Setting Up Data

Below we will use the code setwd() to set our working directory so that R can find our file. Then we will input our data set "Nicotinocin" and call it GroupData.

```
knitr::opts_chunk$set(echo = TRUE)

#setwd("/Users/lauranstoner/Downloads")
GroupData <- read.csv("Nicotinocin.csv", header = TRUE, sep = ",")
```

Sampling Designs and Considerations

Taking a sample from a larger population can enable us to further examine data. The most useful function for sampling is sample(). For this function, we must input the vector we want to sample from, the size of the sample, and we have the option of sampling with or without replacement. Below, we see an example of a sample of 15 different weight values using replacement.

```
# Example:
sample(GroupData$Age, 15, replace = TRUE)
```

```
##  [1] 22 48 51 53 48 27 45 25 31 35 20 26 22 21 26
```

Sampling Variability

Under sampling variability we briefly discuss the types of variables. Broadly, variables are either categorical or quantitative. We can use functions to determine the variable types in a data set, or even change them if need be. Using is._() **can help us determine the type of variable. Using as.**_() can help us change the type of variable. We can either fill the blank with numeric, character, factor, integer, or others depending on what type we want our variable to be.

```

```

```
# Determine variable type in our example data set.
is.numeric(GroupData$Age)
```

```
## [1] TRUE
```

```
# We are returned with a "true" meaning the variable "Age" is numeric.

#Change data type of "Age" to character.
AgeC<- as.character(GroupData$Age, length = 2)
is.character(AgeC)
```

```
## [1] TRUE
```

```
# We have now reformed the variable "Age" into one of character type with a field length of 2.
```

Summarizing and Exploring Data

Numerical Summaries

```
# Frequency Distribution: We can use functions such as length() or table() to examine relative frequenc
table(GroupData$Age)
```

```
##
## 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 38 40 41 42 43 44 45
##  3  4  7  6  6  4  8  9  6  4  4  2  1  3  3  1  3  1  1  2  2  2  2  1  1  2
## 46 47 48 49 50 51 53 58 59 61 62
##  1  1  2  1  1  1  1  1  1  1  1
```

```
length(GroupData$Age)
```

```
## [1] 100
```

```
# With the relative frequency of each value and the length of the variable itself, we can calculate the

# Measures of centre: This describes the typical value of a distribution.
mean(GroupData$Days.with.Symptoms)
```

```
## [1] 12.46
```

```
median(GroupData$Days.with.Symptoms)
```

```
## [1] 12
```

```
# Percentile: Percentiles are a value below which a particular percentage of the distribution lies.
quantile(GroupData$Days.with.Symptoms, 0.75)
```

```
## 75%
##  14
```

```
# Measure of Spread: Values such as range, 5 number summaries, interquartile range, variance, and stand
sd(GroupData$Days.with.Symptoms)
```

```
## [1] 2.645446
```

```
var(GroupData$Days.with.Symptoms)
```

```
## [1] 6.998384
```

```
IQR(GroupData$Days.with.Symptoms)
```

```
## [1] 4
```

```
fivenum(GroupData$Days.with.Symptoms)
```

```
## [1]  8 10 12 14 18
```

```
range(GroupData$Days.with.Symptoms)
```

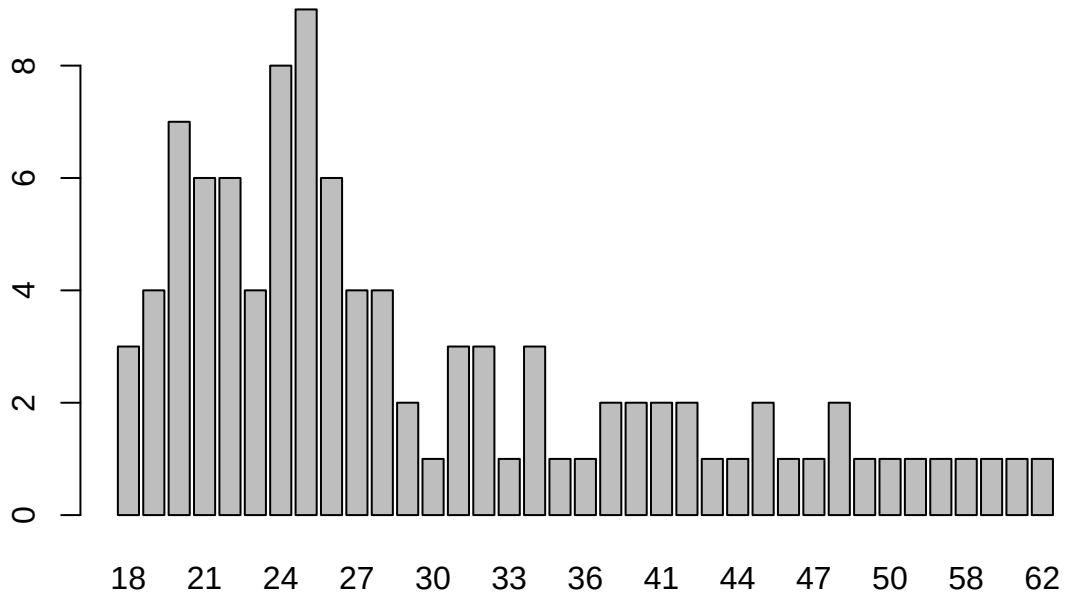
```
## [1]  8 18
```

Graphical Summaries

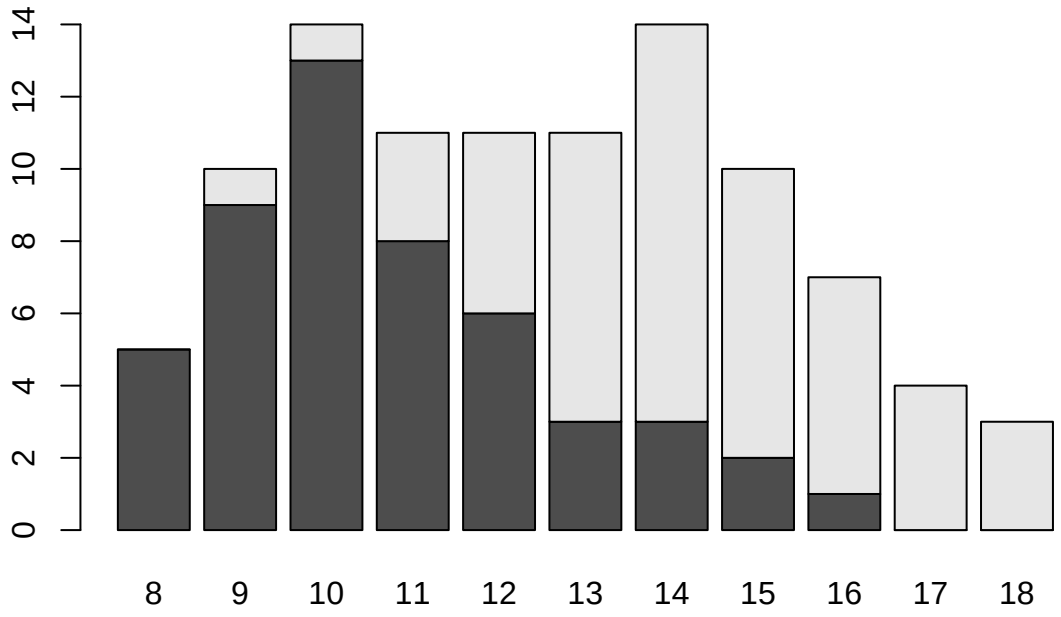
```
# Barplot: Visual representation of the frequency distribution of a variable.
table(GroupData$Age)
```

```
##
## 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 38 40 41 42 43 44 45
##  3  4  7  6  6  4  8  9  6  4  4  2  1  3  3  1  3  1  1  2  2  2  2  1  1  2
## 46 47 48 49 50 51 53 58 59 61 62
##  1  1  2  1  1  1  1  1  1  1  1
```

