Project II: Comparison Of Multiple Distributions

Bushra Tariq Kiyani

2022-11-18

```
library(ggplot2)
library(ggpubr)
library(rstatix)
```

```
##
## Attaching package: 'rstatix'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
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
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
swimming_data <- read.csv("SwimmingTimes.csv")
head(swimming_data)
```

```
##     Category                  Name   Time
## 1 Backstroke       SonneleOeztuerk 133.97
## 2 Backstroke     AnastasyaGorbenko 131.46
## 3 Backstroke CamilaRodriguesRebelo 131.05
## 4 Backstroke            DoraMolnar 129.88
## 5 Backstroke         KatieShanahan 129.82
## 6 Backstroke    CarmenWeilerSastre 131.78
```

```
# Checking Number of Categories
unique(swimming_data$Category)
```

```
## [1] "Backstroke"   "Breaststroke" "Butterfly"    "Freestyle"    "Medley"
```

```r
# Ordering the data based on categories
swimming_data <- swimming_data[order(swimming_data$Category),]
```

```r
# Checking total number of observations in each group the data based on categories
swimming_data %>% group_by(Category) %>%tally()
```

```
## # A tibble: 5 x 2
##   Category         n
##   <chr>        <int>
## 1 Backstroke      16
## 2 Breaststroke    16
## 3 Butterfly       16
## 4 Freestyle       16
## 5 Medley          16
```

```r
# Summary of the Data
summary(swimming_data)
```

```
##    Category             Name                Time
##  Length:80          Length:80          Min.   :117.4
##  Class :character   Class :character   1st Qu.:129.3
##  Mode  :character   Mode  :character   Median :132.4
##                                        Mean   :132.6
##                                        3rd Qu.:135.5
##                                        Max.   :148.4
```

```r
#Check is there any missing values in Data
colSums(is.na(swimming_data))
```

```
## Category     Name     Time
##        0        0        0
```

```r
#same participants in more than one category
swimming_data[swimming_data$Name %in% swimming_data[duplicated(swimming_data$Name),]$Name,]
```

```
##        Category                 Name   Time
## 2    Backstroke    AnastasyaGorbenko 131.46
## 5    Backstroke        KatieShanahan 129.82
## 19 Breaststroke       KristynaHorska 145.55
## 24 Breaststroke        LauraLahtinen 147.60
## 33    Butterfly MireiaBelmonteGarcia 134.01
## 34    Butterfly        KatinkaHosszu 134.54
## 42    Butterfly        LauraLahtinen 131.41
## 51    Freestyle    MarritSteenbergen 117.40
## 67       Medley        KatieShanahan 131.84
## 69       Medley       KristynaHorska 132.99
## 70       Medley    MarritSteenbergen 132.31
## 76       Medley    AnastasyaGorbenko 132.91
## 77       Medley        KatinkaHosszu 132.52
## 79       Medley MireiaBelmonteGarcia 135.47
```

```r
count(swimming_data)
```

```
##    n
## 1 80
```

```r
#removing duplicate participants from one category
swimming_data  <- swimming_data  %>% filter(!(Name == "AnastasyaGorbenko" & Category == "Backstroke"))
```

```r
swimming_data  <- swimming_data  %>% filter(!(Name == "KatieShanahan" & Category == "Backstroke"))
swimming_data  <- swimming_data  %>% filter(!(Name == "KristynaHorska" & Category == "Breaststroke"))
swimming_data  <- swimming_data  %>% filter(!(Name == "LauraLahtinen" & Category == "Breaststroke"))
swimming_data  <- swimming_data  %>% filter(!(Name == "MireiaBelmonteGarcia" & Category == "Butterfly"))
swimming_data  <- swimming_data  %>% filter(!(Name == "KatinkaHosszu" & Category == "Butterfly"))
swimming_data  <- swimming_data  %>% filter(!(Name == "MarritSteenbergen" & Category == "Freestyle"))
```

```r
# No of observation after removing duplicates
swimming_data %>% group_by(Category) %>%tally()
```

```
## # A tibble: 5 x 2
##   Category         n
##   <chr>        <int>
## 1 Backstroke      14
## 2 Breaststroke    14
## 3 Butterfly       14
## 4 Freestyle       15
## 5 Medley          16
```

```r
# Summary after removing duplicates
summary(swimming_data$Time)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   117.7   129.2   132.3   132.4   135.5   148.4
```

