

# Basic Programming in R

## Task 1: Basic Vector practice

### Question 1

```
# Load in pre- and post-treatment data from assignment
pre <- c(130, 128, 116, 124, 133, 134, 118, 126, 114, 127,
        141, 138, 128, 140, 137, 131, 120, 128, 139, 135)
post <- c(114, 98, 113, 99, 107, 116, 113, 111, 119, 117,
        101, 119, 130, 122, 106, 106, 124, 102, 117, 113)
```

### Question 2

```
# Assign names to each observation in pre and post objects
subj <- paste("Subject", 1:20, sep = "_")
names(pre) <- subj
names(post) <- subj
```

### Question 3

```
# Take the difference in measurements for each subject and print the results
diff_op <- pre - post
diff_op
```

Subject_1	Subject_2	Subject_3	Subject_4	Subject_5	Subject_6	Subject_7
16	30	3	25	26	18	5
Subject_8	Subject_9	Subject_10	Subject_11	Subject_12	Subject_13	Subject_14

15	-5	10	40	19	-2	18
Subject_15	Subject_16	Subject_17	Subject_18	Subject_19	Subject_20	
31	25	-4	26	22	22	

#### Question 4

```
# Take the mean of diff_op to find the average decrease in blood pressure
# across the subjects
mean(diff_op)
```

```
[1] 17
```

#### Question 5

```
# Find which patients experienced a decrease in blood pressure
decrease <- which(diff_op > 0)
decrease
```

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				

#### Question 6

```
# Take the subset of patients who experienced a decrease in blood pressure
diff_op_decrease <- diff_op[decrease]
```

#### Question 7

```
# Get the average decrease in blood pressure for the subset of patients
# we just derived
mean(diff_op_decrease)
```

```
[1] 20.64706
```

## Task 2: Basic Data Frame practice

### Question 1

```
# Using objects from Task 1, create a data frame!
patient <- subj
pre_bp <- pre
post_bp <- post
diff_bp <- diff_op

bp_df <- data.frame(patient, pre_bp, post_bp, diff_bp)
rownames(bp_df) <- NULL # No longer need row names to identify patients
```

### Question 2

```
# Subset data frame to find negative diff_bp values
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

### Question 3

```
# Add a true/false column for blood pressures less than 120
bp_df$normal <- ifelse(bp_df$post_bp < 120, TRUE, FALSE)
```

### Question 4

```
knitr::kable(bp_df)
```

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

#### Question 1

```
# Using code from tasks 1 and 2, create a data frame for the placebo
pre_p <- c(138, 135, 147, 117, 152, 134, 114, 121, 131, 130)
post_p <- c(105, 136, 123, 130, 134, 143, 135, 139, 120, 124)

subj_p <- paste("Subject", 1:10, sep = "_")
names(pre_p) <- subj_p
names(post_p) <- subj_p

diff_p <- pre_p - post_p

# Create the dataframe
bp_df_placebo <- data.frame("patient" = subj_p, "pre" = pre_p,
                           "post" = post_p, "diff" = diff_p)
```

```
#Add the normal column
bp_df_placebo$normal <- ifelse(bp_df_placebo$post < 120, TRUE, FALSE)
```

## Question 2

```
# Create a list of two data frames
bp_list <- list("treatment" = bp_df, "placebo" = bp_df_placebo)
```

## Question 3

```
# Access the first element of the list in three different ways
bp_list[1]
```

```
$treatment
  patient pre_bp post_bp diff_bp normal
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
```

```
bp_list[[1]]
```

	patient	pre_bp	post_bp	diff_bp	normal
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

```
bp_list$treatment
```

	patient	pre_bp	post_bp	diff_bp	normal
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

#### Question 4

```
#Access the pre column in the placebo dataset
bp_list[[2]]$pre
```

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

### Task 4: Control Flow Practice

#### Question 1

```
# Create an empty column for the new status variable
bp_list[[1]]$status <- character(20)
bp_list[[2]]$status <- character(10)
```

#### Question 2

