Solution for the assignment of the third class

Kovacs Marton

9/29/2021

Importing data

```
raw <- read_tsv("data/boldog.txt", locale = locale(encoding = "UTF-8"))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double()
## )
## i Use `spec()` for the full column specifications.
```

R sometimes have a problem with special characters. I will fix that now.

```
# first I fix the special Hungarian characters
Encoding(names(raw)) <- 'latin1'
# I translate them to English letters for the sake of simplicity
iconv(names(raw), from = "latin1", to = "ASCII//TRANSLIT")
```

```
##  [1] "index"    "Neme"     "Eletkora" "Gyermeke" "Isk"      "Anyagi"
##  [7] "PElmeny%" "Testi-Fi" "AltLelki" "AltEgAll" "Fizero"   "Arcocska"
## [13] "Aggodalo" "Ideges"   "Feszult"  "Nyugtala" "Diener1"  "Diener2"
## [19] "Diener3"  "Diener4"  "Diener5"  "Diener6"  "Diener7"  "Diener8"
## [25] "Jollet"   "Savor"    "AVhat"    "Onreg"    "Rezil"    "M_Flow"
## [31] "GJerz"    "GJpszi"   "GJszoc"   "GJspir"   "GJerzpsz" "GJspszoc"
## [37] "PIK_MM"   "PIK_AV"   "PIK_Onr"  "PIK_Rez"  "PPozErz"  "PElmely"
## [43] "PPozKapc" "PErtCel"  "PTelj"    "PBoldog"  "PEgeszs"  "PNegErz"
## [49] "PMagany"  "PERMA"
```

Also, I like to use *snake_case* for my variable names, therefore I will transform the variable names.

```
raw <- janitor::clean_names(raw)
```

I will use this dataset for further assignments, so I save the cleaned version in .tsv format.

```
write_tsv(raw, "data/boldog_raw.tsv")
```