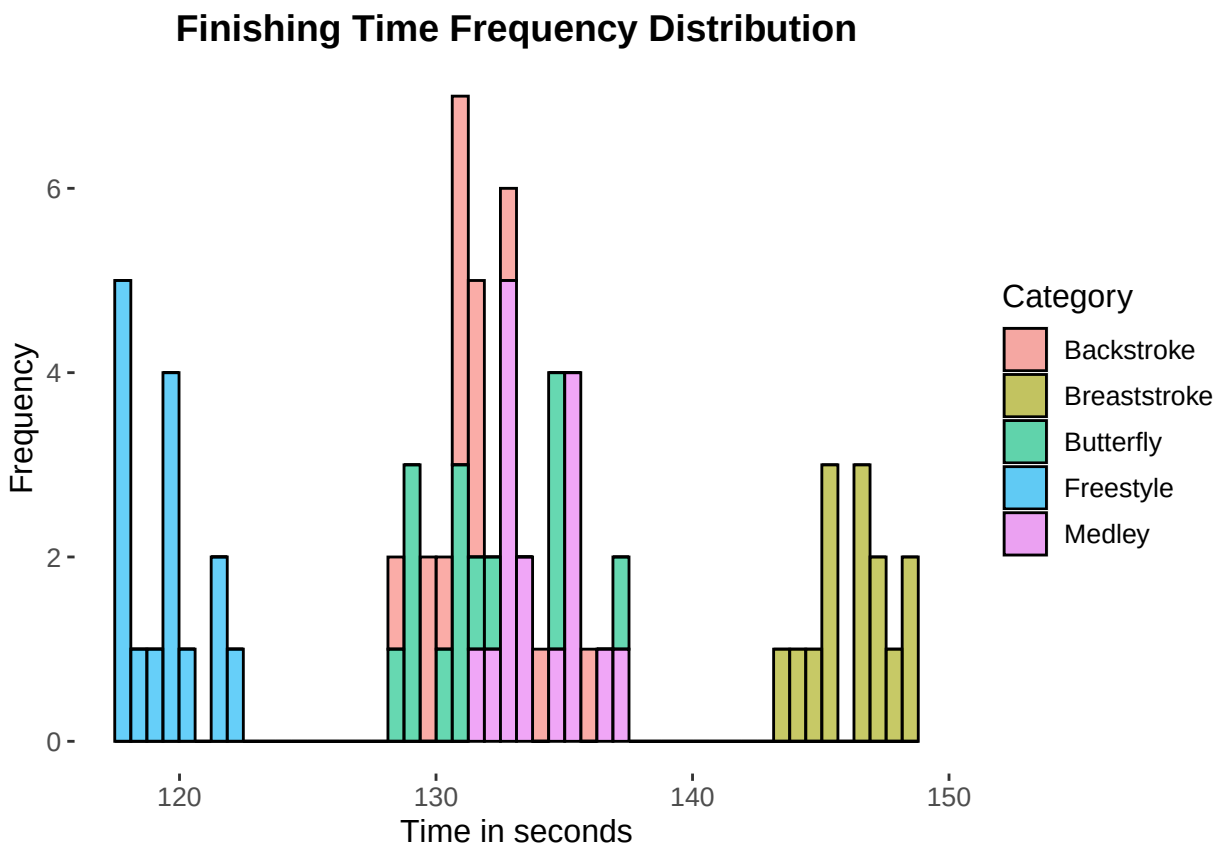
```r
#Standard deviation of time overall
round(sd(swimming_data$Time),3)
```

```
## [1] 8.789
```

```r
# Frequency distribution of finishing time
plot1 <- ggplot(swimming_data, aes(x = Time, fill = Category)) +
  geom_histogram( col = "black", bins= 50, alpha =0.6) +
  xlab("Time in seconds") + ylab("Frequency") +
  ggtitle("Finishing Time Frequency Distribution")+
  theme(plot.title = element_text(face = "bold", hjust = 0.5, size = 14 ),
        panel.background = element_rect(fill = "White"), axis.text=element_text(size=10),
                    axis.title=element_text(size=12), legend.text = element_text(size=10),
        legend.title = element_text(size=12))
ggsave("histogram.pdf",plot = plot1)
```

```
## Saving 6.5 x 4.5 in image
```

```r
plot1
```




```
swimming_data[swimming_data$Time<118,]
```

```
##     Category            Name   Time
## 45 Freestyle      JanjaSegel 117.94
## 47 Freestyle   FreyaAnderson 117.76
## 51 Freestyle CharlotteBonnet 117.73
## 52 Freestyle  NikolettaPadar 117.80
## 53 Freestyle IsabelMarieGose 117.70
```

```
#Table that lists the various statistical measures calculated on the variable sqmPrice
Analysistable <- group_by(swimming_data, Category) %>%
                           summarise(median = sprintf("%0.3f", median(Time, na.rm = TRUE)),
                                     mean = sprintf("%0.3f", mean(Time, na.rm = TRUE)),
                                     sd = sd(Time, na.rm = TRUE),
                                     variance = var(Time, na.rm = TRUE),
                                     minimum = min(Time, na.rm = TRUE),
                                     maximum = max(Time, na.rm = TRUE),
                                     IQR = quantile(Time, 3/4) - quantile(Time, 1/4))
Analysistable
```

```
## # A tibble: 5 x 8
##   Category     median  mean       sd variance minimum maximum   IQR
##   <chr>        <chr>   <chr>   <dbl>    <dbl>   <dbl>   <dbl> <dbl>
## 1 Backstroke   131.115 131.380  1.85     3.43    128.    136.  1.52
## 2 Breaststroke 146.660 146.314  1.51     2.27    144.    148.  2.00
## 3 Butterfly    130.835 131.656  2.61     6.82    128.    137.  4.42
## 4 Freestyle    119.500 119.358  1.56     2.43    118.    122.  2.23
## 5 Medley       133.455 134.040  1.59     2.52    132.    137.  2.47
```

