ANOVA test code

2023-11-01

Import data

```
df <- read.csv("C:/Users/11139/Desktop/train.csv")
```

Create new columns

```
#create new columns
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## 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
```

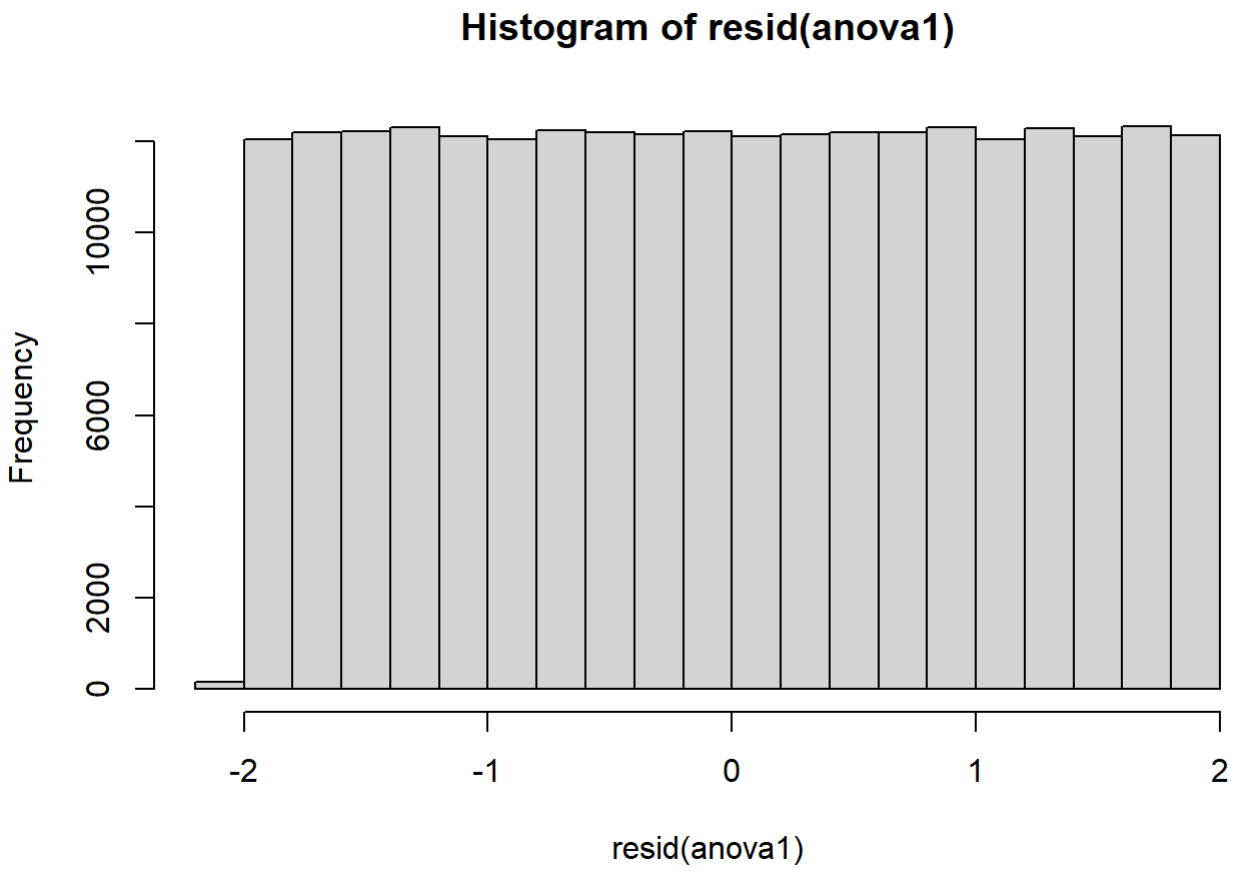
```
df <- df %>%
  mutate(Content = case_when(
    ContentType == "Both" ~ "Both2",
    ContentType == "Movies" ~ "MoviesTV1",
    ContentType == "TV Shows" ~ "MoviesTV1",
    TRUE ~ ContentType
  ))

df <- df %>%
  mutate(Subscription = case_when(
    SubscriptionType == "Basic" ~ "Basic1",
    SubscriptionType == "Premium" ~ "StandardPre2",
    SubscriptionType == "Standard" ~ "StandardPre2",
    TRUE ~ SubscriptionType
  ))
```