Data exploration

```
skimr::skim(raw) %>%
  kable()
```

skim_type	skim_variable	n_missing	complete_rate	numeric.mean	numeric.sd	numeric.p0	numeric.p2
numeric	index	0	1.000	972.344000	486.4327160	3.0	598.0000
numeric	neme	0	1.000	1.500000	0.5005008	1.0	1.0000
numeric	eletkora	0	1.000	52.522000	11.7257196	18.0	45.0000
numeric	gyermeke	0	1.000	1.750000	1.2371585	0.0	1.0000
numeric	isk	0	1.000	2.786000	0.8328212	1.0	2.0000
numeric	anyagi	0	1.000	3.100000	0.5644796	1.0	3.0000
numeric	p_elmeny_percent	0	1.000	58.460000	21.4183541	10.0	40.0000
numeric	testi_fi	3	0.994	4.164990	0.9572209	1.0	4.0000
numeric	alt_lelki	4	0.992	4.161290	1.0722362	1.0	4.0000
numeric	alt_eg_all	7	0.986	4.121704	1.1195761	1.0	3.0000
numeric	fizero	6	0.988	4.153846	1.1867317	1.0	3.0000
numeric	arcocska	0	1.000	2.570000	1.1435539	1.0	2.0000
numeric	aggodalo	0	1.000	2.772000	1.4325223	1.0	2.0000
numeric	ideges	0	1.000	2.490000	1.3746561	1.0	1.0000
numeric	feszult	0	1.000	2.610000	1.3992197	1.0	1.0000
numeric	nyugtala	0	1.000	2.406000	1.4578803	1.0	1.0000
numeric	diener1	0	1.000	5.514000	1.3286590	1.0	5.0000
numeric	diener2	0	1.000	5.228000	1.3219346	1.0	4.0000
numeric	diener3	0	1.000	5.574000	1.2310788	1.0	5.0000
numeric	diener4	0	1.000	5.418000	1.2611332	1.0	5.0000
numeric	diener5	0	1.000	5.756000	1.0727374	1.0	5.0000
numeric	diener6	0	1.000	5.768000	1.1245912	1.0	5.0000
numeric	diener7	0	1.000	5.620000	1.3738176	1.0	5.0000
numeric	diener8	0	1.000	5.532000	1.2148777	1.0	5.0000
numeric	jollet	0	1.000	4.488666	1.0610844	1.0	4.0000
numeric	savor	0	1.000	4.514667	1.0133860	1.0	4.0000
numeric	a_vhat	0	1.000	4.595600	0.9393964	1.0	4.0000
numeric	onreg	0	1.000	4.217332	1.2217368	1.0	3.3333
numeric	rezil	0	1.000	4.044667	1.0210453	1.0	3.3333
numeric	m_flow	0	1.000	4.773997	0.8909798	1.0	4.3333
numeric	g_jerz	0	1.000	4.152800	1.0313775	1.0	3.4000
numeric	g_jpszi	0	1.000	4.282000	0.9892843	1.0	3.7500
numeric	g_jszoc	0	1.000	3.949500	1.1789209	1.0	3.0000
numeric	g_jspir	0	1.000	4.245500	1.2003067	1.0	3.5000
numeric	g_jerzpsz	0	1.000	4.217400	0.9720234	1.1	3.6187
numeric	g_jspszoc	0	1.000	4.097500	1.1382101	1.0	3.3750
numeric	pik_mm	0	1.000	3.027334	0.6267487	1.0	2.6667
numeric	pik_av	0	1.000	3.175000	0.6232189	1.0	2.7500
numeric	pik_onr	0	1.000	3.064000	0.7912331	1.0	2.6667
numeric	pik_rez	0	1.000	3.150668	0.7424924	1.0	2.6667
numeric	p_poz_erz	0	1.000	21.742000	5.2751389	0.0	20.0000
numeric	p_elmely	0	1.000	22.360000	4.5585041	8.0	20.0000
numeric	p_poz_kapc	0	1.000	22.442000	5.5671017	0.0	19.0000
numeric	p_ert_cel	0	1.000	23.374000	4.6790931	0.0	21.0000
numeric	p_telj	0	1.000	22.716000	4.3420099	0.0	21.0000
numeric	p_boldog	0	1.000	7.404000	2.1217416	0.0	7.0000
numeric	p_egeszs	0	1.000	22.260000	5.2248443	0.0	19.0000
numeric	p_neg_erz	0	1.000	9.766000	5.7089779	0.0	5.0000
numeric	p_magany	0	1.000	3.162000	3.0565407	0.0	0.0000
numeric	perma	0	1.000	169.370000	31.2671131	22.0	155.0000

Transforming variables

```
processed <-
  raw %>%
  mutate(neme = case_when(neme == 1 ~ "ferfi",
                          neme == 2 ~ "no",
                          TRUE ~ NA_character_),
         isk = case_when(isk == 1 ~ "altalanos",
                         isk == 2 ~ "kozepiskola",
                         isk == 3 ~ "foiskola",
                         isk == 4 ~ "egyetem",
                         TRUE ~ NA_character_))
```

I will also save this version with the more informative labels for future use.

```
write_tsv(processed, "data/boldog_processed.tsv")
```

1. On which scales do people differ the most in their education level?

To investigate this question I decided to run several one-way ANOVA tests with the education level as the independent variable, and the different scales as the dependent variable. I will include only interval scales. Then, I will compare the differences with a standardized effect size measure, eta-square. The comparison with the biggest eta-square will be the answer.

Also, I will run the ANOVA tests in bulk in one iteration as I did in the second assignment. This way I can save some time typing, which is always nice. The results will be saved in a table format for an easy comparison. Also, a plot with the different standardized effect sizes would be helpful.

