

```
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 errors
```

```
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:dplyr':
##
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarize
##
## The following object is masked from 'package:purrr':
##
##   compact
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

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

```
## corrplot 0.92 loaded
```

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

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

```
library(tidyverse)
```

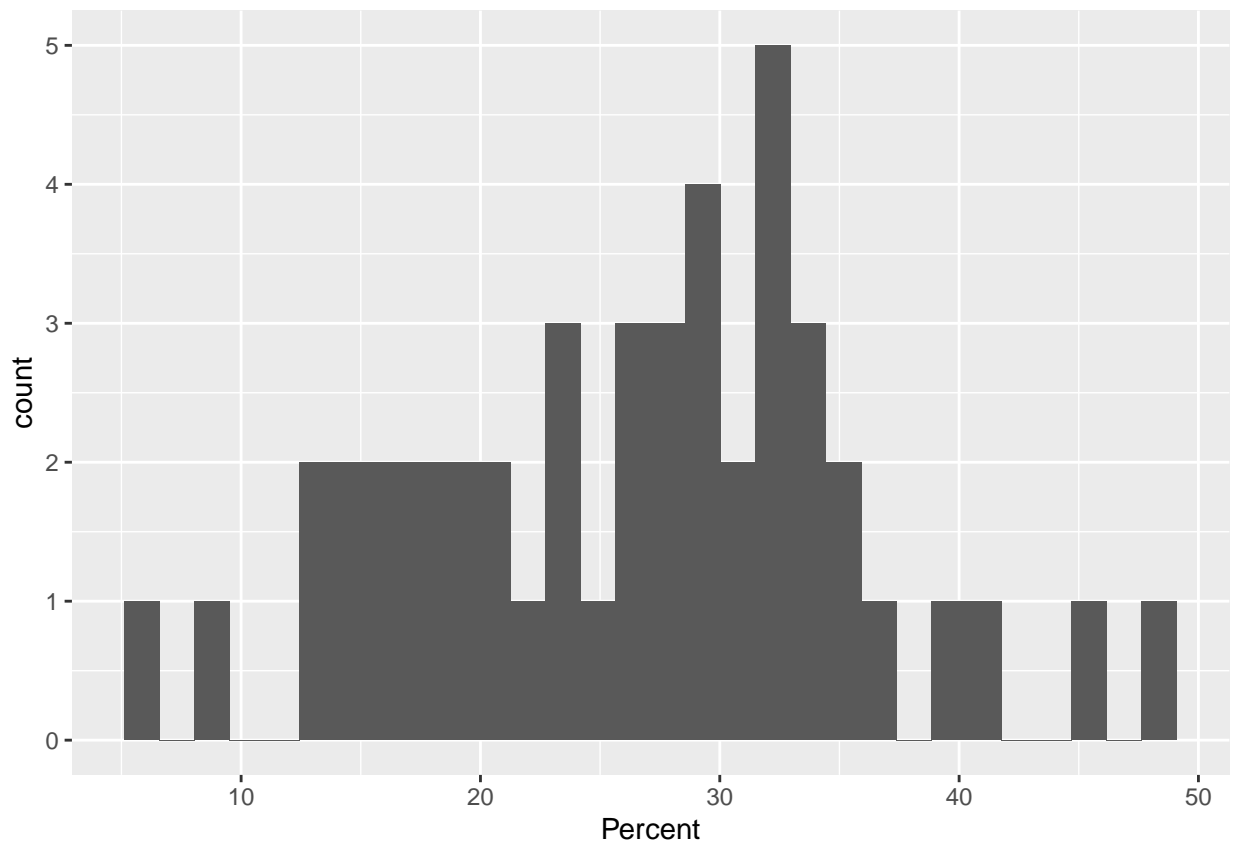
```
jury <- read.csv("jury.csv")
head(jury)
```