ANOVA assumption tests and box-cox transformations

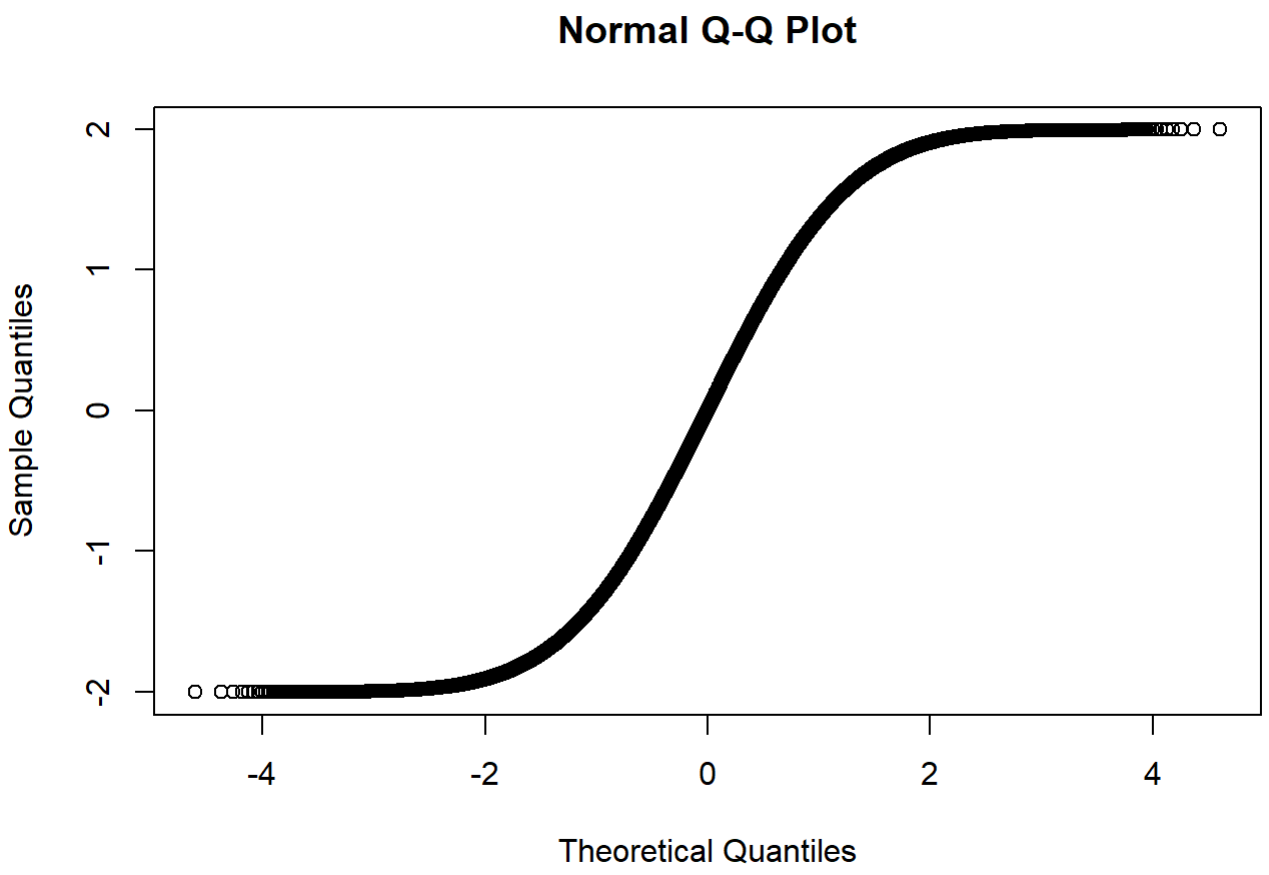
```
#anova assumption tests and box-cox transformations

anova1 <- aov(UserRating~MultiDeviceAccess, data=df)
summary(anova1)
```

```
##                       Df Sum Sq Mean Sq F value Pr(>F)
## MultiDeviceAccess      1      0  0.0244   0.018  0.892
## Residuals         243785 325363  1.3346
```

```
hist(resid(anova1))
```




```
qqnorm(resid(anova1))
```