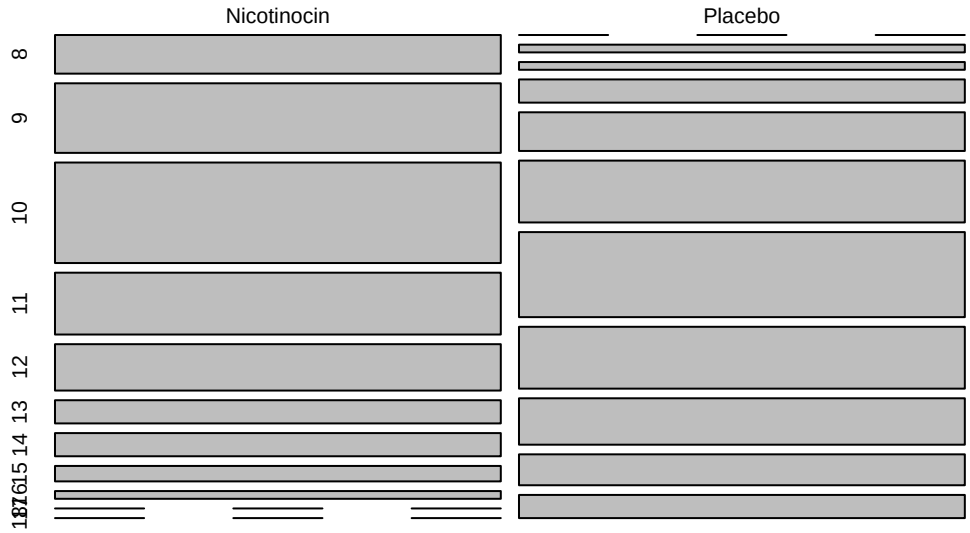
```
barplot(table(GroupData$Age))
```



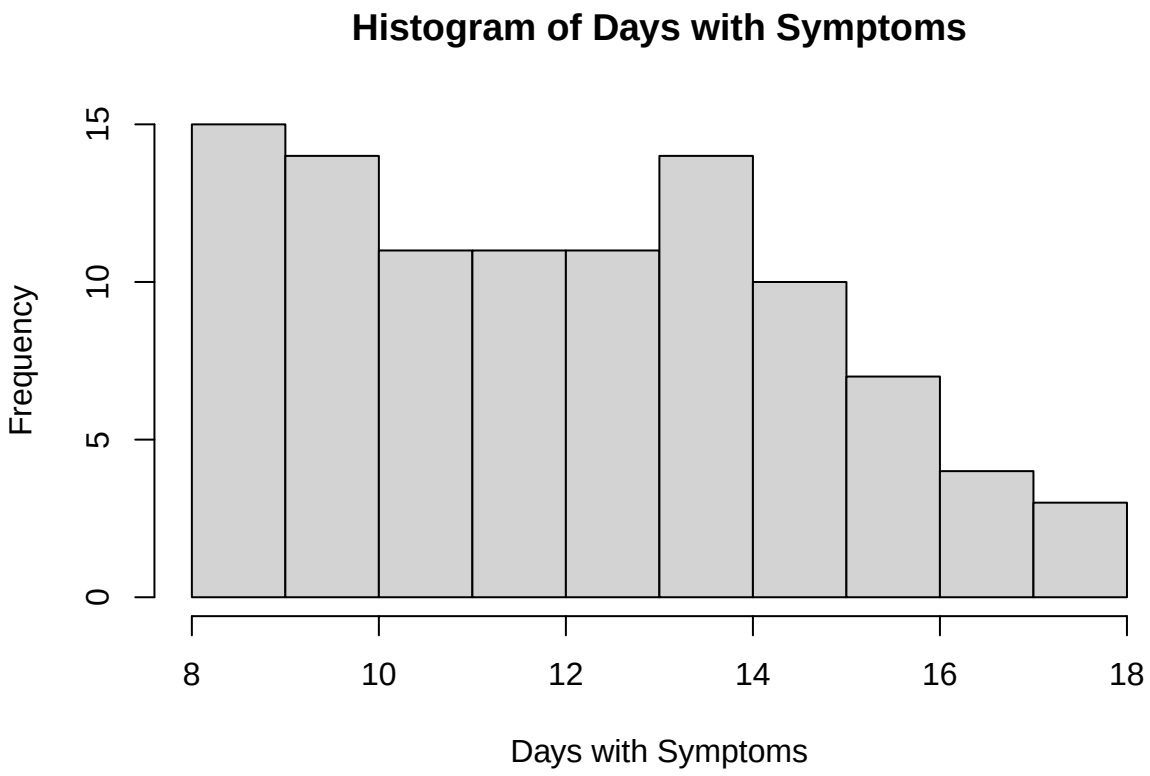

```
# Stacked Barplot: Stacked barplots are a visual representation of a two-way table. Our example below s
StackedPlot <- table(GroupData$Group, GroupData$Days.with.Symptoms)
barplot(StackedPlot)
```



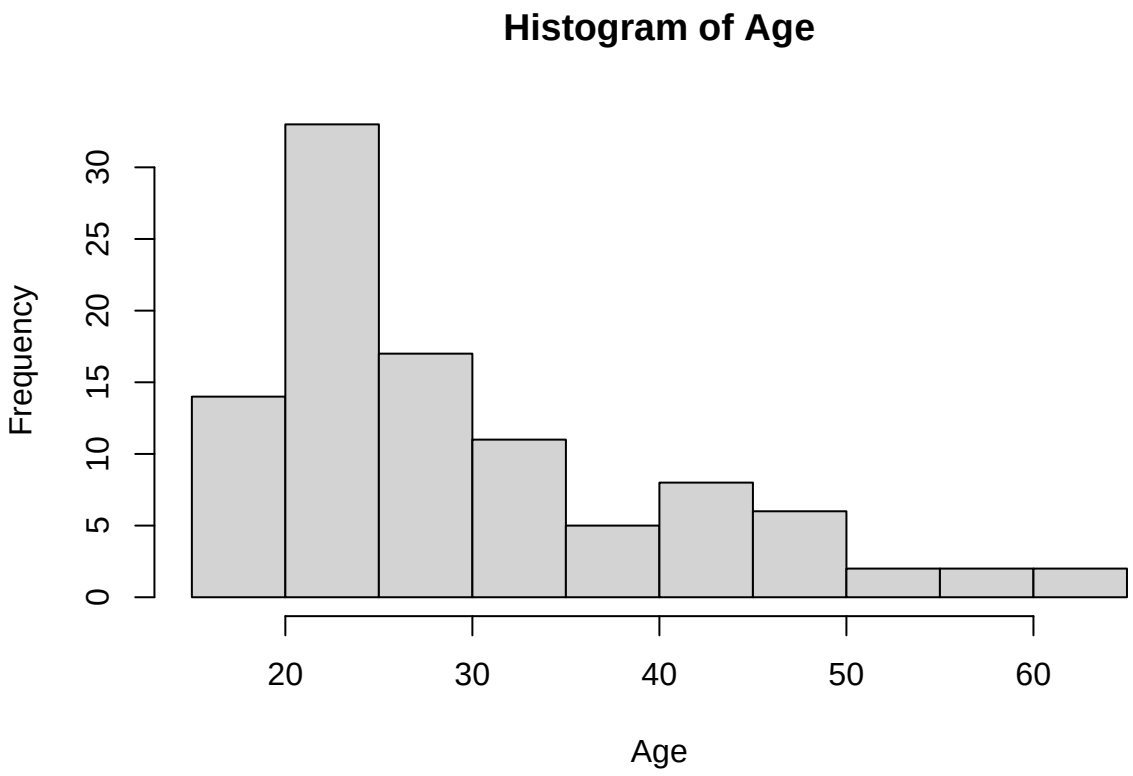

```
# Mosaic Plot: Area of rectangles reflect relative frequencies for two or more variables.
MosaicExample <- table(GroupData$Group, GroupData$Days.with.Symptoms)
mosaicplot(MosaicExample)
```



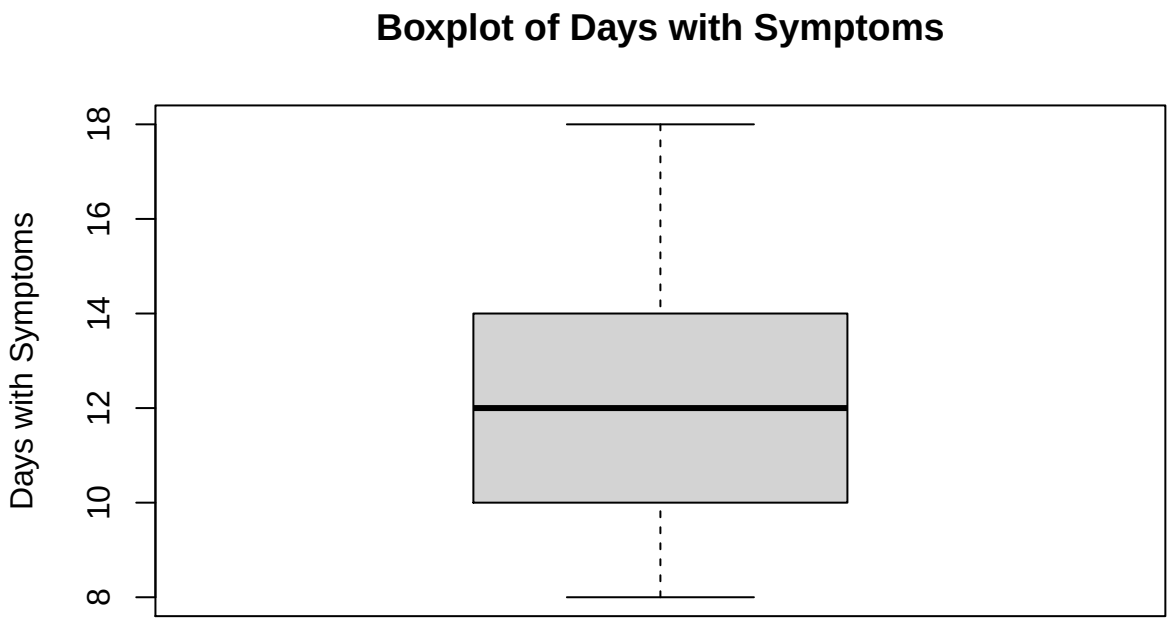

```
# Histogram: Visual representation of frequency distribution for a single quantitative variable.
hist(GroupData$Days.with.Symptoms, xlab = "Days with Symptoms", main = "Histogram of Days with Symptoms
```