```
##   Percent   Judge
## 1     6.4 Spock's
## 2     8.7 Spock's
## 3    13.3 Spock's
## 4    13.6 Spock's
## 5    15.0 Spock's
## 6    15.2 Spock's
```

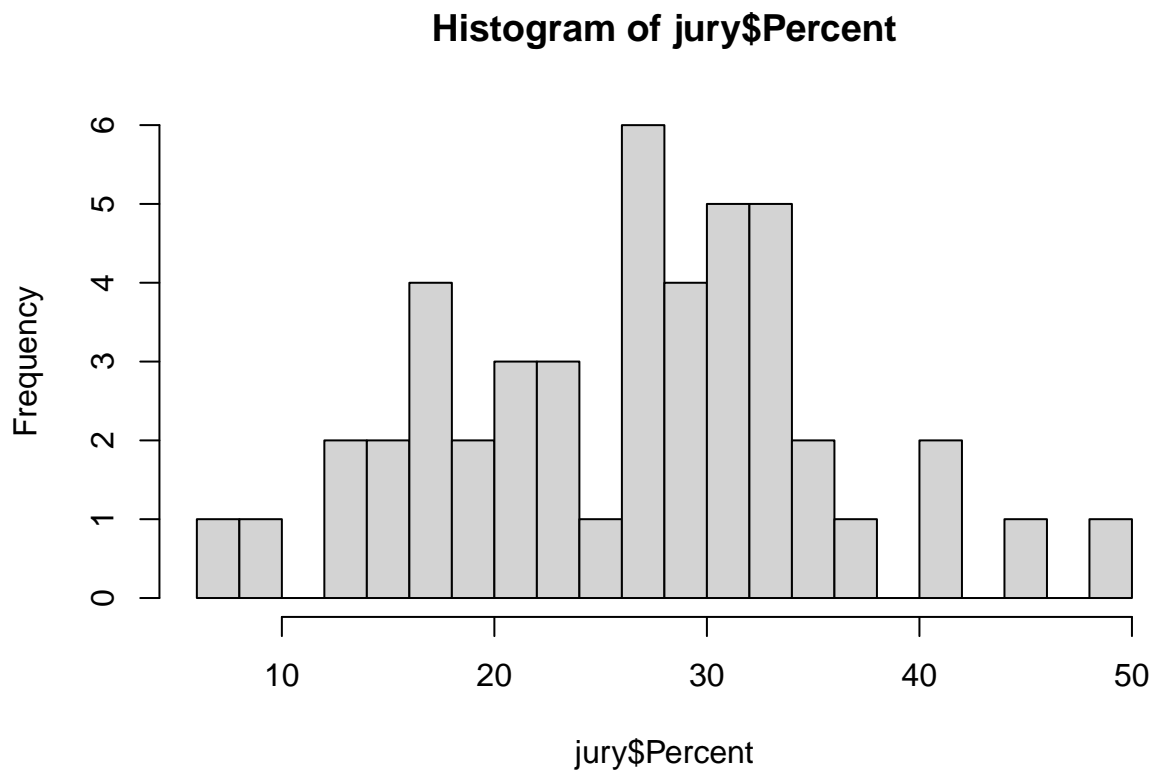
```
##1. EDA ###(1)Histogram and simple graph
```

```
ggplot(jury, aes(Percent)) + geom_histogram()
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



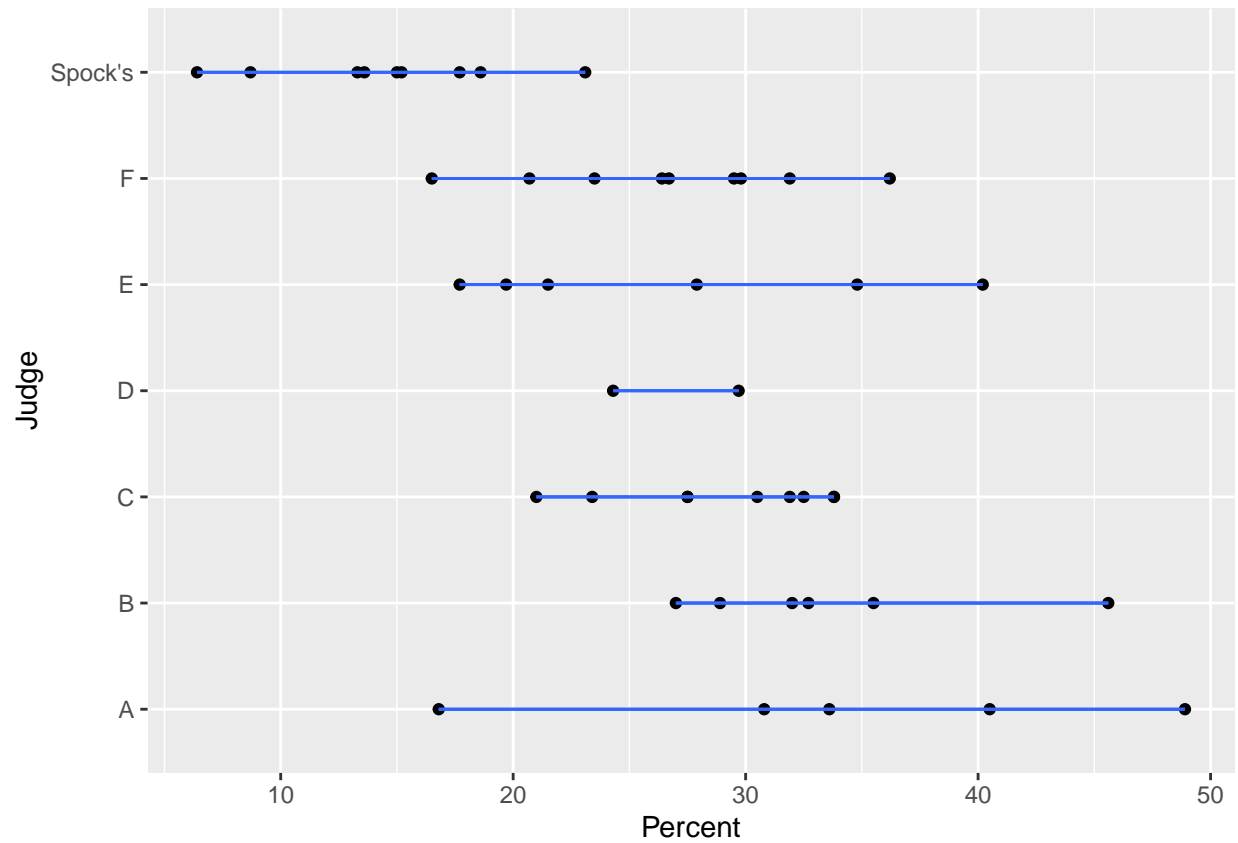
```
hist(jury$Percent, breaks = 30)
```



```
ggplot(jury, aes(Percent, Judge)) +  
  geom_point() +  
  stat_quantile()
```

```
## Smoothing formula not specified. Using: y ~ x  
## Smoothing formula not specified. Using: y ~ x  
## Smoothing formula not specified. Using: y ~ x  
## Smoothing formula not specified. Using: y ~ x  
## Smoothing formula not specified. Using: y ~ x  
## Smoothing formula not specified. Using: y ~ x  
## Smoothing formula not specified. Using: y ~ x
```

