##<1>. EDA work and determine what test to apply

```r
library(gridExtra)
library(utils)
#library(DataExplorer)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::combine() masks gridExtra::combine()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(ggpubr)
library(rstatix)
```

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

```r
#library(factoextra)
library(FactoMineR)
#library(naniar)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
library(cluster)
library(arsenal)
```

```
##
## Attaching package: 'arsenal'
##
## The following object is masked from 'package:lubridate':
##
##     is.Date
```

```r
library(tidyverse)
library(plyr)
```

```
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:rstatix':
##
##     desc, mutate
##
## The following object is masked from 'package:ggpubr':
##
##     mutate
##
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
##
## The following object is masked from 'package:purrr':
##
##     compact
```

```r
chimplearn <- read.csv("chimp-learning.csv")
chimplearn2 <- read.csv("chimp-learning2.csv")

summary(chimplearn)
```

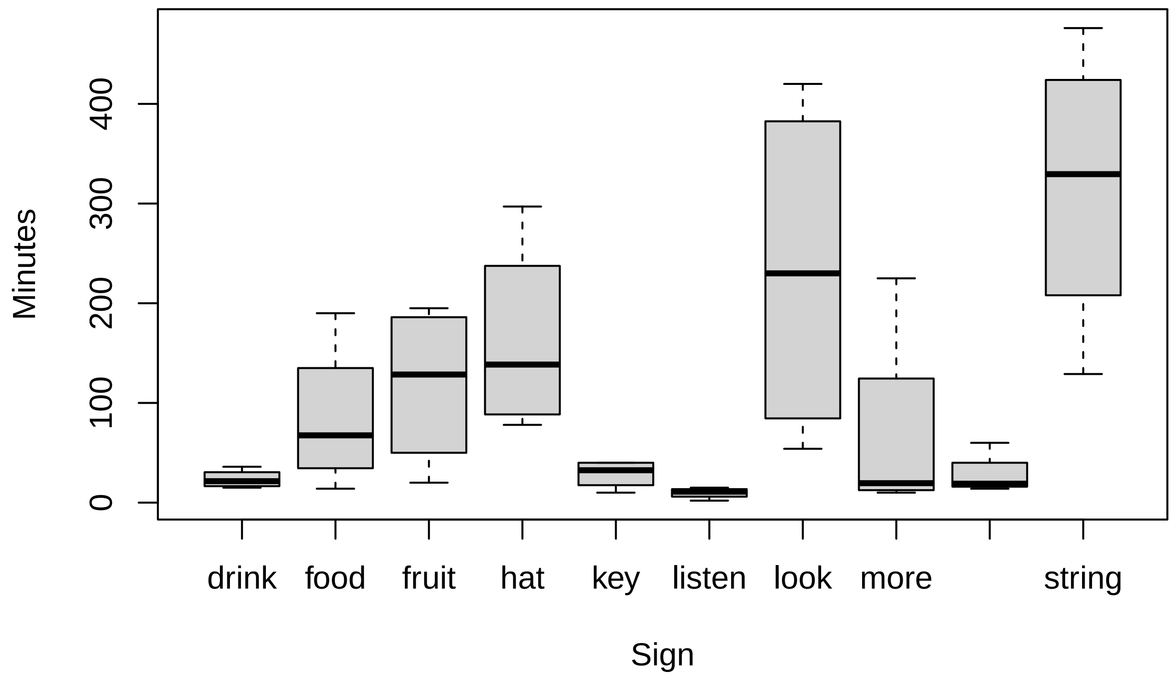
```
##     Minutes          Chimp               Sign               Order
##  Min.   :  2.00   Length:40          Length:40          Min.   : 1.0
##  1st Qu.: 17.25   Class :character   Class :character   1st Qu.: 3.0
##  Median : 47.00   Mode  :character   Mode  :character   Median : 5.5
##  Mean   :107.38                                         Mean   : 5.5
##  3rd Qu.:177.25                                         3rd Qu.: 8.0
##  Max.   :476.00                                         Max.   :10.0
```

```r
head(chimplearn)
```

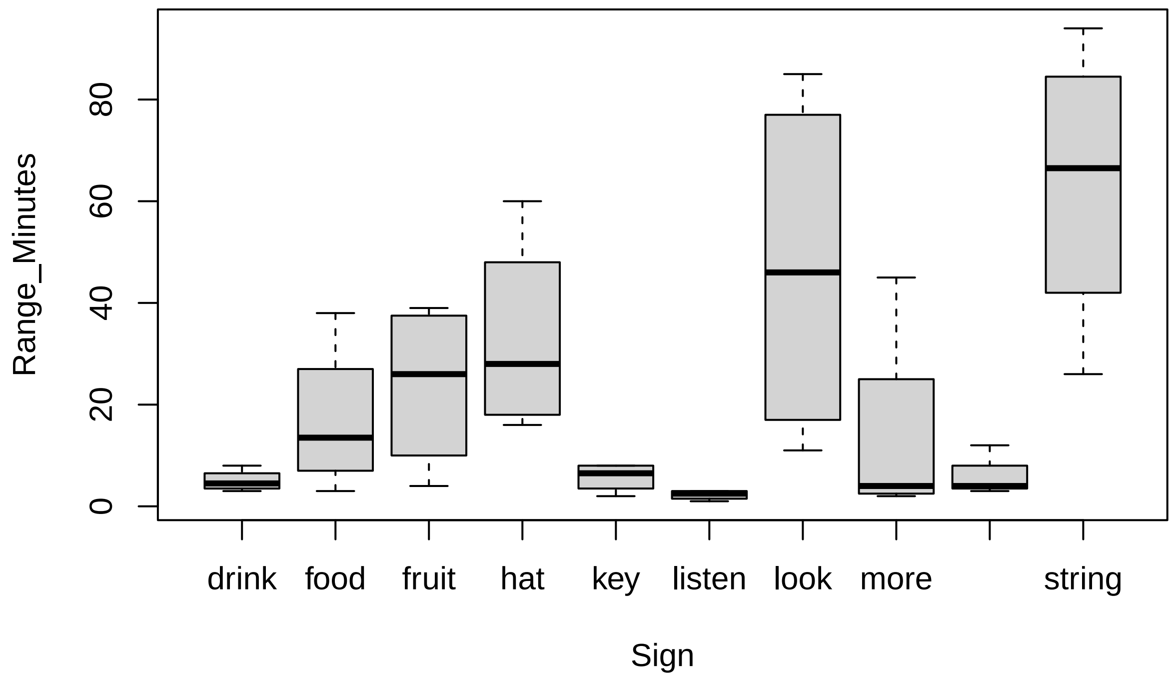
```
##   Minutes Chimp   Sign Order
## 1      12 Booee listen     8
## 2      15 Booee  drink     4
## 3      14 Booee   shoe     2
## 4      10 Booee    key     7
## 5      10 Booee   more     5
## 6      80 Booee   food    10
```

###Assumption 1 : Difficulty of every words are same for any chimp. ###Assumption 2 : All the chimps have same ability to learn the words.

```r
boxplot(Minutes ~ Sign, data = chimplearn )
```




```r
boxplot(Range_Minutes ~ Sign, data = chimplearn2 )
```



From the above boxplot, we can see that words 'look','string' has too big variance. Therefore, we can expect that the difficulty of the word is not same for each chimp since the range of each word is not same. This implies the rejection of 1st aussumption. Also, big range of the words 'look' and 'string' also indicates that the ability of chimps are not same.

So, we need to delete some chimps or words to eliminate this kind of words which bother the indication of same difficulty between the words. »>We need to take some statistical test to measure and claim that the difficulty same/vary even after processing data.

```
library("graphics")
mosaicplot(chimplearn, shade = TRUE, las=30,
           main = "Minutes")
mosaicplot(chimplearn2, shade = TRUE, las=30,
           main = "Range_Minutes")
library("gnm")
chimplearn3 <- select(chimplearn2, "Chimp", "Sign", "Range_Minutes") #"Range_Minutes"
chimplearn3 <- filter(chimplearn3, "Range_Minutes" > 30)
head(chimplearn3)
model <- xtabs(Range_Minutes~Sign + Chimp, data = chimplearn2)
mosaicplot(chimplearn3, gp = shading_max,
            split_vertical = TRUE, shade = TRUE,
            main="Sign")
```

```
library('mosaic')
```

```
## Registered S3 method overwritten by 'mosaic':
```

```
##   method                            from
##   fortify.SpatialPolygonsDataFrame ggplot2

##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.

##
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
##
##     mean

## The following object is masked from 'package:plyr':
##
##     count

## The following objects are masked from 'package:arsenal':
##
##     iqr, relrisk

## The following objects are masked from 'package:rstatix':
##
##     cor_test, prop_test, t_test

## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

## The following object is masked from 'package:purrr':
##
##     cross

## The following object is masked from 'package:ggplot2':
##
##     stat

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
library('data.table')
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library('vcd')
```

```
## Loading required package: grid

##
## Attaching package: 'vcd'

## The following object is masked from 'package:mosaic':
##
##     mplot
```

```
chimplearn3 <- select(chimplearn2, "Chimp", "Sign", "Range_Minutes") #"Range_Minutes"
chimplearn3 <- filter(chimplearn3, "Range_Minutes" > 30)
head(chimplearn3)
```

```
##   Chimp   Sign Range_Minutes
## 1 Booee listen             3
## 2 Booee  drink             3
## 3 Booee   shoe             3
## 4 Booee    key             2
## 5 Booee   more             2
## 6 Booee   food            16
```

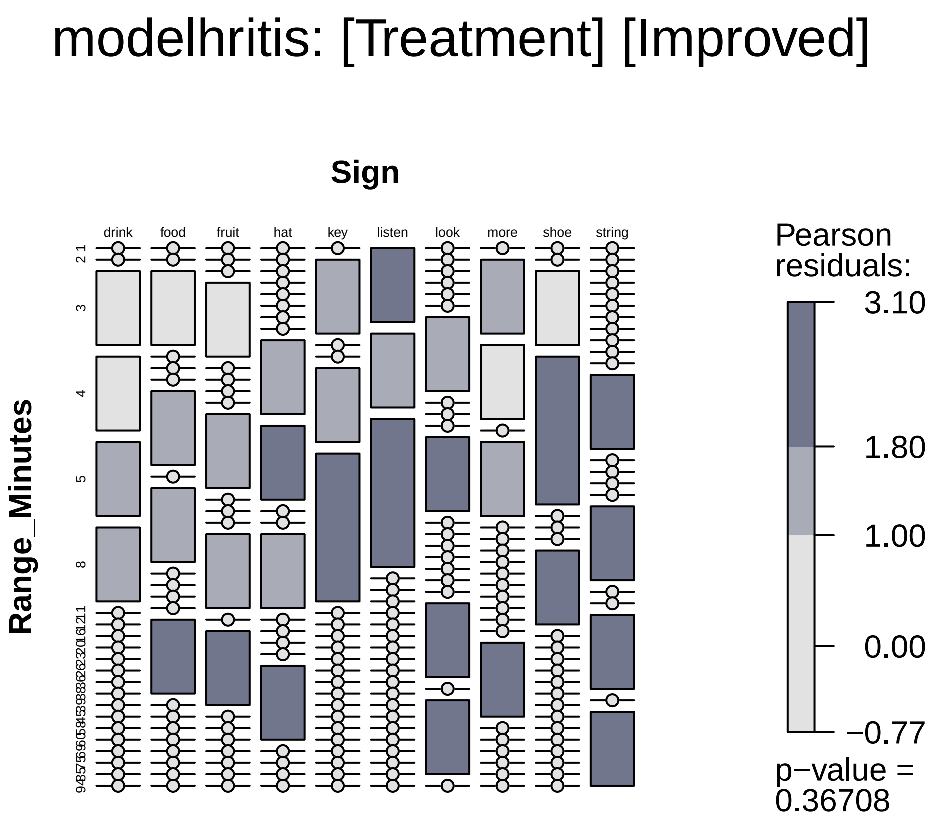
```
# creating dataset with above values
data <- table(chimplearn3)
summary(data)
```

```
## Number of cases in table: 40
## Number of factors: 3
## Test for independence of all factors:
##  Chisq = 840, df = 846, p-value = 0.5517
##  Chi-squared approximation may be incorrect
```

