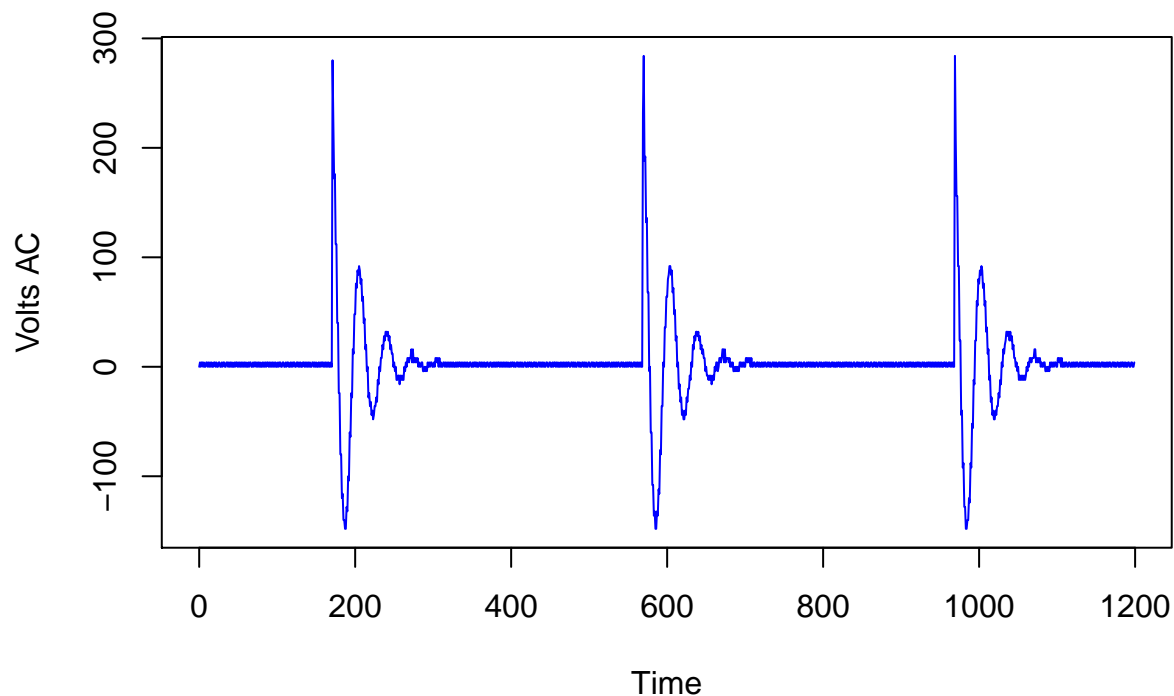


TENS data with R, Markdown and LaTeX... [Focus Numeric](#)

Note: You call LaTeX by using `'(stuff)'` (inline) and double dollar signs for equation blocks.

```
# Pull OscData from the dataset
OscData <- read.csv("OscOutput7.csv")
# Create a plotting vector -- find the length first, then create a plotting sequence
# "plotVec". I want to start the vector at 0 so we have to adjust the length appropriately.
dataLength <- length(OscData$VAC.Out)
plotVec <- 0:(dataLength-1)
# and plot with:
# plot(plotVec,OscData$VAC.Out, type="l", col="blue", main="Raw (unscaled) OscData
# from TENS unit", xlab="Time", ylab="Volts AC")
```

Raw (unscaled) OscData from TENS unit



We scale the output. The maximum output read from the TENS was 3.457 VAC (using my Fluke DMM). However, the maximum value of the data is 284. We can scale the output by $k = \frac{3.457}{284}$, and setup a new set in R:

```
k <- 3.457 / 284
scaledData <- k*OscData$VAC.Out
```

We can also scale the plotting vector so that it accurately represents time. The TENS unit puts out pulses at about 50 Hertz so we want the pulses to happen about 50 times per second. But where does it happen? A useful built-in R function is `'which()'`. Use it to find the first index where the data is original data is over 275.

```
which(OscData$VAC.Out>275)
```

```
[1] 172 571 970
```

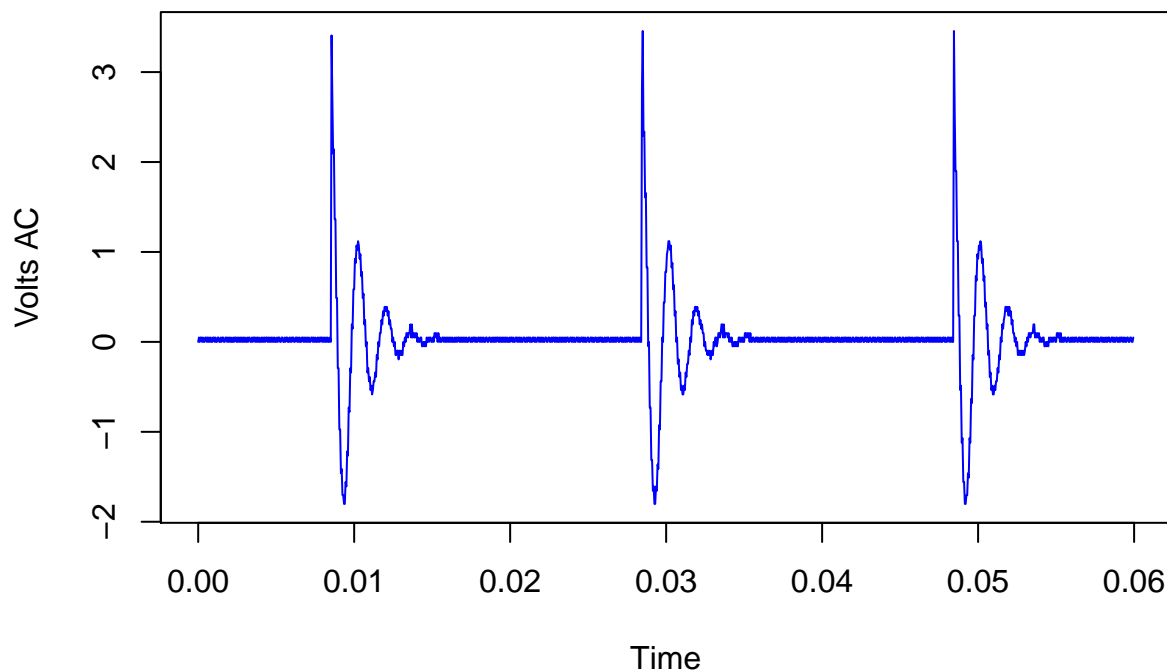
From this we can see that 1/50th of a second happens in about 400 units of our plotting vector plotVec. Thus we can scale the plotting vector using a little stoichiometry:

$$\begin{aligned}400 \text{ units} &= \frac{1}{50} \text{ second} \\20,000 \text{ units} &= 1 \text{ second} \\ \frac{1}{20,000} \text{ second} &= 1 \text{ unit}\end{aligned}$$

```
timeVec <- 1/20000 * plotVec
```

which gives us fully scaled data. Here's an updated plot:

Fully scaled TENS data



Cool. Looks good.

We will use 'approx()' to find a function that approximates the data. Remember: I'm just trying to provide an interesting Markdown example!

```
which(scaledData==0)
```

The output from that command is hidden since it returns many numbers. Based on the results I am going to pick the interval 'scaledData[171:350]'. Here is a plot of the data with 'approx(pulseSubset)':

Fully scaled TENS data – Pulse Subset