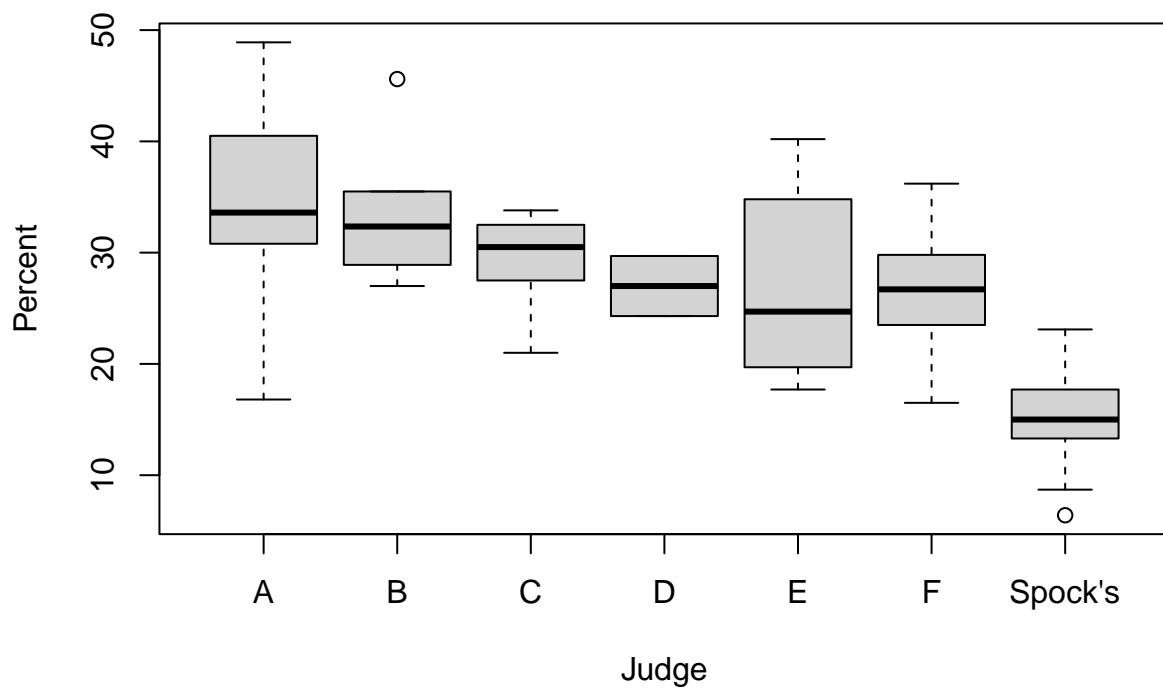
```
## Warning in rq.fit.br(wx, wy, tau = tau, ...): Solution may be nonunique
```



###(2)Box Plot »>Here, I wondered that the Spock's Judge is different with the other specific group with meaningful statistical indicators.

