Merging CBASS Datasets

Katie Lankowicz

2023-06-29

CBASS Data

CBASS data from Portland, 2014-2022, have been cleaned and integrated. We are now ready to begin modeling and analysis. The first step in this process is cross-referencing between dataframes; CBASS contains multiple types of information, all of which are kept in separate files. This is common practice for surveys that produce lots of different information.

In this case, there are three important files which contain the majority of our information. `trips_cleaned.csv` contains all time and environmental data about each net haul. `abundance_cleaned.csv` contains the total number of each species collected at each net haul. And `bio_inf_cleaned.csv` contains biological information about the < 25 individuals of each species collected and subsampled at each net haul.

Loading data

Before we do anything else, we need to load the three dataframes.

```
trips <- read.csv(here('Clean_Data/trips_cleaned.csv'))
head(trips,3)
```

```
##        loc_id year       date site_id                site_name bay_location
## 1 2014_001_06 2014 2014-06-16       6 Mackworth Island - North          Mid
## 2 2014_001_04 2014 2014-06-16       4              Mussel Cove          Mid
## 3 2014_001_13 2014 2014-06-16      13     The Brothers - South          Mid
##    substrate            set_time weather temp_degc do_mg.l salinity_ppt
## 1  mud/shell 2014-06-16 11:20:00   sunny      15.0    10.4        32.83
## 2        mud 2014-06-16 12:20:00   sunny      17.8     9.8        32.41
## 3 sand/shell 2014-06-16 13:10:00   sunny      18.0     8.9        29.18
##   hermit_crabs shrimp    notes week month
## 1           NA     NA     <NA>   25     6
## 2           NA     NA     <NA>   25     6
## 3           NA     NA no catch   25     6
```

```
abund <- read.csv(here('Clean_Data/abundance_cleaned.csv'))
head(abund,3)
```

```
##        loc_id       date                site_name site_id bay_location
## 1 2014_001_06 2014-06-16 Mackworth Island - North       6          Mid
## 2 2014_001_06 2014-06-16 Mackworth Island - North       6          Mid
## 3 2014_001_06 2014-06-16 Mackworth Island - North       6          Mid
##        species_name catch week month year
## 1        green crab     1   25     6 2014
## 2 northern pipefish     1   25     6 2014
## 3 shorthorn sculpin     1   25     6 2014
```

```
bio <- read.csv(here('Clean_Data/bio_inf_cleaned.csv'))
head(bio,3)
```

```
##        loc_id       date                site_name site_id bay_location
## 1 2014_001_06 2014-06-16 Mackworth Island - North       6          Mid
## 2 2014_001_06 2014-06-16 Mackworth Island - North       6          Mid
## 3 2014_001_06 2014-06-16 Mackworth Island - North       6          Mid
##        species_name  sex length_mm             notes week month year
## 1        green crab    f        70              <NA>   25     6 2014
## 2 northern pipefish <NA>       153              <NA>   25     6 2014
## 3 shorthorn sculpin <NA>        29 Sculpin (Photo 1)   25     6 2014
```

Using data from multiple dataframes

The way we cross-reference and merge dataframes will vary depending on the question we'd like to answer. It always helps to think about the data we'll need to address a question, and then think about what variables can be used as an identifier for that information.

Let's do an example. Our example question is: does substrate type affect green crab abundance? The data we'll need are:

- Number of crabs at each net haul (hosted in abundance dataframe)
- Substrate type at eah net haul (hosted in trips dataframe)

Now, let's think about how we would merge these pieces of information when they exist in different dataframes. Luckily, each net haul has been assigned a unique identifying number (called `loc_id`). This identifier is used in both the `trips` and `abundance` dataframes. We can use `loc_id` as a way to merge information from both dataframes. This will be done below.

```
# First, select only necessary variables from the abundance dataframe
gc.abund <-        select(abund,         # The name of the dataframe
                          loc_id,        # Our identifier
                          species_name,  # The species caught
                          catch)         # The number caught of that species)

# Next, we will select only necessary variables from the trips dataframe
gc.trips <-        select(trips,        # The name of the dataframe
                          loc_id,       # Our identifier
                          substrate)    # The substrate type

# Now, we can merge these two dataframes
greencrabs <- merge(gc.trips, gc.abund, by='loc_id')
head(greencrabs)
```

```
##        loc_id substrate      species_name catch
## 1 2014_001_04       mud        green crab     1
## 2 2014_001_04       mud            tomcod     1
## 3 2014_001_06 mud/shell        green crab     1
## 4 2014_001_06 mud/shell northern pipefish     1
## 5 2014_001_06 mud/shell shorthorn sculpin     1
## 6 2014_001_06 mud/shell            tomcod     6
```

