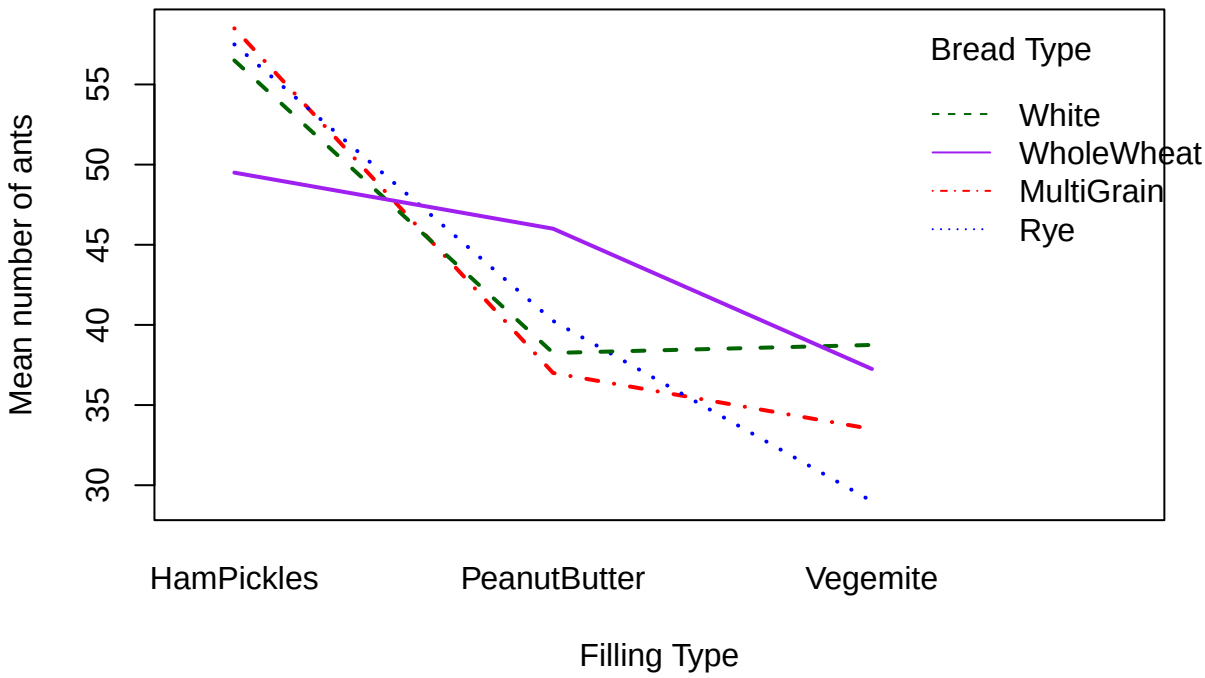
HW#3 part 2

```r
library(readxl)
SandwichAnts <- read_excel("SandwichAnts.xlsx")

SandwichAnts$Bread   <- factor(SandwichAnts$Bread)
SandwichAnts$Filling <- factor(SandwichAnts$Filling)

model <- aov(Ants ~ Bread * Filling, data = SandwichAnts)
summary(model)
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## Bread          3     40    13.5   0.075 0.972862
## Filling        2   3721  1860.3  10.386 0.000275 ***
## Bread:Filling  6    577    96.2   0.537 0.776545
## Residuals     36   6448   179.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
interaction.plot(SandwichAnts$Filling, SandwichAnts$Bread, SandwichAnts$Ants,
                 xlab = "Filling Type",
                 ylab = "Mean number of ants",
                 trace.label = "Bread Type",
                 col = c("red", "blue", "darkgreen", "purple"),
                 lwd = 2)
```




```
library(readr)
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
```