- Box Plot for Each Groups

```
boxplot(Percent ~ Judge, data = jury )
```



-Box Plot for Spock's vs Non-Spock's

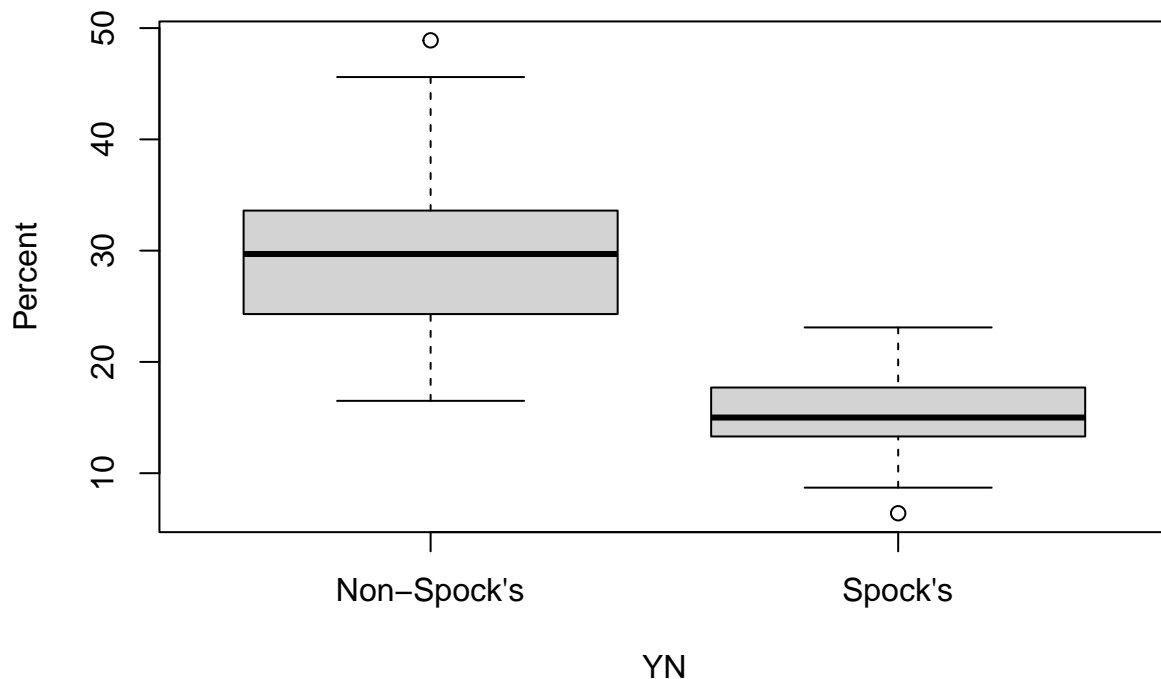
```
jury$YN <- ifelse(jury$Judge == "Spock's", "Spock's", "Non-Spock's" )
```

```
jury
```

##	Percent	Judge	YN
## 1	6.4	Spock's	Spock's
## 2	8.7	Spock's	Spock's
## 3	13.3	Spock's	Spock's
## 4	13.6	Spock's	Spock's
## 5	15.0	Spock's	Spock's
## 6	15.2	Spock's	Spock's
## 7	17.7	Spock's	Spock's
## 8	18.6	Spock's	Spock's
## 9	23.1	Spock's	Spock's
## 10	16.8	A Non-Spock's	
## 11	30.8	A Non-Spock's	
## 12	33.6	A Non-Spock's	
## 13	40.5	A Non-Spock's	
## 14	48.9	A Non-Spock's	
## 15	27.0	B Non-Spock's	
## 16	28.9	B Non-Spock's	
## 17	32.0	B Non-Spock's	
## 18	32.7	B Non-Spock's	
## 19	35.5	B Non-Spock's	

## 20	45.6	B Non-Spock's
## 21	21.0	C Non-Spock's
## 22	23.4	C Non-Spock's
## 23	27.5	C Non-Spock's
## 24	27.5	C Non-Spock's
## 25	30.5	C Non-Spock's
## 26	31.9	C Non-Spock's
## 27	32.5	C Non-Spock's
## 28	33.8	C Non-Spock's
## 29	33.8	C Non-Spock's
## 30	24.3	D Non-Spock's
## 31	29.7	D Non-Spock's
## 32	17.7	E Non-Spock's
## 33	19.7	E Non-Spock's
## 34	21.5	E Non-Spock's
## 35	27.9	E Non-Spock's
## 36	34.8	E Non-Spock's
## 37	40.2	E Non-Spock's
## 38	16.5	F Non-Spock's
## 39	20.7	F Non-Spock's
## 40	23.5	F Non-Spock's
## 41	26.4	F Non-Spock's
## 42	26.7	F Non-Spock's
## 43	29.5	F Non-Spock's
## 44	29.8	F Non-Spock's
## 45	31.9	F Non-Spock's
## 46	36.2	F Non-Spock's

```
boxplot(Percent ~ YN, data = jury )
```



##2. Apply one-way ANOVA test

###(1) Definition of ANOVA test and its assumption Steps for using ANOVA STEP1. Compute the common variance (= Statistically, this is variance within samples or residual variance.) STEP2. Compute the variance between sample means by using variance between sample means. STEP3. Now, compute F-statistics as the ratio of (variance between sample)/(variance within samples).