```
#Loop through each element of the list and assign a status to the
#patient's post-treatment blood pressure
for (i in 1:20) {
  if (bp_list[[1]]$post_bp[i] <= 120) {
    bp_list[[1]]$status[i] <- "Optimal"
  } else if (bp_list[[1]]$post_bp[i] > 120 || bp_list[[1]]$post_bp <= 130) {
    bp_list[[1]]$status[i] <- "Borderline"
  } else if (bp_list[[1]]$post_bp[i] > 130) {
    bp_list[[1]]$status[i] <- "High"
  } else {
    bp_list[[1]]$status[i] <- "ERROR"
  }
}
```

```
# Check the results!
bp_list[[1]]$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"
```

### Question 3

```
#Loop through each element of the list and assign a status to the
#patient's post-treatment blood pressure
for (i in 1:10) {
  if (bp_list[[2]]$post[i] <= 120) {
    bp_list[[2]]$status[i] <- "Optimal"
  } else if (bp_list[[2]]$post[i] > 120 || bp_list[[1]]$post <= 130) {
    bp_list[[2]]$status[i] <- "Borderline"
  } else if (bp_list[[2]]$postp[i] > 130) {
    bp_list[[2]]$status[i] <- "High"
  } else {
    bp_list[[2]]$status[i] <- "ERROR"
  }
}

# Check the results!
bp_list[[2]]$status
```

```
[1] "Optimal"      "Borderline"   "Borderline"   "Borderline"   "Borderline"
[6] "Borderline"   "Borderline"   "Borderline"   "Optimal"      "Borderline"
```

## Task 5: Function Writing

### Question 1

```
# Create summary_stats function
summary_stats <- function(list_obj, stat = "mean") {
  # Sanity checks to ensure correct arguments are passed
```



```

if(!is.list(list_obj)) {
  stop("list_obj argument did not pass a list.")
} else if (!length(list_obj) == 2) {
  stop("list_obj argument is not of length 2.")
} else if (!(is.data.frame(list_obj[[1]]) || is.data.frame(list_obj[[2]]))) {
  stop("One or more objects within list_obj are not of type data.frame")
}

# Get stat type to pass through
my_fun <- get(stat)

# Create a vector of names based on the stat value and an empty vector store values
names_dynamic <- paste(stat, c("pre", "post", "diff", "pre", "post", "diff"),
                        c("trt", "trt", "trt", "placebo", "placebo", "placebo"),
                        sep = "_")

# Apply the my_fun stat to each relevant column and apply the name
vec <- c(colMeans(bp_list[[1]][2:4]), colMeans(bp_list[[2]][2:4]))
names(vec) <- names_dynamic

return(vec)
}

#Test out the brand new function
summary_stats(bp_list)

```

mean_pre_trt	mean_post_trt	mean_diff_trt	mean_pre_placebo
129.35	112.35	17.00	131.90
mean_post_placebo	mean_diff_placebo		
128.90	3.00		

```
summary_stats(bp_list, "var")
```

var_pre_trt	var_post_trt	var_diff_trt	var_pre_placebo
129.35	112.35	17.00	131.90
var_post_placebo	var_diff_placebo		
128.90	3.00		

```
summary_stats(bp_list, "sd")
```

sd_pre_trt	sd_post_trt	sd_diff_trt	sd_pre_placebo	sd_post_placebo
129.35	112.35	17.00	131.90	128.90
sd_diff_placebo				
3.00				

```
summary_stats(bp_list, "min")
```

min_pre_trt	min_post_trt	min_diff_trt	min_pre_placebo
129.35	112.35	17.00	131.90
min_post_placebo	min_diff_placebo		
128.90	3.00		

```
summary_stats(bp_list, "max")
```

max_pre_trt	max_post_trt	max_diff_trt	max_pre_placebo
129.35	112.35	17.00	131.90
max_post_placebo	max_diff_placebo		
128.90	3.00		