Project II Comparison of multiple distributions

Bushra Tariq Kiyani

2023-05-17

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

```
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
```

```
library(rstatix)
```

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

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

```
babies_data <- read.csv("babies.csv")
# Selecting only the "id", wt" and "smoke" columns from the dataset
babies_data <- babies_data[, c("id", "wt", "smoke")]
head(babies_data, 5)
```

```
##   id  wt smoke
## 1 15 120     0
## 2 20 113     0
## 3 58 128     1
## 4 61 123     3
## 5 72 108     1
```

```
# Check the data types of columns in the dataset
str(babies_data)
```

```
## 'data.frame':    1236 obs. of  3 variables:
##  $ id   : int  15 20 58 61 72 100 102 129 142 148 ...
##  $ wt   : int  120 113 128 123 108 136 138 132 120 143 ...
##  $ smoke: int  0 0 1 3 1 2 0 0 0 1 ...
```

```
# check duplicate values based on the 'id' column
duplicates <- babies_data[duplicated(babies_data$id) | duplicated(babies_data$id, fromLast = TRUE),]
count <- length(unique(duplicates$id))
count
```

```
## [1] 10
```

```
sort(unique(duplicates$id))
```

```
##  [1] 1091 2621 6360 6510 7045 7112 7441 8107 8253 8716
```

```
duplicates
```

```
##        id  wt smoke
## 65   1091 128     1
## 159  2621 105     0
## 216  8253  NA     0
## 377  1091  NA     1
## 507  6360  98     0
## 547  6510 123     2
## 567  2621  NA     0
## 634  6510  NA     2
## 660  8716  NA     2
## 730  7045 133     3
## 749  7112 127     0
## 865  7112  NA     0
## 872  7441 125     2
## 886  8107  NA     2
## 1098 7441  NA     2
## 1120 8107 117     2
## 1122 7045  NA     3
## 1159 8253 131     0
## 1163 6360  NA     0
## 1213 8716 138     2
```

```
# Filter the data to keep unique 'id' values where 'wt' is not NA
babies_data <- babies_data %>%
  group_by(id) %>%
  filter(!is.na(wt)) %>%
  distinct(id, .keep_all = TRUE)
```

```
filtered_rows <- subset(babies_data, smoke == 9)

# Display the results
filtered_rows
```

```
## # A tibble: 10 x 3
## # Groups:   id [10]
##       id    wt smoke
##    <int> <int> <int>
##  1  2722   126     9
##  2  3558    90     9
##  3  4396   130     9
##  4  4401   106     9
##  5  6114   115     9
##  6  6692   142     9
##  7  6787   151     9
##  8  6869   141     9
##  9  6876   158     9
## 10  7570   108     9
```

```
# Remove unknown values
babies_data_updated <- babies_data %>%
  filter(smoke != 9)
```

```
# Ordering the data based on categories in smoke
babies_data_updated <- babies_data_updated[order(babies_data_updated$smoke),]
# Checking total number of observations in each group the data based on categories
babies_data_updated %>% group_by(smoke) %>%tally()
```

```
## # A tibble: 4 x 2
##   smoke     n
##   <int> <int>
## 1     0   540
## 2     1   481
## 3     2    95
## 4     3   100
```

```
# Count the number of NA values in the 'wt' column
sum(is.na(babies_data_updated$wt))
```

```
## [1] 0
```

```
any(babies_data_updated$wt == 999)
```

```
## [1] FALSE
```

```
# Summary after removing duplicates
summary(babies_data_updated$wt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    55.0   109.0   120.0   119.6   131.0   176.0
```

```
#Standard deviation of weight overall
round(sd(babies_data_updated$wt),3)
```

```
## [1] 18.14
```

