Wilcoxon Signed Rank Test

Joseph Oliveira

7/4/2020

Loading data in from the `Wilcoxon_Test.Rmd` cleaning procedure

```
faces <- faces %>%
  group_by(Group, `Participant #`, Time, Survey) %>%
  summarise(tot_resp = sum(Response)) %>%
  ungroup()
```

Wilcoxon Test function

1. Cleaned `faces` data is filtered for the Time (`Pre` or `Post`) and Group (`Experimental` or `Control`). Depends one what is being compared
2. Two data sets are created to create the vectors for comparison
3. 3 Cases need to be handled a. If the vector of response variables in data set A are longer than B b. If the vector of response variables in data set B are longer than A c. If the vector of response variables in data set A and B are equal length

```
faces_wilcox <- function(time1, group1, time2, group2, survey, ...) {
  # Filter to data we need for comparison
  faces_ <- faces %>%
    filter(Time %in% c(time1, time2) & Group %in% c(group1, group2))

  # Create 2 datasets
  comp1 <- faces_ %>%
    filter(Time == time1, Group == group1, Survey == survey)

  comp2 <- faces_ %>%
    filter(Time == time2, Group == group2, Survey == survey)

  n1 <- length(comp1$tot_resp)
  n2 <- length(comp2$tot_resp)

  x <- wilcox.test(comp1$tot_resp, comp2$tot_resp, ...)
      x$obs <- n1 + n2
      return(x)
  }
```

Creating the comparison groups for the function above.

```r
distinct_groupings <- faces %>%
  distinct(Group, Time, Survey)

dist_grp_exp_pre <- distinct_groupings %>%
  filter(Group == 'Experimental', Time == 'Pre')

dist_grp_ctrl_pre <- distinct_groupings %>%
  filter(Group == 'Control', Time == 'Pre')

dist_grp_exp_post <- distinct_groupings %>%
  filter(Group == 'Experimental', Time == 'Post')

dist_grp_ctrl_post <- distinct_groupings %>%
  filter(Group == 'Control', Time == 'Post')

first_comp <- inner_join(dist_grp_exp_pre, dist_grp_ctrl_pre, by = c("Survey", "Time"))
secnd_comp <- inner_join(dist_grp_exp_post, dist_grp_ctrl_post, by = c("Survey", "Time"))
third_comp <- inner_join(dist_grp_exp_pre, dist_grp_exp_post, by = c("Survey", "Group"))
forth_comp <- inner_join(dist_grp_ctrl_pre, dist_grp_ctrl_post, by = c("Survey", "Group"))

group_comparison <- bind_rows(first_comp, secnd_comp) %>% mutate_all(as.character)

time_comparison <- bind_rows(third_comp, forth_comp) %>% mutate_all(as.character)
```

Running Wilcoxon tests

test: `pmap` will map specified columns of a tibble to the arguments of a function., and will use each record of the specified columns as an argument in the function specified.

p.value: `map` is like `lapply` but for tibbles. Just like `pmap` it iterates through each record of specified column, passing it to the function call. Here the function calls are subsetting calls, just instead of `x[1]` or `x[[1]]` I am calling `'['(x)` or rater `'['( '['( x ) )`. The back-ticks make R treat the call as a `prefix`, `fn(arg1, arg2)`, instead of `infix`, `arg1 fn arg2`.