We now have a dataframe that tells us the number of each species caught in each net haul AND the type of substrate at that net haul. You'll notice we have every species represented in this dataframe. We only care about green crabs for this question, though, so we can subset the dataframe so it removes all other species.

```
# Subset dataframe so it only has crabs
greencrabs <- subset(greencrabs,                    # Name of dataframe
                     species_name == 'green crab') # Subsetting rule

# Check that there are only crabs in this dataframe
table(greencrabs$species_name)
```

```
##
## green crab
##        356
```

```
# Check the different types of substrate possible
table(greencrabs$substrate)
```

```
##
```

```
##          gravel             mud      mud/gravel        mud/sand       mud/shell
##              12              35               5               5               6
##            sand     sand/gravel sand/mud/gravel      sand/shell
##              36              33               1               9
```

```
# Check the number of crabs caught
summary(greencrabs$catch)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   4.000   9.713  10.000 296.000
```

We have accomplished the main task, merging data from different dataframes. Let's keep going. We can see that crabs were caught over nine types of substrate. Some of these have very few observations; for example, only one crab was caught over a substrate type called `sand/mud/gravel`. It might be beneficial for us to simplify possible substrate types to only consist of gravel, mud, and sand. We can make these changes in our dataframe.

```
# There are quicker ways to do this, but for learning purposes we will be really simple.
# We will instruct R to change the unwanted variable name to a wanted variable name.
greencrabs$substrate[greencrabs$substrate == 'mud/gravel'] <- 'mud'
table(greencrabs$substrate)
```

```
##
##          gravel             mud        mud/sand       mud/shell            sand
##              12              40               5               6              36
##     sand/gravel sand/mud/gravel      sand/shell
##              33               1               9
```

```
# Now all instance of 'mud/gravel' have been replaced with just 'mud'

# Repeat for other unwanted variable names
greencrabs$substrate[greencrabs$substrate == 'mud/sand'] <- 'mud'
greencrabs$substrate[greencrabs$substrate == 'mud/shell'] <- 'mud'
greencrabs$substrate[greencrabs$substrate == 'sand/gravel'] <- 'sand'
greencrabs$substrate[greencrabs$substrate == 'sand/mud/gravel'] <- 'sand'
greencrabs$substrate[greencrabs$substrate == 'sand/shell'] <- 'sand'

# Convert character-type data to factor-type
greencrabs$substrate <- as.factor(greencrabs$substrate)

# Check results
summary(greencrabs$substrate)
```

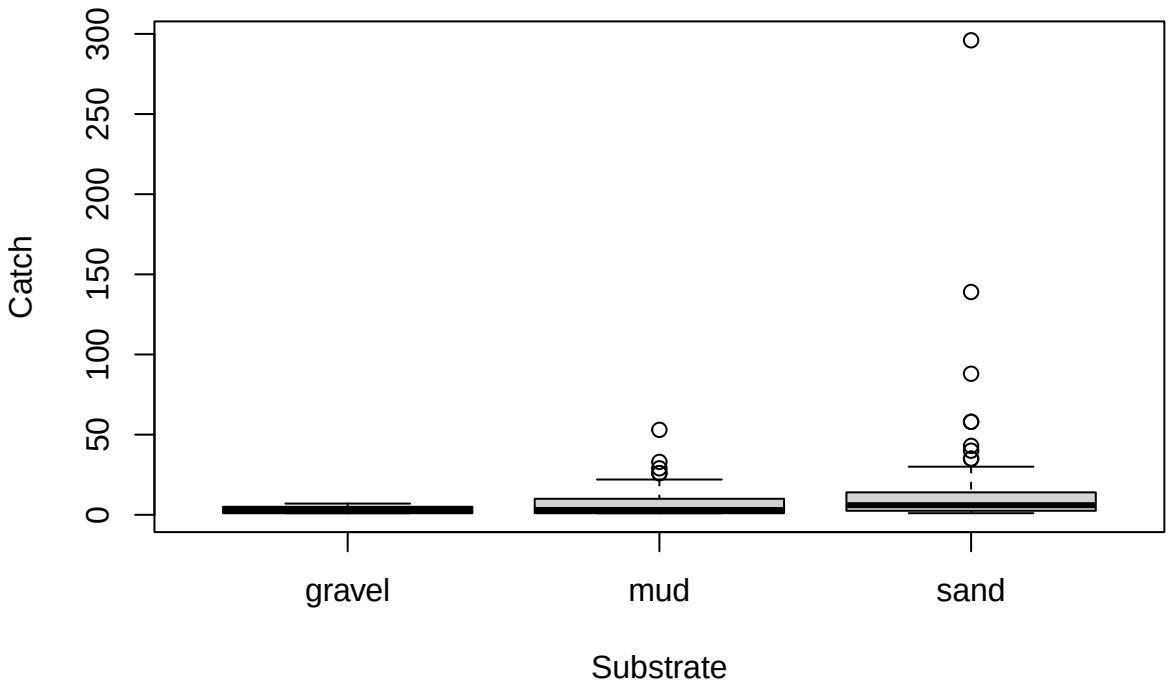
```
## gravel    mud   sand   NA's
##     12     51     79    214
```

```
# There are 214 records that do not have a recorded substrate type.
# We can probably fill these in with information from other sampling days at the
# same locations, but for now we will remove them.
greencrabs <- subset(greencrabs,          # Dataframe name
                     !is.na(substrate))   # Remove NA values (! means "not")
```