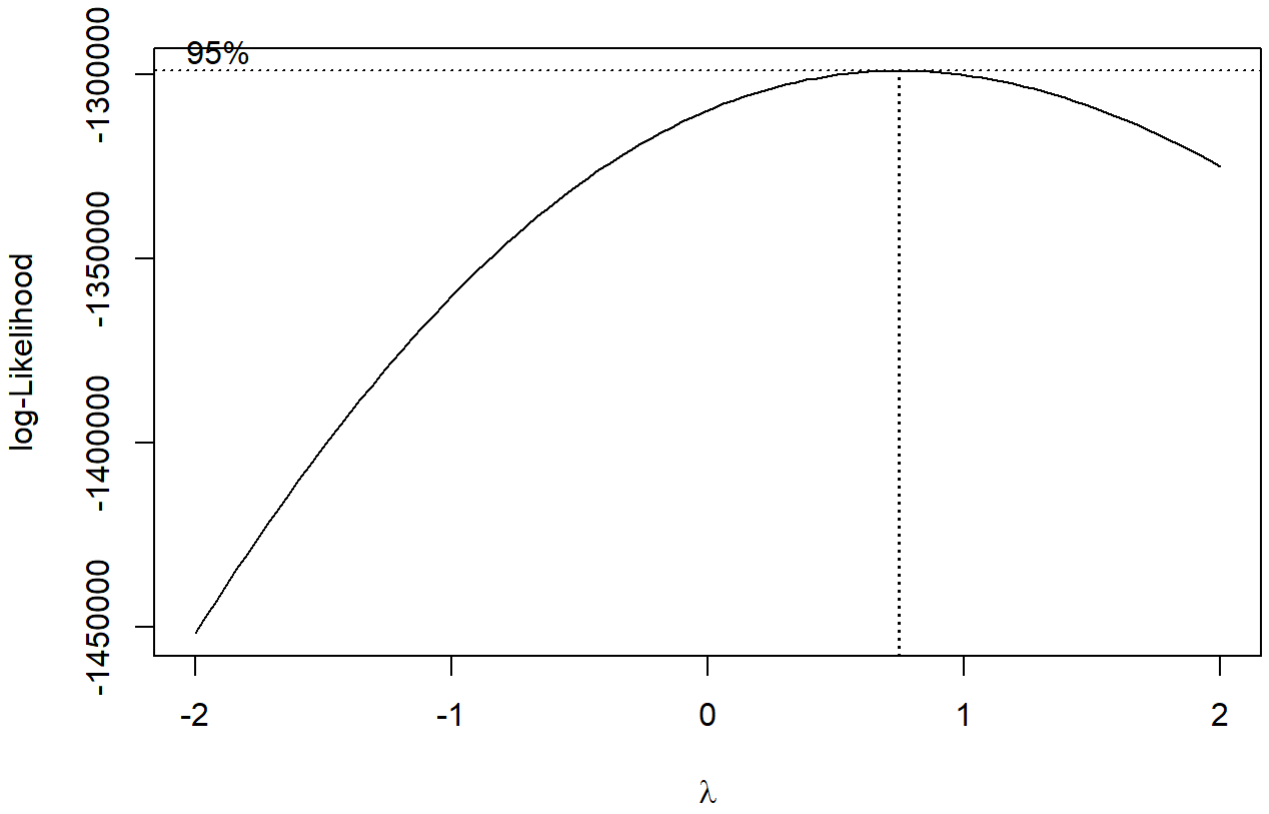



```
#box-cox transformation
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

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

