ANOVA Tutorial

ANOVA in R

This documents includes code for creating a data object in R, displaying summary statistics, creating an informative plot, checking assumptions for ANOVA, and running an Analysis of Variance (ANOVA).

Background

Nicolas Cage is considered the greatest actor of all time by many people in the United States. However, Hollywood scientists think opinions of this amazing actor are related to the culture of individual cities. Thus, to better understand how successful a new movie with Nicolas Cage will be, scientists must understand these spatial patterns. The data below represent a random selection of preference scores by people in three different South Carolina Cities. Preference scores are expressed as a percentage and range from 0 to 100 with 0 indicating low preference (i.e., does not like Nicolas Cage) and 100 indicates high preference (i.e., loves Nicolas Cage).

Hypothesis: If culture influences the opinion of Nicolas Cage, then opinion ratings will be different across cities.

Null Hypothesis: The opinion rating of Nicolas Cage is not different among cities.

Data

The data frame created below contain 15 observations (rows) and 2 variables (columns). Each row represents one individual observation. The first column is opinion rating reported by the individual and the second column is the city the individual is from.

Important points to note. The function used here is "data.frame" which creates a data frame object that is being assigned the name "dat". Everything after "data.frame" are the arguments being passed to the function. The arguments include the column names (rating and city) and the data in each column. Quotes (" ") are used to encapsulate text. This is important so R knows that this column is a categorical variable. The categorical variable will be used as the predictor of rating.

Enter the following code in R:

```
dat = data.frame(row=c(1,2,3,4,5,1,2,3,4,5,1,2,3,4,5),
        rating = c(13,16,8,15,9,
                   29,35,24,27,32,
                   57,59,52,55,60),
        city = as.factor( c("Charleston","Charleston","Charleston","Charleston","Charleston",
                            "Columbia","Columbia","Columbia","Columbia","Columbia",
                            "Florence","Florence","Florence","Florence","Florence")))
```

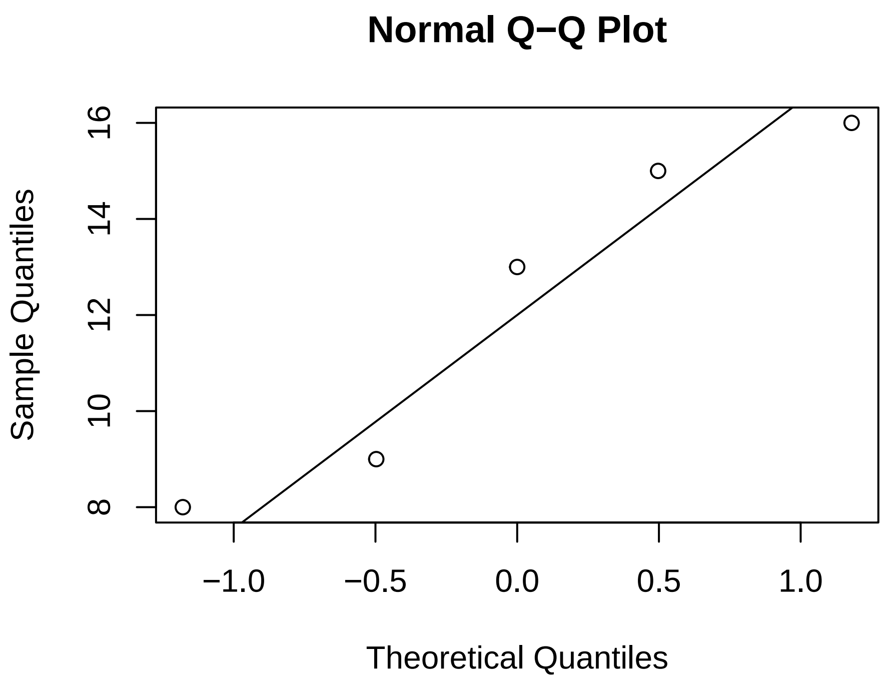
The next line of code prints the data contained in the dat object

```
dat
```