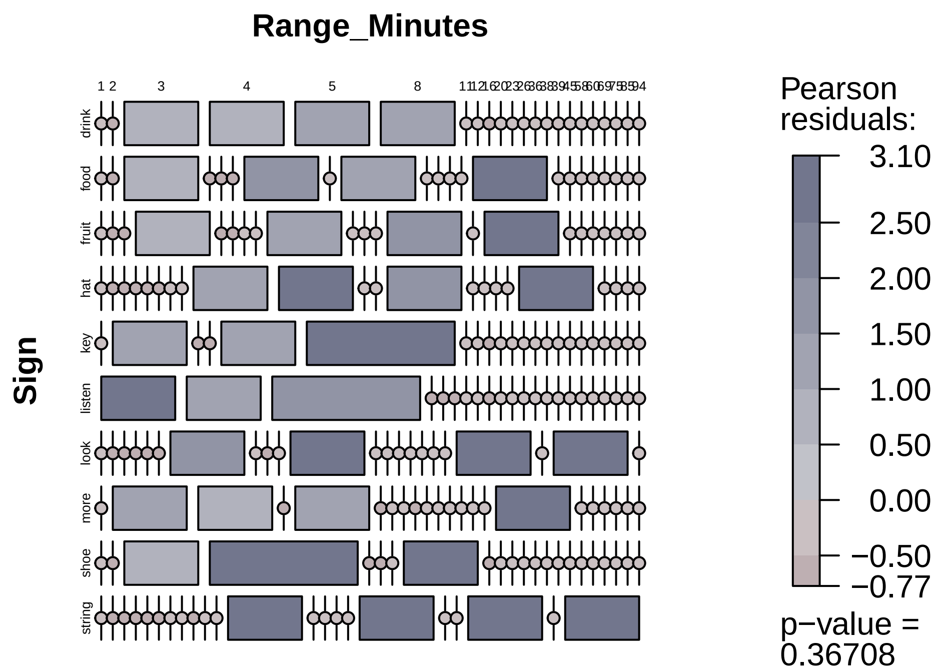
```
# plotting the mosaic chmodel
#setDT(data_frame)
```

```r
mosaic(data, shade=TRUE)
```

```r
model <- xtabs(~Sign + Range_Minutes, data = chimplearn3)
mosaic(model,
            split_vertical = TRUE, shade = TRUE,gp_labels = gpar(fontsize = 5), gp_args = list(interpola
            main="modelhritis: [Treatment] [Improved]")
```




```r
#gp = shading_hcl, gp_args = list(interpolate = c(1, 1.8))
gargs <- list(interpolate=c(-0.5, 0, 0.5, 1, 1.5, 2, 2.5))

mosaic(model, gp = shading_hcl, gp_labels = gpar(fontsize = 5), gp_args=gargs)
```



Let us observe what the above mosaic plot reveal. If all the blocks have same area across categories 'Sign' and 'Range_Minutes', it shows the independence between these categories.

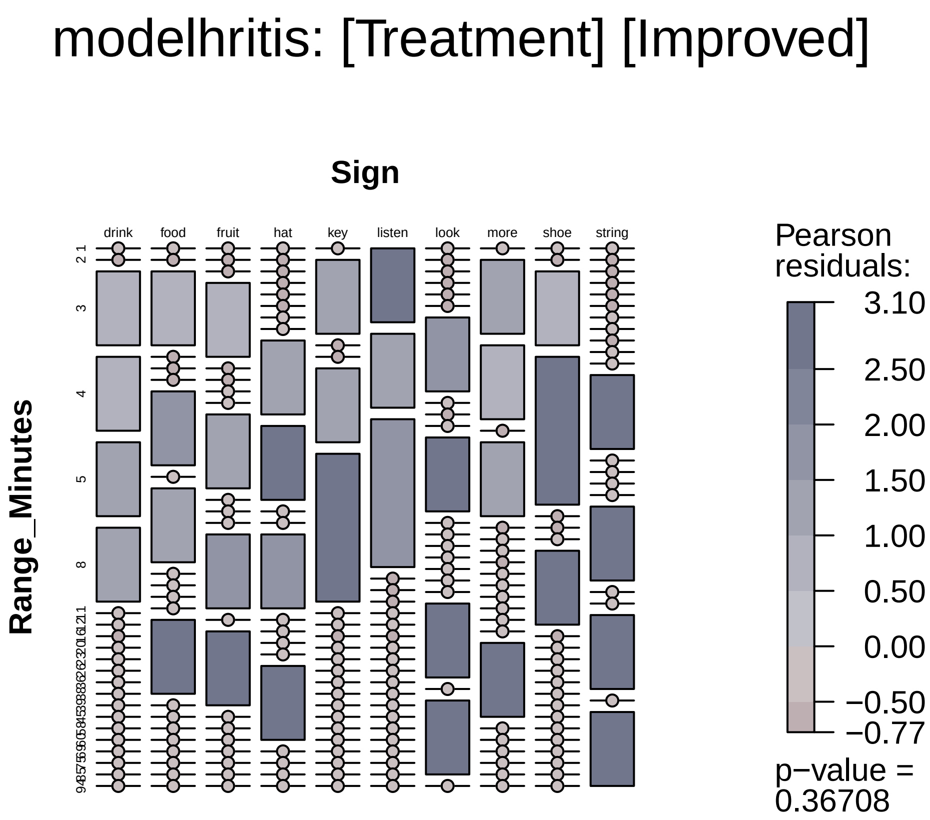
```
summary(model)
```

```
## Call: xtabs(formula = ~Sign + Range_Minutes, data = chimplearn3)
## Number of cases in table: 40
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 195, df = 189, p-value = 0.3671
##  Chi-squared approximation may be incorrect
```

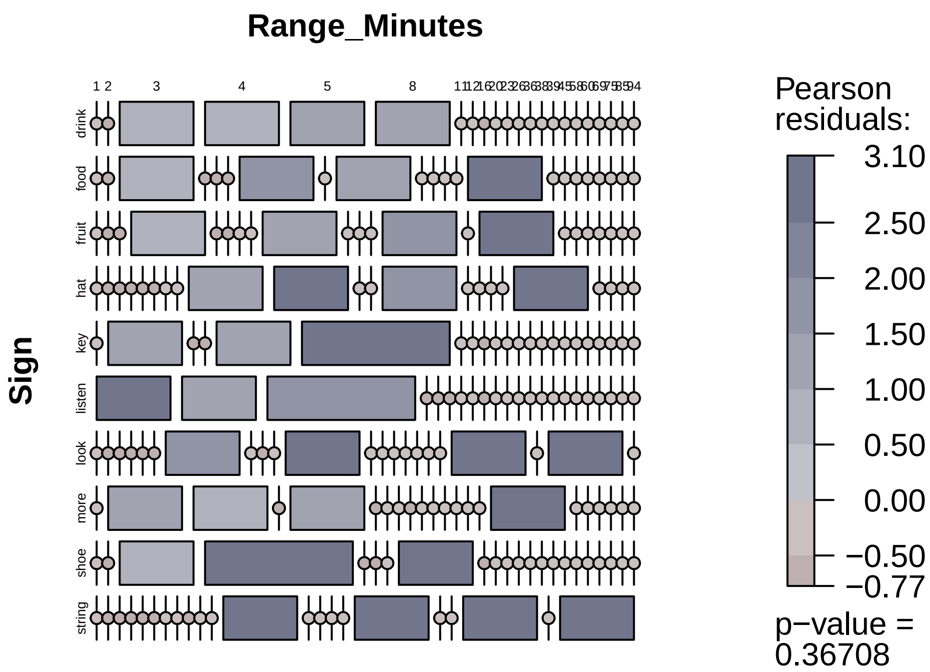
**For the above summary of Chi-square test, the p -value is big enough. In usual, we say two factors are dependent for p-value<0.01. The null hypothesis of independence of Chi-square test is that the two factors are independent, so we cannot reject the hypothesis.

```
model <- xtabs(~Sign + Range_Minutes, data = chimplearn3)
gargs <- list(interpolate=c(-0.5, 0, 0.5, 1, 1.5, 2, 2.5))
largs <- list(set_varnames=list(Sign="Sign",
                                Range_Minutes="Range_Minutes"),
              abbreviate=10)
mosaic(model,
```

```
         split_vertical = TRUE, shade = TRUE,gp_labels = gpar(fontsize = 5), gp_args=gargs,labeling_a
         main="modelhritis: [Treatment] [Improved]")
```




```
#gp = shading_hcl, gp_args = list(interpolate = c(1, 1.8))

mosaic(model,  gp = shading_hcl, gp_labels = gpar(fontsize = 5),
  gp_args=gargs)
```

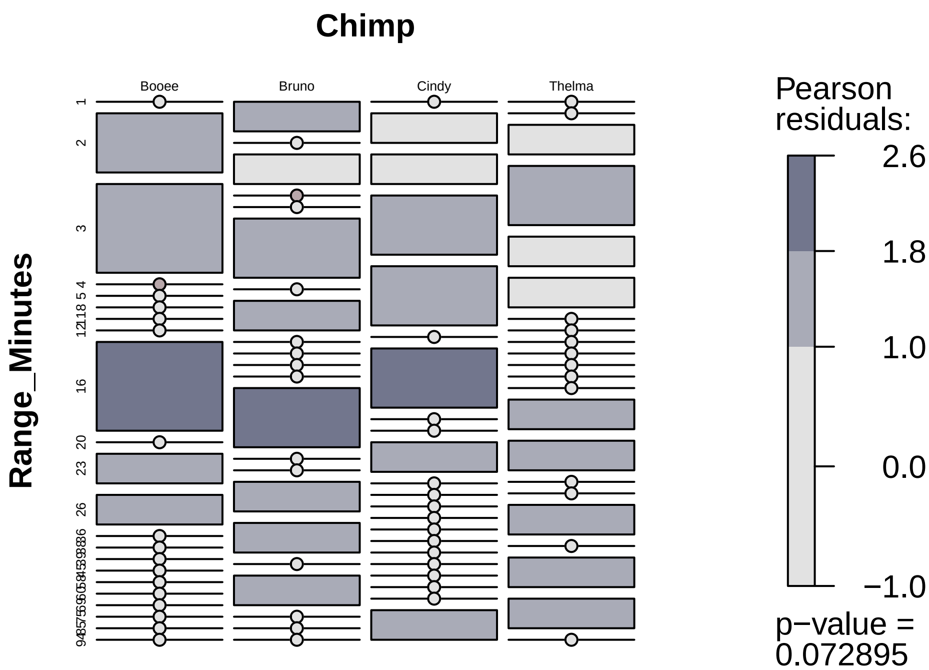



```r
#chimplisten <- select(chimplearn2, "Chimp", "Range_Minutes") #"Range_Minutes"

#data <- table(chimplearn3)
model2 <- xtabs(~Chimp + Range_Minutes, data = chimplearn3)
summary(model2)
```

