homework1

xbw

2025-09-17

* a

iowa.df<-read.csv("D:/postgraduate/data science/mynotes/data/iowa.csv", sep = ';', header=T)

b.33 rows and 10 columns  
  
c.Year Rain0 Temp1 Rain1 Temp2 Rain2 Temp3 Rain3 Temp4 Yield  
  
d. The value of row 5, column 7 of iowa.df is 79.7  
  
e.   
  
  
``` r  
iowa.df[2,]  
```  
  
```  
## Year Rain0 Temp1 Rain1 Temp2 Rain2 Temp3 Rain3 Temp4 Yield  
## 2 1931 14.76 57.5 3.83 75 2.72 77.2 3.3 72.6 32.9  
```

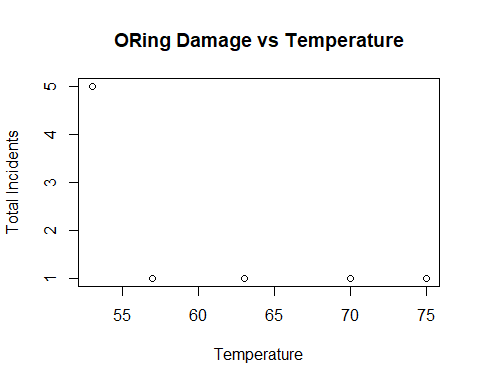
* 1. 因为vector1的元素被定义为character类型，所以vector1的元素在比较和排序时根据字符的ASCII码按照字典序进行，“5”,“12”,“7”,“32”,按照首个字符的ASCII码进行排序，“1”<“3”<“5”<“7”,所以max(vector1)返回“7”,sort(vector1)返回”12” “32” “5” “7” sum()的参数不能为character,所以会报错
  + 当创建向量vector2 <- c(“5”,7,12)时,由于包含了不同类型的元素，所以R会自动进行强制类型转换，以确保所有元素类型相同，因为存在字符串”5”,所以7和12都被转换为character,当执行vector2[2] + vector2[3]时，试图对两个字符元素（即”7”和”12”）使用加法操作符+，在R中+不能用于character，因此会抛出异常。
  + dataframe3 <- data.frame(z1=“5”,z2=7,z3=12)这句代码创建了一个有三列数据的dataframe，列名分别为z1,z2,z3,元素为”5”,7,12 dataframe3[1,2] + dataframe3[1,3] 将dataframe3第1行2列和3列的元素相加，返回19
  + list4[[2]]+list4[[4]]返回结果为168，而list4[2]+list4[4]会报错，因为list4[[2]]返回的是列表list4中的第二个元素，可以直接运算；而list4[2]返回的是包含第二个元素的子列表，列表的数据类型不能进行运算。

1. #design an expression that will give you the sequence of numbers from 1 to 10000 in increments of 372.  
   seq(1,10000,by=372)

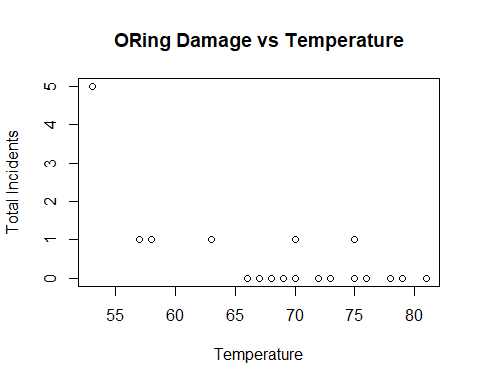
* ## [1] 1 373 745 1117 1489 1861 2233 2605 2977 3349 3721 4093 4465 4837 5209  
  ## [16] 5581 5953 6325 6697 7069 7441 7813 8185 8557 8929 9301 9673
* #Design another that will give you a sequence between 1 and 10000 that is exactly 50 numbers in length.  
  seq(1,10000,by=(10000/50))
* ## [1] 1 201 401 601 801 1001 1201 1401 1601 1801 2001 2201 2401 2601 2801  
  ## [16] 3001 3201 3401 3601 3801 4001 4201 4401 4601 4801 5001 5201 5401 5601 5801  
  ## [31] 6001 6201 6401 6601 6801 7001 7201 7401 7601 7801 8001 8201 8401 8601 8801  
  ## [46] 9001 9201 9401 9601 9801
  1. rep(1:3, times=3) 是将整个向量重复3次，输出为123123123；而rep(1:3, each=3)是将向量中每个元素重复三次，输出为111222333

MB.Ch1.2.

library(DAAG)  
#Create a new data frame by extracting these rows from orings  
orings\_subset <- orings[c(1,2,4,11,13,18), ]   
#plot total incidents against temperature for this new data frame  
plot(orings\_subset$Temperature,orings\_subset$Total,xlab = "Temperature",ylab="Total Incidents", main = "ORing Damage vs Temperature")



#plot for the full dataset  
plot(orings$Temperature,orings$Total,xlab = "Temperature",ylab="Total Incidents", main = "ORing Damage vs Temperature")



MB.Ch1.4. (a).

str(ais)