Assumptions are as follows. - The all Percents of each group are observed independently and randomly. - The all Percents of each group are normally distributed. - These values of percents have a common variance.

###(2) Computing one-way ANOVA test The R function `summary.aov()` processes the analysis of variance model.

```
anova_all <- lm(Percent~Judge, jury)
anova_reduced <- lm(Percent~YN, jury)
#anova to compare Spock's vs Non-spock's
one.way <- aov(Percent ~ YN, data = jury)
summary(one.way)
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## YN         1   1601  1600.6    32.15 1.03e-06 ***
## Residuals 44    2191    49.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
one.way2 <- aov(Percent ~ Judge, data = jury)
summary(one.way2)
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## Judge      6   1927   321.2    6.718 6.1e-05 ***
## Residuals  39   1864    47.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(anova_reduced, anova_all)
```

```
## Analysis of Variance Table
##
## Model 1: Percent ~ YN
## Model 2: Percent ~ Judge
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1      44 2190.9
## 2      39 1864.5  5    326.46 1.3658 0.2582
```

```
#t.test(subset(dt, Y/N, percent), percent.others)
#anova()
```

```
# Computation in the analysis of variance
res.aov <- aov(Percent ~ Judge, data = jury)
# Summary for the analysis
summary(res.aov)
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## Judge      6   1927   321.2    6.718 6.1e-05 ***
## Residuals  39   1864    47.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Interpret the result of one-way ANOVA tests As the p-value is less than the significance level 0.05, we can conclude that there are significant differences between the groups highlighted with “*” in the model summary.

###(3)Pairwise-comparison between the means of groups By taking one-way ANOVA test, the p-value differentiates the group means. However, we cannot get specific pairs of groups, which are different.

Therefore, by using the Tukey multiple pairwise-comparison, we can identify the multiple pairwise-comparison by observing p-value adjustment with mean difference between specific pairs of groups with statistically significance.

- Tukey multiple pairwise-comparisons

As the above explanation, by using R function TukeyHSD(), we can compute Tukey HSD (Tukey Honest Significant Differences) to get multiple pairwise-comparison between the means of groups. This function TukeyHD() computes the fitted ANOVA as an argument.

```
TukeyHSD(res.aov)
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
```



```
##
## Fit: aov(formula = Percent ~ Judge, data = jury)
##
## $Judge
##           diff      lwr      upr      p adj
## B-A      -0.5033333 -13.51242 12.505755 0.9999997
## C-A      -5.0200000 -17.00309  6.963092 0.8470097
## D-A      -7.1200000 -25.09464 10.854638 0.8777485
## E-A      -7.1533333 -20.16242  5.855755 0.6146238
## F-A      -7.3200000 -19.30309  4.663092 0.4936379
## Spock's-A -19.4977778 -31.48087 -7.514686 0.0001992
## C-B       -4.5166667 -15.83962  6.806291 0.8742030
## D-B       -6.6166667 -24.15812 10.924784 0.9003280
## E-B       -6.6500000 -19.05368  5.753679 0.6418002
## F-B       -6.8166667 -18.13962  4.506291 0.5109582
## Spock's-B -18.9944444 -30.31740 -7.671487 0.0001224
## D-C       -2.1000000 -18.89466 14.694661 0.9996956
## E-C       -2.1333333 -13.45629  9.189625 0.9968973
## F-C       -2.3000000 -12.42756  7.827561 0.9914731
## Spock's-C -14.4777778 -24.60534 -4.350216 0.0012936
## E-D       -0.0333333 -17.57478 17.508118 1.0000000
## F-D       -0.2000000 -16.99466 16.594661 1.0000000
## Spock's-D -12.3777778 -29.17244  4.416883 0.2744263
## F-E       -0.1666667 -11.48962 11.156291 1.0000000
## Spock's-E -12.3444444 -23.66740 -1.021487 0.0248789
## Spock's-F -12.1777778 -22.30534 -2.050216 0.0098340
```

-diff: difference between means of the two groups -lwr, upr: the lower and the upper end point of the confidence interval at 95% (default) -p adj: p-value after adjustment for the multiple comparisons.

In the above output, the difference between Spock's and A is the biggest with an adjusted p-value of 0.0001992.

###(4)Multiple comparisons using multcomp package It's possible to use the function `glht()` [in multcomp package] to perform multiple comparison procedures for an ANOVA. `glht` stands for general linear hypothesis tests. The simplified format is as follow:

`glht(model, lincft)` model: a fitted model, for example an object returned by `aov()`. `lincft()`: a specification of the linear hypotheses to be tested. Multiple comparisons in ANOVA models are specified by objects returned from the function `mcp()`. Use `glht()` to perform multiple pairwise-comparisons for a one-way ANOVA:

```
#library(multcomp)
#glht(model, lincft)

#summary(glht(res.aov, lincft = mcp(group = "Tukey")))
```

###(4)Pairwise t-test With the function `pairwise.t.test()`, we can also compute pairwise comparisons between groups of Judge with corrections for multiple testing.