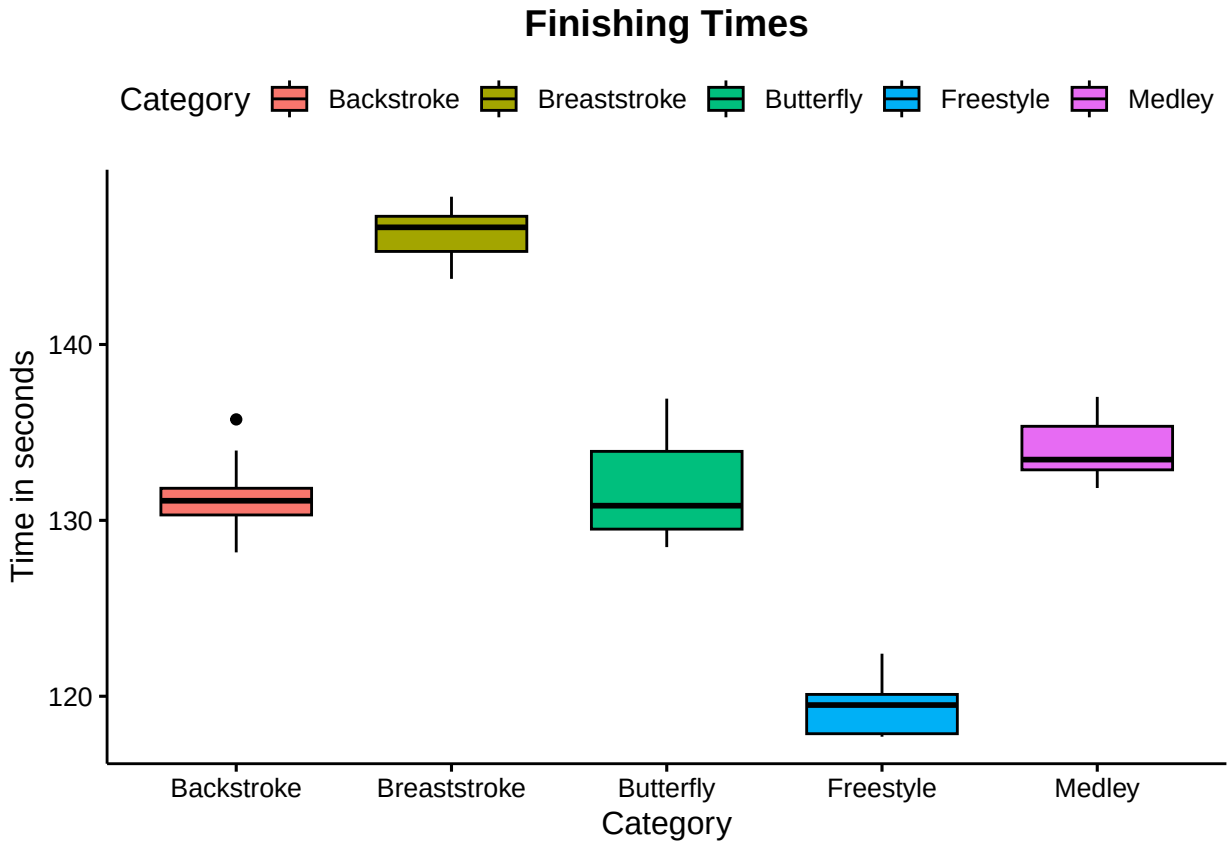
Verifying the Assumptions

1. Homogeneity of variance assumption

```
#Box plot to compare finishing time in the different categories to find the homogeneity in variance
plot2 <- ggboxplot(swimming_data, x = "Category", y = "Time", color = "black", fill = "Category",
                   ylab = "Time in seconds", xlab = "Category") + ggtitle('Finishing Times') +
  theme(plot.title = element_text(face = "bold",hjust = 0.5, size = 14),
        axis.text=element_text(size=10), axis.title=element_text(size=12),
        legend.text = element_text(size = 10),legend.title = element_text(size = 12))
ggsave("boxlot.pdf",plot = plot2)
```

```
## Saving 6.5 x 4.5 in image
```

```
plot2
```

