

# MEGAHAND

*Joe*

*11/8/2018*

## Interoperability

```
# install.packages("tidyverse")
# install.packages("reticulate", dependencies = TRUE)

library(tidyverse)

## -- Attaching packages ----- tidyverse 1.2.1 --
## v ggplot2 3.0.0     v purrr   0.2.5
## v tibble   1.4.2     v dplyr    0.7.6
## v tidyverse 0.8.1     v stringr  1.3.1
## v readr    1.1.1     vforcats  0.3.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()

library(reticulate)
```

With the reticulate package in R, Python code can be integrated into R documents and used alongside R. This is especially convenient in the RMarkdown document format for several reasons:

- R code and Python code can be called in discrete boxes, but within the same document
- Objects built in either environment can be passed back and forth between languages
- RMarkdown offers flexible export formats including pdf, slides, word, and html

This particular aspect of our project interested me due to the scale and diversity of challenges in interoperability.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

## Exploratory Data Analysis and Visualization

Here, I created a Python script that grabs all of the “.csv” files in a folder, and makes a list of the names. The script is saved as “TrainingDataGrabber.py”

```
import os
import glob
path = 'c:\\\\'
extension = 'csv'
os.chdir(path= "C:/Users/joeje/Desktop/Academics/FAES/Intro_to_Python/MEGAHAND/TrainingData")
Training_Data_Files = [i for i in glob.glob('*.{format(extension)})]
print(Training_Data_Files)
```

Here, I used R to source the Python script, create a list object containing all of the file names in the “TrainingData” folder, and then coerced an R DataFrame from that list for display.

```
reticulate::source_python("TrainingDataGrabber.py")  
  
Training_Data_Files  
  
## [1] "Chuck Grip.csv"      "Fine Pinch.csv"      "H. Open.csv"  
## [4] "Hook Grip.csv"       "Key Grip.csv"       "No Move.csv"  
## [7] "Power Grip.csv"      "Thumb Enclosed.csv" "Tool Grip.csv"  
## [10] "W. Abduction.csv"    "W. Adduction.csv"   "W. Extension.csv"  
## [13] "W. Flexion.csv"     "W. Pronation.csv"   "W. Supination.csv"  
  
as.data.frame(Training_Data_Files)  
  
##      Training_Data_Files  
## 1      Chuck Grip.csv  
## 2      Fine Pinch.csv  
## 3      H. Open.csv  
## 4      Hook Grip.csv  
## 5      Key Grip.csv  
## 6      No Move.csv  
## 7      Power Grip.csv  
## 8      Thumb Enclosed.csv  
## 9      Tool Grip.csv  
## 10     W. Abduction.csv  
## 11     W. Adduction.csv  
## 12     W. Extension.csv  
## 13     W. Flexion.csv  
## 14     W. Pronation.csv  
## 15     W. Supination.csv
```

Next, I used the `purrr` package from R to apply a function I made in R that tidys the data (removing extraneous columns and formatting) and then creates a pre-set visualization for all of the files from the list (that was made in Python.)

```
source("C:/Users/joeje/Desktop/Academics/FAES/Intro_to_Python/MEGAHAND/Megamunge_Jitter.R")
library(purrr)
setwd('TrainingData')
map(Training_Data_Files, Megamunge)

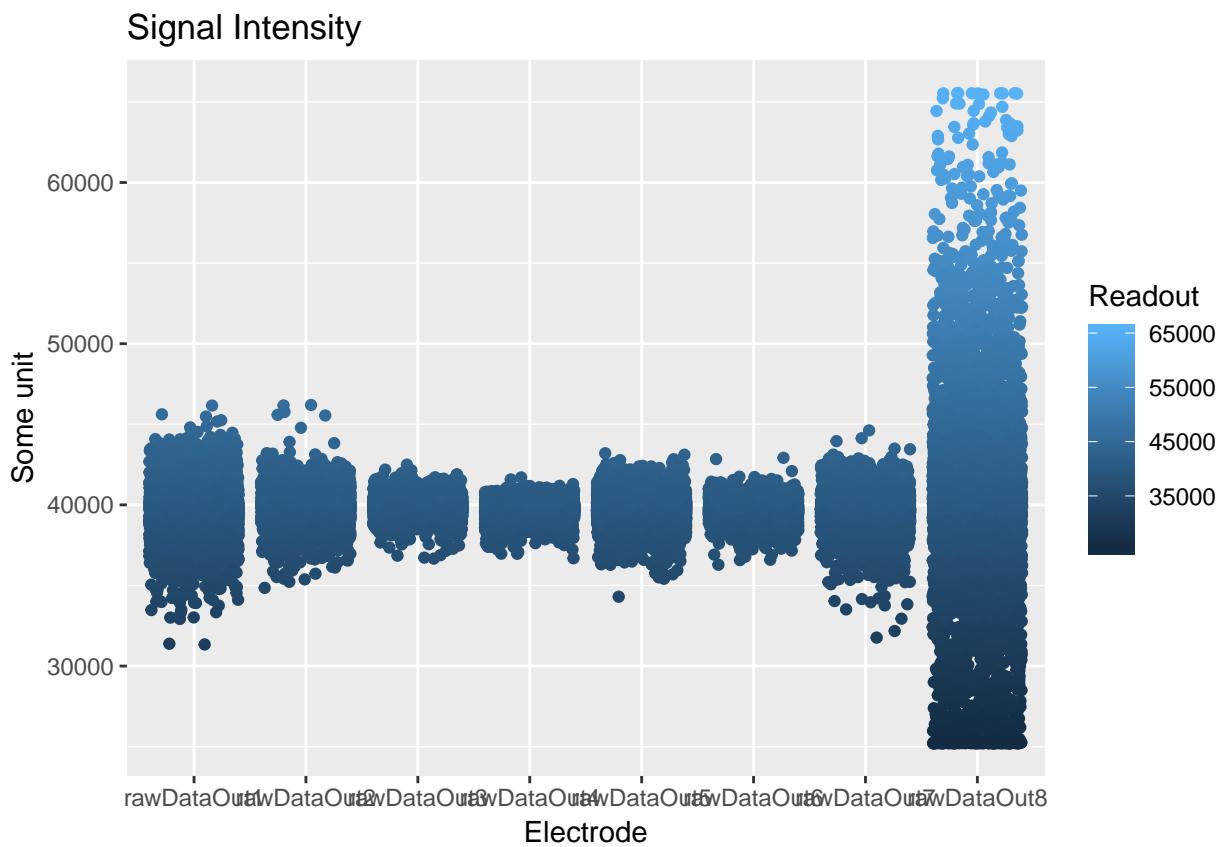
## Parsed with column specification:
## cols(
##   rawDataOut1 = col_integer(),
##   rawDataOut2 = col_integer(),
##   rawDataOut3 = col_integer(),
##   rawDataOut4 = col_integer(),
##   rawDataOut5 = col_integer(),
##   rawDataOut6 = col_integer(),
##   rawDataOut7 = col_integer(),
##   rawDataOut8 = col_integer(),
##   Action = col_character()
## )
## Parsed with column specification:
## cols(
##   rawDataOut1 = col_integer(),
##   rawDataOut2 = col_integer(),
##   rawDataOut3 = col_integer(),
##   rawDataOut4 = col_integer(),
```





```
## rawDataOut5 = col_integer(),
## rawDataOut6 = col_integer(),
## rawDataOut7 = col_integer(),
## rawDataOut8 = col_integer(),
## Action = col_character()
## )
## Parsed with column specification:
## cols(
##   rawDataOut1 = col_integer(),
##   rawDataOut2 = col_integer(),
##   rawDataOut3 = col_integer(),
##   rawDataOut4 = col_integer(),
##   rawDataOut5 = col_integer(),
##   rawDataOut6 = col_integer(),
##   rawDataOut7 = col_integer(),
##   rawDataOut8 = col_integer(),
##   Action = col_character()
## )
## Parsed with column specification:
## cols(
##   rawDataOut1 = col_integer(),
##   rawDataOut2 = col_integer(),
##   rawDataOut3 = col_integer(),
##   rawDataOut4 = col_integer(),
##   rawDataOut5 = col_integer(),
##   rawDataOut6 = col_integer(),
##   rawDataOut7 = col_integer(),
##   rawDataOut8 = col_integer(),
##   Action = col_character()
## )
## Parsed with column specification:
## cols(
##   rawDataOut1 = col_integer(),
##   rawDataOut2 = col_integer(),
##   rawDataOut3 = col_integer(),
##   rawDataOut4 = col_integer(),
##   rawDataOut5 = col_integer(),
##   rawDataOut6 = col_integer(),
##   rawDataOut7 = col_integer(),
##   rawDataOut8 = col_integer(),
##   Action = col_character()
## )
## Parsed with column specification:
## cols(
##   rawDataOut1 = col_integer(),
##   rawDataOut2 = col_integer(),
##   rawDataOut3 = col_integer(),
##   rawDataOut4 = col_integer(),
##   rawDataOut5 = col_integer(),
##   rawDataOut6 = col_integer(),
##   rawDataOut7 = col_integer(),
##   rawDataOut8 = col_integer(),
##   Action = col_character()
## )
## Parsed with column specification:
## cols(
##   rawDataOut1 = col_integer(),
##   rawDataOut2 = col_integer(),
##   rawDataOut3 = col_integer(),
##   rawDataOut4 = col_integer(),
##   rawDataOut5 = col_integer(),
##   rawDataOut6 = col_integer(),
##   rawDataOut7 = col_integer(),
##   rawDataOut8 = col_integer(),
##   Action = col_character()
## )
```

