Code ▾

Supplementary file 1

Sympathetic nervous system functioning during the Face-to-face Still-face paradigm in the first year of life

prepare data for modelling

Hide

```
# make list of dead channels
dead_channels <- output %>%
  mutate(phase = fct_relevel(phase,"play","sf","reunion")) %>%
  group_by(participant) %>%
  summarise(avg_conduc = mean(conduc)) %>%
  filter(avg_conduc < 5 | avg_conduc > 120) %>%
  distinct(participant)

# exclude dead channels and collapse accelerometer columns
output_filtered <- output %>%
  group_by(participant, device, phase) %>%
  anti_join(dead_channels, by = "participant") %>%
  mutate(accel_xyz = abs(accel_x) + abs(accel_y) + abs(accel_z))

### randomise devices ------------------------------------------------------------------------------

library(permute)
# randomly assign participants by device
rand <- output_clean %>%
  ungroup() %>%
  select(participant, device) %>%
  group_by(participant, device) %>%
  unique()

rand %<>%
  group_by(participant) %>%
  mutate(assign = sample(c(0,1), size = 1, replace = TRUE))

rand <- rand[shuffle(rand),]
rand$dup <- duplicated(rand[,c(1,3)])
rand <- rand[-which(rand$dup),]
rand %<>% select(1:2) %>%
  arrange(participant)
```

UPDATED 28-10-20 inverting time-series by phase and centering final sample on 0

Hide

```
glimpse(sf_phase,width=100)
```

```
Rows: 909
Columns: 3
Groups: participant, phase [909]
$ participant <fct> 3115, 3115, 3115, 3125, 3125, 3125, 3128, 3128, 3128, 3134, 3134, 3134, 3143,…
$ phase       <fct> play, sf, reunion, play, sf, reunion, play, sf, reunion, play, sf, reunion, p…
$ data        <list> [<tbl_df[1870 x 11]>, <tbl_df[3456 x 11]>, <tbl_df[892 x 11]>, <tbl_df[2400 …
```

preliminary modelling for each participant by phase with resistance and accel_xyz as covariates

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0.503274	0.5024754	5.418775	630.1994	7.340882e-283	3	-5811.469	11632.94	11660.61	54791.59

1 row | 1-10 of 12 columns

extract the model for each participant by phase

Hide

```
glimpse(sf_px,width=100)
```

```
Rows: 909
Columns: 8
Groups: participant, phase [909]
$ id          <fct> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22…
$ participant <fct> 3115, 3115, 3115, 3125, 3125, 3125, 3128, 3128, 3128, 3134, 3134, 3134, 3143,…
$ phase       <fct> play, sf, reunion, play, sf, reunion, play, sf, reunion, play, sf, reunion, p…
$ rsq         <dbl> 0.50327402, 0.64524279, 0.89820582, 0.92333142, 0.97245447, 0.98188630, 0.978…
$ intercept   <dbl> 31.6192225, 31.1828560, 25.0140960, 59.3018506, 63.6439644, 67.1201623, 39.16…
$ slope       <dbl> 1.192065e-02, 6.007435e-04, 1.273463e-02, 1.735436e-02, -1.640232e-02, 7.6365…
$ resist      <dbl> -1.333648e-04, -2.226587e-01, -1.375338e-01, -4.653077e-01, -7.434886e-01, -1…
$ accel_xyz   <dbl> -1.5348880199, 0.0046802277, 0.0197610030, -0.5148904233, -0.3642863326, -0.0…
```

examine assumptions prior to mixed-effects modelling

Hide

```
# check for non-finite values
any(sapply(sf_px,is.infinite))
```

```
[1] FALSE
```

Hide

```r
# check for NA values
any(sapply(sf_px,is.na))
```

```
[1] FALSE
```

Hide

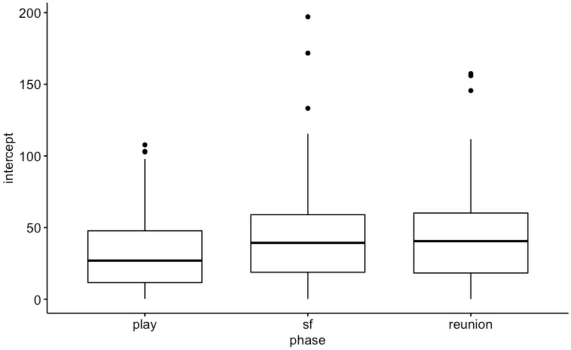
```r
# compute summary statistics
sf_px %>%
  ungroup() %>% group_by(phase) %>%
  rstatix::get_summary_stats(c(intercept,slope), type = "mean_sd")
```

phase <fctr>	variable <chr>	n <dbl>	mean <dbl>	sd <dbl>
play	intercept	303	32.338	27.193
play	slope	303	0.004	0.019
sf	intercept	303	42.095	31.441
sf	slope	303	0.015	0.054
reunion	intercept	303	40.629	29.990
reunion	slope	303	-0.012	0.067

6 rows

Hide

```r
# boxplot visualisation of summary
sf_px %>%
  filter(intercept > 0, intercept < 200) %>%
  ggpubr::ggboxplot(x = "phase", y = "intercept")
```



Hide

```r
# identify outliers
sf_px %>%
  ungroup() %>% group_by(phase) %>%
  rstatix::identify_outliers(intercept)
```

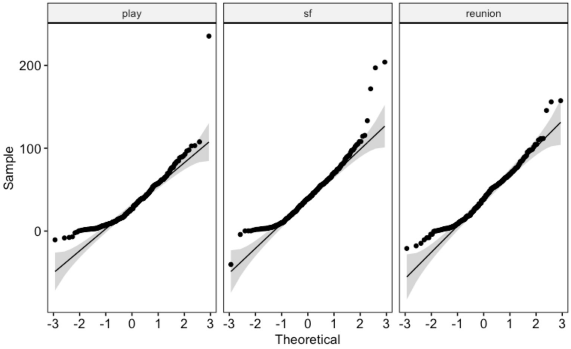
phase <fctr>	id <fctr>	participant <fctr>	rsq <dbl>	intercept <dbl>	slope <dbl>	resist <dbl>	accel_xyz <dbl>	is.outlier <lgl>
play	103	3225	0.9962092	102.7755	9.427977e-05	-2.6152434294	-0.01872356	TRUE
play	460	3496	0.9373980	235.3433	1.323292e-01	-0.0006758406	-8.81168381	TRUE
play	859	3857	0.9883752	103.1334	-3.787348e-03	-0.0054840828	-2.83824511	TRUE
play	871	3865	0.9574710	107.6952	1.693493e-02	-1.2084946894	-1.29270925	TRUE
sf	254	3360	0.9537277	133.2100	3.639317e-01	-0.5497827834	-4.17090982	TRUE
sf	461	3496	0.9848103	197.0903	2.037723e-02	-0.0004063428	-7.06205389	TRUE
sf	839	3838	0.9726583	203.9941	-1.033213e-01	0.0169899770	-7.13139977	TRUE
sf	872	3865	0.9795633	171.7328	3.844066e-02	-0.5815441343	-4.68577077	TRUE
reunion	255	3360	0.9604438	145.5756	-3.038226e-02	-1.7831424177	-4.15362720	TRUE
reunion	462	3496	0.9885657	155.9504	1.805155e-02	-0.0001426843	-5.40428194	TRUE

1-10 of 11 rows | 1-9 of 10 columns

Previous 1 2 Next

Hide

```r
# visually check for outliers
ggqqplot(sf_px, "intercept", facet.by = "phase")
```



