

# MEGAHAND

*Joe*

11/8/2018

## Interoperability

```
# install.packages("tidyverse")
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.0.0
##   v ggplot2 3.0.0     v purrr    0.2.5
##   v tibble   1.4.2     v dplyr    0.7.6
##   v tidyr    0.8.1     v stringr  1.3.1
##   v readr    1.1.1     v forcats 0.3.0

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

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

This is 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.)

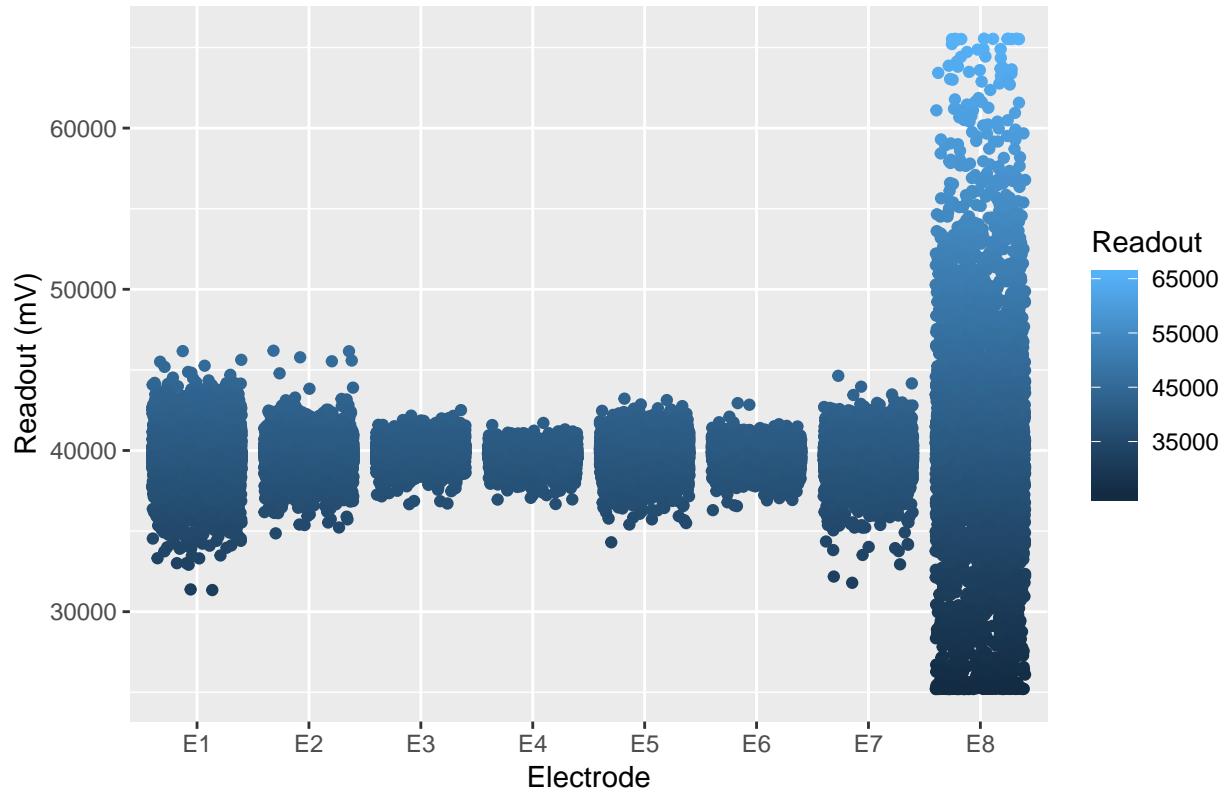
```