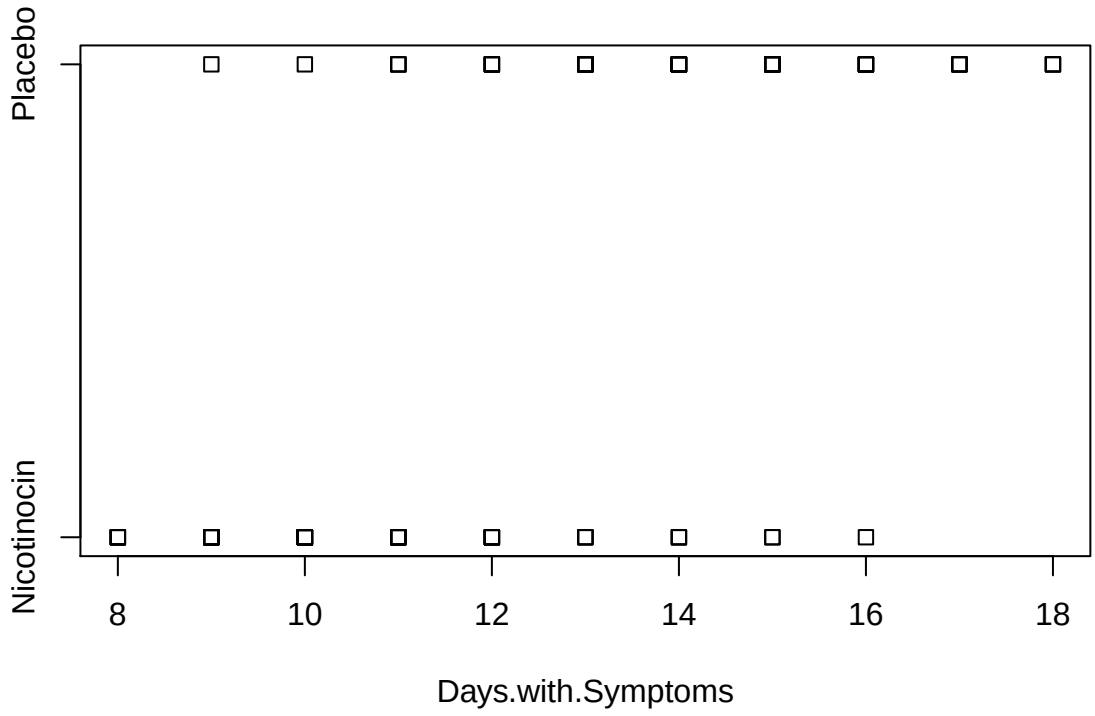

```
# Evaluating Symmetry: We can describe distributions by their degree of symmetry. In our example below,
# Evaluating Modality: We can also describe the distributions by modality. Modality is the number of pr
hist(GroupData$Age, xlab = "Age", main = "Histogram of Age")
```



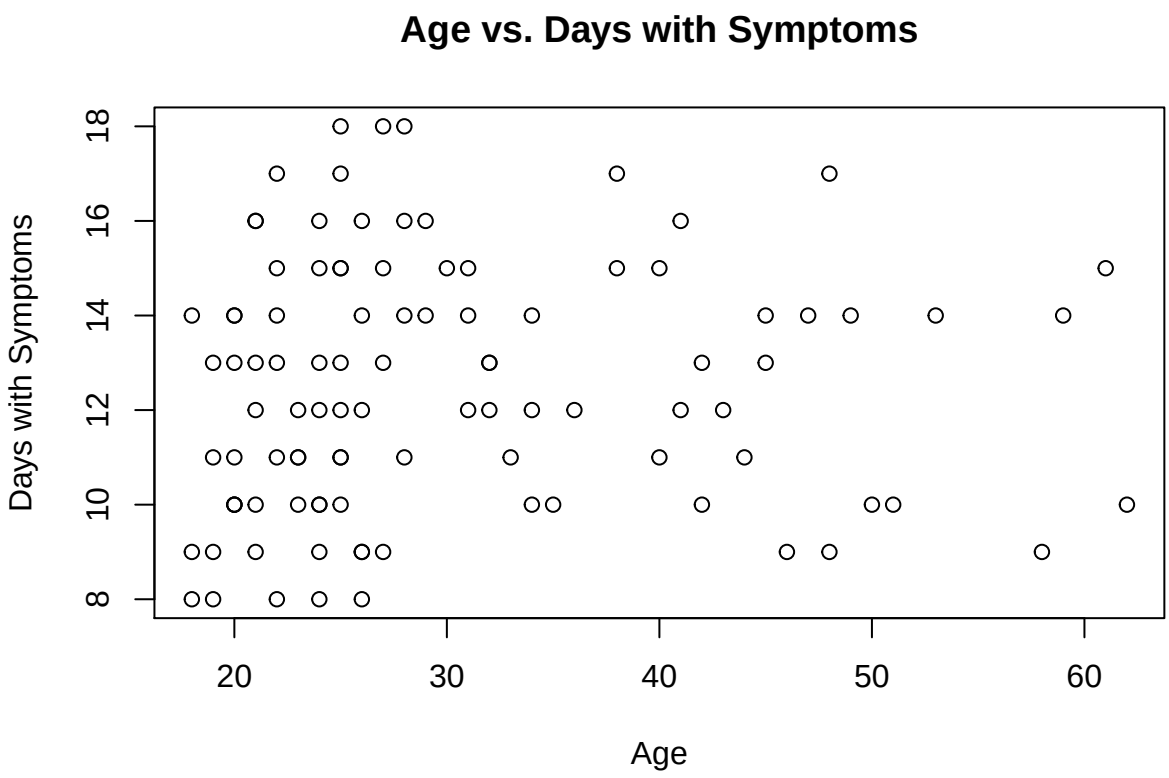

```
# Boxplot: Boxplots are an incredibly informative summary of data. Boxplots incorporate the min, first
boxplot(GroupData$Days.with.Symptoms, ylab = "Days with Symptoms", main = "Boxplot of Days with Symptom
```




```
# Stripchart: Stripcharts plot data points of a quantitative variable across categorical groups.
stripchart(Days.with.Symptoms ~ Group, data = GroupData)
```




```
# Scatterplot: Scatterplots plot quantitative variables as x,y coordinates.
plot(GroupData$Age, GroupData$Days.with.Symptoms, xlab = "Age", ylab = "Days with Symptoms", main = "Ag
```



Probability Models: Binomial

Binomial models can be described by their shape, center, and spread. This means we can use curves, mean functions, and SD functions as shown above. We can also use separate functions curated just for binomial models to work with our data.

```
# Using dbinom to find the probability at a certain value, first compute the probability of a participa
lessthan10nic <- nrow(GroupData[(GroupData$Days.with.Symptoms<10) & (GroupData$Group == "Nicotinocin"),]
numtotal2 <- nrow(GroupData[(GroupData$Group == "Nicotinocin"),])
lessthan10nic
```

```
## [1] 14
```

```
numtotal2
```

```
## [1] 50
```

```
nicolessthan10 <-lessthan10nic/numtotal2
nicolessthan10
```

```
## [1] 0.28
```

```r
#Using dbinom, what is the probability that EXACTLY 5 out of 50 participants on nicotinocin will show l
#x is the number of successes you want, size is the sample size, and the prob argument is the probabili
dbinom(x=5, size=50, prob=0.28)
```

```
## [1] 0.001386227
```

```r
# Using pbinom to find the cumulative probability,
#in this case, we will find what is the probability that 5 OR LESS of the 50 participants under nicotin
pbinom(q=5, size=50, prob=0.28)
```

```
## [1] 0.001873605
```

```r
# Using qnorm to find the value given a certain probability.
#In this case, how many participants under nicotinocin will be expected to show less than 10 days of sy
qbinom(p=pbinom(q=5, size=50, prob=0.28), size=50, prob=0.28)
```

```
## [1] 5
```

Probability Models: Normal

Normal distribution are defined by their mean and SD, so we can use these functions as presented in previous sections. We can also use separately curated function to work with the data.

```r
# Using dnorm to find density at a certain value. what is the probability that a participant under nico
#Compute mean and standard deviation of nicotinocin group
Mean_n<-mean(GroupData[(GroupData$Group == "Nicotinocin"), "Days.with.Symptoms"])
Stan_n<-sd(GroupData[(GroupData$Group == "Nicotinocin"), "Days.with.Symptoms"])
Mean_n
```

```
## [1] 10.76
```

```r
Stan_n
```

```
## [1] 1.985252
```

```r
dnorm(x=10, mean=Mean_n, sd=Stan_n)
```

```
## [1] 0.1867544
```

```r
# Using pnorm to find the cumulative probability given a certain value.
#What is the probability that a participant under nicotinocin treatment will show 10 or less days of sy
pnorm(q=10, mean=Mean_n, sd=Stan_n)
```

```
## [1] 0.3509255
```

```r
# Using qnorm to find the value given a certain cumulative probability.
#Given probability from the pnorm scenario above:
qnorm(p=pnorm(q=10, mean=Mean_n, sd=Stan_n), mean=Mean_n, sd=Stan_n)
```

```
## [1] 10
```

T Confidence Intervals for the Mean

T distributions are used when we are relying on a sample s to estimate the population standard deviation. T distributions are a type of density curve that centers around a mean of zero, is unimodal, bell-shaped, and its spread is described by degrees of freedom. We can use this distribution to compute confidence intervals if our data originates from an SRS and the data comes from a normal population.

```
# Using pt for cumulative probability. In this scenario, our lower boundary is -1 in the distribution a
pt(q=-1, df=99)
```

```
## [1] 0.1598742
```

```
# Using qt to find value of t given a cumulative probability. In this scenario, we will use the previou
qt(p=0.1598742, df=99)
```

```
## [1] -1
```