Hide

```r
### Normality -----------------------------------------------------------------

# Kolgomorov-Smirnov less sensitive to large sample size
ks_fx <- function (df) {
  temp <- df[[1]]
  data <- ks.test(x = temp, y = 'pnorm', alternative = 'two.sided')
  return(data)
}

sf_residuals <- phase_models %>%
  mutate(
    res = mod %>% map(stats::residuals),
    norm = res %>% map(ks_fx),
    ks = norm %>% map_dbl("p.value")
  )
```

implement peak and artefact detection following Taylor et al., 2017

Hide

```r
artefact_filepaths <- list.files("output/artefact-detection",pattern="*.csv",full.names=TRUE) %>%
  as_tibble() %>%
  rename(filepath_in = "value") %>%
  mutate(channels = substring(filepath_in,27,35))

peak_filepaths <- list.files("output/peak-detection",pattern="*.csv",full.names=TRUE) %>%
  as_tibble() %>%
  rename(filepath_in = "value") %>%
  mutate(channels = substring(filepath_in,23,31))

# prepare output dataframe
combined <- peak_filepaths[[2]] %>%
  as_tibble() %>%
  rename(channels = value)
```

timestamp peaks

Hide

```r
# helper function to produce epochs in peak data
epochr <- function(value,base) {
  base*floor(value/base)
}

phase_ts <- output_prep %>%
  select(1:5) %>%
  group_by(participant,device,phase) %>%
  filter(phase_t == 0) %>%
  mutate(
    timestamp = as.numeric(as.POSIXct(timestamp)),
    epoch_start = epochr(timestamp,5),
    channels = paste0(participant,"_",device)
    ) %>%
  select(1:2,4:3,6:7) %>%
  nest(phases = c(1:5)) %>%
  semi_join(combined,by="channels") %>%
  left_join(artefact_filepaths,by="channels") %>%
  left_join(peak_filepaths,by="channels") %>%
  mutate(analyses = NA)

for (i in seq_along(phase_ts[[1]])) {
  library(lubridate)
  artefacts <- read_csv(file.path(phase_ts[[3]][i])) %>%
    mutate(epoch_start = epochr(time_length(interval(ymd("1970-01-01"),StartTime)),5),
```

```
           epoch_end = epochr(time_length(interval(ymd("1970-01-01"),EndTime)),5)) %>%
    select(1,6,4:5)

  peaks <- read_csv(file.path(phase_ts[[4]][[i]])) %>%
    mutate(timestamp = as.numeric(time_length(interval(ymd("1970-01-01"),Timestamp)))) %>%
    filter(EDA < 50) %>%
    mutate(epoch_start = epochr(timestamp,5)) %>%
    select(10:9,2:8)

  phase_ts[[5]][[i]] <- full_join(artefacts,peaks,by="epoch_start") %>%
    arrange(epoch_start) %>%
    as_tibble() %>%
    nest(analyses = everything())
}
```

create epoched database

Hide

```
phase_ts <- phase_ts %>%
  unnest(c(analyses)) %>%
  mutate(master = NA)

for (i in seq_along(phase_ts[[1]])) {

  temp_a <- phase_ts[[1]][i]

  temp_b <- phase_ts %>%
    filter(channels == temp_a) %>%
    unnest(phases) %>%
    select(1:6)

  # adding phase label to constrain end of experiment appropriately
  temp_d <- temp_b[3,] %>%
    mutate(phase = "terminated",
           timestamp = timestamp+240,
           epoch_start = epochr(timestamp,5))
  temp_e <- rbind(temp_b,temp_d)

  temp_c <- phase_ts %>%
  filter(channels == temp_a) %>%
  unnest(analyses) %>%
  select(1,5:16)

  phase_ts[[6]][[i]] <- full_join(temp_e,temp_c,by="epoch_start") %>%
    arrange(epoch_start) %>%
    fill(1:4) %>%
    select(2:4,8,6,9:18) %>%
    rename(peak = timestamp.y) %>%
    nest(clean = everything())
}

phase_ts <- phase_ts %>%
  unnest(c(master))
```

explore peak data

Hide

```
# how many peak events have been detected per phase by participant by device?
phase_ts %>%
  select(clean) %>%
  unnest(clean) %>%
  group_by(participant,device,phase) %>%
  filter(phase!="terminated",!is.na(EDA),!is.na(participant)) %>%
  rstatix::reorder_levels("phase", order = c("play","sf","reunion")) %>%
  count(phase)
```

participant <fctr>	**device** <fctr>	**phase** <fctr>	**n** <int>
3115	CD1C	play	54
3115	CD1C	sf	23
3115	CD2B	play	43
3115	CD2B	sf	10
3115	CD2B	reunion	11
3125	CD1C	play	8
3125	CD1C	sf	9
3125	CD1C	reunion	3
3125	CD2B	play	5
3125	CD2B	sf	11

1-10 of 1,188 rows Previous **1** 2 3 4 5 6 ... 100 Next

Hide

```
# what is the average number of peak events by phase?
phase_ts %>%
  select(clean) %>%
  unnest(clean) %>%
  group_by(participant,device,phase) %>%
  filter(phase!="terminated",!is.na(EDA),!is.na(participant)) %>%
  count(phase) %>%
  ungroup() %>%
  group_by(phase) %>%
  rstatix::reorder_levels("phase", order = c("play","sf","reunion")) %>%
  summarise(by_phase = mean(n))
```

phase <fctr>	by_phase <dbl>
play	8.088889
sf	7.498734
reunion	10.855670

3 rows

Hide

```
# what is the proportion of peak events by phase?
phase_ts %>%
  select(clean) %>%
  unnest(clean) %>%
  group_by(participant,device,phase) %>%
  filter(phase!="terminated",!is.na(EDA),!is.na(participant)) %>%
  rstatix::reorder_levels("phase", order = c("play","sf","reunion")) %>%
  count(phase) %>%
  mutate(N = case_when(phase == "play" | phase == "reunion" ~ n/240*100,
                       phase == "sf" ~ n/90*100)) %>%
  ungroup() %>%
  group_by(phase) %>%
  summarise(by_phase = mean(N))
```

phase <fctr>	by_phase <dbl>
play	3.370370
sf	8.331927
reunion	4.523196

3 rows

Hide

```
# which phase did children have the most artefacts in?
phase_ts %>%
  select(clean) %>%
  unnest(clean) %>%
  group_by(phase) %>%
  filter(phase != "terminated",!is.na(MulticlassLabels),!is.na(participant)) %>%
  rstatix::reorder_levels("phase", order = c("play","sf","reunion")) %>%
  select(1:3,7) %>%
  count(MulticlassLabels) %>%
  mutate(phase_total = sum(n),
         proportion = round((n/phase_total*100),2)) %>%
  select(-4)
```

phase <fctr>	MulticlassLabels <dbl>	n <int>	proportion <dbl>
play	-1	68	0.64
play	0	9522	89.75
play	1	1020	9.61
sf	-1	78	0.73
sf	0	9579	89.51
sf	1	1044	9.76
reunion	-1	252	1.43
reunion	0	14405	82.01
reunion	1	2908	16.56

9 rows

create summary statistics for peak data

Hide

```
peaks <- phase_ts %>%
  select(clean) %>%
  unnest(clean) %>%
  group_by(participant,device,phase) %>%
  filter(phase!="terminated",!is.na(EDA),!is.na(participant)) %>%
  rstatix::reorder_levels("phase", order = c("play","sf","reunion")) %>%
  count(phase) %>%
  mutate(N = case_when(phase == "play" | phase == "reunion" ~ n/240*100,
                       phase == "sf" ~ n/90*100) %>% round(.,2)) %>%
  select(-n) %>%
  ungroup()
# write_csv(peaks,file.path(getwd(),"REAL_FF-SF_peaks_29-01-21.csv"))

# do the phases differ substantially in number of peak events?
peaks %>%
  ungroup() %>% group_by(phase) %>%
  count(phase) %>%
  ungroup() %>%
  mutate(`% diff` = c(0,(405-395)/405*100,(405-388)/405*100) %>%
                       round(.,2))
```

phase	n	% diff
<fctr>	<int>	<dbl>
play	405	0.00
sf	395	2.47
reunion	388	4.20

3 rows

Hide

```
# peak events by device per phase
peaks %>%
  ungroup() %>% group_by(device) %>%
  count(phase)
```

device	phase	n
<fctr>	<fctr>	<int>
CD1C	play	207
CD1C	sf	206
CD1C	reunion	197
CD2B	play	198
CD2B	sf	189
CD2B	reunion	191