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

## [[1]]

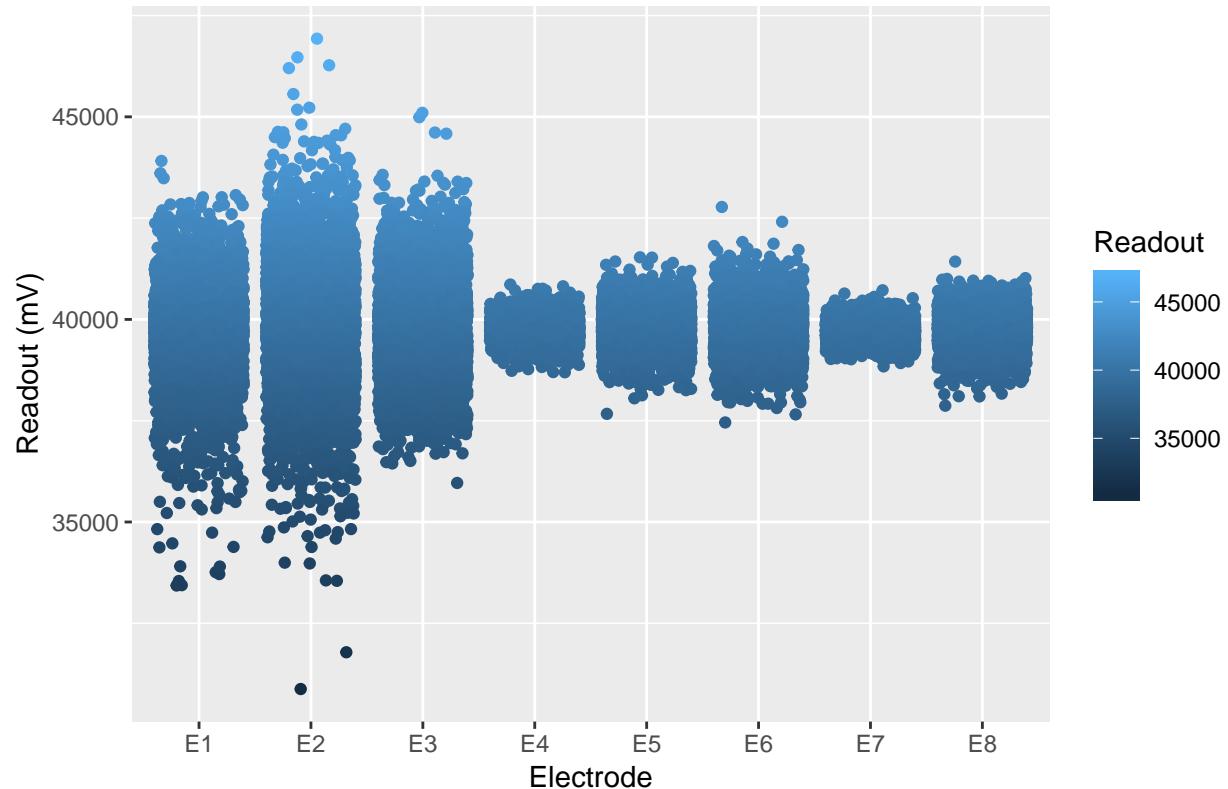
```

## Signal Intensities for Chuck Grip



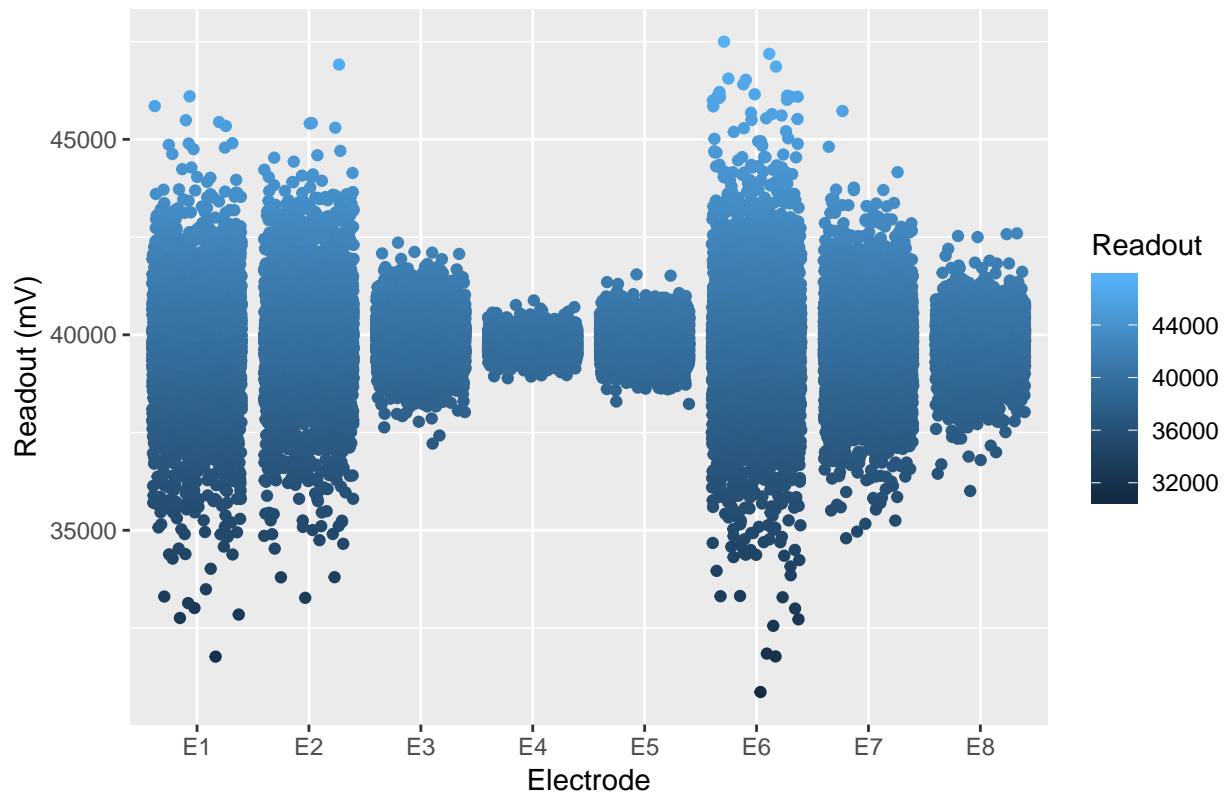
```
##  
## [[2]]
```

## Signal Intensities for Fine Pinch



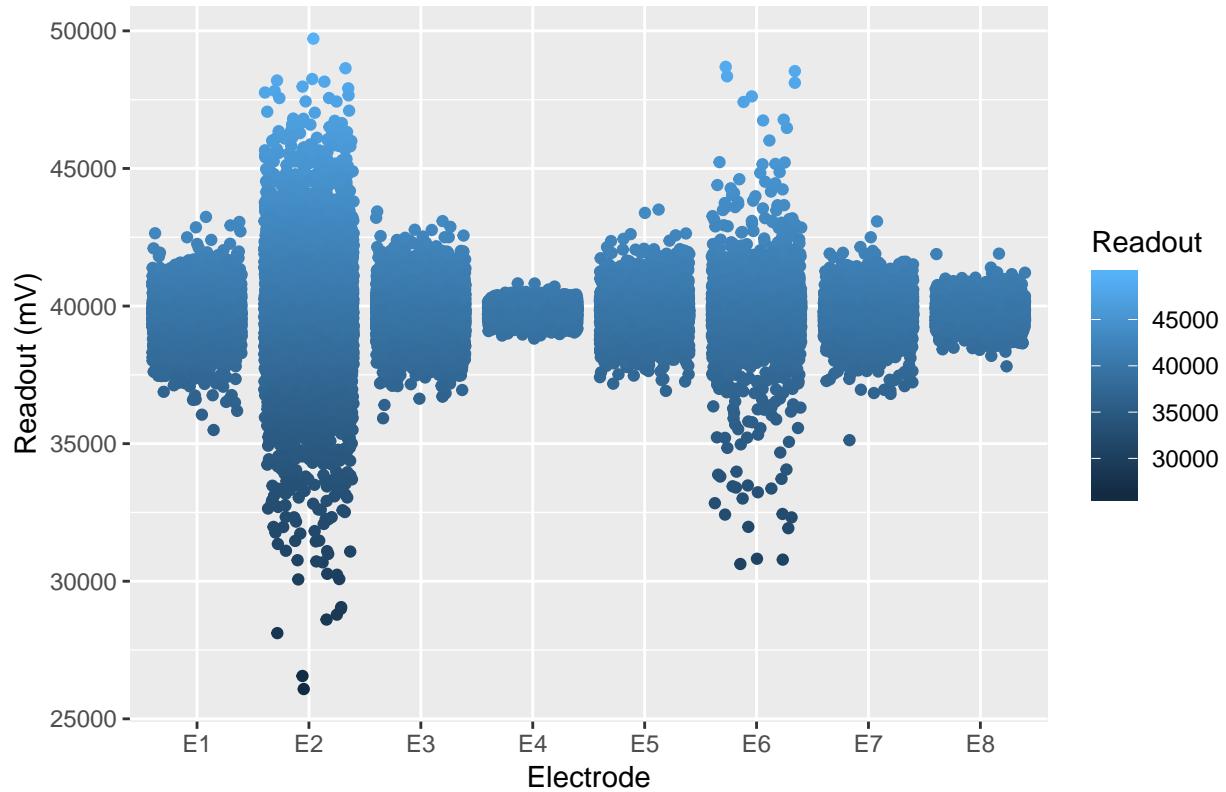
