Non-Parametric Modeling

ANTH 3720-001 Archaeological and Forensic Science Lab Methods: Data Analysis with R

Elic Weitzel

Jan. 29-31, 2021

1 Non-Parametric Modeling

We've now covered t-tests, ANOVAs, and Pearson's Correlations and discussed how all are really just specific types of linear models to be used in certain situations. When your data are grouped into two categories, you might be able to use a t-test. When your data are grouped into more than two categories, consider an ANOVA. When you only care about the direction and strength of the association between two variables, perhaps a Pearson's test would work.

But as discussed, all of these tests make assumptions about your data. In fact, all three of these tests make pretty much the same basic assumptions: your data originate from a Gaussian (normal) distribution. When your data are continuous and unbounded, any of these tests could be used. But our data are not always continuous and unbounded, as we learned in the lecture on probability distributions. Sometimes our data are continuous but bounded (from a log normal distribution), or discrete and bounded by 0 and 1 (from a binomial distribution), or discrete integers (from a Poisson distribution). In these cases, the basic assumptions of many of these tests are violated and we might have to try something different.

While we do have several options when our data violate the assumptions of an ordinary Gaussian linear model, one of the simplest things we can do is simply to apply a non-parametric test.

2 Non-Parametric Tests

Non-parametric tests are sometimes called "distribution free" tests. This is because they use methods of calculating statistics that avoid assuming the data originate from any distribution. In all of the tests we've learned thus far, the data are assumed to originate from a Gaussian distribution. This makes them parametric tests: they assume that the data are derived from certain probability distribution parameters. Specifically, a Gaussian or normal distribution in the cases of t-tests, ANOVAs, Pearson's correlation, and ordinarly linear models.

When your data don't come from such a distribution, you have two main choices. You could find a more complex parametric model that uses a more relevant distribution like binomial or Poisson. Or, you could use a simpler model that is non-parametric, and therefore agnostic to the distribution of your data.

Non-parametric tests avoid making assumptions about probability distributions by ranking data ordinally instead of assessing the actual values of your data.

As an example, Group A contains values of 3, 2, 1, and 4. Group B contains 2, 4, 3, and 5. The mean of A is 2.5 while the mean of B is 3.5. As a parametric test, a t-test would compare those real means as if they originated from a normal distribution. A non-parametric test would instead rank the observations of each group in numerical order and compare the ordinal ranks, not the real means of each group. Therefore it doesn't really matter how the data are distributed: the ordinal ranks are the only thing that matter in order for such a test to work.

2.1 Preparing the Data

To learn what to do in such cases, let's load the Ernest Witte Cemetery data from the `archdata` package.

```
library(archdata)
```

```
## Warning: package 'archdata' was built under R version 4.1.2
```

```
data(EWBurials)
```

Let's say we're interested in whether males and females in this skeletal population are the same age or not.

However, the age classes in this dataset are not numerical. They're simply such ordinal categories as "Fetus" or "Middle Adult". For this reason, let's assign a dummy variable to each age class that corresponds to its ordinal rank. Since there are 8 different age classes ranging from Fetus to Old Adult, let's number them 1 through 8. You could also assign a mean age in years if you knew what that were for each age class.

There are a few different ways to assign a dummy variable like this, but the most efficient is to use a function called `recode()` in the `dplyr` package. The `dplyr` package is one of a set of packages in what is known as the *tidyverse*. The tidyverse was conceived of by a famous R programmer named Hadley Wickham, and is intended to improve upon the syntax, efficiency, and data structures of base R. To use this package, you'll have to install it (either with the `install.packages()` function or by clicking Packages > Install) and then load it using the `library()` function. If you try to run this `recode()` function without first installing and loading the `dplyr` package, it won't work: the function only exists in the dplyr package, not in base R.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
EWBurials$DummyAge <- recode(EWBurials$Age, "Fetus" = 1,
                                            "Infant" = 2,
                                            "Child" = 3,
                                            "Adolescent" = 4,
                                            "Young Adult" = 5,
                                            "Adult" = 6,
                                            "Middle Adult" = 7,
                                            "Old Adult" = 8)
```

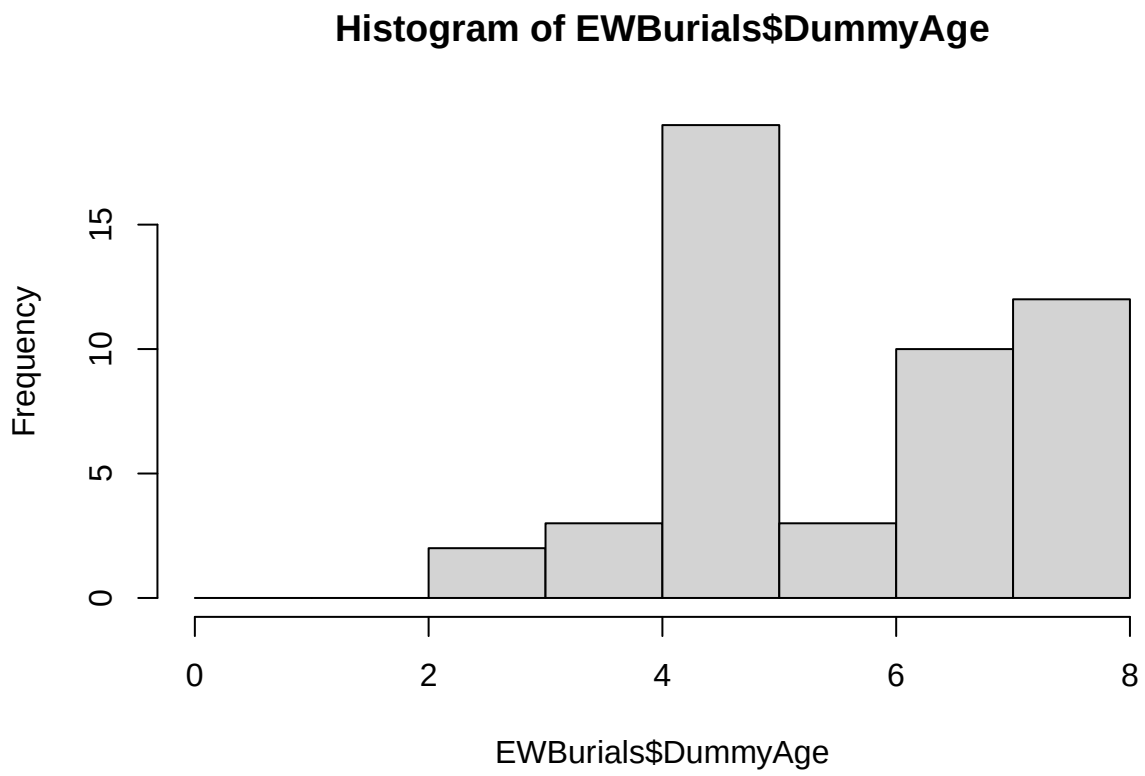
Running this code will create a new column in the EWBurials data frame called "DummyAge" which will contain numbers instead of the character strings for the age classes. You can inspect the EWBurials object by clicking on it or using `head(EWBurials)` to see how this works. To write this code, I looked at all the values of the Age column in EWBurials and made each one equal to the corresponding number.

Now let's get a sense for what different age classes exist and how many individuals belong to each.