6 rows

Hide

```
#  what are the overall summary statistics by phase?
peaks %>%
  ungroup() %>% group_by(phase) %>%
  rstatix::get_summary_stats(N,type="mean_sd")
```

phase	variable	n	mean	sd
<fctr>	<chr>	<dbl>	<dbl>	<dbl>
play	N	405	3.371	2.437
sf	N	395	8.332	6.290
reunion	N	388	4.523	2.792

3 rows

Hide

```
# what are the summary statistics when disaggregating device?
peaks %>%
  ungroup() %>% group_by(device,phase) %>%
  rstatix::get_summary_stats(N,type="mean_sd")
```

device	phase	variable	n	mean	sd
<fctr>	<fctr>	<chr>	<dbl>	<dbl>	<dbl>
CD1C	play	N	207	3.380	2.527
CD1C	sf	N	206	8.975	7.612
CD1C	reunion	N	197	4.514	2.807
CD2B	play	N	198	3.361	2.345
CD2B	sf	N	189	7.630	4.335
CD2B	reunion	N	191	4.533	2.783

6 rows

Hide

```
# is there a difference in peak events between devices?
peaks %>%
  ungroup() %>% group_by(device) %>%
  rstatix::get_summary_stats(N,type="mean_sd")
```

device	variable	n	mean	sd
<fctr>	<chr>	<dbl>	<dbl>	<dbl>
CD1C	N	610	5.636	5.487
CD2B	N	578	5.144	3.714

2 rows

Hide

```
# are there any outliers? if yes, isolate extreme values
peaks %>%
  ungroup() %>% group_by(device,phase) %>%
  rstatix::identify_outliers(N) %>%
  filter(is.extreme==TRUE) %>%
  select(device,participant) %>%
  distinct()
```

device	participant
<fctr>	<fctr>
CD1C	3115
CD1C	3189
CD1C	3237
CD1C	3246
CD1C	3247
CD1C	3738
CD1C	3819
CD2B	3115
CD2B	3421
CD2B	3564

1-10 of 10 rows

2 x (Device: CD1C, CD2B) by 3 x (Phase: Play, SF, Reunion) mixed ANOVA

Hide

```
peaks_anova <- anova_test(dv=N,wid=participant,between=device,within=phase,
                          data=peaks,effect.size="pes",type=3)

# applying Hyunh-Feldt correction
get_anova_table(peaks_anova,correction="HF")
```

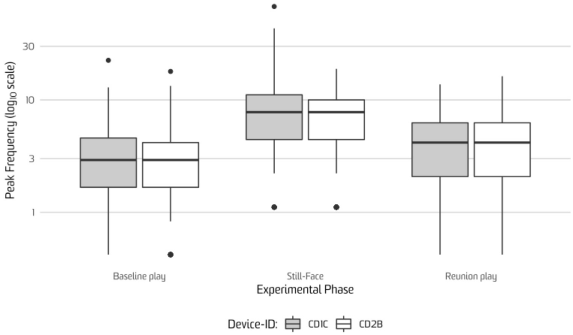
```
ANOVA Table (type III tests)

        Effect DFn    DFd       F        p p<.05     pes
1       device 1.0 360.00   0.079 7.78e-01       0.00022
2        phase 1.6 577.21 220.163 5.21e-61     * 0.37900
3 device:phase 1.6 577.21   0.738 4.51e-01       0.00200
```

plot peak events data

Hide

```
# reshape dataframe for plotting
peaks_vis <- peaks %>%
  mutate(phase = recode(phase,play="Baseline play",
                              sf="Still-Face",
                              reunion="Reunion play")) %>%
  rename("Device-ID:"=device,
         "Experimental Phase"=phase,
         "Peak Frequency (log scale)"=N)

### Plot peak frequencies -------------------------------------------------------
library(ggthemes)
library(wesanderson)
library(showtext)

# setup custom fonts
fontpath <- sysfonts::font_paths()
font_add(family="Sommet",regular="Sommet.otf",
           bold="SommetBold.otf",
           italic="SommetItalic.otf")
showtext::showtext_auto()

# prepare plot
peaks_vis %>%
  ggplot(aes(`Experimental Phase`,`Peak Frequency (log scale)`,fill = `Device-ID:`)) +
  theme_hc() %+replace% theme(
    text=element_text(family="Sommet"),
    axis.ticks = element_blank()) +
  # scale_fill_manual(values=wes_palette(name="GrandBudapest2")) +
```

```
  geom_boxplot() +
  scale_fill_grey(start = 0.8,end = 1) + # greyscale version
  scale_y_continuous(trans = 'log10') +
  ylab(expression(paste("Peak Frequency (log"[10]," scale)")))
```



simulation analysis

Hide

```
# 197 dead channels in total
dead_channels <- output %>%
  group_by(participant,device) %>%
  select(1:2,8) %>%
  mutate(avg_conduc = mean(conduc)) %>%
  filter(avg_conduc < 10 | avg_conduc > 60) %>%
  distinct(participant,device)

# 119 participants affected
dead_channels %>%
  ungroup() %>%
  select(participant) %>%
  distinct()

# exclude dead channels and collapse accelerometer columns
output_clean <- output %>%
  group_by(participant, device, phase) %>%
  # anti_join(dead_channels, by = "participant") %>%
  mutate(phase = fct_relevel(phase,"play","sf","reunion"),
         accel_xyz = abs(accel_x) + abs(accel_y) + abs(accel_z))

# 387 clean channels for 194 participants included
# 193 infants with 2 channels; 1 with 1 channel
output_clean %>%
  ungroup() %>%
  distinct(participant,device) %>%
  count(participant) %>%
  filter(n == 1)

### Randomise devices ----------------------------------------------------------

sim <- function (data) {

  # simple multivariate linear modeling function
  phase_fx <- function (df) {lm(conduc ~ phase_t + accel_xyz, data = df)}

    # randomly assign participants by device
    rand <- data %>%
      ungroup() %>%
      select(participant, device) %>%
      group_by(participant, device) %>%
      unique()
    rand %<>% group_by(participant) %>% mutate(assign = sample(c(0,1), size = 1, replace = TRUE))
    rand <- rand[permute::shuffle(rand),]
    rand$dup <- duplicated(rand[,c(1,3)])
    rand <- rand[-which(rand$dup),]
    rand %<>% select(1:2) %>% arrange(participant)

    # merge rand with `frog_lm` and delete rejected channels
    temp <- data %>%
      anti_join(rand, by = c("participant", "device")) %>%
      ungroup() %>% group_by(participant,device,phase) %>%
      nest() %>%
      mutate(
        mod = map(data, phase_fx),
        tidy = mod %>% map(broom::tidy),
        intercept = tidy %>% map("estimate") %>% map(1),# this 'plucks' the intercept value
        se = tidy %>% map("estimate") %>% map(2)
      ) %>%
      select(participant,device,phase,intercept,se)

  return(temp)
```

```
}
```

execute simulation

Hide

```
channels_simulated <- replicate(10000,sim(output_prep),simplify = FALSE)
```

extract simulated data

Hide

```
channels_simulated_clean <- channels_simulated %>%
  imap(~mutate(.x, rep = .y)) %>%
  map_df(as_tibble) %>%
  group_by(participant,device,phase) %>%
  mutate(
    rep = as.factor(rep),
    intercept = as.numeric(intercept),
    se = as.numeric(se)
  ) %>%
  select(6,1:5) %>%
  arrange(participant)
```

explore simulated differences

Hide

```
# what is the between-device difference in average intercept by participant by phase?
channels_simulated_clean %>%
  ungroup() %>%
  select(2:6) %>%
  filter(intercept > 0, intercept < 200) %>%
  group_by(participant,device,phase) %>%
  summarise(intercept_average = mean(intercept),
            se_average = mean(se)) %>%
  pivot_wider(names_from = device,
              values_from = c(intercept_average,se_average)) %>%
  mutate(intercept_difference = abs((intercept_average_CD1C-intercept_average_CD2B)),
         se_difference = abs((se_average_CD1C-se_average_CD2B))) %>%
  select(participant,phase,intercept_difference,se_difference)
```

participant <fctr>	phase <fctr>	intercept_difference <dbl>	se_difference <dbl>
3115	play	1.751190e+01	5.906846e-02
3115	reunion	4.004243e+00	1.879286e-01
3115	sf	1.289040e+00	3.976086e-02
3125	play	2.770892e+01	3.649738e-02
3125	reunion	2.067455e+01	3.333160e-02
3125	sf	2.657454e+01	5.149647e-02
3128	play	1.411913e+00	7.175652e-03
3128	reunion	1.927665e+00	2.916890e-03
3128	sf	1.743195e+00	4.494172e-02
3134	play	8.401885e+00	6.835411e-02