```
##  
## [[3]]
```

## Signal Intensities for H. Open



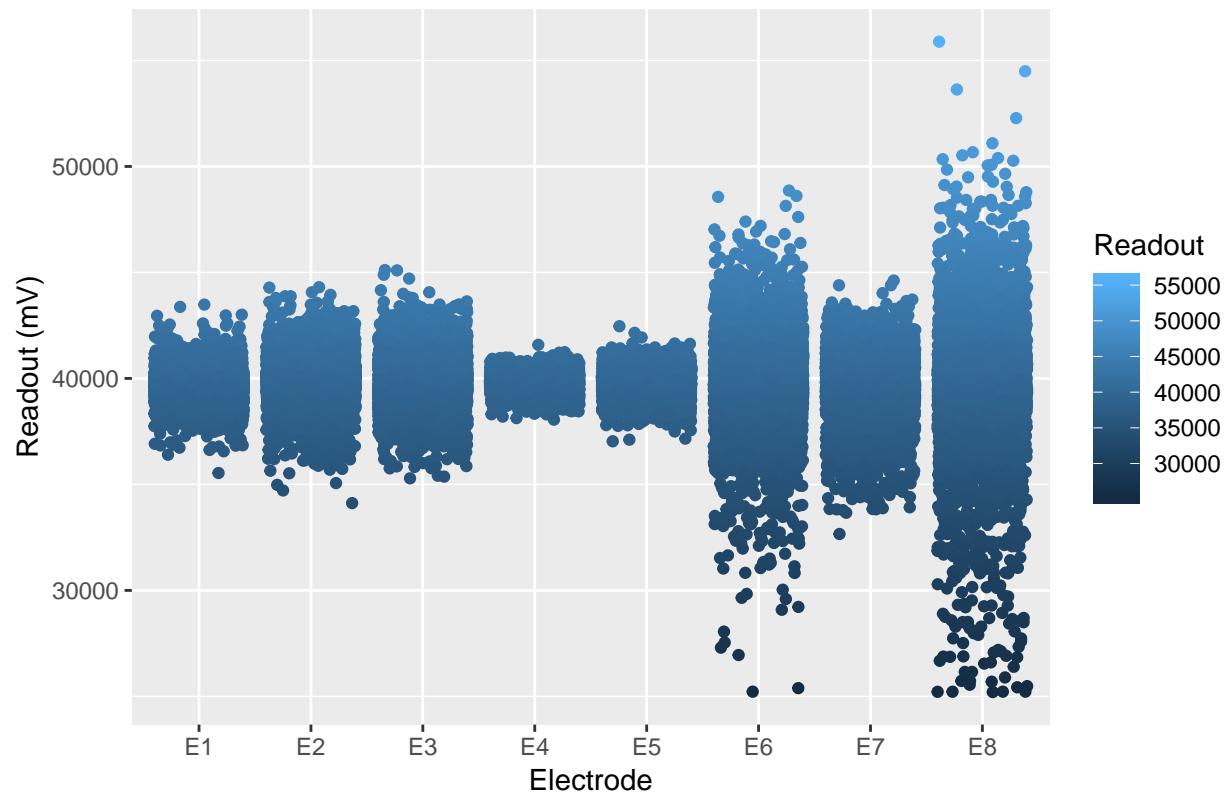
```
##  
## [[4]]
```

## Signal Intensities for Hook Grip



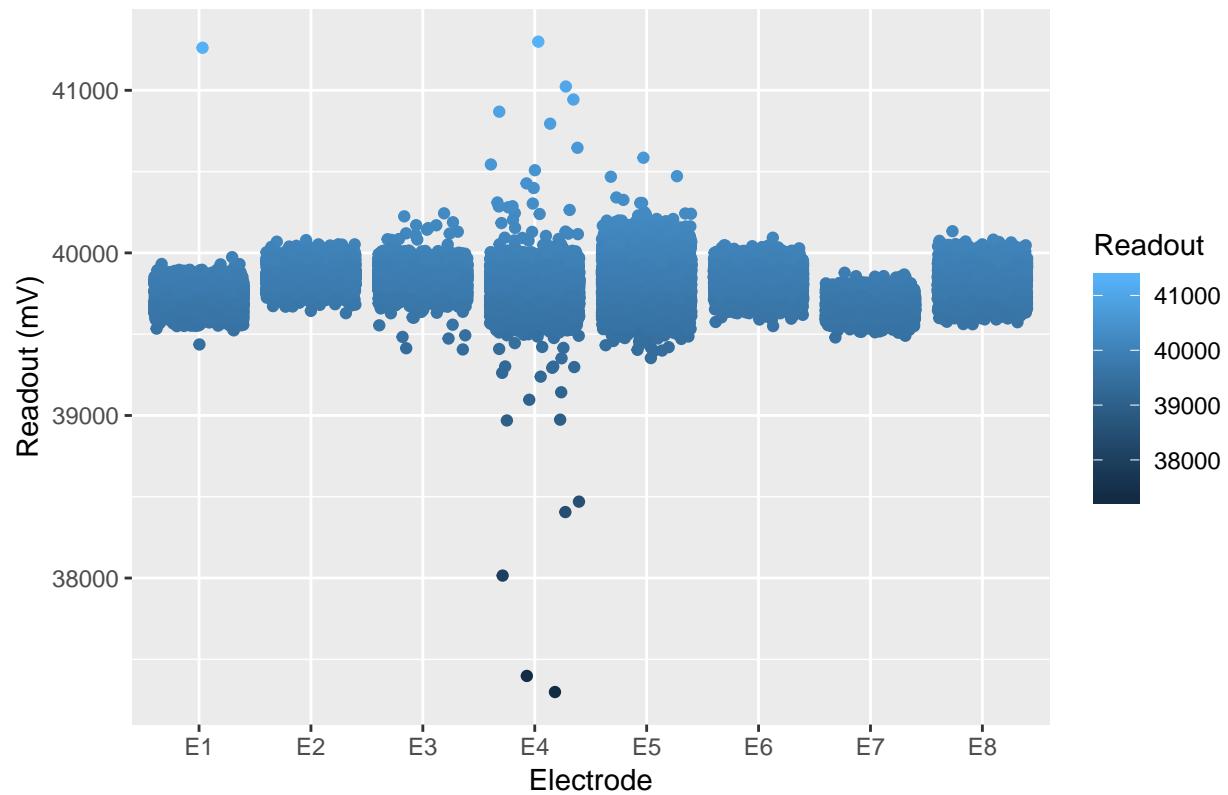
```
##  
## [[5]]
```

## Signal Intensities for Key Grip



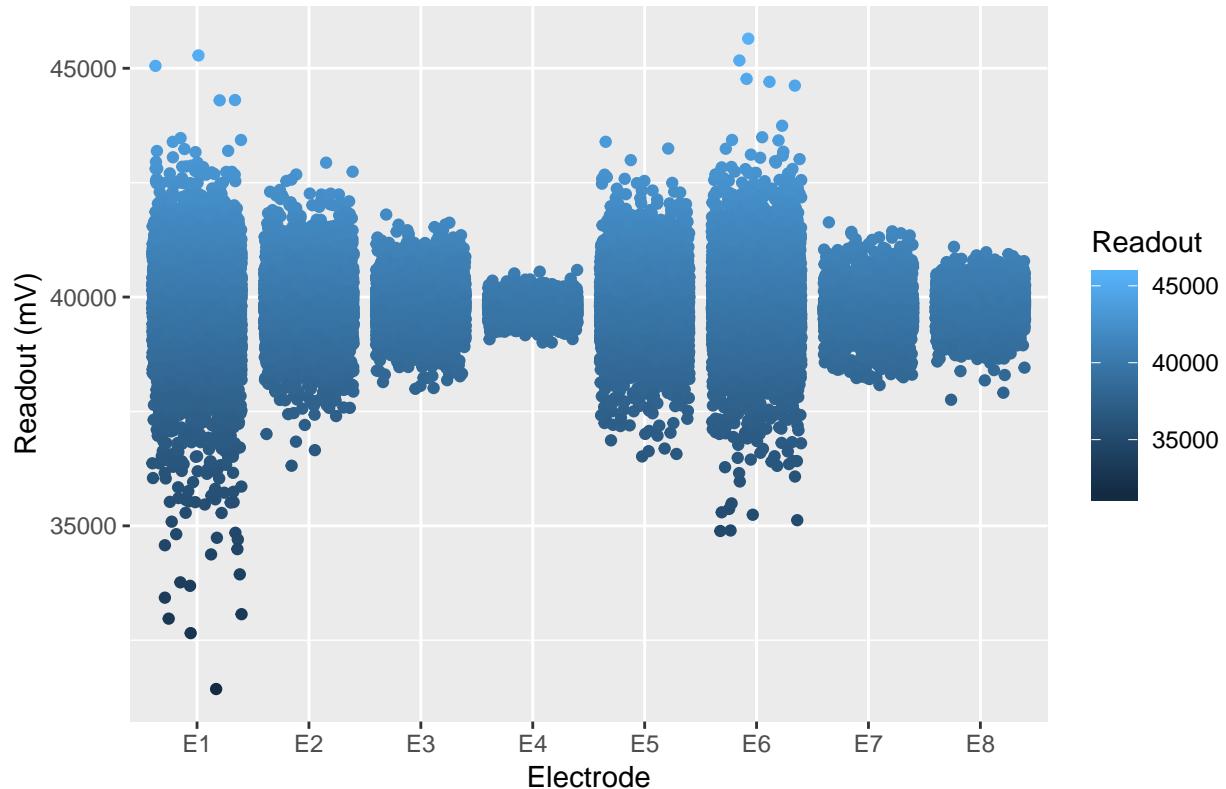
