year
avg_year = c()
for (i in 1:61) {
  avg_year[i] = mean(data$rainmm[seq(336*i-335,336*i,1)])

}
avg_year
#avg_year[1] = mean(rain(all station in year 1960))
#avg_year[61] = mean(rain(all station in year 2020))
year = seq(1960,2020,1)
avg_rain = data.frame(year,avg_year)
fit <- lm(avg_year~year, data=avg_rain)
plot(avg_rain ,type = "l", xlab = "Year",ylab = "Average Rainfall", lwd = 2)
lines(avg_rain$year, fitted(fit), col="cyan4", lwd = 2)
summary(fit)
# rain = -347.49 + 0.22*year
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
#                                                                               #
#                             Linear Model                                      #
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
library(lmtest)
data$month = as.factor(data$month)

# with interactions forward regression
fit = lm(rainmm ~ heightm + easting + northing + distance +
            month + year + heightm:easting + heightm:distance +
            easting:northing + easting:distance + northing:distance +
            easting:year, data = data)
summary(fit)

#### breusch-pagan test
#null : Homoscedasticity is present (equal variance)
#alternate: Heteroscedasticity is present (not equal distribution of variances)
# pvalue < 0.05 => reject null hypo
bptest(fit)
# pvalue very small => heteroscedastity is present
# => try weighted least square regression model

# Weighted Least square regression
wt <- 1 / lm(abs(fit$residuals) ~ fit$fitted.values)$fitted.values^2
```

```r
wls_model <- lm(rainmm ~ heightm + easting + northing + distance +
                   month + year + heightm:easting + heightm:distance +
                   easting:northing + easting:distance +
                   northing:distance + easting:year, data = data,
                 weights=wt)

summary(wls_model)
par(mfrow=c(2,2))
plot(wls_model)
# Log-likelihood ratio
lrtest(wls_model)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
#                                                                              #
#                           TSA                                                #
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
library(TSA)
library(tseries)
library(forecast)
library(fastDummies) # dummy variables
library(magrittr)

# Dummy Variables
data = dummy_cols(data,select_columns = "month")
# drop month main column
data$month <- NULL
# move rain to last column
library(dplyr)
data <- data %>% relocate(rainmm, .after = last_col())
# replace month_1 all values to 1 since the 1-hot encoding was rank decificient
data$month_1 = rep(1,20496)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
adf.test(data$rainmm)
# p-value < 0.05 => Stationary ie no unit root ie do not difference
tsdisplay(data$rainmm, col = "cyan3")

mdl <- auto.arima(data$rainmm, xreg = as.matrix(data[c(4,5,7:8,10:20)]),
                  method = "ML")
summary(mdl)
# Residual analysis
res = residuals(mdl)
par(mfrow=c(1,2))
acf(res)
pacf(res)
## Check Residuals for model
checkresiduals(mdl)
# Ljung box- Ho: model fits Ha: model does not fit well
# p-value < 0.05 => Reject Ho ie I want pvalue > 0.05
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
#                                                                              #
#                          Spatial Modeling                                    #
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
# Data Transformation
data = read.csv("D:/Anisha/UCD/Sem 3/Thesis/spatial data.csv")
```

```
data$X <- rep(seq(1,12,1),1708)
df1 = data[!(data$year>1990),] #1960 - 1990
df2 = data[!(data$year<1991),] #1991 - 2020

### Month Wise Averages for 1960-1990 for each station
n1 = c()
for (i in 1:12) {
  n1[i]= mean(df1$rain[seq(i,372,12)])
}

n2 = c()
for (i in 1:12) {
  n2[i]= mean(df1$rain[seq(372+i,372*2,12)])
}

n3 = c()
for (i in 1:12) {
  n3[i]= mean(df1$rain[seq(744+i,372*3,12)])
}

n4 = c()
for (i in 1:12) {
  n4[i]= mean(df1$rain[seq(1116+i,372*4,12)])
}

n5 = c()
for (i in 1:12) {
  n5[i]= mean(df1$rain[seq(1488+i,372*5,12)])
}
n6 = c()
for (i in 1:12) {
  n6[i]= mean(df1$rain[seq(1860+i,372*6,12)])
}
n7 = c()
for (i in 1:12) {
  n7[i]= mean(df1$rain[seq(2232+i,372*7,12)])
}
n8 = c()
for (i in 1:12) {
  n8[i]= mean(df1$rain[seq(2604+i,372*8,12)])
}
n9 = c()
for (i in 1:12) {
  n9[i]= mean(df1$rain[seq(2976+i,372*9,12)])
}
n10 = c()
for (i in 1:12) {
  n10[i]= mean(df1$rain[seq(3348+i,372*10,12)])
}
n11 = c()
for (i in 1:12) {
  n11[i]= mean(df1$rain[seq(3720+i,372*11,12)])
}
```