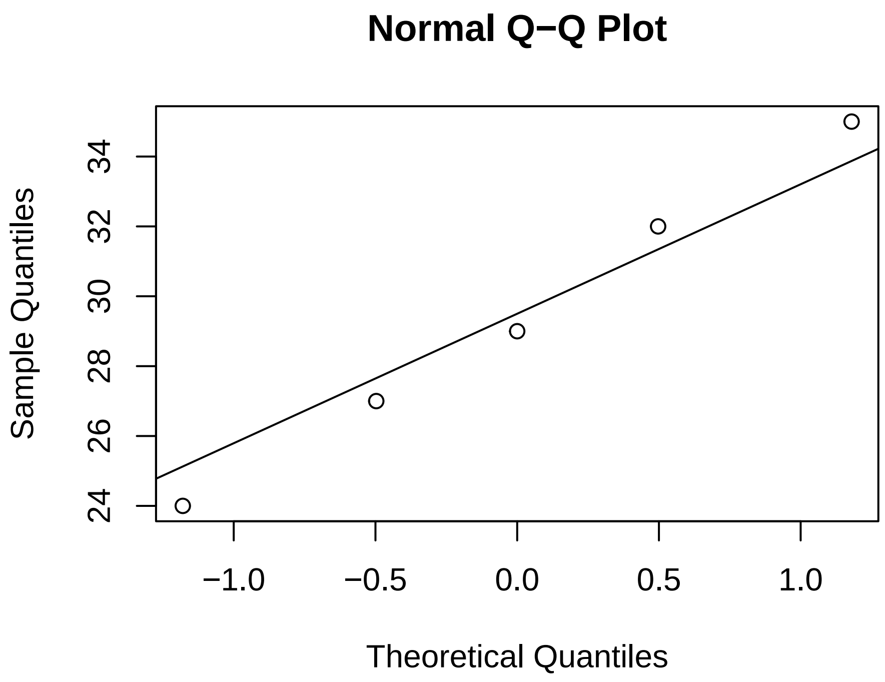
```
##    row rating       city
## 1    1     13 Charleston
## 2    2     16 Charleston
## 3    3      8 Charleston
## 4    4     15 Charleston
## 5    5      9 Charleston
## 6    1     29   Columbia
## 7    2     35   Columbia
## 8    3     24   Columbia
## 9    4     27   Columbia
## 10   5     32   Columbia
## 11   1     57   Florence
## 12   2     59   Florence
## 13   3     52   Florence
## 14   4     55   Florence
## 15   5     60   Florence
```

Check Normality Assumption

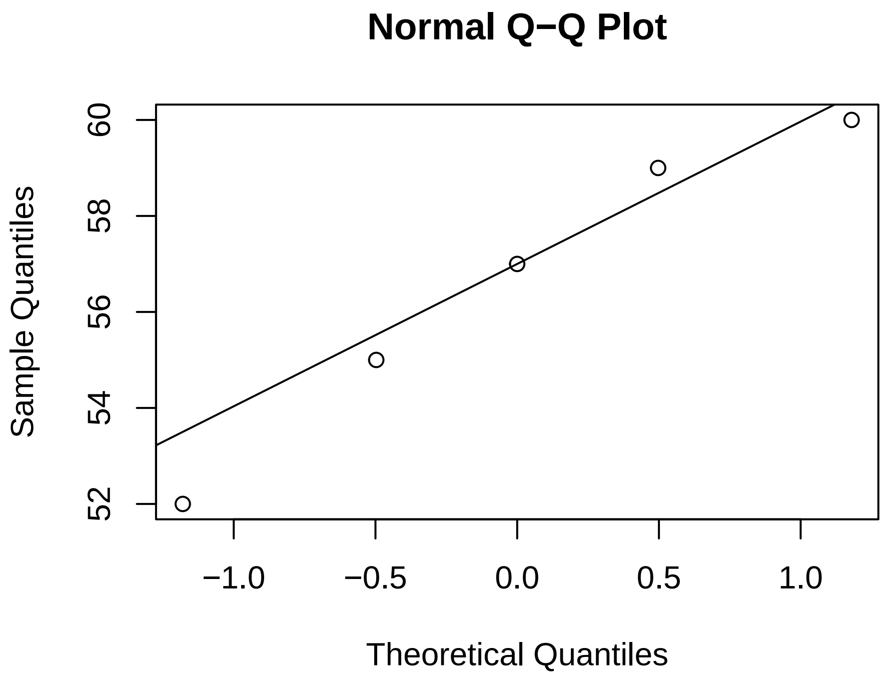
```
#Load required packages. These might need to be installed
library(reshape2)
#Convert from long to wide format
wide_dat = dcast(dat, row~city,value.var="rating")
#Check normality
qqnorm(wide_dat$Charleston)
qqline(wide_dat$Charleston)
```