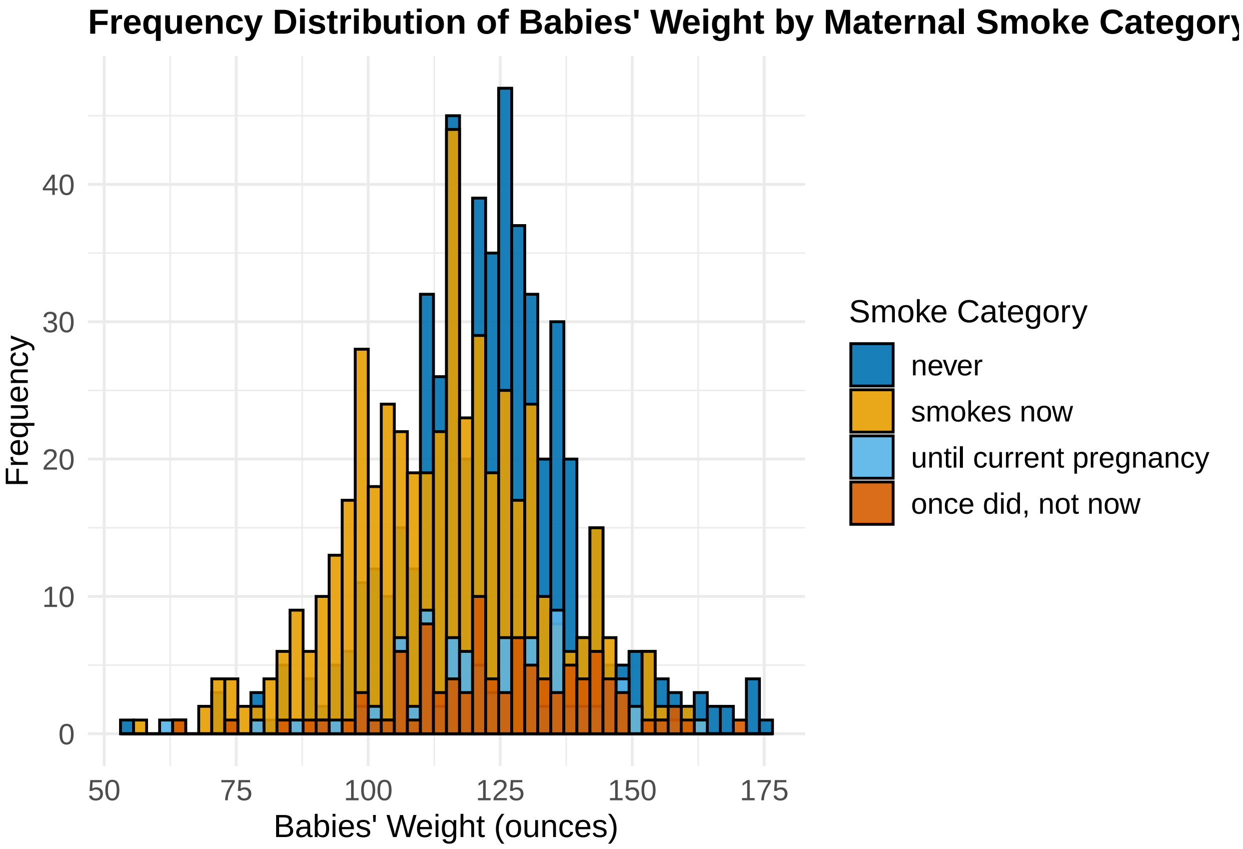
```
wt_summary <- babies_data_updated %>%
  group_by(smoke) %>%
  summarize(mean_wt = mean(wt),
            median_wt = median(wt),
            sd_wt = sd(wt),
            min_wt = min(wt),
            max_wt = max(wt),
            var_wt = var(wt))

# Display summary statistics for the 'wt' variable
wt_summary
```

```
## # A tibble: 4 x 7
##   smoke mean_wt median_wt sd_wt min_wt max_wt var_wt
##   <int>   <dbl>     <dbl> <dbl>  <int>  <int>  <dbl>
## 1     0    123.       124  17.1     55    176   291.
## 2     1    114.       115  18.0     58    163   323.
## 3     2    123.       122  17.8     62    163   317.
## 4     3    125.       124.  18.6     65    170   345.
```

```
# Create a histogram for the 'wt' variable, grouped by the 'smoke' variable
plot_freq <- ggplot(data = babies_data_updated, aes(x = wt, fill = as.factor(smoke))) +
  geom_histogram(col = "black", position = "identity", alpha = 0.9, bins = 50) +
  scale_fill_manual(values = c("#0072B2", "#E69F00", "#56B4E9", "#D55E00",
          "#CC79A7"),
                    name = "Smoke Category",
                    labels = c("never", "smokes now", "until current pregnancy", "once did, not now")) +
  labs(title = "Frequency Distribution of Babies' Weight by Maternal Smoke Category",
       x = "Babies' Weight (ounces)",
       y = "Frequency") +
  theme_minimal()

plot_freq2 <- plot_freq + theme(plot.title = element_text(face = "bold"),
                                 axis.text = element_text(size = 11),
               axis.title = element_text(size = 12),
               legend.text = element_text(size = 11),
               legend.title = element_text(size = 12))
ggsave("freq.pdf", plot = plot_freq2, width = 8.5, height = 4, units = "in")
plot_freq2
```