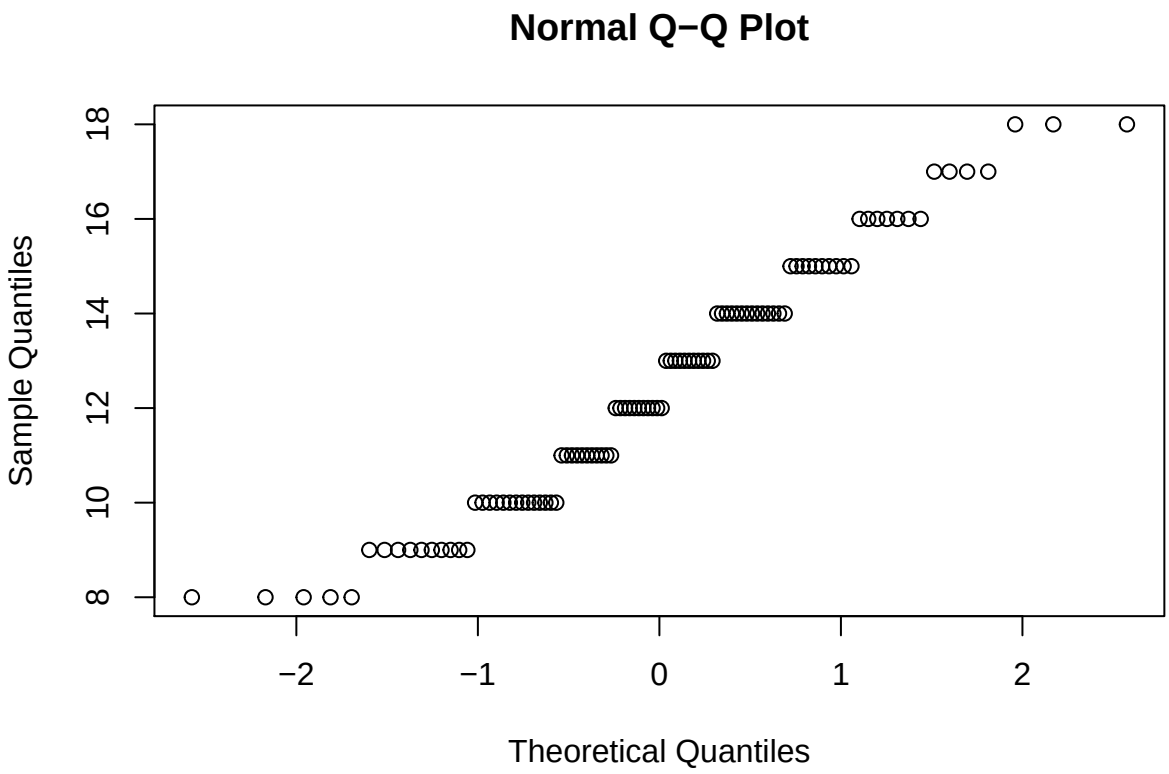
```
# Using t.test to compute a confidence interval with 95% confidence. For this argument, our default arg
t.test(GroupData$Days.with.Symptoms)
```

```
##
##  One Sample t-test
##
## data:  GroupData$Days.with.Symptoms
## t = 47.1, df = 99, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  11.93509 12.98491
## sample estimates:
## mean of x
##     12.46
```

Large Sample Confidence Intervals for Proportion

Large sample confidence intervals are computed given 4 conditions: data is from a SRS, count of successes can be approximated by a binomial model, normal approximation of binomial model is reasonable, and np and nq must be at least 15. Due to this being a normal model, we are also able to use pnorm and qnorm functions as well.

```
# The qqnorm function can be used to determine if a normal approximation is reasonable. We can also use
qqnorm(GroupData$Days.with.Symptoms)
```




```
nrow(GroupData)
```

```
## [1] 100
```

```
# Prop.test enables us to compute a confidence interval around the data. We use x (5) as a count of our
prop.test(5, 20, conf.level = 0.90, correct = FALSE)
```

```
##
##  1-sample proportions test without continuity correction
##
## data:  5 out of 20, null probability 0.5
## X-squared = 5, df = 1, p-value = 0.02535
## alternative hypothesis: true p is not equal to 0.5
## 90 percent confidence interval:
##  0.1273772 0.4322018
## sample estimates:
##    p
## 0.25
```

Large Sample Test for Proportion

The large sample Test for Proportion has 3 conditions to be used: the sample data is from a SRS, count of successes can be described by a binomial model, and lastly, the normal approximation of the binomial model must be reasonable. However, in this case, we must check that np0 and nq0 are greater or equal to

10 (where q0 is the compliment of p0). It is important to emphasize that these values are based on the hypothesis values and not the sample estimates.

```
lessthan10pla <- nrow(GroupData[(GroupData$Days.with.Symptoms<10) & (GroupData$Group == "Placebo"),])
numtotal <- nrow(GroupData[(GroupData$Group == "Placebo"),])
lessthan10pla
```

```
## [1] 1
```

```
numtotal
```

```
## [1] 50
```

```
placebolessthan10 <-lessthan10pla/numtotal
placebolessthan10
```

```
## [1] 0.02
```

As such, we can see that our null hypothesis value will be 0.02, since we expect the participants that take nicotinocin to experience less days with symptoms than with the placebo, we can suggest the alternative hypothesis: The frequency of participants experiencing less than 10 days of symptoms using the nicotinocin treatment will be higher than 0.02. Therefore,

Null hypothesis: $p < 0.02$ Alternative hypothesis: $p > 0.02$

```
lessthan10nic <- nrow(GroupData[(GroupData$Days.with.Symptoms<10) & (GroupData$Group == "Nicotinocin"),]
numtotal2 <- nrow(GroupData[(GroupData$Group == "Nicotinocin"),])
lessthan10nic
```

```
## [1] 14
```

```
numtotal2
```

```
## [1] 50
```

```
nicolessthan10 <-lessthan10nic/numtotal2
nicolessthan10
```

```
## [1] 0.28
```

```
#Using this information, we conduct a prop.test: In this case, our x value is the count of "successes"
prop.test(x=14, n=50, p=0.02, alternative="greater", correct=FALSE)
```

```
## Warning in prop.test(x = 14, n = 50, p = 0.02, alternative = "greater", : Chi-
## squared approximation may be incorrect
```

```
##
##  1-sample proportions test without continuity correction
##
## data:  14 out of 50, null probability 0.02
```

```
## X-squared = 172.45, df = 1, p-value < 2.2e-16
## alternative hypothesis: true p is greater than 0.02
## 95 percent confidence interval:
##  0.1889395 1.0000000
## sample estimates:
##    p
## 0.28
```

```
#Since we would like to include a Confidence interval, we will have to rerun the test without the alter
prop.test(x=14, n=50,correct=FALSE)
```

```
##
##  1-sample proportions test without continuity correction
##
## data:  14 out of 50, null probability 0.5
## X-squared = 9.68, df = 1, p-value = 0.001863
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.1747417 0.4166512
## sample estimates:
##    p
## 0.28
```

We can conclude from this data that the proportion of participants using nicotinocin (95% Cl:0.175-0.417)is greater than the participants using the placebo(P=2.2e-16, n=50).

T Test for the Mean

Since we are dealing with another hypothesis test, we will need to define our null and alternative hypothesis. In this case, our null hypothesis value will equal the parameter (u=u0) and our alternative hypothesis will show a difference in u and u0 (Can be left tailed, right tailed, or two tailed). There are 2 conditions that must be fulfilled to use the T test for the mean. The first is that the sample must be from a SRS. Additionally, the sample must come from a population that follows a normal distribution as we will need a t distribution for this test. However, if we have a large enough sample, in this case n=50, we can consider this condition fulfilled. From the t distribution with degrees of freedom equal to n-1, we can calculate t test, that is, (Sample statistic - NULL value)/SE_mean. This test statistic is appropriate for pairwise difference in data as well as testing against a null value for a given single quantitative variable.

```
#the x argument will compute what data values to take from. the mu argument will indicate the true valu
#t.test(x=diff,mu=0,alternative="less")
```

T Test for Differences in Means

The T test for difference in means is used when comparing 2 population means. Again, we will need to define our null and alternative hypothesis. In this case, the null hypothesis is that there is no difference between the mean number of days of symptoms in the nicotinocin group, and the placebo group. The alternative hypothesis is that there is a difference in the mean number of days of symptoms between the nicotinocin group, and the placebo group.

Null hypothesis(H_0): μn - μp ≤ 0 Alternative hypothesis (H_A) μn - μp $\neq 0$

There are 2 conditions for T test for difference in means: Both samples must be SRS, and the samples must be independent from each other and are normally distributed. As stated in the PPDAC, our sample is not a

SRS, however, since we have a dataset of n=50, it fulfills the criteria for central limit thereom and therefore is normally distributed.

```
#the x represents the first data set you will take from while the y argument represents the second data
nicotinocingroup<-GroupData[GroupData$Group == "Nicotinocin", ("Days.with.Symptoms")]
placebogroup<-GroupData[GroupData$Group == "Placebo", ("Days.with.Symptoms")]
t.test(x=nicotinocingroup, y=placebogroup, mu=0, alternative="two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  nicotinocingroup and placebogroup
## t = -8.3745, df = 97.814, p-value = 4.051e-13
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.205705 -2.594295
## sample estimates:
## mean of x mean of y
##     10.76     14.16
```

As our p-value is lower than 0.05 which is our critical value, we can reject our null hypothesis

Conclusion: The participants using nicotinocin treatment ($\bar{x}$=10.76, s=1.99, n=50) showed lower mean number of days with symptoms than the participants on the placebo treatment ($\bar{x}$=14.16, s=2.07, n=50)