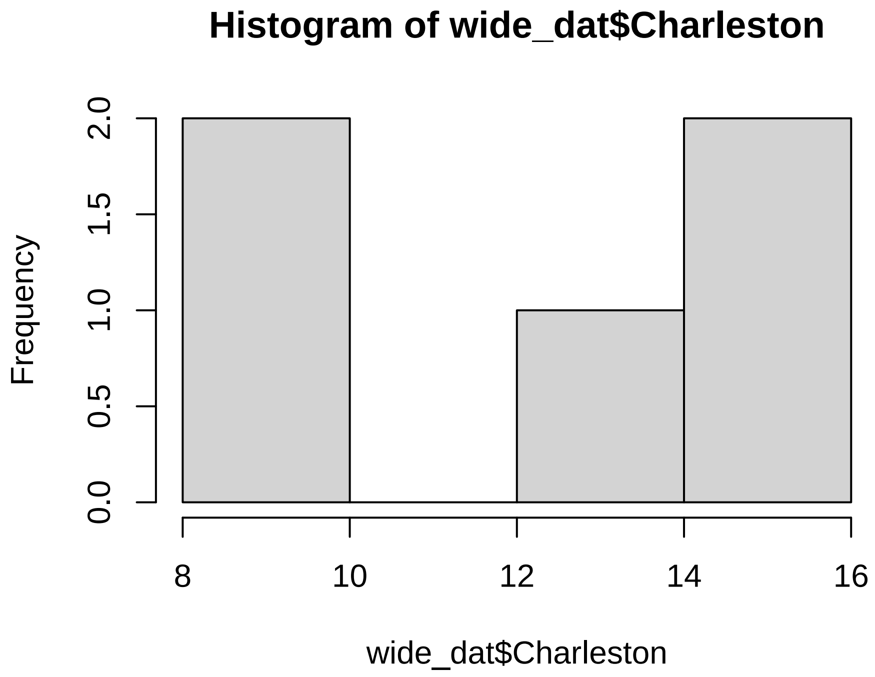
```
qqnorm(wide_dat$Columbia)
qqline(wide_dat$Columbia)
```