1-10 of 861 rows Previous 1 2 3 4 5 6 ... 87 Next

Hide

```
# what is overall between-device difference?
channels_simulated_clean %>%
  ungroup() %>%
  select(2:6) %>%
  filter(intercept > 0, intercept < 200) %>%
  group_by(participant,device,phase) %>%
  summarise(intercept_average = mean(intercept),
            se_average = mean(se)) %>%
  pivot_wider(names_from = device,
              values_from = c(intercept_average,se_average)) %>%
  mutate(intercept_difference = (intercept_average_CD1C-intercept_average_CD2B),
         se_difference = (se_average_CD1C-se_average_CD2B)) %>%
  select(participant,phase,intercept_difference,se_difference) %>%
  ungroup() %>%
  drop_na(intercept_difference,se_difference) %>%
  group_by(phase) %>%
  summarise(overall_difference = mean(intercept_difference),
            overall_se = mean(se_difference))
```

phase <fctr>	overall_difference <dbl>	overall_se <dbl>
play	1.213338	0.0007301041
reunion	1.718470	0.0013098158
sf	1.448525	0.0007712224

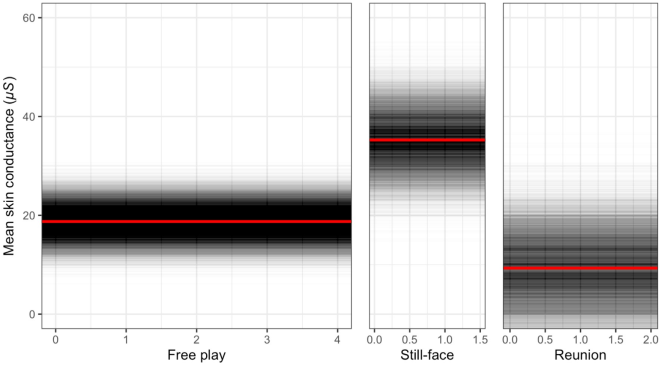
3 rows

build plot to visualise simulation results

Hide

```r
y_title <- expression(
  paste(
    "Mean skin conductance (",
    italic("µS"),
    ")"
  ))

play_sim <- channels_sim_outliers %>%
  ungroup() %>%
  group_by(rep,phase) %>%
  summarise(intercept = mean(intercept),
            se = mean(se)) %>%
  filter(phase == "play",intercept < 200) %>% {
  ggplot(.) + theme_bw() + xlim(0,4) + ylim(0, 60) +
  geom_abline(intercept = .$intercept, slope = 0, alpha = .006) +
  geom_hline(yintercept = mean(.$intercept), colour = "red", size = 1) +
  labs(x = "Free play", y = y_title)
  }

sf_sim <- channels_sim_outliers %>%
  ungroup() %>%
  group_by(rep,phase) %>%
  summarise(intercept = mean(intercept),
            se = mean(se)) %>%
  filter(phase == "sf",intercept < 200) %>% {
  ggplot(.) + theme_bw() + xlim(0,1.5) + ylim(0, 60) +
  geom_abline(intercept = .$intercept, slope = 0, alpha = .006) +
  geom_hline(yintercept = mean(.$intercept), colour = "red", size = 1) +
  labs(x = "Still-face")
  }

reunion_sim <- channels_sim_outliers %>%
  ungroup() %>%
  group_by(rep,phase) %>%
  summarise(intercept = mean(intercept),
            se = mean(se)) %>%
  filter(phase == "reunion",intercept < 200) %>% {
  ggplot(.) + theme_bw() + xlim(0,2) + ylim(0, 60) +
  geom_abline(intercept = .$intercept, slope = 0, alpha = .006) +
  geom_hline(yintercept = mean(.$intercept), colour = "red", size = 1) +
  labs(x = "Reunion")
  }

### Splice graphs together -----------------------------------------------------

library(ggpubr)
library(grid)
library(egg)

sf_figure <- ggarrange(
  play_sim,
  sf_sim +
    theme(axis.text.y = element_blank(),
                 axis.ticks.y = element_blank(),
                 axis.title.y = element_blank()),
  reunion_sim +
    theme(axis.text.y = element_blank(),
                 axis.ticks.y = element_blank(),
                 axis.title.y = element_blank()),
  widths = c(4,1.5,2),
  nrow = 1)

main_title <- expression(
  paste(
    "Simulation of mean conductance intercepts by participant by phase (",
    italic("n"),
    "=10000)"
  ))

annotate_figure(sf_figure,
  top = textGrob(
    main_title,
    hjust = 1.3, x = 1.09
  ),
  bottom = textGrob(
    "CBRC: Louis Klein, 29-10-20",
    gp = gpar(fontface = 3, fontsize = 9),
    hjust = 1.07, x = 1
  )
)
```

Simulation of mean conductance intercepts by participant by phase (*n*=10000)



CBRC: Louis Klein, 29-10-20

examine reasons for exclusion

Hide

```
load("/Users/moriarty/Dropbox/r-projects/REAL-EDA/workspace_14-07-20.RData")

exclusions %>%
  mutate(participant = substr(datafile,1,4),
         channel = substr(datafile,23,26)) %>%
  mutate_at(vars(participant,channel),~as.factor(.)) %>%
  count(issue) %>%
  mutate(N = round(n/sum(n)*100))
```

issue <chr>	**n** <int>	**N** <dbl>
Accel_LN_X is missing	18	13
Accel_LN_Y is missing	18	13
Accel_LN_Z is missing	18	13
Event marker cannot identify phase	66	47
Event_Marker is missing	20	14

5 rows

Hide

```
# there are 2 participants with only 1 channel affected out of 34
exclusions %>%
  mutate(participant = substr(datafile,1,4),
         channel = substr(datafile,23,26)) %>%
  mutate_at(vars(participant,channel),~as.factor(.)) %>%
  distinct(participant,channel) %>%
  count(participant) %>%
  count(n)
```

n <int>	**nn** <int>
1	2
2	32

2 rows

linear mixed-effects modelling

Hide

```
# make contrasts by device
nlme_cont <- matrix(c(-.5,.5), ncol=1)
contrasts(nlme_prep$device) <- nlme_cont

### Mixed-effects modelling ----------------------------------------------------

nlme_frog <- lme(conduc ~ 0 + phase_t*phase + device,
                 random = ~ phase_t*phase | participant,
                 # weights = ~ I(1/accel_xyz),
                 control = lmeControl(maxIter=100),
                 data = nlme_prep)

nlme_frog
```

```
Linear mixed-effects model fit by REML
  Data: nlme_prep
  Log-restricted-likelihood: -3804759
  Fixed: conduc ~ 0 + phase_t * phase + device
             phase_t            phaseplay         phasereunion              phasesf              device1
         0.004543738         19.758245126         23.128194571         26.735471152         -1.028378161
phase_t:phasereunion      phase_t:phasesf
        -0.036337410          0.036356070

Random effects:
 Formula: ~phase_t * phase | participant
 Structure: General positive-definite, Log-Cholesky parametrization
                     StdDev      Corr
(Intercept)          13.23505626 (Intr) phas_t phsrnn phassf phs_t:
phase_t               0.03839229  0.516
phasereunion         10.27415896 -0.405 -0.404
phasesf              11.04222245 -0.475 -0.467  0.545
phase_t:phasereunion  0.09398719 -0.145 -0.340  0.666 -0.049
phase_t:phasesf       0.10819359 -0.520 -0.584  0.475  0.859  0.033
Residual              6.51887098

Number of Observations: 1153738
Number of Groups: 255
```