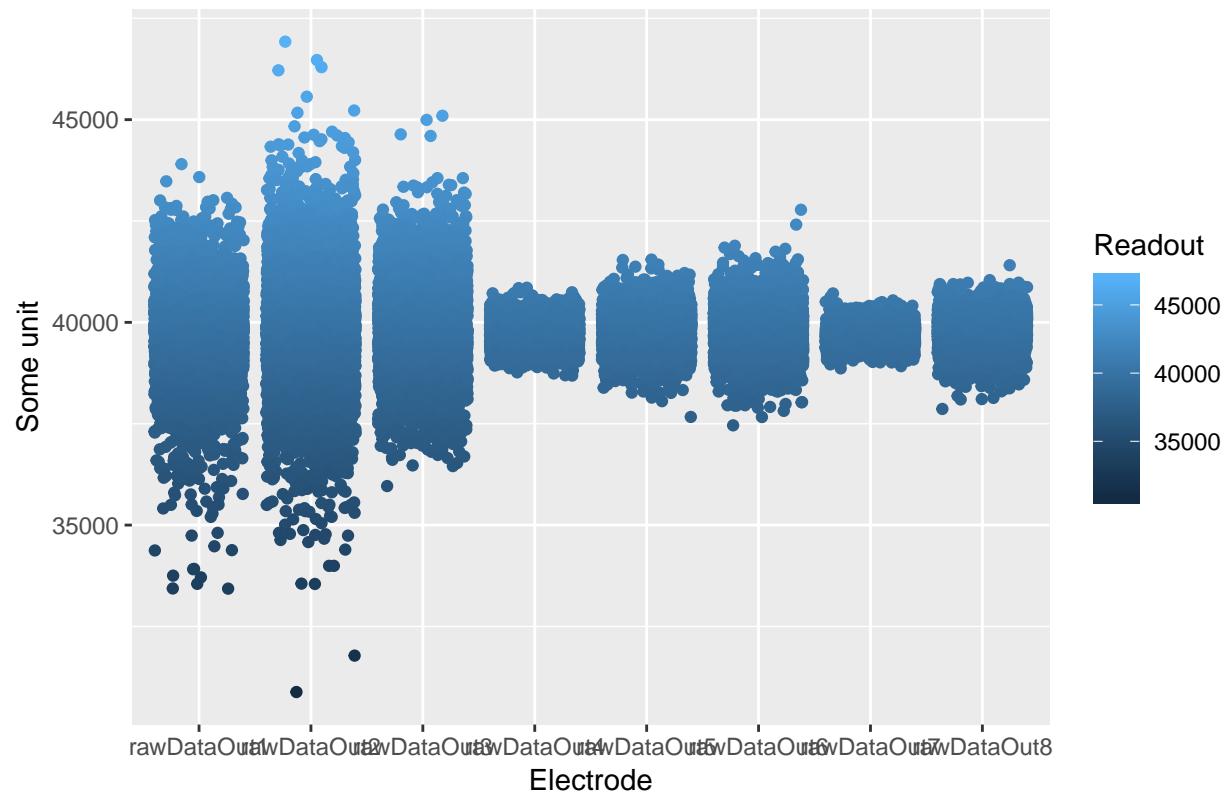
```
## [[1]]
```



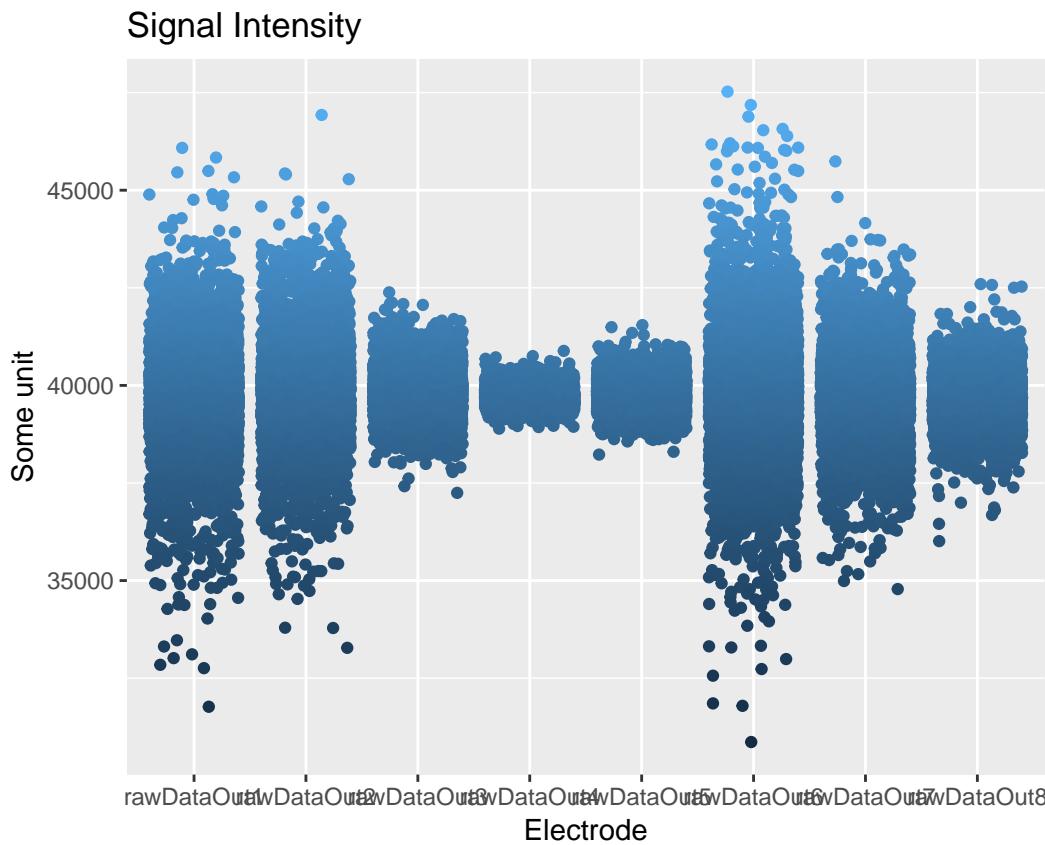
```
