Week3-Rmarkdown

Tianheng Z.

2021/10/10

1.Random Experiments, events and sample spaces

```
#Load the library
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.0     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.5
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

2.Tidy data and iteration

2.1 Missing Data and iteration

```
#Should use the purr package
impute_by_mean<-function(x){
  mu<-mean(x,na.rm=1) # first compute the mean of x
  impute_f<-function(z){ # coordinate-wise imputation
    if(is.na(z)){
    return(mu) # if z is na replace with mean
} else{
  return(z) # otherwise leave in place
  }
}
  return(map_dbl(x,impute_f)) # apply the map function to impute across vector
}
```

Now Create the impute_by_median

```
impute_by_median <- function(x){
  med <- median(x,na.rm = 1)
  impute_f <- function(z){
    if(is.na(z)){
      return(med)
    }
```

```
    else{
      return(z)
    }
  }
  return(map_dbl(x,impute_f))

}
#Test
v <- c(1,2,NA,4)
impute_by_median(v)
```

```
## [1] 1 2 2 4
```

Now Create df_xy

```
x <- seq(0,10,0.1)
y <- 5 *x +1
df_xy <- data.frame(x,y)
df_xy %>% head(5)
```

```
##     x   y
## 1 0.0 1.0
## 2 0.1 1.5
## 3 0.2 2.0
## 4 0.3 2.5
## 5 0.4 3.0
```

```
df_xy %>% mutate(z = map2_dbl(x,y,~.x+.y)) %>% head(5)
```

```
##     x   y   z
## 1 0.0 1.0 1.0
## 2 0.1 1.5 1.6
## 3 0.2 2.0 2.2
## 4 0.3 2.5 2.8
## 5 0.4 3.0 3.4
```

```
sometimes_missing <- function(index,value){
  mis_fuc <- function(a,b){
    if(a %% 5 ==0){
      return(NA)
    }
    else{
      return(b)
    }
  }
  return (map2_dbl(index,value,mis_fuc))
}
sometimes_missing(14,25)
```

```
## [1] 25
```

```
sometimes_missing(15,25)
```

```
## [1] NA
```

```
#Generate df_xy_missing
x <- df_xy$x
y <- map2_dbl(row_number(df_xy$y),df_xy$y,sometimes_missing)
```

```
df_xy_missing <- data.frame(x,y)
df_xy_missing %>% head(10)
```

```
##      x   y
## 1  0.0 1.0
## 2  0.1 1.5
## 3  0.2 2.0
## 4  0.3 2.5
## 5  0.4  NA
## 6  0.5 3.5
## 7  0.6 4.0
## 8  0.7 4.5
## 9  0.8 5.0
## 10 0.9  NA
```

```
x <- df_xy$x
y <- impute_by_median(df_xy_missing$y)
df_xy_imputed <- data.frame(x,y)
df_xy_imputed %>% head(10)
```

