Project_2

Rahul

2024-04-24

#Loadind and setups

```r
library(haven)
library(ggplot2)
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
```

```r
dataset <- read_sav("Sandwich.sav")

# Renaming the covariates
names(dataset) <- c("Ants", "Bread", "Topping", "Butter")

dataset$Bread <- as.factor(dataset$Bread)
levels(dataset$Bread) <- c("wholegrain", "multigrain", "rye", "white bread")

dataset$Topping <- as.factor(dataset$Topping)
levels(dataset$Topping) <- c("ham pickles", "peanut butter", "yeast spread")

dataset$Butter <- as.factor(dataset$Butter)
levels(dataset$Butter) <- c ("with", "without")

names(dataset)
```

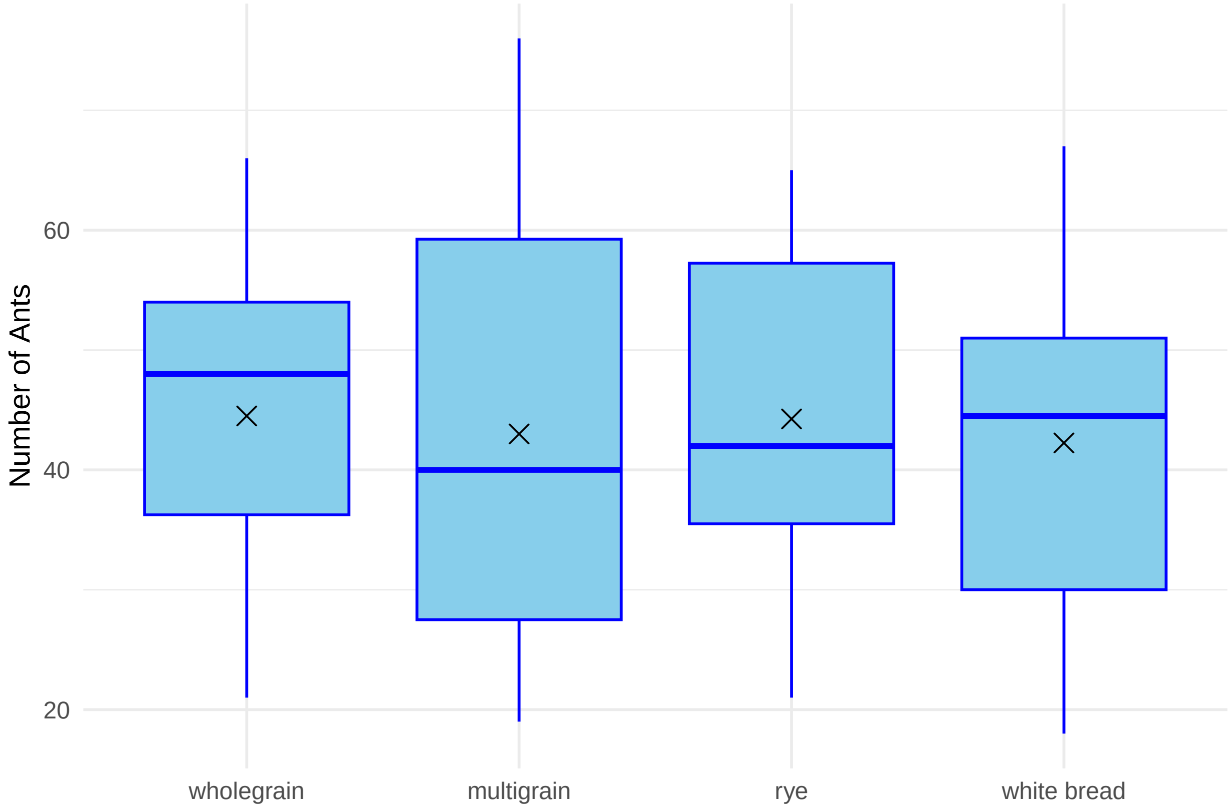
```
## [1] "Ants"    "Bread"   "Topping" "Butter"
```