##
```

```
## [[2]]
```

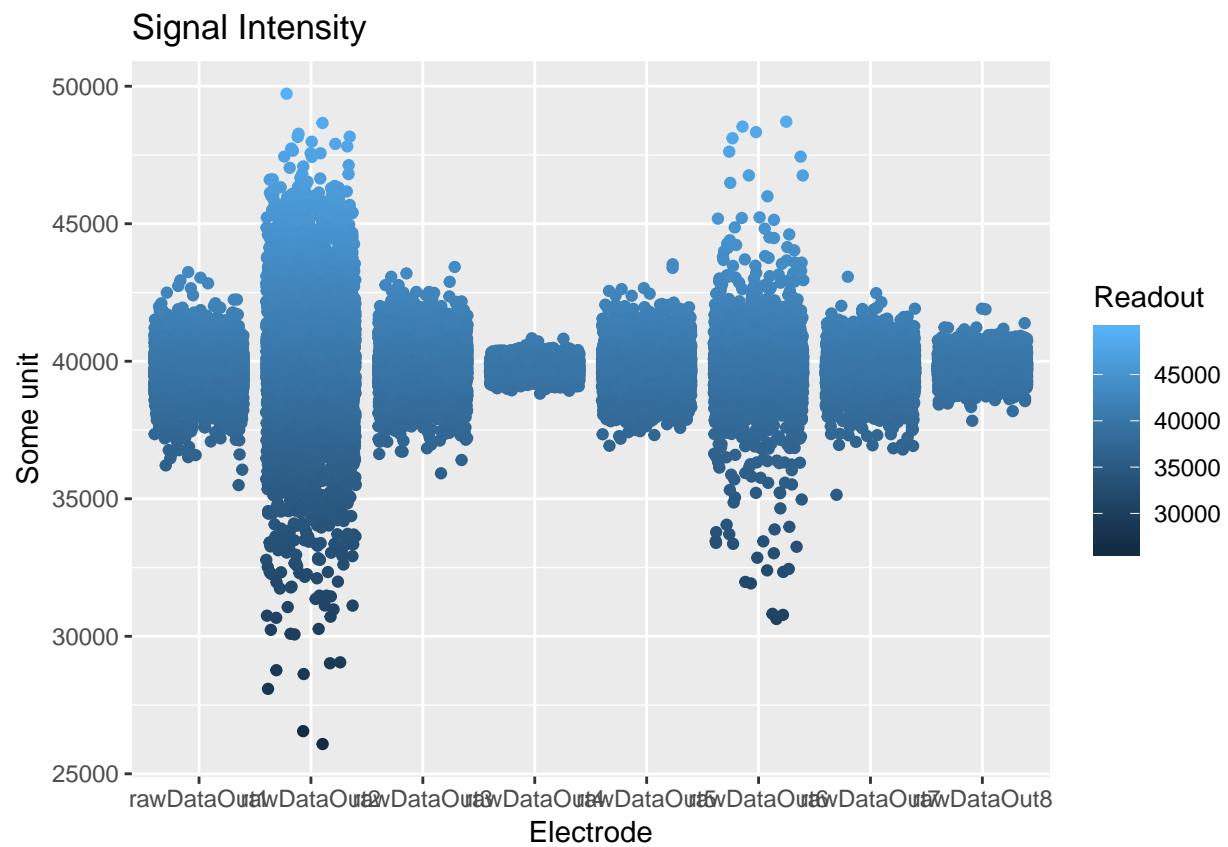
## Signal Intensity



```
##  
## [[3]]
```