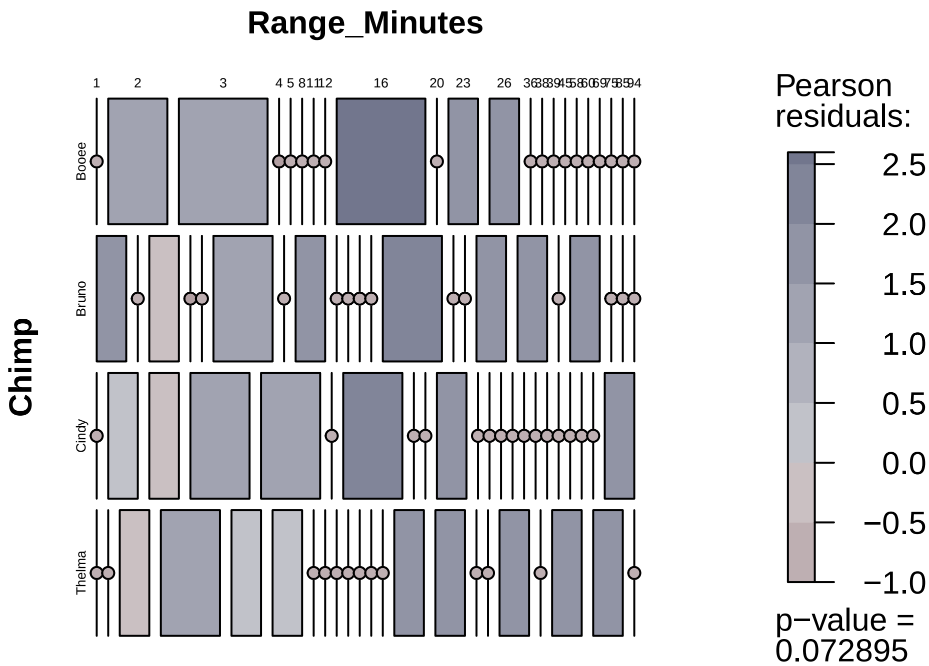
```
## Call: xtabs(formula = ~Chimp + Range_Minutes, data = chimplearn3)
## Number of cases in table: 40
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 80, df = 63, p-value = 0.0729
##  Chi-squared approximation may be incorrect
```

```r
mosaic(model2,
            split_vertical = TRUE, shade = TRUE,gp_labels = gpar(fontsize = 5), gp_args = list(interpola
            main="modelhritis: [Treatment] [Improved]")
```

modelhritis: [Treatment] [Improved]




```
#gp = shading_hcl, gp_args = list(interpolate = c(1, 1.8))

mosaic(model2, gp = shading_hcl, gp_labels = gpar(fontsize = 5), gp_args=gargs)
```



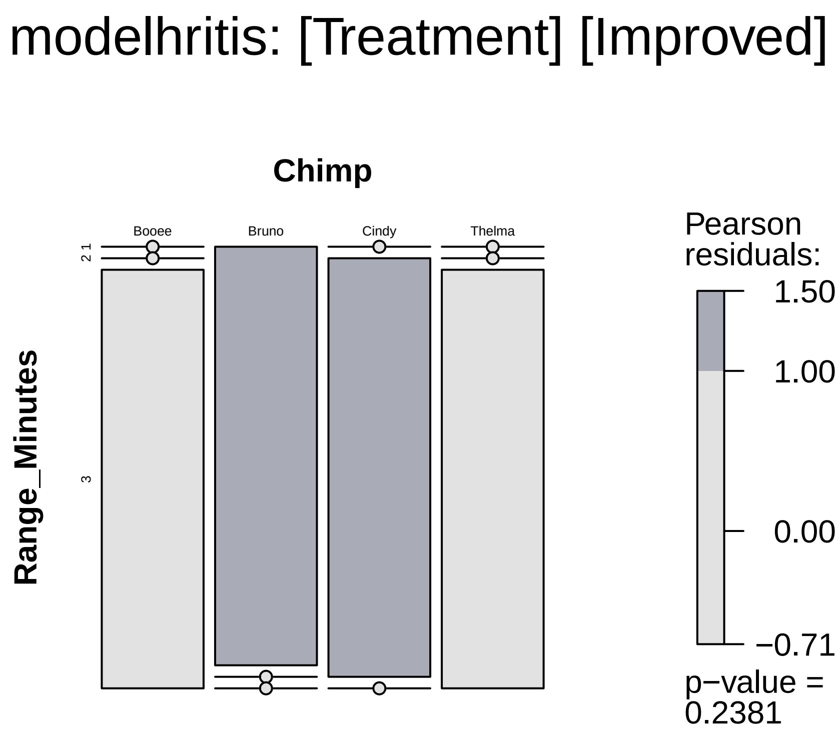

```r
chimplisten <- subset(chimplearn2, Sign == "listen")
chimplisten <- as_tibble(chimplisten)
head(chimplisten)
```

```
## # A tibble: 4 x 5
##   Minutes Chimp  Sign   Order Range_Minutes
##     <int> <chr>  <chr>  <int>         <int>
## 1      12 Booee  listen     8             3
## 2      10 Cindy  listen     8             2
## 3       2 Bruno  listen     8             1
## 4      15 Thelma listen     8             3
```

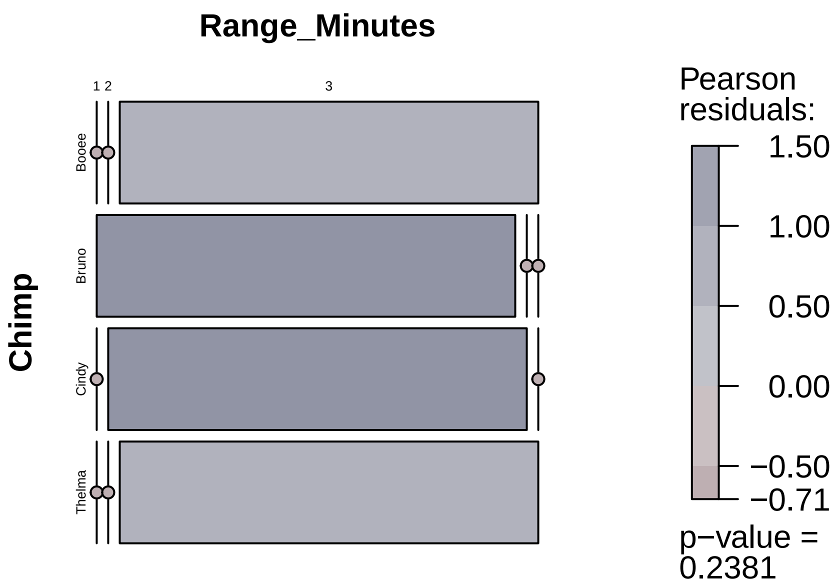
```r
model3 <- xtabs(~Chimp + Range_Minutes, data = chimplisten)
summary(model3)
```

```
## Call: xtabs(formula = ~Chimp + Range_Minutes, data = chimplisten)
## Number of cases in table: 4
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 8, df = 6, p-value = 0.2381
##  Chi-squared approximation may be incorrect
```

```
mosaic(model3,
            split_vertical = TRUE, shade = TRUE,gp_labels = gpar(fontsize = 5), gp_args = list(interpola
            main="modelhritis: [Treatment] [Improved]")
```




```
#gp = shading_hcl, gp_args = list(interpolate = c(1, 1.8))

mosaic(model3, gp = shading_hcl, gp_labels = gpar(fontsize = 5), gp_args=gargs)
```




```
chimplook <- subset(chimplearn2, Sign == "look")
chimplook <- as_tibble(chimplook)
head(chimplook)
```

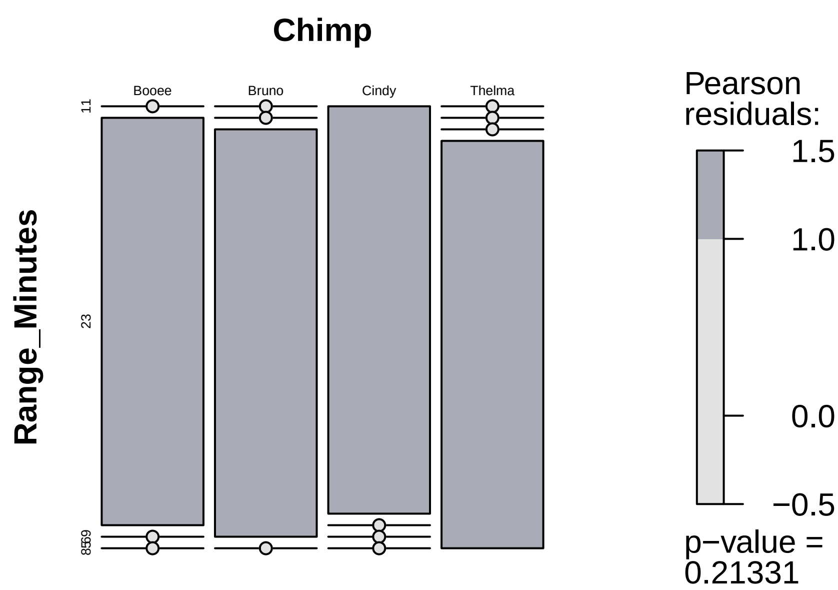
```
## # A tibble: 4 x 5
##   Minutes Chimp  Sign  Order Range_Minutes
##     <int> <chr>  <chr> <int>         <int>
## 1     115 Booee  look      6            23
## 2      54 Cindy  look      6            11
## 3     345 Bruno  look      6            69
## 4     420 Thelma look      6            85
```

```
model4 <- xtabs(~Chimp + Range_Minutes, data = chimplook)
summary(model4)
```

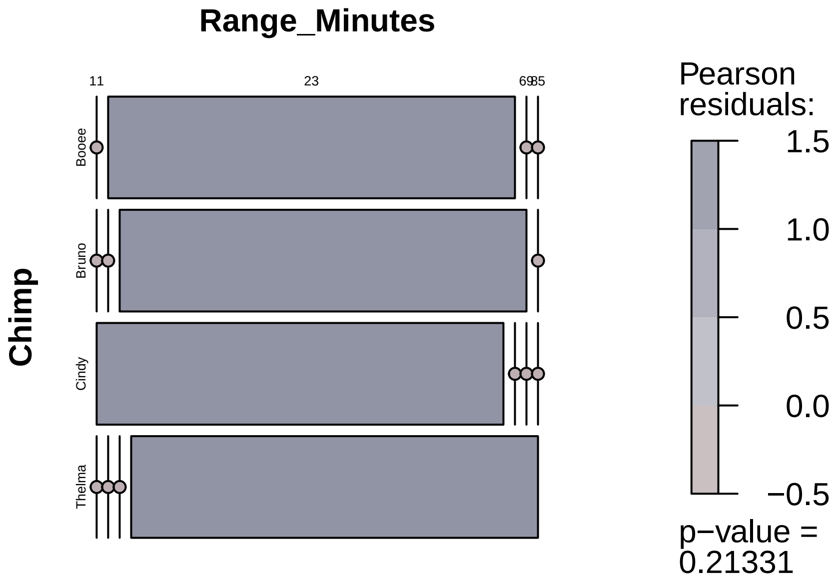
```
## Call: xtabs(formula = ~Chimp + Range_Minutes, data = chimplook)
## Number of cases in table: 4
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 12, df = 9, p-value = 0.2133
##  Chi-squared approximation may be incorrect
```

```
mosaic(model4,
            split_vertical = TRUE, shade = TRUE,gp_labels = gpar(fontsize = 5), gp_args = list(interpola
            main="modelhritis: [Treatment] [Improved]")
```