```
lambda <- boxcox(UserRating ~ 1, data = df)
```




```
best_lambda <- lambda$x[which.max(lambda$y)]
df$UserRating_transformed <- ifelse(best_lambda == 0, log(df$UserRating), (df$UserRating^best_la
mbda - 1) / best_lambda)
model <- aov(UserRating_transformed ~ MultiDeviceAccess, data = df)
summary(model)
```

```
##                       Df    Sum Sq   Mean Sq F value Pr(>F)
## MultiDeviceAccess      1 0.000e+00 7.923e-24   0.998  0.318
## Residuals         243785 1.936e-18 7.941e-24
```

```
#Assume Ordinal or numeric data, no needs of two assumptions
kruskal.test(UserRating~MultiDeviceAccess, data=df)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  UserRating by MultiDeviceAccess
## Kruskal-Wallis chi-squared = 0.018856, df = 1, p-value = 0.8908
```

WatchlistSize~SubscriptionType

```
#This returns a "difference in location" measure of -4.65. The documentation for the wilcox.test
() function states this "does not estimate the difference in medians (a common misconception) bu
t rather the median of the difference between a sample from x and a sample from y."
#The confidence interval is fairly wide due to the small sample size, but it appears we can safe
ly say the median weight of company A's packaging is at least -0.1 less than the median weight o
f company B's packaging.

#Using the Kruskal-Wallis test to discover median differences, and then employing the wilcox.tes
t to determine the direction of group differences.
library(FSA)
```

```
## Warning: package 'FSA' was built under R version 4.2.3
```

```
## ## FSA v0.9.5. See citation('FSA') if used in publication.
## ## Run fishR() for related website and fishR('IFAR') for related book.
```

```
kruskal.test(WatchlistSize~SubscriptionType, data=df)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  WatchlistSize by SubscriptionType
## Kruskal-Wallis chi-squared = 5.1301, df = 2, p-value = 0.07692
```

```
dunnTest(WatchlistSize~SubscriptionType, data=df,method="bonferroni")
```

```
## Warning: SubscriptionType was coerced to a factor.
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison
```

```
##   p-values adjusted with the Bonferroni method.
```

```
##           Comparison         Z    P.unadj     P.adj
## 1    Basic - Premium -2.023208 0.04305169 0.1291551
## 2   Basic - Standard -1.894773 0.05812248 0.1743674
## 3 Premium - Standard  0.135192 0.89246005 1.0000000
```

