Programming in Base R

Task 1: Basic Vector practice

1.Create two vectors

```
pre_bp <-c(130,128,116,124,133,134,118,126,114,127,141,138,128,140,137,131,120,128,139,135)
post_bp <-c(114,98,113,99,107,116,113,111,119,117,101,119,130,122,106,106,124,102,117,113)
```

2.Assign names to the vector elements

```
paste("Subject", 1:20,sep="_")
```

```
 [1] "Subject_1"  "Subject_2"  "Subject_3"  "Subject_4"  "Subject_5"
 [6] "Subject_6"  "Subject_7"  "Subject_8"  "Subject_9"  "Subject_10"
[11] "Subject_11" "Subject_12" "Subject_13" "Subject_14" "Subject_15"
[16] "Subject_16" "Subject_17" "Subject_18" "Subject_19" "Subject_20"
```

```
names(pre_bp) <-c(paste("Subject", 1:20,sep="_"))
names(post_bp) <-c(paste("Subject", 1:20,sep="_"))
```

3.Calculate the change in blood pressure

```
diff_bp <- c(pre_bp - post_bp)
```

4.Calculate the average decrease in blood pressure

```r
mean(diff_bp)
```

```
[1] 17
```

5. Determine positive change

```r
which(diff_bp>0)
```

```
 Subject_1  Subject_2  Subject_3  Subject_4  Subject_5  Subject_6  Subject_7
         1          2          3          4          5          6          7
 Subject_8 Subject_10 Subject_11 Subject_12 Subject_14 Subject_15 Subject_16
         8         10         11         12         14         15         16
Subject_18 Subject_19 Subject_20
        18         19         20
```

6.Subset the vector of differences only return positive change

```r
positive_change <- diff_bp[c(which(diff_bp>0))]
```

7. Calculate average decrease in blood pressure

```r
mean(positive_change)
```

```
[1] 20.64706
```

Task 2: Basic Data Frame practice

1. Create a data frame object with 4 columns

```r
patient <-c(paste("Subject", 1:20,sep="_"))
bp_df <-data.frame(patient,pre_bp,post_bp,diff_bp,row.names=NULL)
```

2. Return only rows with negative diff_bp

```
`[`(bp_df,bp_df$diff_bp<0, )
```

```
      patient pre_bp post_bp diff_bp
9   Subject_9    114     119      -5
13 Subject_13    128     130      -2
17 Subject_17    120     124      -4
```

```
bp_df[bp_df$diff_bp<0, ]
```

```
      patient pre_bp post_bp diff_bp
9   Subject_9    114     119      -5
13 Subject_13    128     130      -2
17 Subject_17    120     124      -4
```

```
subset(bp_df,diff_bp<0)
```

```
      patient pre_bp post_bp diff_bp
9   Subject_9    114     119      -5
13 Subject_13    128     130      -2
17 Subject_17    120     124      -4
```

3. Add a new column (TRUE if post_bp<120)

```
bp_df$post_bp_less_120 <-ifelse(bp_df$post_bp<120,TRUE,FALSE)
```

4. Print the data frame out

```
knitr::kable(bp_df,align='c',"simple")
```

patient	pre_bp	post_bp	diff_bp	post_bp_less_120
Subject_1	130	114	16	TRUE
Subject_2	128	98	30	TRUE

patient	pre_bp	post_bp	diff_bp	post_bp_less_120
Subject_3	116	113	3	TRUE
Subject_4	124	99	25	TRUE
Subject_5	133	107	26	TRUE
Subject_6	134	116	18	TRUE
Subject_7	118	113	5	TRUE
Subject_8	126	111	15	TRUE
Subject_9	114	119	-5	TRUE
Subject_10	127	117	10	TRUE
Subject_11	141	101	40	TRUE
Subject_12	138	119	19	TRUE
Subject_13	128	130	-2	FALSE
Subject_14	140	122	18	FALSE
Subject_15	137	106	31	TRUE
Subject_16	131	106	25	TRUE
Subject_17	120	124	-4	FALSE
Subject_18	128	102	26	TRUE
Subject_19	139	117	22	TRUE
Subject_20	135	113	22	TRUE