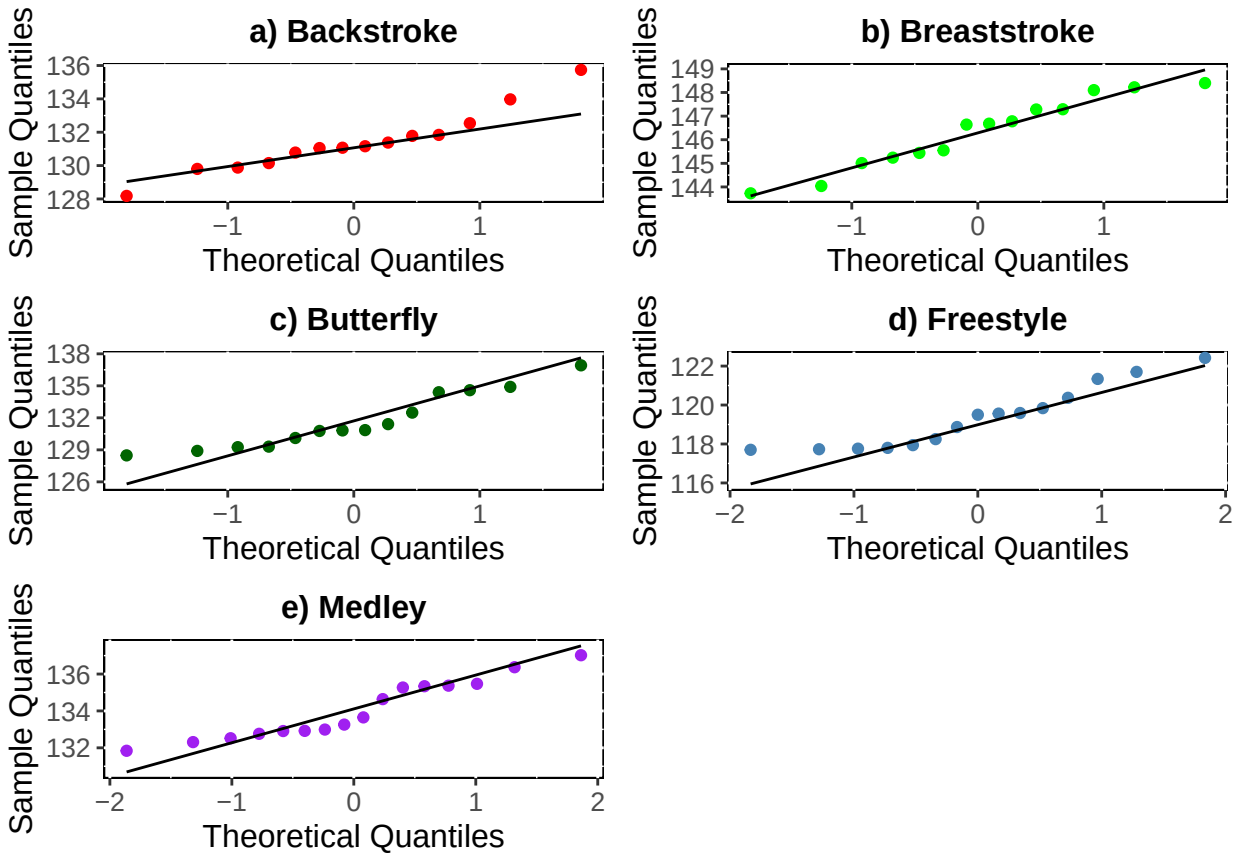


2. Normality assumption

```
Backstroke <- swimming_data %>% filter(Category == "Backstroke")
Breaststroke <- swimming_data %>% filter(Category == "Breaststroke")
Butterfly <- swimming_data %>% filter(Category == "Butterfly")
Freestyle <- swimming_data %>% filter(Category == "Freestyle")
Medley <- swimming_data %>% filter(Category == "Medley")
```

```
plot4 <- ggplot(Backstroke) + stat_qq(aes(sample = Time), color= "red")+
  stat_qq_line(aes(sample = Time)) +  scale_x_continuous(name = "Theoretical Quantiles") +
  scale_y_continuous(name = "Sample Quantiles") + ggtitle("a) Backstroke") +
```

```
  theme(panel.background = element_rect(fill = "White", color = "black"),
        plot.title = element_text(face = "bold",hjust = 0.5, size = 12),
        axis.text=element_text(size=10),
        axis.title=element_text(size=12), legend.text = element_text(size = 12))


plot5 <- ggplot(Breaststroke) + stat_qq(aes(sample = Time), color= "green")+
  stat_qq_line(aes(sample = Time)) +  scale_x_continuous(name = "Theoretical Quantiles") +
  scale_y_continuous(name = "Sample Quantiles") + ggtitle("b) Breaststroke") +
  theme(panel.background = element_rect(fill = "White", color = "black"),
        plot.title = element_text(face = "bold",hjust = 0.5, size = 12),
        axis.text=element_text(size=10), axis.title=element_text(size=12),
        legend.text = element_text(size = 12))

plot6 <- ggplot(Butterfly) + stat_qq(aes(sample = Time), color= "darkgreen")+
  stat_qq_line(aes(sample = Time)) +  scale_x_continuous(name = "Theoretical Quantiles") +
  scale_y_continuous(name = "Sample Quantiles") + ggtitle("c) Butterfly") +