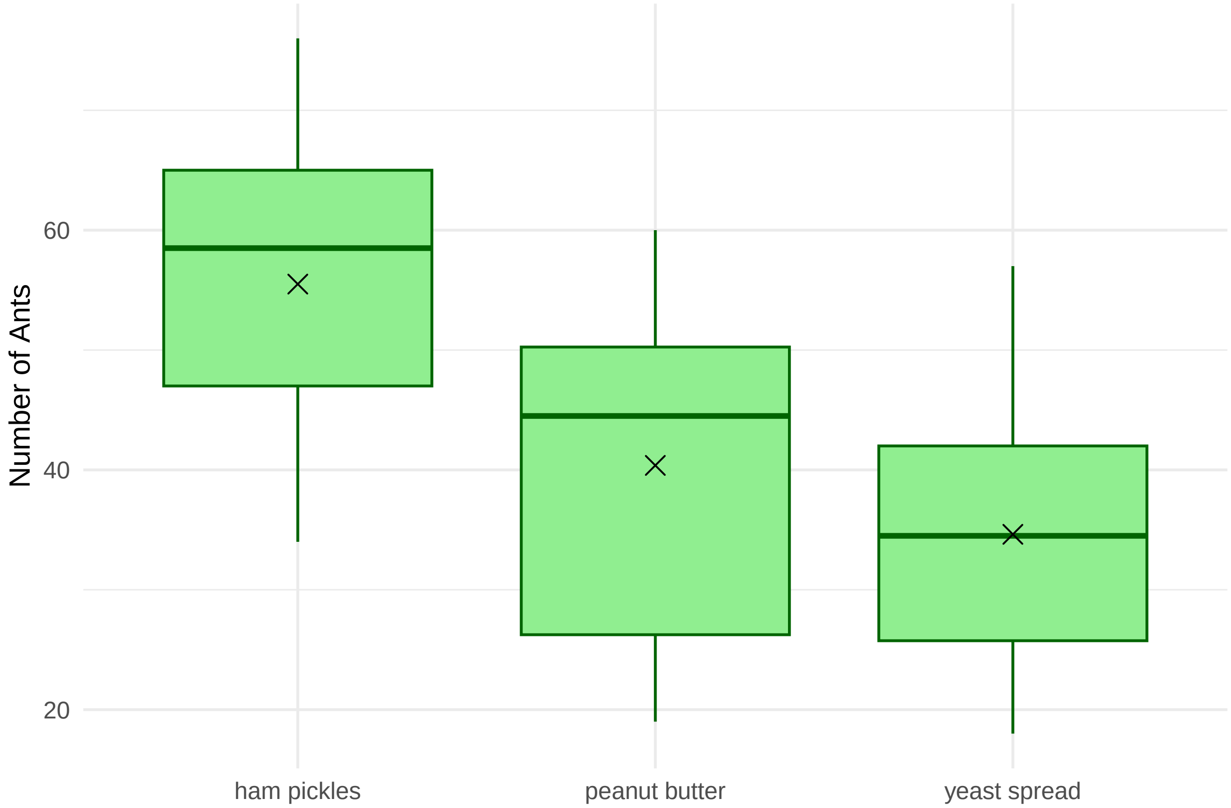
```r
summary(dataset)
```

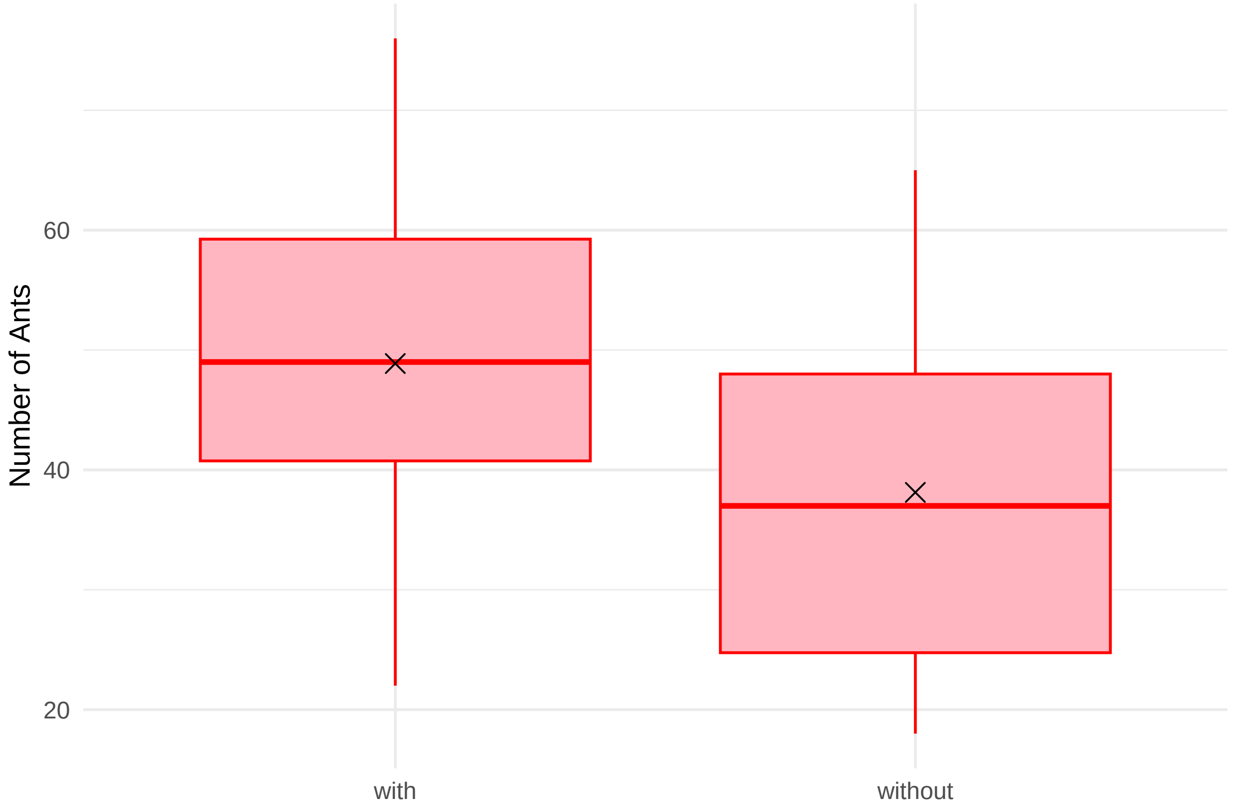
```
##       Ants              Bread              Topping       Butter
##  Min.   :18.0   wholegrain :12   ham pickles   :16   with   :24
##  1st Qu.:30.5   multigrain :12   peanut butter:16   without:24
##  Median :43.0   rye        :12   yeast spread :16
##  Mean   :43.5   white bread:12
##  3rd Qu.:57.0
##  Max.   :76.0
```

#Box Plots for each factor

```r
# Bread
gg.Bread <- ggplot(dataset, aes(x = Bread, y = Ants)) +
  geom_boxplot(fill = "skyblue", color = "blue") +
  stat_summary(fun = mean, geom = "point", shape = 4, size = 3,
               color = "black") +  # Add mean point
  labs(x = "", y = "Number of Ants"
       # , title = "Box Plot of Ants by Bread type"
       ) +
  theme_minimal()



# Toppings
gg.Topping <- ggplot(dataset, aes(x = Topping, y = Ants)) +
  geom_boxplot(fill = "lightgreen", color = "darkgreen") +
  stat_summary(fun = mean, geom = "point", shape = 4, size = 3,
               color = "black") +  # Add mean point
  labs(x = "", y = "Number of Ants"
       # , title = "Box Plot of Ants by Topping type"
       ) +
  theme_minimal()



# Butter
gg.Butter <- ggplot(dataset, aes(x = Butter, y = Ants)) +
  geom_boxplot(fill = "lightpink", color = "red") +
  stat_summary(fun = mean, geom = "point", shape = 4, size = 3,
               color = "black") +  # Add mean point
  labs(x = "", y = "Number of Ants"
       # , title = "Box Plot of Ants by with or without Butter"
       ) +
  theme_minimal()



gg.Bread; gg.Topping; gg.Butter
```