Large Sample Test for Difference in Proportion

A large sample test for difference in proportion is useful when we want to compare a difference in proportions of more than one group. Our conditions for this situation are that both samples are from an SRS, the count of successes can be described by a binomial model, the normal approximation of the binomial model is reasonable, and we must have at least 5 successes and 5 failures for each sample (np and nq). For a large sample confidence interval in this case, we must note that we require 10 successes and failures for each group instead of 5.

```
Placebo <- subset(GroupData, GroupData$Group=="Placebo")
Nicotinocin <- subset(GroupData, GroupData$Group=="Nicotinocin")

Placebo10 <- subset(Placebo, Placebo$Days.with.Symptoms>10)
Nicotinocin10 <- subset(Nicotinocin, Nicotinocin$Days.with.Symptoms>10)

nrow(Placebo)
```

```
## [1] 50
```

```
nrow(Nicotinocin)
```

```
## [1] 50
```

```
nrow(Placebo10)
```

```
## [1] 48
```

```r
nrow(Nicotinocin10)
```

```
## [1] 23
```

```r
# We see that we have 48 success for placebo out of 50 total, and 23 successes for nicotinocin out of 5

prop.test(x=c(48,23), n=c(50,50))
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(48, 23) out of c(50, 50)
## X-squared = 27.975, df = 1, p-value = 1.229e-07
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.3315594 0.6684406
## sample estimates:
## prop 1 prop 2
##   0.96   0.46
```

Use R to run t procedures to conduct inference on two population means

The independent two sample t-test and confidence interval are parametric methods appropriate for examining the difference in means for two populations. They can also be thought of as a way of examining the relationship between a numeric outcome or Y variable and a categorical explanatory or X variable.

```r
# Compare the Symptoms separated on groups
#The null hypothesis is that the mean difference is 0 and the alt is two-sided
#The confidence of 95 percent for the confidence interval
# The variances are not equal and these two groups are not paired
t.test(GroupData$Days.with.Symptoms~GroupData$Group, mu=0, alt="two.sided", conf=0.95, var.eq=F, paired=
```

```
##
##  Welch Two Sample t-test
##
## data:  GroupData$Days.with.Symptoms by GroupData$Group
## t = -8.3745, df = 97.814, p-value = 4.051e-13
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.205705 -2.594295
## sample estimates:
## mean in group Nicotinocin     mean in group Placebo
##                     10.76                     14.16
```

The output returned by R:
The test statistic of negative 8.375, the p-value is 4.051e-13 The 95 percent confidence interval for the difference in means running from negative 4.206 to negative 2.594; as well the sample group means of 10.76 to 14.16

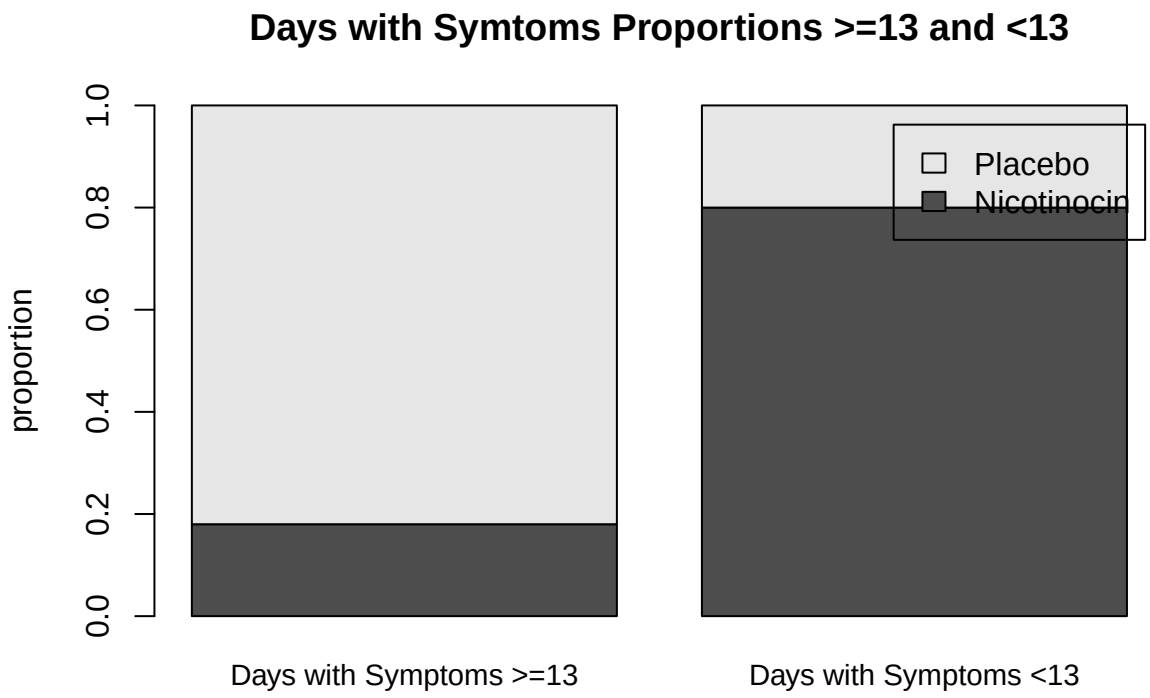
User R to run large sample procedures to conduct inference on two proportions

Create a table with two rows: treatment group "Nicotinocin" & treatment group "Placebo" and two columns: Days with Symptoms ≥ 13 & Days with Symptoms <13

```
#Create the table

#Total number of Nicotinocin items
total_Nicotinocin <- nrow(GroupData[(GroupData$Group=="Nicotinocin"),])
#Total number of Placebo items
total_Placebo <- nrow(GroupData[(GroupData$Group=="Placebo"),])
#Total number of Nicotinocin items with Days with Symptoms >= 13
gl13_Nicotinocin <- nrow(GroupData[(GroupData$Group=="Nicotinocin" &
                                    GroupData$Days.with.Symptoms >=13),])
#Total number of Placebo items with Days with Symptoms >= 13
gl13_Placebo <- nrow(GroupData[(GroupData$Group=="Placebo" &
                                  GroupData$Days.with.Symptoms >=13),])

matrixGL13 <- matrix(c(gl13_Nicotinocin, total_Nicotinocin-gl13_Nicotinocin,
                       gl13_Placebo,total_Placebo-gl13_Placebo), ncol=2)

rownames(matrixGL13) <- c("Nicotinocin", "Placebo")
colnames(matrixGL13) <- c("Days with Symptoms >=13", "Days with Symptoms <13" )

matrixGL13 = as.table(matrixGL13)

#display table
matrixGL13
```

```
##             Days with Symptoms >=13 Days with Symptoms <13
## Nicotinocin                       9                     40
## Placebo                          41                     10
```

```
#display barplot
barplot(prop.table(matrixGL13, margin = 2), beside=FALSE, ylim=c(0,1),
        legend = rownames(matrixGL13),
        main="Days with Symtoms Proportions >=13 and <13", ylab="proportion", cex.names = 0.9 )
```




```
# gl13_Nicotinocin: Days with Symptoms is greater than 13 in crop Nicotinocin
# gl13_Placebo: Days with Symptoms is greater than 13 in Placebo
# total_Nicotinocin Total number of items in treatment group Nicotinocin
# total_Placebo Total number of items in treatment group Placebo

prop.test(c(gl13_Nicotinocin, gl13_Placebo), c(total_Nicotinocin,total_Placebo))
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(gl13_Nicotinocin, gl13_Placebo) out of c(total_Nicotinocin, total_Placebo)
## X-squared = 36.014, df = 1, p-value = 1.959e-09
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.7937292 -0.4462708
## sample estimates:
## prop 1 prop 2
##   0.18   0.80
```

R Output for the prop.test() function

Hypothesis test of $H_0 : p1 = p2 vs H_A : p1 \neq p2$ A 95 percent confidence interval for p1-p2 is (-0.794, -0.446) The proportion estimates based on these data are 0.18 (Days with Symptoms $\geq$ 13) and 0.80 (Days with Symptoms $<$ 13)

Simple Linear Regressions

Simple linear regression is useful for examining or modelling the relationship between two numeric variables.

Fit simple linear regression models

```
#Calculate the Pearson's correlation between Day and Weight
cor(GroupData$Age, GroupData$Days.with.Symptoms)
```

```
## [1] 0.06102452
```

```
#Fit a linear regression and save it in the object: fit
fit <-lm( GroupData$Age~GroupData$Days.with.Symptoms)
```

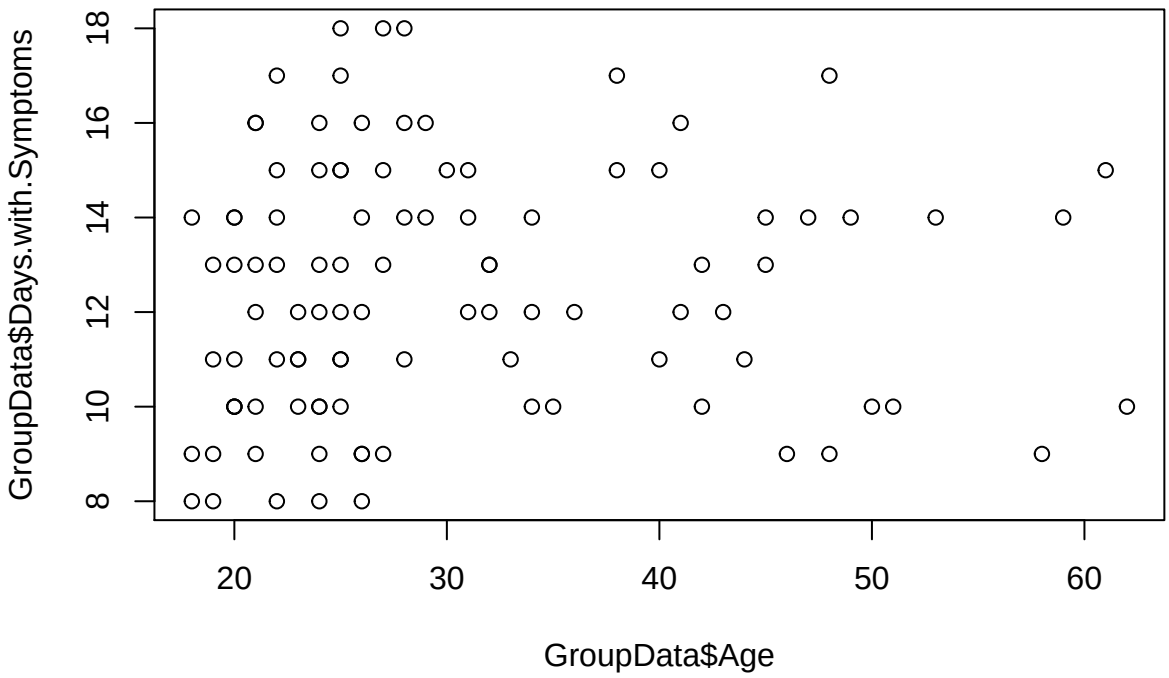
Summary function returns a summary for the residuals, the estimate of the intercept, its standard error as well as the test statistic, and p-value for a hypothesis test that the intercept is zero.

```
#Summary of the model fit
summary(fit)
```