```
table(EWBurials$DummyAge)
```

```
##
##  3  4  5  6  7  8
##  2  3 19  3 10 12
```

```
hist(EWBurials$DummyAge, breaks = seq(0, 8, by=1))
```



We can see from the `table()` output that there are no individuals in age classes 1 or 2, only 3-8. The histogram reveals this as well.

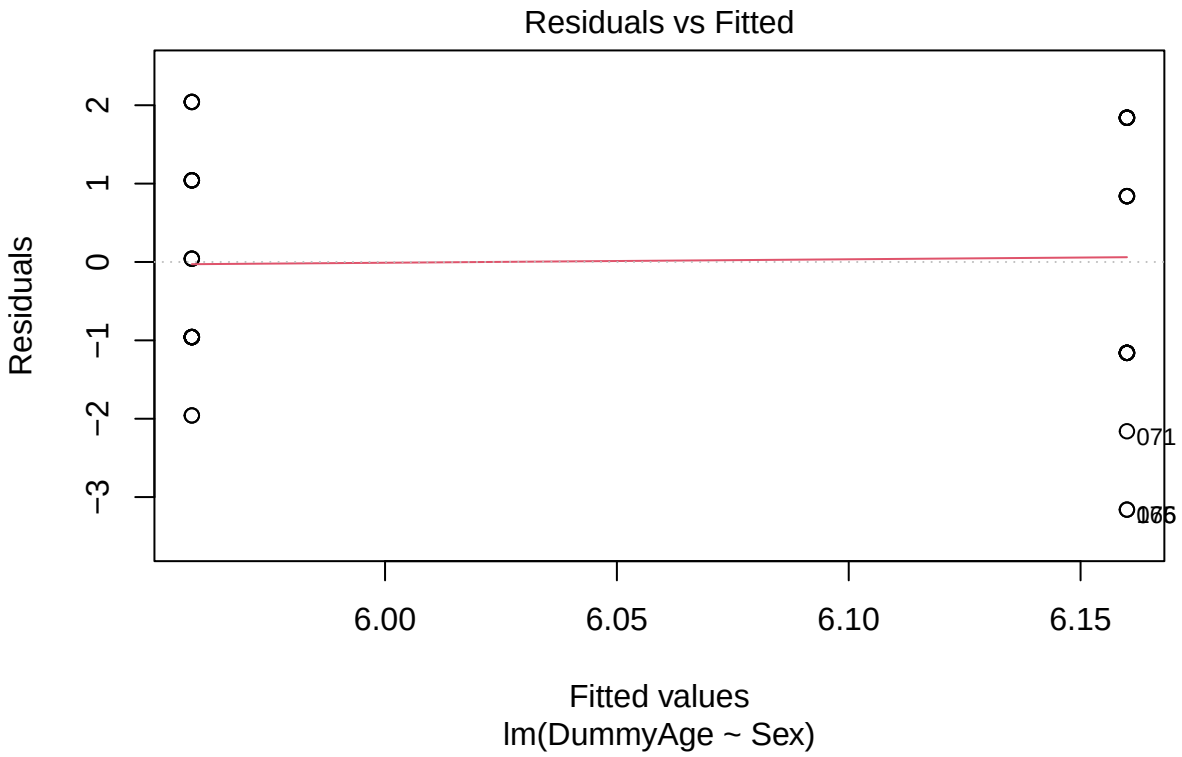
2.2 Why Linear Modeling Won't Work Here

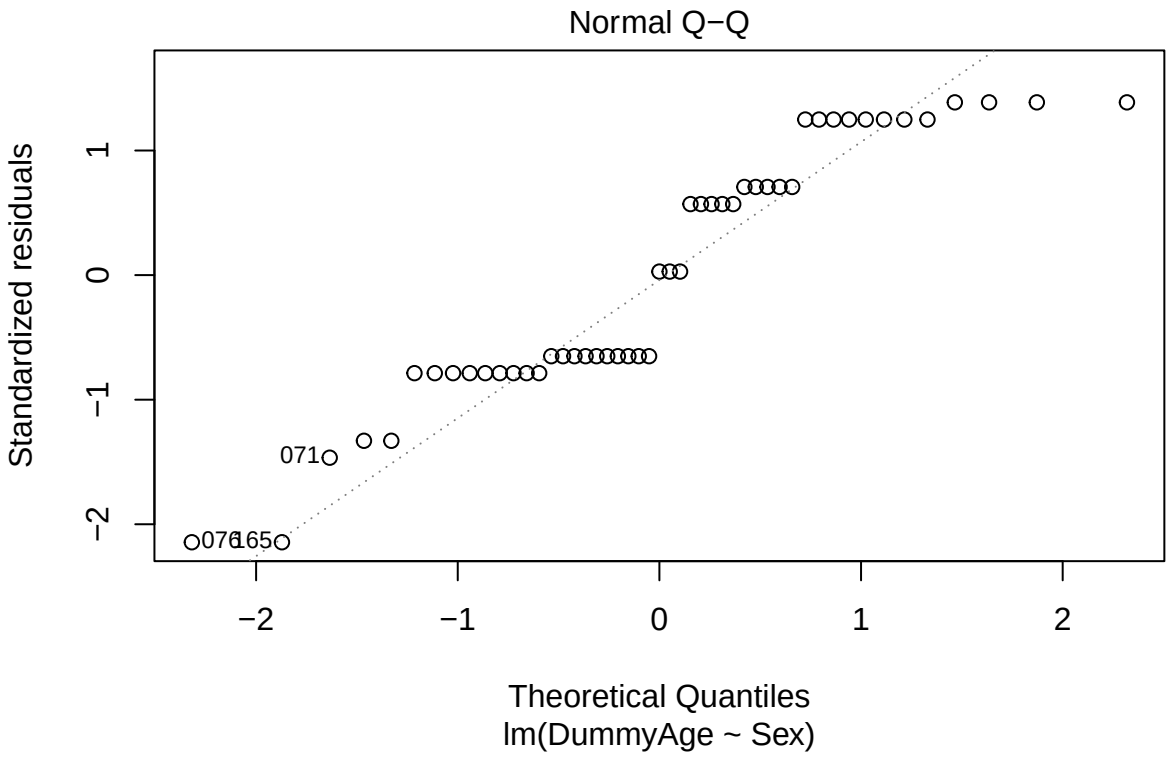
Based on what we learned previously, this would seemingly be a time to use a t-test. We have two groups, males and females, and we want to know if the mean age differs between them. Let's build a linear model to do this so that we can see exactly why this would be a bad choice.

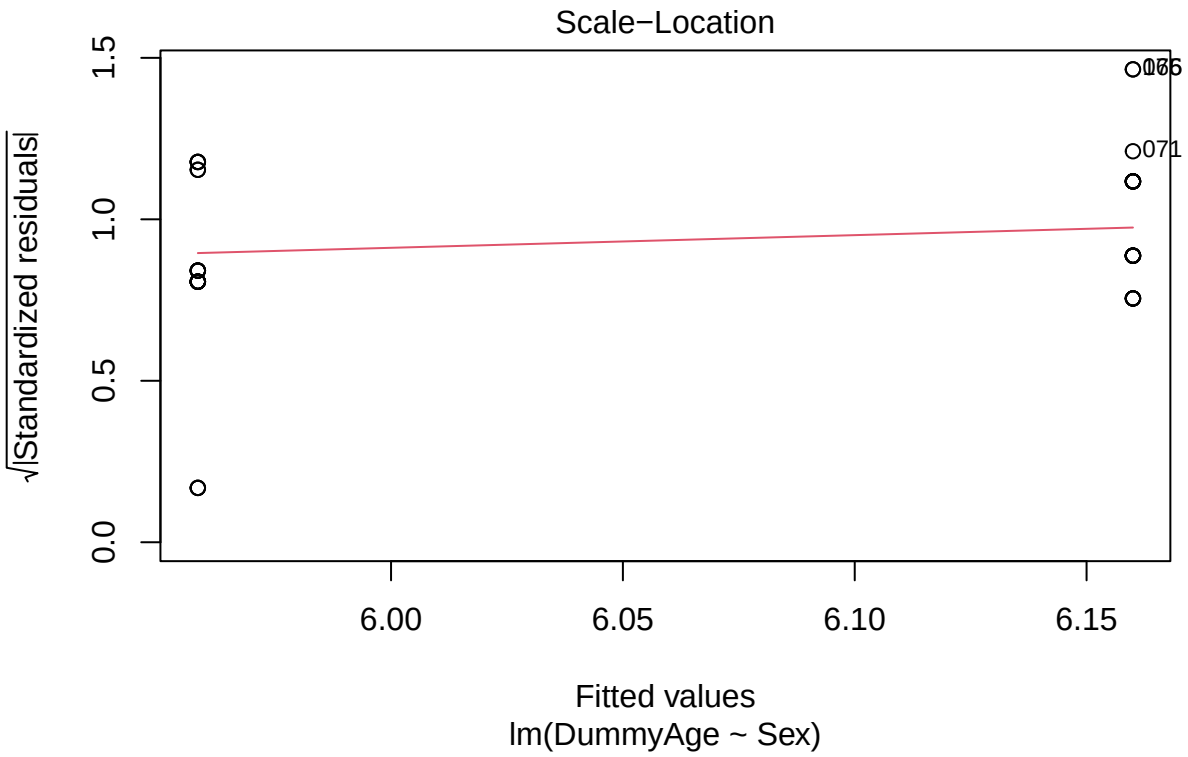
First, build the linear model and then run the diagnostic plots.

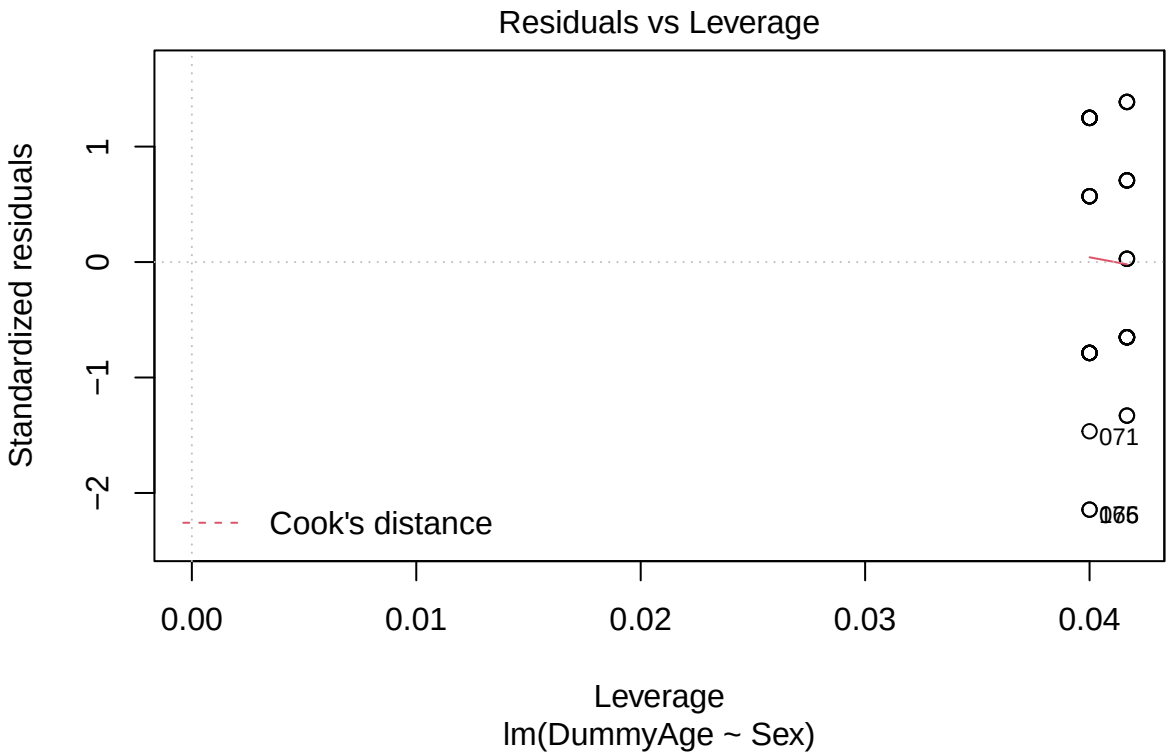
```
bad.mod <- lm(DummyAge ~ Sex, data = EWBurials)