```
pairwise.t.test(jury$Percent, jury$Judge,
                p.adjust.method = "BH")
```

```
##
## Pairwise comparisons using t tests with pooled SD
```

```
##
## data: jury$Percent and jury$Judge
##
##      A      B      C      D      E      F
## B    0.99532 -      -      -      -      -
## C    0.36470 0.36470 -      -      -      -
## D    0.36470 0.37244 0.86439 -      -      -
## E    0.21788 0.21788 0.73715 0.99532 -      -
## F    0.18091 0.18091 0.67844 0.99532 0.99532 -
## Spock's 0.00011 0.00011 0.00050 0.09629 0.00682 0.00314
##
## P value adjustment method: BH
```

The above is a table of p-values on each of the pairwise comparisons. With the the Benjamini-Hochberg method, p-values have been adjusted.

####(5)Validity of the test

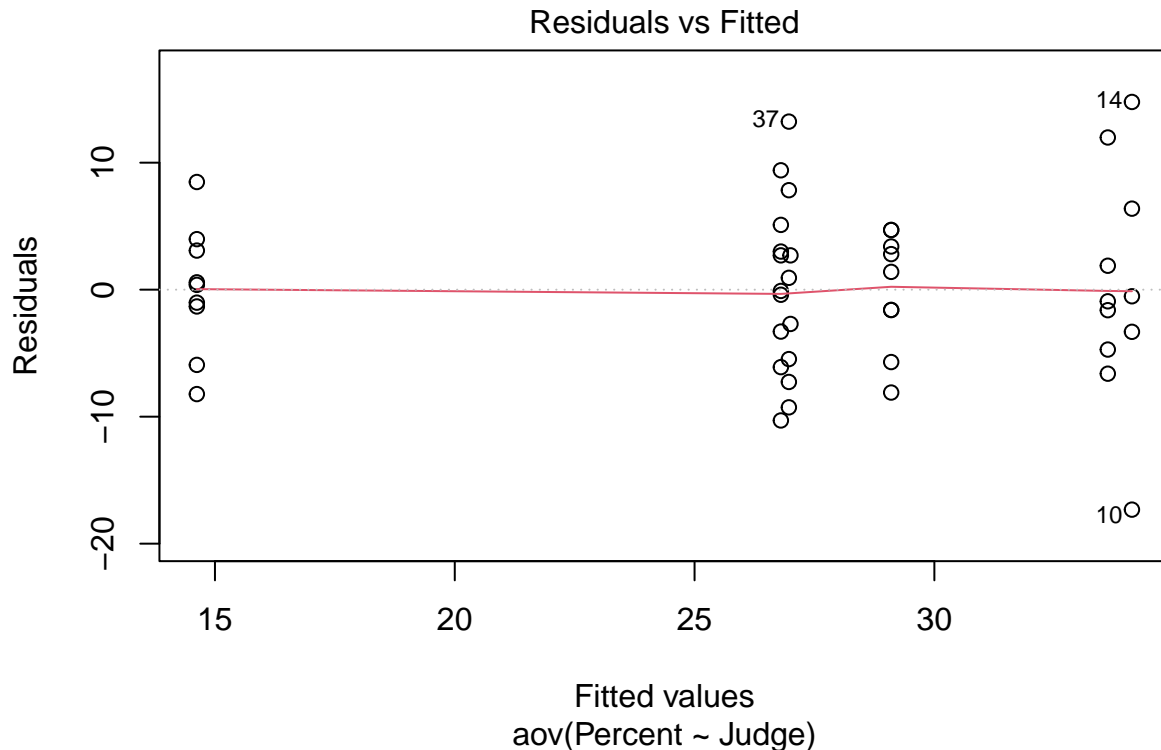
- Check ANOVA assumptions

From our assumptions before, we will check the normal distribution of the variance of groups and homogeneity of the variance through the groups, via drawing the diagnostic plots.

- Check the homogeneity of variance for assumption The residuals versus fits plot can be used to check the homogeneity of variances.

In the plot below, which is the residuals versus fits, there is no dependency between residuals and fitted values (=the mean of each groups). So the result as the plot is in nice level. From this, it is possible to assume the homogeneity of variances.

```
# 1. Homogeneity of variances
plot(res.aov, 1)
```



We've found some outlier data points at positions 37, 14, and 10, which can mess up the normality and homogeneity of variances. To use our assumptions for testing, we might want to eliminate these points.

We can also use two tests, Bartlett's or Levene's, to see if the data has consistent variances. We can use Levene's test because it's not picky about normality of variances. To do this, we'll use a function called "leveneTest()" from the "car" package.