  theme(panel.background = element_rect(fill = "White", color = "black"),
        plot.title = element_text(face = "bold",hjust = 0.5, size = 12),
        axis.text=element_text(size=10), axis.title=element_text(size=12),
        legend.text = element_text(size = 12))

plot7 <- ggplot(Freestyle) + stat_qq(aes(sample = Time), color= "steelblue")+
  stat_qq_line(aes(sample = Time)) +  scale_x_continuous(name = "Theoretical Quantiles") +
  scale_y_continuous(name = "Sample Quantiles") + ggtitle("d) Freestyle") +
  theme(panel.background = element_rect(fill = "White", color = "black"),
        plot.title = element_text(face = "bold",hjust = 0.5, size = 12),
        axis.text=element_text(size=10), axis.title=element_text(size=12),
        legend.text = element_text(size = 12))

plot8 <- ggplot(Medley) + stat_qq(aes(sample = Time), color= "purple")+
  stat_qq_line(aes(sample = Time)) +  scale_x_continuous(name = "Theoretical Quantiles") +
  scale_y_continuous(name = "Sample Quantiles") + ggtitle("e) Medley") +
  theme(panel.background = element_rect(fill = "White", color = "black"),
        plot.title = element_text(face = "bold",hjust = 0.5, size = 12),
        axis.text=element_text(size=10), axis.title=element_text(size=12),
        legend.text = element_text(size = 12))

final_plot1 <- grid.arrange(plot4, plot5, plot6, plot7,plot8, ncol=2, nrow = 3)
```




```
ggsave("QQplots.pdf",plot = final_plot1)
```

```
## Saving 6.5 x 4.5 in image
```

```
final_plot1
```

```
## TableGrob (3 x 2) "arrange": 5 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
## 5 5 (3-3,1-1) arrange gtable[layout]
```