plot(bad.mod)
```


Residuals vs Fitted
Residuals
2
1
0
−1
−2
−3
071
6.00
6.05
6.10
6.15
Fitted values
lm(DummyAge ~ Sex)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(DummyAge ~ Sex)
071
076
165
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
1.5
1.0
0.5
0.0
071
6.00
6.05
6.10
6.15
Fitted values
lm(DummyAge ~ Sex)



The second plot, the q-q plot, is where we most clearly see the problem. Our standardized residuals are not normally distributed. There are several horizontal rows of residuals in this plot: the residuals are not continuously distributed along the dotted line as they need to be for a linear model to be appropriate. Normally distributed residuals need to be continuously distributed around the model fit because a normal distribution is a continuous distribution. Our data are not continuous and the residuals are clearly not meeting the criteria for normality that this technique requires. So what do we do?

2.3 The Mann-Whitney U Test

To more appropriately assess the difference in age between males and females in this skeletal population, we could use the *Mann-Whitney U test*. The Mann-Whitney U test is a non-parametric alternative to a t-test. It works on two categorical groups of data and assesses differences using rankings. We can run such a test using the `wilcox.test()` function.

```
wilcox.test(EWBurials$DummyAge ~ EWBurials$Sex)
```

```
## Warning in wilcox.test.default(x = c(7, 5, 7, 8, 5, 7, 8, 6, 8, 6, 7, 5, :
## cannot compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  EWBurials$DummyAge by EWBurials$Sex
## W = 272.5, p-value = 0.5731
## alternative hypothesis: true location shift is not equal to 0
```

We use the `wilcox.test()` function because of similarities between the Mann Whitney U test and another set of tests called Wilcoxon tests. You can read more about this family of tests on your own, but we won't get into it here.

The result of this test shows that there is no significant difference in age between males and females in this skeletal assemblage (W = 272.5, p = 0.5731).

You'll note that on degrees of freedom are reported in this output. This is because we're now dealing with non-parametric tests, which don't make any assumptions about the probability distributions from which data come. By ranking data and ignoring distributions, degrees of freedom and the test statistics of t and F that we're more familiar with become obsolete.

There is a warning returned as well that says that the function `cannot compute exact p-value with ties`. This means that some values in the model are tied in rank: the same value appears numerous times and thus has the same rank in each case, which makes the calculations less exact. This isn't usually a problem, but you should read about what this warning message means on your own if you use this test and encounter this issue.

As a note, an alternative to the Mann Whitney U test is the Kolmogorov-Smirnov test (`ks.test()`). This test is also a non-parametric alternative to a t-test, but is slightly different from a Mann Whitney U test. Either one usually gets you where you need to go, and often the choice of using one over the other comes down more to disciplinary precedent than anything else.

2.3.1 Mann-Whitney U as a Linear Model

As discussed previously, pretty much all of the basic statistical tests you'll encounter are just special cases of a linear model. The Mann-Whitney U test is no exception.

We can program a linear model to do basically the same thing that a Mann Whitney U test is doing by simply ranking the response variable - `DummyAge` in this case - and checking out the p-value in the summary output.

```
mannwhitmod <- lm(rank(DummyAge) ~ Sex, data = EWBurials)
summary(mannwhitmod)
```

```
##
## Call:
## lm(formula = rank(DummyAge) ~ Sex, data = EWBurials)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.600 -11.100   2.146   8.646  19.646
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   23.854      2.815   8.474 5.04e-11 ***
## SexMale        2.246      3.941   0.570    0.571
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.79 on 47 degrees of freedom
## Multiple R-squared:  0.006862,   Adjusted R-squared:  -0.01427
## F-statistic: 0.3248 on 1 and 47 DF,  p-value: 0.5715
```

Our p-value from the `wilcox.test()` function was 0.5731, and our p-value from this linear model equivalent using `lm()` is 0.5715. There is a slight difference between the two p-values since the `wilcox.test()` function

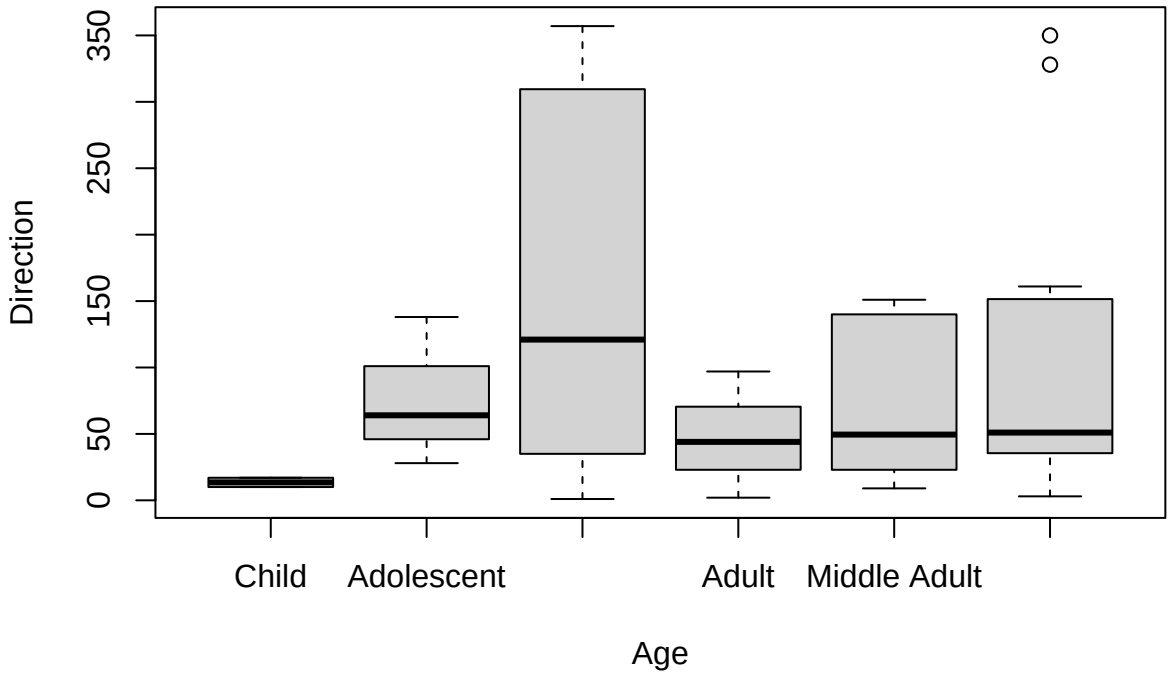
isn't literally doing the same thing, but conceptually, these are equivalent models. All that a Mann Whitney U test is doing is ranking the values of your response variable and comparing the ranks between the two groups instead of the true values.

2.4 The Kruskal-Wallis Test

The Mann-Whitney U test is a good alternative to a t-test when assumptions of the test are violated, but what about ANOVAs?

To explore a case where an ANOVA isn't appropriate, let's analyze the direction each individual was buried (in degrees east of north) as a function of age class. As usual, let's visually inspect these data before building any sort of model.

