Stats 109

Final Project - 5/3/2021

William (Bill) Pottle

Validation of Energy Impact Sensing Vests for Martial Arts

Abstract

Marital Arts has always been plagued by a central question – how is it possible to practice dangerous things safely? For martial sports, the question turns into a closely related one - how can we judge which competitor is better able to defend themselves, without the other competitor suffering injury? Certain organizations, for instance the UFC, allow full contact. While the matches have a comparatively high degree of realism, they require a ringside doctor and are not scalable to the broader population. This has led to the development of protective vests (*hogus*) that measure the impact of a strike on the body of an opponent. Electronic Hogus from 2 major brands (Daedo and 2020 Armor) and helmets from Daedo were repeatedly tested by dropping known weights from various heights in order to answer several key questions. 1) Which brand has more variation within a hogu? 2) Which brand has more variation between 2 hogus? 3) Is there a statistically significant difference in how difficult it is to score on 2 different hogus from the same brand? 4) Is there a linear model which relates the energy of a blow to a reading for each brand? 5) Is there a linear model which relates the reading on the helmet to the force of the blow?

This project might be different from others in the class in that it relies on an experimentally generated dataset. However, as someone who owned a martial arts school for 15 years and as a promoter of the most popular Olympic Style Taekwondo tournament series in Colorado, this topic is one that I am very passionate about.

Besides this, there are several other key reasons why this project was chosen.

- A study like this has never been published. All previous data has only been used internally by private companies. While the results of the vest readings are of course available to their owners, users have no idea how much trust they should place in those readings.

- *Daedo scores only in dimensionless units.* Any type of conversion to energy could help thousands of schools, tournaments, referees, and medics around the world connect a certain reading to the actual energy delivered. Right now, if someone shows up to a hospital after taking a blow to the head, the only data that can be relayed to a doctor is "*The blow was a level 17*". This gives the doctor no actionable information verses knowing the actual energy that was delivered.

- The company Swiss Timing is responsible for all timing events at the Olympics until the 2032 games. Sometime during 2022, they will choose a product for Taekwondo scoring for the 2024 games. While they will conduct their own investigation, the data from this project will be made available to them as well as the companies producing vests. Daedo has been used for the 2012 and 2016 games[1].

- Energy Scoring Leagues are popping up and becoming more common. In order for them to gain in popularity, fans will need to trust the statistics. For instance, in American Football if a play went for 9 yards, there is no ambiguity in that statement.

- Independent validation of energy impacts to the head could help this technology be used in football or other sports to prevent and inform treatment of concussions.

Background

Grappling martial arts such as Brazilian JiuJitsu and Judo have the concept of 'tapping out'. This is possible because there is some delay in the execution of a technique and the injury that results from it. For instance, player A can apply a chokehold to player B. There are a few seconds where player B knows that his brain is losing oxygen and that he is unable to escape, but before he becomes unconscious, he will tap player A to signal his unconditional surrender. Player A lets go and is declared the winner, and no one is injured.

Striking martial arts, however, cannot separate the impact of the hit with the pain that comes from it. There is no way to '*really know*' that you '*would have*' hit someone aside from actually hitting them. The '*purest*' way to score is by measuring the damage that each player can inflict on the other. Tournaments in the 1960s and 1970s were frequently full contact, and only a very small percentage of the population could participate. However, in the early 1980s with movies like the Karate Kid and the invention of dipped-foam protective padding, there was a huge push to make martial arts available to children, women, older adults, etc.

This led to the development of 'points' as a proxy measure of impact. Rather than sparring until one player became too injured to continue, participants could spar for something like 3 rounds of 2 minutes, and judges would award points for strikes delivered under certain conditions. For instance, head contact must be controlled with younger age divisions, etc.

The development of points was great in that it allowed **literally hundreds of millions of people** around the world to participate in sparring. However, there were many issues. Different martial arts would give more points for a punch of more for a kick, etc. In Olympic Taekwondo, the leadership changes the point rules every year or two, as a way of trying to force the style of the athletes to be more 'exciting.' Current rules reward: a punch to the body (**1 pt**), a kick to the body (**2 pts**), a kick to the head (**3 pts**), a spinning kick to the body (**4 pts**), and a spinning kick to the head (**5 pts**). [2]

[1] http://en.mastkd.com/2018/12/daedo-selected-again-as-pss-supplier-for-olympic-games/

[2] WT Official Sparring Rules, worldtaekwondo.org

A key difficulty in awarding points is knowing what is ‘hard enough contact’. Taekwondo used to define this as causing ‘trembling shock’ to the opponent’s body. With the development of hogus that sense kicking force, this has been replaced with thresholds. However, the thresholds need to adjust for males vs. females as well as for each weight / age division[3]. Thus, one athlete could hit another ten times with 99% of the required force each time, while the other could hit with 101% of the required force once, and win 1 to 0. Intuitively, this doesn’t seem right. With energy-based scoring, whoever imparted the most energy to their partner’s body would be the winner. Energy-based scoring instead of point-based scoring could alleviate this issue, but how accurate are the vests?

Physics of Martial Arts Strikes – A ‘hard’ strike is surprisingly difficult to model completely.[4] The reason is that from a physics perspective, most strikes have both a force component and an energy component. For instance, when the knee is bent at the moment of impact, the kick delivers more force, which is used to move an opponent. When the knee is straight at the moment of impact, the kick delivers more energy, which is used to damage an opponent. This is why knees are virtually always straight when doing applications like board breaking. While pushing kicks are used in sparring to create distance or disrupt an opponent’s momentum, the main impact desired in sparring is the energy impact. This is what the sparring vests are made to measure.

Materials and Methods

Originally testing was undertaken with the hogu flat on the ground, but the manufactures do not recommend testing this way as some kind of motion of the device is necessary in order to properly record impacts. Thus, the following setup was adopted:

A torso from Century’s Body Opponent Bag (BOB) lays flat on the floor, with a Daedo helmet and hogu. Each contains a transmitter which sends a signal to the receiver and laptop.[5]

A 5 lb (2.26 kg) weight has a magnet extracted from a Daedo Sock taped to it. The hogu will not record points without the magnet attached.

The setup for 2020 armor is similar, yet the vest connects to a smartphone via Bluetooth.

---

[3] https://www.tkdscore.com/thresholds

[4] *Taekwondo: A Practical Guide to The World’s Most Popular Martial Art*: Pottle, 2010

[5] http://www.daedotruescoreusa.com/electronic-systems-c13/

To deliver the impact, the weight was repeatedly dropped from 20cm, 40cm, 60cm, 80cm, 100cm, 120cm and 140cm heights. Energy was calculated by

$Ke = \frac{1}{2} m * v^2$

Where the velocity was calculated from:

$$v = \sqrt{2gh}$$

Height (m)	Final Velocity (m/s)	Impact Energy (J)
0.20	1.98	4.37
0.40	2.80	8.74
0.60	3.43	13.11
0.80	3.96	17.48
1.00	4.43	21.85
1.20	4.85	26.22
1.40	5.24	30.60

Data Setup

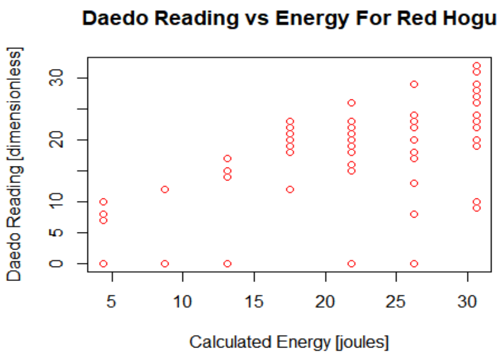
The data file (composite.csv) has 4 columns. **Energy** is amount of energy in Joules calculated from the weight and fall height. **dReading** is the reading from the sensor for Daedo in dimensionless units. **eReading** is the reading from the sensor for 2020 armor, in Joules. **Device** is a factor variable representing which device the reading came from with the following values.