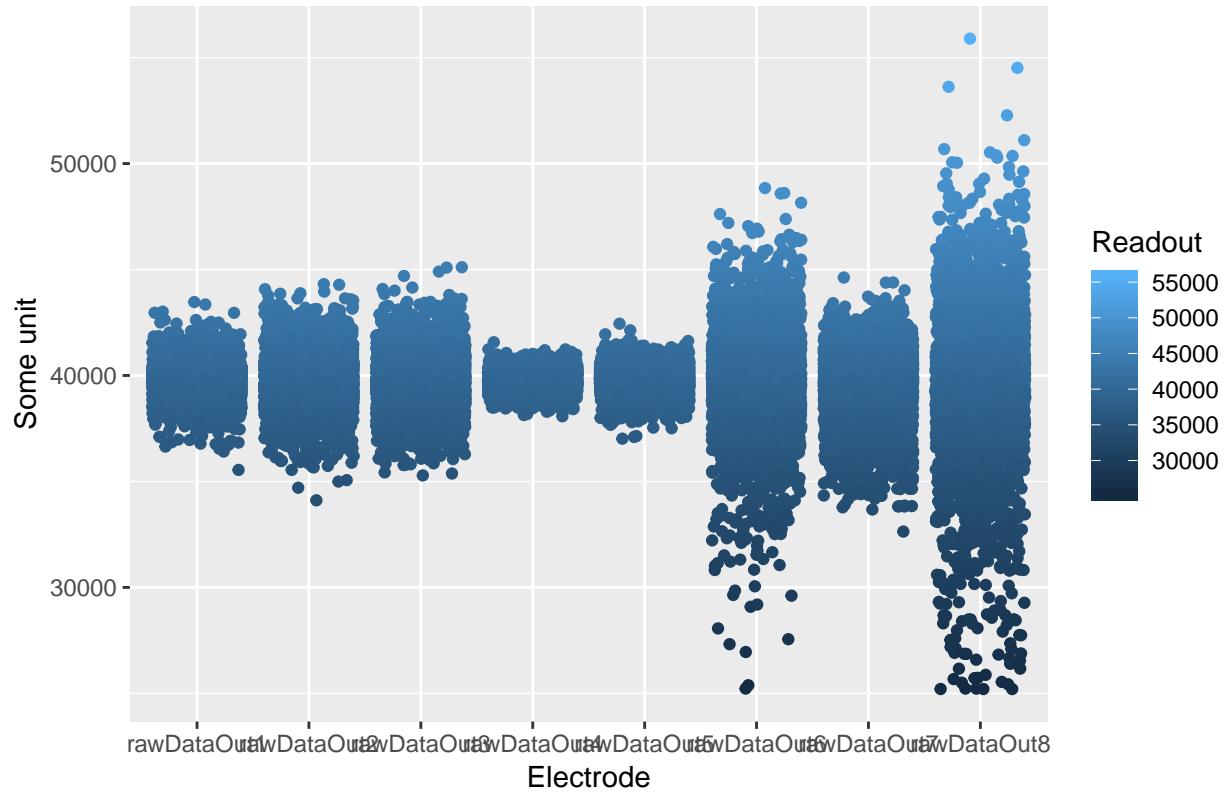


```
##  
## [[4]]
```

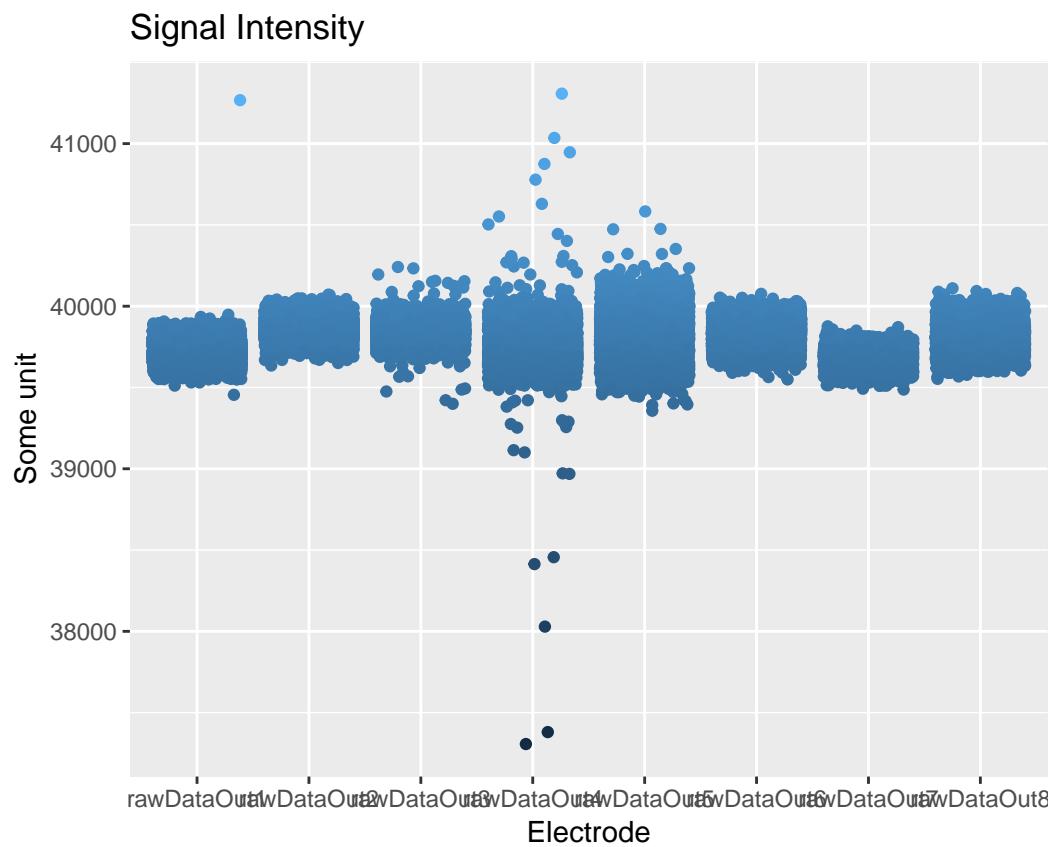


```
##  
## [[5]]
```

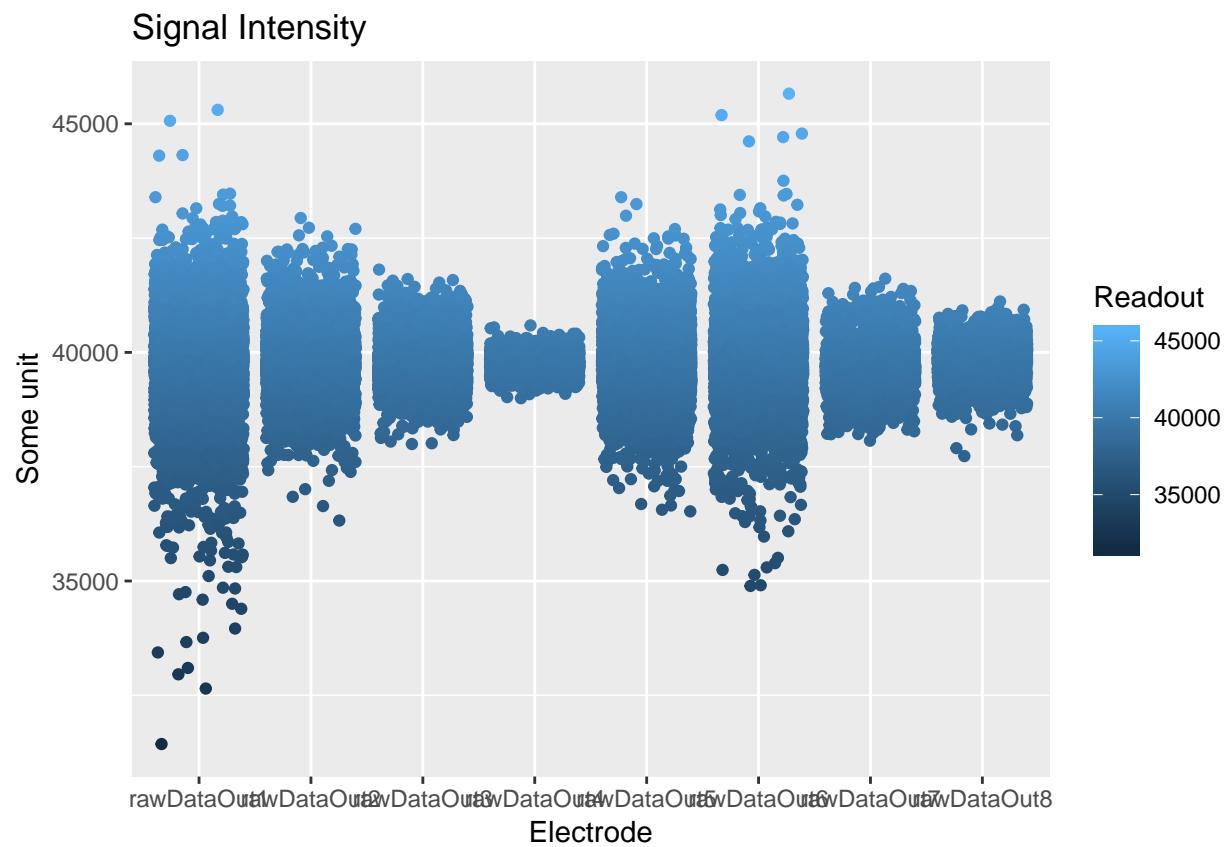