```
##  
## [[6]]
```

## Signal Intensities for No Move



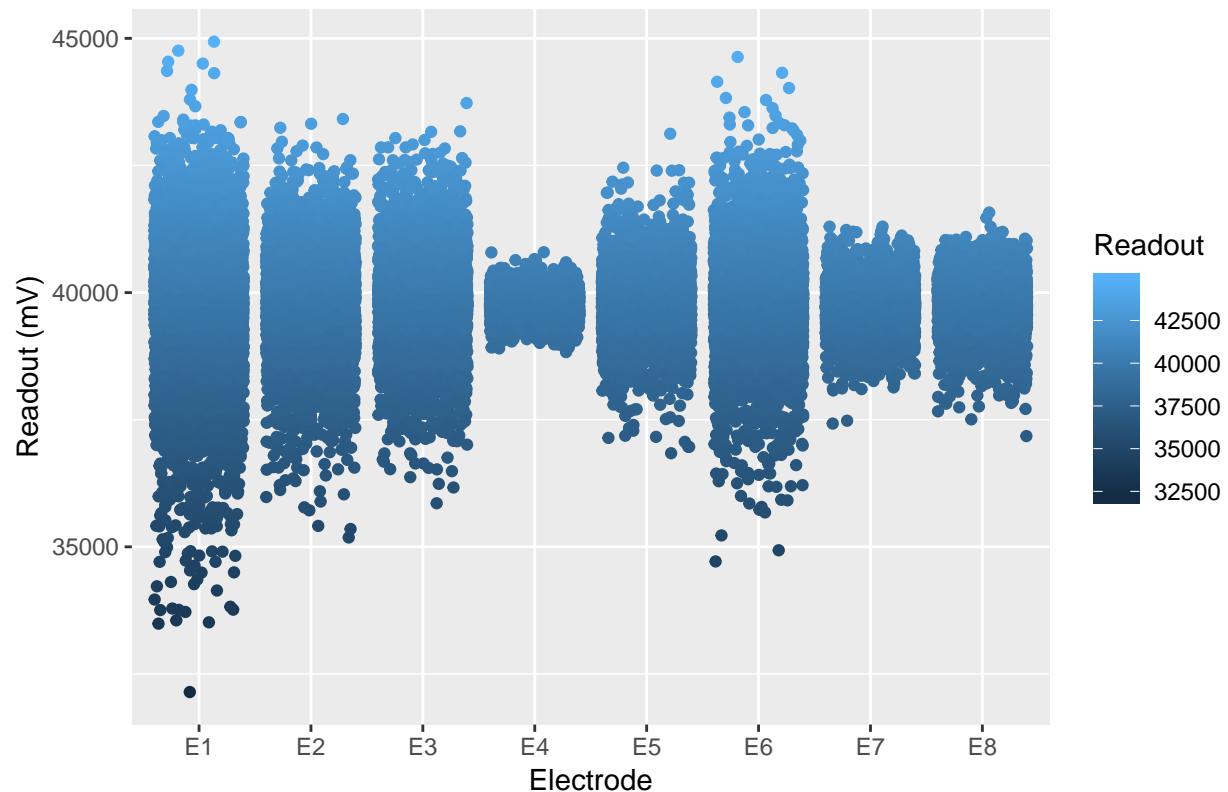
```
##  
## [[7]]
```

## Signal Intensities for Power Grip



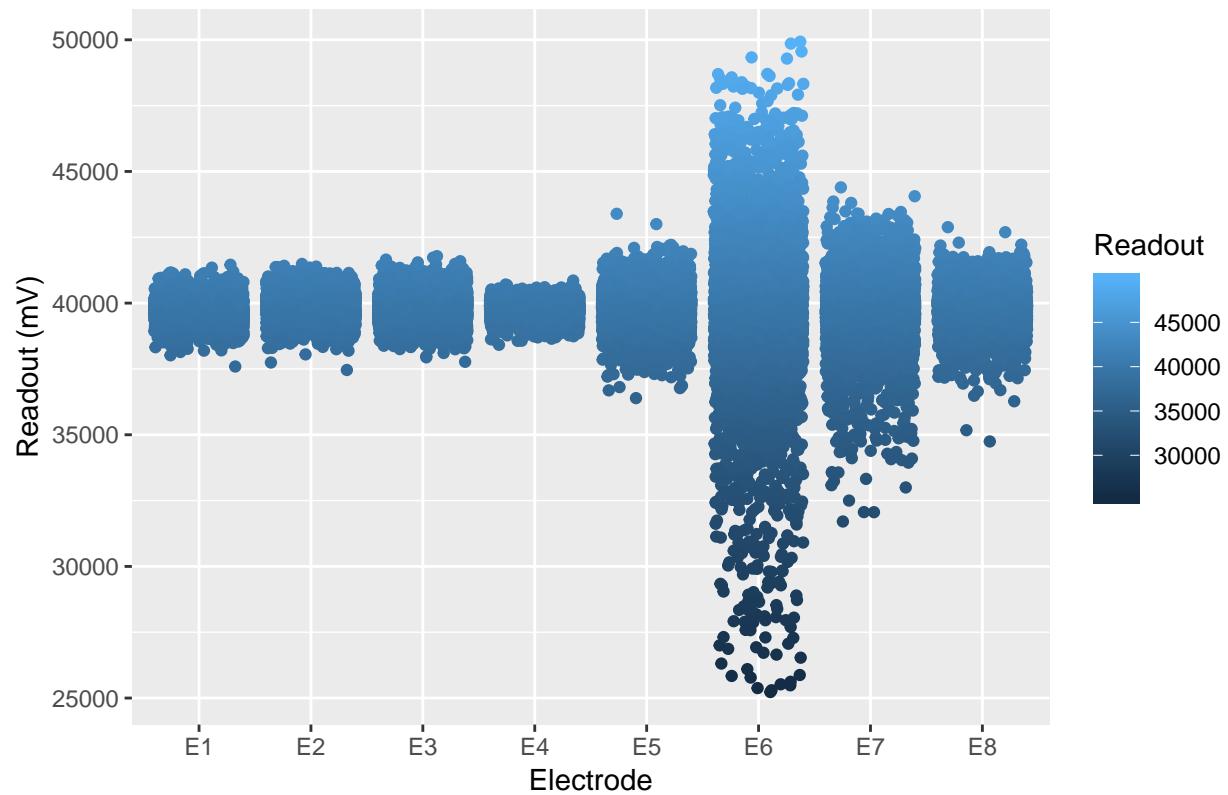
```
##  
## [[8]]
```

