```
#!/usr/bin/env Rscript
#Chapter 8.8 Final Practical

rm(list = ls())
library(stats) # for plot.ts
library(ggplot2)
library(pracma) # for movavg

# Collect the data
MyData = as.data.frame(read.csv("../Data/KeyWest.CSV"))


#load("../Data/KeyWestAnnualMeanTemperature.RData")
#Years = ats[[1]]
Temps = MyData[[3]]
#MyData = as.data.frame(cbind(Years,Temps))


#Examine = function(Data){
  # plot simple time seriesgraphics.off()

pdf("../Results/TAutocorrtimeseries1.pdf")

ggplot(MyData, aes(x = Year, y = Temp))+
 labs(title = "Time series of temperature data for KeyWest 1900 - 2000", x = "date", y =
"Temperature")+
  geom_line()

dev.off()

pdf("../Results/TAutocorrtimeseries2.pdf")
par(mfrow = c(2,2))
sapply(1:4, function(x) plot(MyData$Temp[-c(100:(100-x+1))], MyData$Temp[-c(1:x)], ylab =
"Year", xlab = "year"))
dev.off()

#autocorrelation coef is Sum(Y[i+1]-AveY)(Y[i] -AveY)/sum(sqr(Y[i]-AveY))


# calculate sum[(Yi+1 - ave)(Yi - ave)] -- numerator of autocorr coef
#Calc_numerator = function(avector){
num = vector("numeric",99)
for (i in seq_along((Temps))) {
  if (i <100) {
  num[i] = as.vector( Temps[i+1] - mean(Temps) ) * ( Temps[i] - mean(Temps) )
  }
 else
 totalnum = sum(num)}



#Calc_denom = function(avector){
```

```
#  calcuate sum[(Yi - ave)^2] -- denom of auto corr coef
denom = vector("numeric",99)
for (i in seq_along((Temps))) {
  if (i <100) {
    denom[i] = as.vector(( Temps[i] - mean(Temps) )^2)
  }
  else
    totaldenom = sum(denom)}



#Calc_acf = function(avector){
   autocorrcoef = totalnum/totaldenom
   print("autocorrelatoin coefficient for lag 1 is ")
   print(autocorrcoef)


#generate 1000 acf from random sampling of Temps
  acfs = vector("numeric",1000)

  for (j in 1:1000){
    num = vector("numeric",99)
    denom = vector("numeric",99)

      for (i in seq_along((Temps))) {
        RTemp = sample(Temps,100)

        if (i <100) {
          num[i] = as.vector( RTemp[i+1] - mean(RTemp) ) * (RTemp[i] - mean(RTemp) )
          denom[i] = as.vector(( RTemp[i] - mean(RTemp) )^2)
      }
        else
          totalnum = sum(num)
          totaldenom = sum(denom)

      }
        acfs[j] = totalnum/totaldenom
  }


#Calculate p value
  p = length(acfs[acfs > autocorrcoef])/1000
  print("p value for autocorrelation coeffice=ient of lag 1 is ")
  print(p)
  message = "p value for autocorrelation coeffice=ient of lag 1 is "
  output = c(message,p)
  write(output,'../Results/pvalue.txt' )

 pdf("../Results/TAutocorrHist.pdf")
hist(acfs, main = paste("distribution of random acfs. acf for sequential data 0.309"))
  #Since p value indicates correlation between points, lets look at moving average
  #and plot a trend line
```

```
dev.off()

  ma = movavg(MyData$Temp, 2, "s") # simple moving average with 2 points

  MyData = as.data.frame(cbind(MyData,ma)) # need a dataframe for ggplot

  lm = summary(lm(MyData$ma ~ MyData$Year, MyData)) #a linear model of moving averages
  pdf("../Results/TAutocorrmovingavg.pdf")
  ggplot(MyData, aes(y = MyData$ma, x = MyData$Year , colour  = abs(lm$residuals)))+
  geom_point()+
  labs(x = "Years", title = "Scatterplot of 2 point moving average and linear fit", y = "Moving
Average")+
  geom_abline(intercept = lm$coefficients[1][1],
              slope = lm$coefficients[2][1],
               colour = "red")

  dev.off()
```