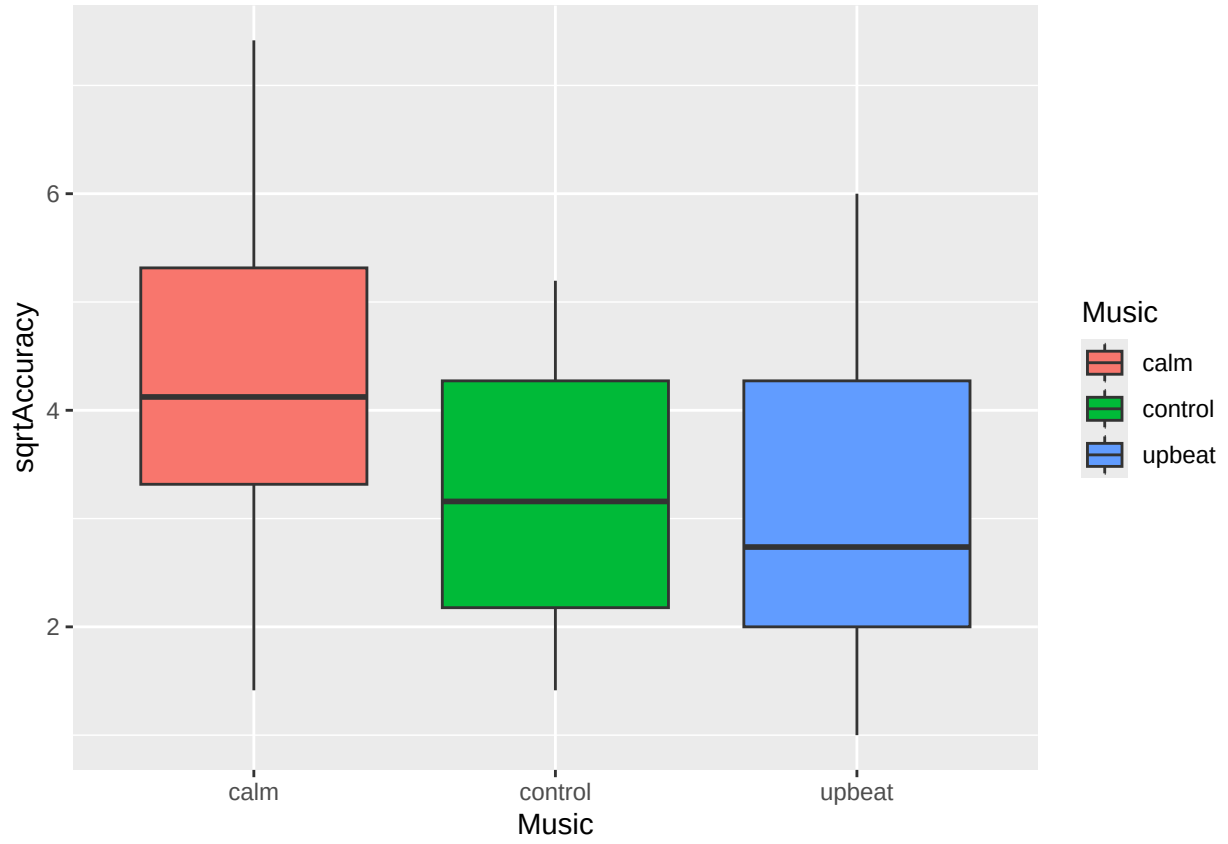
```
library(ggplot2)
library(ez)

MusicTime <- read_csv("MusicTme.csv")
```

```
## Rows: 60 Columns: 6
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): MusicBg, Subject, Sex, Music
```

```