I am running type 2 ANOVA tests.

```
# Selecting and saving interval scales
interval_vars <-
  processed %>%
  select(26:50) %>%
  names()

# Converting participant id and isk to factor variable
isk_scale_data <-
  processed %>%
  mutate(isk = as.factor(isk),
         index = as.factor(index))

# Checking the order of levels for isk as a factor variable
levels(processed$isk)
```

```
## NULL
```

```r
# Function to run the one-way ANOVA
isk_anova <- function(data, dv_var) {
  eval(
    substitute(
      # The ANOVA
      ezANOVA(data = data, dv = dv_var, wid = index, between = isk, detailed = TRUE),
      list(dv_var = dv_var)
      )
    )
}

# Running the ANOVA tests
isk_scale_res <-
  tibble(
    variable = interval_vars,
    anova_res =  map(variable,
                     ~ isk_anova(data = isk_scale_data, dv_var = .x)),
    f_value = map_dbl(anova_res, ~ pluck(.x, "ANOVA", "F")),
    p_value = map_dbl(anova_res, ~ pluck(.x, "ANOVA", "p")),
    generalized_eta_square = map_dbl(anova_res, ~ pluck(.x, "ANOVA", "ges"))
    ) %>%
  select(-anova_res) %>%
  arrange(desc(generalized_eta_square))

isk_scale_res %>%
  kable(
    format = "latex",
    col.names = c("Variable", "F value", "p value", "Generalized Eta-squared"),
    align = c("l", "c", "c", "c"),
    caption = "One-way ANOVA Results for Each Scale Variable by Three Education Level"
  ) %>%
  row_spec(row = 0, align = "c") %>%
  kable_styling(full_width = TRUE) %>%
  add_footnote("Results are arranged in descending order by the size of the effec size.")
```

To answer the question I only consider significant effects. The highest difference between the educational levels were present on the *avhat* variable which is a subscale of the MET test (Vargha et al. (2020)).

Lets see the results on a plot as well.

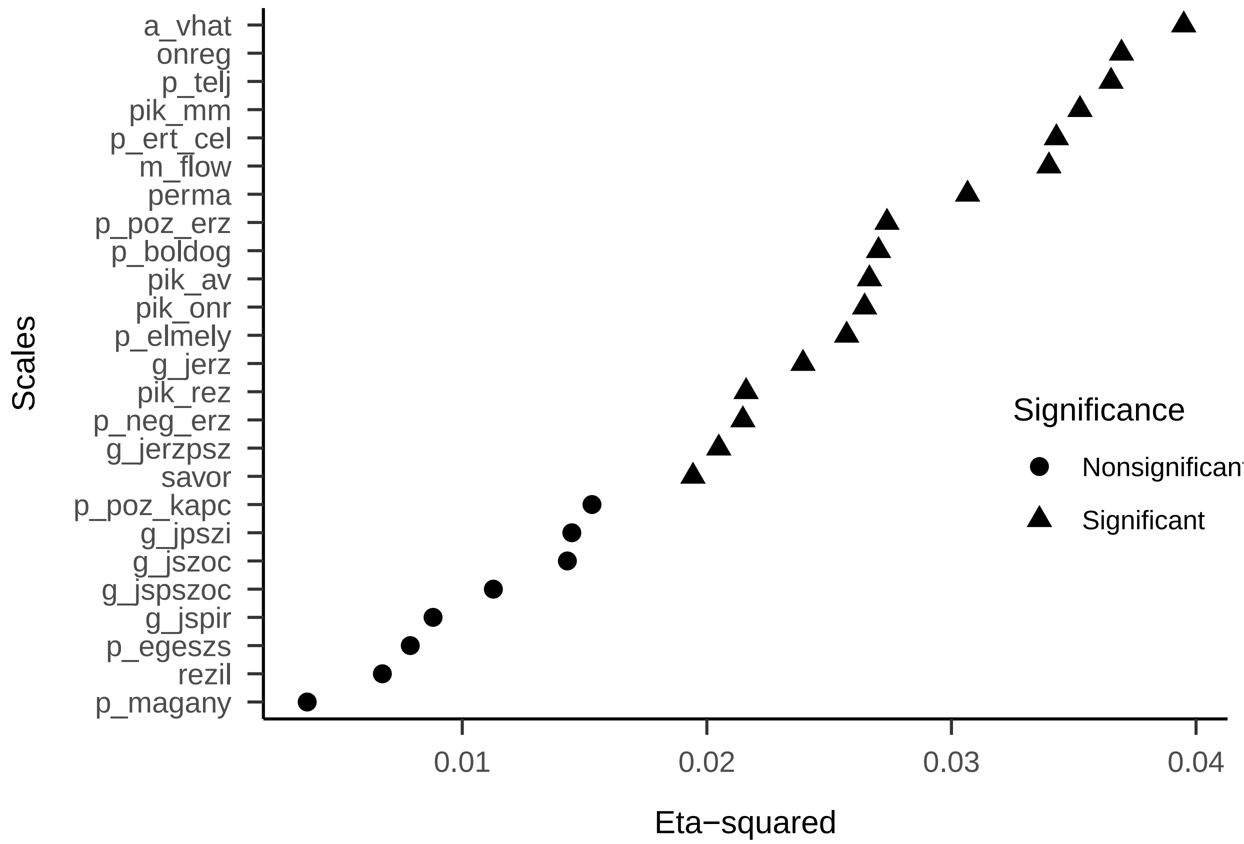
```r
isk_scale_res %>%
  mutate(significant = case_when(p_value <= 0.05 ~ "Significant",
                                 p_value > 0.05 ~ "Nonsignificant"),
         variable = fct_reorder(variable, generalized_eta_square)) %>%
  ggplot() +
  aes(x = variable, y = generalized_eta_square, shape = significant) +
  geom_point(size = 3) +
  coord_flip() +
  labs(
    y = "Eta-squared",
    x = "Scales",
    shape = "Significance"
  ) +
  papaja::theme_apa() +
```