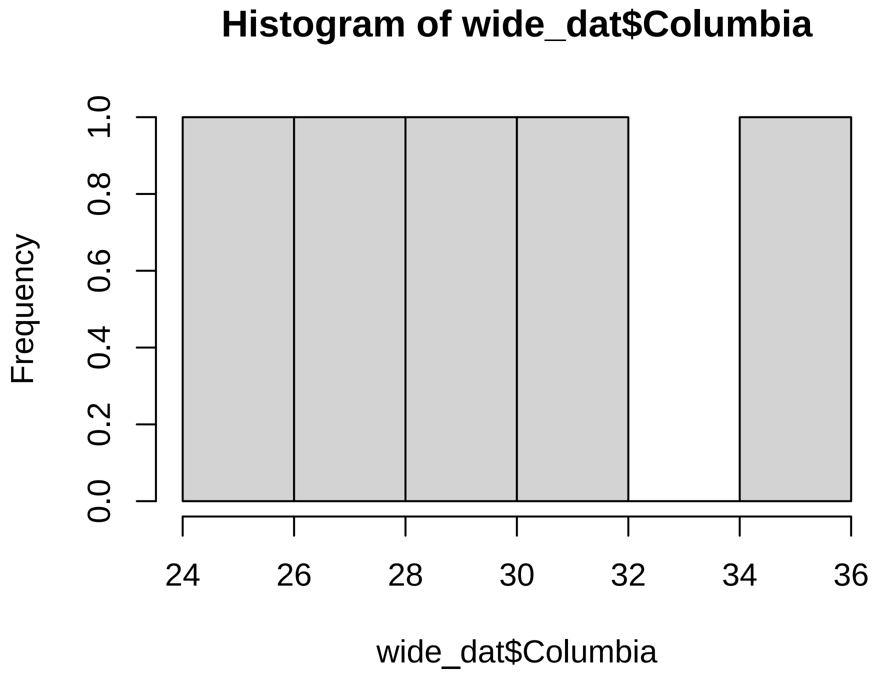
```
qqnorm(wide_dat$Florence)
qqline(wide_dat$Florence)
```



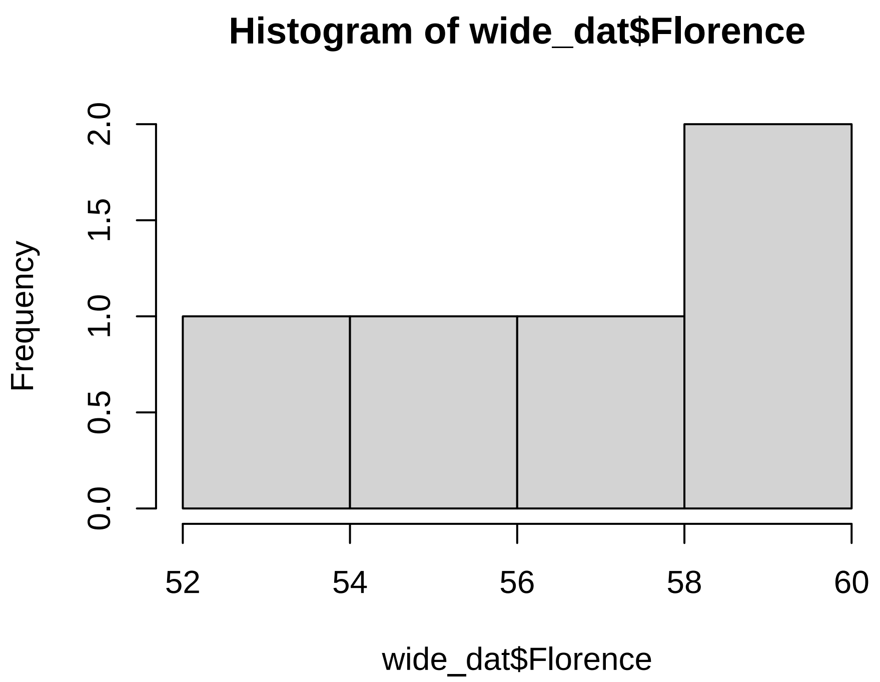
```
#Histogram
hist(wide_dat$Charleston)
```



```
hist(wide_dat$Columbia)
```



```r
hist(wide_dat$Florence)
```



```r
#Shapiro test
shapiro.test(wide_dat$Charleston)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  wide_dat$Charleston
## W = 0.90096, p-value = 0.4152
```

```r
shapiro.test(wide_dat$Columbia)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  wide_dat$Columbia
## W = 0.99117, p-value = 0.9836
```

```r
shapiro.test(wide_dat$Florence)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  wide_dat$Florence
## W = 0.95802, p-value = 0.7941
```