```
library(car)
```

```
## Loading required package: carData
```

```
##
```

```
## Attaching package: 'car'
```

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

```
##
```

```
## recode
```

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

```
##
```

```
## some
```

```
leveneTest(Percent ~ Judge, data = jury)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group 6  1.2625 0.2969
##      39
```

From the output above we can see that the p-value is not less than the significance level of 0.05. This means that there is no evidence to suggest that the variance across groups is statistically significantly different. Therefore, we can assume the homogeneity of variances in the different treatment groups.

###(6)Relaxing the homogeneity of variance assumption: In our example, the Levene test is not significant; the assumption of equal variances for all groups, required by the classical one-way ANOVA test, turned out to be fine.

How do we save our ANOVA test in a situation where the homogeneity of variance assumption is violated? An alternative procedure, the Welch one-way test, that does not require that assumption, has been implemented in the function “oneway.test()”.

–ANOVA test with no assumption of equal variances

```
oneway.test(Percent ~ Judge, data = jury)
```

```
##
## One-way analysis of means (not assuming equal variances)
##
## data: Percent and Judge
## F = 7.4923, num df = 6.0000, denom df = 9.9246, p-value = 0.003093
```

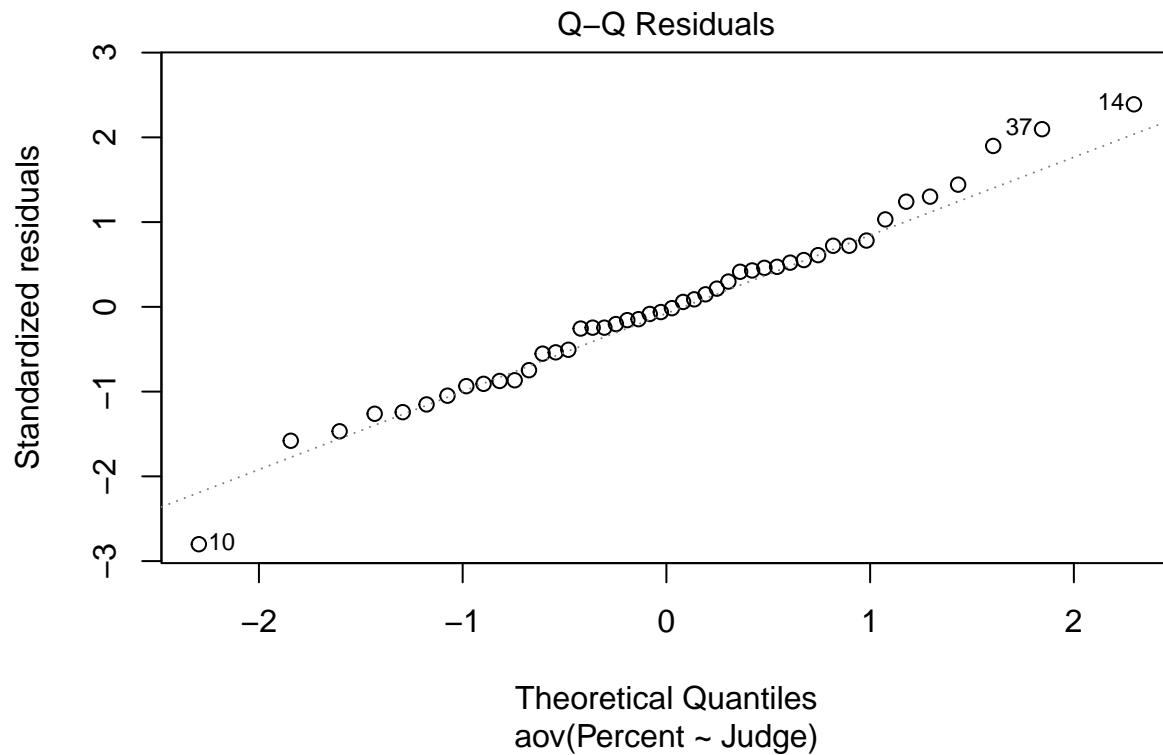
–Pairwise t-tests with no assumption of equal variances

```
pairwise.t.test(jury$Percent, jury$Judge,
                p.adjust.method = "BH", pool.sd = FALSE)
```

```
##
## Pairwise comparisons using t tests with non-pooled SD
##
## data: jury$Percent and jury$Judge
##
##      A      B      C      D      E      F
## B    0.9945 -      -      -      -      -
## C    0.5746 0.3807 -      -      -      -
## D    0.4922 0.3807 0.7529 -      -      -
## E    0.4922 0.3807 0.7529 0.9945 -      -
## F    0.4846 0.2056 0.5611 0.9945 0.9945 -
## Spock's 0.0788 0.0019 0.0002 0.2056 0.0788 0.0019
##
## P value adjustment method: BH
```

###(7)Check the normality assumption To assess the assumption of normality, we examine a normality plot of residuals. In the plot below, residuals’ quantiles are compared to those of the normal distribution, with a 45-degree reference line included. This plot helps verify the assumption that residuals are normally distributed, indicated by an approximately straight line.