Hide

```
summary(nlme_frog)
```

```
Linear mixed-effects model fit by REML
 Data: nlme_prep
```

	AIC	BIC	logLik
	<dbl>	<dbl>	<dbl>
	7609576	7609923	-3804759

1 row

```

Random effects:
 Formula: ~phase_t * phase | participant
 Structure: General positive-definite, Log-Cholesky parametrization
                     StdDev      Corr
(Intercept)          13.23505626 (Intr) phas_t phsrnn phassf phs_t:
phase_t               0.03839229  0.516
phasereunion         10.27415896 -0.405 -0.404
phasesf              11.04222245 -0.475 -0.467  0.545
phase_t:phasereunion  0.09398719 -0.145 -0.340  0.666 -0.049
phase_t:phasesf       0.10819359 -0.520 -0.584  0.475  0.859  0.033
Residual              6.51887098

Fixed effects: conduc ~ 0 + phase_t * phase + device
```

	Value	Std.Error	DF	t-value	p-value
	<chr>	<chr>	<chr>	<chr>	<chr>
phase_t	0.004544	0.0024309	1153477	1.86918	0.0616
phaseplay	19.758245	0.8310558	1153477	23.77487	0.0000
phasereunion	23.128195	0.8264224	1153477	27.98592	0.0000
phasesf	26.735471	0.7905368	1153477	33.81939	0.0000
device1	-1.028378	0.0122131	1153477	-84.20322	0.0000
phase_t:phasereunion	-0.036337	0.0060398	1153477	-6.01635	0.0000
phase_t:phasesf	0.036356	0.0068584	1153477	5.30092	0.0000

7 rows

```
 Correlation:
                     phas_t phsply phsrnn phassf devic1 phs_t:phsr
phaseplay             0.515
phasereunion          0.201  0.686
phasesf               0.131  0.631  0.675
device1               0.001  0.000  0.000  0.000
phase_t:phasereunion -0.336 -0.143  0.389 -0.190  0.000
phase_t:phasesf      -0.579 -0.516 -0.150  0.212  0.000  0.035

Standardized Within-Group Residuals:
         Min           Q1          Med           Q3          Max
-9.058921085 -0.398864941 -0.004807634  0.407580199 17.734410933

Number of Observations: 1153738
Number of Groups: 255
```

random effects variance-covariance matrix

Hide

```
getVarCov(nlme_frog)
```

```
Random effects variance covariance matrix
                     (Intercept)    phase_t phasereunion    phasesf phase_t:phasereunion phase_t:phasesf
(Intercept)            175.17000  0.2621300    -55.12400 -69.483000          -0.17996000     -0.74461000
phase_t                  0.26213  0.0014740     -0.15921  -0.197990          -0.00122610     -0.00242510
phasereunion           -55.12400 -0.1592100    105.56000  61.847000           0.64359000      0.52756000
phasesf                -69.48300 -0.1979900     61.84700 121.930000          -0.05042100      1.02600000
phase_t:phasereunion    -0.17996 -0.0012261      0.64359  -0.050421           0.00883360      0.00033771
phase_t:phasesf         -0.74461 -0.0024251      0.52756   1.026000           0.00033771      0.01170600
  Standard Deviations: 13.235 0.038392 10.274 11.042 0.093987 0.10819
```

create mixed effects summary

term	estimate	std.error	df	t.statistic	p.value	sig	lower.ci	upper.ci	d
<chr>	<chr>	<chr>	<dbl>	<chr>	<chr>	<chr>	<chr>	<chr>	<dbl>
phase_t	0.005	0.00	1153477	1.87	0.062	NA	0.000	0.009	NA
phaseplay	19.758	0.83	1153477	23.77	0.000	***	18.129	21.387	0.04
phasereunion	23.128	0.83	1153477	27.99	0.000	***	21.508	24.748	0.05
phasesf	26.735	0.79	1153477	33.82	0.000	***	25.186	28.285	0.06
device1	-1.028	0.01	1153477	-84.20	0.000	***	-1.052	-1.004	-0.16
phase_t:phasereunion	-0.036	0.01	1153477	-6.02	0.000	***	-0.048	-0.024	-0.01
phase_t:phasesf	0.036	0.01	1153477	5.30	0.000	***	0.023	0.050	0.01

7 rows

extract random effects for plotting

Hide

```
nlme_ranef <- coef(nlme_frog) %>%
  rownames_to_column() %>%
  rename(participant = "rowname",play = "phaseplay",sf = "phasesf",reunion = "phasereunion") %>%
  select(participant,play,sf,reunion) %>%
  pivot_longer(!participant,names_to="phase",values_to="ranef")

nlme_ranef_slopes <- coef(nlme_frog) %>%
  rownames_to_column() %>%
  rename(participant="rowname",play="phase_t",sf="phase_t:phasesf",reunion="phase_t:phasereunion") %>%
  pivot_longer(!participant,names_to="phase",values_to="ranef") %>%
  filter(!str_detect(phase,"phaseplay|phasesf|phasereunion|(Intercept)|accel_xyz|device1")) %>%
  rename(slope=ranef)

nlme_visu <- nlme_prep %>%
  full_join(nlme_ranef,by=c("participant","phase")) %>%
  full_join(nlme_prep,by=c("participant","device","phase","phase_t","conduc","accel_xyz")) %>%
  arrange(participant,device) %>%
  full_join(nlme_ranef_slopes,by=c("participant","phase")) %>%
  mutate(ranef = case_when(device=="CD2B"~ranef-1.028378,
                         device=="CD1C"~ranef+1.028378))
```

make individual plots for visual inspection with fixed and random effects overlaid