3. Independence assumption

```
#same participants in more than one category
swimming_data[swimming_data$Name %in% swimming_data[duplicated(swimming_data$Name),]$Name,]
```

```
## [1] Category Name     Time
## <0 rows> (or 0-length row.names)
```

Task 1

One-way ANOVA test

```
# Compute the analysis of variance
one_way_anova <- aov(Time ~ Category, data = swimming_data)
# Summary of the analysis
summary(one_way_anova)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Category     4   5327  1331.8   385.9 <2e-16 ***
## Residuals   68    235     3.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Task 2

Multiple T-Tests

```
#List of pairs made of the 5 Categories
pair_category  <- c("Backstroke_Breaststroke","Backstroke_Butterfly","Backstroke_Freestyle",
                    "Backstroke_Medley","Breaststroke_Butterfly","Breaststroke_Freestyle",
                    "Breaststroke_Medley","Butterfly_Freestyle",
                   "Butterfly_Medley","Freestyle_Medley")
```

```
#Filtering data for pairwise t-test
Backstroke_Breaststroke  <- swimming_data  %>% filter(Category %in% c("Backstroke","Breaststroke"))
Backstroke_Butterfly  <- swimming_data  %>% filter(Category %in% c("Backstroke","Butterfly"))
Backstroke_Freestyle  <- swimming_data  %>% filter(Category %in% c("Backstroke","Freestyle"))
Backstroke_Medley  <- swimming_data  %>% filter(Category %in% c("Backstroke","Medley"))
Breaststroke_Butterfly  <- swimming_data  %>% filter(Category %in% c("Breaststroke","Butterfly"))
Breaststroke_Freestyle  <- swimming_data  %>% filter(Category %in% c("Breaststroke","Freestyle"))
Breaststroke_Medley  <- swimming_data  %>% filter(Category %in% c("Breaststroke","Medley"))
Butterfly_Freestyle  <- swimming_data  %>% filter(Category %in% c("Butterfly","Freestyle"))
Butterfly_Medley  <- swimming_data  %>% filter(Category %in% c("Butterfly","Medley"))
Freestyle_Medley  <- swimming_data  %>% filter(Category %in% c("Freestyle","Medley"))
```

```
#t-tests
test_1  <- t.test(Time ~ Category, data = Backstroke_Breaststroke, var.equal = TRUE)
test_2  <- t.test(Time ~ Category, data = Backstroke_Butterfly, var.equal = TRUE)
test_3  <- t.test(Time ~ Category, data = Backstroke_Freestyle, var.equal = TRUE)
test_4  <- t.test(Time ~ Category, data = Backstroke_Medley, var.equal = TRUE)
test_5  <- t.test(Time ~ Category, data = Breaststroke_Butterfly, var.equal = TRUE)
test_6  <- t.test(Time ~ Category, data = Breaststroke_Freestyle, var.equal = TRUE)
test_7  <- t.test(Time ~ Category, data = Breaststroke_Medley, var.equal = TRUE)
test_8  <- t.test(Time ~ Category, data = Butterfly_Freestyle, var.equal = TRUE)
test_9  <- t.test(Time ~ Category, data = Butterfly_Medley, var.equal = TRUE)
test_10 <- t.test(Time ~ Category, data = Freestyle_Medley, var.equal = TRUE)
```