```
#0.13,0.17 Basic vs Premium&Standard
wilcox.test(WatchlistSize~Subscription, data=df,conf.int = TRUE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  WatchlistSize by Subscription
## W = 6557936306, p-value = 0.02376
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -6.638373e-08 -4.473917e-05
## sample estimates:
## difference in location
##          -9.717818e-06
```

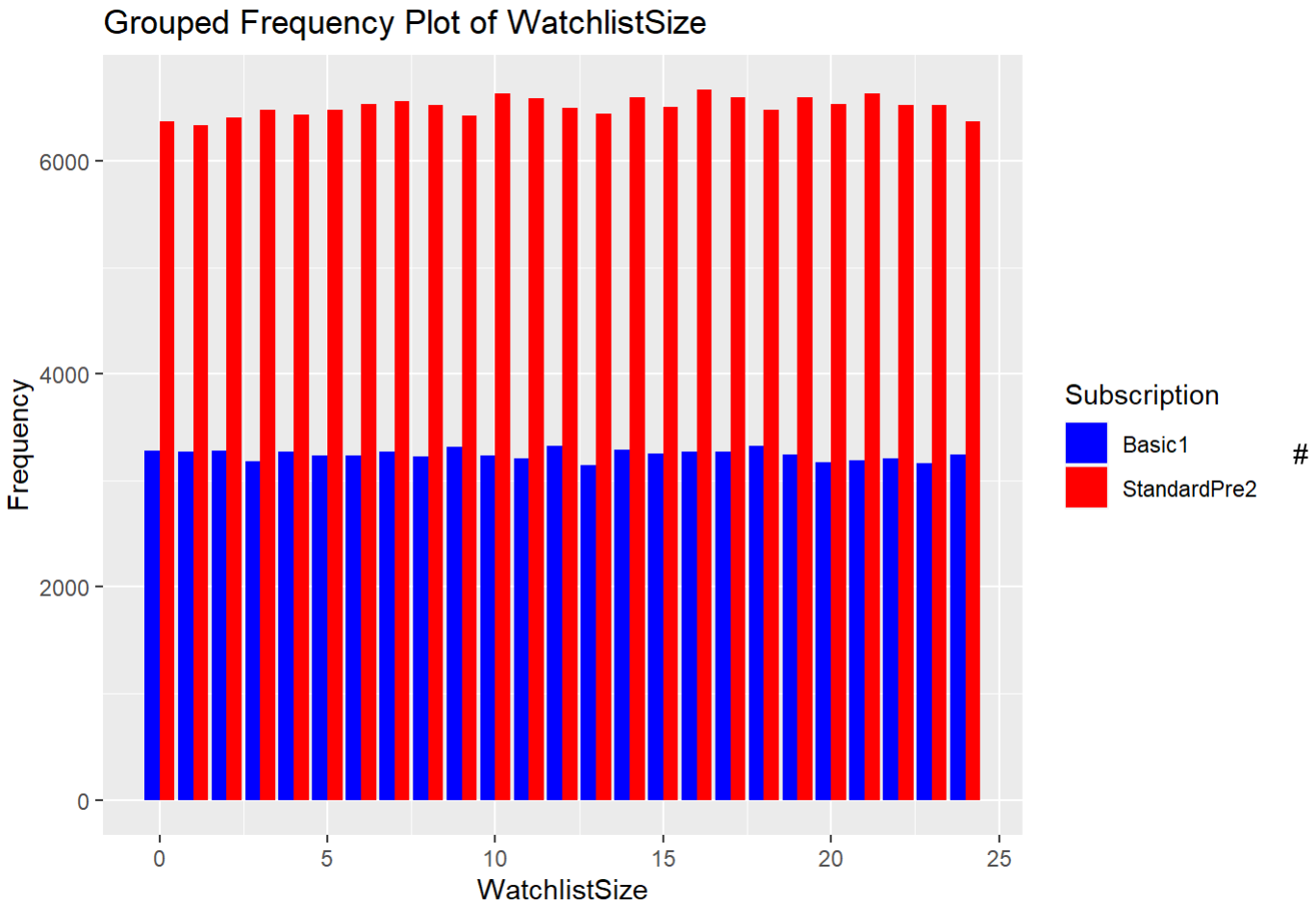
```
#0.024, -0.0, More WatchlistSize when subsscription is non-basic
table(df$SubscriptionType)
```