```
# 2. Normality
plot(res.aov, 2)
```



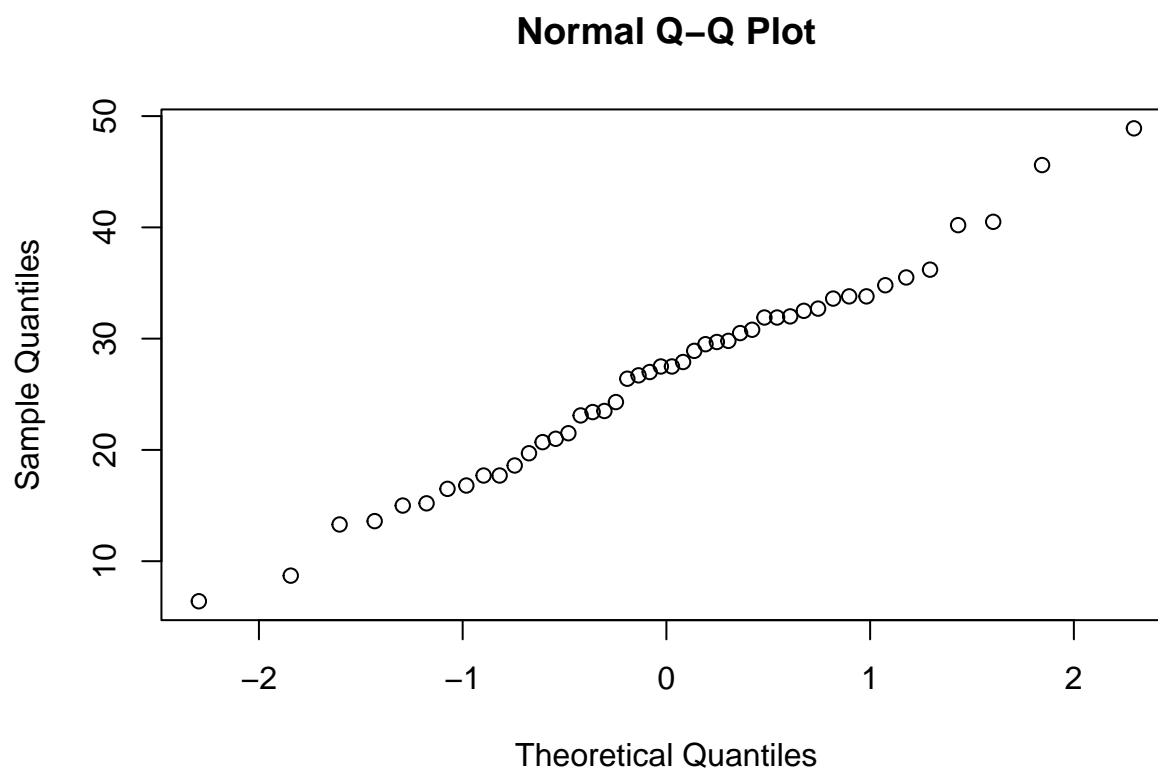
We can assume normality as all the points fall approximately along this reference line.

The conclusion above is supported by the Shapiro-Wilk test on the ANOVA residuals ($W = 0.96$, $p = 0.6$), which indicate that normality holds.

```
# Extract the residuals
aov_residuais <- residuals(object = res.aov )
# Run Shapiro-Wilk test
shapiro.test(x = aov_residuais )
```

```
##
##  Shapiro-Wilk normality test
##
## data:  aov_residuais
## W = 0.98817, p-value = 0.9167
```

```
qqnorm(jury$Percent)
```



```
head(jury)
```

```
##   Percent   Judge    YN
## 1     6.4 Spock's Spock's
## 2     8.7 Spock's Spock's
## 3    13.3 Spock's Spock's
## 4    13.6 Spock's Spock's
## 5    15.0 Spock's Spock's
## 6    15.2 Spock's Spock's
```

```
library(rstatix)
```

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

## The following objects are masked from 'package:plyr':
##
##   desc, mutate

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

```
t.test(jury$Percent ~ jury$YN, var.equal=TRUE)
```

```
##  
## Two Sample t-test  
##  
## data: jury$Percent by jury$YN  
## t = 5.6697, df = 44, p-value = 1.03e-06  
## alternative hypothesis: true difference in means between group Non-Spock's and group Spock's is not 0  
## 95 percent confidence interval:  
## 9.584045 20.155294  
## sample estimates:  
## mean in group Non-Spock's      mean in group Spock's  
## 29.49189 14.62222
```