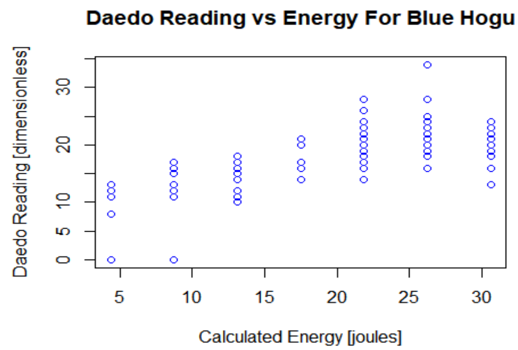
Device 1	Red Daedo Hogu
Device 2	Blue Deado Hogu
Device 3	Red Daedo Helmet
Device 4	Blue Daedo Helmet
Device 5	2020 Armor Hogu 1
Device 6	2020 Armor Hogu 2

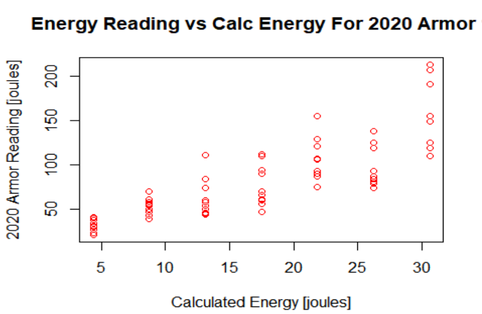
Statistical Analysis

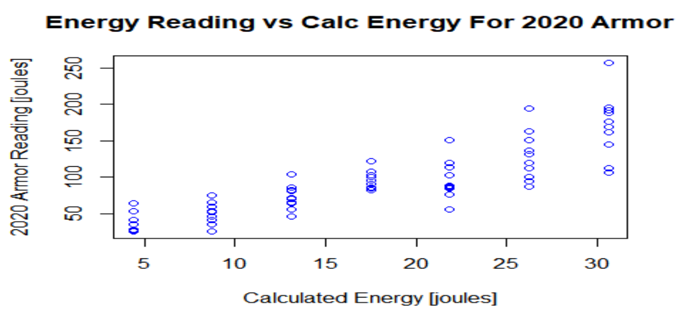
First the file is loaded, summary statistics are checked, and the data is broken into subsets for each separate device.

The reading for each of the 4 hogus are plotted on the y axis with the energy calculated from fall height and weight as the independent variable.









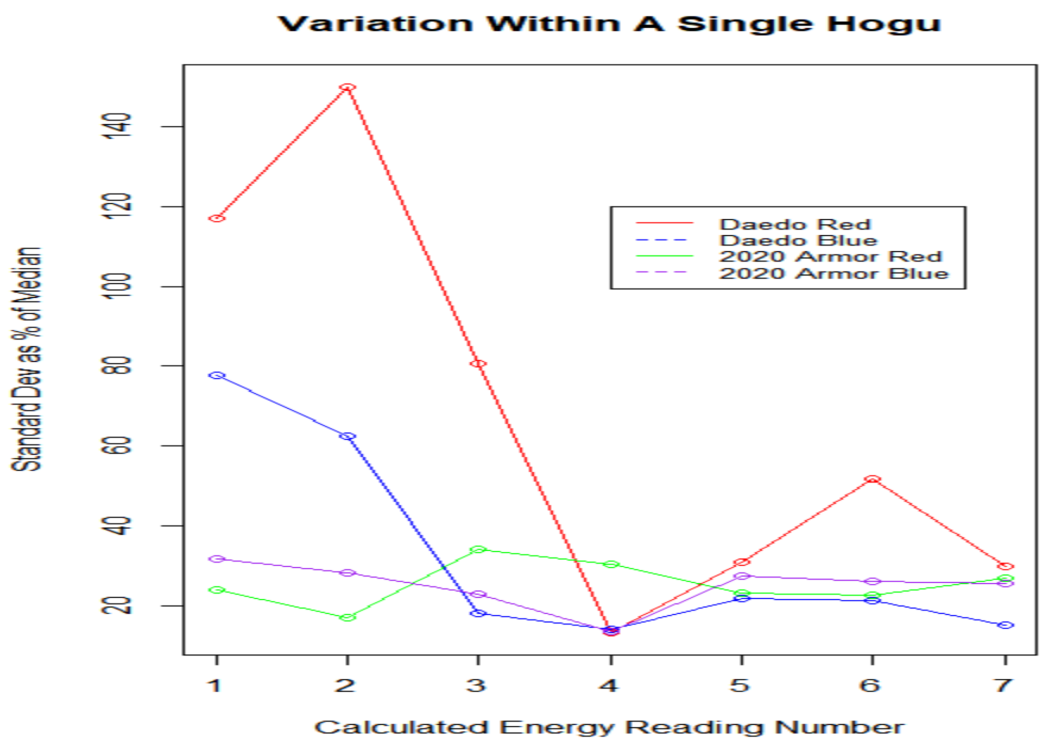
The graphs look as expected, with vertical lines of points representing the different values received at each specific drop height.

Theoretically, the hogus should give the same y value (Joules or Daedo Dimensionless Units) for each x value (Joules of Calculated Energy).

Which Brand has More Variation Within A Hogu? - In order to answer this question, first we calculate the standard deviation and mean of the readings at each x value. Because the reading gets higher when the calculated energy is higher, we normalize by expressing the standard deviation as a percentage of the mean.

When we extend to all 4 hogus and plot the variation vs. the reading number, we create the following graph.

It is interesting to note that in general, variation is higher at the lower extreme. This is particularly true with the Daedo hogus. At low values of impact energy, the standard deviation is even greater than the mean. This is a clue that a lot of noise is getting added. Of course, we would expect that when the signal (strike) is low, the signal to noise ratio is also smaller. There were many strikes that recorded a ‘0’ on the receiver.



Mean Variation Across all Data Points by Device (Lower is better)

Daedo Red	Daedo Blue	Daedo All	2020 Armor Red	2020 Armor Blue	2020 Armor All
67.57	32.88	**50.23**	25.38	24.99	**25.18**

The Daedo hogus have approximately twice as much variation for a given energy impact value as the 2020 armor hogus. It's also important to note that the Daedo Red hogu has more than double the variation of the Daedo Blue hogu, while the two 2020 armor hogus are very similar. However, even the best Daedo hogu has about a third more variation than the worst 2020 armor unit.

Which Brand has More Variation Between Hogus, and does this give one player a statistically significant advantage? This is an important question when it comes to competitive fairness. If one hogu is harder to score on than another, it would be the same as one runner being on a shorter track or only one swimmer going against a current.

We use the Welch Two Sample T Test to compare means of the values captured with the same x value (energy impact) on the red and blue hogus from the same manufacturer. This code was run over all sample values.

The null hypothesis is that the means are equal. The alternative hypothesis is that there is a true difference in the means. In a perfect world, we would expect little variation between two items made with the same manufacturer, made around the same time, and subjected to the same patterns of historical use. We should fail to reject the null hypothesis across all striking values.

The following table lists the p values of the test. Non-significant p values are in white boxes, while significant values have the cell colored with the color of the higher mean, i.e., the hogu that is easier to score on.