## Signal Intensity



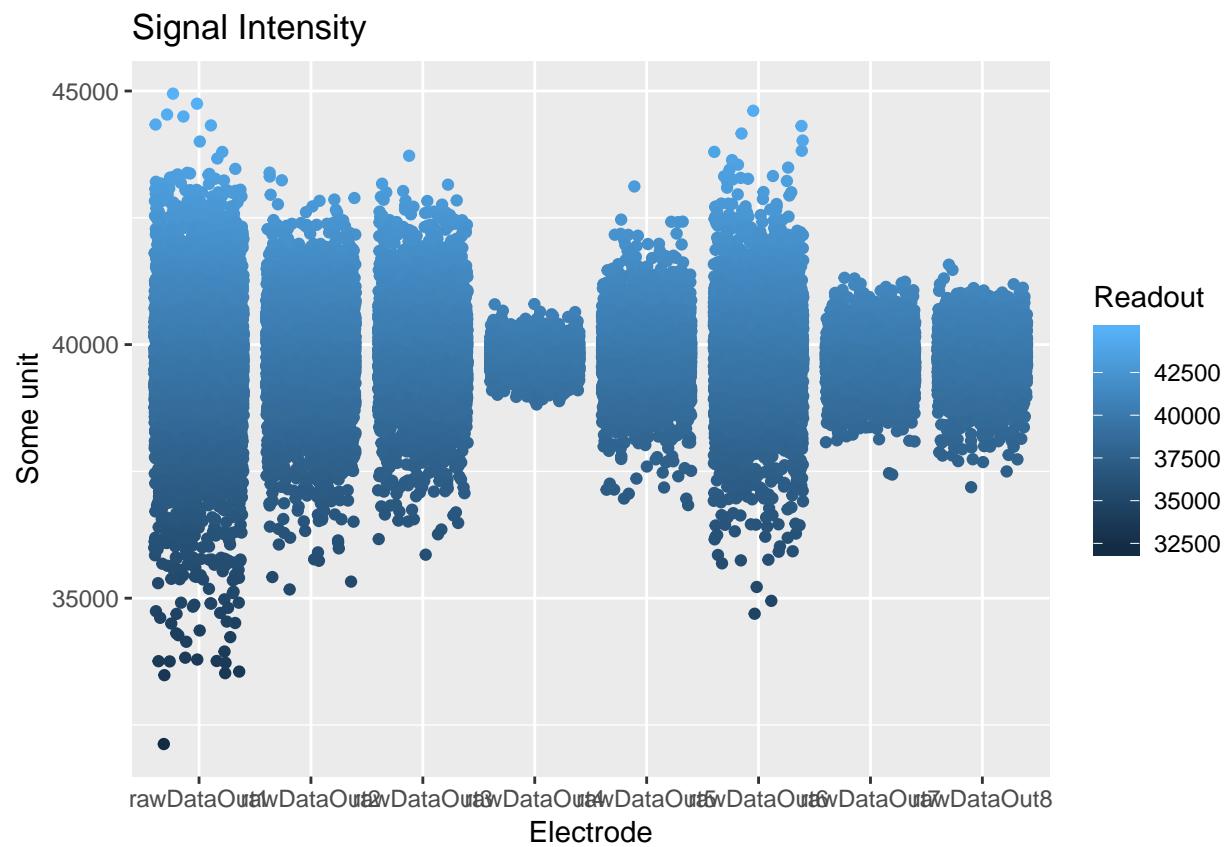
```
##  
## [[6]]
```



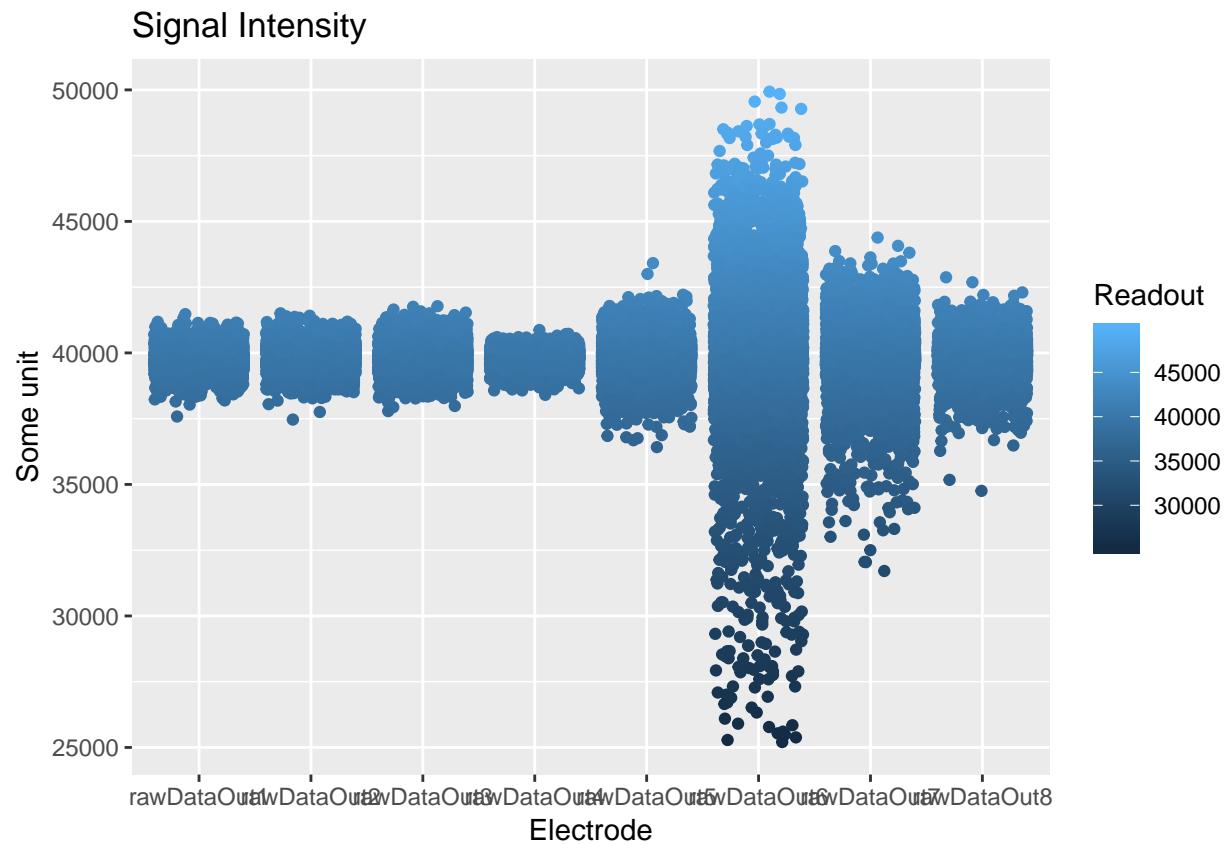
```
##  
## [[7]]
```



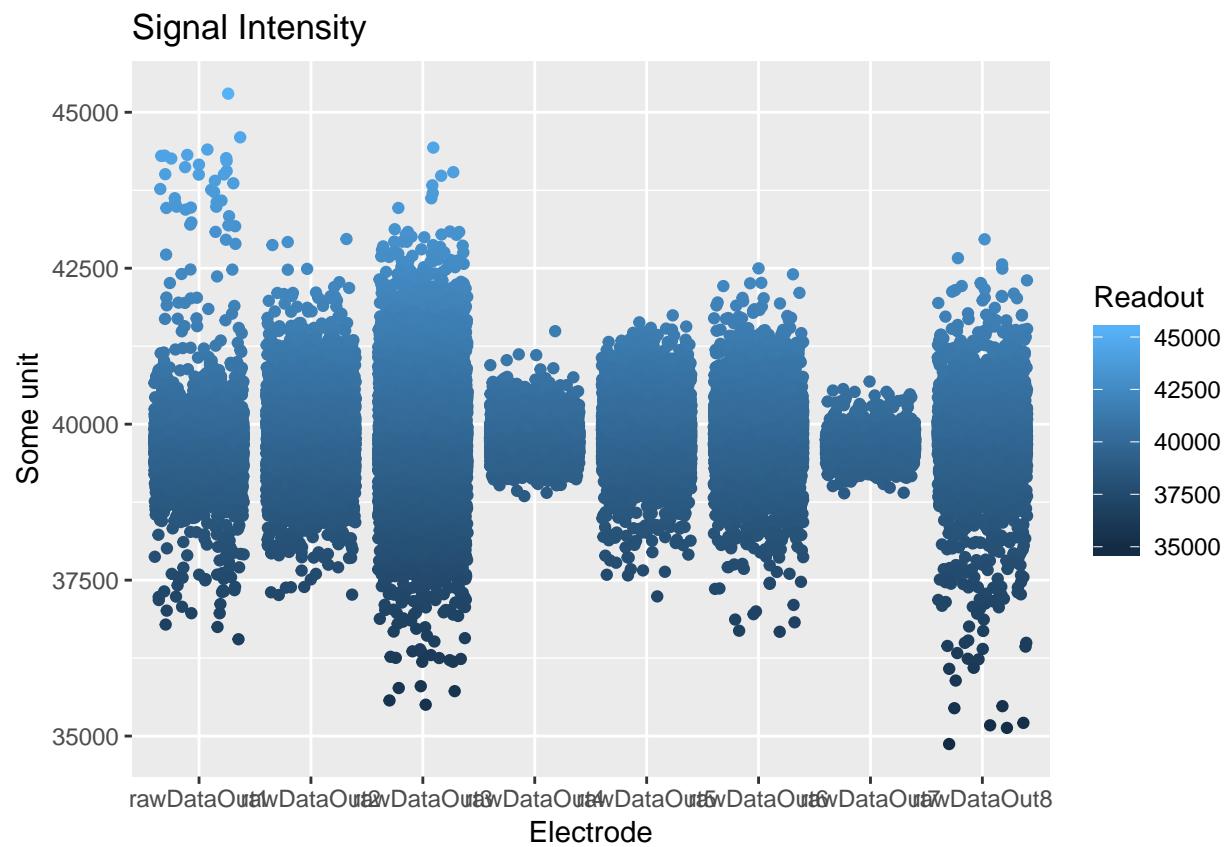
```
##  
## [[8]]
```



