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 (CITE).
- Despite significant innovation (CITE), this burgeoning field lacks sophisticated tools for managing the quantity and quality of psychophysiological data.
- Currently, most data cleaning and preprocessing requires proprietary software. These software produce several output files per-subject. These files often vary in their formatting and require extensive data wrangling.
- When conducted by hand, wrangling introduces opportunities for bias and error (CITE).
- Error-ridden data produces error-driven inferences.
- Unstandardized formatting obstructs shareability and reproducibility.

The Solution

- R provides an existing infrastructure for open-source software development.
- We introduce a new R package suite, `psyphr`, capitalizing on larger discourses re: open science and computational reproducibility
- [Point about computational reproducibility - cite- and tidyverse principles -cite-] Laraway, Bizzego, NatlAca, Piccolo, Pelicans, Stodden, Wickham
- This poster illustrates the purpose and functionality of our first R package within the `psyphr` suite.
- `psyphr.read` is designed to help researchers wrangle and tidy psychophysiological data from proprietary data collection systems.

Load Packages

```
# devtools::install_github("psyphr-dev/psyphr.read")
library(psyphr.read)
library(tidyverse)
library(tibble)
library(readxl)
library(ggplot2)
```

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.

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

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

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

```r
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"
```

```r
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

```r
pilot1 <- psyphr.read::MW_study(pilot1_path)
```

```r
pilot2 <- psyphr.read::MW_study(pilot2_path)
```

Use **psyphr.read::unnest_data()** to unnest study

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

```
pilot2_unnested <- pilot2 %>% psyphr.read::unnest_data()
```

HRV: Respiratory sinus arrhythmia

Pilot 1 data preparation

```
hrv1_baseline <- pilot1_unnested %>%
  filter(format == "HRV") %>%
  select(id_1:id_4, `HRV Stats`) %>%
  filter(id_4 == "Baseline") %>%
  unnest() %>%
  filter(`Segment Duration` == "30") %>%
  select(id_1:id_4, `Segment Number`, RSA) %>%
  mutate(rsa_base = mean(RSA)) # calculate average baseline RSA

hrv1_rls <- pilot1_unnested %>%
  filter(format == "HRV") %>%
  select(id_1:id_4, `HRV Stats`) %>%
  filter(id_4 == "RLS") %>%
  unnest() %>%
  filter(`Segment Duration` == "30") %>%
  select(id_1:id_4, `Segment Number`, RSA) %>%
  mutate(rsa_react_rls = RSA - hrv1_baseline$rsa_base) # calculate reactivity
```

Pilot 2 data preparation

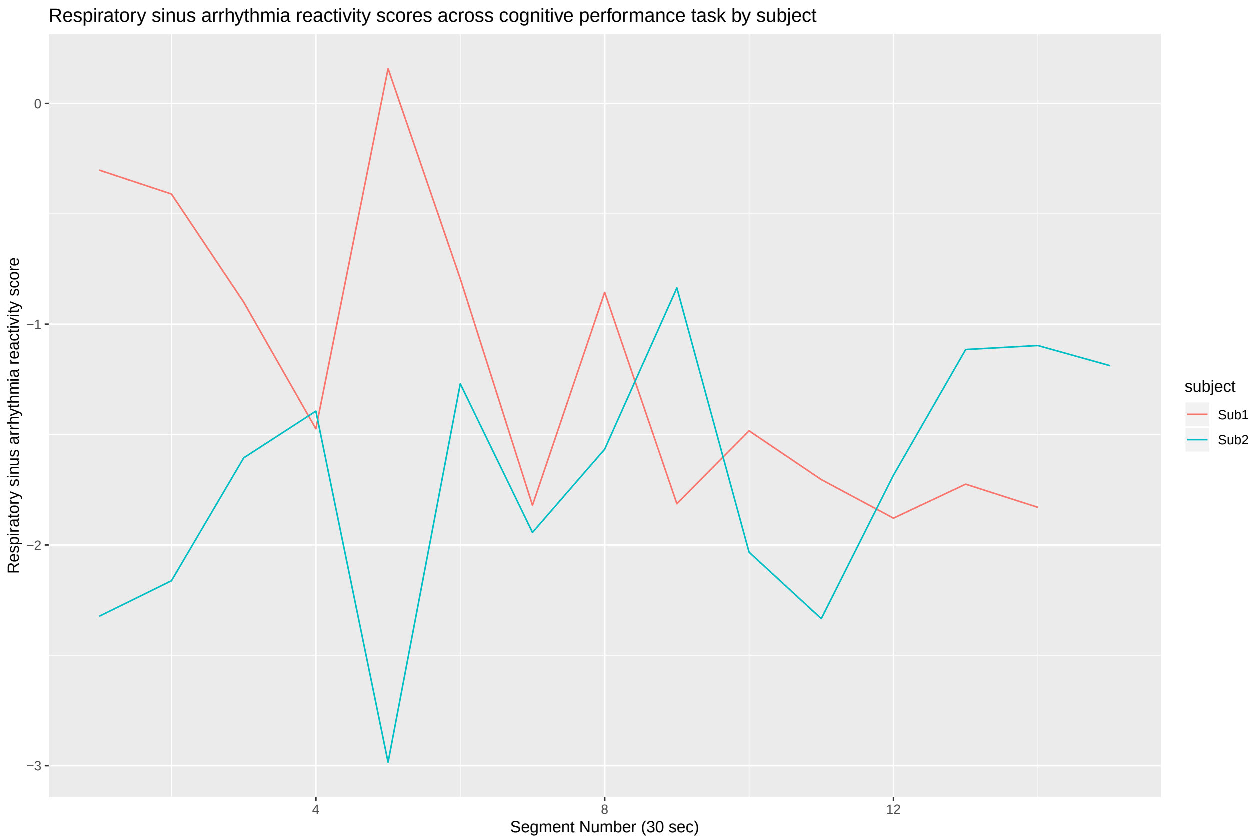
```
hrv2_baseline <- pilot2_unnested %>%
  filter(format == "HRV") %>%
  select(id_1:id_4, `HRV Stats`) %>%
  filter(id_4 == "Baseline") %>%
  unnest() %>%
  filter(`Segment Duration` == "30") %>%
  select(id_1:id_4, `Segment Number`, RSA) %>%
  mutate(rsa_base = mean(RSA)) # calculate average baseline RSA