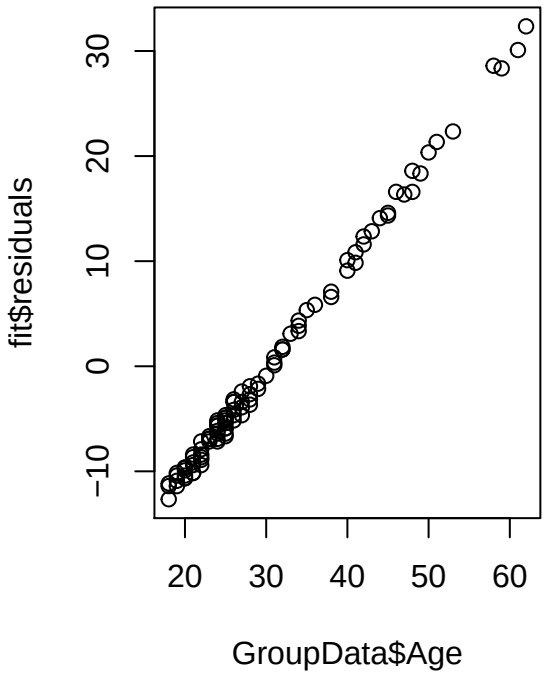
```
##
## Call:
## lm(formula = GroupData$Age ~ GroupData$Days.with.Symptoms)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.656  -7.344  -4.404   5.471  32.346
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    27.1483     5.2715   5.150 1.35e-06 ***
## GroupData$Days.with.Symptoms    0.2505     0.4139   0.605    0.546
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.9 on 98 degrees of freedom
## Multiple R-squared:  0.003724,   Adjusted R-squared:  -0.006442
## F-statistic: 0.3663 on 1 and 98 DF,  p-value: 0.5464
```

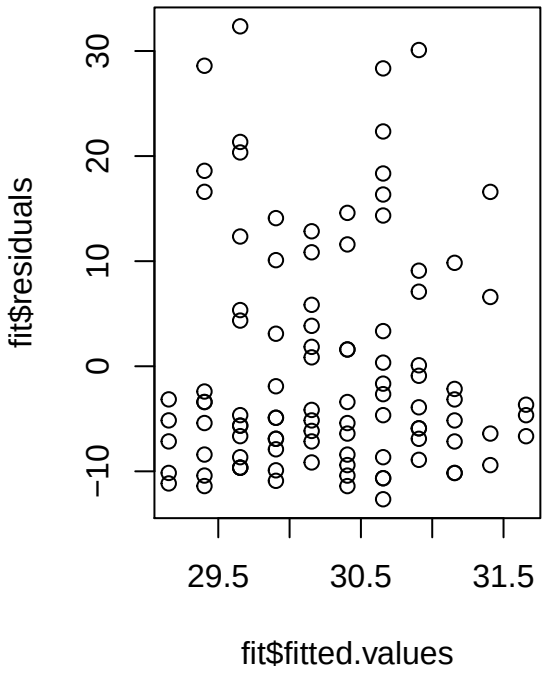
Produce regression diagnostics

```
#Add fitted line to scatterplot
plot(GroupData$Age, GroupData$Days.with.Symptoms)
abline(fit, col=2, lwd=3)
```

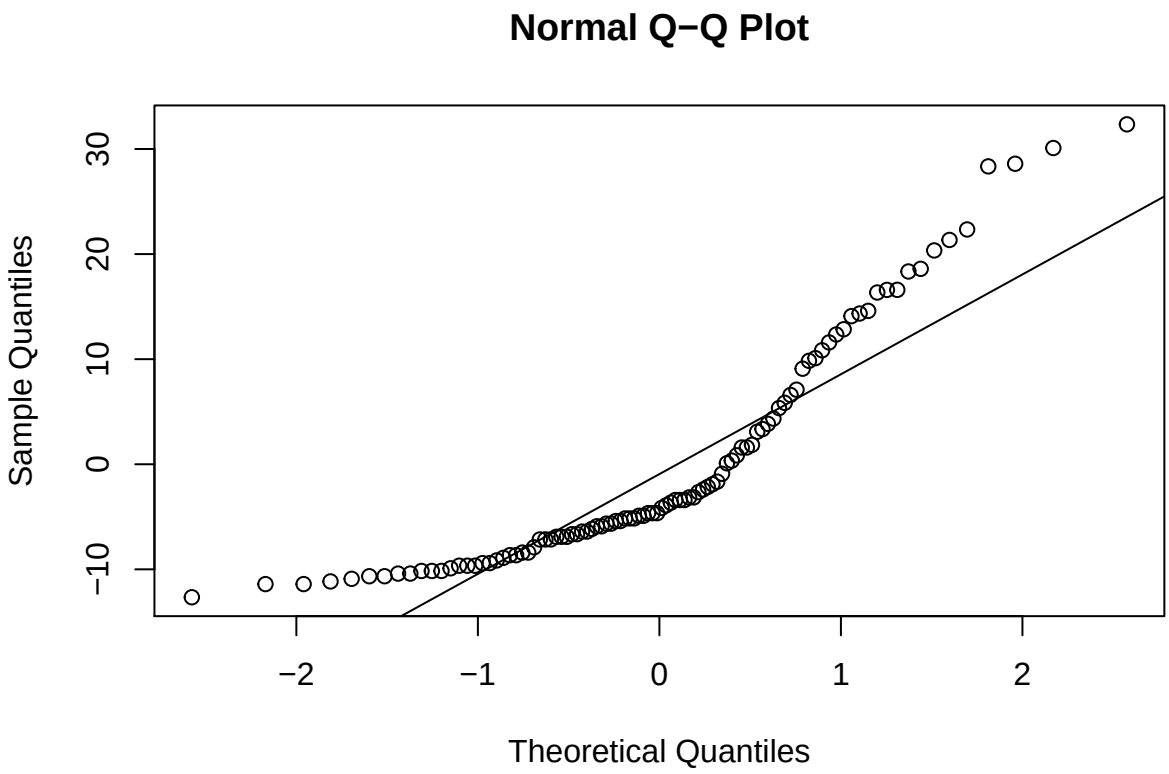



```
#Produce residual plot
par(mfrow=c(1,2))
plot(GroupData$Age, fit$residuals)
plot(fit$fitted.values, fit$residuals)
```



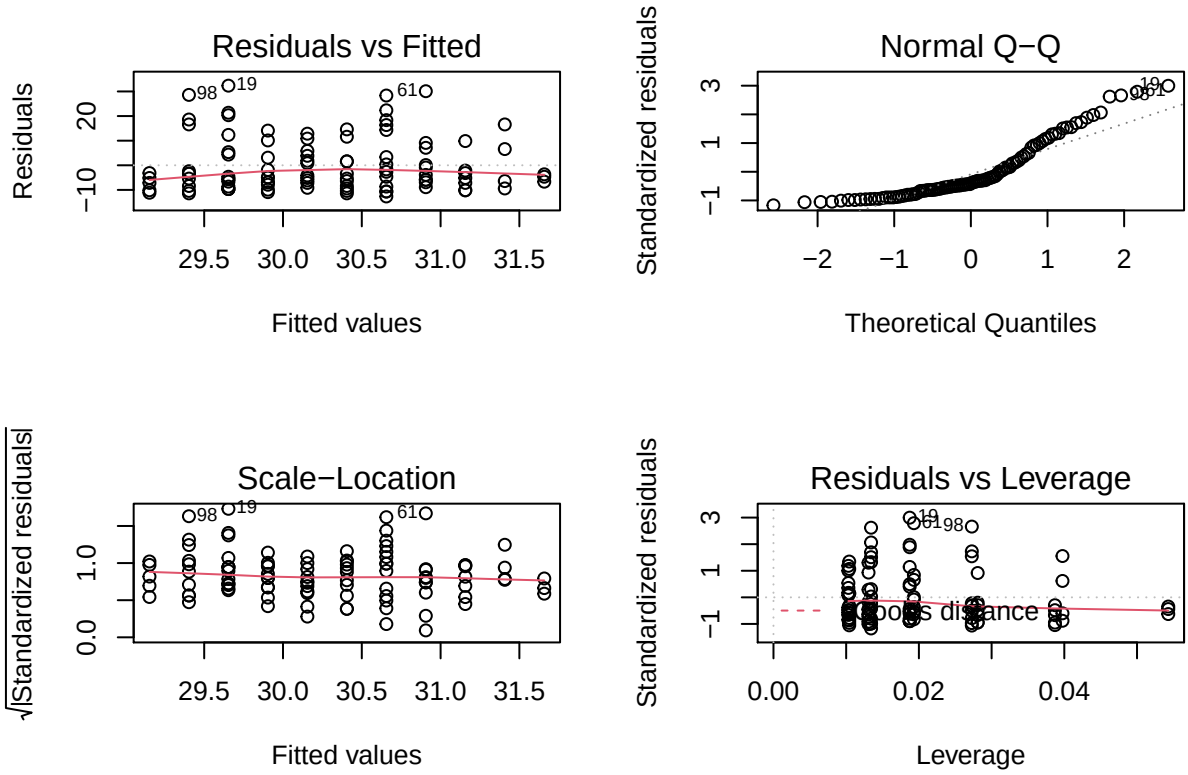



```
#produce QQPlot of residuals
qqnorm(fit$residuals)
qqline(fit$residuals)
```