```
##      x    y
## 1  0.0  1.0
## 2  0.1  1.5
## 3  0.2  2.0
## 4  0.3  2.5
## 5  0.4 26.0
## 6  0.5  3.5
## 7  0.6  4.0
## 8  0.7  4.5
## 9  0.8  5.0
## 10 0.9 26.0
```

Combine the df

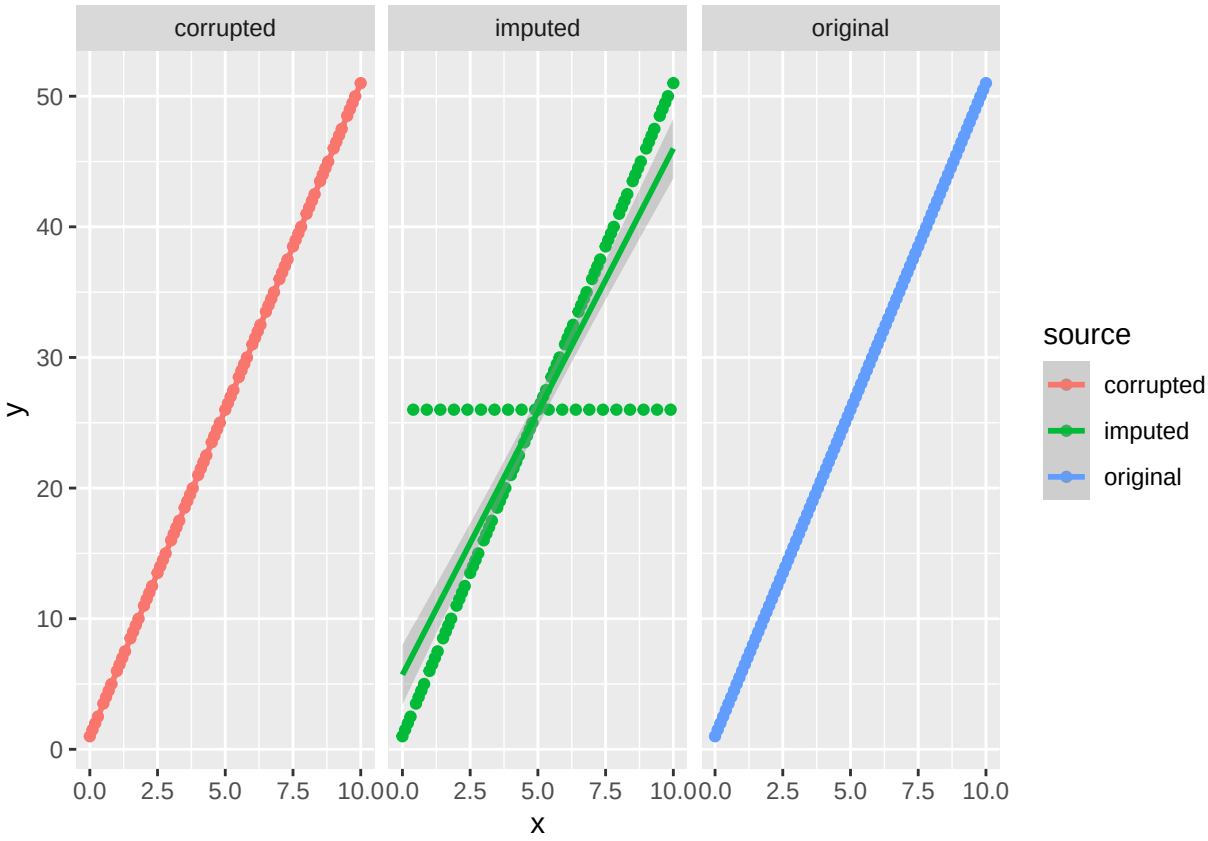
```
df_xy<-df_xy%>%
mutate(source="original")
df_xy_missing<-df_xy_missing%>%
mutate(source="corrupted")
df_xy_imputed<-df_xy_imputed%>%
mutate(source="imputed")
df_combined<-rbind(df_xy,df_xy_missing,df_xy_imputed)

ggplot(df_combined,aes(x=x,y=y,color = source)) + geom_point()+
  facet_wrap(~source) + geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 20 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 20 rows containing missing values (geom_point).
```



2.2 Tidying data with pivot functions

```
if(!require("readxl"))
install.packages("readxl")
```

```
## Loading required package: readxl
```

```
## Warning: package 'readxl' was built under R version 3.6.3
```

```
library(readxl)
library(tidyverse)
folder_path <- paste("C:\\Users\\zth2\\Desktop\\Bristol\\SCEM\\Week3\\FirstRproject", "\\Week3",sep = "
file_name <- "HockeyLeague.xlsx"

file_path <- paste(folder_path,file_name,sep = "\\")
wins_data_frame <- read_excel(file_path,sheet = "Wins")
```