```
n12 = c()
for (i in 1:12) {
  n12[i]= mean(df1$rain[seq(4092+i,372*12,12)])
}
n13 = c()
for (i in 1:12) {
  n13[i]= mean(df1$rain[seq(4464+i,372*13,12)])
}
n14 = c()
for (i in 1:12) {
  n14[i]= mean(df1$rain[seq(4836+i,372*14,12)])
}
n15 = c()
for (i in 1:12) {
  n15[i]= mean(df1$rain[seq(5208+i,372*15,12)])
}
n16 = c()
for (i in 1:12) {
  n16[i]= mean(df1$rain[seq(5580+i,372*16,12)])
}
n17 = c()
for (i in 1:12) {
  n17[i]= mean(df1$rain[seq(5952+i,372*17,12)])
}
n18 = c()
for (i in 1:12) {
  n18[i]= mean(df1$rain[seq(6324+i,372*18,12)])
}
n19 = c()
for (i in 1:12) {
  n19[i]= mean(df1$rain[seq(6696+i,372*19,12)])
}
n20 = c()
for (i in 1:12) {
  n20[i]= mean(df1$rain[seq(7068+i,372*20,12)])
}
n21 = c()
for (i in 1:12) {
  n21[i]= mean(df1$rain[seq(7440+i,372*21,12)])
}
n22 = c()
for (i in 1:12) {
  n22[i]= mean(df1$rain[seq(7812+i,372*22,12)])
}
n23 = c()
for (i in 1:12) {
  n23[i]= mean(df1$rain[seq(8184+i,372*23,12)])
}
n24 = c()
for (i in 1:12) {
  n24[i]= mean(df1$rain[seq(8556+i,372*24,12)])
}
n25 = c()
```

```r
for (i in 1:12) {
  n25[i]= mean(df1$rain[seq(8928+i,372*25,12)])
}
n26 = c()
for (i in 1:12) {
  n26[i]= mean(df1$rain[seq(9300+i,372*26,12)])
}
n27 = c()
for (i in 1:12) {
  n27[i]= mean(df1$rain[seq(9672+i,372*27,12)])
}
n28 = c()
for (i in 1:12) {
  n28[i]= mean(df1$rain[seq(10044+i,372*28,12)])
}
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
### Month Wise Averages for 1991-2020 for each station
m1 = c()
for (i in 1:12) {
  m1[i]= mean(df2$rain[seq(i,360,12)])
}

m2 = c()
for (i in 1:12) {
  m2[i]= mean(df2$rain[seq(360+i,360*2,12)])
}
m3 = c()
for (i in 1:12) {
  m3[i]= mean(df2$rain[seq(720+i,360*3,12)])
}
m4 = c()
for (i in 1:12) {
  m4[i]= mean(df2$rain[seq(1080+i,360*4,12)])
}
m5 = c()
for (i in 1:12) {
  m5[i]= mean(df2$rain[seq(1440+i,360*5,12)])
}
m6 = c()
for (i in 1:12) {
  m6[i]= mean(df2$rain[seq(1800+i,360*6,12)])
}
m7 = c()
for (i in 1:12) {
  m7[i]= mean(df2$rain[seq(2160+i,360*7,12)])
}
m8 = c()
for (i in 1:12) {
  m8[i]= mean(df2$rain[seq(2520+i,360*8,12)])
}
m9 = c()
for (i in 1:12) {
  m9[i]= mean(df2$rain[seq(2880+i,360*9,12)])
```

```
}
m10 = c()
for (i in 1:12) {
  m10[i]= mean(df2$rain[seq(3240+i,360*10,12)])
}
m11 = c()
for (i in 1:12) {
  m11[i]= mean(df2$rain[seq(3600+i,360*11,12)])
}
m12 = c()
for (i in 1:12) {
  m12[i]= mean(df2$rain[seq(3960+i,360*12,12)])
}
m13 = c()
for (i in 1:12) {
  m13[i]= mean(df2$rain[seq(4320+i,360*13,12)])
}
m14 = c()
for (i in 1:12) {
  m14[i]= mean(df2$rain[seq(4680+i,360*14,12)])
}
m15 = c()
for (i in 1:12) {
  m15[i]= mean(df2$rain[seq(5040+i,360*15,12)])
}
m16 = c()
for (i in 1:12) {
  m16[i]= mean(df2$rain[seq(5400+i,360*16,12)])
}
m17 = c()
for (i in 1:12) {
  m17[i]= mean(df2$rain[seq(5760+i,360*17,12)])
}
m18 = c()
for (i in 1:12) {
  m18[i]= mean(df2$rain[seq(6120+i,360*18,12)])
}
m19 = c()
for (i in 1:12) {
  m19[i]= mean(df2$rain[seq(6480+i,360*19,12)])
}
m20 = c()
for (i in 1:12) {
  m20[i]= mean(df2$rain[seq(6840+i,360*20,12)])
}
m21 = c()
for (i in 1:12) {
  m21[i]= mean(df2$rain[seq(7200+i,360*21,12)])
}
m22 = c()
for (i in 1:12) {
  m22[i]= mean(df2$rain[seq(7560+i,360*22,12)])
}
```