```
#p-values from the t-tests
p_values  <- c(test_1$p.value,test_2$p.value,test_3$p.value,test_4$p.value,test_5$p.value,
               test_6$p.value,test_7$p.value,test_8$p.value,test_9$p.value,test_10$p.value)
p_values
```

```
##  [1] 5.387969e-19 7.492059e-01 3.900106e-17 2.220449e-04 2.630470e-16
##  [6] 1.675051e-27 5.253854e-19 5.578581e-15 4.793850e-03 1.275451e-21
```

```
#Tabulating the P-value
df1 <- data.frame(data.frame(pair_category),data.frame(p_values))
names(df1)[1] <- "Categories pair"
names(df1)[2] <- "p-values"

df1["Reject Yes/No"] <- with(df1, ifelse(df1$`p-values` < 0.05, "Yes", "No"))
df1
```

```
##            Categories pair     p-values Reject Yes/No
## 1  Backstroke_Breaststroke 5.387969e-19           Yes
## 2     Backstroke_Butterfly 7.492059e-01            No
## 3     Backstroke_Freestyle 3.900106e-17           Yes
## 4        Backstroke_Medley 2.220449e-04           Yes
## 5   Breaststroke_Butterfly 2.630470e-16           Yes
## 6   Breaststroke_Freestyle 1.675051e-27           Yes
## 7      Breaststroke_Medley 5.253854e-19           Yes
## 8      Butterfly_Freestyle 5.578581e-15           Yes
## 9         Butterfly_Medley 4.793850e-03           Yes
## 10        Freestyle_Medley 1.275451e-21           Yes
```

Multiple Tests Adjustment Methods

```
#Adjusting methods, Bonferroni, Benjamini and hochberg
p_values_bonferroni  <- p.adjust(p = p_values, method = "bonferroni", n = 10)
p_values_holm  <- p.adjust(p = p_values, method = "holm", n = 10)
```

Bonferroni's Correction

```
#Tabulating the P-value after bonferroni correction method
df2 <- data.frame(data.frame(pair_category),data.frame(p_values_bonferroni))
names(df2)[1] <- "Categories pair"
names(df2)[2] <- "Adjusted p-values"

df2["Reject Yes/No"] <- with(df2, ifelse(df2$`Adjusted p-values` < 0.05, "Yes", "No"))

df2
```

```
##            Categories pair Adjusted p-values Reject Yes/No
## 1  Backstroke_Breaststroke      5.387969e-18           Yes
## 2     Backstroke_Butterfly      1.000000e+00            No
## 3     Backstroke_Freestyle      3.900106e-16           Yes
## 4        Backstroke_Medley      2.220449e-03           Yes
## 5   Breaststroke_Butterfly      2.630470e-15           Yes
## 6   Breaststroke_Freestyle      1.675051e-26           Yes
## 7      Breaststroke_Medley      5.253854e-18           Yes
## 8      Butterfly_Freestyle      5.578581e-14           Yes
## 9         Butterfly_Medley      4.793850e-02           Yes
## 10        Freestyle_Medley      1.275451e-20           Yes
```

Bonferroni-Holm method

```
#Tabulating the P-value before bonferroni-holm adjustment method
df3 <- data.frame(data.frame(pair_category),data.frame(p_values_holm))
names(df3)[1] <- "Categories pair"
names(df3)[2] <- "Adjusted p-values"

df3["Reject Yes/No"] <- with(df3, ifelse(df3$`Adjusted p-values` < 0.05, "Yes", "No"))

df3
```

```
##            Categories pair Adjusted p-values Reject Yes/No
## 1  Backstroke_Breaststroke      4.203083e-18           Yes
## 2     Backstroke_Butterfly      7.492059e-01            No
## 3     Backstroke_Freestyle      2.340064e-16           Yes
## 4        Backstroke_Medley      6.661346e-04           Yes
## 5   Breaststroke_Butterfly      1.315235e-15           Yes
## 6   Breaststroke_Freestyle      1.675051e-26           Yes
## 7      Breaststroke_Medley      4.203083e-18           Yes
## 8      Butterfly_Freestyle      2.231432e-14           Yes
## 9         Butterfly_Medley      9.587700e-03           Yes
## 10        Freestyle_Medley      1.147906e-20           Yes
```