Hide

```
library(grid)
library(egg)
library(patchwork)

# create custom theme for missing/removed data
  theme_NA <- function (base_size = 11, base_family = "") {
    theme_bw() %+replace%
    theme(
      panel.grid.minor=element_blank(),
      panel.grid.major=element_blank(),
      panel.background=element_blank(),
      panel.border=element_blank()
    )}

for (i in seq_along(unique(nlme_visu$participant))) {

  temp_px <- unique(nlme_visu$participant)[i]
  temp_df <- nlme_visu %>% filter(participant == temp_px)

  ### Extract random effects ---------------------------------------------------

  CD1C_play_intercept <- temp_df %>% filter(device=="CD1C",phase=="play") %>% pull(ranef) %>% first()
  CD1C_play_slope <- temp_df %>% filter(device=="CD1C",phase=="play") %>% pull(slope) %>% first()
  CD1C_sf_intercept <- temp_df %>% filter(device=="CD1C",phase=="sf") %>% pull(ranef) %>% first()
  CD1C_sf_slope <- temp_df %>% filter(device=="CD1C",phase=="sf") %>% pull(slope) %>% first()
  CD1C_reunion_intercept <- temp_df %>% filter(device=="CD1C",phase=="reunion") %>% pull(ranef) %>% first()
  CD1C_reunion_slope <- temp_df %>% filter(device=="CD1C",phase=="reunion") %>% pull(slope) %>% first()

  CD2B_play_intercept <- temp_df %>% filter(device=="CD2B",phase=="play") %>% pull(ranef) %>% first()
  CD2B_play_slope <- temp_df %>% filter(device=="CD2B",phase=="play") %>% pull(slope) %>% first()
  CD2B_sf_intercept <- temp_df %>% filter(device=="CD2B",phase=="sf") %>% pull(ranef) %>% first()
  CD2B_sf_slope <- temp_df %>% filter(device=="CD2B",phase=="sf") %>% pull(slope) %>% first()
  CD2B_reunion_intercept <- temp_df %>% filter(device=="CD2B",phase=="reunion") %>% pull(ranef) %>% first()
  CD2B_reunion_slope <- temp_df %>% filter(device=="CD2B",phase=="reunion") %>% pull(slope) %>% first()
  # n.b. fixed effects are inherited from environment

  ### Build plots --------------------------------------------------------------
```

```
  CD1C_play_temp <- temp_df %>%
        filter(device=="CD1C",phase=="play") %>%
        ungroup() %>% group_by(participant)

  CD1C_play_shader <- CD1C_play_temp %>%
    ungroup() %>%
    filter(conduc<1) %>%
    pull(phase_t)

  if (all(rapportools::is.empty(CD1C_play_temp$conduc))==FALSE) {
    CD1C_play_temp <- CD1C_play_temp %>%
      ggplot(aes(x=phase_t,y=conduc,group=participant)) +
      theme_bw() + theme(axis.title.y=element_blank(),
                         axis.text.x=element_blank(),
                         axis.ticks.x=element_blank(),
                         axis.title.x=element_blank(),
                         plot.margin=margin(r=2,l=0,t=10,b=0)) +
      geom_line(alpha=.8,size=.2) +
      geom_vline(xintercept=CD1C_play_shader,colour="grey92") +
      coord_cartesian(ylim=c(0,80)) +
      scale_x_continuous(expand=c(0,0)) + scale_y_continuous(expand=c(0,0)) +
      geom_abline(intercept=nlme_fixed_play_intercept,slope=nlme_fixed_play_slope,colour="red",size=.3,alpha=.5) +
      geom_abline(intercept=CD1C_play_intercept,slope=CD1C_play_slope,colour="blue",size=.3,alpha=.5)
  } else {
    CD1C_play_temp <- tibble(x=1,y=1) %>%
      ggplot(aes(x,y)) +
      theme_NA() + theme(axis.text.y=element_blank(),
                         axis.ticks.y=element_blank(),
                         axis.title.y=element_blank(),
                         axis.text.x=element_blank(),
                         axis.ticks.x=element_blank(),
                         axis.title.x=element_blank(),
                         plot.margin=margin(r=2,l=0,t=10,b=0))
  }

  CD2B_play_temp <- temp_df %>%
    filter(device=="CD2B",phase=="play") %>%
    ungroup() %>% group_by(participant)

  CD2B_play_shader <- CD2B_play_temp %>%
    ungroup() %>%
    filter(conduc<1) %>%
    pull(phase_t)

  if (all(rapportools::is.empty(CD2B_play_temp$conduc))==FALSE) {
    CD2B_play_temp <- CD2B_play_temp %>%
      ggplot(aes(x=phase_t,y=conduc,group=participant)) +
      theme_bw() + theme(axis.title.y=element_blank(),
                         plot.margin=margin(r=2,l=0,t=10,b=0)) +
      geom_line(alpha=.8,size=.2) +
      geom_vline(xintercept=CD2B_play_shader,colour="grey92") +
      coord_cartesian(ylim=c(0,80)) +
      scale_x_continuous(expand=c(0,0)) + scale_y_continuous(expand=c(0,0)) +
      geom_abline(intercept=nlme_fixed_play_intercept,slope=nlme_fixed_play_slope,colour="red",size=.3,alpha=.5) +
      geom_abline(intercept=CD2B_play_intercept,slope=CD2B_play_slope,colour="blue",size=.3,alpha=.5) +
      labs(x="Free play")
  } else {
    CD2B_play_temp <- tibble(x=1,y=1) %>%
      ggplot(aes(x,y)) +
      theme_NA() + theme(axis.title.y=element_blank(),
                         axis.text.y=element_blank(),
                         axis.ticks.y=element_blank(),
                         axis.text.x=element_blank(),
                         axis.ticks.x=element_blank(),
                         plot.margin=margin(r=2,l=0,t=10,b=0)) +
      labs(x="Free play")
  }

  CD1C_sf_temp <- temp_df %>%
    filter(device=="CD1C",phase=="sf") %>%
    ungroup() %>% group_by(participant)

  CD1C_sf_shader <- CD1C_sf_temp %>%
    ungroup() %>%
    filter(conduc<1) %>%
    pull(phase_t)

  if (all(rapportools::is.empty(CD1C_sf_temp$conduc))==FALSE) {
    CD1C_sf_temp <- CD1C_sf_temp %>%
      ggplot(aes(x=phase_t,y=conduc,group=participant)) +
      theme_bw() + theme(strip.background=element_blank(),
                         strip.text.y=element_blank(),
                         axis.text.y=element_blank(),
                         axis.ticks.y=element_blank(),
                         axis.title.y=element_blank(),
                         axis.text.x=element_blank(),
                         axis.ticks.x=element_blank(),
                         axis.title.x=element_blank(),
                         plot.margin=margin(r=2,l=2,t=10,b=0)) +
      geom_line(alpha=.8,size=.2) +
      geom_vline(xintercept=CD1C_sf_shader,colour="grey92") +
      coord_cartesian(ylim=c(0,80)) +
      scale_x_continuous(expand=c(0,0)) + scale_y_continuous(expand=c(0,0)) +
      geom_abline(intercept=nlme_fixed_sf_intercept,slope=nlme_fixed_sf_slope,colour="red",size=.3,alpha=.5) +
      geom_abline(intercept=CD1C_sf_intercept,slope=CD1C_sf_slope,colour="blue",size=.3,alpha=.5)
  } else {
    CD1C_sf_temp <- tibble(x=1,y=1) %>%
```

```
      ggplot(aes(x,y)) +
      theme_NA() + theme(strip.background=element_blank(),
                         strip.text.y=element_blank(),
                         axis.text.y=element_blank(),
                         axis.ticks.y=element_blank(),
                         axis.title.y=element_blank(),
                         axis.text.x=element_blank(),
                         axis.ticks.x=element_blank(),
                         axis.title.x=element_blank(),
                         plot.margin=margin(r=2,l=2,t=10,b=0))
  }

  CD2B_sf_temp <- temp_df %>%
    filter(device=="CD2B",phase=="sf") %>%
    ungroup() %>% group_by(participant)

  CD2B_sf_shader <- CD2B_sf_temp %>%
    ungroup() %>%
    filter(conduc<1) %>%
    pull(phase_t)

  if (all(rapportools::is.empty(CD2B_sf_temp$conduc))==FALSE) {
    CD2B_sf_temp <- CD2B_sf_temp %>%
      ggplot(aes(x=phase_t,y=conduc,group=participant)) +
      theme_bw() + theme(strip.background=element_blank(),
                         strip.text.y=element_blank(),
                         axis.text.y=element_blank(),
                         axis.ticks.y=element_blank(),
                         axis.title.y=element_blank(),
                         plot.margin=margin(r=2,l=2,t=10,b=0)) +
      geom_line(alpha=.8,size=.2) +
      geom_vline(xintercept=CD2B_sf_shader,colour="grey92") +
      coord_cartesian(ylim=c(0,80)) +
      scale_x_continuous(expand=c(0,0)) + scale_y_continuous(expand=c(0,0)) +
      geom_abline(intercept=nlme_fixed_sf_intercept,slope=nlme_fixed_sf_slope,colour="red",size=.3,alpha=.5) +
      geom_abline(intercept=CD2B_sf_intercept,slope=CD2B_sf_slope,colour="blue",size=.3,alpha=.5) +
      labs(x="Still-face")
  } else {
    CD2B_sf_temp <- tibble(x=1,y=1) %>%
      ggplot(aes(x,y)) +
      theme_NA() + theme(strip.background=element_blank(),
                         strip.text.y=element_blank(),
                         axis.text.y=element_blank(),
                         axis.ticks.y=element_blank(),
                         axis.title.y=element_blank(),
                         axis.text.x=element_blank(),
                         axis.ticks.x=element_blank(),
                         plot.margin=margin(r=2,l=2,t=10,b=0)) +
      labs(x="Still-face")
  }

  CD1C_reunion_temp <- temp_df %>%
    filter(device=="CD1C",phase=="reunion") %>%
    ungroup() %>% group_by(participant)

  CD1C_reunion_shader <- CD1C_reunion_temp %>%
    ungroup() %>%
    filter(conduc<1) %>%
    pull(phase_t)

  if (all(rapportools::is.empty(CD1C_reunion_temp$conduc))==FALSE) {
    CD1C_reunion_temp <- CD1C_reunion_temp %>%
      ggplot(aes(x=phase_t,y=conduc,group=participant)) +
      theme_bw() + theme(strip.background=element_blank(),
                         strip.text.y=element_blank(),
                         axis.text.y=element_blank(),
                         axis.ticks.y=element_blank(),
                         axis.title.y.left=element_blank(),
                         axis.text.x=element_blank(),
                         axis.ticks.x=element_blank(),
                         axis.title.x=element_blank(),
                         plot.margin=margin(r=0,l=2,t=10,b=0)) +
      geom_line(alpha=.8,size=.2) +
      geom_vline(xintercept=CD1C_reunion_shader,colour="grey92") +
      coord_cartesian(ylim=c(0,80)) +
      scale_x_continuous(expand=c(0,0)) + scale_y_continuous(expand=c(0,0),sec.axis=sec_axis(~.,name="CD1C")) +
      geom_abline(intercept=nlme_fixed_reunion_intercept,slope=nlme_fixed_reunion_slope,colour="red",size=.3,alph
a=.5) +
      geom_abline(intercept=CD1C_reunion_intercept,slope=CD1C_reunion_slope,colour="blue",size=.3,alpha=.5)
  } else {
    CD1C_reunion_temp <- tibble(x=1,y=1) %>%
      ggplot(aes(x,y)) +
      theme_NA() + theme(strip.background=element_blank(),
                         strip.text.y=element_blank(),
                         axis.text.y=element_blank(),
                         axis.ticks.y=element_blank(),
                         axis.title.y.left=element_blank(),
                         axis.text.x=element_blank(),
                         axis.ticks.x=element_blank(),
                         axis.title.x=element_blank(),
                         plot.margin=margin(r=0,l=2,t=10,b=0))
  }

  CD2B_reunion_temp <- temp_df %>%
    filter(device=="CD2B",phase=="reunion") %>%
    ungroup() %>% group_by(participant)

  CD2B_reunion_shader <- CD2B_reunion_temp %>%
    ungroup() %>%
```

```
    filter(conduc<1) %>%
    pull(phase_t)

  if (all(rapportools::is.empty(CD2B_reunion_temp$conduc))==FALSE) {
    CD2B_reunion_temp <- CD2B_reunion_temp %>%
      ggplot(aes(x=phase_t,y=conduc,group=participant)) +
      theme_bw() + theme(strip.background=element_blank(),
                         strip.text.y=element_blank(),
                         axis.text.y=element_blank(),
                         axis.ticks.y=element_blank(),
                         axis.title.y.left=element_blank(),
                         plot.margin=margin(r=0,l=2,t=10,b=0)) +
      geom_line(alpha=.8,size=.2) +
      geom_vline(xintercept=CD2B_reunion_shader,colour="grey92") +
      coord_cartesian(ylim=c(0,80)) +
      scale_x_continuous(expand=c(0,0)) + scale_y_continuous(expand=c(0,0),sec.axis=sec_axis(~.,name="CD2B")) +
      geom_abline(intercept=nlme_fixed_reunion_intercept,slope=nlme_fixed_reunion_slope,colour="red",size=.3,alph
a=.5) +
      geom_abline(intercept=CD2B_reunion_intercept,slope=CD2B_reunion_slope,colour="blue",size=.3,alpha=.5) +
      labs(x="Reunion")
  } else {
    CD2B_reunion_temp <- tibble(x=1,y=1) %>%
      ggplot(aes(x,y)) +
      theme_NA() + theme(strip.background=element_blank(),
                         strip.text.y=element_blank(),
                         axis.text.y=element_blank(),
                         axis.ticks.y=element_blank(),
                         axis.title.y.left=element_blank(),
                         plot.margin=margin(r=0,l=2,t=10,b=0)) +
      labs(x="Reunion")
    # + geom_text(aes(label="NA",fontface="italic"))
  }

  CD1C_temp_plot <- CD1C_play_temp + CD1C_sf_temp + CD1C_reunion_temp + plot_layout(widths = c(4,1.5,2),nrow = 1)
  CD2B_temp_plot <- CD2B_play_temp + CD2B_sf_temp + CD2B_reunion_temp + plot_layout(widths = c(4,1.5,2),nrow = 1)

  temp_plot <- (CD1C_temp_plot/CD2B_temp_plot) + plot_annotation(
    title=expression(paste("Skin conductance data for participant ",temp_px,".")),
    subtitle="Dummy text",
    caption=expression(paste(italic("CBRC"),": Louis Klein, 13-11-20"))
  )

  ggsave(filename=paste0(temp_px,"_ranef.png"))
}
```