```
boxplot(Direction ~ Age, data = EWBurials)
```

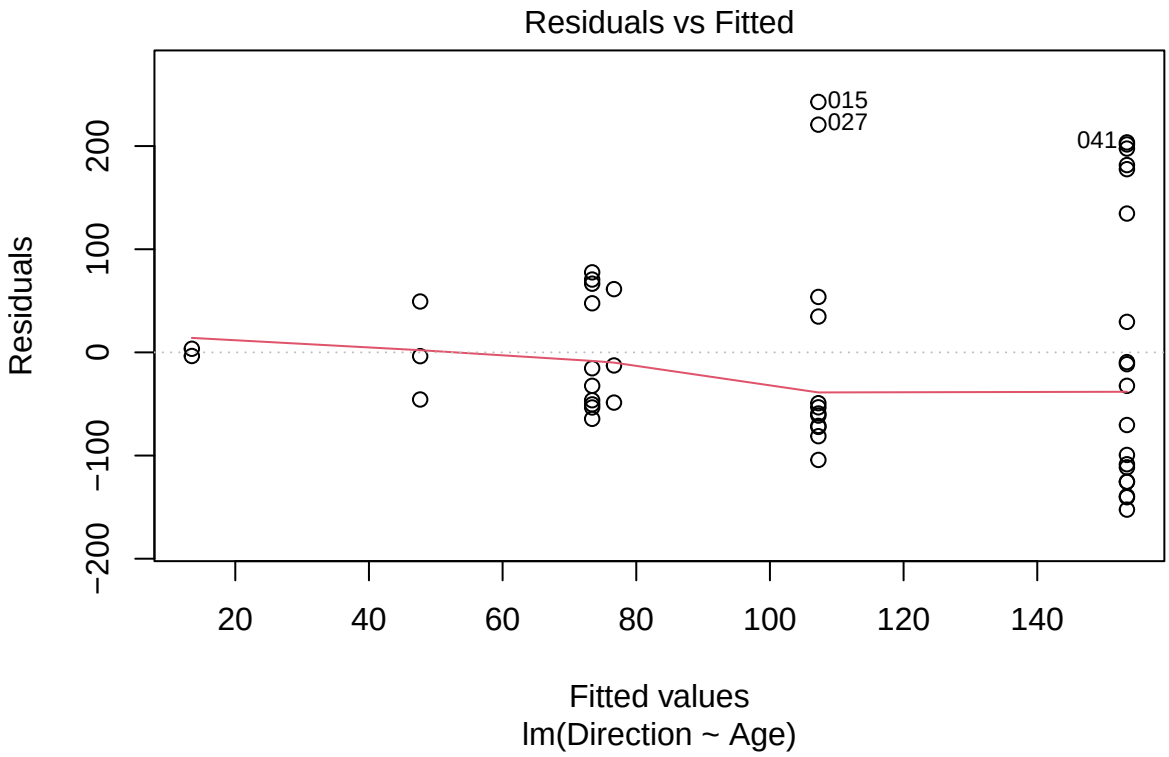


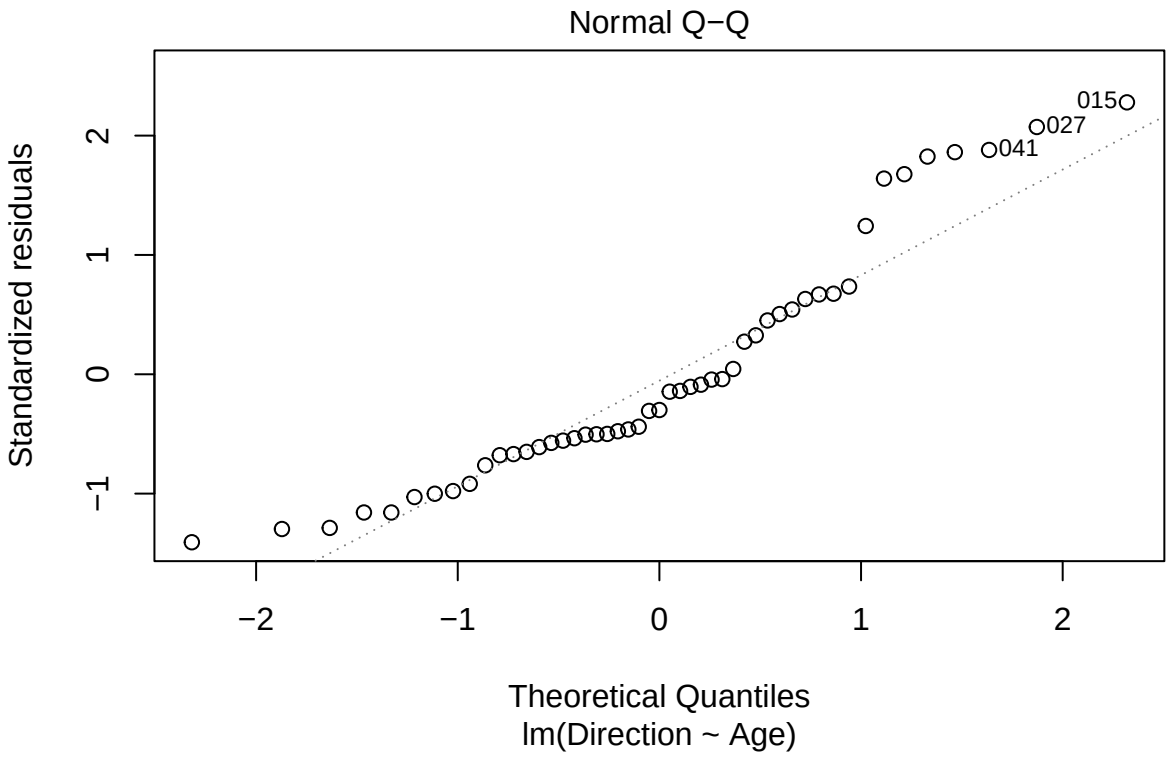
It looks like there may be some differences here between the groups, but these also might not be significant. The boxplots overlap by a good degree in all cases except for the Child category, but we know that category only has two data points in it. This means that a model will have a hard time determining whether this lower mean Direction value for the Child category is a fluke or not, and will probably say that it is and that the Child category may be the same as the rest.

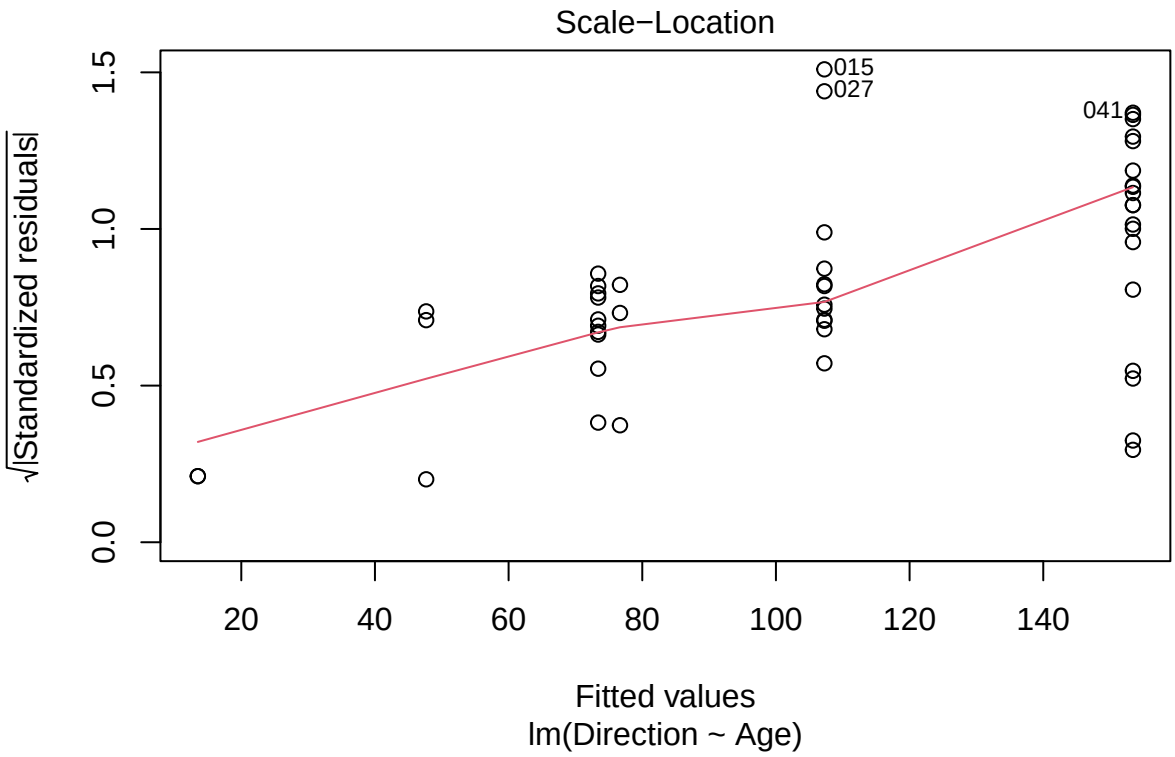
But let's build a linear model of this relationship and then inspect the diagnostic plots.

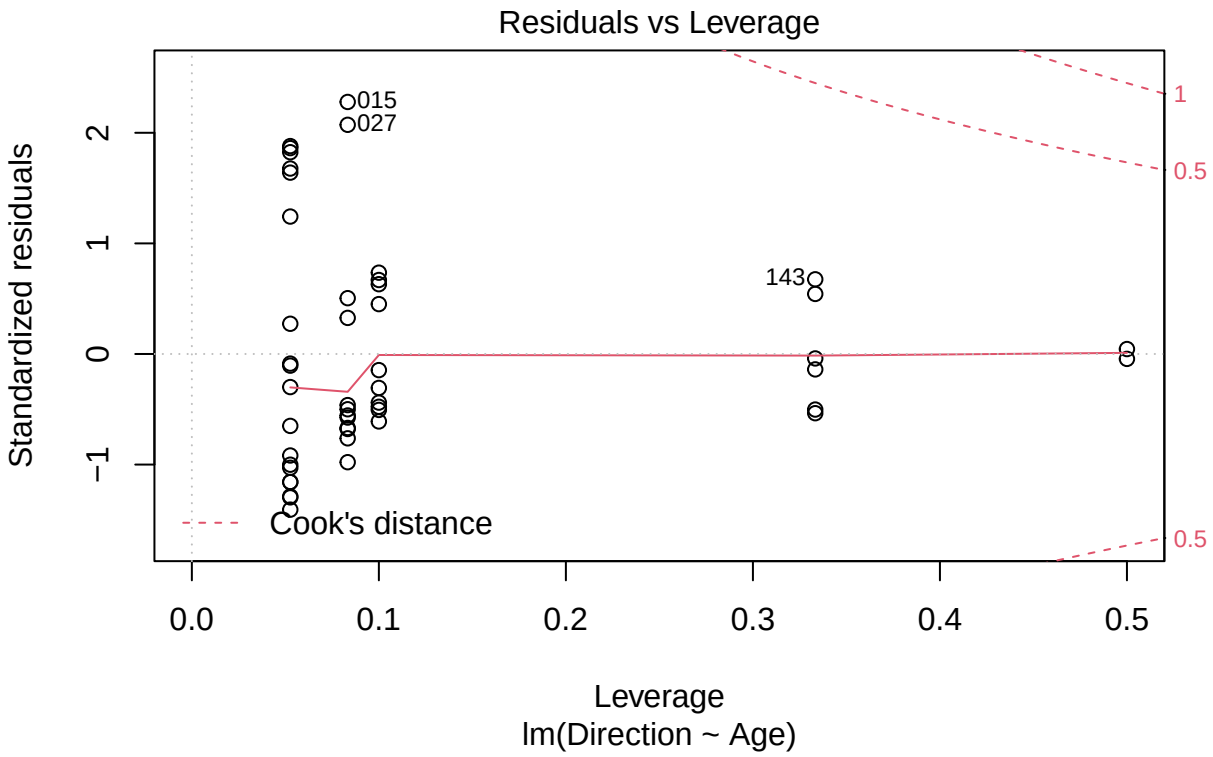
```
bad.mod.2 <- lm(Direction ~ Age, data = EWBurials)