- For each test result, I'm pulling out the p.value: `test[[x]]$p.value`

```r
group_wilcoxon <- group_comparison %>%
  # defined the test result
  mutate(test = pmap(list(time1 = Time, group1 = Group.x,
                          time2 = Time, group2 = Group.y,
                          survey = Survey),
                     faces_wilcox, paired = F, alternative = "two.sided"),
         test_statistic = unlist(map(test, function(x) `$`(`[`(`[`(`[`(x)), 'statistic'))),
         p.value = unlist(map(test, function(x) `$`(`[`(`[`(x)), 'p.value'))),
         effect_size =  test_statistic / unlist(map(test, function(x) `$`(`[`(`[`(x)), 'obs')))**0.5)

time_wilcoxon  <- time_comparison %>%
  mutate(test = pmap(list(time1 = Time.x, group1 = Group,
                          time2 = Time.y, group2 = Group,
                          survey = Survey),
                     faces_wilcox, paired = T, alternative = "less"),
```

```
        test_statistic = unlist(map(test, function(x) `$`(`[`(`[`(x)), 'statistic'))),
        p.value = unlist(map(test, function(x) `$`(`[`(`[`(x)), 'p.value'))),
        effect_size = test_statistic / unlist(map(test, function(x) `$`(`[`(`[`(x)), 'obs')))**0.5)

c1 <- faces %>%
  filter(Time == 'Pre' & Group == 'Experimental' & Survey == 'FACES') %>%
  pull(tot_resp)
c2 <- faces %>%
  filter(Time == 'Post' & Group == 'Experimental' & Survey == 'FACES') %>%
  pull(tot_resp)
wilcox.test(c1, c2, paired = T, alternative = 'less')
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  c1 and c2
## V = 18, p-value = 0.1792
## alternative hypothesis: true location shift is less than 0
```

```
time_wilcoxon
```

```
## # A tibble: 12 x 8
##    Group        Time.x Survey Time.y test    test_statistic p.value effect_size
##    <chr>        <chr>  <chr>  <chr>  <list>           <dbl>   <dbl>       <dbl>
##  1 Experimental Pre    FACES  Post   <htest>             18 0.179         4.02
##  2 Experimental Pre    AKS    Post   <htest>             16 0.131         3.58
##  3 Experimental Pre    FES    Post   <htest>              1 0.00195       0.224
##  4 Experimental Pre    SCS    Post   <htest>              0 0.0108        0
##  5 Experimental Pre    FPPS   Post   <htest>              8 0.175         2
##  6 Experimental Pre    SEAS   Post   <htest>              3 0.0195        0.75
##  7 Control      Pre    FACES  Post   <htest>             10 0.791         2.89
##  8 Control      Pre    AKS    Post   <htest>             11 0.584         3.18
##  9 Control      Pre    FES    Post   <htest>              2 0.0888        0.577
## 10 Control      Pre    SCS    Post   <htest>              1 0.0987        0.354
## 11 Control      Pre    FPPS   Post   <htest>              2 0.188         0.707
## 12 Control      Pre    SEAS   Post   <htest>              7 0.5           2.02
```

Wilcoxon Test Results

Experimental vs Control

Time	Survey	Group.x	Group.y	test_statistic	p.value	effect_size
Pre	AKS	Experimental	Control	21.0	0.3438231	5.250000
Post	AKS	Experimental	Control	36.0	0.5469450	9.000000
Post	FACES	Experimental	Control	42.0	0.2109217	10.500000
Pre	FACES	Experimental	Control	30.0	1.0000000	7.500000
Post	FES	Experimental	Control	43.0	0.1742049	10.750000
Pre	FES	Experimental	Control	33.0	0.7861121	8.250000
Pre	FPPS	Experimental	Control	28.0	0.0499907	8.082904
Post	FPPS	Experimental	Control	27.5	0.0603629	7.938566
Pre	SCS	Experimental	Control	21.5	0.3949381	6.206515

Time	Survey	Group.x	Group.y	test_statistic	p.value	effect_size
Post	SCS	Experimental	Control	17.5	0.8641908	5.051815
Post	SEAS	Experimental	Control	32.5	0.2990292	8.685990
Pre	SEAS	Experimental	Control	22.5	0.8971665	6.013378

Pre vs Post

Group	Survey	Time.x	Time.y	test_statistic	p.value	effect_size
Experimental	AKS	Pre	Post	16	0.1309413	3.5777088
Control	AKS	Pre	Post	11	0.5839463	3.1754265
Experimental	FACES	Pre	Post	18	0.1791632	4.0249224
Control	FACES	Pre	Post	10	0.7907539	2.8867513
Experimental	FES	Pre	Post	1	0.0019531	0.2236068
Control	FES	Pre	Post	2	0.0887649	0.5773503
Experimental	FPPS	Pre	Post	8	0.1745382	2.0000000
Control	FPPS	Pre	Post	2	0.1875000	0.7071068
Experimental	SCS	Pre	Post	0	0.0107697	0.0000000
Control	SCS	Pre	Post	1	0.0987330	0.3535534
Experimental	SEAS	Pre	Post	3	0.0195313	0.7500000
Control	SEAS	Pre	Post	7	0.5000000	2.0207259