Frequency Distribution of Babies' Weight by Maternal Smoke Categor

```r
never_lt_55 <- babies_data_updated[babies_data_updated$smoke == 3 & babies_data_updated$wt > 165, ]
never_lt_55
```

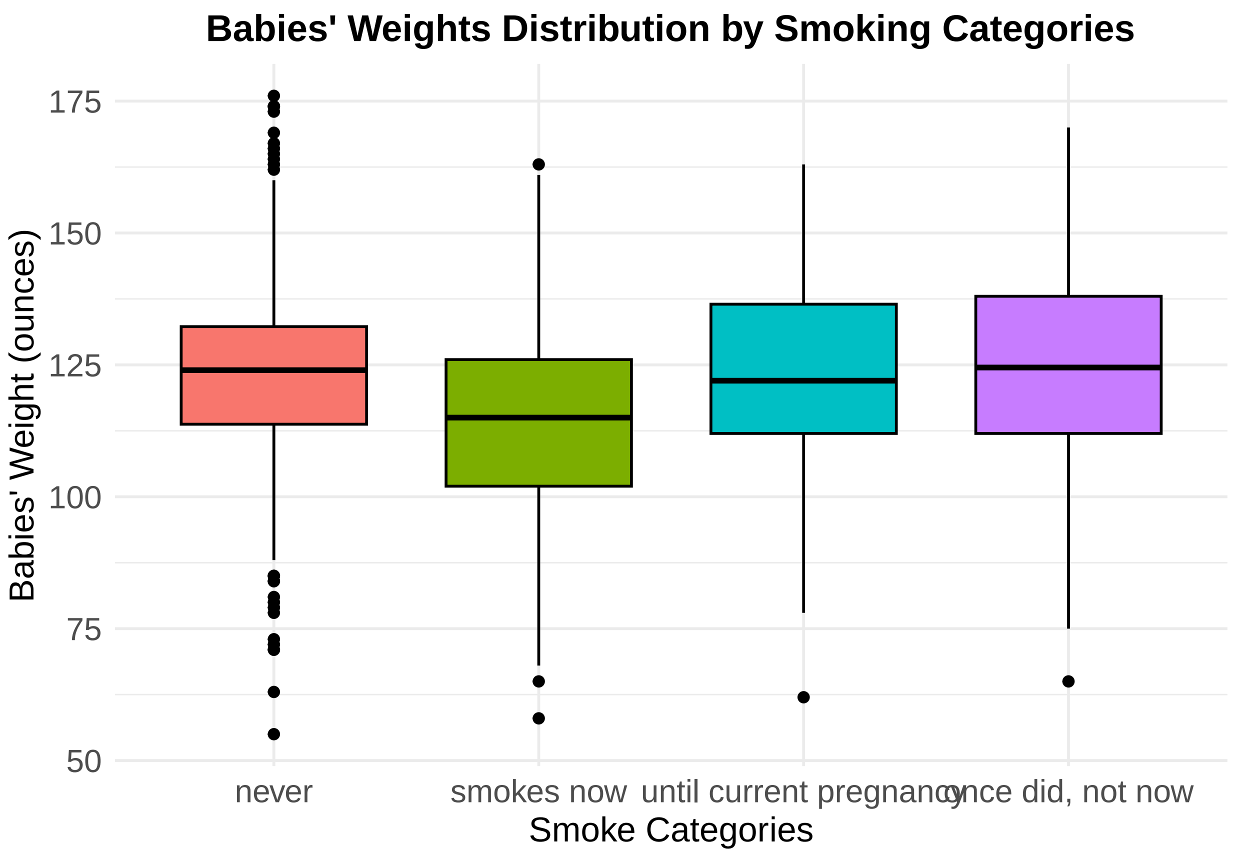
```
## # A tibble: 1 x 3
## # Groups:   id [1]
##      id    wt smoke
##   <int> <int> <int>
## 1  6660   170     3
```