## 'data.frame': 202 obs. of 13 variables:  
## $ rcc : num 3.96 4.41 4.14 4.11 4.45 4.1 4.31 4.42 4.3 4.51 ...  
## $ wcc : num 7.5 8.3 5 5.3 6.8 4.4 5.3 5.7 8.9 4.4 ...  
## $ hc : num 37.5 38.2 36.4 37.3 41.5 37.4 39.6 39.9 41.1 41.6 ...  
## $ hg : num 12.3 12.7 11.6 12.6 14 12.5 12.8 13.2 13.5 12.7 ...  
## $ ferr : num 60 68 21 69 29 42 73 44 41 44 ...  
## $ bmi : num 20.6 20.7 21.9 21.9 19 ...  
## $ ssf : num 109.1 102.8 104.6 126.4 80.3 ...  
## $ pcBfat: num 19.8 21.3 19.9 23.7 17.6 ...  
## $ lbm : num 63.3 58.5 55.4 57.2 53.2 ...  
## $ ht : num 196 190 178 185 185 ...  
## $ wt : num 78.9 74.4 69.1 74.9 64.6 63.7 75.2 62.3 66.5 62.9 ...  
## $ sex : Factor w/ 2 levels "f","m": 1 1 1 1 1 1 1 1 1 1 ...  
## $ sport : Factor w/ 10 levels "B\_Ball","Field",..: 1 1 1 1 1 1 1 1 1 1 ...

colSums(is.na(ais))#统计ais中每一列缺失值的数量，结果都为0，所以全都没有缺失值

## rcc wcc hc hg ferr bmi ssf pcBfat lbm ht wt   
## 0 0 0 0 0 0 0 0 0 0 0   
## sex sport   
## 0 0

(b).

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

#Make a table that shows the numbers of males and females for each different sport  
sport\_sex\_table <- table(ais$sport, ais$sex)  
sport\_sex\_df <- as.data.frame(sport\_sex\_table) #将table转换为data frame  
colnames(sport\_sex\_df) <- c("Sport", "Sex", "Count") #修改列名  
grouped\_data <- group\_by(sport\_sex\_df,Sport) #将数据根据运动分组  
#统计  
summarized\_data <- summarise(  
 grouped\_data,  
 Male = sum(Count[Sex == "m"]),  
 Female = sum(Count[Sex == "f"]),  
 Total = sum(Count),  
 Ratio = round(Male / Female, 2)  
)  
#筛选出男女比例大于2的运动  
imbalanced\_sports <- filter(summarized\_data,Ratio > 2 | Ratio < 0.5 | is.infinite(Ratio))  
print(imbalanced\_sports)

## # A tibble: 4 × 5  
## Sport Male Female Total Ratio  
## <fct> <int> <int> <int> <dbl>  
## 1 Gym 0 4 4 0   
## 2 Netball 0 23 23 0   
## 3 T\_Sprnt 11 4 15 2.75  
## 4 W\_Polo 17 0 17 Inf

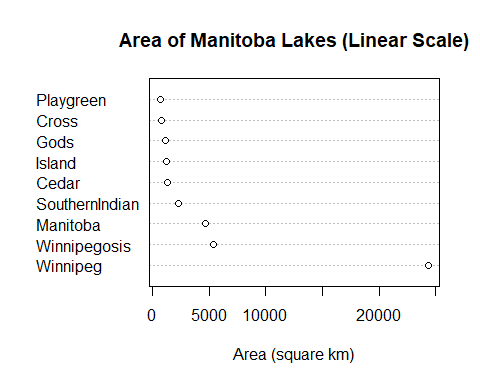
#或者使用管道运算符  
imbalanced\_sports <- sport\_sex\_df %>% group\_by(Sport) %>% summarise(   
 Male = sum(Count[Sex == "m"]),  
 Female = sum(Count[Sex == "f"]),  
 Total = sum(Count),  
 Ratio = round(Male / Female, 2)) %>% filter(Ratio > 2 | Ratio < 0.5 | is.infinite(Ratio))

MB.Ch1.6.

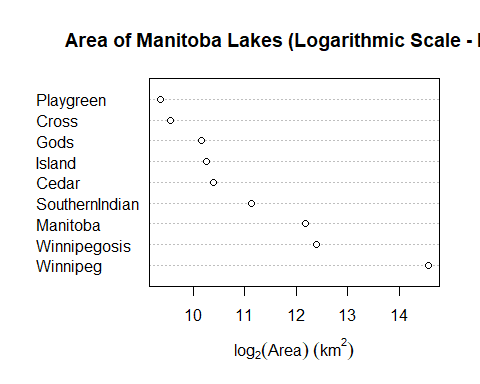
col1 <- c("Winnipeg","Winnipegosis","Manitoba","SouthernIndian","Cedar","Island","Gods","Cross","Playgreen")  
 col2 <- c(217,254,248,254,253,227,178,207,217)  
 col3 <- c(24387,5374,4624,2247,1353,1223,1151,755,657)  
 Manitoba.lakes <- data.frame(col2, col3) #生成data frame  
 Manitoba.lakes <- data.frame("elevation" = col2,"area" = col3) #设置列名  
 row.names(Manitoba.lakes) <- col1 #设置行名

MB.Ch1.7. (a)

dotchart(Manitoba.lakes$area, labels = row.names(Manitoba.lakes), main = "Area of Manitoba Lakes (Linear Scale)",xlab = "Area (square km)")



dotchart(log2(Manitoba.lakes$area), labels = row.names(Manitoba.lakes),main = "Area of Manitoba Lakes (Logarithmic Scale - log2)", xlab = expression(log[2](Area) ~ (km^2)))



MB.Ch1.8.

#计算了所有列出的湖泊面积之和，作为Manitoba被水覆盖面积的下界  
total\_water\_area <- sum(Manitoba.lakes$area)