Task 3: List practice

1. Create a new data frame

```
patient <-paste("Subject", 1:10,sep="_")
pre_placebo <-c(138,135,147,117,152,134,114,121,131,130)
post_placebo <-c(105,136,123,130,134,143,135,139,120,124)
diff_placebo <- pre_placebo - post_placebo
placebo_df <-data.frame(patient,pre_bp=pre_placebo,post_bp=post_placebo,diff_bp=diff_placebo)
placebo_df$post_bp_less_120 <-ifelse(placebo_df$post_bp<120,TRUE,FALSE)
```

2. Create and store a list with two elements

```
my_list <- list(treatment=bp_df, placebo=placebo_df)
```

3. Access the first list element using three different types of syntax

```
my_list[1]
```

```
$treatment
      patient pre_bp post_bp diff_bp post_bp_less_120
1   Subject_1    130     114      16             TRUE
2   Subject_2    128      98      30             TRUE
3   Subject_3    116     113       3             TRUE
4   Subject_4    124      99      25             TRUE
5   Subject_5    133     107      26             TRUE
6   Subject_6    134     116      18             TRUE
7   Subject_7    118     113       5             TRUE
8   Subject_8    126     111      15             TRUE
9   Subject_9    114     119      -5             TRUE
10 Subject_10    127     117      10             TRUE
11 Subject_11    141     101      40             TRUE
12 Subject_12    138     119      19             TRUE
13 Subject_13    128     130      -2            FALSE
14 Subject_14    140     122      18            FALSE
15 Subject_15    137     106      31             TRUE
16 Subject_16    131     106      25             TRUE
17 Subject_17    120     124      -4            FALSE
18 Subject_18    128     102      26             TRUE
19 Subject_19    139     117      22             TRUE
20 Subject_20    135     113      22             TRUE
```

```
my_list[[1]]
```

```
      patient pre_bp post_bp diff_bp post_bp_less_120
1   Subject_1    130     114      16             TRUE
2   Subject_2    128      98      30             TRUE
3   Subject_3    116     113       3             TRUE
4   Subject_4    124      99      25             TRUE
5   Subject_5    133     107      26             TRUE
6   Subject_6    134     116      18             TRUE
7   Subject_7    118     113       5             TRUE
8   Subject_8    126     111      15             TRUE
9   Subject_9    114     119      -5             TRUE
10 Subject_10    127     117      10             TRUE
```

```
11 Subject_11    141     101      40             TRUE
12 Subject_12    138     119      19             TRUE
13 Subject_13    128     130      -2            FALSE
14 Subject_14    140     122      18            FALSE
15 Subject_15    137     106      31             TRUE
16 Subject_16    131     106      25             TRUE
17 Subject_17    120     124      -4            FALSE
18 Subject_18    128     102      26             TRUE
19 Subject_19    139     117      22             TRUE
20 Subject_20    135     113      22             TRUE
```

```
my_list$treatment
```

```
      patient pre_bp post_bp diff_bp post_bp_less_120
1   Subject_1    130     114      16             TRUE
2   Subject_2    128      98      30             TRUE
3   Subject_3    116     113       3             TRUE
4   Subject_4    124      99      25             TRUE
5   Subject_5    133     107      26             TRUE
6   Subject_6    134     116      18             TRUE
7   Subject_7    118     113       5             TRUE
8   Subject_8    126     111      15             TRUE
9   Subject_9    114     119      -5             TRUE
10 Subject_10    127     117      10             TRUE
11 Subject_11    141     101      40             TRUE
12 Subject_12    138     119      19             TRUE
13 Subject_13    128     130      -2            FALSE
14 Subject_14    140     122      18            FALSE
15 Subject_15    137     106      31             TRUE
16 Subject_16    131     106      25             TRUE
17 Subject_17    120     124      -4            FALSE
18 Subject_18    128     102      26             TRUE
19 Subject_19    139     117      22             TRUE
20 Subject_20    135     113      22             TRUE
```

4. In one line, access the placebo data frame, pre_bp column

```
my_list$placebo$pre_bp
```

```
 [1] 138 135 147 117 152 134 114 121 131 130
```

Task 4: Control Flow Practice

1.create a new column called status

```
my_list$treatment$status <- character(20)
my_list$placebo$status <-character(10)
```