Verifying the Assumptions

1. Homogeneity of variance assumption

```r
# Box plot to compare finishing time in the different categories to find the homogeneity in variance
box_plot <- ggboxplot(babies_data_updated, x = "smoke", y = "wt", color = "black", fill = "smoke",
                   ylab = "Babies' Weight (ounces)", xlab = "Smoking Status") +
  labs(title = "Babies' Weights Distribution by Smoking Categories") +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", hjust = 0.5, size = 14),
        axis.text = element_text(size = 12),
        axis.title = element_text(size = 13),
        legend.text = element_text(size = 12),
        legend.title = element_text(size = 13),
        legend.position = "none") +
  scale_x_discrete(name = "Smoke Categories",
                   labels = c("never", "smokes now", "until current pregnancy", "once did, not now"))

ggsave("box_plot2.pdf", plot = box_plot, width = 8.5, height = 4, units = "in")
box_plot
```



2. Normality assumption

```
Never <- babies_data_updated %>% filter(smoke == 0)
Smokes_now <- babies_data_updated %>% filter(smoke == 1)
Until_cur_pregnancy <- babies_data_updated %>% filter(smoke == 2)
Once_did <- babies_data_updated %>% filter(smoke == 3)
```

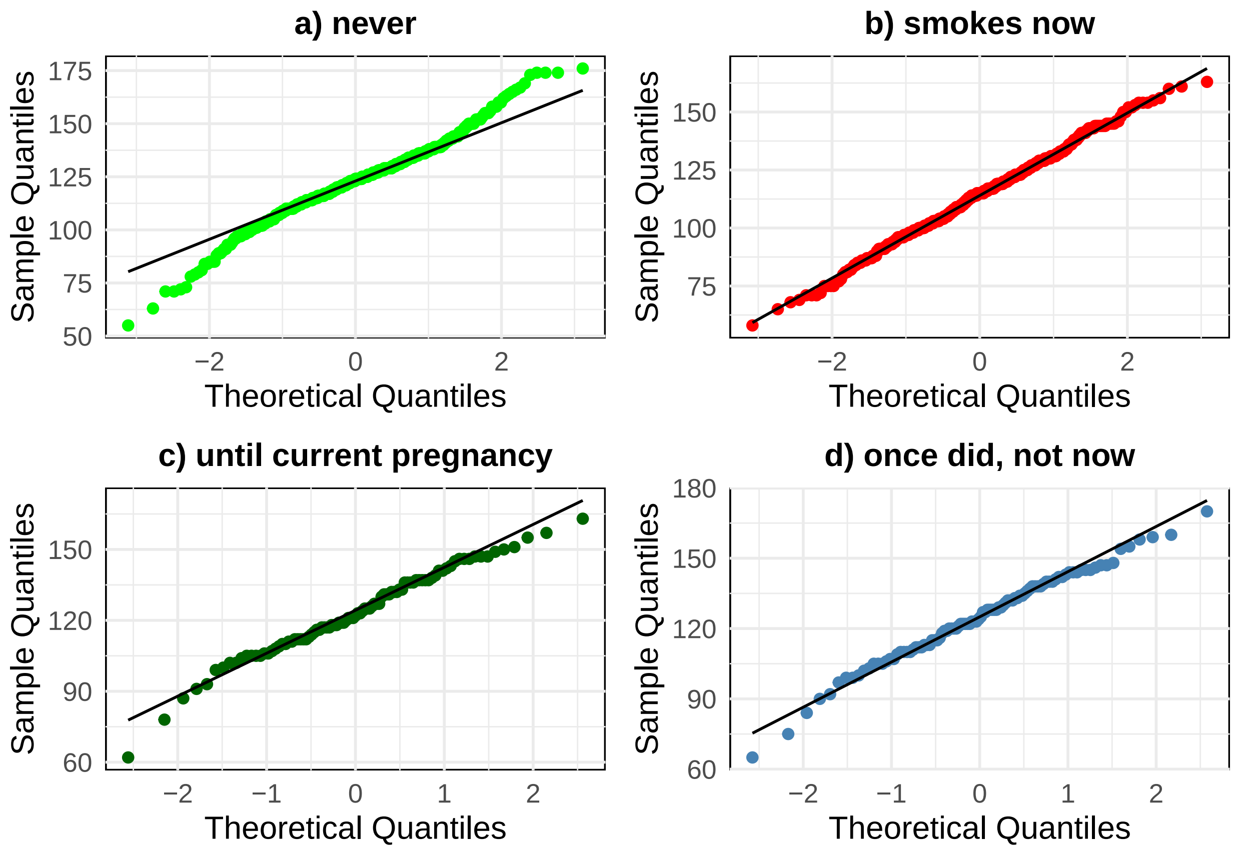
```
plot4 <- ggplot(Never) + stat_qq(aes(sample = wt), color= "green")+
  stat_qq_line(aes(sample = wt)) +  scale_x_continuous(name = "Theoretical Quantiles") +
  scale_y_continuous(name = "Sample Quantiles") + ggtitle("a) never") + theme_minimal()
  plot4 <- plot4 + theme(panel.background = element_rect(fill = "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(Smokes_now) + stat_qq(aes(sample = wt), color= "red")+
  stat_qq_line(aes(sample = wt)) +  scale_x_continuous(name = "Theoretical Quantiles") +
  scale_y_continuous(name = "Sample Quantiles") + ggtitle("b) smokes now")  + theme_minimal()
  plot5 <- plot5 + 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(Until_cur_pregnancy) + stat_qq(aes(sample = wt), color= "darkgreen")+
  stat_qq_line(aes(sample = wt)) +  scale_x_continuous(name = "Theoretical Quantiles") +
  scale_y_continuous(name = "Sample Quantiles") + ggtitle("c) until current pregnancy") + theme_minimal
  plot6 <- plot6 +  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(Once_did) + stat_qq(aes(sample = wt), color= "steelblue")+
  stat_qq_line(aes(sample = wt)) +  scale_x_continuous(name = "Theoretical Quantiles") +
  scale_y_continuous(name = "Sample Quantiles") + ggtitle("d) once did, not now") + theme_minimal()
  plot7 <- plot7 +  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, ncol=2, nrow = 2)
```



```
ggsave("QQplots.pdf",plot = final_plot1, width = 8.5, height = 4.5, units = "in")
final_plot1
```

```
## TableGrob (2 x 2) "arrange": 4 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]
```

