In Class Work Week 10

Peyton Hall

03/14/2024

```
rm(list=ls())
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(Lahman)
```

```
# this is in the wider format
data <- data.frame(year=c(2000,2001), apple=c(50,60), bananna=c(30,35))
# data

data_long <- data %>%
  pivot_longer(cols = c(apple, bananna),names_to = "fruit",values_to = "Count")

# data_long

pivot_longer(data,cols = c(apple, bananna),names_to = "fruit",values_to = "Count")
```

```
## # A tibble: 4 x 3
##    year fruit   Count
##   <dbl> <chr>   <dbl>
## 1  2000 apple      50
## 2  2000 bananna    30
## 3  2001 apple      60
## 4  2001 bananna    35
```

```
data_wide<-data_long%>%
  pivot_wider(names_from = fruit,values_from = Count)

data_wide
```

```
## # A tibble: 2 x 3
##    year apple bananna
##   <dbl> <dbl>   <dbl>
## 1  2000    50      30
## 2  2001    60      35
```

```
BP_wide<-data.frame(subjectw=c("BHO", "GWB", "WJC"), before=c(160,120,105),after=c(115,135,145))
BP_wide
```

```
##   subjectw before after
## 1      BHO    160   115
## 2      GWB    120   135
## 3      WJC    105   145
```

```
BP_long<-BP_wide%>%
  pivot_longer(cols = c("before", "after"), names_to = "when",values_to = "SBP")
BP_long
```

```
## # A tibble: 6 x 3
##   subjectw when     SBP
##   <chr>    <chr>  <dbl>
## 1 BHO      before   160
## 2 BHO      after    115
## 3 GWB      before   120
## 4 GWB      after    135
## 5 WJC      before   105
## 6 WJC      after    145
```

```
mat <- matrix(1:9, nrow=3)
mat
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```
row_sums <- apply(mat,1, sum)
mat %>% apply(1,sum)
```

```
## [1] 12 15 18
```

```
row_sums
```

```
## [1] 12 15 18
```

```
range_diff<-function(x){
  return(max(x)-min(x))
}

mat %>%
  apply(1, range_diff)
```

```
## [1] 6 6 6
```

```
# install.packages("Lahman")
library(Lahman)

TeamsData<-Teams %>%
  select(Rank, G, W, L, R, RA)%>%
  apply(2, mean)

TeamsData
```

```
##       Rank          G          W          L          R         RA
##   4.028192 150.125373  74.674627  74.674627 681.158872 681.157877
```

```
# str(Teams)
TeamsRowmean<-Teams %>%
  select(Rank, G, W, L, R, RA)%>%
  apply(1, mean)
# TeamsRowmean
```

```
TeamsData2<-Teams %>%
  select(teamID, Rank, G, W, L, R, RA)%>%
  apply(2,mean, na.rm=TRUE)
```

```
## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
## returning NA
```

```
TeamsData2
```

```
## teamID   Rank      G      W      L      R     RA
##     NA     NA     NA     NA     NA     NA     NA
```

```
TeamsData3<-Teams %>%
  select(teamID, Rank, G, W, L, R, RA)%>%
  sapply(MARGIN=2,mean)
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
TeamsData3
```

```
##     teamID       Rank          G          W          L          R         RA
##         NA   4.028192 150.125373  74.674627  74.674627 681.158872 681.157877
```

```
TeamsData4<-Teams %>%
  select(teamID, Rank, G, W, L, R, RA)%>%
  lapply(MARGIN=2,mean)
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
TeamsData4
```

```
## $teamID
## [1] NA
##
## $Rank
## [1] 4.028192
##
## $G
## [1] 150.1254
##
## $W
## [1] 74.67463
##
## $L
## [1] 74.67463
##
## $R
## [1] 681.1589
##
## $RA
## [1] 681.1579
```