plot(bad.mod.2)
```


Residuals vs Fitted
Residuals
200
100
0
−100
−200
015
027
041
20
40
60
80
100
120
140
Fitted values
lm(Direction ~ Age)


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Direction ~ Age)
015
027
041
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
015
027
041
20
40
60
80
100
120
140
Fitted values
lm(Direction ~ Age)



In these plots, specifically the first and third, you can see that there is a problem: heteroskedasticity. The residual variance increases with the fitted values. This violates a key assumption of an ANOVA and makes interpreting such a test problematic. It might not be a huge problem, but if we can, we should probably avoid using an ANOVA in this case...

A common non-parametric version of the ANOVA is the Kruskal-Wallis test. It works in a similar way to the Mann Whitney U test, by ranking observations to avoid making assumptions about distributions. We can run this test using the `kruskal.test()` function.

```
kruskal.test(Direction ~ Age, data = EWBurials)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Direction by Age
## Kruskal-Wallis chi-squared = 6.3314, df = 5, p-value = 0.2753
```

This test tells us that there is no significant difference in direction between the various age classes ($X2 = 6.33$, $df = 5$, $p = 0.275$). Indeed, as we saw in the boxplots of these data, while the Young Adult category had some high values and the Child category had a low mean, there is enough overlap between all age classes that they're not significantly different from each other.

2.4.1 Post Hoc Tests

Just as with ANOVAs, if the p-value from your Kruskal-Wallis test is significant, you may want to then know which pairwise comparison is driving that result. In the example above, our p-value wasn't significant so there's no reason to do a post hoc test. But let's explore a case where there is a significant difference.

2.4.1.1 Prepare the Data Let's pretend that there is a reason for us to divide these data up into four groups, called A, B, C, and D. We're going to cook the data here to illustrate the point. Here, we index the EWBurials dataset so that all rows with Direction values of less than 27 degrees are assigned to group A. Then the rest of the rows are divided up into groups B, C, and D as specified.

```
A <- EWBurials[EWBurials$Direction < 27, ]
BCD <- EWBurials[EWBurials$Direction >= 27, ]
B <- BCD[1:12, ]
C <- BCD[13:25, ]
D <- BCD[26:38, ]
```

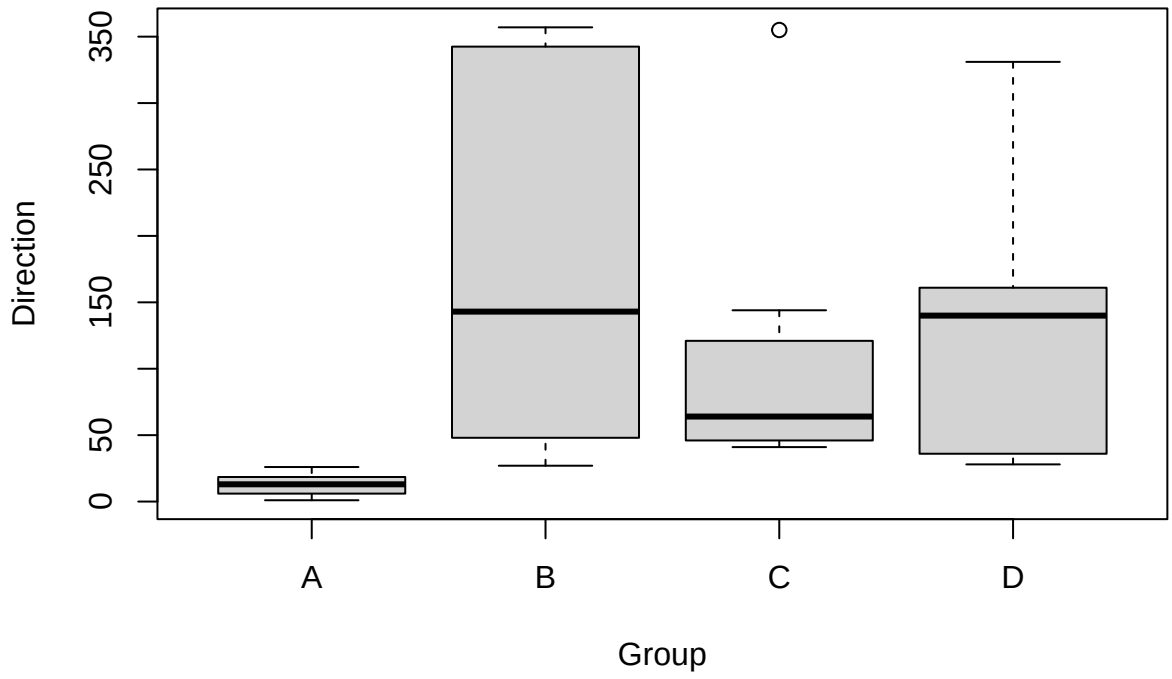
Now, let's make it so that the Group column for each group is labeled with that group's letter. Then, use the `rbind()` function, which stands for "row bind", to combine these four objects into one object called `burial.data`.