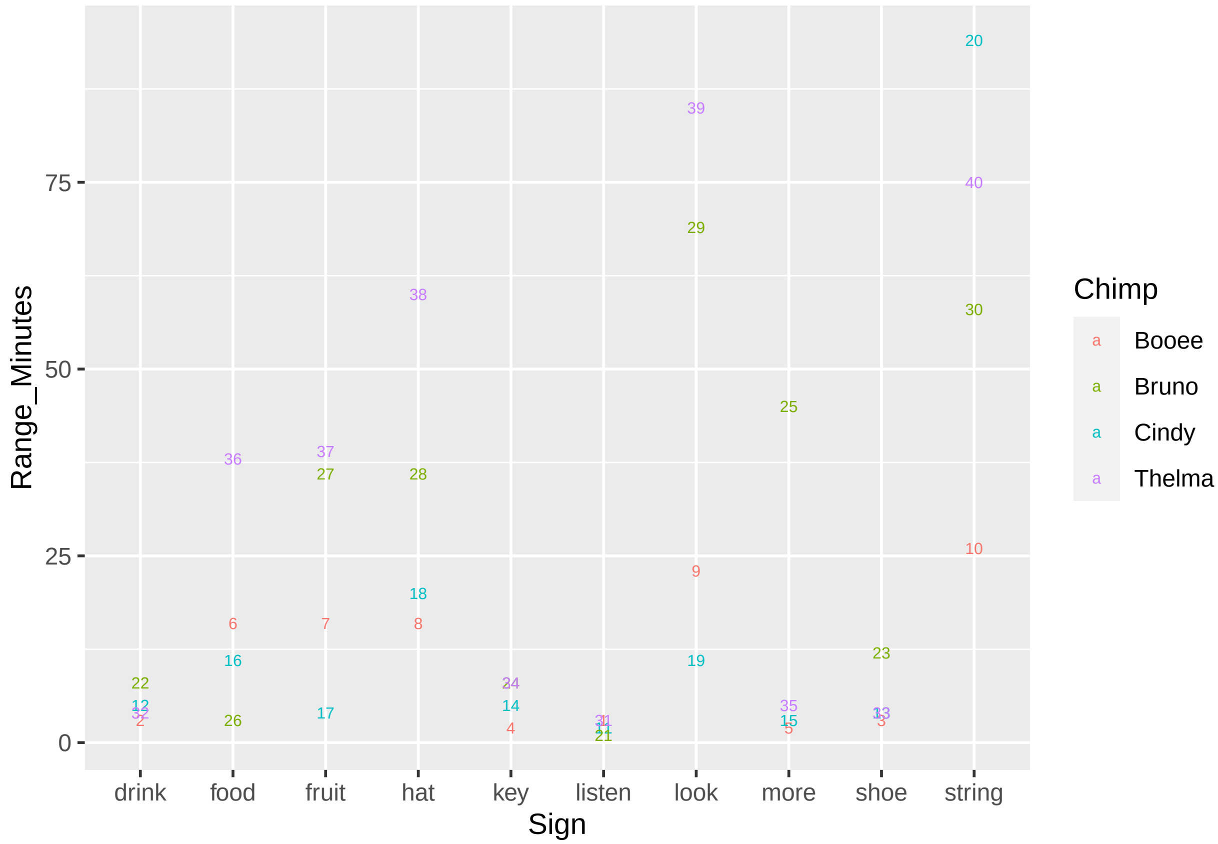
modelhritis: [Treatment] [Improved]



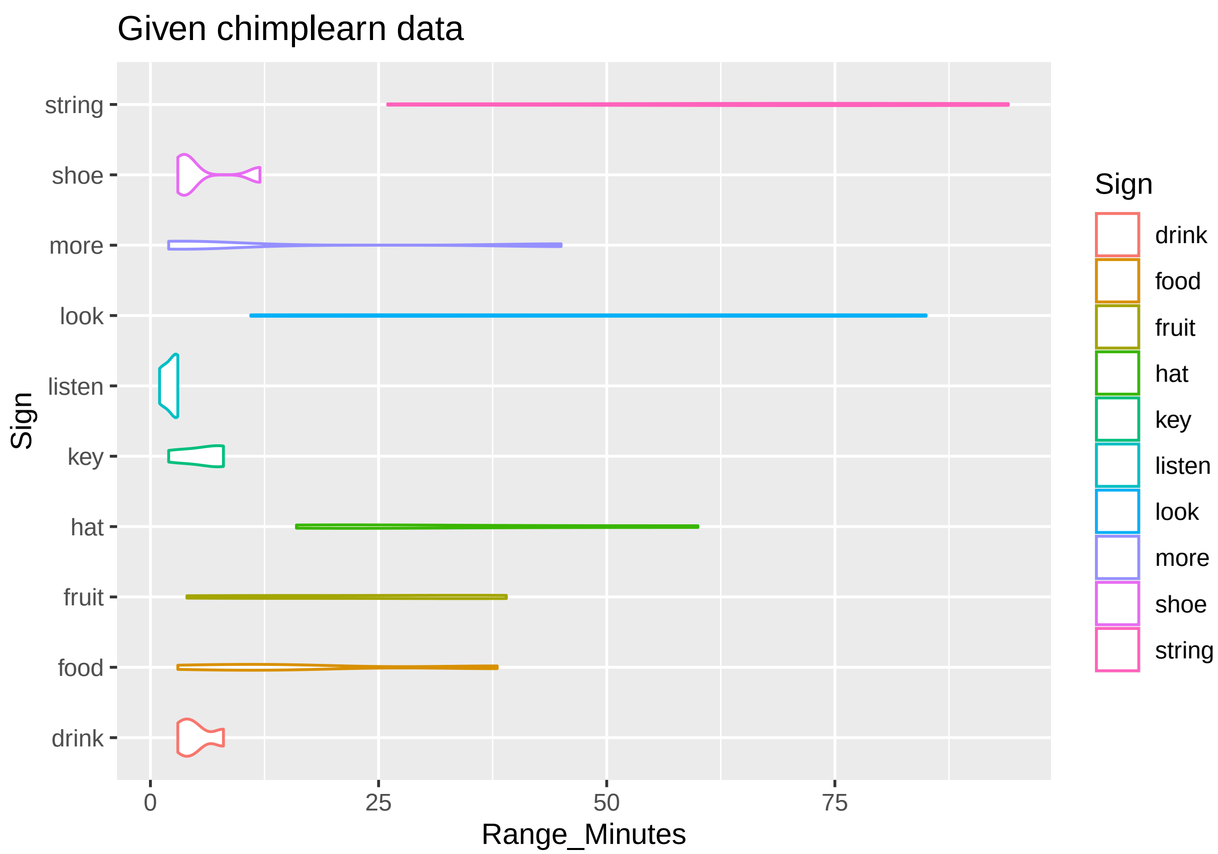

```r
#gp = shading_hcl, gp_args = list(interpolate = c(1, 1.8))

mosaic(model4, gp = shading_hcl, gp_labels = gpar(fontsize = 5), gp_args=gargs)
```




```r
## text geom
ggplot(chimplearn2, aes(x = Sign, y = Range_Minutes, label = rownames(chimplearn2), color=Chimp)) + geom
```