Task 2

Do the babies birth weights differ between the categories? Conduct a global test.

```
# Convert 'wt' to numeric
babies_data_updated$wt <- as.numeric(babies_data_updated$wt)

# Convert 'smoke' to factor
babies_data_updated$smoke <- as.factor(babies_data_updated$smoke)
```

```
# Perform one-way ANOVA test
anova_result <- aov(wt ~ smoke, data = babies_data_updated)
summary(anova_result)
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## smoke          3  23932    7977   25.72 3.91e-16 ***
## Residuals   1212 375862     310
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Extract p-value from ANOVA result
anova_p_value <- summary(anova_result)[[1]]["smoke", "Pr(>F)"]

# Check if the p-value is less than the significance level (0.05)
if (anova_p_value < 0.05) {
  cat("There is a significant difference in babies' birth weights between the smoking categories.\n")
} else {
  cat("There is no significant difference in babies' birth weights between the smoking categories.\n")
}
```

```
## There is a significant difference in babies' birth weights between the smoking categories.
```

Task 3

Multiple T-Tests

```
#0=never", "1=smokes now", "2=until current pregnancy", "3=once did, not now", "9=unknown"
#List of pairs made of the 5 Categories
pair_category  <- c("never_smokesNow","never_untilCurrentPregnancy","never_onceDidNotNow",
                    "smokesNow_untilCurrentPregnancy","smokesNow_onceDidNotNow",
                    "untilCurrentPregnancy_onceDidNotNow")
```

```
#Filtering data for pairwise t-test
never_smokesNow  <- babies_data_updated  %>% filter(smoke %in% c(0,1))
never_untilCurrentPregnancy  <- babies_data_updated  %>% filter(smoke %in% c(0,2))
never_onceDidNotNow  <- babies_data_updated  %>% filter(smoke %in% c(0,2))
smokesNow_untilCurrentPregnancy  <- babies_data_updated  %>% filter(smoke %in% c(1,2))
smokesNow_onceDidNotNow <- babies_data_updated  %>% filter(smoke %in% c(1,3))
untilCurrentPregnancy_onceDidNotNow  <- babies_data_updated  %>% filter(smoke %in% c(2,3))
```

```
#t-tests
test_1  <- t.test(wt ~ factor(smoke), data = never_smokesNow, var.equal = TRUE)
test_2  <- t.test(wt ~ factor(smoke), data = never_untilCurrentPregnancy, var.equal = TRUE)
```

```r
test_3  <- t.test(wt ~ factor(smoke), data = never_onceDidNotNow, var.equal = TRUE)
test_4  <- t.test(wt ~ factor(smoke), data = smokesNow_untilCurrentPregnancy, var.equal = TRUE)
test_5  <- t.test(wt ~ factor(smoke), data = smokesNow_onceDidNotNow, var.equal = TRUE)
test_6  <- t.test(wt ~ factor(smoke), data = untilCurrentPregnancy_onceDidNotNow, var.equal = TRUE)
```

```r
#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)
p_values
```

```
## [1] 3.935037e-15 9.070791e-01 9.070791e-01 1.008024e-05 1.681502e-07
## [6] 5.540177e-01
```

```r
#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                       never_smokesNow 3.935037e-15           Yes
## 2         never_untilCurrentPregnancy 9.070791e-01            No
## 3                 never_onceDidNotNow 9.070791e-01            No
## 4     smokesNow_untilCurrentPregnancy 1.008024e-05           Yes
## 5             smokesNow_onceDidNotNow 1.681502e-07           Yes
## 6 untilCurrentPregnancy_onceDidNotNow 5.540177e-01            No
```