```
A$Group <- "A"
B$Group <- "B"
C$Group <- "C"
D$Group <- "D"

burial.data <- rbind(A, B, C, D)
```

Boxplots of the data suggest that there may be quite a big difference between Group A and the other three groups (since I deliberately divided up the groups so that this would occur...).

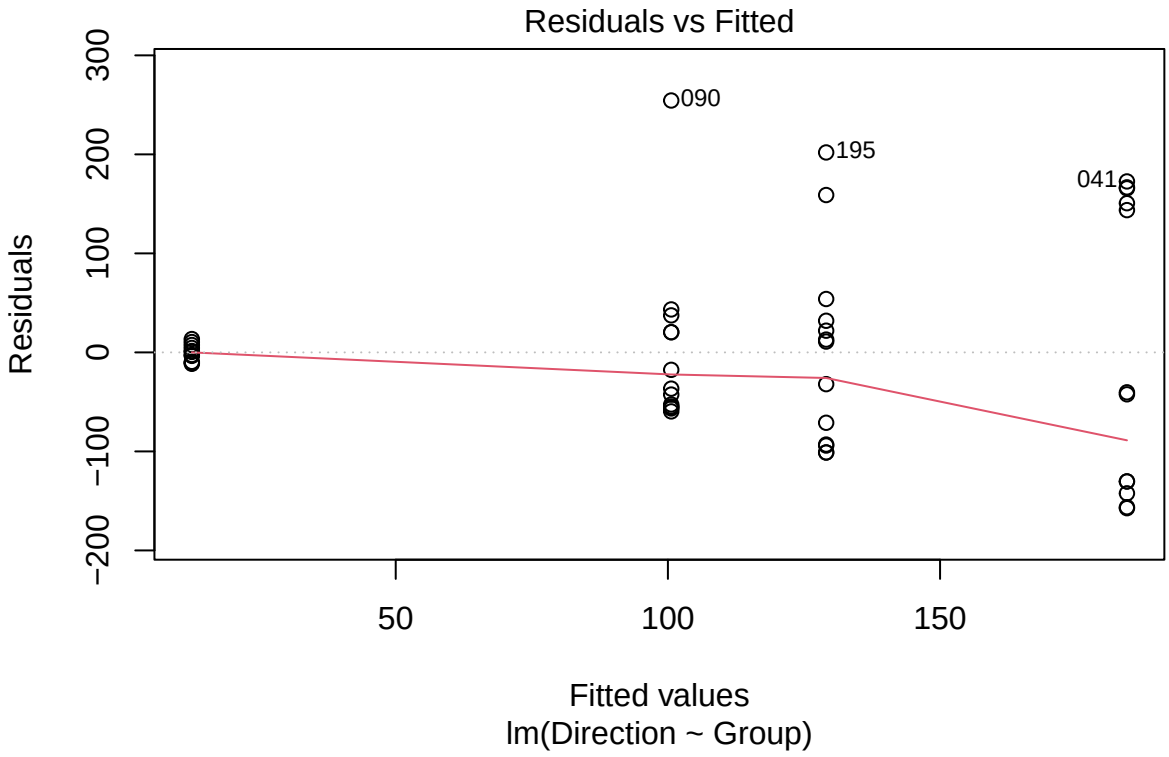
```
boxplot(Direction ~ Group, data = burial.data)
```

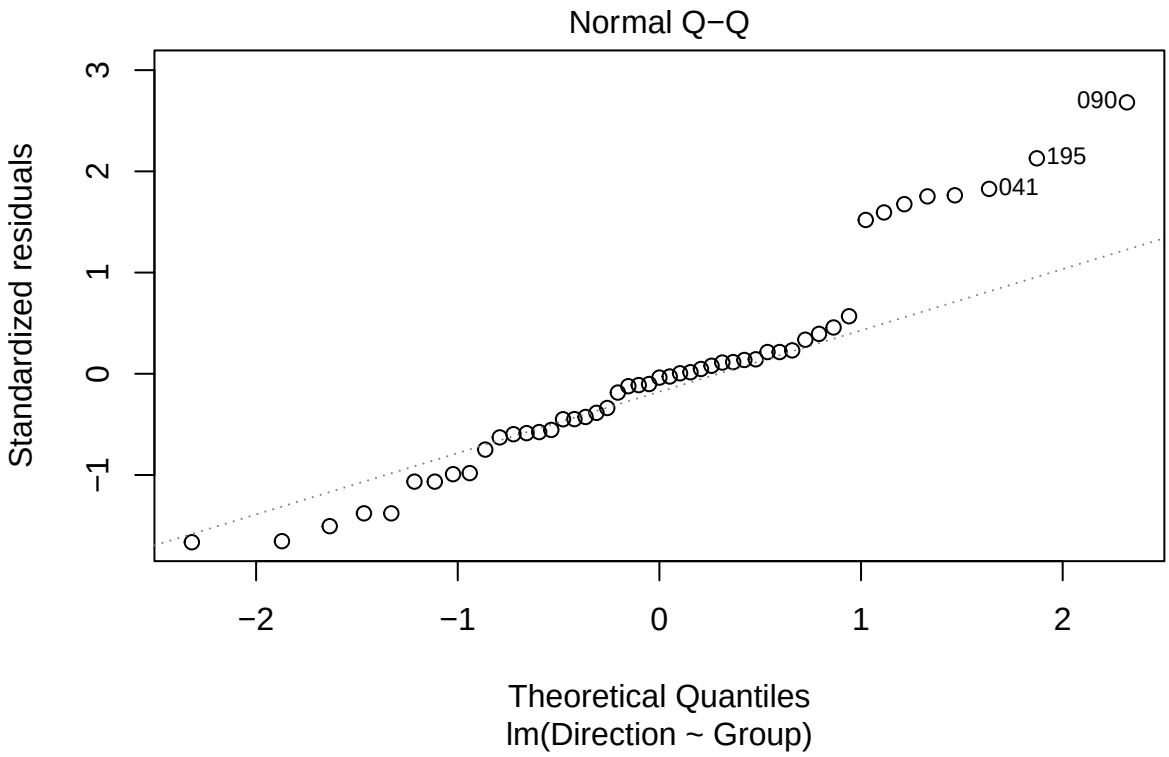


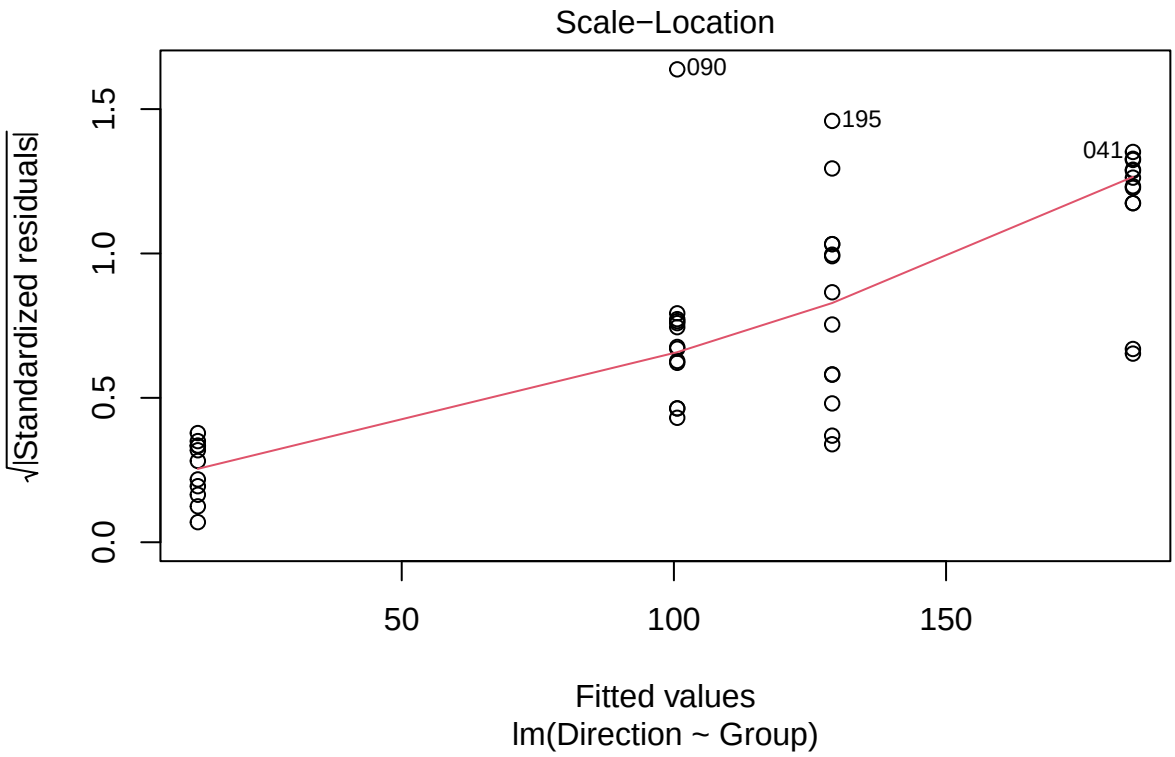
2.4.1.2 Run the KW Test With our newly grouped dataset, let's try to build a regular linear model under the assumptions of a normal distribution (an ANOVA).

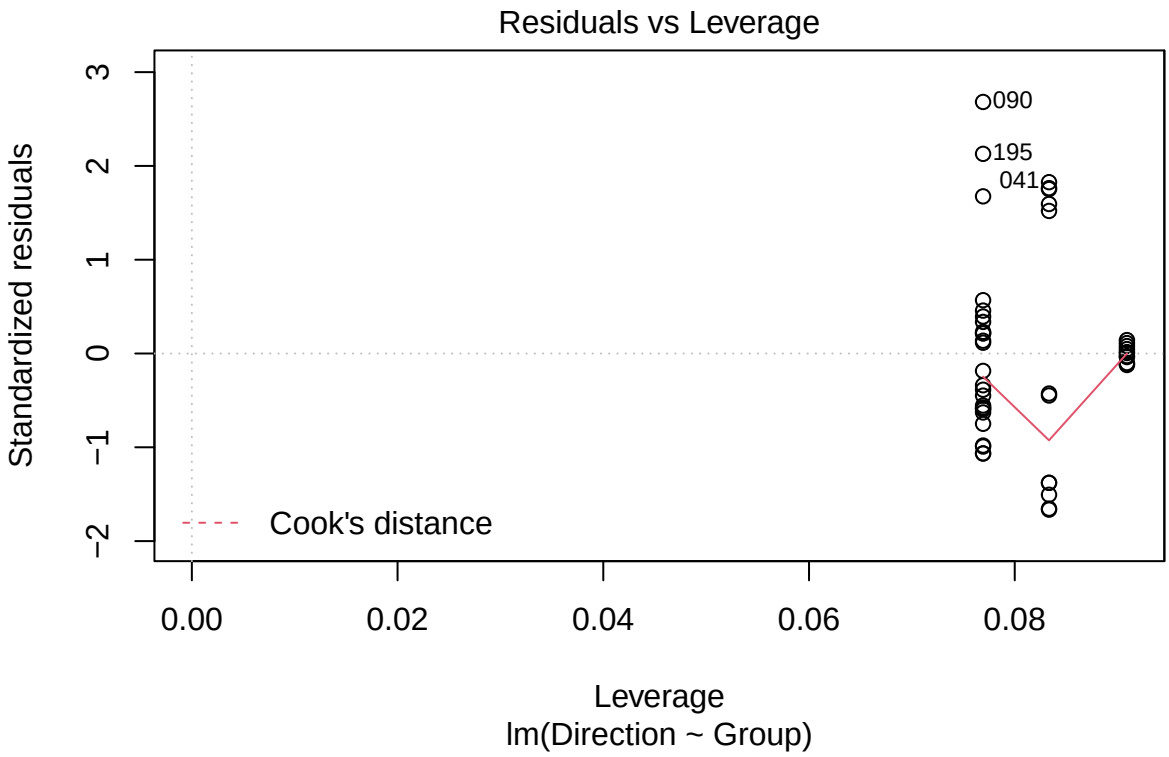
```
bad.mod2 <- lm(Direction ~ Group, data = burial.data)