```
##
##    Basic  Premium Standard
##    81050    80817    81920
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
ggplot(df, aes(x = WatchlistSize, fill = Subscription)) +
  geom_bar(position = "dodge", stat = "count") +
  labs(title = "Grouped Frequency Plot of WatchlistSize",
       x = "WatchlistSize",
       y = "Frequency") +
  scale_fill_manual(values = c("Basic1" = "blue", "StandardPre2" = "red"))
```



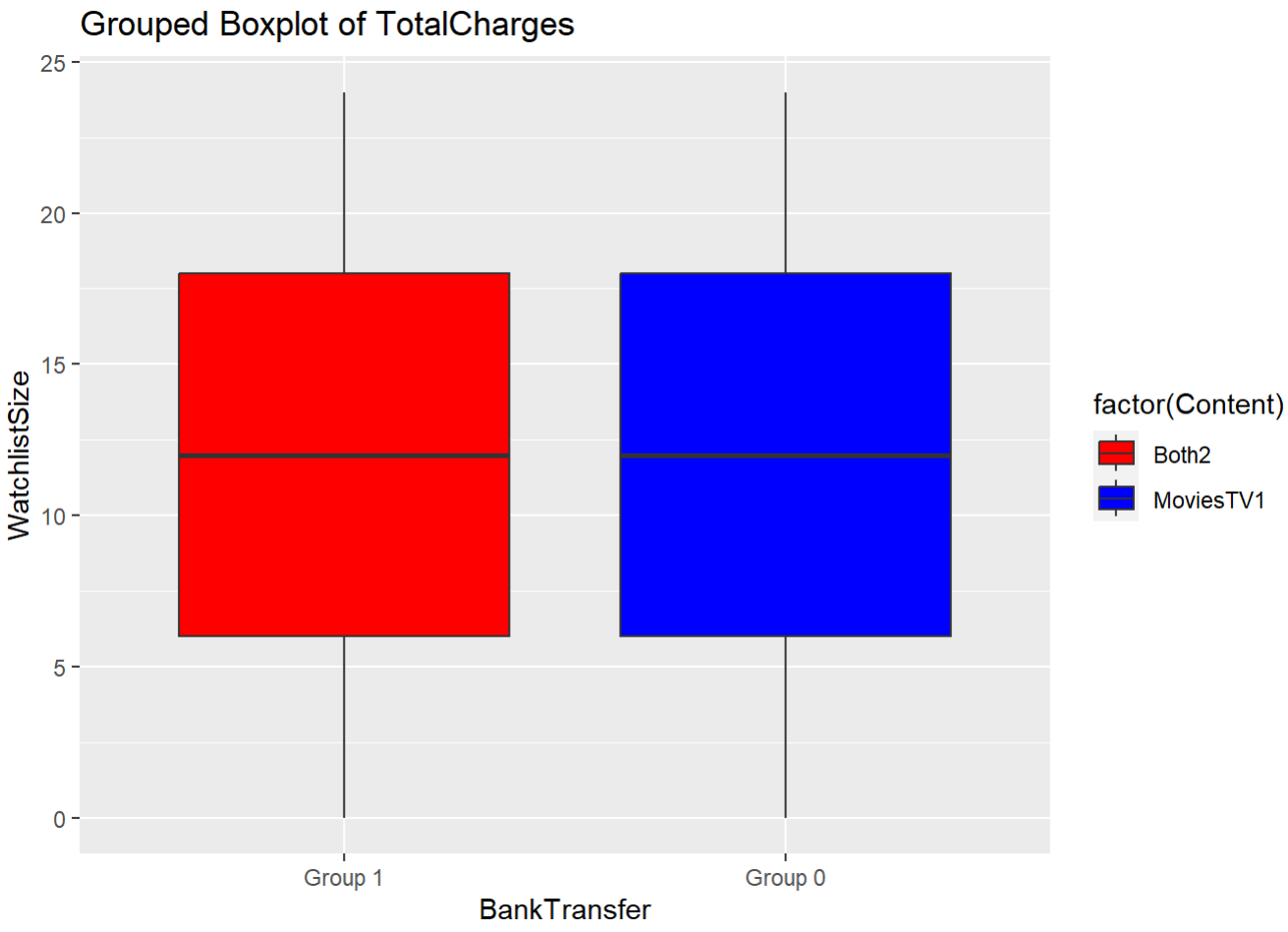
Significance: (WatchlistSize~SubscriptionType) have p-value as 0.02376 and negative values in difference of location. People with advanced subscriptions tend to use the app more frequently. When a customer with a basic but lengthy watchlist size appears, we can offer them a targeted upgrade discount. Once they upgrade to a Standard or Premium membership, the company can benefit from their potential longer-term commitment due to the higher-ranked subscription types and the enhanced benefits associated with the higher prices they pay.

TotalCharges~Content

```
wilcox.test(TotalCharges~Content, data=df,conf.int = TRUE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  TotalCharges by Content
## W = 6589897962, p-value = 0.04527
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -8.16062272 -0.08652015
## sample estimates:
## difference in location
##              -4.118148
```