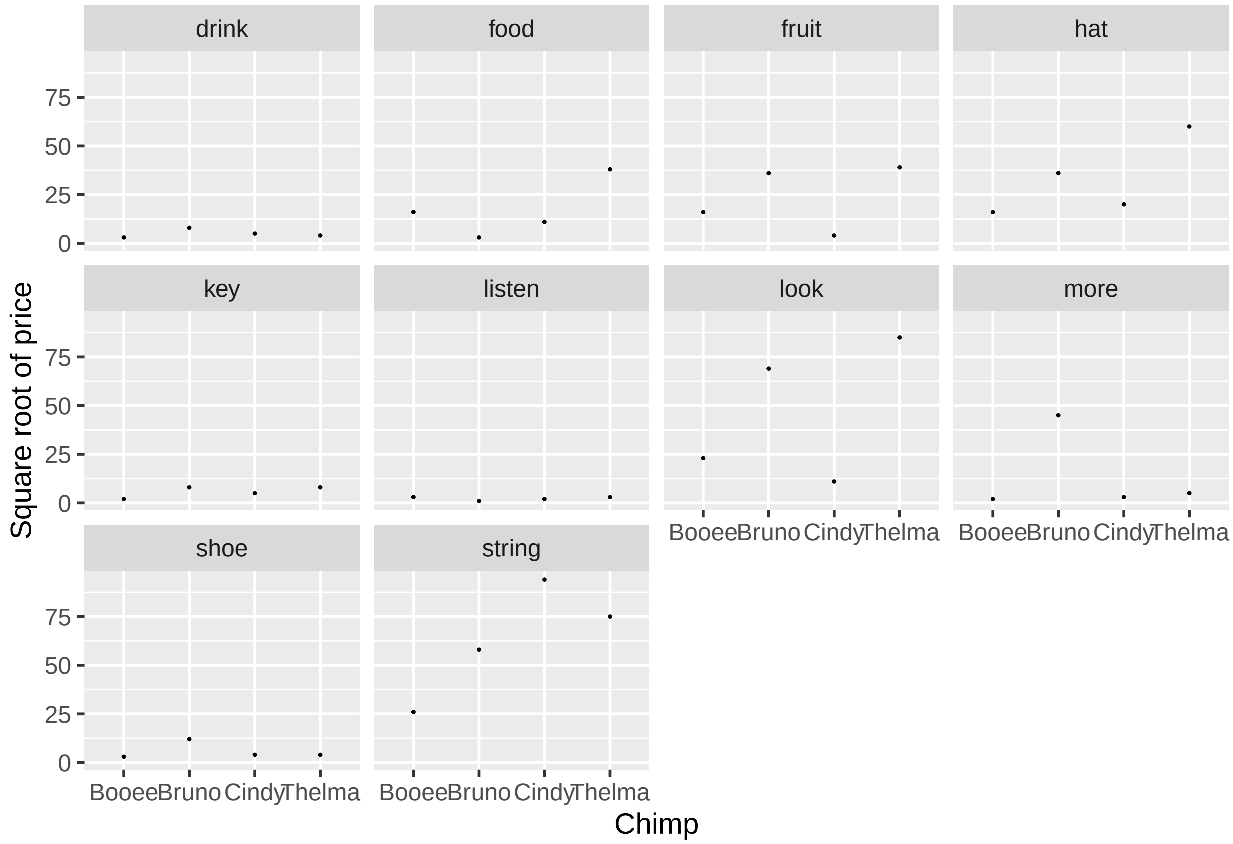

```r
ggplot(chimplearn2, aes(x = Sign, y = Range_Minutes, color = Sign)) +
    geom_violin() + coord_flip() + ggtitle("Given chimplearn data")
```




```
ggplot(chimplearn2, aes(x = Sign, y = Range_Minutes, color = Sign)) +
    geom_violin() + coord_flip() + ggtitle("chimplearn")
```

```
ggplot(chimplearn2, aes(x = Chimp, y = Range_Minutes)) +
    geom_point(size = .1) + stat_smooth(method = "lm", se = FALSE) +
    facet_wrap(~ Sign) +ylab("Square root of price")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

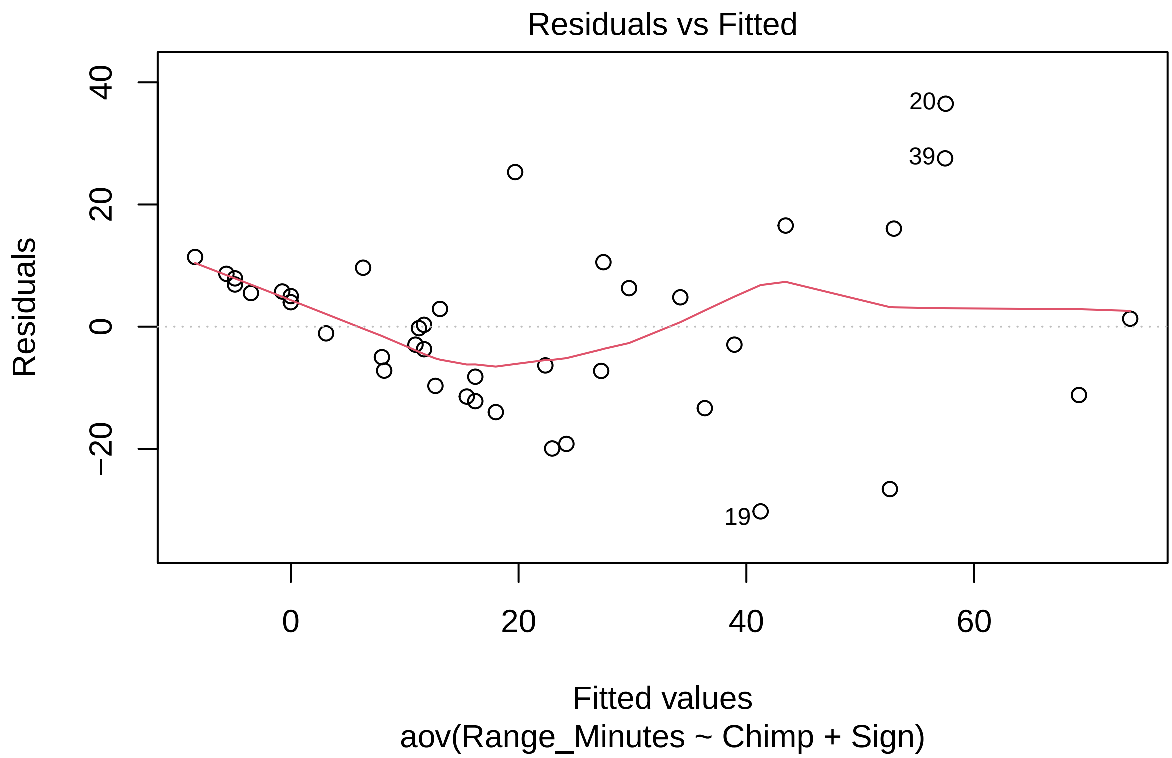


By using Anova having hypothesis statement as no difference between means of two factors, we determine whether we exclude some variables of Chimp and Sign or not.

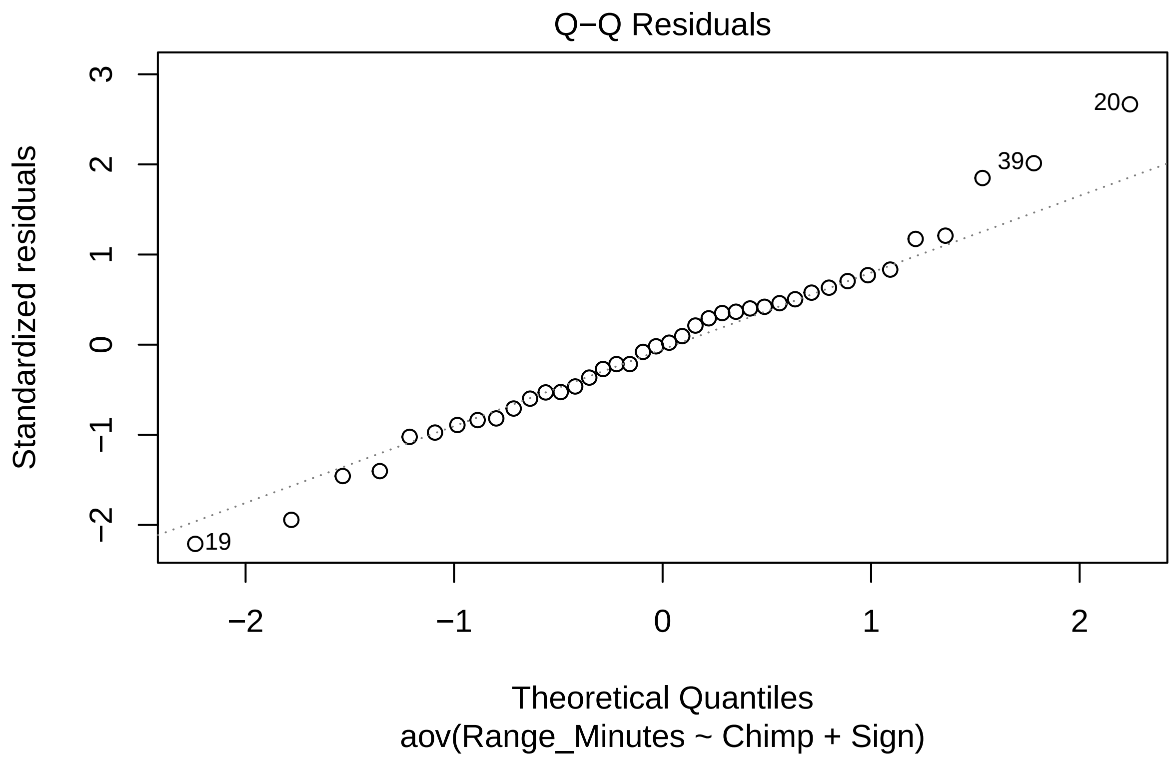
```r
preaov <- aov(Range_Minutes ~ Chimp + Sign , data = chimplearn2)


summary(preaov)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## Chimp        3   2911   970.3   3.497 0.02902 *
## Sign         9  14999  1666.5   6.006 0.00013 ***
## Residuals   27   7492   277.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(preaov, 1)
```




```
plot(preaov, 2)
```



TukeyHSD operates fitted ANOVA for pairwise variables as below.

```
TukeyHSD(preaov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Range_Minutes ~ Chimp + Sign, data = chimplearn2)
##
## $Chimp
##              diff         lwr       upr     p adj
## Bruno-Booee   16.6  -3.7857162 36.985716 0.1411374
## Cindy-Booee    4.9 -15.4857162 25.285716 0.9118830
## Thelma-Booee  21.1   0.7142838 41.485716 0.0404604
## Cindy-Bruno  -11.7 -32.0857162  8.685716 0.4115817
## Thelma-Bruno   4.5 -15.8857162 24.885716 0.9298895
## Thelma-Cindy  16.2  -4.1857162 36.585716 0.1559996
##
## $Sign
##                       diff        lwr        upr     p adj
## food-drink     1.200000e+01 -28.514364  52.514364 0.9883541
## fruit-drink    1.875000e+01 -21.764364  59.264364 0.8406603
## hat-drink      2.800000e+01 -12.514364  68.514364 0.3764914
## key-drink      7.500000e-01 -39.764364  41.264364 1.0000000
## listen-drink  -2.750000e+00 -43.264364  37.764364 0.9999999
## look-drink     4.200000e+01   1.485636  82.514364 0.0375878
```

```
## more-drink     8.750000e+00 -31.764364  49.264364 0.9988604
## shoe-drink     7.500000e-01 -39.764364  41.264364 1.0000000
## string-drink   5.825000e+01  17.735636  98.764364 0.0012369
## fruit-food     6.750000e+00 -33.764364  47.264364 0.9998583
## hat-food       1.600000e+01 -24.514364  56.514364 0.9295134
## key-food      -1.125000e+01 -51.764364  29.264364 0.9925757
## listen-food   -1.475000e+01 -55.264364  25.764364 0.9559806
## look-food      3.000000e+01 -10.514364  70.514364 0.2893668
## more-food     -3.250000e+00 -43.764364  37.264364 0.9999997
## shoe-food     -1.125000e+01 -51.764364  29.264364 0.9925757
## string-food    4.625000e+01   5.735636  86.764364 0.0160660
## hat-fruit      9.250000e+00 -31.264364  49.764364 0.9982502
## key-fruit     -1.800000e+01 -58.514364  22.514364 0.8690437
## listen-fruit  -2.150000e+01 -62.014364  19.014364 0.7145923
## look-fruit     2.325000e+01 -17.264364  63.764364 0.6226729
## more-fruit    -1.000000e+01 -50.514364  30.514364 0.9968475
## shoe-fruit    -1.800000e+01 -58.514364  22.514364 0.8690437
## string-fruit   3.950000e+01  -1.014364  80.014364 0.0605024
## key-hat       -2.725000e+01 -67.764364  13.264364 0.4124820
## listen-hat    -3.075000e+01 -71.264364   9.764364 0.2603470
## look-hat       1.400000e+01 -26.514364  54.514364 0.9680010
## more-hat      -1.925000e+01 -59.764364  21.264364 0.8201361
## shoe-hat      -2.725000e+01 -67.764364  13.264364 0.4124820
## string-hat     3.025000e+01 -10.264364  70.764364 0.2794643
## listen-key    -3.500000e+00 -44.014364  37.014364 0.9999995
## look-key       4.125000e+01   0.735636  81.764364 0.0434494
## more-key       8.000000e+00 -32.514364  48.514364 0.9994374
## shoe-key      -1.065814e-14 -40.514364  40.514364 1.0000000
## string-key     5.750000e+01  16.985636  98.014364 0.0014577
## look-listen    4.475000e+01   4.235636  85.264364 0.0217969
## more-listen    1.150000e+01 -29.014364  52.014364 0.9913314
## shoe-listen    3.500000e+00 -37.014364  44.014364 0.9999995
## string-listen  6.100000e+01  20.485636 101.514364 0.0006761
## more-look     -3.325000e+01 -73.764364   7.264364 0.1785333
## shoe-look     -4.125000e+01 -81.764364  -0.735636 0.0434494
## string-look    1.625000e+01 -24.264364  56.764364 0.9231921
## shoe-more     -8.000000e+00 -48.514364  32.514364 0.9994374
## string-more    4.950000e+01   8.985636  90.014364 0.0081701
## string-shoe    5.750000e+01  16.985636  98.014364 0.0014577
```