Number of Ants
60
40
20
wholegrain
multigrain
rye
white bread


Number of Ants
60
40
20
ham pickles
peanut butter
yeast spread



Test Assumptions

1 Equal variance among the groups. 2 The data is normally distributed 3 The data is independent.

#Statistical Summary for each factor

```
#Bread : 4 levels
summary.Brot <- dataset %>% group_by(Bread) %>%
  summarise(
    count = n(),
    min = min(Ants, na.rm = TRUE),
    # Q1  = quantile(Ants,0.25),
    median = median(Ants, na.rm = TRUE),
    # Q3  = quantile(Ants,0.75),
    max  = max(Ants, na.rm = TRUE),
    mean = mean(Ants, na.rm = TRUE),
    sd = sd(Ants, na.rm = TRUE)
    # ,
    # IQR = quantile(Ants,0.75) - quantile(Ants,0.25)
  )
print(summary.Brot)
```

```
## # A tibble: 4 x 7
##   Bread       count   min median   max  mean    sd
##   <fct>       <int> <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 wholegrain     12    21     48    66  44.5  14.6
## 2 multigrain     12    19   40      76  43    18.2
## 3 rye            12    21   42      65  44.2  13.4
## 4 white bread    12    18   44.5    67  42.2  15.9
```

```r
summary.Brot$sd
```

```
## [1] 14.60697 18.23084 13.39691 15.86377
```

```r
summary.Brot$sd[-2][3] - summary.Brot$sd[-2][2]
```

```
## [1] 2.466854
```

```r
#Topping : 3 levels
summary.Topping <- dataset %>% group_by(Topping) %>%
  summarise(
    count = n(),
    min = min(Ants, na.rm = TRUE),
    # Q1  = quantile(Ants,0.25),
    median = median(Ants, na.rm = TRUE),
    # Q3  = quantile(Ants,0.75),
    max  = max(Ants, na.rm = TRUE),
    mean = mean(Ants, na.rm = TRUE),
    sd = sd(Ants, na.rm = TRUE)
    # ,
    # IQR = quantile(Ants,0.75) - quantile(Ants,0.25)
  )
print(summary.Topping)
```

```
## # A tibble: 3 x 7
##   Topping        count   min median   max  mean    sd
##   <fct>          <int> <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 ham pickles       16    34   58.5    76  55.5  12.1
## 2 peanut butter     16    19   44.5    60  40.4  14.2
## 3 yeast spread      16    18   34.5    57  34.6  11.2
```

```r
summary.Topping$sd
```

```
## [1] 12.05543 14.18391 11.15870
```

```r
max(summary.Topping$sd) - min(summary.Topping$sd)
```