```
library(ggplot2)
ggplot(df, aes(x = factor(Content), y = WatchlistSize, fill = factor(Content))) +
  geom_boxplot() +
  labs(title = "Grouped Boxplot of TotalCharges",
       x = "BankTransfer",
       y = "WatchlistSize") +
  scale_fill_manual(values = c("MoviesTV1" = "blue", "Both2" = "red")) +
  scale_x_discrete(labels = c("MoviesTV1" = "Group 0", "Both2" = "Group 1"))
```




```
#-4.118, 4 more totalcharges when Content both
# MonthlyCharges and AccountAge are not significant
```

Significance:

(TotalCharges~ContentType) has p-value 0.04527 and a negative value in difference of location. Customers who watch both TV and movies tend to have higher charges. The company may consider sending promotions or advertisements for other types of content to customers who prefer a specific content type.

#UserRating~PaymentMethod

```
kruskal.test(UserRating~GenrePreference, data=df)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  UserRating by GenrePreference
## Kruskal-Wallis chi-squared = 1.3353, df = 4, p-value = 0.8554
```

```
kruskal.test(ContentDownloadsPerMonth~PaymentMethod, data=df)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  ContentDownloadsPerMonth by PaymentMethod
## Kruskal-Wallis chi-squared = 5.8368, df = 3, p-value = 0.1198
```

```
kruskal.test(UserRating~PaymentMethod, data=df)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  UserRating by PaymentMethod
## Kruskal-Wallis chi-squared = 9.1857, df = 3, p-value = 0.02692
```

```
library(FSA)
df$PaymentMethod <- as.factor(df$PaymentMethod)
dunnTest(UserRating~PaymentMethod, data=df,method="bonferroni")
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison
```

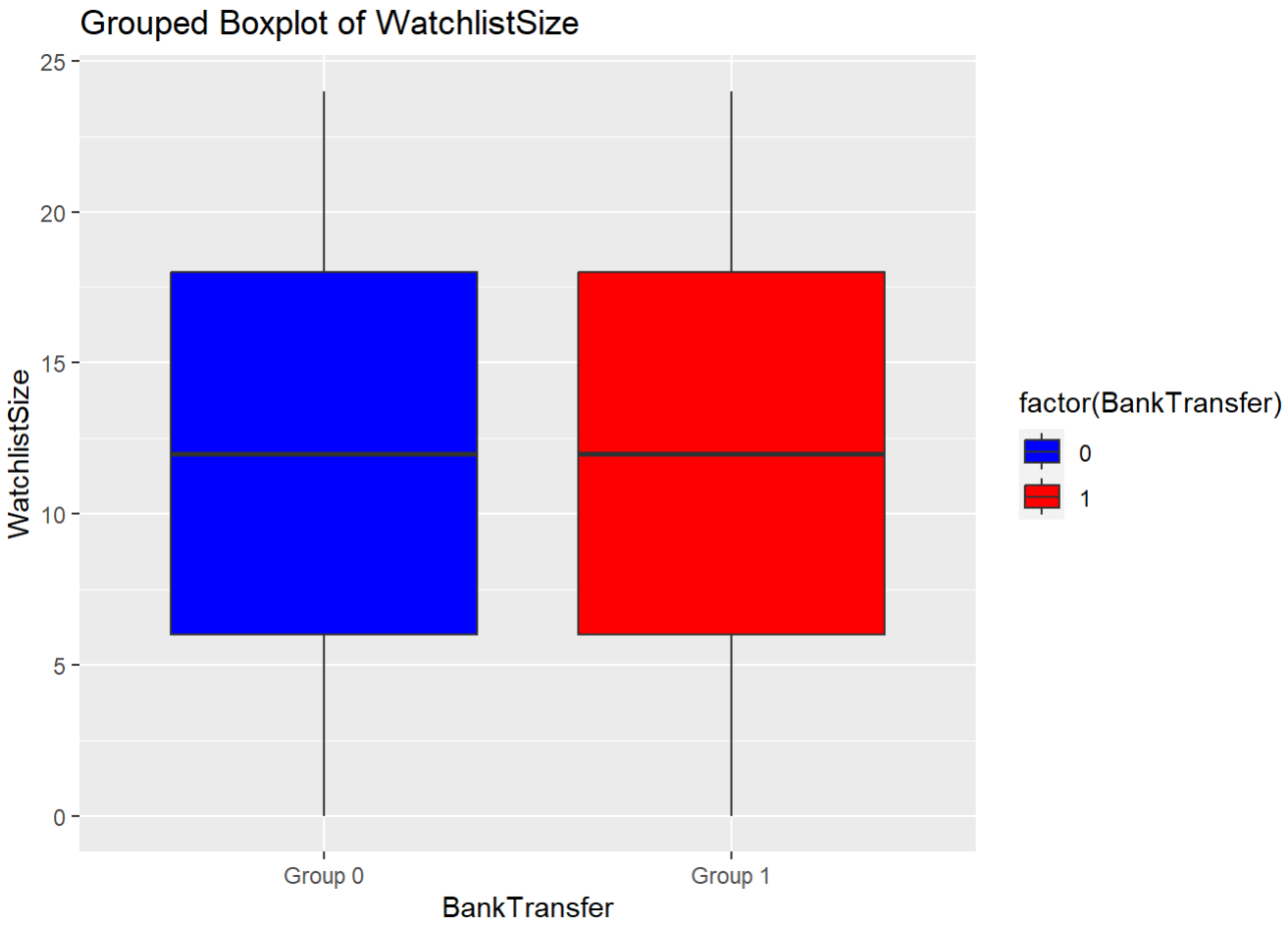
```
##   p-values adjusted with the Bonferroni method.
```