```
#t.test(Range_Minutes ~ Chimp|Sign, data = chimplearn2)
```

We have null hypothesis as significant difference between two variables. For the above p-values, small p-values imply the statistically meaningful difference between two designated variables for permitting this hypothesis. For the 'Chimp' variable, small p adj means that the Thelma has big different word learning minuites mean value compared with other Chimps. So we can delete the Thelma from the Chimp variables.

For the p-values between the vairables in 'Sign', string has small p adj values with other sign variables. So we can delete 'string' from the 'Sign'variable, since it has significantly different learning minutes mean value, compared with other sign variables.

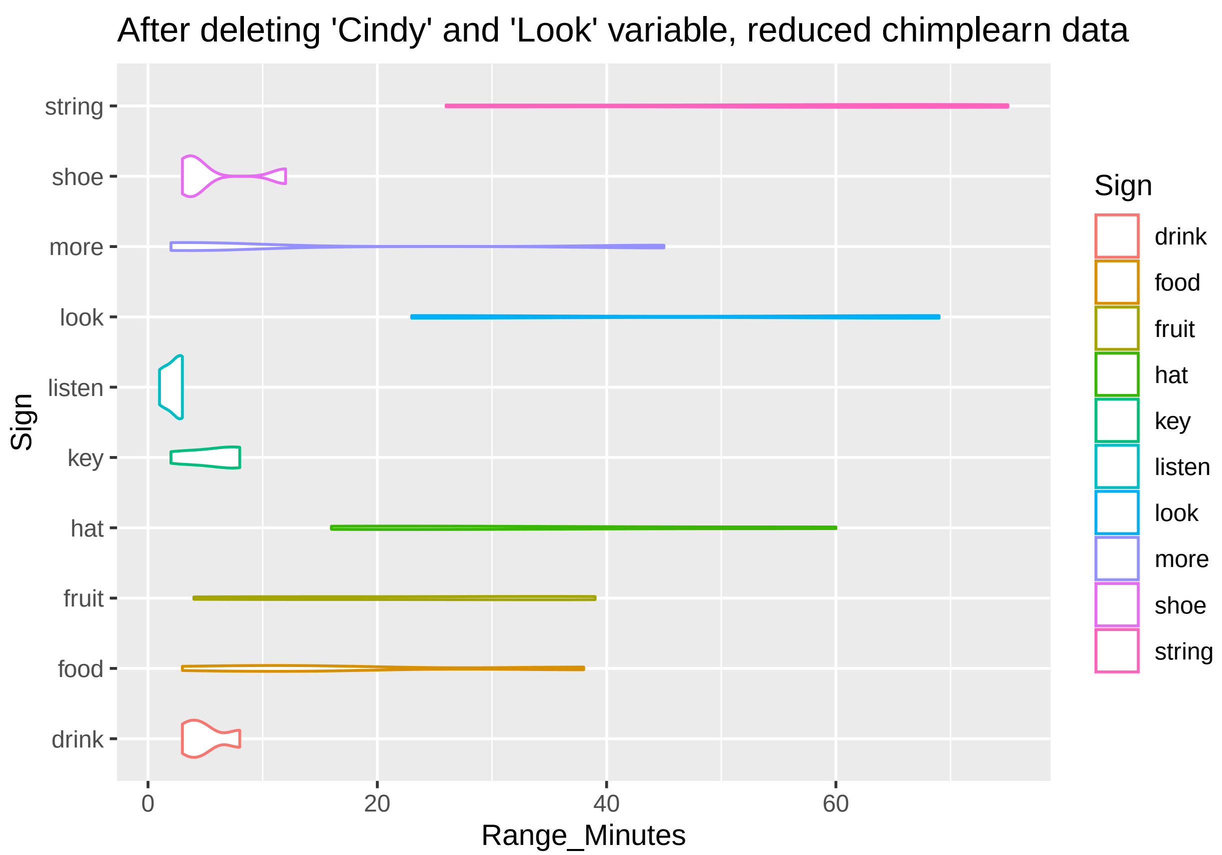
In the result, we can derive the following model by deleting the above two variable 'Thelma' and 'string'.

##<2>Deleting the Two variables

```r
chimp_reduced <- subset(chimplearn2, Chimp != "Cindy" | Sign != "string") #, Sign != "string"
chimp_reduced <- subset(chimp_reduced,  Chimp != "Cindy" |  Sign != "look")
chimp_reduced <- subset(chimp_reduced,  Chimp != "Thelma" | Sign != "look")
chimp_reduced <- as_tibble(chimp_reduced)
summary(chimp_reduced)
```

```
##     Minutes          Chimp               Sign               Order
##  Min.   :  2.00   Length:37          Length:37          Min.   : 1.000
##  1st Qu.: 15.00   Class :character   Class :character   1st Qu.: 3.000
##  Median : 40.00   Mode  :character   Mode  :character   Median : 5.000
##  Mean   : 90.41                                         Mean   : 5.378
##  3rd Qu.:129.00                                         3rd Qu.: 8.000
##  Max.   :372.00                                         Max.   :10.000
##  Range_Minutes
##  Min.   : 1.00
##  1st Qu.: 3.00
##  Median : 8.00
##  Mean   :18.27
##  3rd Qu.:26.00
##  Max.   :75.00
```

```r
ggplot(chimp_reduced, aes(x = Sign, y = Range_Minutes, color = Sign)) +
    coord_flip() + ggtitle("After deleting 'Cindy' and 'Look' variable, reduced chimplearn data") + geom
```

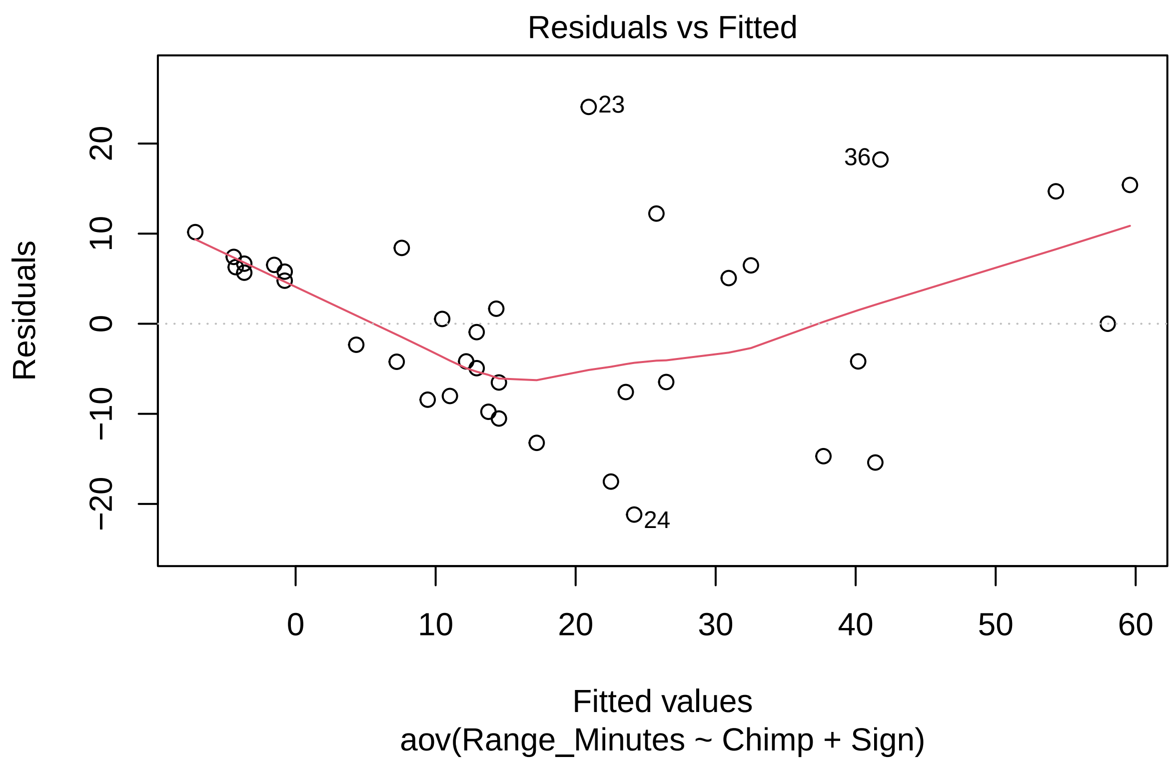


##<3>Setting model to identify difficulty of Sign to learn is common for the Chimps.

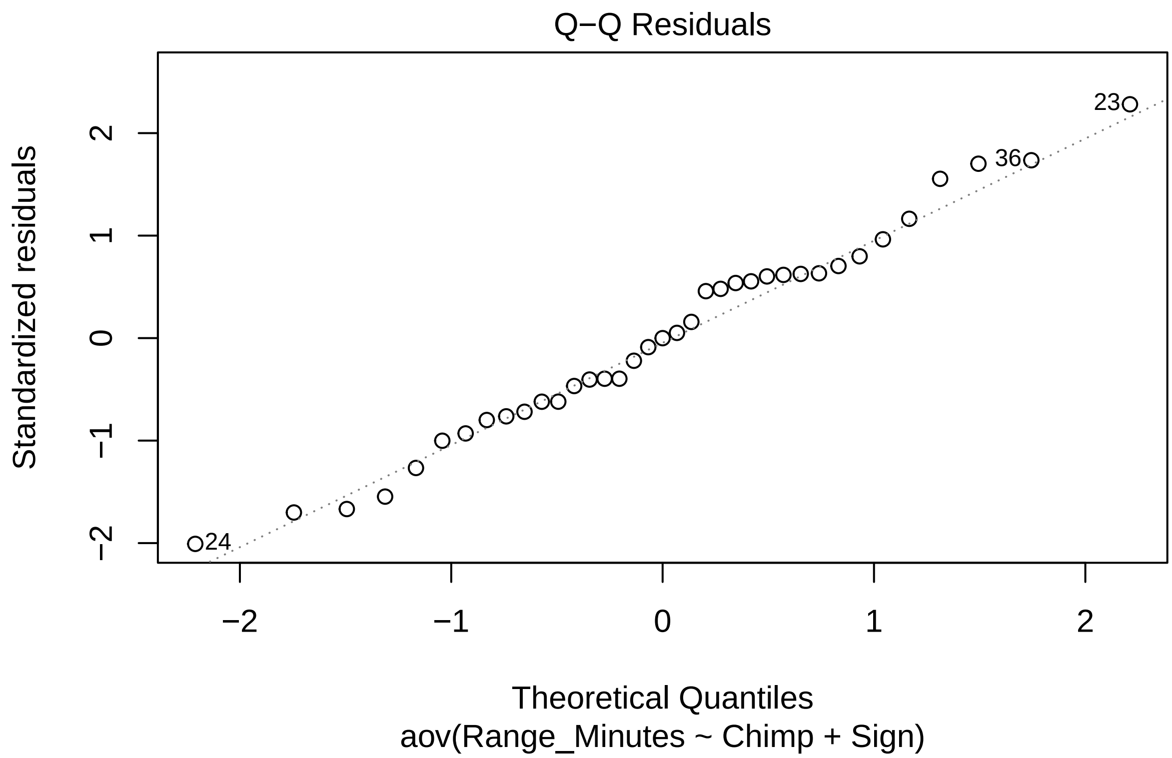
```r
aov <- aov(Range_Minutes ~ Chimp + Sign, data = chimp_reduced)
summary(aov)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## Chimp        3   3030  1009.9   6.090 0.003118 ** 
## Sign         9   8607   956.4   5.767 0.000279 ***
## Residuals   24   3980   165.8                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(aov, 1)
```



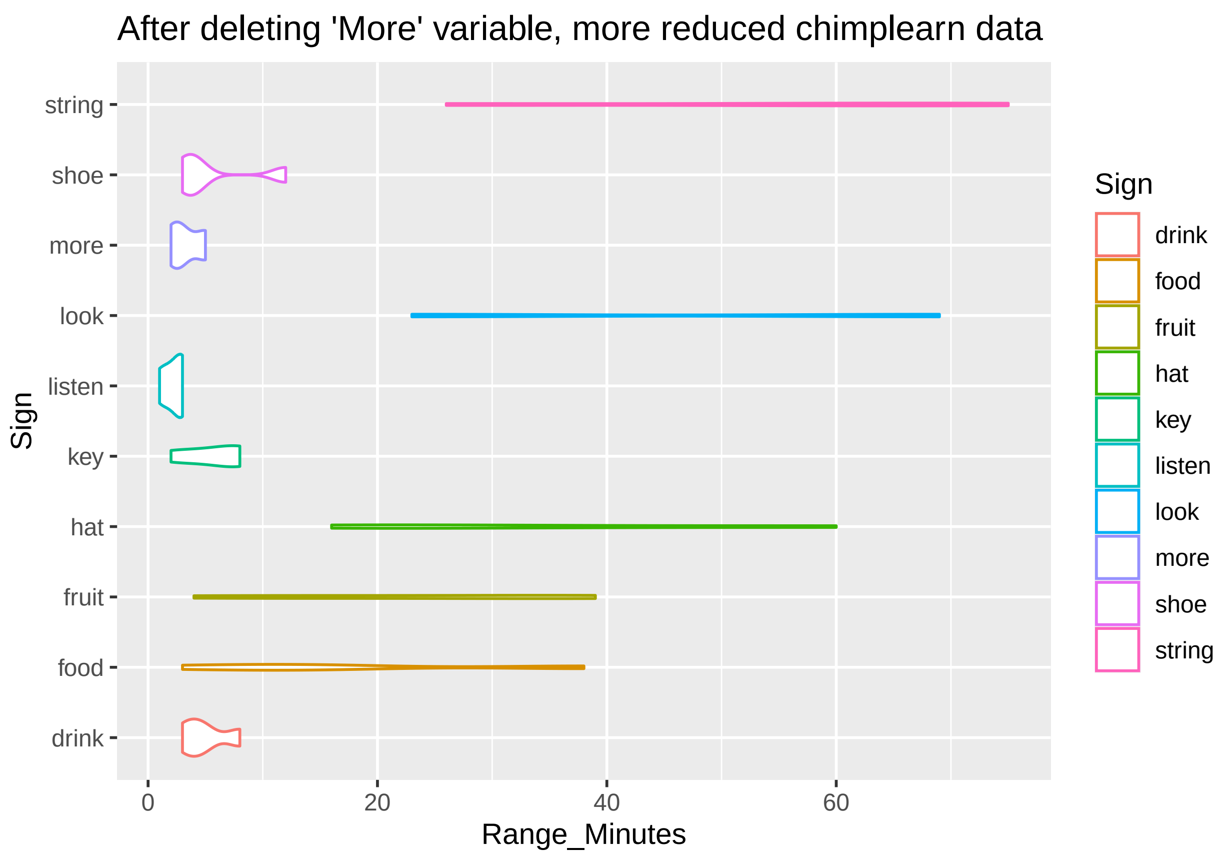