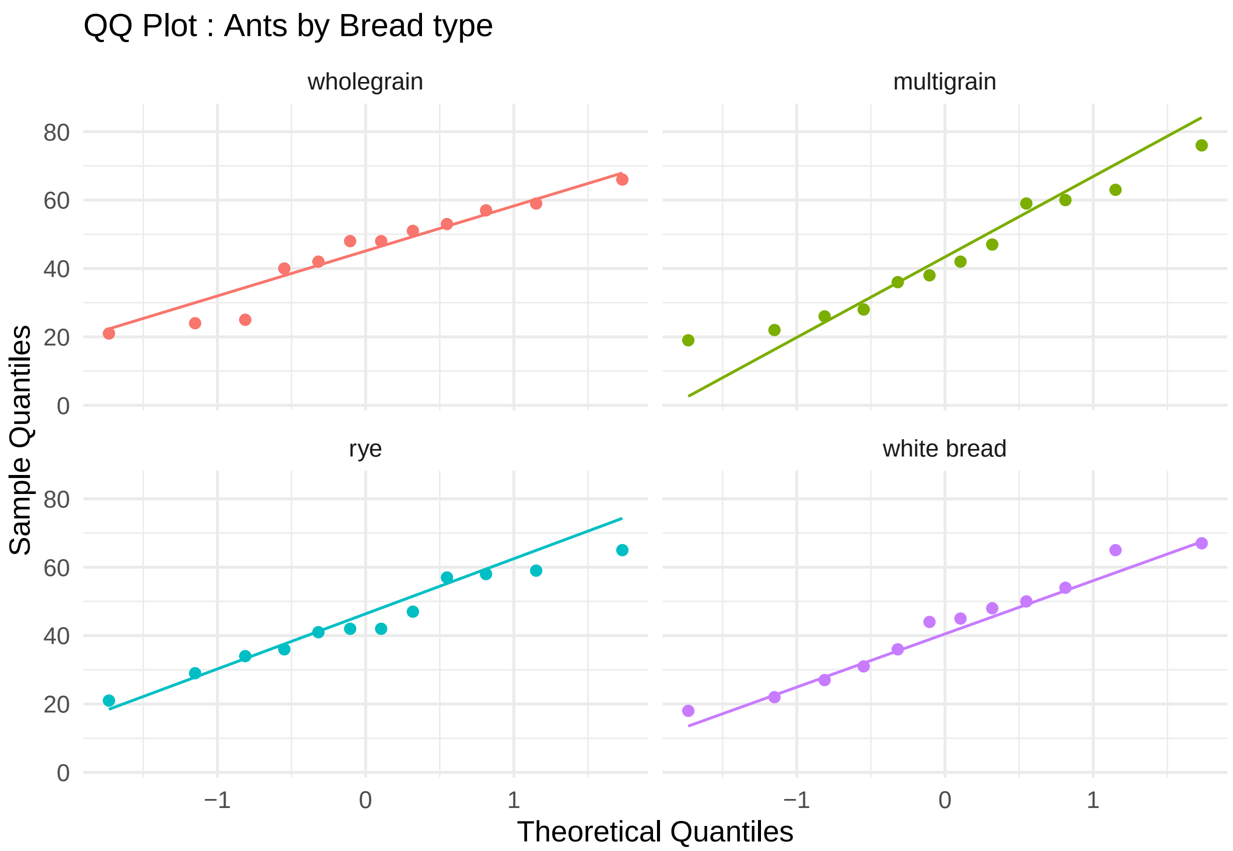
```
## [1] 3.025207
```

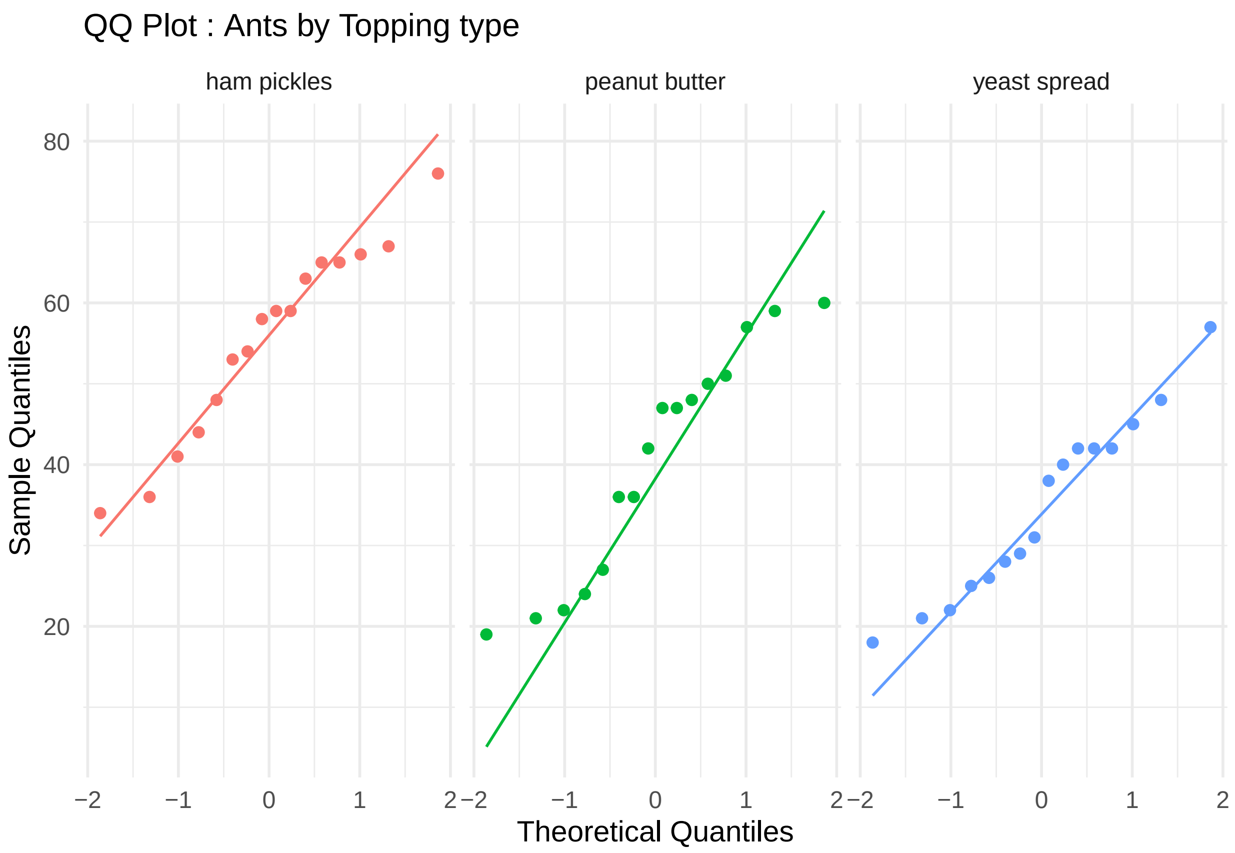
```r
#Butter : 2 levels
summary.Butter <- dataset %>% group_by(Butter) %>%
  summarise(
    count = n(),
    min = min(Ants, na.rm = TRUE),
    # Q1  = quantile(Ants,0.25),
    median = median(Ants, na.rm = TRUE),
    # Q3  = quantile(Ants,0.75),
    max  = max(Ants, na.rm = TRUE),
    mean = mean(Ants, na.rm = TRUE),
    sd = sd(Ants, na.rm = TRUE)
    # ,
    # IQR = quantile(Ants,0.75) - quantile(Ants,0.25)
  )
print(summary.Butter)
```