```
m23 = c()
for (i in 1:12) {
  m23[i]= mean(df2$rain[seq(7920+i,360*23,12)])
}
m24 = c()
for (i in 1:12) {
  m24[i]= mean(df2$rain[seq(8280+i,360*24,12)])
}
m25 = c()
for (i in 1:12) {
  m25[i]= mean(df2$rain[seq(8640+i,360*25,12)])
}
m26 = c()
for (i in 1:12) {
  m26[i]= mean(df2$rain[seq(9000+i,360*26,12)])
}
m27 = c()
for (i in 1:12) {
  m27[i]= mean(df2$rain[seq(9360+i,360*27,12)])
}
m28 = c()
for (i in 1:12) {
  m28[i]= mean(df2$rain[seq(9720+i,360*28,12)])
}
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
new_df = read.csv("D:/Anisha/UCD/Sem 3/Thesis/new_geo.csv")
new_df$month <- rep(seq(1,12,1),28)
new_df$Distance.from.the.sea <- new_df$Distance.from.the.sea*1000
# dummy variables
library(fastDummies)
library(magrittr)
new_df = dummy_cols(new_df,select_columns = "month")
# drop month main column
new_df$month <- NULL
# replace month_1 all values to 1 since the 1-hot encoding was rank decificient
new_df$month_1 = rep(1,336)
# create new columns
avg1 = c(n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,n14,n15,n16,n17,n18,n19,n20,
         n21,n22,n23,n24,n25,n26,n27,n28)
avg2 = c(m1,m2,m3,m4,m5,m6,m7,m8,m9,m10,m11,m12,m13,m14,m15,m16,m17,m18,m19,m20,
         m21,m22,m23,m24,m25,m26,m27,m28)
# add new columns to the data
new_df$avg1 <- avg1 # monthly avg from 1960 - 1990
new_df$avg2 <- avg2 # monthly avg from 1991 - 2020
write.csv(new_df,"D:/Anisha/UCD/Sem 3/Thesis/final_geo.csv")
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#

library(geoR)
#### 28*12 = 336 obs
data = read.table("D:/Anisha/UCD/Sem 3/Thesis/final_geo.csv",header=TRUE,
                  sep=",")

#data$month <- as.factor(rep(seq(1,12,1),28))
```

```r
### Months matrix causes error as they cant be inverted
data = subset(data[c(2:8,21,22)])
colnames(data) = c("Station ID","Station Name", "County","Height","Easting",
                   "Northing","Distance","Avg_1960_1990","Avg_1991_2020")

############## 1960 - 1990
# make data spatial
geo1 = as.geodata(data, coords.col = 5:6,covar.col = c(4,7), data.col = 8)
names(geo1)
###### Matern

## Kappa = 0.5

mtrn1 = likfit(geo1, cov.model = "mat", kappa = 0.5,ini.cov.pars = c(600,20000),
               nugget = 650, trend = ~Height+Distance)
mtrn1
summary(mtrn1)
## Kappa = 1.5
mtrn2 = likfit(geo1, cov.model = "mat", kappa = 1.5,ini.cov.pars = c(600,20000),
               nugget = 650, trend = ~Height+Distance)
mtrn2
summary(mtrn2)
## Kappa = 2.5
mtrn3 = likfit(geo1, cov.model = "mat", kappa = 2.5,ini.cov.pars = c(600,20000),
               nugget = 650, trend = ~Height+Distance)
smtrn3
summary(mtrn3)
plot(variog(geo1), main = "Variogram for 1960-1990")
lines(mtrn3, lwd = 2, col = "cyan4")
###### Exponential
exp1 = likfit(geo1, cov.model = "exponential",ini.cov.pars = c(600,20000),
              nugget = 650, trend = ~Height+Distance)
exp1
summary(exp1)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
############## 1991 - 2020
# make data spatial
geo2 = as.geodata(data, coords.col = 5:6,covar.col = c(4,7), data.col = 9)
names(geo2)
###### Matern
### Months matrix causes error as they cant be inverted

## Kappa = 0.5
mtrn4 = likfit(geo2, cov.model = "mat", kappa = 0.5,ini.cov.pars = c(600,20000),
               nugget = 750, trend = ~Height+Distance)
mtrn4
summary(mtrn4)
## Kappa = 1.5
mtrn5 = likfit(geo2, cov.model = "mat", kappa = 1.5,ini.cov.pars = c(600,20000),
               nugget = 750, trend = ~Height+Distance)
mtrn5
summary(mtrn5)
## Kappa = 2.5
```

```
mtrn6 = likfit(geo2, cov.model = "mat", kappa = 2.5,ini.cov.pars = c(600,20000),
               nugget = 750, trend = ~Height+Distance)
mtrn6
summary(mtrn6)
plot(variog(geo2),main = "Variogram for 1991-2020")
lines(mtrn6, lwd = 2, col = "cyan4")
###### Exponential
exp2 = likfit(geo2, cov.model = "exponential",ini.cov.pars = c(600,20000),
              nugget = 750, trend = ~Height+Distance)
exp2
summary(exp2)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~#
## Calculate the change in amount of rainfall.
chg = (mean(data$Avg_1991_2020)-mean(data$Avg_1960_1990))/mean(data$Avg_1960_1990)
chg = chg*100
chg
```