prepare sociodemographics for manuscript

Hide

```
load("~/Dropbox/r-projects/REAL-EDA/snapshot_28-12-20.RData")

real <- read_csv("datafiles/WMGFR_NBQ&SF_LK.csv") %>%
  select(record_id,bcweight,bgest,bcgender,bmyearsaust,bcethnicity,bmdob) %>%
  mutate_at(vars(bcweight,bgest,bcgender,bmyearsaust,bcethnicity),~case_when(.==99~NA_real_,TRUE~.)) %>%
  mutate_at(vars(record_id,bcgender,bcethnicity),~as.factor(.)) %>%
  full_join(nlme_prep %>%
              ungroup() %>%
              select(participant) %>%
              transmute(record_id = participant) %>%
              distinct() %>%
              mutate(eda = 1)) %>%
  mutate(eda = case_when(is.na(eda)==TRUE~0,
                         TRUE~eda) %>% as.factor(.),
         bmdob = lubridate::parse_date_time(bmdob,"dmy")) %>%
  group_by(eda)

glimpse(real,width=100)
```

```
Rows: 788
Columns: 8
Groups: eda [2]
$ record_id   <fct> 3100, 3101, 3102, 3103, 3104, 3105, 3106, 3107, 3108, 3109, 3110, 3111, 3112,…
$ bcweight    <dbl> 3030, 3710, 3770, 3660, 3800, 3540, 3400, 3650, 3460, 3380, 3200, 3650, 3300,…
$ bgest       <dbl> 39, 40, 42, 38, 39, 41, 37, 39, 39, 39, 39, 40, 39, 35, 41, 39, 39, 38, 42, 3…
$ bcgender    <fct> 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0,…
$ bmyearsaust <dbl> 2, 5, 1, 5, 5, 5, 3, 1, 5, 5, 2, 2, 5, 1, 5, 1, 5, 5, 2, 5, 2, 4, 3, 5, 5, 3,…
$ bcethnicity <fct> 2, 1, 9, 1, 1, 1, 4, 2, 4, 4, 5, 6, 8, 5, 2, 5, 4, 1, 3, 1, 5, 2, 6, 1, 5, 5,…
$ bmdob       <dttm> 1976-05-09, 1978-12-11, 1990-10-05, 1987-05-18, 1988-05-25, 1986-01-17, 1983…
$ eda         <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,…
```

is there a difference in ages between total recruited and final sample?

Hide

```
(age_summary <- real %>%
  full_join(
    read_csv("datafiles/WMGFRDataProject_DATA_2021-04-23_2008.csv") %>%
      select(record_id,cmdate) %>%
      drop_na() %>%
      mutate(record_id = as.factor(record_id),
             cmdate = lubridate::parse_date_time(cmdate,"dmy")),
  by="record_id") %>%
  mutate(age_at_test = as.numeric((cmdate-bmdob)/365),
         record_id = as.numeric(as.character(record_id))) %>%
  relocate(record_id,bmdob,cmdate,age_at_test) %>%
  select(record_id,eda,age_at_test) %>%
  drop_na(age_at_test) %>%
  group_by(eda) %>%
  add_count(record_id) %>%
  summarise(age_mean = mean(age_at_test),
            age_sd = sd(age_at_test),
            age_n = as.numeric(sum(n))))
```

eda <fctr>	age_mean <dbl>	age_sd <dbl>	age_n <dbl>
0	31.85102	8.296038	143
1	31.70724	4.976025	253

2 rows

t-test on means from above

Hide

```
(age_test <- DescTools::TTestA(
  mx=31.85102,my=31.70724,
  sx=8.296038,sy=4.976025,
  nx=143,ny=253
  ))
```

```

	Welch Two Sample t-test

data:  x and y
t = 0.18893, df = 200.94, p-value = 0.8503
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.356837  1.644397
sample estimates:
mean of x mean of y
 31.85102  31.70724
```

is there a difference between genders between total recruited and final sample?