```
## New names:
## * `` -> ...1
```

```
wins_data_frame %>%tibble() %>% select(1:5) %>% head(3)
```

```
## # A tibble: 3 x 5
##   ...1   `1990`   `1991`   `1992`   `1993`
##   <chr>  <chr>    <chr>    <chr>    <chr>
## 1 Ducks  30 of 50 11 of 50 30 of 50 12 of 50
## 2 Eagles 24 of 50 12 of 50 37 of 50 14 of 50
```

```
## 3 Hawks  20 of 50 22 of 50 33 of 50 11 of 50
```

Not tidy data

```
wins_tidy <- wins_data_frame %>%
  rename(Team = ...1) %>%
  pivot_longer(cols = !Team,names_to = "Year") %>%
  separate(col = value,into = c("Wins","Total"),sep = "of")
wins_tidy %>% dim()
```

```
## [1] 248   4
```

```
wins_tidy %>% head(5)
```

```
## # A tibble: 5 x 4
##   Team  Year  Wins  Total
##   <chr> <chr> <chr> <chr>
## 1 Ducks 1990  "30 " " 50"
## 2 Ducks 1991  "11 " " 50"
## 3 Ducks 1992  "30 " " 50"
## 4 Ducks 1993  "12 " " 50"
## 5 Ducks 1994  "24 " " 50"
```

Create losses_tidy

```
losses_data_frame <- read_excel(file_path,sheet = "Losses")
```

```
## New names:
## * `` -> ...1
```

```
losses_tidy <- losses_data_frame %>%
  tibble() %>%
  rename(Team = ...1) %>%
  pivot_longer(cols = !Team,names_to = "Year") %>%
  separate(col = value, into= c("Losses","Total"),sep = "of")

losses_tidy %>%head(5)
```

```
## # A tibble: 5 x 4
##   Team  Year  Losses Total
##   <chr> <chr> <chr>  <chr>
## 1 Ducks 1990  "20 "   " 50"
## 2 Ducks 1991  "37 "   " 50"
## 3 Ducks 1992  "1 "    " 50"
## 4 Ducks 1993  "30 "   " 50"
## 5 Ducks 1994  "7 "    " 50"
```

Combine two dataframe

```
hockey_df <- wins_tidy %>%
  inner_join(losses_tidy) %>%
  mutate(Wins = as.integer(Wins),Total = as.integer(Total),
```

```
        Losses = as.integer(Losses)) %>%
  mutate(Draws = Total - Wins- Losses,
        Wins_rt = Wins/Total,
        Losses_rt = Losses / Total,
        Draws_rt = Draws/Total)
```

```
## Joining, by = c("Team", "Year", "Total")
```

```
hockey_df %>% head(5)
```

```
## # A tibble: 5 x 9
##   Team  Year   Wins Total Losses Draws Wins_rt Losses_rt Draws_rt
##   <chr> <chr> <int> <int>  <int> <int>   <dbl>     <dbl>    <dbl>
## 1 Ducks 1990     30    50     20     0    0.6       0.4      0
## 2 Ducks 1991     11    50     37     2    0.22      0.74     0.04
## 3 Ducks 1992     30    50      1    19    0.6       0.02     0.38
## 4 Ducks 1993     12    50     30     8    0.24      0.6      0.16
## 5 Ducks 1994     24    50      7    19    0.48      0.14     0.38
```

Conclude

```
hockey_df %>% group_by(Team)%>% summarise(median_win_rt = median(Wins_rt),
                        mean_win_rt = mean(Wins_rt),
                        median_Losses_rt = median(Losses_rt),
                        mean_Losses_rt = mean(Losses_rt),
                        median_Draws_rt = median(Draws_rt),
                        mean_Draws_rt = mean(Draws_rt)) %>%
  arrange(desc(median_win_rt))
```