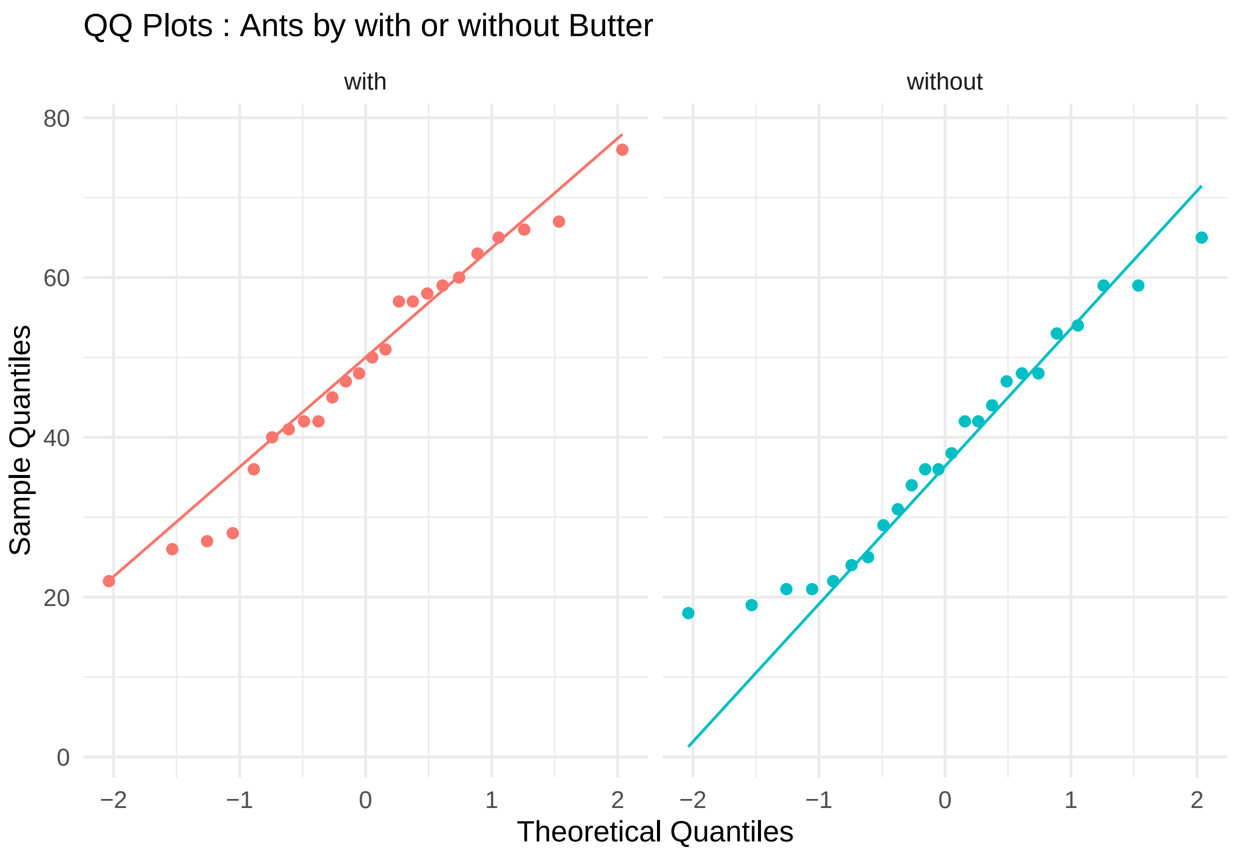
```
## # A tibble: 2 x 7
##   Butter  count   min median   max  mean    sd
##   <fct>   <int> <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 with       24    22     49    76  48.9  14.5
## 2 without    24    18     37    65  38.1  14.1
```

QQ plot for normality test for each factor and their levels

```
# Brot(Bread) QQ
qq_plot_Bread <- ggplot(dataset, aes(sample = Ants, color = Bread)) +
  stat_qq() +
  stat_qq_line() +
  labs(title = "QQ Plot : Ants by Bread type") +
  facet_wrap(~ Bread) +  # Create separate plots for each category of Bread
  theme_minimal() +  # Set theme to minimal
  theme(plot.title = element_text(size = 12), legend.position = "none") + # Decrease title font size
  xlab("Theoretical Quantiles") +  # Add x-axis label
  ylab("Sample Quantiles")




# Belag(Topping) QQ
qq_plot_Topping <- ggplot(dataset, aes(sample = Ants, color = Topping)) +
  stat_qq() +
  stat_qq_line() +
  labs(title = "QQ Plot : Ants by Topping type") +
  facet_wrap(~ Topping) +  # Create separate plots for each category of Belag
  theme_minimal() +  # Set theme to minimal
  theme(plot.title = element_text(size = 12), legend.position = "none")  + # Decrease title font size
  xlab("Theoretical Quantiles") +  # Add x-axis label
  ylab("Sample Quantiles")




# Butter QQ
qq_plot_Butter <- ggplot(dataset, aes(sample = Ants, color = Butter)) +
  stat_qq() +
  stat_qq_line() +
  labs(title = "QQ Plots : Ants by with or without Butter") +
  facet_wrap(~ Butter) +  # Create separate plots for each category of Butter
  theme_minimal() +  # Set theme to minimal
  theme(plot.title = element_text(size = 12), legend.position = "none") + # Decrease title font size
  xlab("Theoretical Quantiles") +  # Add x-axis label
  ylab("Sample Quantiles")


qq_plot_Bread; qq_plot_Topping; qq_plot_Butter
```


