HW5\_567

Matthew Stoebe

2024-11-20

library(ggplot2)

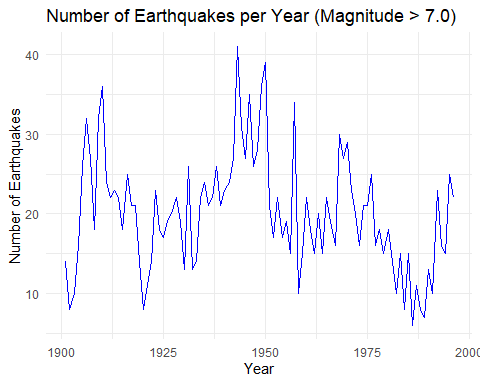
#Question 1

earthquakes <- read.csv("Data/earthquakes.csv")  
head(earthquakes)

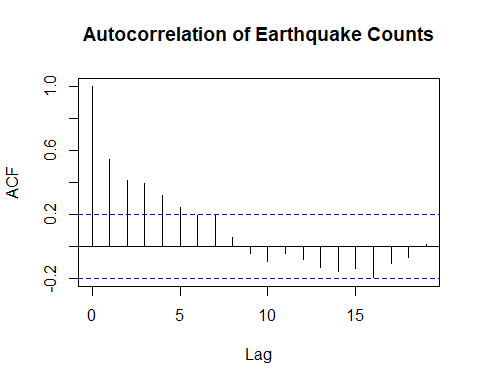
## year quakes  
## 1 1901 14  
## 2 1902 8  
## 3 1903 10  
## 4 1904 16  
## 5 1905 26  
## 6 1906 32

##a Create summary plots of the data. Specifically, create a (line) plot of number of earthquakes per year. Also use acf() to create an autocorrelation plot.

ggplot(data = earthquakes, aes(x = year, y = quakes)) +  
 geom\_line(color = "blue") +  
 labs(title = "Number of Earthquakes per Year (Magnitude > 7.0)",  
 x = "Year",  
 y = "Number of Earthquakes") +  
 theme\_minimal()



acf(earthquakes$quakes, main = "Autocorrelation of Earthquake Counts")



##b Provide the estimated mean and simple t-based confidence interval for the mean. We will use this naive approach for purposes of comparison only!

n <- nrow(earthquakes)  
mean\_eq <- mean(earthquakes$quakes)  
sd\_eq <- sd(earthquakes$quakes)  
  
df <- n - 1  
  
  
se\_eq <- sd\_eq / sqrt(n)  
  
# t-value for 95% confidence interval  
alpha <- 0.05  
t\_value <- qt(1 - alpha/2, df)  
  
lower\_bound <- mean\_eq - t\_value \* se\_eq  
upper\_bound <- mean\_eq + t\_value \* se\_eq  
  
cat("Estimated Mean:", mean\_eq, "\n")

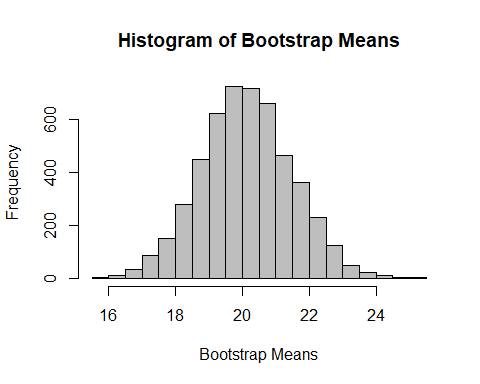
## Estimated Mean: 20.13542

cat("95% t-based Confidence Interval: [", lower\_bound, ",", upper\_bound, "]\n")

## 95% t-based Confidence Interval: [ 18.65037 , 21.62047 ]

##c Now perform a non-moving block bootstrap with a block length of 8 and R = 5000. Write a function to do this. Show the 12 blocks and the histogram of the resulting bootstrap means

set.seed(420) # For reproducibility  
  
block\_length <- 8  
num\_blocks <- floor(n / block\_length)  
  
# Create blocks  
blocks <- split(earthquakes$quakes, rep(1:num\_blocks, each = block\_length))  
  
  
non\_moving\_block\_bootstrap <- function(data, block\_length, R) {  
 n <- length(data)  
 num\_blocks <- floor(n / block\_length)  
 blocks <- split(data, rep(1:num\_blocks, each = block\_length))  
   
 bootstrap\_means <- numeric(R)  
   
 for (i in 1:R) {  
 # Sample blocks with replacement  
 sampled\_blocks <- sample(blocks, size = num\_blocks, replace = TRUE)  
 # Concatenate sampled blocks  
 sampled\_data <- unlist(sampled\_blocks)  
 # Calculate mean  
 bootstrap\_means[i] <- mean(sampled\_data)  
 }  
 return(bootstrap\_means)  
}  
  
# Perform bootstrap  
R <- 5000  
bootstrap\_means <- non\_moving\_block\_bootstrap(earthquakes$quakes, block\_length, R)  
hist(bootstrap\_means, breaks = 30, main = "Histogram of Bootstrap Means",  
 xlab = "Bootstrap Means", col = "grey", border = "black")

 ##d Using your boostrap results, construct a 95% confidence interval using the percentile method.