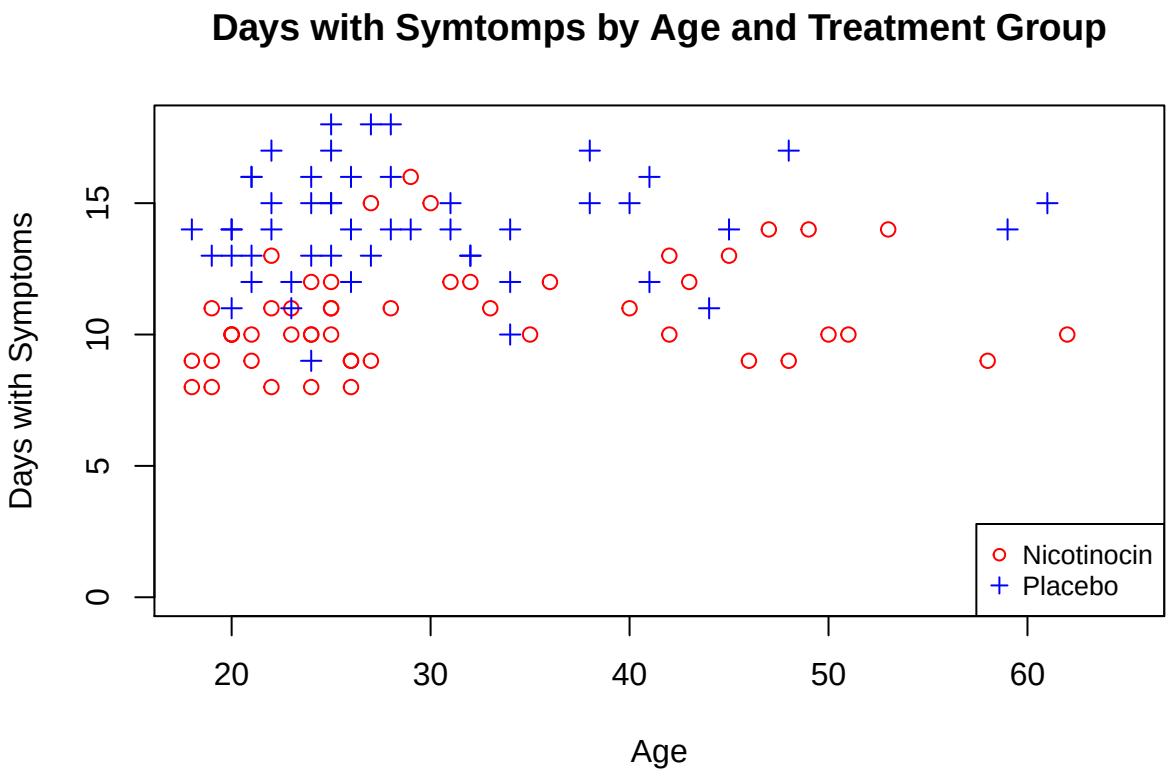
Instead of starting from scratch for all plots, we can plot fit to display all four graphs

```
#Use plot function with fit object to print all four plots in one page
par(mfrow=c(2,2))
plot(fit)
```



Adding a predictor

```
#Adding  the third variable crop using different colors

plot(GroupData$Age[GroupData$Group == 'Nicotinocin'],
     GroupData$Days.with.Symptoms[GroupData$Group == 'Nicotinocin'],
     pch=1, col="red", ylim=c(0,18), xlim=c(18,65), xlab = "Age", ylab = "Days with Symptoms",
     main = "Days with Symtomps by Age and Treatment Group")

points(GroupData$Age[GroupData$Group =='Placebo'],
       GroupData$Days.with.Symptoms[GroupData$Group == 'Placebo'],
       pch=3, col="blue")

legend("bottomright", legend=c("Nicotinocin", "Placebo"), pch=c(1,3), col=c("red", "blue"), cex=0.8)
```



Allow each line to have different Days with Symptoms depending on treatment group

```
#Having different Days with Symptoms depending on the Treatment Group

fit2 <- lm(GroupData$Days.with.Symptoms~GroupData$Age+factor(GroupData$Group))
summary(fit2)
```

```
##
## Call:
## lm(formula = GroupData$Days.with.Symptoms ~ GroupData$Age + factor(GroupData$Group))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0021 -1.3550 -0.1377  1.2264  5.3093
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     9.81686    0.65147  15.069  < 2e-16 ***
## GroupData$Age                   0.03013    0.01872   1.610    0.111
## factor(GroupData$Group)Placebo  3.46207    0.40458   8.557 1.73e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.014 on 97 degrees of freedom
## Multiple R-squared:  0.4323, Adjusted R-squared:  0.4206
## F-statistic: 36.93 on 2 and 97 DF,  p-value: 1.189e-12
```

Allow each line to have different heights & slopes depending on crop type

```
#Having a different height & slopes depending on the crop type
fit3 <- lm(GroupData$Days.with.Symptoms~GroupData$Age*factor(GroupData$Group))
summary(fit3)
```

```
##
## Call:
## lm(formula = GroupData$Days.with.Symptoms ~ GroupData$Age * factor(GroupData$Group))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0859 -1.2766 -0.1495  1.2216  5.3342
##
## Coefficients:
##                                             Estimate Std. Error t value
## (Intercept)                                  9.47778    0.81267  11.663
## GroupData$Age                                0.04097    0.02431   1.685
## factor(GroupData$Group)Placebo               4.26875    1.21973   3.500
## GroupData$Age:factor(GroupData$Group)Placebo -0.02682    0.03825  -0.701
##                                             Pr(>|t|)
## (Intercept)                                  < 2e-16 ***
## GroupData$Age                               0.095192 .
## factor(GroupData$Group)Placebo              0.000708 ***
## GroupData$Age:factor(GroupData$Group)Placebo 0.484828
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.019 on 96 degrees of freedom
## Multiple R-squared:  0.4352, Adjusted R-squared:  0.4175
## F-statistic: 24.66 on 3 and 96 DF,  p-value: 6.502e-12
```

One-Factor ANOVA

ANOVA is a parametric method appropriate for comparing the Means for two or more independent populations

We can conduct an analysis of variance using the "aov" command/function in R

```
# Compare Days with Symptoms separated by treatment Group
#Store aov results in R object
anova1 <- aov(GroupData$Days.with.Symptoms~GroupData$Group)
anova1
```

```
## Call:
##    aov(formula = GroupData$Days.with.Symptoms ~ GroupData$Group)
##
## Terms:
##                 GroupData$Group Residuals
## Sum of Squares           289.00    403.84
## Deg. of Freedom               1        98
##
## Residual standard error: 2.029979
## Estimated effects may be unbalanced
```

Call summary function to return the Sum of Squares, the Mean Squares between groups ['crop' row], and errors ['residuals' row] of computing F test statistic.

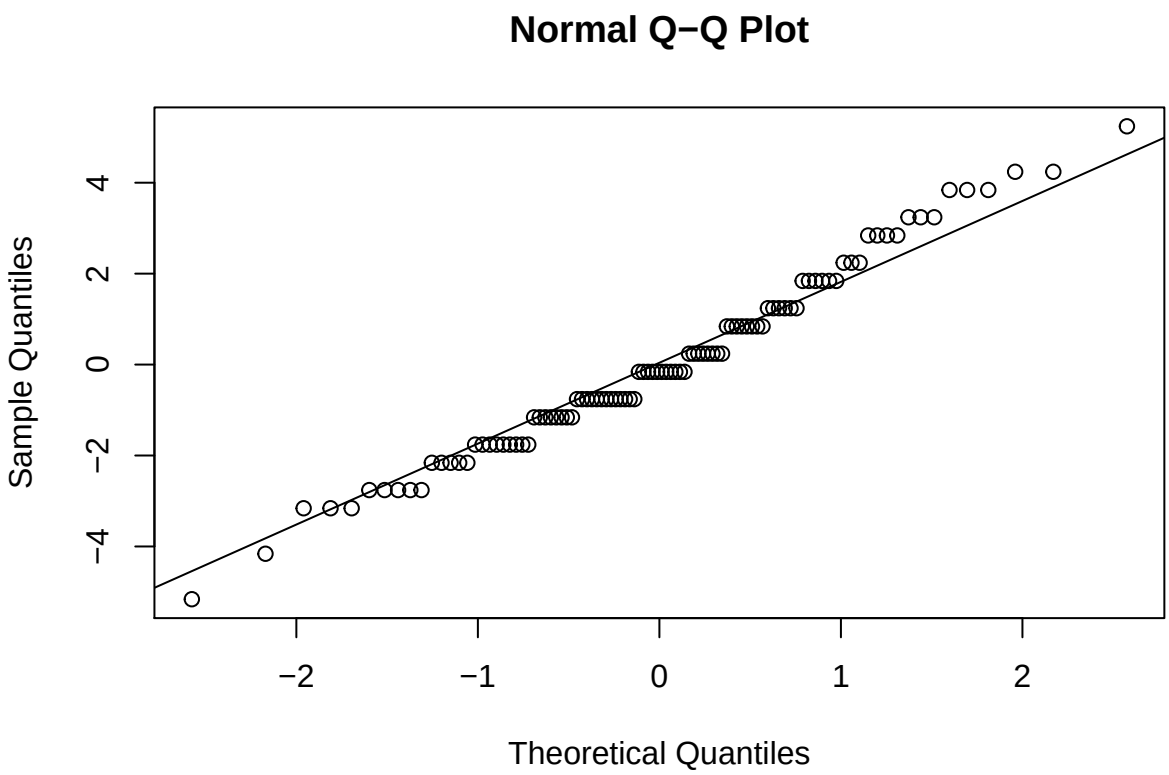
F test statistic and P-value for test of:
H_0: all population means are the same
H_A: at least one population mean differs

```
# Use summary command to display ANOVA table
summary(anova1)
```