plot(bad.mod2)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Direction ~ Group)
090
195
041
−2
−1
0
1
2
3


Scale−Location
090
195
041
√|Standardized residuals|
0.0
0.5
1.0
1.5
50
100
150
Fitted values
lm(Direction ~ Group)



The third diagnostic plot shows that we have a problem with heteroskedasticity: the residuals are not evenly distributed around the model fit. So a better option could be a non-parametric Kruskal-Wallis test.

```
kruskal.test(Direction ~ Group, data = burial.data)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Direction by Group
## Kruskal-Wallis chi-squared = 25.774, df = 3, p-value = 1.064e-05
```

This Kruskal-Wallis test shows that there is a significant difference between these four groups ($X2 = 25.77$, $df = 3$, $p < 0.0001$).

2.4.1.3 Run the Post Hoc Test (Dunn's Test)

Now that we know there's a significant difference between the groups, but we don't know which groups, we should run a post hoc test.

With ANOVAs, we can run pairwise t-tests with a multiple comparisons correction. But with a Kruskal-Wallis, we can't run t-tests: we've already shown that our data don't meet the assumptions necessary for a parametric test that assumes normality. So we need something else.

There are of course many options, but one common one is called Dunn's Test. To use this test, you'll need to install the `dunn.test` package and then use the `library()` function to load this package.

```
library(dunn.test)
```

```
## Warning: package 'dunn.test' was built under R version 4.1.1
```

```
dunn.test(burial.data$Direction, burial.data$Group, method="bh")
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 25.7736, df = 3, p-value = 0
##
##
##                            Comparison of x by group
##                              (Benjamini-Hochberg)
## Col Mean-|
## Row Mean |          A          B          C
## ---------+---------------------------------
##        B |  -4.541846
##          |    0.0000*
##          |
##        C |  -3.824893   0.821635
##          |    0.0001*     0.3085
##          |
##        D |  -4.140348   0.498809  -0.329482
##          |    0.0001*     0.3707     0.3709
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

The output in the R Console shows the results of this Dunn's Test.

In our function above, we specified that our p-value adjustment method should be "bh", which stands for Benjamini-Hochberg. As with post hoc tests for ANOVAs, we need to account for the multiple comparisons problem and the increasing likelihood of false positives with all of these pairwise tests. There are numerous alternative methods besides Benjamini-Hochberg, and you should read up on them to see what makes sense for you, but Benjamini-Hochberg is a common one.

The `dunn.test()` function starts off by actually running a Kruskal-Wallis test on your data. We see our same chi-squared, degrees of freedom, and p-value from above repeated here (with rounding error).

Then the function returns a table, just as the pairwise t-test function that we used did. There are two values in this table. The value on top is the column mean minus the row mean. This helps you to know which mean is higher or lower when comparing the groups listed in the rows and columns.

Then, the second value is the p-value. This package is kind enough to include asterisks for significant values, so we can easily see that we have three significant p-values and three non-significant ones. The differences between A and B, A and C, and A and D are all significant, and very strongly so. However, the differences between groups B and C (0.3085), B and D (0.3707), and C and D (0.3709) are not significant. This is exactly what we expected given the look of the boxplots we created above.

So now we know that it's the orientations of Group A burials that are different from the other groups!

2.4.2 Kruskal-Wallis as a Linear Model

The Kruskal-Wallis test is, of course, just a specific variety of linear model. If you still need more proof, we can re-run our original `kruskal.test()` function from above and compare it to a linear model equivalent using the `lm()` function.