hrv2_rls <- pilot2_unnested %>%
  filter(format == "HRV") %>%
  select(id_1:id_4, `HRV Stats`) %>%
  filter(id_4 == "RLS") %>%
  unnest() %>%
  filter(`Segment Duration` == "30") %>%
  select(id_1:id_4, `Segment Number`, RSA) %>%
  mutate(rsa_react_rls = RSA - hrv1_baseline$rsa_base) # calculate reactivity

hrv_rls_data <- full_join(hrv1_rls, hrv2_rls)
```

Plot RSA reactivity scores across cognitive performance task

```
hrv_rls_data %>%
  rename("study" = id_1,
         "subject" = id_2,
         "stream" = id_3,
         "task" = id_4) %>%
  group_by(subject) %>%
  ggplot(aes(x = `Segment Number`, y = rsa_react_rls, color = subject)) +
  geom_line() +
  ylab("Respiratory sinus arrhythmia reactivity score") +
  xlab("Segment Number (30 sec)") +
  ggtitle("Respiratory sinus arrhythmia reactivity scores across cognitive performance task by subject")
```



IMP: Pre-ejection period

Pilot 1 data preparation

```
imp1_baseline <- pilot1_unnested %>%
  filter(format == "IMP") %>%
  select(id_1:id_4, `Impedance Stats`) %>%
  filter(id_4 == "Baseline") %>%
  unnest() %>%
  filter(`Segment Duration` == "30") %>%
  select(id_1:id_4, `Segment Number`, PEP) %>%
```