Statistical tests

We can now check the distribution of green crabs caught among the different substrate types. This can be done with a table, but is probably easier to understand as a figure. A boxplot would work well here, as we are comparing a quantitative variable among different levels of a categorical variable.

```
boxplot(greencrabs$catch ~ greencrabs$substrate,  # Catch as y variable, substrate as x
        xlab = 'Substrate',                       # X axis name
        ylab = 'Catch')                           # Y axis name
```

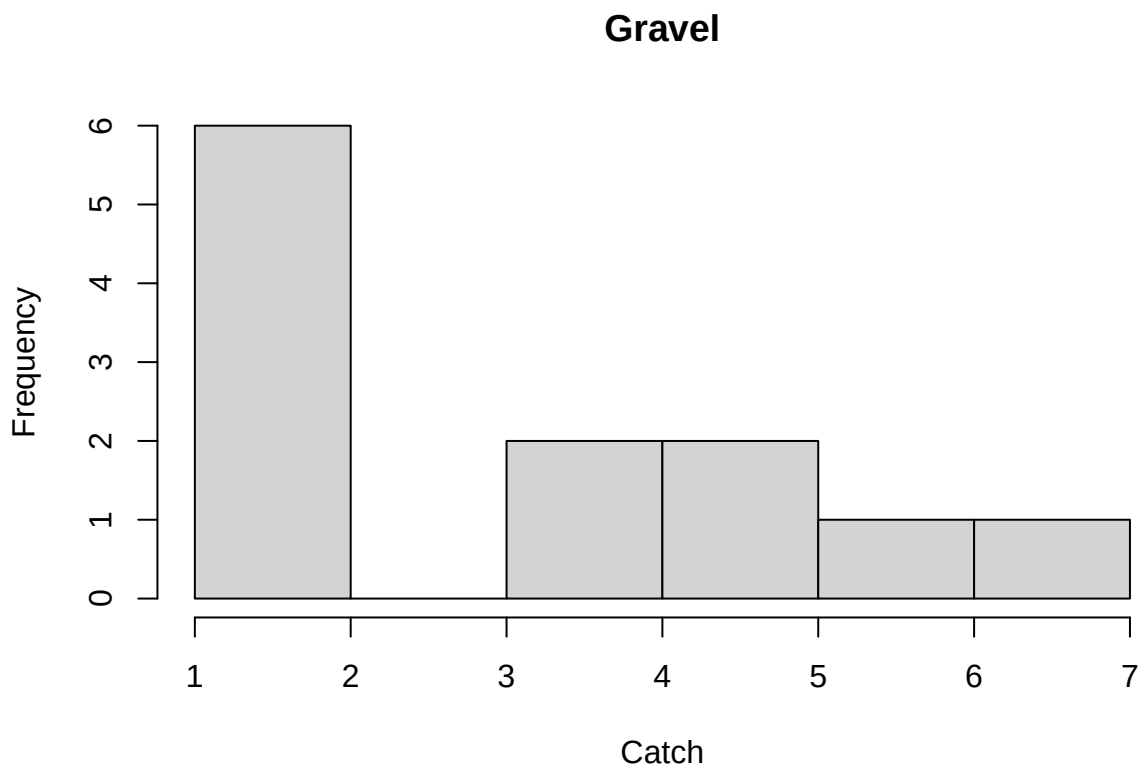


Visually, it looks like substrate may have an effect on the number of green crabs at a site. We can do a statistical test to get a more robust answer. Usually, we would choose to do an ANOVA, which tests whether samples from different categorical groups all come from the same distribution. Essentially, we'd use it to test if there is a difference in mean crab abundance for each of our three substrate types.