```
kruskal.test(Direction ~ Age, data = EWBurials)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Direction by Age
## Kruskal-Wallis chi-squared = 6.3314, df = 5, p-value = 0.2753
```

```
kwmod <- lm(rank(Direction) ~ Age, data = EWBurials)
summary(kwmod)
```

```
##
## Call:
## lm(formula = rank(Direction) ~ Age, data = EWBurials)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.263 -10.263   0.875  11.850  20.375
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)         6.500      9.944   0.654    0.517
## AgeAdolescent      18.500     12.837   1.441    0.157
## AgeYoung Adult     22.763     10.454   2.177    0.035 *
## AgeAdult           10.833     12.837   0.844    0.403
## AgeMiddle Adult    15.650     10.893   1.437    0.158
## AgeOld Adult       19.125     10.740   1.781    0.082 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.06 on 43 degrees of freedom
## Multiple R-squared:  0.1319, Adjusted R-squared:  0.03096
## F-statistic: 1.307 on 5 and 43 DF,  p-value: 0.2791
```

Our kruskal.test() function returns a p-value of 0.2753 while our lm() function returns a p-value of 0.2791. Again, not perfectly equivalent (since the former uses a chi-square distribution and the latter an F distribution), but they're close enough to prove the concept: the Kruskal-Wallis test is just a specific form of linear model.

2.5 Spearman's Correlation

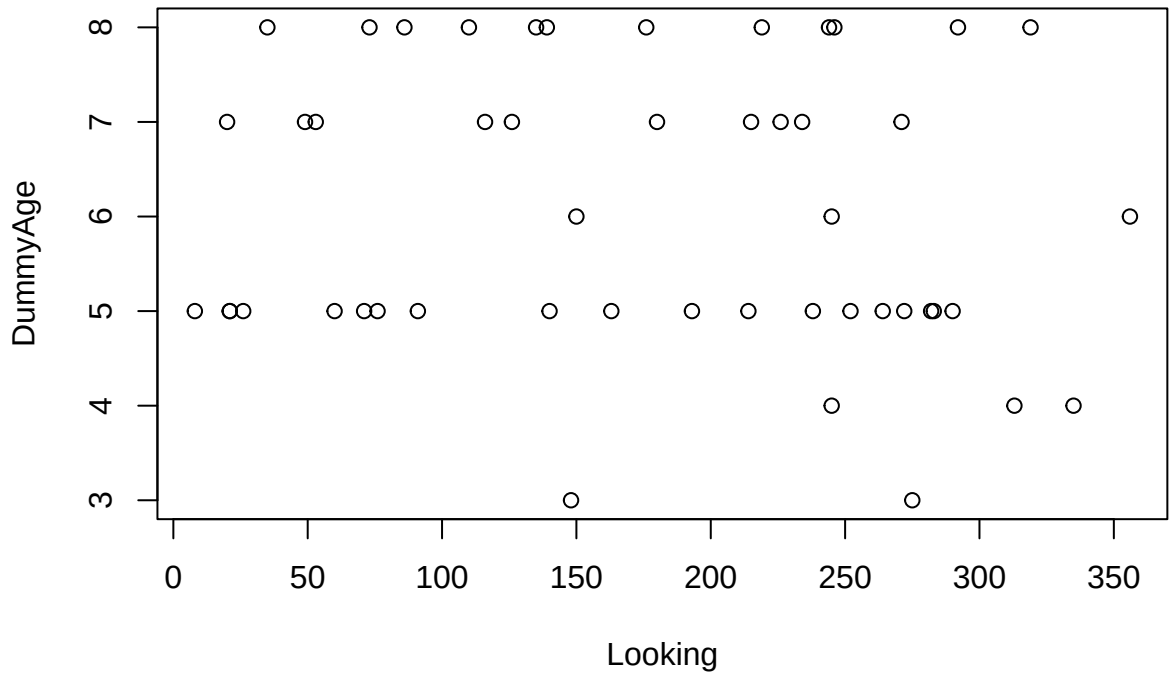
Now that we've covered the non-parametric versions of t-tests and ANOVAs, we can introduce a non-parametric correlation analysis: Spearman's Correlation.

Spearman's test accomplishes the same goal as Pearson's, but does so non-parametrically by ranking the observations ordinally and correlating the ranked values instead of the real values. For this reason, you may sometimes see it referred to as Spearman's rank-order correlation test.

With the EWBurials dataset we've been using, let's examine the correlation between the direction a burial is looking and age. In this dataset, age is an ordinal variable: it's rank-order. Adolescents are older than children and Old Adults are older than Young Adults, but we don't actually know by how much. The real values of age aren't provided, only these ordinally-ranked groups. This means that Spearman's test could potentially be quite useful here since it deals with rank-order correlations.

Let's plot the relationship between age and the direction a burial is looking using the `DummyAge` variable that we made.

```
plot(DummyAge ~ Looking, data = EWBurials)
```



It looks like there could be a relationship here, but if there is, it's probably pretty weak and not particularly meaningful. The overall trend appears to be negative, but given the spread in the data, it might not be strong enough to be statistically meaningful. Let's run the analysis and see!

We can run Spearman's Correlation using the same function we used to run Pearson's: `cor.test()`. But now, since we're running a non-parametric version, we have to add an argument. This time around, we'll set the `method` argument equal to `"spearman"` as follows.

```
cor.test(EWBurials$DummyAge, EWBurials$Looking, method = "spearman")
```

```
## Warning in cor.test.default(EWBurials$DummyAge, EWBurials$Looking, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
```

```
##
## data:  EWBurials$DummyAge and EWBurials$Looking
## S = 22421, p-value = 0.3239
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.1439143
```

We now see the output for a Spearman's rank-order correlation test. The statistics for this test get reported a little bit differently. Now we have an S value of 22421 and a p-value of 0.3239. That's not significant, so it appears that there is no significant relationship between age and the direction a burial is looking.

At the bottom of the output, we see the statistic `rho` (ρ) is equal to -0.1439. This is our correlation coefficient for a Spearman's test: ρ = -0.144. A correlation coefficient of -0.144 is negative as we suspected based on the plot, but it's pretty weak. Based on that p-value and rho statistic, there's seemingly not much going on between age and the direction a burial is looking.

2.5.1 Spearman's Test as a Linear Model

As I'm sure you're tired of hearing by now, surprise! Spearman's test is also just a specific type of linear model. We can once again build an equivalent model using the `lm()` function to prove this point, and simply rank the data we plug into it.

```
spearmod <- lm(rank(DummyAge) ~ rank(Looking), data = EWBurials)
summary(spearmod)
```

```
##
## Call:
## lm(formula = rank(DummyAge) ~ rank(Looking), data = EWBurials)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.0517 -11.5171   0.5862   9.4309  21.5343
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    28.4481     3.9734   7.160 4.68e-09 ***
## rank(Looking)  -0.1379     0.1383  -0.997    0.324
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.69 on 47 degrees of freedom
## Multiple R-squared:  0.02071,    Adjusted R-squared:  -0.0001246
## F-statistic: 0.994 on 1 and 47 DF,  p-value: 0.3239
```

From this rank-ordered linear model, we get a p-value of 0.3239 and an r^2 of 0.02071. Because the correlation coefficient r is simply the square root of the coefficient of determination r^2, we can convert one to the other quite easily.

```
sqrt(0.02071)
```

```
## [1] 0.1439097
```

By taking the square root of our r^2 value of 0.02071, we get our original r value of 0.1439 from the `cor.test()` function. This rank-transformed linear model of our data is equivalent to Spearman's correlation!

3 Problems with Non-Parametric Tests

As discussed, when your data don't originate from a Gaussian distribution, the common set of parametric tests (t-tests, ANOVAs, etc.) become problematic. This is because they generally assume your data do originate from a normal distribution. So if this main assumption (and all of the many other assumptions that go along with it) is violated, non-parametric tests can be a good alternative.

But non-parametric tests are not without their own problems.

The main problem is that these tests are weak. Because they rank observations instead of actually assessing the real values, you end up with a less powerful statistical test. Ranking observations makes non-parametric tests good at overcoming the problems of incorrectly assuming a certain distribution, but it also makes them weaker because they assume no distribution at all. This means that you're more likely to get an erroneous result from them, such as failing to get a significant p-value even when the null hypothesis is false and you should.

One potential solution to this problem is to continue to use parametric statistics, but simply change the assumed probability distribution you use. Not all linear models assume a Gaussian/normal distribution. Other types of models utilize Poisson or binomial or other distributions and therefore better model your data than a non-parametric test can because they elect to specify an appropriate distribution instead of ignoring distributions entirely. This class of linear models are called generalized linear models (GLMs) and are the subject of future tutorials.