Daedo

4.37 J	8.74 J	13.11 J	17.48 J	21.85 J	26.22 J	30.6 J
0.1541	**0.0269**	**0.01884**	0.1872	0.5786	**0.04458**	0.1717

2020 Armor

4.37 J	8.74 J	13.11 J	17.48 J	21.85 J	26.22 J	30.6 J
0.1274	**0.6469**	**0.258**	0.0408	**0.47**	**0.2353**	**0.389**

The results indicate that for Daedo, with a 95% level of statistical significance, the blue hogu was easier to score on than the red hogu for 3 out of the 7 impact levels. For 2020 armor, the blue hogu was easier to score on for just one of the 7 impact levels. Thus, the Daedo hogus have more variation between two units.

Daedo Hogus Linear Model – One of the central questions is whether there is a linear model that relates the calculated energy impact with the dimensionless Daedo reading. This would allow all owners of Daedo hogus to know how much energy is measured from their strikes.

We create a linear model with the Daedo Reading as the dependent variable and the Calculated Energy as the independent variable. The results are as follows:

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.22936    1.00287   7.209 1.42e-11 ***
## dReading     0.71353    0.05711  12.493  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.347 on 184 degrees of freedom
## Multiple R-squared:  0.4589, Adjusted R-squared:  0.456
## F-statistic: 156.1 on 1 and 184 DF,  p-value: < 2.2e-16
```

The model developed is:

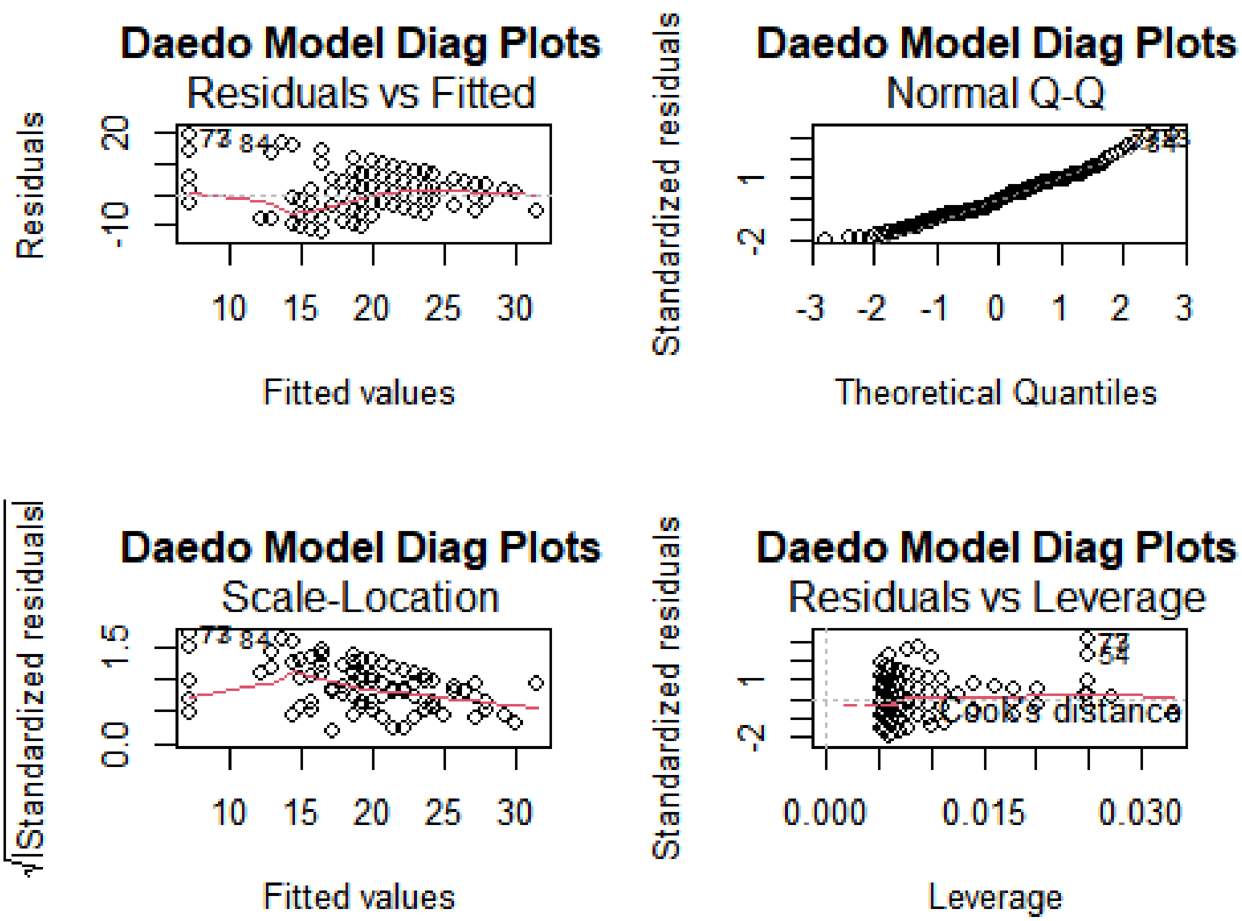
Energy [Joules] = 0.71353 * Daedo Reading + 7.23

The model is very interesting in that the r^2 value is not that high, indicating that just 46% of the energy impact is accounted for by the reading. This puts the correlation in the 'medium to substantial' correlation level. However, the P values for the intercept and slope are highly significant, at 10^-11 and 10^-16 respectively.

The confidence interval is:

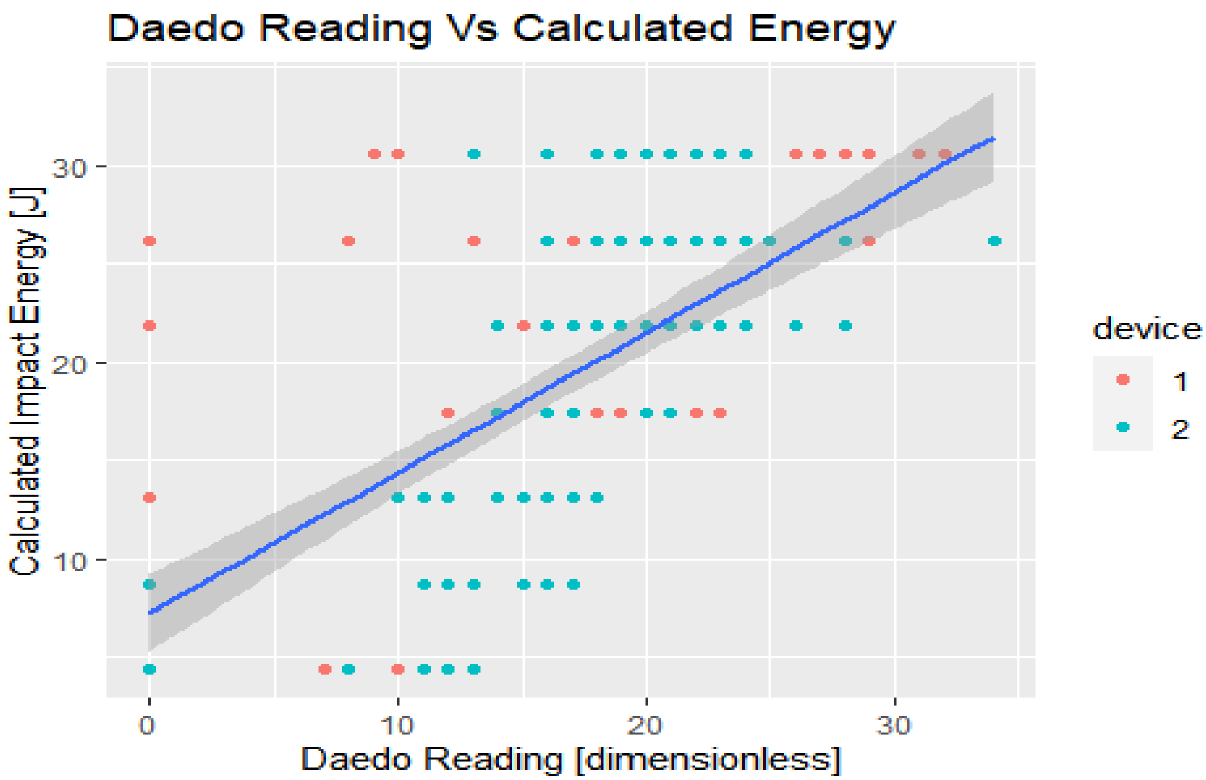
```
##                   2.5 %    97.5 %
## (Intercept) 5.2507660 9.2079558
## dReading    0.6008425 0.8262083
```

This matches well with real world experience. Stronger kicks certainly result in higher scores, but there is tremendous variation in scores between strikes with the same amount of energy. Next we look at the standard diagnostic plots.



