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


prepare_workspace = function(){
  rm(list = ls())
  library(stats) # for plot.ts
  library(ggplot2)
  library(pracma) # for movavg
  graphics.off()
}

# get the data and make two vectors coz easier for loops

GetData = function(){
  load("../Data/KeyWestAnnualMeanTemperature.RData")
  Years = ats[[1]]
  Temps = ats[[2]]
  MyData = as.data.frame(cbind(Years,Temps))
  return(MyData)
}

Examine = function(Data){
  # plot simple time series
  plot.ts(Data)
  #plot terms with lag of 1 to 4 years against each other M Crawley p787
  par(mfrow = c(2,2))
  sapply(1:4, function(x) plot(Data[-c(100:(100-x+1))], Data[-c(1:x)]))
}

#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((avector))) {
  if (i <100) {
  num[i] = as.vector( avector[i+1] - mean(avector) ) * ( avector[i] -
mean(avector) )
  }
 else
 totalnum = sum(num)
}
return(totalnum)}


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


Calc_acf = function(avector){
  num = Calc_numerator(avector)
  denom = Calc_denom(avector)
  autocorrcoef = num/denom
```

```
}

#generate 1000 acf from random sampling of Temps
Calc_Random_acfs = function(avector){
  acfs = vector("numeric",1000)
  for (i in 1:1000){
    acfs[i] = Calc_acf(sample(avector,100))
  }
  return(acfs)
  }

Calc_p_Value = function(avector,avalue){
  p = length(avector[avector > avalue])
 }

#Since p value indicates correlation between points, lets look at moving average
#and plot a trend line

Final_plots = function(DataIn){
  ma = movavg(DataIn$Temps, 2, "s") # simple moving average with 2 points

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

  lm = summary(lm(MyData$ma ~ MyData$Years, MyData)) #a linear model of moving
averages

  ggplot(MyData, aes(y = MyData$ma, x = MyData$Years , colour  =
abs(lm$residuals)))+
  geom_point()+
  geom_abline(intercept = lm$coefficients[1][1],
              slope = lm$coefficients[2][1],
               colour = "red")
}

######################

prepare_workspace()
TempData = GetData()
Examine(TempData$Temps)
autocorr =Calc_acf(TempData$Temps)
print ("autocorrelation coefficient is ")
print(autocorr)
randacf = Calc_Random_acfs(TempData$Temps)
p = Calc_p_Value(randacf,autocorr)
print("p value is ")
print(p)
Final_plots(TempData)
```