### Signal Intensities for Thumb Enclosed



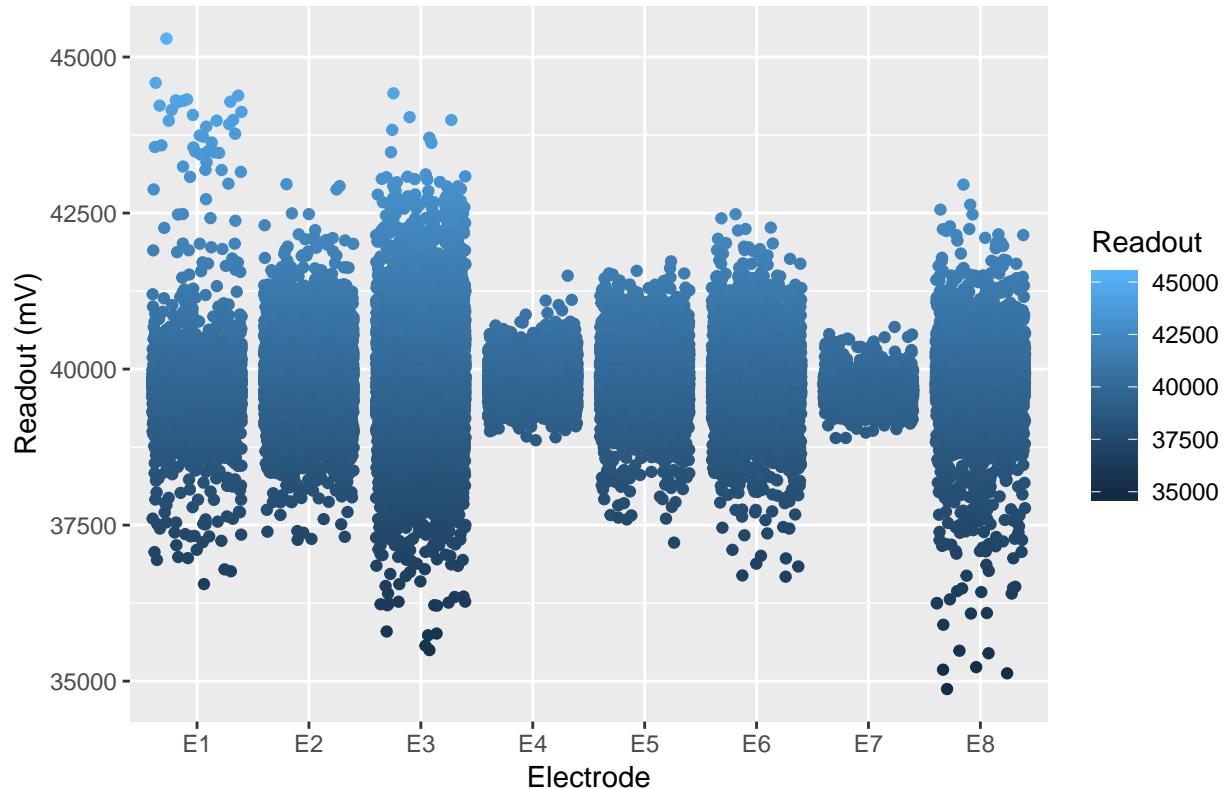
```
##  
## [[9]]
```

### Signal Intensities for Tool Grip



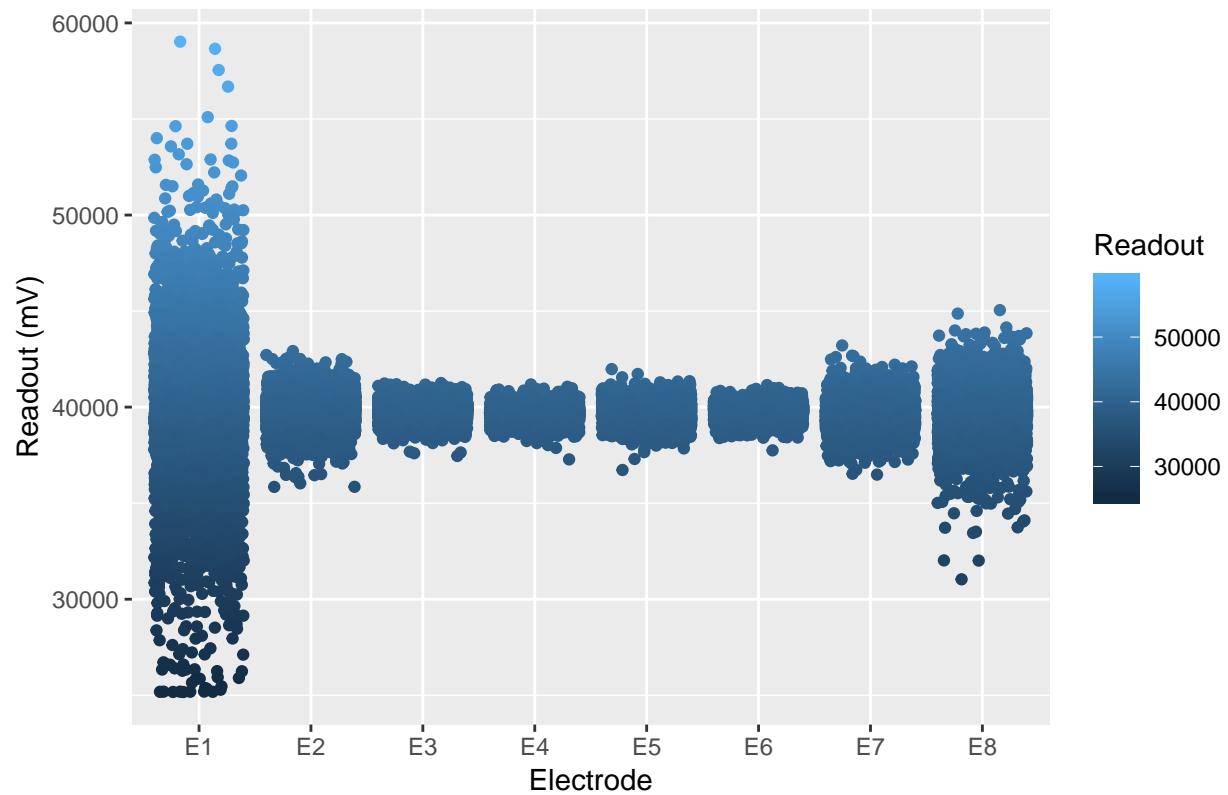
```
##  
## [[10]]
```

### Signal Intensities for W. Abduction



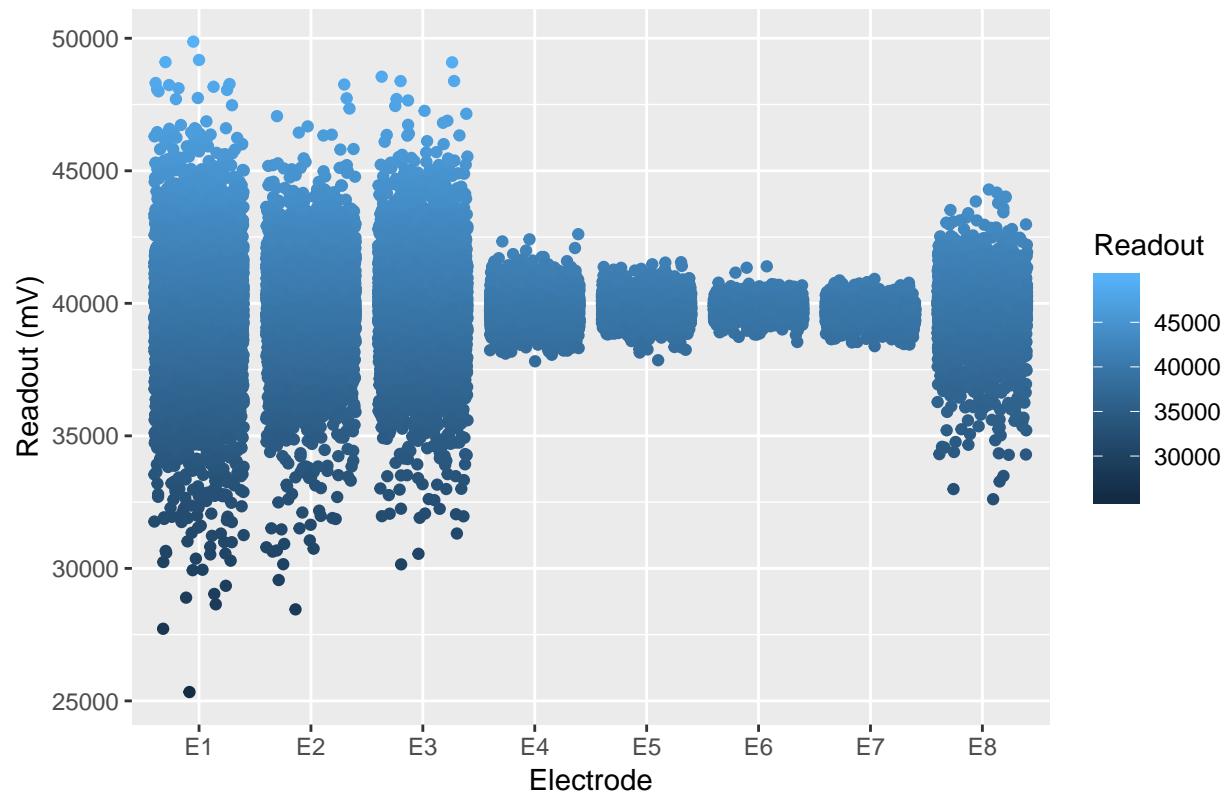
```