## dbl (2): TimeGuess, Accuracy
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

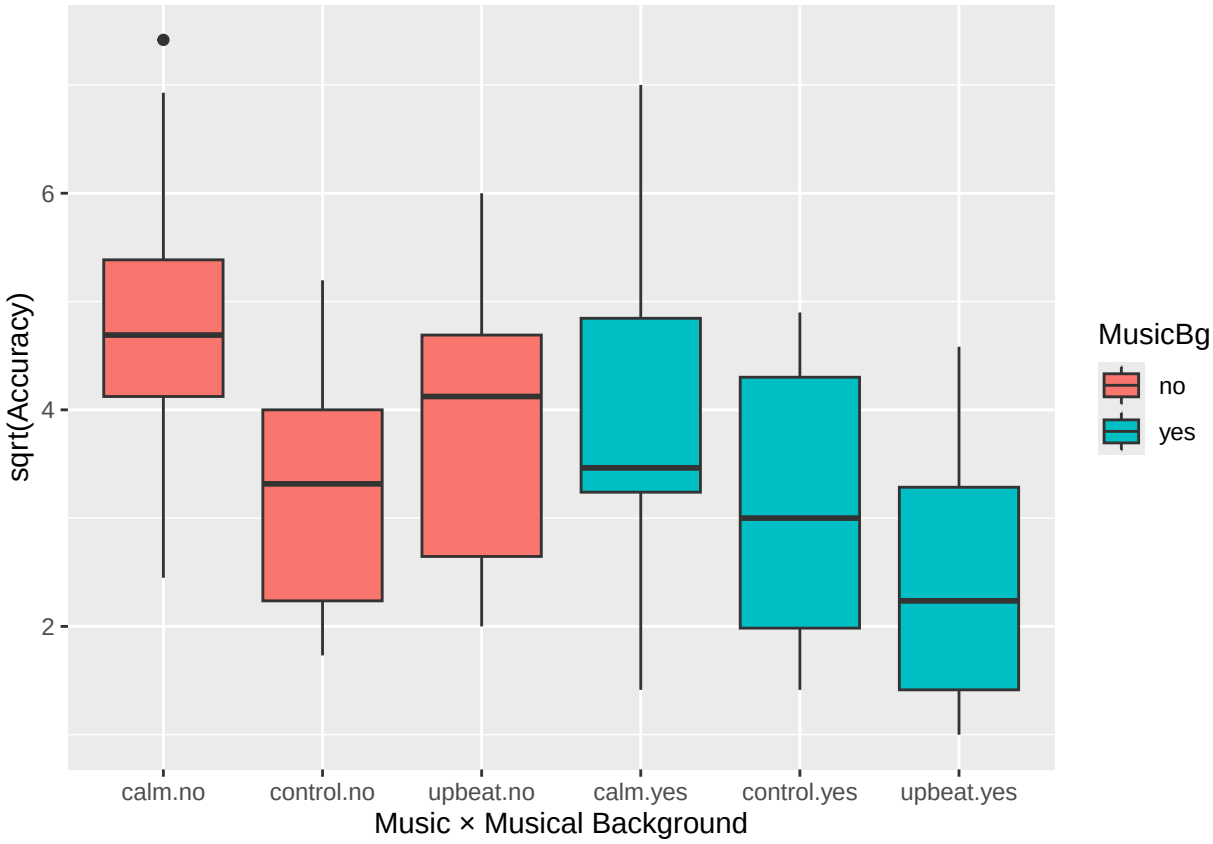
```
MusicTime$sqrtAccuracy <- sqrt(MusicTime$Accuracy)
MusicTime$Music <- factor(MusicTime$Music)
MusicTime$MusicBg <- factor(MusicTime$MusicBg)
MusicTime$Subject <- factor(MusicTime$Subject)