QQ Plot : Ants by Bread type
wholegrain
multigrain
rye
white bread
Sample Quantiles
Theoretical Quantiles
0
20
40
60
80
−1
0
1


QQ Plot : Ants by Topping type
ham pickles
peanut butter
yeast spread
Sample Quantiles
Theoretical Quantiles
80
60
40
20
−2
−1
0
1
2



#ANOVA compares the mean in different group are equal or not # ie. We will test for each factor with different levels

```
# Step 1: Load necessary libraries
library(stats)     # For statistical functions

#ANOVA - Bread
cat("Bread Group \n")
```

```
## Bread Group
```

```
Bread_anova <- aov(Ants ~ Bread, data = dataset)
summary(Bread_anova)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Bread        3     41    13.5   0.055  0.983
## Residuals   44  10745   244.2
```

```
summary_Bread_anova <- summary(Bread_anova)
summary_Bread_anova[[1]]$`Pr(>F)`[1]
```

```
## [1] 0.9826739
```

```
#ANOVA - Topping
cat("\n Topping Group \n")
```

```
##
##  Topping Group
```

```
Topping_anova <- aov(Ants ~ Topping, data = dataset)
summary(Topping_anova)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## Topping      2   3721    1860   11.85 7.35e-05 ***
## Residuals   45   7065     157                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Summarize ANOVA results
summary_Topping_anova <- summary(Topping_anova)

# Extract p-value
summary_Topping_anova[[1]]$`Pr(>F)`[1]
```

```
## [1] 7.351161e-05
```

T-test for each factor with different levels

ie. pair-wise t test among different levels

```
# Pairwise T-test for Topping ===================================

# To store p-values for each combination of levels of Topping
unadjusted.p.values <- c()


## (1) "ham pickles"   &   (2) "peanut butter"
t.Topping.H01 <- t.test(dataset$Ants[dataset$Topping == "ham pickles"],
       dataset$Ants[dataset$Topping == "peanut butter"],
       alternative = "two.sided", paired = FALSE, var.equal = TRUE)

unadjusted.p.values[1] <- t.Topping.H01$p.value



## (1) "ham pickles"   &   (3) "yeast spread"
t.Topping.H02 <- t.test(dataset$Ants[dataset$Topping == "ham pickles"],
       dataset$Ants[dataset$Topping == "yeast spread"],
       alternative = "two.sided", paired = FALSE, var.equal = TRUE)

unadjusted.p.values[2] <- t.Topping.H02$p.value



## (2) "peanut butter"   &   (3) "yeast spread"
t.Topping.H03 <- t.test(dataset$Ants[dataset$Topping == "peanut butter"],
       dataset$Ants[dataset$Topping == "yeast spread"],
       alternative = "two.sided", paired = FALSE, var.equal = TRUE)

unadjusted.p.values[3] <- t.Topping.H03$p.value


# adjusting p-values with "bonferroni" method
```

```
adjusted.p.values <- p.adjust(unadjusted.p.values, method = "bonferroni")
adjusted.p.values
```

```
## [1] 8.536451e-03 5.527728e-05 6.368751e-01
```

```
## Which is nothing but ....
unadjusted.p.values * 3
```

```
## [1] 8.536451e-03 5.527728e-05 6.368751e-01
```

```
# T-test for Butter ===================================
# For Butter : No adjustment needed -----------------------------
t.Butter.H01 <- t.test(dataset$Ants[dataset$Butter == "with"],
       dataset$Ants[dataset$Butter == "without"],
       alternative = "two.sided", paired = FALSE, var.equal = TRUE)

t.Butter.H01$p.value
```

```
## [1] 0.01232967
```