# Calculate the 2.5th and 97.5th percentiles  
ci\_lower <- quantile(bootstrap\_means, 0.025)  
ci\_upper <- quantile(bootstrap\_means, 0.975)  
  
# Output the bootstrap confidence interval  
cat("95% Bootstrap Confidence Interval (Percentile Method): [", ci\_lower, ",", ci\_upper, "]\n")

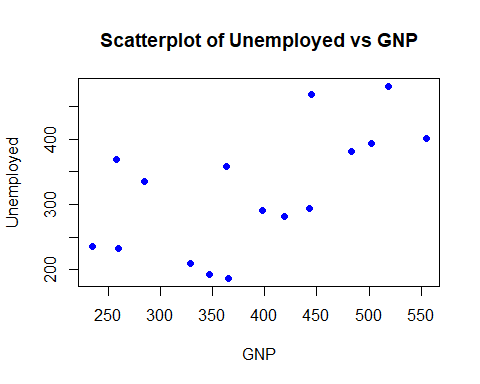
## 95% Bootstrap Confidence Interval (Percentile Method): [ 17.46875 , 22.80208 ]

##e Compare your bootstrap confidence interval to the simple t-based confidence interval.Which is wider? Briefly explain why this is the case. Our bootstrap Confidence interval is wider because it accounts for the autocorrelation in the data.The t based method assumes independent samples which is often not the case with time series data.

#Question 2

##a Create a scatterplot of Unemployed (y) vs GNP (x).

data(longley)  
plot(longley$GNP, longley$Unemployed,  
 main = "Scatterplot of Unemployed vs GNP",  
 xlab = "GNP",  
 ylab = "Unemployed",  
 pch = 19, col = "blue")



##b Use cor.test() to calculate (default) Pearson correlation, p-value and 95% confidence interval. You can just print the output

pearson\_result <- cor.test(longley$Unemployed, longley$GNP)  
print(pearson\_result)

##   
## Pearson's product-moment correlation  
##   
## data: longley$Unemployed and longley$GNP  
## t = 2.8376, df = 14, p-value = 0.01317  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.1549766 0.8464304  
## sample estimates:  
## cor   
## 0.6042609

##c Use cor.test() again but this time with method = ”spearman” and print the result. Note that a confidence interval is NOT provided.

spearman\_result <- cor.test(longley$Unemployed, longley$GNP, method = "spearman")  
print(spearman\_result)

##   
## Spearman's rank correlation rho  
##   
## data: longley$Unemployed and longley$GNP  
## S = 246, p-value = 0.009375  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.6382353

##d Use bootstrap() to perform case resampling bootrap for correlation with R = 5000.Provide a 95% confidence interval using the percentile method.

# I was having issues with the bootstrap() function so i swapped to this one.   
library(boot)

## Warning: package 'boot' was built under R version 4.4.2

data\_longley <- longley[, c("Unemployed", "GNP")]  
correlation\_stat <- function(data, indices) {  
 resampled\_data <- data[indices, ]  
 return(cor(resampled\_data$Unemployed, resampled\_data$GNP))  
}  
  
# Perform bootstrapping  
R <- 5000  
boot\_results <- boot(data\_longley, statistic = correlation\_stat, R = R)  
bootstrap\_estimates <- boot\_results$t  
  
  
# Calculate the 95% confidence interval using the percentile method  
ci\_lower <- quantile(bootstrap\_estimates, 0.025)  
ci\_upper <- quantile(bootstrap\_estimates, 0.975)  
  
  
cat("95% Bootstrap Confidence Interval: [", ci\_lower, ",", ci\_upper, "]\n")

## 95% Bootstrap Confidence Interval: [ 0.2564638 , 0.8303361 ]

##e Finally, implement a permutation test of H0 : ρ = 0 where ρ is the population correlation. Perform 5000 permutations and report your permutation based p-value (twosided).

obs\_cor <- cor(longley$Unemployed, longley$GNP, method='spearman')  
  
n\_perm <- 5000  
perm\_cor <- numeric(n\_perm)  
  
for (i in 1:n\_perm) {  
 permuted\_gnp <- sample(longley$GNP)  
 perm\_cor[i] <- cor(longley$Unemployed, permuted\_gnp, method='spearman')  
}  
  
# two-sided p-value  
p\_value <- (sum(abs(perm\_cor) >= abs(obs\_cor)) + 1) / (n\_perm + 1)  
cat("Permutation p-value:", p\_value)

## Permutation p-value: 0.00859828