```
## # A tibble: 8 x 7
##   Team  median_win_rt mean_win_rt median_Losses_rt mean_Losses_rt
##   <chr>         <dbl>       <dbl>            <dbl>          <dbl>
## 1 Eagl~         0.45        0.437            0.25           0.279
## 2 Peng~         0.45        0.457            0.3            0.310
## 3 Hawks         0.417       0.388            0.233          0.246
## 4 Ducks         0.383       0.362            0.34           0.333
## 5 Owls          0.32        0.333            0.3            0.33
## 6 Ostr~         0.3         0.309            0.4            0.395
## 7 Stor~         0.3         0.284            0.22           0.283
## 8 King~         0.233       0.245            0.34           0.360
## # ... with 2 more variables: median_Draws_rt <dbl>, mean_Draws_rt <dbl>
```

2.3 Most correlated variables

```
max_cor_var<-function(df,col_name){
# function to determine the variable with maximal correlation
v_col<-df%>%select(all_of(col_name))
# extract variable based on col_name
df_num<-df%>%
select_if(is.numeric)%>%
select(-all_of(col_name))
# select all numeric variables excluding col_name
```

```
correlations<-unlist(map(df_num,
function(x){cor(x,v_col,use="complete.obs")}))
# compute correlations with all other numeric variables
max_abs_cor_var<-names(which(abs(correlations)==max(abs(correlations))))
# extract the variable name
cor<-as.double(correlations[max_abs_cor_var])
# compute the correlation
return(data.frame(var_name=max_abs_cor_var,cor=cor))
# return dataframe
}
```

```
top_correlates_by_var <- function(df){
  df_names <- names(df)

  #Five numeric cols
  df_num2 <- df %>% select_if(is.numeric) %>%names()
  #print(df_num2)

  df_map_func <- function(df_colname){
    df_num <- df%>% select_if(is.numeric) %>%
      select(-all_of(df_colname))  #other 4 cols

    df_mine <- df %>% select(df_colname)

    #print(df_colname)
    correlations<-unlist(map(df_num,function(x){cor(x,df_mine,use="complete.obs")}))
    # compute correlations with all other numeric variables
    max_abs_cor_var<-names(which(abs(correlations)==max(abs(correlations))))
    #print(max_abs_cor_var)
    # extract the variable name
    cor<-as.double(correlations[max_abs_cor_var])
    return(as.character(max_abs_cor_var))
  }

  #Should add results first then change the names
  results <- map_chr(df_num2,df_map_func)
  results_df<- as.data.frame(matrix(nrow = 0,ncol = length(df_num2)))
  final_results <- results_df %>%rbind(results)
  colnames(final_results) <- df_num2

  return(final_results %>% mutate(across(everything(),as.character)))

}
```

```
library(palmerpenguins)
```

```
## Warning: package 'palmerpenguins' was built under R version 3.6.3
```

```
penguins%>%top_correlates_by_var()
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(df_colname)` instead of `df_colname` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
##      bill_length_mm     bill_depth_mm flipper_length_mm       body_mass_g
```

```
## 1 flipper_length_mm flipper_length_mm       body_mass_g flipper_length_mm
##                year
## 1 flipper_length_mm
```

```
Adelie_re <- penguins %>% filter(species == "Adelie") %>% top_correlates_by_var()
Gentoo_re <- penguins %>% filter(species == "Gentoo") %>% top_correlates_by_var()
Chinstrap_re <- penguins %>% filter(species == "Chinstrap") %>% top_correlates_by_var()

as_tibble(rbind(Adelie_re,Gentoo_re,Chinstrap_re)) %>%
  mutate(species = c("Adelie","Gentoo","Chinstrap")) %>%
  select(6,1:5) %>% mutate(across(everything(),as.character),species = as.factor(species))
```

```
## # A tibble: 3 x 6
##   species  bill_length_mm bill_depth_mm  flipper_length_mm body_mass_g  year
##   <fct>    <chr>          <chr>          <chr>             <chr>        <chr>
## 1 Adelie   body_mass_g    body_mass_g    body_mass_g       bill_depth_~ flipper~
## 2 Gentoo   body_mass_g    body_mass_g    bill_depth_mm     bill_depth_~ bill_de~
## 3 Chinstr~ bill_depth_mm  bill_length_mm body_mass_g       flipper_len~ flipper~
```

3. Elementary set theory

4. Introduction to probability