The standard diagnostic plots do not reveal any significant areas of concern. It is worth noting that the Residuals vs. Fitted and Normal Q-Q plots both show the difficulty that Daedo systems have with strikes with less power. Errors are much higher when strikes are at the lower end.

The scale-location plot shows that the data is mostly homoscedastic, i.e. with the noted errors at lower power levels. The Residuals vs. Leverage plot shows that there are no points with a high enough Cook's distance to invalidate the model. Next we graph the model with the 95% confidence interval:



The graph looks like what we would expect. There is a clear positive linear correlation, but a lot of noise.

2020 Armor Linear Model – Next we create the model for the 2020 armor hogus for comparison.

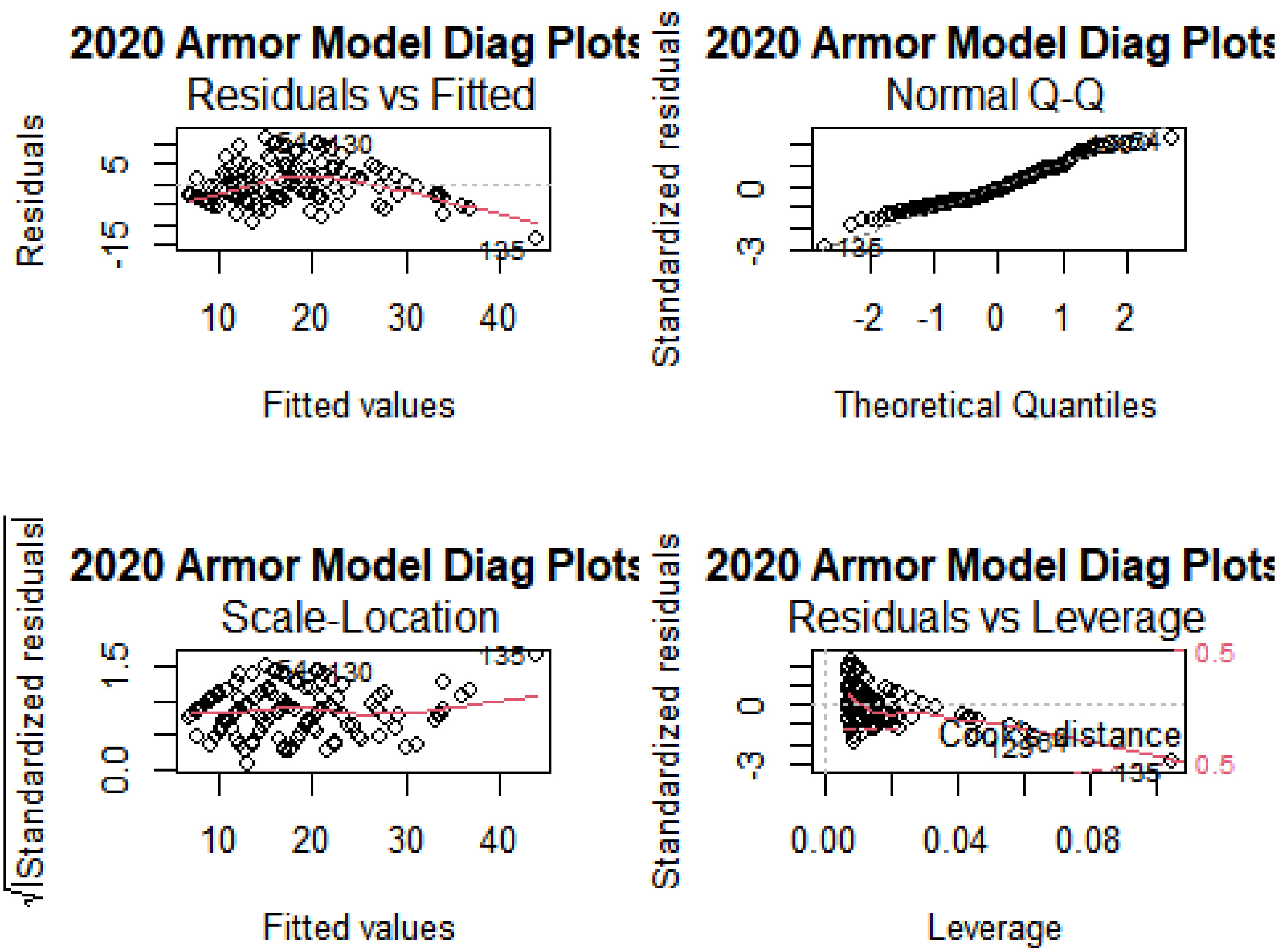
```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.724078   0.895829   4.157 5.65e-05 ***
## eReading    0.156103   0.009055  17.240  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.924 on 137 degrees of freedom
## Multiple R-squared:  0.6845, Adjusted R-squared:  0.6822
## F-statistic: 297.2 on 1 and 137 DF,  p-value: < 2.2e-16
```

The resulting equation states:

Energy [J] = 0.156 * 2020 armor energy reading [J] + 3.72

Similar to the Daedo model, the P values for both the slope and intercept are extremely low. Thus we can reject the null hypothesis that the 2020 reading does not depend on impact energy. The r^2 value here is much better than before at 0.68, which puts it into the 'Very Strong' correlation range.

We see some interesting things in the diagnostic plots.



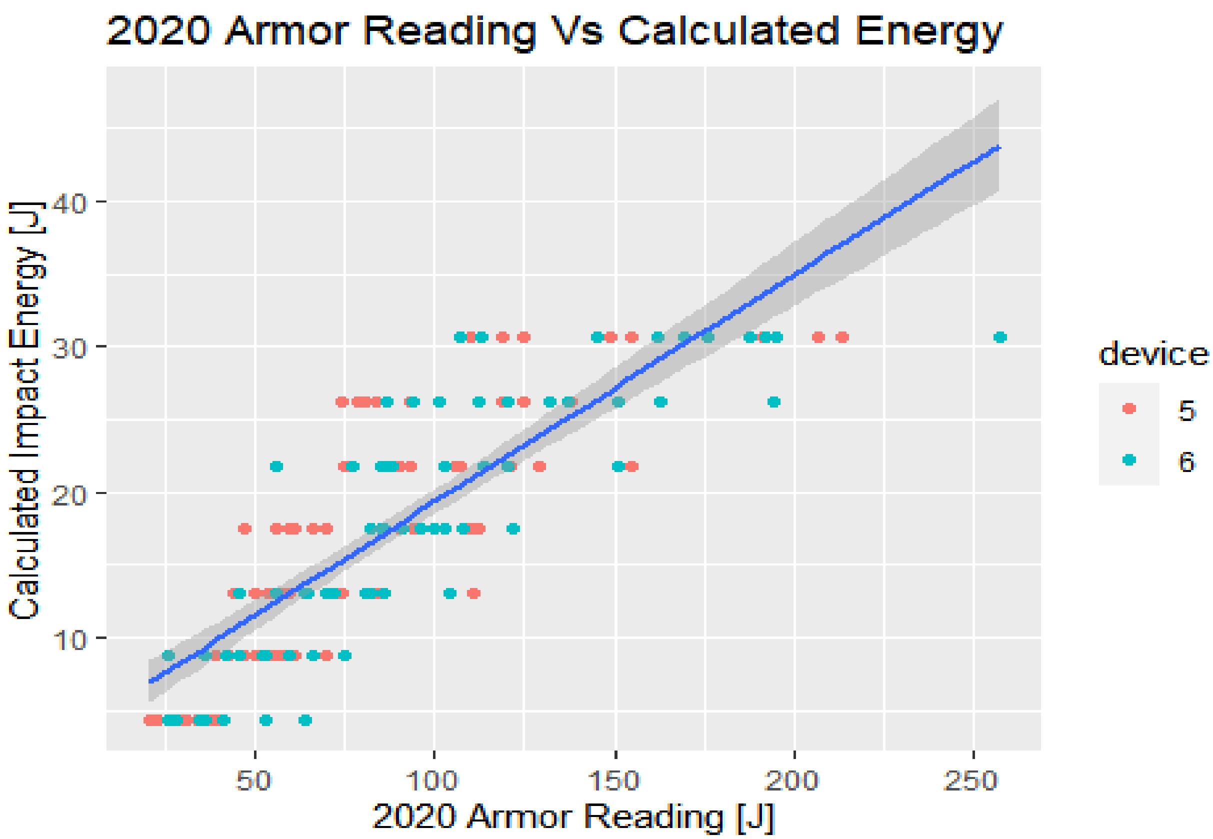
There is one high leverage point at 135 here that pulls the regression up and has a Cook's distance of 0.5. The Residuals vs. Fitted plot roughly shows that the linear model is a good fit, because the points seem to be evenly distributed around a mean of 0, although the point at 135 does act to pull the slope of the line downwards.

The Normal Q-Q plot shows the same, with a linear range through the middle of the data set, with slight breaks at the lower and higher end of the data.

The scale-location plot shows that the data is homoscedastic, i.e. that the magnitude of the error does not depend on the value of the predictor variable. Again, the point at 135 pulls the line upwards. The Residuals vs. Leverage plot shows that point 135 does have an oversize impact on the model.

The confidence interval is:

```
##                   2.5 %    97.5 %
## (Intercept) 1.9526377 5.4955187
## eReading    0.1381979 0.1740087
```

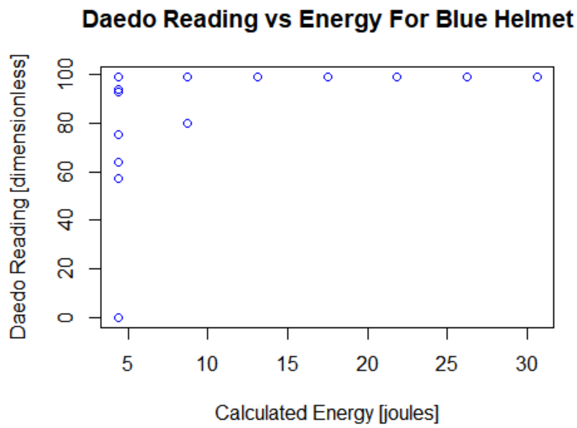


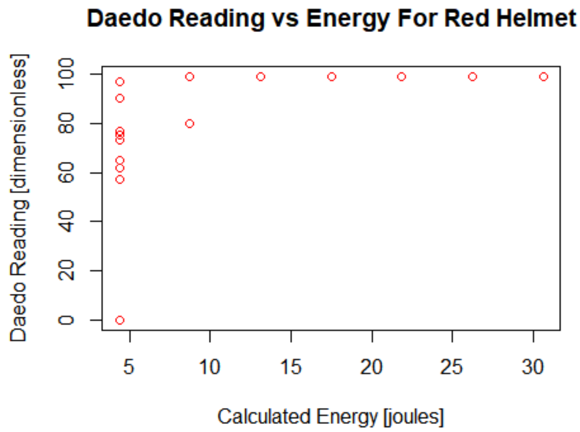
There is a tighter confidence interval around these points, but there is also a lot of similarities with the Daedo graph. There is a clear positive linear relationship, but also a lot of variation in the data.

Helmet Analysis – The helmets are another important part of the system. Helmets from 2020 Armor will not be available until June 2021, and thus were not included in this report.

First, we graph the Daedo reading vs. the calculated energy impact.

The helmet data shows the need to be careful about the range of data used for constructing a linear model. At the third energy value (60 cm drop) and above, all readings were 99. It is clear that this is the highest reading possible with the Daedo system.





Log transformations can improve on the linear model somewhat, but they are not appropriate because they ignore the fundamental issue of saturation of the sensor and reporting hardware. Therefore, the best method is to use a linear model with a cutoff of the point where the data saturates.

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.07729    1.40208   1.482    0.144
## dReading     0.06893    0.01540   4.477 3.77e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.785 on 56 degrees of freedom
## Multiple R-squared:  0.2636, Adjusted R-squared:  0.2504
## F-statistic: 20.04 on 1 and 56 DF,  p-value: 3.766e-05
```

The slope is statistically significant with a p value < 0.001. The intercept, however, is not statistically significant. This makes sense, because we do not have a lot of data or assumptions about what happens at low energy levels as the hogu is not able to record low impacts.

The r^2 value of 0.25 puts this only in the ‘Weak to Moderate’ range.

Because the intercept is not significant, we repeat the regression without it.

```
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## dReading 0.090951   0.004058   22.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.815 on 57 degrees of freedom
## Multiple R-squared:  0.8981, Adjusted R-squared:  0.8963
## F-statistic: 502.3 on 1 and 57 DF,  p-value: < 2.2e-16
```

The results this time are much better, with the p value dropping 11 orders of magnitude and the r^2 value going up to 0.90, putting this in the correlation in the ‘extremely strong’ category.