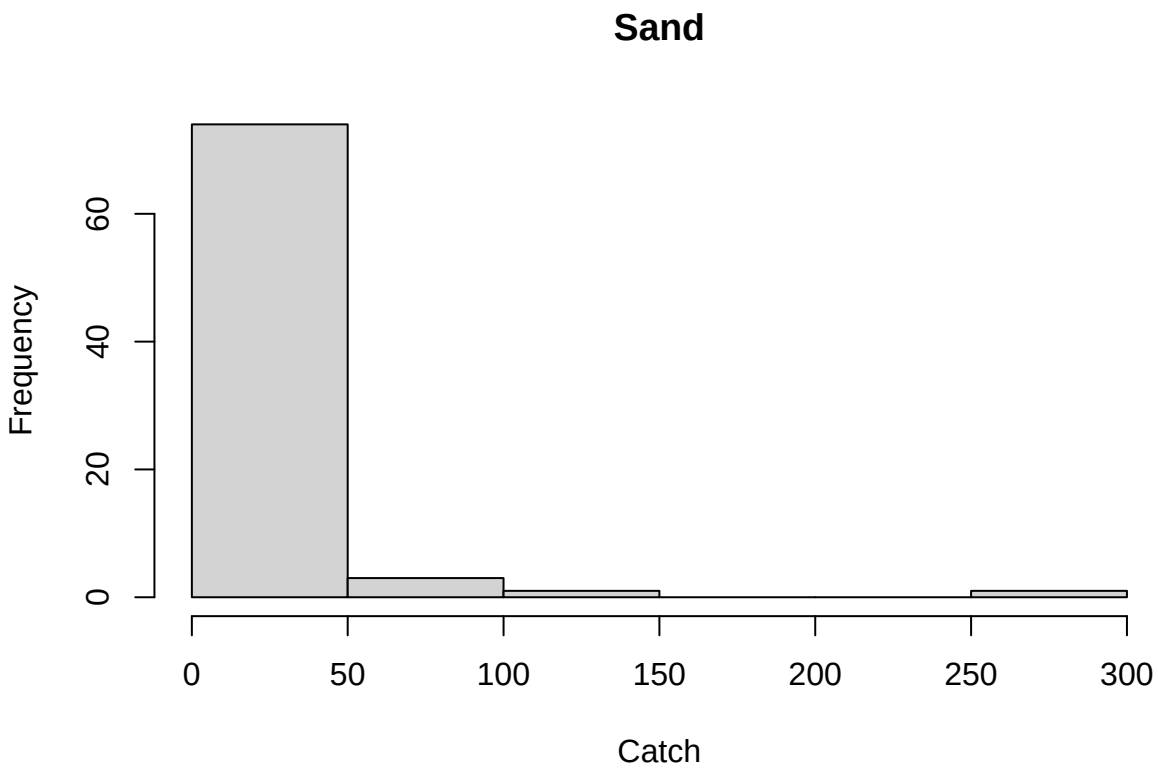
Checking assumptions Before we proceed, we need to consider the assumptions of ANOVA. Parametric tests like ANOVA assume normal population distribution, equal variance between groups, and independence between groups. It also helps to have about the same number of samples within each group. We can assume for now that the data are independent, because we did one net haul at each site about every two weeks. But we may have problems with the other assumptions.

We know that the number of samples at each type of substrate is unbalanced from our earlier tables. It also looks a lot like the distribution of catch is non-normal for each substrate type. Let's look more closely at this using histograms and tests of normality.