2. create a for loop and use if/then/else logic to create status values for non-placebo data frame

```
# my_list$treatment$status <- ifelse(my_list$treatment$post_bp<=120,"optimal",ifelse(my_list$

for (i in 1:20){
  if (my_list$treatment$post_bp[i] <=120) {
    my_list$treatment$status[i]<-"optimal"
  } else if (my_list$treatment$post_bp[i] >130) {
    my_list$treatment$status[i]<-"high"
  } else {
    my_list$treatment$status[i]<-"borderline"
  }
}
my_list$treatment$status
```

```
 [1] "optimal"    "optimal"    "optimal"    "optimal"    "optimal"
 [6] "optimal"    "optimal"    "optimal"    "optimal"    "optimal"
[11] "optimal"    "optimal"    "borderline" "borderline" "optimal"
[16] "optimal"    "borderline" "optimal"    "optimal"    "optimal"
```

3.create a for loop and use if/then/else logic to create status values for placebo data frame

```
for (i in 1:10) {
  if (my_list$placebo$post_bp[i]<=120){
    my_list$placebo$status[i]<-"optimal"
  } else if (my_list$placebo$post_bp[i]>130) {
    my_list$placebo$status[i]<-"high"
  } else {my_list$placebo$status[i]<-"borderline"
```

```
  }
}
my_list$placebo$status
```

```
 [1] "optimal"    "high"       "borderline" "borderline" "high"
 [6] "high"       "high"       "high"       "optimal"    "borderline"
```

Task 5: Function Writing

1.Write a function that

*takes in a list with two data frames in it

```
take_in_list <- function(data_frame1,data_frame2){
  list(treatmnet=data_frame1,placebo=data_frame2)
}
```

*takes in an R function find a summary of a numeric column (default set to "mean")

```
summary_mean <- function(column){
  if(!is.numeric(column)){
    stop("Not numeric column")
  }
  mean(column)
}
summary_mean(positive_change)
```

```
[1] 20.64706
```

- find statistic of interest for the pre,post,diff columns of both data frames

```
find_stat <- function(stat){
  my_fun <-get(stat)
  bp_pre<-my_fun(bp_df$pre_bp)
  bp_post<-my_fun(bp_df$post_bp)
  bp_diff<-my_fun(bp_df$diff_bp)
  pla_pre<-my_fun(placebo_df$pre_bp)
  pla_post<-my_fun(placebo_df$post_bp)
  pla_diff<-my_fun(placebo_df$diff_bp)
  return(list(bp_pre,bp_post,bp_diff,pla_pre,pla_post,pla_diff))
```

```
  #return six values as a named list
  value_list<-find_stat(stat)
  #create a vector of names
   if (mean) {
    vec_name=c(paste("mean","_",1:6,sep=''))
  } else if (var) {
   vec_name=c(paste("var","_",1:6,sep=''))
  } else if (sd) {
   vec_name=c(paste("sd","_",1:6,sep=''))
  } else if (min) {
   vec_name=c(paste("min","_",1:6,sep=''))
  } else if (max) {
   vec_name=c(paste("max","_",1:6,sep=''))
  }

  #create a vector of stat values
  vec_value<-c(value_list[1],value_list[2],value_list[3],
               value_list[4],value_list[5],value_list[6])
  #assign names() to the vector
  names(vec_value)=c(vec_name)

}

#apply function (var,sd,min,max)
find_stat("mean")
```

```
[[1]]
[1] 129.35

[[2]]
[1] 112.35

[[3]]
[1] 17

[[4]]
[1] 131.9

[[5]]
[1] 128.9
```

```
[[6]]
[1] 3
```

```
find_stat('var')
```

```
[[1]]
[1] 64.55526

[[2]]
[1] 74.76579

[[3]]
[1] 153.6842

[[4]]
[1] 149.8778

[[5]]
[1] 124.9889

[[6]]
[1] 341.3333
```

```
find_stat('sd')
```

```
[[1]]
[1] 8.034629

[[2]]
[1] 8.646721

[[3]]
[1] 12.39694

[[4]]
[1] 12.24246

[[5]]
[1] 11.17984

[[6]]
[1] 18.47521
```

```
find_stat("min")
```

```
[[1]]
[1] 114

[[2]]
[1] 98

[[3]]
[1] -5

[[4]]
[1] 114

[[5]]
[1] 105

[[6]]
[1] -21
```

```
find_stat("max")
```

```
[[1]]
[1] 141

[[2]]
[1] 130

[[3]]
[1] 40

[[4]]
[1] 152

[[5]]
[1] 143

[[6]]
[1] 33
```