ggplot(MusicTime, aes(Music, sqrtAccuracy, fill = Music)) +
  geom_boxplot()
```




```
MusicTime |>
  group_by(Music) |>
  summarise(mean = mean(sqrtAccuracy),
            sd = sd(sqrtAccuracy))
```

```
## # A tibble: 3 x 3
##   Music    mean    sd
##   <fct>   <dbl> <dbl>
## 1 calm     4.32  1.63
## 2 control  3.20  1.26
## 3 upbeat   3.12  1.42
```

```
ggplot(MusicTime, aes(interaction(Music, MusicBg), sqrtAccuracy, fill = MusicBg)) +
  geom_boxplot() +
  labs(x = "Music × Musical Background", y = "sqrt(Accuracy)")
```




```
MusicTime |>
  group_by(Music, MusicBg) |>
  summarise(mean = mean(sqrtAccuracy), sd = sd(sqrtAccuracy))
```

```
## `summarise()` has grouped output by 'Music'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 6 x 4
## # Groups:   Music [3]
##   Music   MusicBg  mean    sd
##   <fct>   <fct>   <dbl> <dbl>
## 1 calm    no       4.86  1.60
## 2 calm    yes      3.88  1.59
## 3 control no       3.27  1.23
## 4 control yes      3.15  1.34
## 5 upbeat  no       3.90  1.26
## 6 upbeat  yes      2.48  1.25
```

```r
anova_result <- ezANOVA(
  data = MusicTime,
  dv = sqrtAccuracy,
  wid = Subject,
  within = .(Music),
  between = .(MusicBg),
  detailed = TRUE
)
```

```
## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().
```

```r
anova_result
```

```
## $ANOVA
##          Effect DFn DFd        SSn      SSd          F            p p<.05
## 1   (Intercept)   1  18 754.720068 56.18968 241.769668 7.053773e-12     *
## 2       MusicBg   1  18  10.512729 56.18968   3.367684 8.306769e-02
## 3         Music   2  36  18.031115 48.14145   6.741801 3.259430e-03     *
## 4 MusicBg:Music   2  36   4.404954 48.14145   1.647004 2.068097e-01
##          ges
## 1 0.87855074
## 2 0.09153932
## 3 0.14735848
## 4 0.04051051
##
## $`Mauchly's Test for Sphericity`
##          Effect         W          p p<.05
## 3         Music 0.7069301 0.05244448
## 4 MusicBg:Music 0.7069301 0.05244448
##
## $`Sphericity Corrections`
##          Effect       GGe       p[GG] p[GG]<.05      HFe       p[HF] p[HF]<.05
## 3         Music 0.7733534 0.007081708         * 0.832278 0.005783399         *
## 4 MusicBg:Music 0.7733534 0.213183170           0.832278 0.211739552
```

```r
df <- data.frame(
  Month = factor(c(1,2,3,4,5, 1,2,3,4,5, 1,2,3,4,5, 1,2,3,4,5, 1,2,3,4,5)),
  Home = factor(c("H1","H1","H1","H1","H1",
                  "H2","H2","H2","H2","H2",
                  "H3","H3","H3","H3","H3",
                  "H4","H4","H4","H4","H4",
                  "H5","H5","H5","H5","H5")),
  Brand = factor(c("A","B","C","D","E",
                   "B","C","D","E","A",
                   "C","D","E","A","B",
                   "D","E","A","B","C",
                   "E","A","B","C","D")),
  CADR = c(162,115,149,229,328,
           89,192,273,273,205,
           160,164,238,175,321,
           146,296,227,71,208,
```

```
           241,142,103,119,333)
)

model <- aov(CADR ~ Month + Home + Brand, data = df)
summary(model)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Month        4  44513   11128   3.854 0.0308 *
## Home         4   2178     544   0.189 0.9398
## Brand        4  58034   14509   5.025 0.0130 *
## Residuals   12  34647    2887
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(model, "Brand")
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = CADR ~ Month + Home + Brand, data = df)
##
## $Brand
##      diff         lwr       upr     p adj
## B-A -42.4 -150.721561  65.92156 0.7255499
## C-A -16.6 -124.921561  91.72156 0.9869863
## D-A  46.8  -61.521561 155.12156 0.6522142
## E-A  93.0  -15.321561 201.32156 0.1060932
## C-B  25.8  -82.521561 134.12156 0.9375271
## D-B  89.2  -19.121561 197.52156 0.1270915
## E-B 135.4   27.078439 243.72156 0.0127592
## D-C  63.4  -44.921561 171.72156 0.3839672
## E-C 109.6    1.278439 217.92156 0.0469018
## E-D  46.2  -62.121561 154.52156 0.6623672
```