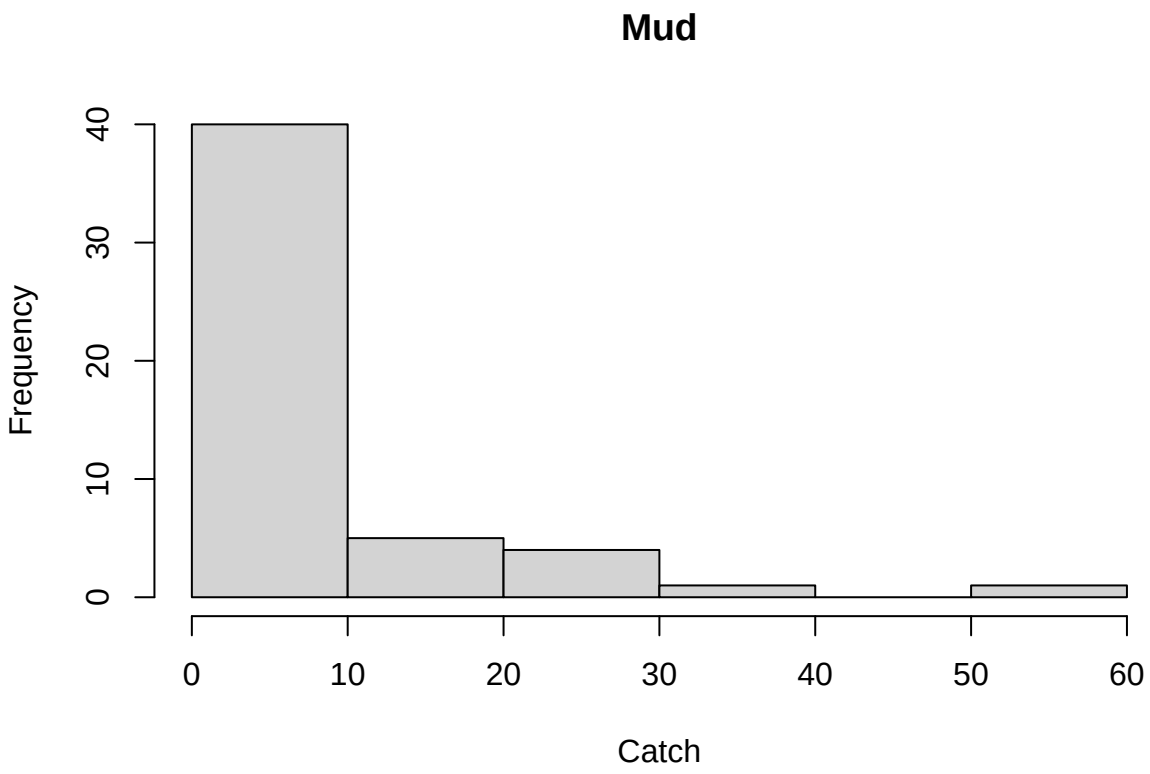
```
# We'll make a histogram of crabs caught for each substrate type
# Gravel
hist(greencrabs$catch[greencrabs$substrate == 'gravel'], # Subset gravel
     main='Gravel',                                      # Main title
     xlab='Catch')                                       # X axis name
```




```
# Sand
hist(greencrabs$catch[greencrabs$substrate == 'sand'],   # Subset sand
     main='Sand',                                        # Main title
     xlab='Catch')                                       # X axis name
```




```
# Gravel
hist(greencrabs$catch[greencrabs$substrate == 'mud'],     # Subset mud
     main='Mud',                                          # Main title
     xlab='Catch')                                        # X axis name
```


Mud
Frequency
0
10
20
30
40
0
10
20
30
40
50
60
Catch

These histograms are all very right-skewed, which means that the "tail" on the right side is much longer than it would be for a normal (bell-curve) distribution. We can also use a Shapiro-Wilk test of normality to draw a final conclusion, though normally I'd say these histograms are proof enough.

For a Shapiro-Wilk test of normality, the null hypothesis is a normal distribution. The alternative hypothesis is non-normality. If the p-value is greater than 0.05, we fail to reject the null hypothesis and can assume that the distribution is normal. If the p-value is less than 0.05, we reject the null hypothesis and can assume the alternative hypothesis of non-normality.

```
# Run Shapiro test on gravel substrate
shapiro.test(greencrabs$catch[greencrabs$substrate == 'gravel'])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  greencrabs$catch[greencrabs$substrate == "gravel"]
## W = 0.84515, p-value = 0.032
```

```
# Run Shapiro test on sand substrate
shapiro.test(greencrabs$catch[greencrabs$substrate == 'sand'])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  greencrabs$catch[greencrabs$substrate == "sand"]
## W = 0.38558, p-value < 2.2e-16
```

```
# Run Shapiro test on mud substrate
shapiro.test(greencrabs$catch[greencrabs$substrate == 'mud'])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  greencrabs$catch[greencrabs$substrate == "mud"]
## W = 0.69078, p-value = 4.483e-09
```