```r
plot(aov, 2)
```




```
chimp_reduced2 <- subset(chimp_reduced,  Chimp != "Bruno" |  Sign != "food")
chimp_reduced2 <- subset(chimp_reduced,  Chimp != "Bruno" |  Sign != "more")
chimp_reduced2 <- as_tibble(chimp_reduced2)
head(chimp_reduced2)
```

```
## # A tibble: 6 x 5
##   Minutes Chimp Sign   Order Range_Minutes
##     <int> <chr> <chr>  <int>         <int>
## 1      12 Booee listen     8             3
## 2      15 Booee drink      4             3
## 3      14 Booee shoe       2             3
## 4      10 Booee key        7             2
## 5      10 Booee more       5             2
## 6      80 Booee food      10            16
```

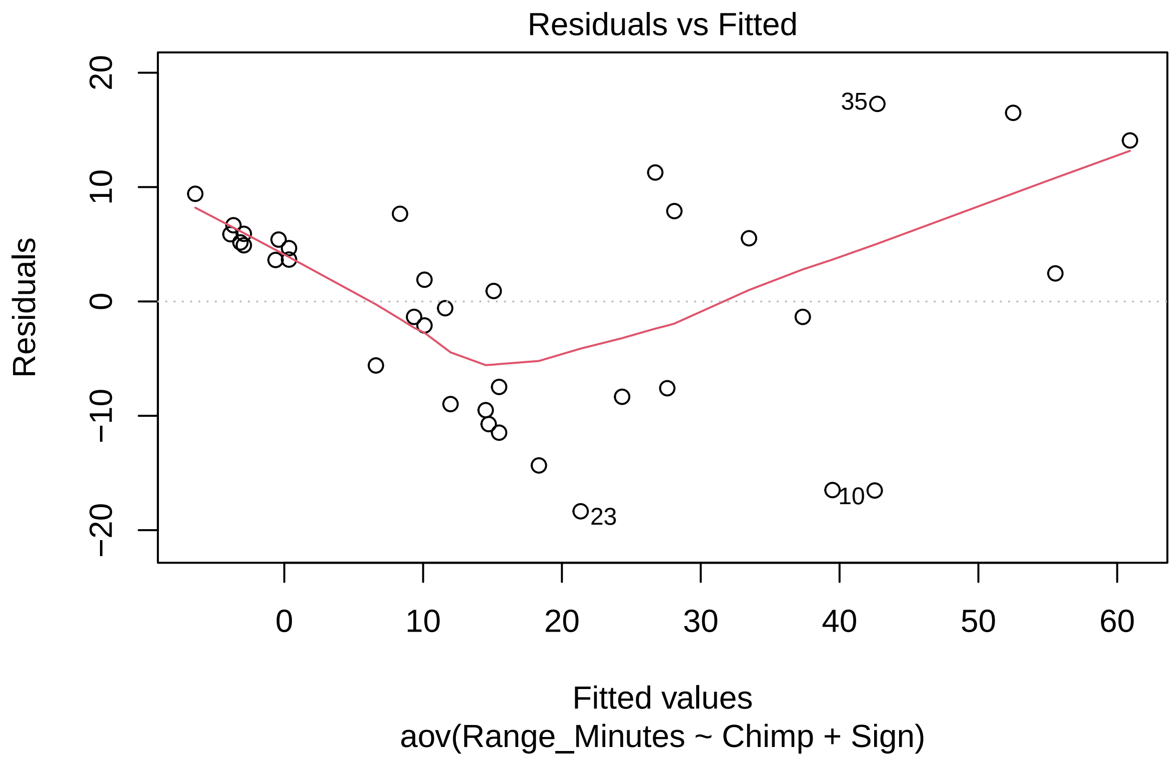
```
ggplot(chimp_reduced2, aes(x = Sign, y = Range_Minutes, color = Sign)) +
    coord_flip() + ggtitle("After deleting 'More' variable, more reduced chimplearn data") + geom_violi
```



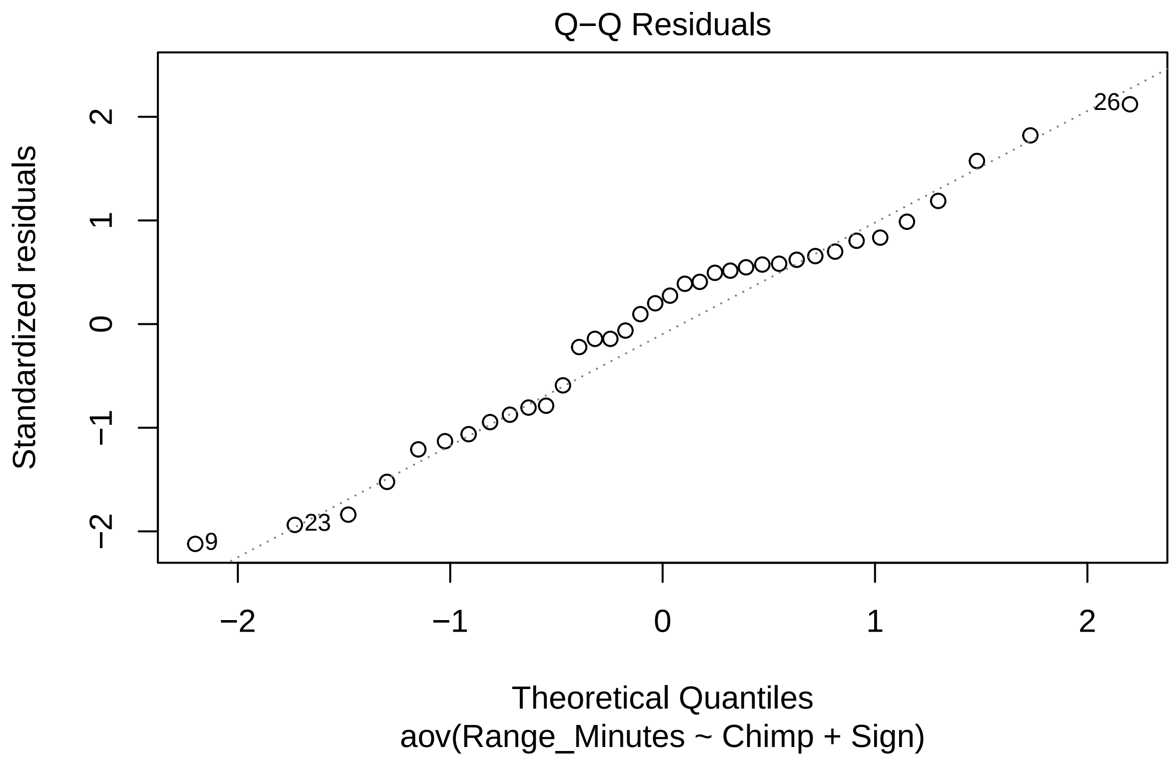

```
aov2 <- aov(Range_Minutes ~ Chimp + Sign, data = chimp_reduced2)
summary(aov2)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Chimp        3   2632   877.3   6.474 0.00245 **
## Sign         9   9134  1014.9   7.489 4.72e-05 ***
## Residuals   23   3117   135.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(aov2, 1)
```


Residuals vs Fitted
Residuals
20
10
0
−10
−20
35
23
10
0
10
20
30
40
50
60
Fitted values
aov(Range_Minutes ~ Chimp + Sign)