Thus the equation is

If Daedo Reading < 99: **Impact Energy [J] = 0.091 * Daedo Reading**

If Daedo Reading = 99: **Impact Energy [J] >= 14**

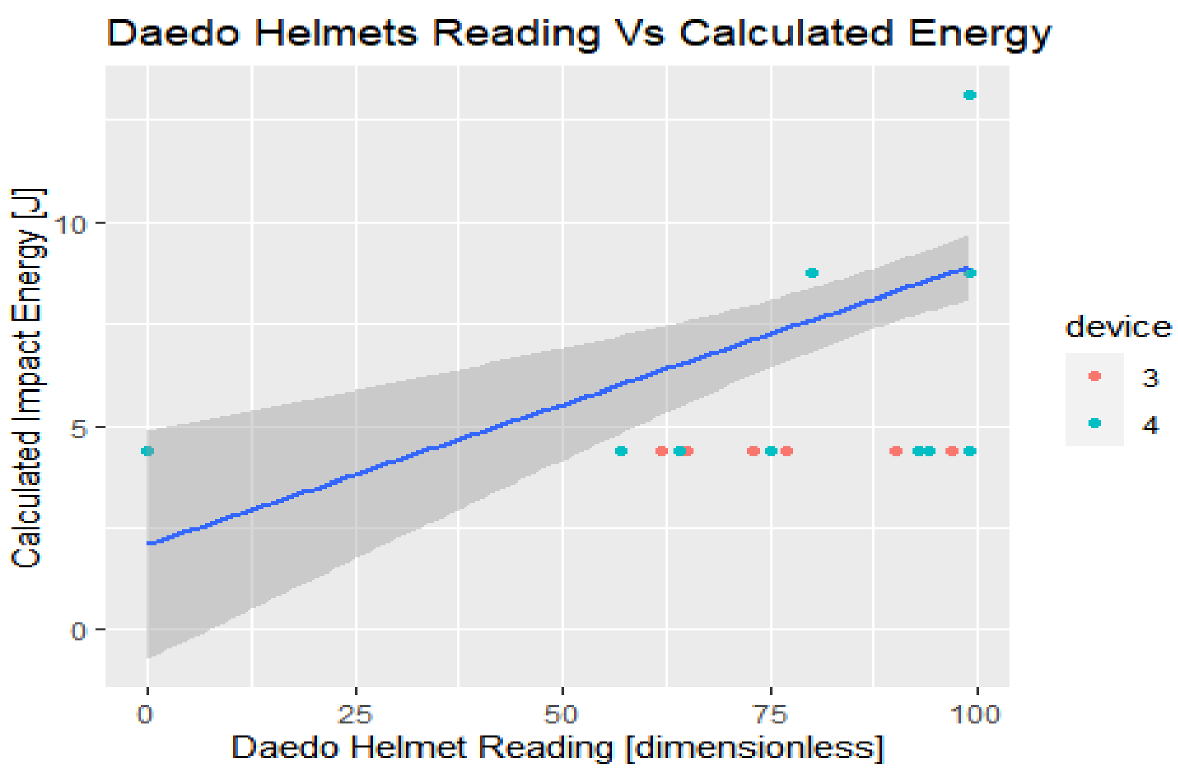
The confidence interval is quite wide for both the intercept and slope when including both:

```
##                   2.5 %     97.5 %
## (Intercept) -0.73141289 4.88599147
## dReading     0.03808845 0.09977279

However, this interval narrows considerably when removing the intercept from
the model.

##                 2.5 %     97.5 %
## dReading 0.08282499 0.09907777
```

This uncertainty about the intercept is also reflected in the chart below, with a very wide confidence interval around the left side of the graph.



The graph shows a clear linear trend, but a very wide confidence interval, especially in the lower levels of Daedo Readings.

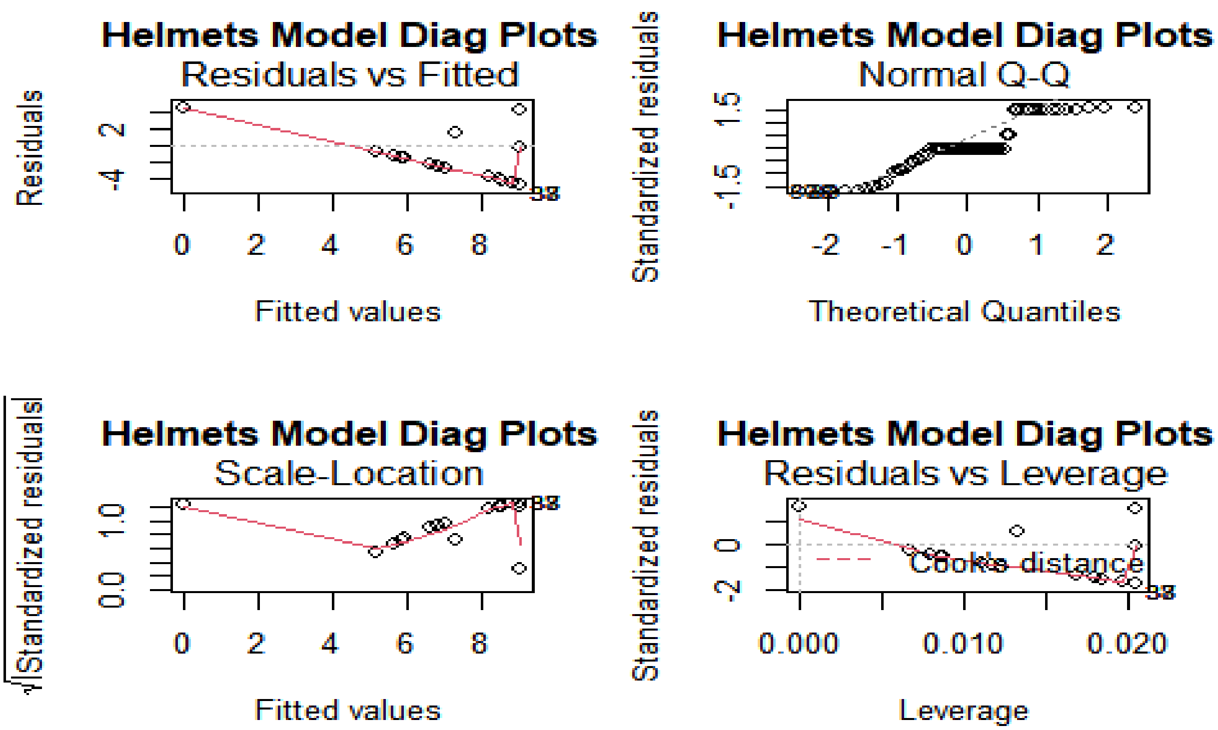
We check the standard diagnostics plots for this situation as well.

The Residuals vs. Fitted plot roughly shows that the linear model is not a good fit, because the residuals decrease as the fitted values increase.

The Normal Q-Q plot shows the same, with a stepwise pattern rather than the line we would expect from a good fit.

The scale-location plot shows that the data is not homoscedastic, i.e. that the magnitude of the error **does** depend on the value of the predictor variable. The Residuals vs. Leverage plot shows that there are multiple outliers influencing the model.

With all these issues in the diagnostic plots, it's somewhat surprising that our r^2 value is still high. However, this could be a case of overfitting to a very limited dataset.



Improvements And Future Work

These preliminary findings are a potential cause for concern from a standpoint of competitive fairness. The next step would be to test a much larger sample size of hogus to see if the variability issues persist across the entire population.

Another key improvement would be to reduce experimental error. There were times when the weight 'bounced' on the hogu, and two quick strikes in succession may have influenced the reading. It would be better to have access to the raw voltage data coming from the embedded sensors. Experimental error could also be reduced by testing in a wind tunnel or using a machine that always strikes with the same amount of energy.

More information could also be found by testing different parts of the hogu and helmet, rather than striking only in the front of the chest.

Olympic Results

Taekwondo is a popular sport worldwide because it is known as a sport where any country can have a chance to win an Olympic medal. For instance, in order to be competitive in swimming, a country must have a network of hundreds of pools, high school and collegiate programs, etc. However, Taekwondo only requires a few athletes and a coach. In fact, Vietnam[6], Gabon[7] and Afghanistan[8] all won their first ever Olympic medals in Taekwondo. Could variation in the hogus be giving athletes from smaller countries a higher chance vs. traditional powerhouse countries than they otherwise should have? In the 2012 Olympics, the 32 medals were won by athletes from 21 different countries[9]. In the 2016 Rio games, the 32 medals were won by athletes from 20 different countries[10]. These raw numbers are somewhat suspicious, since Daedo hogus were used in both games. However, in the 2008 games which used manual scoring, the 32 medals were won by athletes from 22 different countries.[11]

There is not enough data here to reach any conclusions, but it is an interesting area for future study.

Challenges Faced

There were numerous challenges faced with this project because it involved much more than just data analysis. The experimental design required determining what could lead to statistically significant results, while still being feasible to carry out in a reasonable amount of time and with a reasonable amount of materials.

The other big challenge was getting the Daedo systems working – I had to install new versions of the software and radio drivers on my computer and one of my hogu transmitters was not connecting properly. Even though I was testing only one device at a time, the software would not work without both transmitters running. After hours troubleshooting with the main US distributor, I ultimately was unable to get it to work, but fortunately I was able to borrow a transmitter from another martial arts instructor.

I spoke with the manufacturer for 2020 Armor, but was unable to get their helmets in time for this experiment. Their supply chain has been affected by COVID and they only have a handful of prototypes available.