Hide

```
comp_g <- as.table(rbind(c(286,247),c(121,134)))
dimnames(comp_g) <- list(sample=c("total","final"),
                         gender=c("male","female"))

comp_g_chisq <- chisq.test(comp_g,correct=TRUE)
(comp_g_chisq)
```

```

	Pearson's Chi-squared test with Yates' continuity correction

data:  comp_g
X-squared = 2.4187, df = 1, p-value = 0.1199
```

Hide

```
# effect size function
chisq.es <- function (htest) {
  return(
    unname(
      sqrt(htest$statistic/sum(htest$observed))
    ))}

chisq.es(comp_g_chisq)
```

```
[1] 0.0554019
```

generalised linear model to examine influence of sociodemgraphic predictor vars

Hide

```
comp_glm <- glm(formula=eda~bcweight+bgest+bmyearsaust,data=drop_na(real),family=binomial)
summary(comp_glm)
```

```

Call:
glm(formula = eda ~ bcweight + bgest + bmyearsaust, family = binomial,
    data = drop_na(real))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0081  -0.9055  -0.8647   1.4540   1.8085

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.8293029  1.6287293  -2.351   0.0187 *
bcweight    -0.0001101  0.0001682  -0.655   0.5127
bgest        0.0880762  0.0491282   1.793   0.0730 .
bmyearsaust  0.0156227  0.0474232   0.329   0.7418
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 973.76  on 771  degrees of freedom
Residual deviance: 969.85  on 768  degrees of freedom
AIC: 977.85

Number of Fisher Scoring iterations: 4
```

extract confidence intervals for GLM

Hide

```
round(exp(cbind(OR = coef(comp_glm),confint(comp_glm))),2)
```

```
              OR 2.5 % 97.5 %
(Intercept) 0.02  0.00   0.48
bcweight    1.00  1.00   1.00
bgest       1.09  0.99   1.21
bmyearsaust 1.02  0.93   1.12
```

Hosmer and Lemeshow goodness of fit (GOF) test

Hide

```
library(ResourceSelection)

real2 <- real %>%
  ungroup() %>%
  column_to_rownames("record_id") %>%
  select(!c(bmdob,bcethnicity,bcgender)) %>%
  mutate_all(~as.numeric(as.character(.))) %>%
  relocate(eda) %>%
  drop_na() # note: hoslem.test() breaks on NA values

real_glm <- glm(formula=eda~bcweight+bmyearsaust+bgest,data=real2,family=binomial)

ResourceSelection::hoslem.test(real2$eda,fitted(real_glm),g=10)
```

```

    Hosmer and Lemeshow goodness of fit (GOF) test

data:  real2$eda, fitted(real_glm)
X-squared = 13.452, df = 8, p-value = 0.0972
```

print session info

Hide

```
sessionInfo()
```

```
R version 3.6.1 (2019-07-05)
Platform: x86_64-apple-darwin15.6.0 (64-bit)
Running under: macOS Catalina 10.15.5

Matrix products: default
BLAS:   /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versions/A/libBLA
S.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib

locale:
[1] en_AU.UTF-8/en_AU.UTF-8/en_AU.UTF-8/C/en_AU.UTF-8/en_AU.UTF-8

attached base packages:
[1] grid      stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
 [1] ResourceSelection_0.3-5 broom.mixed_0.2.6       egg_0.4.5               gridExtra_2.3
 [5] nlme_3.1-150            showtext_0.9-1          showtextdb_3.0          sysfonts_0.8.2
 [9] wesanderson_0.3.6       ggthemes_4.2.0          rstatix_0.6.0.999       ggpubr_0.4.0
[13] broomExtra_4.1.0        broom_0.7.2             kableExtra_1.3.1        magrittr_2.0.1
[17] lubridate_1.7.9.2       tsibble_0.9.3           forcats_0.5.0           stringr_1.4.0
[21] dplyr_1.0.2             purrr_0.3.4             readr_1.4.0             tidyr_1.1.2
[25] tibble_3.0.4            ggplot2_3.3.2           tidyverse_1.3.0         knitr_1.30
[29] easypackages_0.1.0

loaded via a namespace (and not attached):
```

```
  [1] readxl_1.3.1        backports_1.2.0     Hmisc_4.4-1         plyr_1.8.6          TMB_1.7.18
  [6] splines_3.6.1       usethis_1.6.3       digest_0.6.27       htmltools_0.5.0     rsconnect_0.8.16
 [11] lmerTest_3.1-3      fansi_0.4.1         checkmate_2.0.0     memoise_1.1.0       cluster_2.1.0
 [16] openxlsx_4.2.3      remotes_2.2.0       modelr_0.1.8        anytime_0.3.9       prettyunits_1.1.1
 [21] jpeg_0.1-8.1        colorspace_2.0-0    rvest_0.3.6         haven_2.3.1         xfun_0.19
 [26] callr_3.5.1         crayon_1.3.4        jsonlite_1.7.1      Exact_2.1           lme4_1.1-25
 [31] survival_3.2-7      glue_1.4.2          gtable_0.3.0        emmeans_1.5.3       webshot_0.5.2
 [36] sjstats_0.18.0      sjmisc_2.8.5        car_3.0-10          pkgbuild_1.1.0      abind_1.4-5
 [41] rapportools_1.0     scales_1.1.1        mvtnorm_1.1-1       DBI_1.1.0           broman_0.71-6
 [46] Rcpp_1.0.5          viridisLite_0.3.0   xtable_1.8-4        performance_0.5.1   htmlTable_2.1.0
 [51] reticulate_1.18     foreign_0.8-75      Formula_1.2-4       httr_1.4.2          htmlwidgets_1.5.2
 [56] RColorBrewer_1.1-2  ellipsis_0.3.1      farver_2.0.3        pkgconfig_2.0.3     nnet_7.3-14
 [61] dbplyr_2.0.0        utf8_1.1.4          labeling_0.4.2      reshape2_1.4.4      tidyselect_1.1.0
 [66] rlang_0.4.9         effectsize_0.4.1    munsell_0.5.0       cellranger_1.1.0    tools_3.6.1
 [71] cli_2.2.0           generics_0.1.0      sjlabelled_1.1.7    devtools_2.3.2      evaluate_0.14
 [76] arm_1.11-2          yaml_2.2.1          processx_3.4.4      fs_1.5.0            zip_2.1.1
 [81] pander_0.6.3        rootSolve_1.8.2.1   xml2_1.3.2          compiler_3.6.1      rstudioapi_0.13
 [86] curl_4.3            png_0.1-7           e1071_1.7-4         testthat_3.0.0      ggsignif_0.6.0
 [91] reprex_0.3.0        statmod_1.4.35      DescTools_0.99.39   stringi_1.5.3       highr_0.8
 [96] ps_1.4.0            parameters_0.10.1   desc_1.2.0          lattice_0.20-41     Matrix_1.2-18
[101] nloptr_1.2.2.2      permute_0.9-5       vctrs_0.3.5         pillar_1.4.7        lifecycle_0.2.0
[106] estimability_1.3    cowplot_1.1.0       data.table_1.13.2   insight_0.11.0      lmom_2.8
[111] patchwork_1.1.0     R6_2.5.0            latticeExtra_0.6-29 rio_0.5.16          gld_2.6.2
[116] sessioninfo_1.1.1   boot_1.3-25         MASS_7.3-53         assertthat_0.2.1    pkgload_1.1.0
[121] rprojroot_2.0.2     withr_2.3.0         bayestestR_0.8.0    expm_0.999-5        hms_0.5.3
[126] DataCombine_0.2.21  rpart_4.1-15        coda_0.19-4         class_7.3-17        minqa_1.2.4
[131] EMAtools_0.1.3      rmarkdown_2.5       carData_3.0-4       numDeriv_2016.8-1.1 base64enc_0.1-3
[136] tinytex_0.27
```