The Shapiro-Wilk test and all the histograms point towards the same conclusion– the data are not normally distributed. Let's keep that in mind as we test for equal variance. The simplest way to do this is to calculate the standard deviation for each group. If any one standard deviation is more than twice the value of another standard deviation, the assumption of equal variance is violated. Let's test that.

```
sd(greencrabs$catch[greencrabs$substrate == 'gravel'])
```

```
## [1] 2.249579
```

```
sd(greencrabs$catch[greencrabs$substrate == 'sand'])
```

```
## [1] 38.02423
```

```
sd(greencrabs$catch[greencrabs$substrate == 'mud'])
```

```
## [1] 10.30777
```

The calculations reveal that our smallest standard deviation value is ~2 (gravel) and the largest is ~38 (mud). This is a huge difference; the mud standard deviation is nearly 19 times bigger than the gravel standard deviation. The equal variance assumption is violated.

Selecting a test The violation of assumptions are problematic for using parametric statistical tests. Another problem is the unequal number of samples in each substrate type. Therefore, we should not use a parametric statistical test to compare these groups. Instead, we will use a non-parametric test. The non-parametric equivalent of ANOVA is called Kruskal-Wallis, and it relaxes the assumption of normality and equal variance. It still requires independence of the data.

The null hypothesis of the Kruskal test is equal means of all groups. The alternative hypothesis is that at least two groups have means that are statistically different. If the p-value is greater than 0.05, we fail to reject the null hypothesis and can assume that the mean response of all groups is statistically the same. If the p-value is less than 0.05, we reject the null hypothesis and can assume that the groups have different means.

```
kruskal.test(greencrabs$catch ~ greencrabs$substrate)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  greencrabs$catch by greencrabs$substrate
## Kruskal-Wallis chi-squared = 7.971, df = 2, p-value = 0.01858
```

Interpreting results The resulting p-value is less than 0.05, which means at least two of our three groups have statistically different means. The Kruskal-Wallis test does not tell us which grous have different means. For that, we need to use a Dunn test of multiple comparisons. You will need to install a new package called `dunn.test`. Do not do package installations in RMarkdown documents like this. Instead, click over to your console (bottom left) and type in `install.packages("dunn.test")`. Then proceed with this code chunk.

```
library(dunn.test)
dunn.test(x = greencrabs$catch,         # Numeric variable
          g = greencrabs$substrate,     # Categorical variable
          method = "bonferroni",        # p-adjustment for multiple comparisons
          list = TRUE,                  # Show us output in list format
          table = FALSE)                # Do not show us output in table format
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 7.971, df = 2, p-value = 0.02
##
##
##                            Comparison of x by group
##                                  (Bonferroni)
##
## List of pairwise comparisons: Z statistic (adjusted p-value)
## ----------------------------------
## gravel - mud  : -1.276372 (0.3027)
## gravel - sand : -2.442941 (0.0219)*
## mud - sand    : -1.933819 (0.0797)
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

```
# You can play around with table vs. list for output. I think list is easier to read.
```

There is a statistically-significant difference in means between the gravel and sand group. The difference in means between the gravel and mud groups is not statistically significant, nor is the difference between the mud and sand groups.

Drawing conclusions

We have now gone through the process of developing a research question, determining the data we need to address that question, manipulating the data to a format we can use in R, checking assumptions for statistical tests, selecting a test, and executing that test. Now, we can talk about our findings.

Sampling site substrate seems to have an effect on the number of crabs caught. We found that mean crab abundance was statistically significant when comparing gravel and sand sites, with abundance at sand sites being much higher. There was no statistically significant difference between the abundance of crabs at gravel and mud or mud and sand sites, though typically fewer crabs were caught at mud sites than sand sites.

At this point in a scientific article or report, we'd talk about why we think we got this result. This was mostly meant to be an example on data manipulation, so I'll keep the discussion brief. Green crabs burrow in sediment to camouflage themselves from predators. This is less possible in substrate with lots of gravel. Green crabs are generalists, and can feed on plants, gastropods, bivalves, amphipods, polychaetes, and other crustaceans (including smaller green crabs). We have no evidence to prove it, but it's possible that food availability is better in sandy or muddy habitats rather than gravel. At any rate, the result is supported both by our statistical tests and what is generally known about green crab behavior and distribution.