Multiple Tests Adjustment Method: Bonferroni's Correction

```r
#Adjusting method: Bonferroni
p_values_bonferroni <- p.adjust(p = p_values, method = "bonferroni", n = 6)
```

```r
#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                     never_smokesNow      2.361022e-14           Yes
## 2         never_untilCurrentPregnancy      1.000000e+00            No
## 3                 never_onceDidNotNow      1.000000e+00            No
## 4     smokesNow_untilCurrentPregnancy      6.048143e-05           Yes
## 5             smokesNow_onceDidNotNow      1.008901e-06           Yes
## 6 untilCurrentPregnancy_onceDidNotNow      1.000000e+00            No
```

Tukey-Kramer test

TukeyHSD() function adjust for unequal sizes as well (Performs Tukey-Kramer when sizes are unequal across the categories)

```r
# Perform Tukey-Kramer test
tukey_hsd <- TukeyHSD(anova_result)
tukey_hsd
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = wt ~ smoke, data = babies_data_updated)
##
## $smoke
##           diff        lwr       upr     p adj
## 1-0 -8.7530030 -11.593395 -5.912611 0.0000000
## 2-0  0.2230994  -4.817277  5.263476 0.9994719
## 3-0  1.7688889  -3.163166  6.700943 0.7927303
## 2-1  8.9761024   3.889689 14.062516 0.0000366
## 3-1 10.5218919   5.542798 15.500986 0.0000004
## 3-2  1.5457895  -4.944892  8.036471 0.9280679
```

```r
# Extract p-values and convert them into a data frame
upr <- tukey_hsd[[1]][, "upr"]
lwr <- tukey_hsd[[1]][, "lwr"]
diff <- tukey_hsd[[1]][, "diff"]
confi <- data.frame(
  Categories_pair = pair_category,
  diff = diff,
  lwr = lwr,
  upr = upr
)

confi
```

```
##                          Categories_pair       diff        lwr       upr
## 1-0                      never_smokesNow -8.7530030 -11.593395 -5.912611
## 2-0          never_untilCurrentPregnancy  0.2230994  -4.817277  5.263476
## 3-0                  never_onceDidNotNow  1.7688889  -3.163166  6.700943
## 2-1      smokesNow_untilCurrentPregnancy  8.9761024   3.889689 14.062516
## 3-1              smokesNow_onceDidNotNow 10.5218919   5.542798 15.500986
## 3-2 untilCurrentPregnancy_onceDidNotNow  1.5457895  -4.944892  8.036471
```

```r
# Extract p-values and convert them into a data frame
p_val_t <- tukey_hsd[[1]][, "p adj"]
p_val_t_df <- data.frame(
  Categories_pair = pair_category,
  adj_p_value = p_val_t
)
# Add a column to indicate rejection or acceptance of the null hypothesis
p_val_t_df["Reject Yes/No"] <- with(p_val_t_df, ifelse(p_val_t_df$adj_p_value < 0.05, "Yes", "No"))

p_val_t_df
```

```
##                          Categories_pair  adj_p_value Reject Yes/No
## 1-0                      never_smokesNow 0.000000e+00           Yes
## 2-0          never_untilCurrentPregnancy 9.994719e-01            No
## 3-0                  never_onceDidNotNow 7.927303e-01            No
## 2-1      smokesNow_untilCurrentPregnancy 3.661824e-05           Yes
## 3-1              smokesNow_onceDidNotNow 3.928815e-07           Yes
## 3-2 untilCurrentPregnancy_onceDidNotNow 9.280679e-01            No
```