Conclusions

[6] https://www.abc.net.au/news/2016-08-07/vietnam-wins-first-ever-olympic-gold-medal/7698692

[7] https://olympics.com/tokyo-2020/en/news/only-olympic-medal-history-maker-obame-targets-more-glory-for-gabon

[8] https://www.rferl.org/a/Afghanistan_Celebrates_Its_First_Ever_Olympic_Medal/1192877.html

[9] https://en.wikipedia.org/wiki/Taekwondo_at_the_2012_Summer_Olympics

[10] https://en.wikipedia.org/wiki/Taekwondo_at_the_2016_Summer_Olympics

[11] https://en.wikipedia.org/wiki/Taekwondo_at_the_2016_Summer_Olympics

The main conclusions of this work are twofold. Hogus made by 2020 Armor have significantly less variation both within a single unit and between multiple units than their counterparts manufactured by Daedo. Secondly, there is high statistical significance in the fact that harder strikes on both hogus will result in higher scores. However, there is a concerning amount of variation in these scores that should be addressed by future work.

How to Reproduce

The full source data (CSV Format) has been posted to the Interplanetary File System (IPFS) at hash: QmY8J7bYGzaFWjvMEHafDpJ5vm6k3zwRkSPs7Fra3kycpb

The full R code follows:

```
# Libraries in order of use
library(readr)

library(dplyr)

library(ggplot2)

# Read in the data from the file on disk

composite <- read_csv("C:/Users/bill/OneDrive/Data Science/Stats 109/Project/
composite.csv")

# We want the device number to be a factor

# Device 1 - Daedo Red Hogu
# Device 2 - Daedo Blue Hogu
# Device 3 - 2020 Armor Hogu 1
# Device 4 - 2020 Armor Hogu 2
# Device 5 - Daedo Red Helmet
# Device 6 - Daedo Blue Helmet
composite$device<- as.factor(composite$device)


# Create subsets of the data for each factor variable
daedo_red <- subset(composite, device =='1')
daedo_blue <- subset(composite, device =='2')
armor_2020_red <- subset(composite, device =='5')
armor_2020_blue <- subset(composite, device =='6')
helmet_daedo_red <- subset(composite, device =='3')
helmet_daedo_blue <- subset(composite, device =='4')

# Plot Energy Vs Daedo Reading for the Daedo Hogus
plot(daedo_red$dReading ~ daedo_red$energy, col='red', main='Daedo Reading vs
Energy For Red Hogu', ylab = 'Daedo Reading [dimensionless]', xlab= 'Calculat
ed Energy [joules]')
```

```r
plot(daedo_blue$dReading ~ daedo_blue$energy, col='blue', main='Daedo Reading
vs Energy For Blue Hogu', ylab = 'Daedo Reading [dimensionless]', xlab= 'Calc
ulated Energy [joules]')



# Plot Calculated Energy Vs Energy Reading for the 2020 Armor Hogus
plot(armor_2020_red$eReading ~ armor_2020_red$energy, col='red', main='Energy
Reading vs Calc Energy For 2020 Armor 1', ylab = '2020 Armor Reading [joules]
', xlab= 'Calculated Energy [joules]')
```

```r
plot(armor_2020_blue$eReading ~ armor_2020_blue$energy, col='blue', main='Ene
rgy Reading vs Calc Energy For 2020 Armor 2', ylab = '2020 Armor Reading [jou
les]', xlab= 'Calculated Energy [joules]')
```

```r
# For each reading, filter by device and energy level
daedo_red_1 <- daedo_red %>% filter(energy == '4.37')
daedo_blue_1 <- daedo_blue %>% filter(energy == '4.37')

daedo_red_2 <- daedo_red %>% filter(energy == '8.74')
daedo_blue_2 <- daedo_blue %>% filter(energy == '8.74')

daedo_red_3 <- daedo_red %>% filter(energy == '13.11')
daedo_blue_3 <- daedo_blue %>% filter(energy == '13.11')

daedo_red_4 <- daedo_red %>% filter(energy == '17.48')
daedo_blue_4 <- daedo_blue %>% filter(energy == '17.48')

daedo_red_5 <- daedo_red %>% filter(energy == '21.85')
daedo_blue_5 <- daedo_blue %>% filter(energy == '21.85')

daedo_red_6 <- daedo_red %>% filter(energy == '26.22')
daedo_blue_6 <- daedo_blue %>% filter(energy == '26.22')

daedo_red_7 <- daedo_red %>% filter(energy == '30.6')
daedo_blue_7 <- daedo_blue %>% filter(energy == '30.6')

# Do the same for 2020 armor
armor_2020_red_1 <- armor_2020_red %>% filter(energy == '4.37')
armor_2020_blue_1 <- armor_2020_blue %>% filter(energy == '4.37')

armor_2020_red_2 <- armor_2020_red %>% filter(energy == '8.74')
armor_2020_blue_2 <- armor_2020_blue %>% filter(energy == '8.74')

armor_2020_red_3 <- armor_2020_red %>% filter(energy == '13.11')
armor_2020_blue_3 <- armor_2020_blue %>% filter(energy == '13.11')
```

```
armor_2020_red_4 <- armor_2020_red %>% filter(energy == '17.48')
armor_2020_blue_4 <- armor_2020_blue %>% filter(energy == '17.48')

armor_2020_red_5 <- armor_2020_red %>% filter(energy == '21.85')
armor_2020_blue_5 <- armor_2020_blue %>% filter(energy == '21.85')

armor_2020_red_6 <- armor_2020_red %>% filter(energy == '26.22')
armor_2020_blue_6 <- armor_2020_blue %>% filter(energy == '26.22')

armor_2020_red_7 <- armor_2020_red %>% filter(energy == '30.6')
armor_2020_blue_7 <- armor_2020_blue %>% filter(energy == '30.6')

# Use t test for Daedo to see if the means are different
t.test(daedo_red_1$dReading, daedo_blue_1$dReading)

t.test(daedo_red_2$dReading, daedo_blue_2$dReading)

t.test(daedo_red_3$dReading, daedo_blue_3$dReading)

t.test(daedo_red_4$dReading, daedo_blue_4$dReading)

t.test(daedo_red_5$dReading, daedo_blue_5$dReading)

t.test(daedo_red_6$dReading, daedo_blue_6$dReading)

t.test(daedo_red_7$dReading, daedo_blue_7$dReading)

# Do the t test for 2020 armor to see if the means are different
t.test(armor_2020_red_1$eReading, armor_2020_blue_1$eReading)

t.test(armor_2020_red_2$eReading, armor_2020_blue_2$eReading)

t.test(armor_2020_red_3$eReading, armor_2020_blue_3$eReading)

t.test(armor_2020_red_4$eReading, armor_2020_blue_4$eReading)

t.test(armor_2020_red_5$eReading, armor_2020_blue_5$eReading)

t.test(armor_2020_red_6$eReading, armor_2020_blue_6$eReading)

t.test(armor_2020_red_7$eReading, armor_2020_blue_7$eReading)

# Calculate stdev, mean, and stdev as a % of mean for Daedo red hogu
s <- sd(daedo_red_1$dReading)
m <- mean(daedo_red_1$dReading)
dr1 <- s / m * 100

s <- sd(daedo_red_2$dReading)
m <- mean(daedo_red_2$dReading)
dr2 <- s / m * 100

s <- sd(daedo_red_3$dReading)
m <- mean(daedo_red_3$dReading)
```

```r
dr3 <- s / m * 100

s <- sd(daedo_red_4$dReading)
m <- mean(daedo_red_4$dReading)
dr4 <- s / m * 100

s <- sd(daedo_red_5$dReading)
m <- mean(daedo_red_5$dReading)
dr5 <- s / m * 100

s <- sd(daedo_red_6$dReading)
m <- mean(daedo_red_6$dReading)
dr6 <- s / m * 100

s <- sd(daedo_red_7$dReading)
m <- mean(daedo_red_7$dReading)
dr7 <- s / m * 100

dr <- c(dr1, dr2, dr3, dr4, dr5, dr6, dr7)

# Calculate stdev, mean, and stdev as a % of mean for Daedo blue hogu
s <- sd(daedo_blue_1$dReading)
m <- mean(daedo_blue_1$dReading)
db1 <- s / m * 100

s <- sd(daedo_blue_2$dReading)
m <- mean(daedo_blue_2$dReading)
db2 <- s / m * 100

s <- sd(daedo_blue_3$dReading)
m <- mean(daedo_blue_3$dReading)
db3 <- s / m * 100

s <- sd(daedo_blue_4$dReading)
m <- mean(daedo_blue_4$dReading)
db4 <- s / m * 100

s <- sd(daedo_blue_5$dReading)
m <- mean(daedo_blue_5$dReading)
db5 <- s / m * 100

s <- sd(daedo_blue_6$dReading)
m <- mean(daedo_blue_6$dReading)
db6 <- s / m * 100

s <- sd(daedo_blue_7$dReading)
m <- mean(daedo_blue_7$dReading)
db7 <- s / m * 100
```

```r
db <- c(db1, db2, db3, db4, db5, db6, db7)


# Calculate stdev, mean, and stdev as a % of mean for 2020 Armor blue hogu
s <- sd(armor_2020_blue_1$eReading)
m <- mean(armor_2020_blue_1$eReading)
ab1 <- s / m * 100

s <- sd(armor_2020_blue_2$eReading)
m <- mean(armor_2020_blue_2$eReading)
ab2 <- s / m * 100

s <- sd(armor_2020_blue_3$eReading)
m <- mean(armor_2020_blue_3$eReading)
ab3 <- s / m * 100

s <- sd(armor_2020_blue_4$eReading)
m <- mean(armor_2020_blue_4$eReading)
ab4 <- s / m * 100

s <- sd(armor_2020_blue_5$eReading)
m <- mean(armor_2020_blue_5$eReading)
ab5 <- s / m * 100

s <- sd(armor_2020_blue_6$eReading)
m <- mean(armor_2020_blue_6$eReading)
ab6 <- s / m * 100

s <- sd(armor_2020_blue_7$eReading)
m <- mean(armor_2020_blue_7$eReading)
ab7 <- s / m * 100

ab <- c(ab1, ab2, ab3, ab4, ab5, ab6, ab7)


# Calculate stdev, mean, and stdev as a % of mean for 2020 Armor Red hogu
s <- sd(armor_2020_red_1$eReading)
m <- mean(armor_2020_red_1$eReading)
ar1 <- s / m * 100

s <- sd(armor_2020_red_2$eReading)
m <- mean(armor_2020_red_2$eReading)
ar2 <- s / m * 100

s <- sd(armor_2020_red_3$eReading)
m <- mean(armor_2020_red_3$eReading)
ar3 <- s / m * 100

s <- sd(armor_2020_red_4$eReading)
m <- mean(armor_2020_red_4$eReading)
```

```r
ar4 <- s / m * 100

s <- sd(armor_2020_red_5$eReading)
m <- mean(armor_2020_red_5$eReading)
ar5 <- s / m * 100

s <- sd(armor_2020_red_6$eReading)
m <- mean(armor_2020_red_6$eReading)
ar6 <- s / m * 100

s <- sd(armor_2020_red_7$eReading)
m <- mean(armor_2020_red_7$eReading)
ar7 <- s / m * 100

ar <- c(ar1, ar2, ar3, ar4, ar5, ar6, ar7)


# Plot the bar chart to compare variation within a single hogu
plot(dr,type = "o",col = "red", xlab = "Calculated Energy Reading Number", yl
ab = "Standard Dev as % of Median",
   main = "Variation Within A Single Hogu")

lines(db, type = "o", col = "blue")

lines(ar, type="o", col="green")
lines(ab, type="o", col="purple")

# Add a legend
legend(4, 120, legend=c("Daedo Red", "Daedo Blue", "2020 Armor Red", "2020 Ar
mor Blue"),
       col=c("red", "blue", "green", "purple"), lty=1:2, cex=0.8)
```

```r
# print the overall means
mean(ar)

mean(ab)

mean(dr)

mean(db)

# Get all daedo hogus together
daedo <- subset(composite, (device == '1' | device == '2'))

# Create the composite model
daedo_model <- lm(energy ~ dReading, data = daedo)
summary(daedo_model)

# Find the 95% confidence interval
confint(daedo_model)
```

```r
# Display diagnostic plots
par(mfrow = c(2,2))
plot(daedo_model, main='Daedo Model Diag Plots')
```

```r
# Display the model with data
daedo %>% ggplot(aes(y = energy, x = dReading )) +
geom_point(aes(colour = device)) +
geom_smooth(method = 'lm') +
ggtitle('Daedo Reading Vs Calculated Energy') +
xlab('Daedo Reading [dimensionless]') +
ylab('Calculated Impact Energy [J]')

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