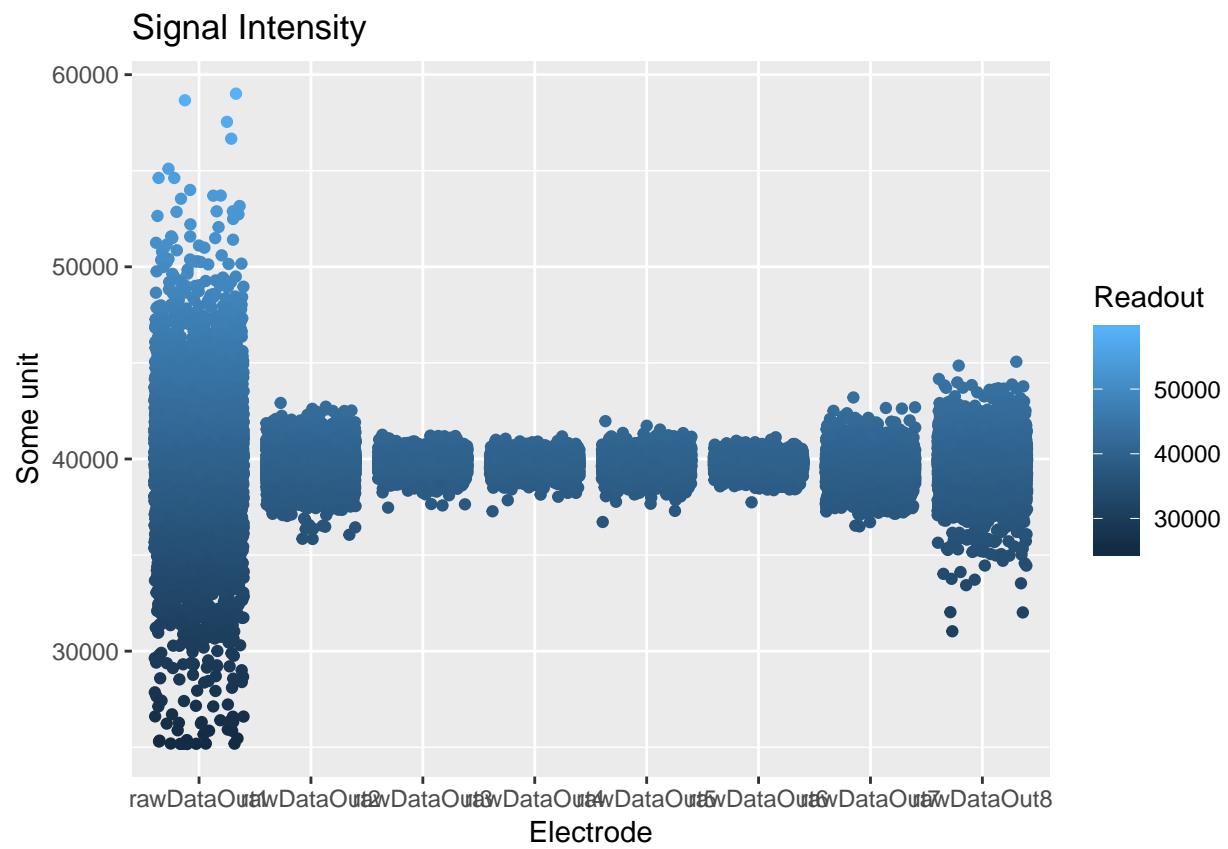
```
##  
## [[9]]
```



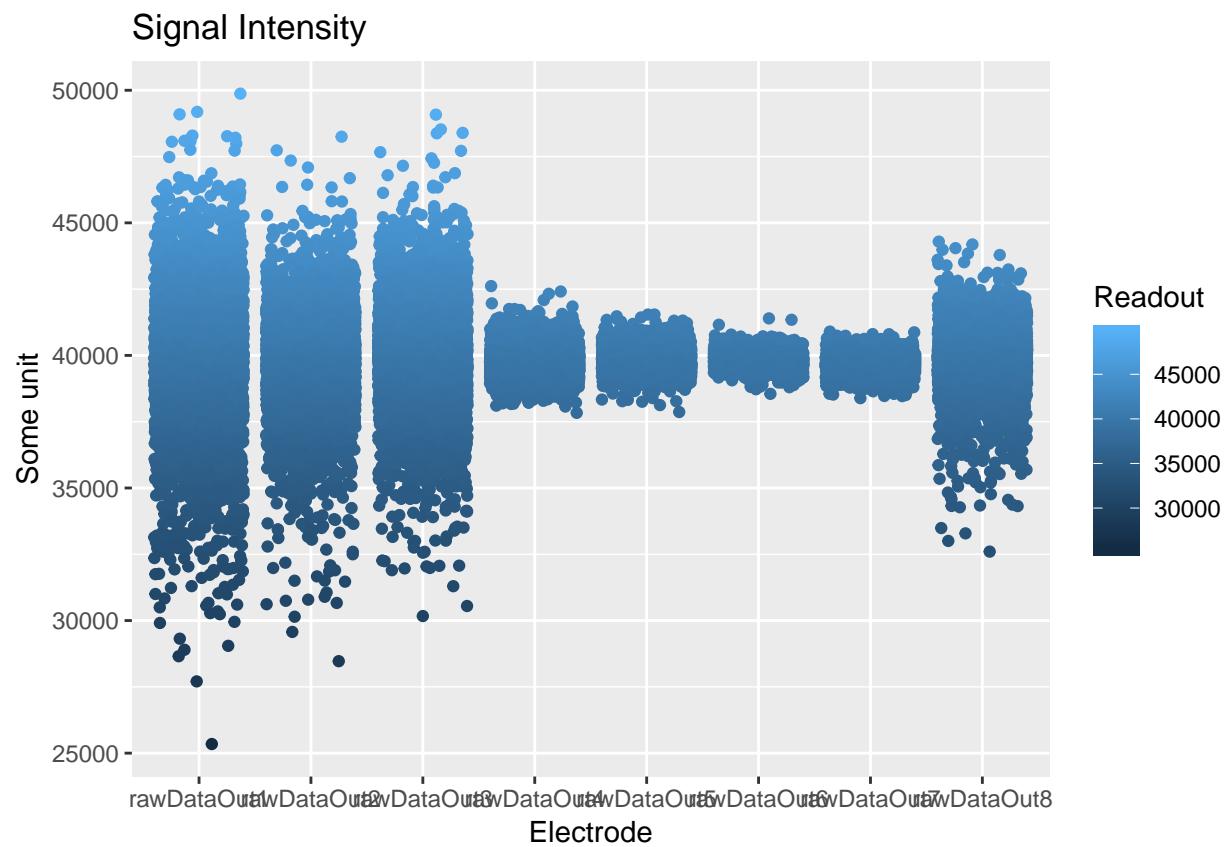
```
##  
## [[10]]
```



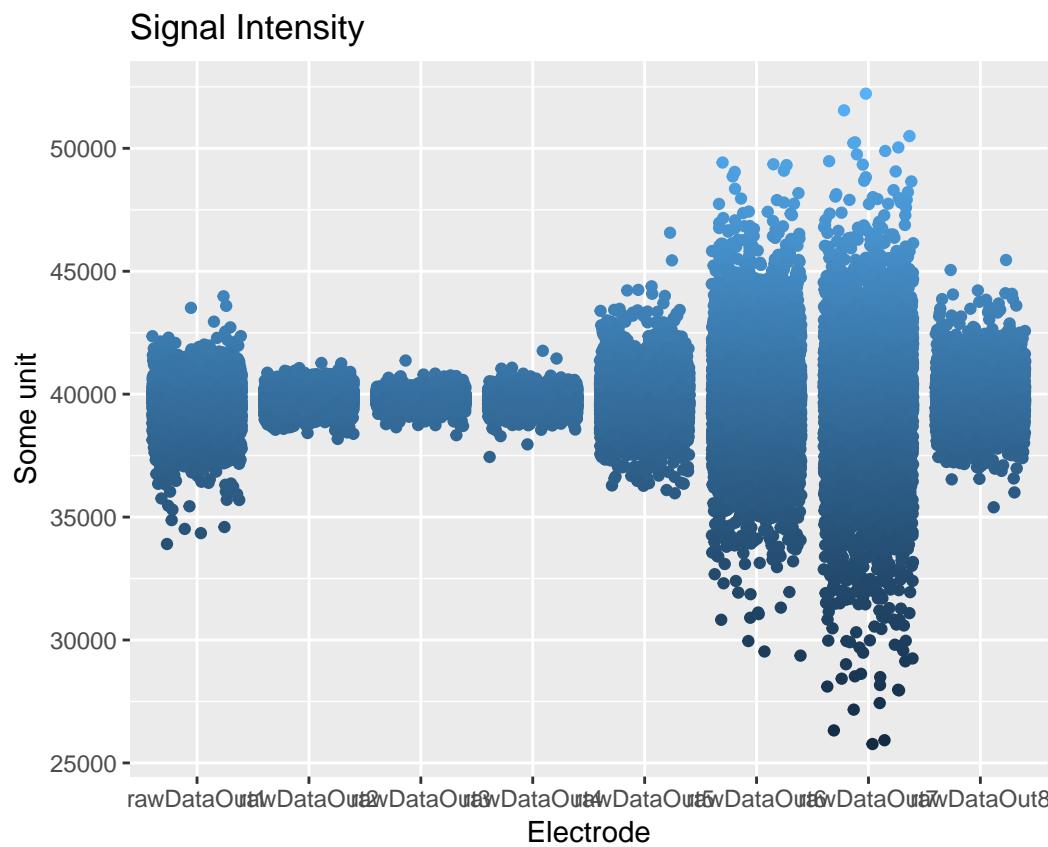
```
##  
## [[11]]
```