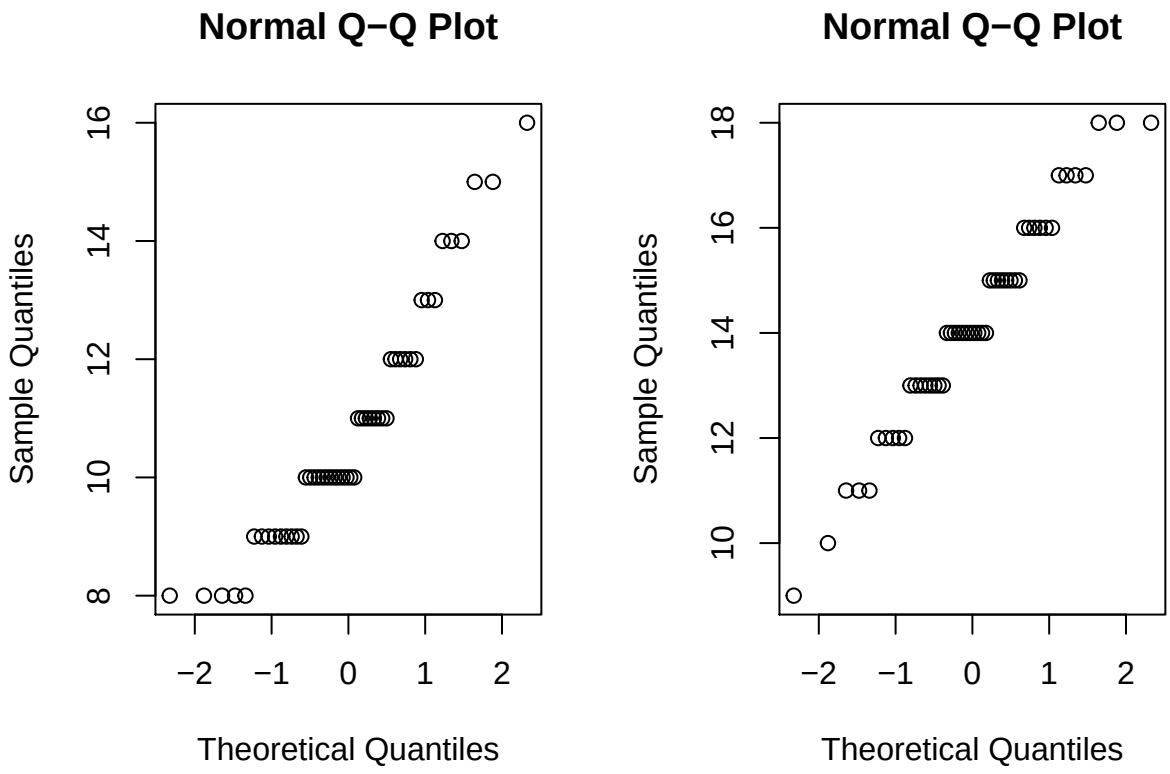
```
##                 Df Sum Sq Mean Sq F value Pr(>F)
## GroupData$Group  1  289.0  289.00   70.13  4e-13 ***
## Residuals       98  403.8    4.12
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Assessing conditions

```
# Produce Normal Quantile (QQ) plots of residuals:
qqnorm(anova1$residuals)
qqline(anova1$residuals)
```



```r
# Produce Normal Quantile (QQ) plots of response for each group:
par(mfrow=c(1,2))
qqnorm(GroupData$Days.with.Symptoms[GroupData$Group=="Nicotinocin"])
qqnorm(GroupData$Days.with.Symptoms[GroupData$Group=="Placebo"])
```

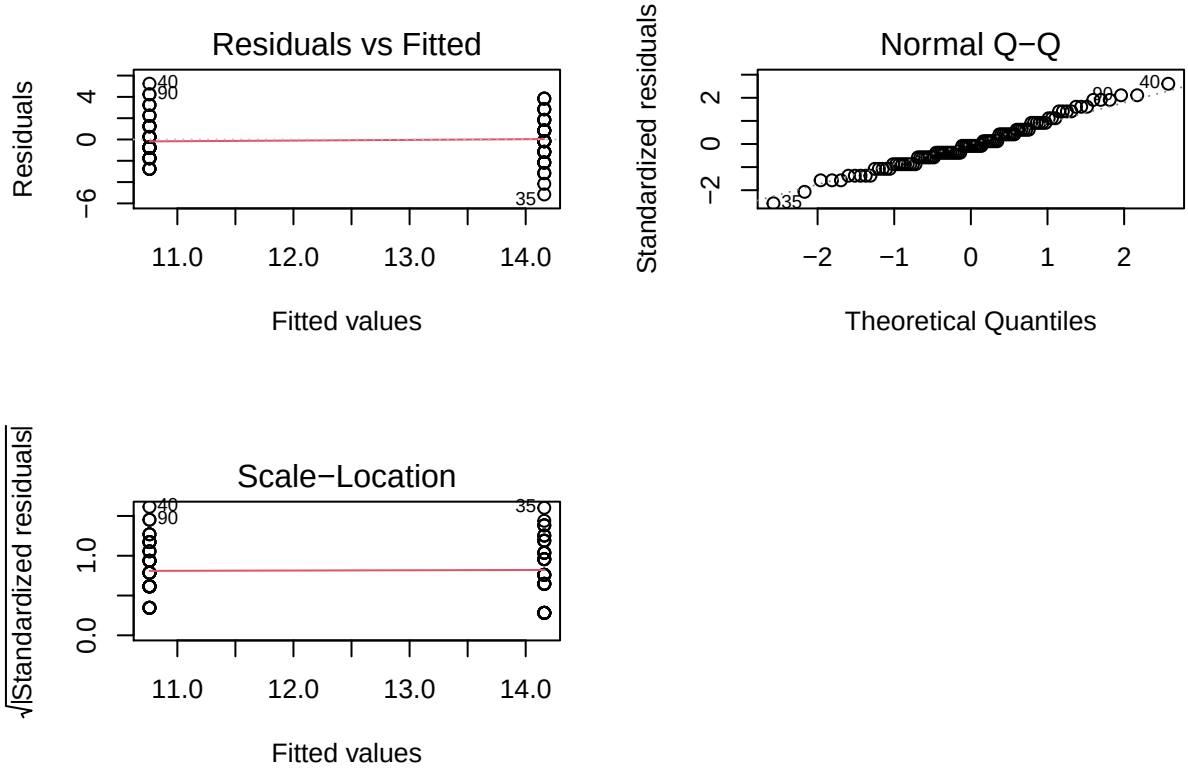



```r
# Produce stripchart of residuals
stripchart(anova1$residuals~GroupData$Group, vertical=TRUE)
```

Instead of starting from scratch for all plots, we can plot anova to display all four graphs

```
# Use plot(anoval1) to display plots for diagnostics
par(mfrow=c(2,2))
plot(anova1)
```

```
## hat values (leverages) are all = 0.02
##  and there are no factor predictors; no plot no. 5
```



Two sample t-tests for pairwise comparisons:

Paired t-test is a parametric approach (or large sample approach) used to compare means of two paired groups (dependent groups or matched groups)

```
#Data vectors: response=Dayswith Symptoms ; factor = Group
pairwise.t.test(GroupData$Days.with.Symptoms, GroupData$Group)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  GroupData$Days.with.Symptoms and GroupData$Group
##
##         Nicotinocin
## Placebo 4e-13
##
## P value adjustment method: holm
```

Tukey's HSD tests and confidence intervals:

TukeyHSD function generates P values and confidence intervals for all possible pairwise populations when combining means of more than two populations. Multiple comparisons helps us decide which Means or groups may differ from the others.

Overall 95% confidence intervals for the difference in Means of pairs B-A and an adjusted P-value are returned:

```
#Conducts all possible pair-wise comparisons for the analysis of variance fit.
TukeyHSD(anova1)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = GroupData$Days.with.Symptoms ~ GroupData$Group)
##
## $`GroupData$Group`
##                       diff      lwr      upr p adj
## Placebo-Nicotinocin  3.4 2.594314 4.205686     0
```

We can see a visual display by using "plot"; using "las=1" to rotate the labels on the y-axis

```
#Summary of anova1 object
plot(TukeyHSD(anova1), las=1)
```