```r
# Get all daedo hogus
armor_2020 <- subset(composite, (device == '5' | device == '6'))

# Create the composite model
armor_2020_model <- lm(energy ~ eReading, data = armor_2020)
summary(armor_2020_model)

# Find the 95% confidence interval
confint(armor_2020_model)

# Display diagnostic plots
par(mfrow = c(2,2))
plot(armor_2020_model, main='2020 Armor Model Diag Plots')
```

```r
# Display the model with data
armor_2020 %>% ggplot(aes(y = energy, x = eReading )) +
geom_point(aes(colour = device)) +
geom_smooth(method = 'lm') +
ggtitle('2020 Armor Reading Vs Calculated Energy') +
xlab('2020 Armor Reading [J]') +
ylab('Calculated Impact Energy [J]')

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

```r
# Plot Energy Vs Daedo Reading for the Daedo Helmets
par(mfrow = c(1,1))
plot(helmet_daedo_red$dReading ~ helmet_daedo_red$energy, col='red', main='Da
edo Reading vs Energy For Red Helmet', ylab = 'Daedo Reading [dimensionless]'
, xlab= 'Calculated Energy [joules]')
```

```r
plot(helmet_daedo_blue$dReading ~ helmet_daedo_blue$energy, col='blue', main=
'Daedo Reading vs Energy For Blue Helmet', ylab = 'Daedo Reading [dimensionle
ss]', xlab= 'Calculated Energy [joules]')
```

```r
# Get the helmets data
helmets <- subset(composite,(device ==3 | device ==4))

# Take a subset of the helmets before the sensors are saturated
helmets_linear <- subset(helmets, energy < 14)

# Create the linear model
helmets_linear_model <- lm(energy ~ dReading, data= helmets_linear)
summary(helmets_linear_model)

# Repeat without intercept.
helmets_linear_model <- lm(energy ~ dReading -1, data= helmets_linear)
summary(helmets_linear_model)

# Find the 95% confidence interval
confint(helmets_linear_model)

# Analyze the model
par(mfrow = c(2,2))
plot(helmets_linear_model, main = 'Helmets Model Diag Plots')

# Graph the model for the subset of data before saturation
par(mfrow = c(1,1))
helmets_linear %>% ggplot(aes(y = energy, x = dReading )) +
geom_point(aes(colour = device)) +
geom_smooth(method = 'lm') +
ggtitle('Daedo Helmets Reading Vs Calculated Energy') +
xlab('Daedo Helmet Reading [dimensionless]') +
ylab('Calculated Impact Energy [J]')

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