```
##                         Comparison         Z     P.unadj      P.adj
## 1      Bank transfer - Credit card 2.1281200 0.033327137 0.19996282
## 2 Bank transfer - Electronic check 2.4512449 0.014236306 0.08541784
## 3   Credit card - Electronic check 0.3199124 0.749034729 1.00000000
## 4     Bank transfer - Mailed check 2.7090480 0.006747657 0.04048594
## 5       Credit card - Mailed check 0.5827260 0.560077735 1.00000000
## 6  Electronic check - Mailed check 0.2639644 0.791807350 1.00000000
```

```
#0.04, bank transfer-mailed check
#0.03, bank transfer-credit card
#0.01, bank transfer-electronic check
# Assuming you have a data frame named df
df$BankTransfer <- ifelse(df$PaymentMethod == "Bank transfer", 1, 0)
wilcox.test(UserRating~BankTransfer, data=df,conf.int = TRUE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  UserRating by BankTransfer
## W = 5517909868, p-value = 0.002939
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.026768756 -0.005468087
## sample estimates:
## difference in location
##            -0.01609408
```

```
# 0.003, -0.02, higher 0.02 UserRating/Customer satisfaction rating when the payment method is B
ankTransfer
ggplot(df, aes(x = factor(BankTransfer), y = WatchlistSize, fill = factor(BankTransfer))) +
  geom_boxplot() +
  labs(title = "Grouped Boxplot of WatchlistSize",
       x = "BankTransfer",
       y = "WatchlistSize") +
  scale_fill_manual(values = c("0" = "blue", "1" = "red")) +
  scale_x_discrete(labels = c("0" = "Group 0", "1" = "Group 1"))
```



#Significance: (UserRating~Paymethod) has 0.002939 and a negative value in difference of location. Specifically, individuals who paid with BankTransfer demonstrated higher satisfaction. Notably, we have already examined various factors, such as gender, watchlist preferences, subscription type, genre preferences, and other dataset columns, and found no significant differences among users based on their payment methods. Therefore, the noteworthy higher satisfaction observed among BankTransfer users warrants further investigation. If the cause is operational convenience, enhancing the functionality of other payment types to improve overall payment convenience may contribute to higher customer ratings.