```r
  mutate(pep_base = mean(PEP)) # calculate average baseline PEP

imp1_rls <- pilot1_unnested %>%
  filter(format == "IMP") %>%
  select(id_1:id_4, `Impedance Stats`) %>%
  filter(id_4 == "RLS") %>%
  unnest() %>%
  filter(`Segment Duration` == "30") %>%
  select(id_1:id_4, `Segment Number`, PEP) %>%
  mutate(pep_react_rls = PEP - imp1_baseline$pep_base) # calculate reactivity
```

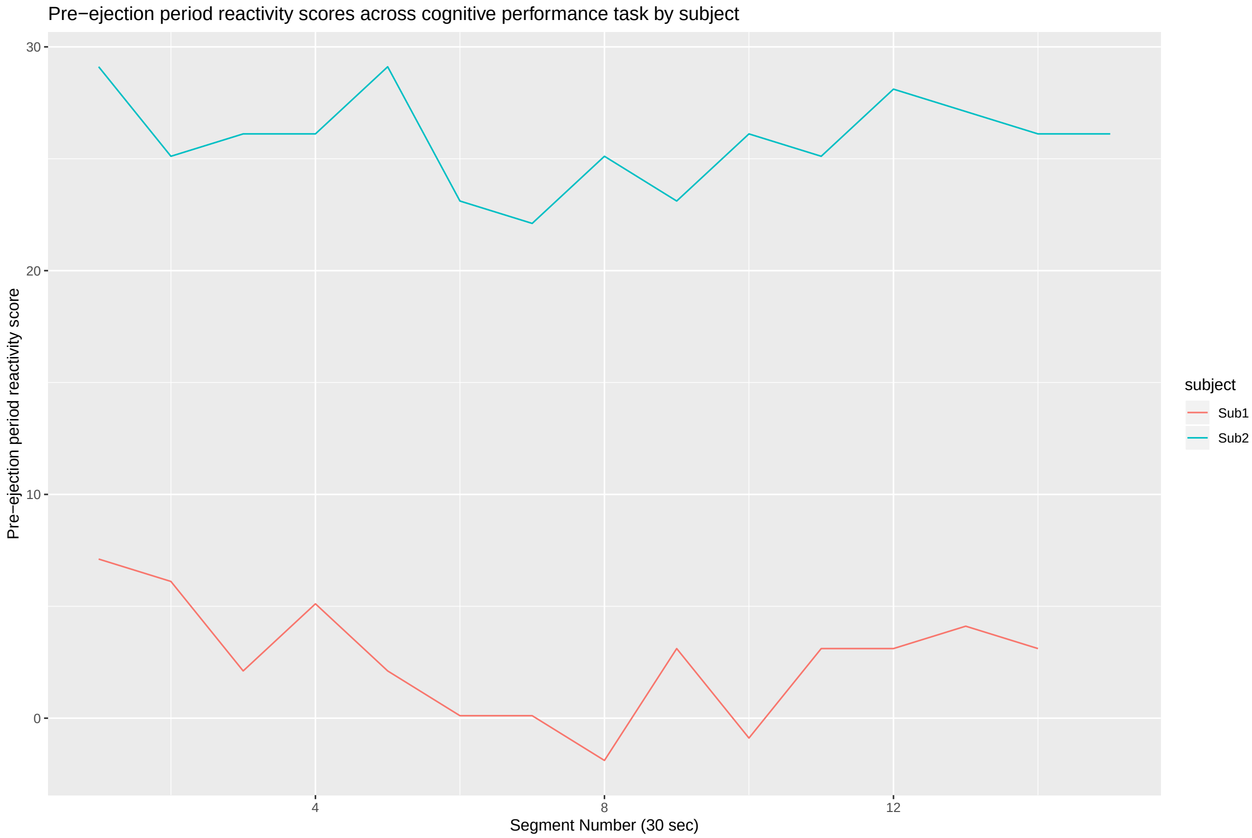
Pilot 2 data preparation

```r
imp2_baseline <- pilot2_unnested %>%
  filter(format == "IMP") %>%
  select(id_1:id_4, `Impedance Stats`) %>%
  filter(id_4 == "Baseline") %>%
  unnest() %>%
  filter(`Segment Duration` == "30") %>%
  select(id_1:id_4, `Segment Number`, PEP) %>%
  mutate(pep_base = mean(PEP)) # calculate average baseline PEP

imp2_rls <- pilot2_unnested %>%
  filter(format == "IMP") %>%
  select(id_1:id_4, `Impedance Stats`) %>%
  filter(id_4 == "RLS") %>%
  unnest() %>%
  filter(`Segment Duration` == "30") %>%
  select(id_1:id_4, `Segment Number`, PEP) %>%
  mutate(pep_react_rls = PEP - imp1_baseline$pep_base) # calculate reactivity
```

```r
imp_rls_data <- full_join(imp1_rls, imp2_rls)
```

```r
imp_rls_data %>%
  rename("study" = id_1,
         "subject" = id_2,
         "stream" = id_3,
         "task" = id_4) %>%
  group_by(subject) %>%
  ggplot(aes(x = `Segment Number`, y = pep_react_rls, color = subject)) +
  geom_line() +
  ylab("Pre-ejection period reactivity score") +
  xlab("Segment Number (30 sec)") +
  ggtitle("Pre-ejection period reactivity scores across cognitive performance task by subject")
```


Pre−ejection period reactivity scores across cognitive performance task by subject
Pre−ejection period reactivity score
30
20
10
0
4
8
12
Segment Number (30 sec)
subject
Sub1
Sub2