##  
## [[11]]
```

### Signal Intensities for W. Adduction



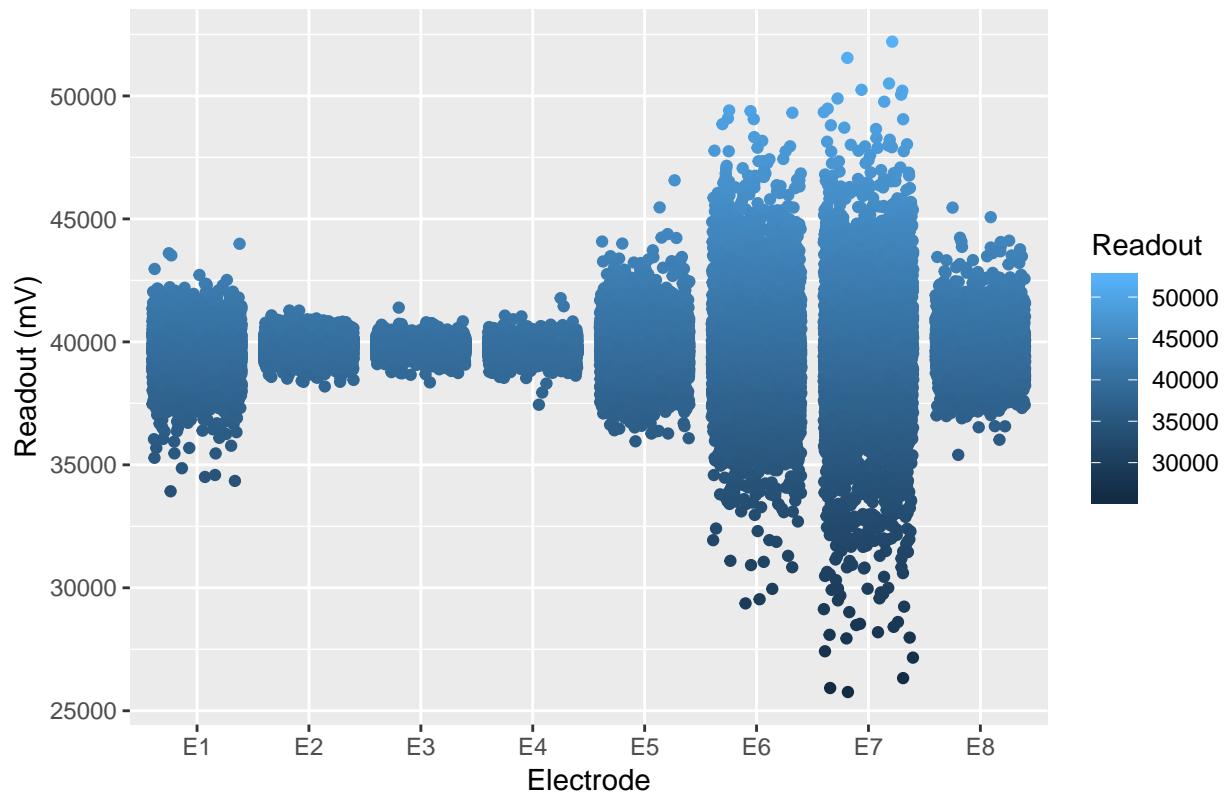
```
##  
## [[12]]
```

### Signal Intensities for W. Extension



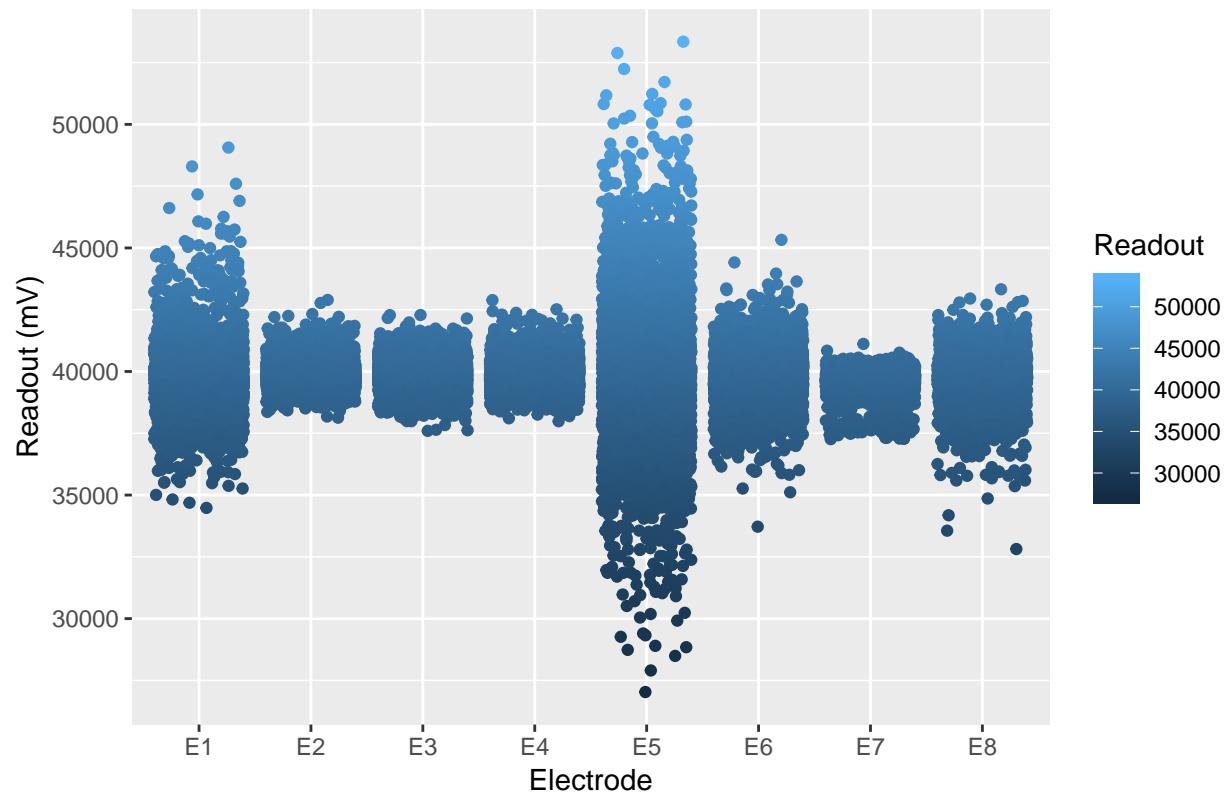
```
##  
## [[13]]
```

### Signal Intensities for W. Flexion



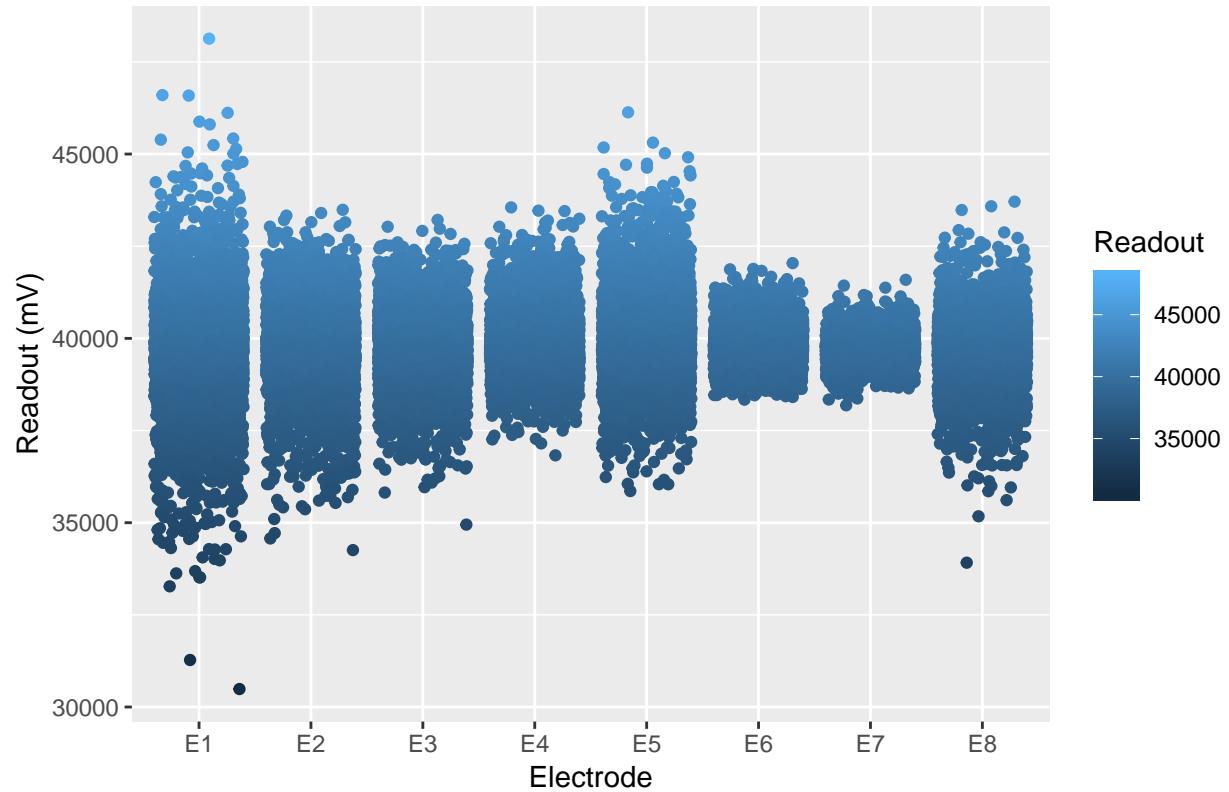
```
##  
## [[14]]
```

### Signal Intensities for W. Pronation



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

## Signal Intensities for W. Supination



## Machine Learning

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