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

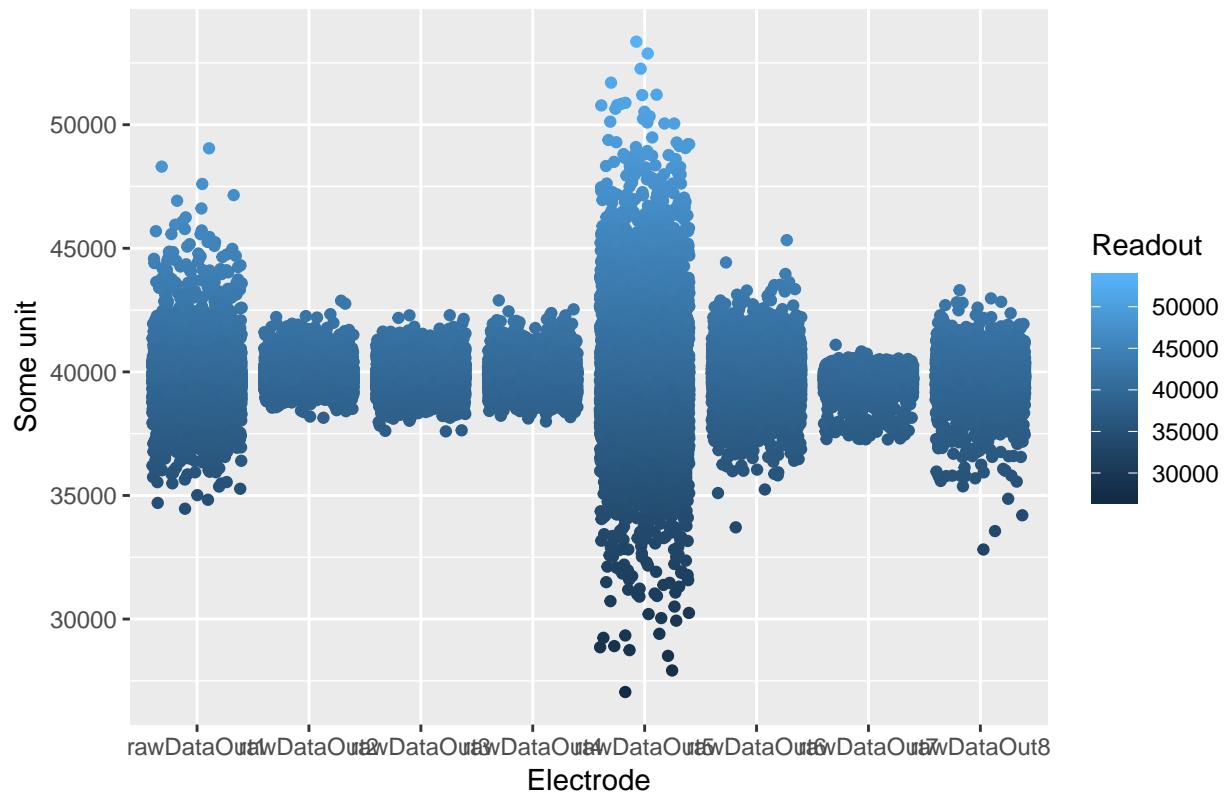


```
##  
## [[13]]
```



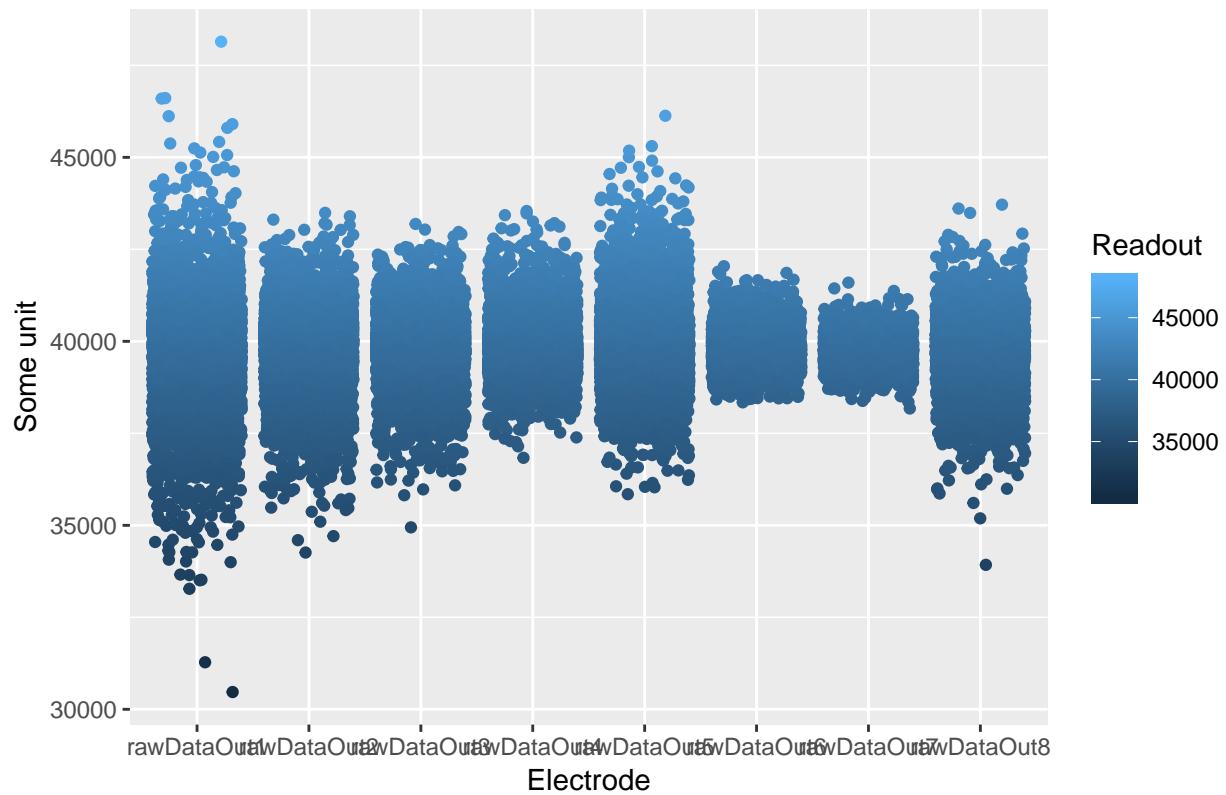
```
##  
## [[14]]
```

## Signal Intensity



```
##  
## [[15]]
```

## Signal Intensity



## Machine Learning

```
install.packages("tensorflow", dependencies = TRUE)
install.packages("keras", dependencies = TRUE)
library(tensorflow)
library(keras)
```