Check Homogeneity of Variance Assumption

```
#Load required packages. These might need to be installed
library(car)
```

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

```
#Bartlett's Test
bartlett.test(dat$rating~dat$city)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  dat$rating by dat$city
## Bartlett’s K-squared = 0.30997, df = 2, p-value = 0.8564
```

```
#If data were not normal then use Lavene's Test from the car package
#Bartlett's test
leveneTest(dat$rating~dat$city)
```

```
## Levene’s Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  2   0.189 0.8302
##       12
```

Run Equal Variance ANOVA

This code chuck performs an ANOVA, displays summary data, then performs a Tukey post-hoc pairwise comparisons test.

```
res = aov(dat$rating~dat$city)
summary(res)
TukeyHSD(res)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## dat$city     2   5012  2505.9     182 1.06e-09 ***
## Residuals   12    165    13.8
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = dat$rating ~ dat$city)
##
## $‘dat$city‘
##                     diff      lwr      upr    p adj
## Columbia-Charleston 17.2 10.93951 23.46049 2.52e-05
## Florence-Charleston 44.4 38.13951 50.66049 0.00e+00
## Florence-Columbia   27.2 20.93951 33.46049 2.00e-07
```

Unequal Variance Welch's ANOVA

This code chuck performs a Welch's ANOVA, displays summary data, then performs a Games-Howell post-hoc pairwise comparisons test.

```
#Load required packages. These might need to be installed
library(rstatix)
res = oneway.test(dat$rating~dat$city, var.equal=FALSE)
res
games_howell_test(dat, rating~city)
```

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

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

##
##  One-way analysis of means (not assuming equal variances)
##
## data:  dat$rating and dat$city
## F = 202.75, num df = 2.0000, denom df = 7.8989, p-value = 1.628e-07

## # A tibble: 3 x 8
##   .y.    group1     group2   estimate conf.low conf.high     p.adj p.adj.signif
## * <chr>  <chr>      <chr>       <dbl>    <dbl>     <dbl>     <dbl> <chr>
## 1 rating Charleston Columbia     17.2     10.0      24.4   3.7 e-4 ***
## 2 rating Charleston Florence     44.4     38.3      50.5   7.51e-8 ****
## 3 rating Columbia   Florence     27.2     20.3      34.1   1.51e-5 ****
```

Non-parametric Kruskal-Wallis Test

This code chuck performs a Kruskal-Wallis Test, displays summary data, then performs a Dunn post-hoc pairwise comparisons test. Note, the data provided on this tutorial meet assumptions of an ANOVA. The Kruskal-Wallis test is provided for your information.

```
#Load required packages. These might need to be installed
library(FSA)
res = kruskal.test(dat$rating~dat$city)
res
dunnTest(dat$rating~dat$city)
```

```
## Registered S3 methods overwritten by 'FSA':
##   method       from
##   confint.boot car
##   hist.boot    car

## ## FSA v0.9.3.9000. See citation('FSA') if used in publication.
## ## Run fishR() for related website and fishR('IFAR') for related book.
```

```
##
## Attaching package: 'FSA'

## The following object is masked from 'package:car':
##
##     bootCase

##
##  Kruskal-Wallis rank sum test
##
## data:  dat$rating by dat$city
## Kruskal-Wallis chi-squared = 12.5, df = 2, p-value = 0.00193

## Dunn (1964) Kruskal-Wallis multiple comparison

##   p-values adjusted with the Holm method.

##              Comparison         Z     P.unadj       P.adj
## 1 Charleston - Columbia -1.767767 0.077099872 0.154199743
## 2 Charleston - Florence -3.535534 0.000406952 0.001220856
## 3   Columbia - Florence -1.767767 0.077099872 0.077099872
```