

# GSS Poster Presentation

*Kathleen E. Wendt, Siqi Zhang, & Mallory J. Feldman*

*Draft by 11/04/2019*

## Introduction - @iqis - can condense introductory sections into poster space by taking key concepts and key words

### The Problem

- Contemporary perspectives on the embodied mind have increased interest in psychophysiological processes and their measurement.
- While there has been significant theoretical and technological innovation within this domain, the field still lacks sophisticated tools for managing the quantity and quality of data collected with physiological measurements.
- We begin to address this need, by introducing a novel pipeline for wrangling, visualizing, and analyzing certain types of physiological data.
- R has robust existing infrastructure for open source software development (see *Advanced R* and *R Packages* books).
- This pipeline is implemented in R and capitalizes on larger discourses surrounding open science practices, computational reproducibility, and open source software development.
- Collecting and cleaning physiological data requires multiple software which are versioned and expensive.
- Using these software, the typical psychophysiology study produces several output files per-subject  
\*\*[SEE DIAGRAM]
- These output files vary in their formatting and require extensive data wrangling (e.g., cleaning, exclusions, compilation, alignment) \*\*[NOT SURE THESE EXAMPLES ARE BEST]
- When conducted via copy-and-paste (e.g., Excel and SPSS) these processes introduce several researchers' degrees of freedom and opportunities for error
- **Error-ridden data produces error-driven inferences** [main point!]
- Because there is no standardized pipeline for data wrangling, analysis, and visualization, data is difficult to share (both internally and externally) and results are difficult to reproduce.

### Rise of computational reproducibility (see lit folder for references)

- R ecosystem
- Rise of data science, open science, and open source software development (e.g., Unconferences to promote open source software development and interdisciplinary collaboration)
- Tidyverse principles (see lit folder for Wickham papers)

### The Solution

- Our answer: **psyphr** - R package suite (currently supporting recent versions of MindWare Technologies workbooks)
- This poster to illustrate purpose and functionality of first R package in suite: **psyphr.read** : wrangle and tidy psychophysiological data from proprietary data collection systems

```
getwd()
```

```
## [1] "C:/Users/Siqi/Desktop/rplay/psyphr-dev/writing/gss_poster/R"
```

## Load Packages

```
# devtools::install_github("psyphr-dev/psyphr.read")
library(psyphr.read)
library(tidyverse)
library(readxl)
```

## Example data

Would it be better to take snapshots of the file directory structure and even the Excel workbook? I am trying to think about the best way to show that these data are disorganized, messy, and in need of **psyphr** tools. I don't think it would be particularly useful to include all the code required to "wrangle" one Excel workbook.

```
list.files("gss_poster/data")
```

```
## character(0)
```

```
readxl::read_xlsx("../data/Pilot1/Pilot_Sub1_ECG_Baseline.xlsx", sheet = 1) %>%
  tibble::glimpse()
```

```
## Observations: 23
```

```
## Variables: 11
```

```
## $ `Segment Number` <chr> "Start Event", "Start Time", "End Event", "En...
## $ `1` <chr> "M_Test_0_0.mwi:Acquisition PC:Keyboard:F1:Ba...
## $ `2` <chr> "M_Test_0_0.mwi:Acquisition PC:Keyboard:F1:Ba...
## $ `3` <chr> "M_Test_0_0.mwi:Acquisition PC:Keyboard:F1:Ba...
## $ `4` <chr> "M_Test_0_0.mwi:Acquisition PC:Keyboard:F1:Ba...
## $ `5` <chr> "M_Test_0_0.mwi:Acquisition PC:Keyboard:F1:Ba...
## $ `6` <chr> "M_Test_0_0.mwi:Acquisition PC:Keyboard:F1:Ba...
## $ `7` <chr> "M_Test_0_0.mwi:Acquisition PC:Keyboard:F1:Ba...
## $ `8` <chr> "M_Test_0_0.mwi:Acquisition PC:Keyboard:F1:Ba...
## $ `9` <chr> "M_Test_0_0.mwi:Acquisition PC:Keyboard:F1:Ba...
## $ `10` <chr> "M_Test_0_0.mwi:Acquisition PC:Keyboard:F1:Ba...
```

Create lists of file names - @iqis, what do you think is the best way to manage the multiple person/task/stream problem? I don't think `psyphr.read::MW()` is best used on a folder split by person and task. It would be best for the user to be able to map all of the files in a study at once and have internal labels for person, task, and data stream type that can be filtered.

@wendtke This problem should be covered. If the user needs to read a bunch of files in different folders, he is encouraged to structure them as flat, with `psyphr.read::flatten_study_dir()`, then and `psyphr.read::MW_study()` can work.

```
pilot1_path <- "../data/Pilot1"
list.files(pilot1_path, ".xlsx")
```

```
## [1] "Pilot_Sub1_BP_Baseline.xlsx" "Pilot_Sub1_ECG_Baseline.xlsx"
## [3] "Pilot_Sub1_ECG_RLS.xlsx"    "Pilot_Sub1_EDA_Baseline.xlsx"
```

```
## [5] "Pilot_Sub1_EDA_RLS.xlsx"      "Pilot_Sub1_ICG_Baseline.xlsx"
## [7] "Pilot_Sub1_ICG_RLS.xlsx"

pilot2_path <- "../data/Pilot2"
list.files(pilot2_path, ".xlsx")
```

```
## [1] "Pilot_Sub2_BP_Baseline.xlsx"  "Pilot_Sub2_BP_RLS.xlsx"
## [3] "Pilot_Sub2_ECG_Baseline.xlsx" "Pilot_Sub2_ECG_RLS.xlsx"
## [5] "Pilot_Sub2_EDA_Baseline.xlsx" "Pilot_Sub2_EDA_RLS.xlsx"
## [7] "Pilot_Sub2_ICG_Baseline.xlsx" "Pilot_Sub2_ICG_RLS.xlsx"
```

Use `psyphr.read::MW_study()` to read data

```
pilot1 <- psyphr.read::MW_study(pilot1_path)
pilot1_unnested <- pilot1 %>% psyphr.read::unnest_data()
```

`View()` both

## Visualize

Once I learn how to properly use `psyphr.read`, we can move on to plotting example data. I think we should have two time series plots. One for Respiratory Sinus Arrhythmia (under HRV Stats) and one for Tonic Skin Conductance Level (under EDA Stats). If we are ambitious, we can include data from both baseline and RLS task.