```
plot(aov2, 2)
```



From above p-values for each factor Chimp and Sign, we know that Minutes to learn each sign is significantly effected by Sign, not by Chimp.

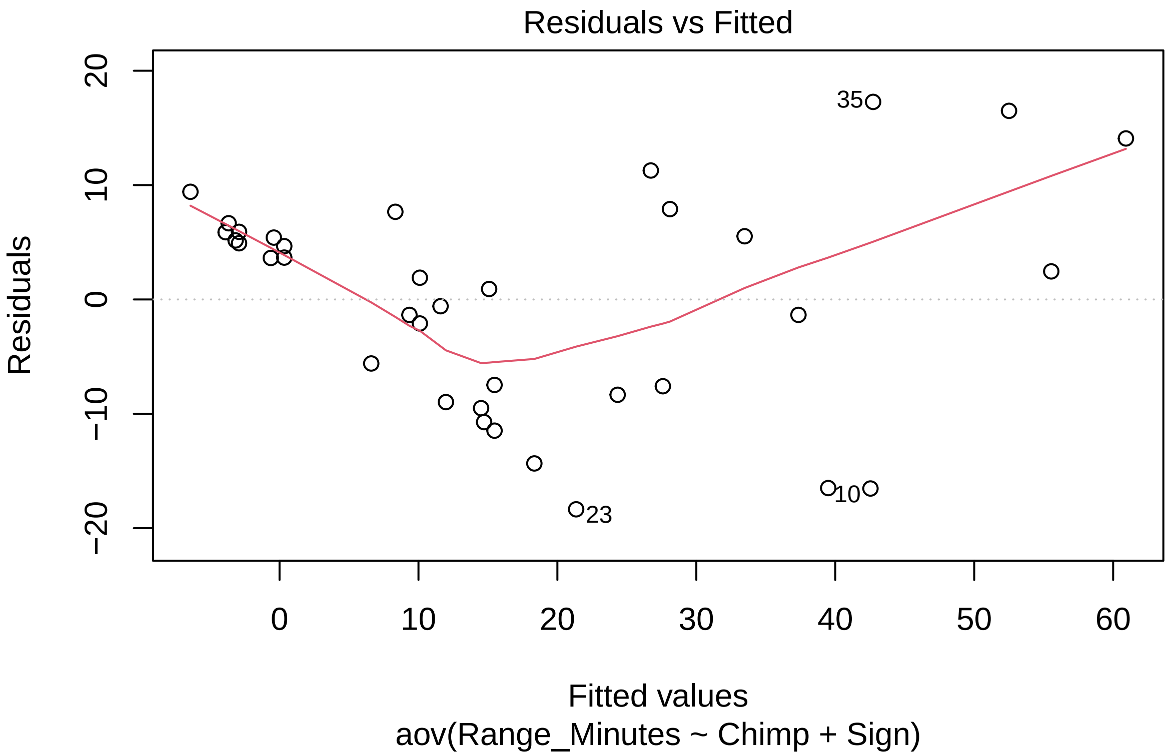
As a result the above chimplearn boxplot indicates ranking of the difficulty of the sign(word) to learn.

##<4>Test validity for the ANOVA Assumption For the ANOVA test, we assume data distribution is normal and the variance for all groups are homogeneous. By using the following plots, we can diagnose these to check.

###1. Check the homogeneity of variance assumption

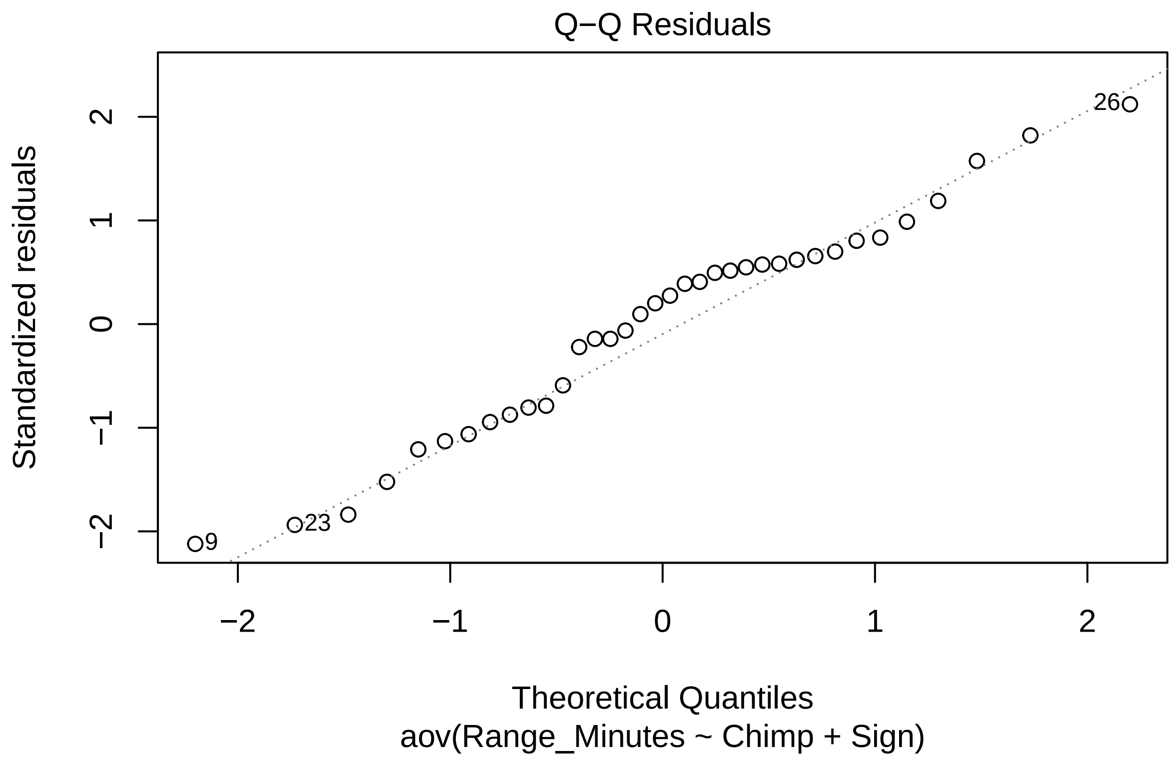
To check the homogeneity of variances, we use residuals versus fits plot. In this plot, residuals and fitted values are not related. This implies assumption for the homogeneity of variances is strong.

```
#Homogeneity of variances
plot(aov2, 1)
```



###2. Check the normal distribution of our data

```
#Normal distribution
plot(aov2, 2)
```



The above Normality plot of the residuals indicates the graph of quantiles ofthe residuals versus quantiles of the normal distribution. They have 45-degree reference line and thus this plot verify the assumption that the residuals has normal distribution. As the plot follow more the straight line, the normal probability plot of the residuals show stronger normality.

```
aov3 <- aov(log(Minutes) ~ Chimp + Sign, data = chimp_reduced2)
#Compare the mean of multiple groups using ANOVA test
aov4 <- chimp_reduced2 %>% anova_test(Minutes ~ Chimp + Sign)
aov4
```

```
## ANOVA Table (type II tests)
##
##   Effect DFn DFd     F        p p<.05   ges
## 1  Chimp   3  23 4.835 9.00e-03     * 0.387
## 2   Sign   9  23 7.500 4.67e-05     * 0.746
```

```
aov5 <- chimp_reduced2 %>% anova_test(log(Minutes) ~ Chimp + Sign)
aov5
```

```
## ANOVA Table (type II tests)
##
##   Effect DFn DFd     F        p p<.05   ges
## 1  Chimp   3  23 2.570 7.90e-02       0.251
## 2   Sign   9  23 8.974 1.11e-05     * 0.778
```

```r
aov6 <- chimplearn %>% anova_test(Minutes ~ Chimp + Sign)
aov6
```

```
## ANOVA Table (type II tests)
##
##   Effect DFn DFd     F        p p<.05   ges
## 1  Chimp   3  27 3.380 0.033000     * 0.273
## 2   Sign   9  27 5.947 0.000141     * 0.665
```

```r
aov7 <- chimplearn %>% anova_test(log(Minutes) ~ Chimp + Sign)
aov7
```

```
## ANOVA Table (type II tests)
##
##   Effect DFn DFd     F       p p<.05   ges
## 1  Chimp   3  27 2.719 6.4e-02       0.232
## 2   Sign   9  27 7.765 1.5e-05     * 0.721
```

```r
##########As we can see below, after deleting variable, we can see more clear evidence on that the line
```

```r
pairwise<- pairwise_t_test(Minutes ~ Sign, p.adjust.method = "bonferroni", data = chimp_reduced2)
print(pairwise, n = 100)              # Print 20 rows of tibble
```

```
## # A tibble: 45 x 9
##    .y.     group1 group2    n1    n2         p p.signif   p.adj p.adj.signif
##  * <chr>   <chr>  <chr>  <int> <int>     <dbl> <chr>      <dbl> <chr>
##  1 Minutes drink  food       4     4 0.225     ns       1      ns
##  2 Minutes drink  fruit      4     4 0.0662    ns       1      ns
##  3 Minutes food   fruit      4     4 0.506     ns       1      ns
##  4 Minutes drink  hat        4     4 0.00884   **       0.398  ns
##  5 Minutes food   hat        4     4 0.124     ns       1      ns
##  6 Minutes fruit  hat        4     4 0.37      ns       1      ns
##  7 Minutes drink  key        4     4 0.916     ns       1      ns
##  8 Minutes food   key        4     4 0.266     ns       1      ns
##  9 Minutes fruit  key        4     4 0.0817    ns       1      ns
## 10 Minutes hat    key        4     4 0.0114    *        0.511  ns
## 11 Minutes drink  listen     4     4 0.782     ns       1      ns
## 12 Minutes food   listen     4     4 0.14      ns       1      ns
## 13 Minutes fruit  listen     4     4 0.0372    *        1      ns
## 14 Minutes hat    listen     4     4 0.0045    **       0.202  ns
## 15 Minutes key    listen     4     4 0.703     ns       1      ns
## 16 Minutes drink  look       4     2 0.00207   **       0.0931 ns
## 17 Minutes food   look       4     2 0.0235    *        1      ns
## 18 Minutes fruit  look       4     2 0.0749    ns       1      ns
## 19 Minutes hat    look       4     2 0.277     ns       1      ns
## 20 Minutes key    look       4     2 0.00258   **       0.116  ns
## 21 Minutes listen look       4     2 0.00116   **       0.0521 ns
## 22 Minutes drink  more       4     3 0.894     ns       1      ns
## 23 Minutes food   more       4     3 0.21      ns       1      ns
## 24 Minutes fruit  more       4     3 0.0672    ns       1      ns
## 25 Minutes hat    more       4     3 0.0106    *        0.475  ns
## 26 Minutes key    more       4     3 0.817     ns       1      ns
```

```
## 27 Minutes listen more       4     3 0.903     ns       1       ns
## 28 Minutes look   more       2     3 0.00242   **       0.109   ns
## 29 Minutes drink  shoe       4     4 0.928     ns       1       ns
## 30 Minutes food   shoe       4     4 0.26      ns       1       ns
## 31 Minutes fruit  shoe       4     4 0.0793    ns       1       ns
## 32 Minutes hat    shoe       4     4 0.011     *        0.493   ns
## 33 Minutes key    shoe       4     4 0.988     ns       1       ns
## 34 Minutes listen shoe       4     4 0.714     ns       1       ns
## 35 Minutes look   shoe       2     4 0.0025    **       0.112   ns
## 36 Minutes more   shoe       3     4 0.828     ns       1       ns
## 37 Minutes drink  string     4     3 0.000128  ***      0.00576 **
## 38 Minutes food   string     4     3 0.00252   **       0.114   ns
## 39 Minutes fruit  string     4     3 0.0115    *        0.519   ns
## 40 Minutes hat    string     4     3 0.0724    ns       1       ns
## 41 Minutes key    string     4     3 0.000166  ***      0.00747 **
## 42 Minutes listen string     4     3 0.0000647 ****     0.00291 **
## 43 Minutes look   string     2     3 0.612     ns       1       ns
## 44 Minutes more   string     3     3 0.000197  ***      0.00888 **
## 45 Minutes shoe   string     4     3 0.00016   ***      0.0072  **
```