Table 1: One-way ANOVA Results for Each Scale Variable by Three Education Level

Variable	F value	p value	Generalized Eta-squared
a__vhat	6.7985628	0.0001691	0.0394962
onreg	6.3436851	0.0003163	0.0369513
p__telj	6.2677818	0.0003511	0.0365253
pik__mm	6.0417033	0.0004792	0.0352543
p__ert__cel	5.8707103	0.0006062	0.0342907
m__flow	5.8165120	0.0006530	0.0339849
perma	5.2291004	0.0014630	0.0306580
p__poz__erz	4.6515348	0.0032269	0.0273644
p__boldog	4.5911821	0.0035044	0.0270190
pik__av	4.5260853	0.0038303	0.0266461
pik__onr	4.4917908	0.0040140	0.0264495
p__elmely	4.3638893	0.0047796	0.0257157
g__jerz	4.0535990	0.0072940	0.0239310
pik__rez	3.6508889	0.0125978	0.0216049
p__neg__erz	3.6280982	0.0129924	0.0214729
g__jerzpsz	3.4581228	0.0163474	0.0204875
savor	3.2765573	0.0208788	0.0194328
p__poz__kapc	2.5696220	0.0536523	0.0153042
g__jpszi	2.4290833	0.0645856	0.0144793
g__jszoc	2.3981773	0.0672662	0.0142977
g__jspszoc	1.8849284	0.1311545	0.0112723
g__jspir	1.4689255	0.2221533	0.0088064
p__egeszs	1.3123623	0.2696230	0.0078752
rezil	1.1208468	0.3401198	0.0067337
p__magany	0.6074972	0.6103794	0.0036609

[a] Results are arranged in descending order by the size of the effec size.

```
  theme(
    legend.position = c(.9, .35)
  )
```



2. On which scales do people differ more if we merge the data of the two lowest educational level?

```
# Merge the levels
isk_scale_merged_data <-
  processed %>%
  mutate(isk = case_when(isk == "altalanos" ~ "altalanos_es_kozepiskola",
                         isk == "kozepiskola" ~ "altalanos_es_kozepiskola",
                         TRUE ~ isk),
         isk = as.factor(isk))

# Running the ANOVA tests
isk_scale_merged_res <-
  tibble(
    variable = interval_vars,
    anova_res =  map(variable,
                     ~ isk_anova(data = isk_scale_merged_data, dv_var = .x)),
    f_value = map_dbl(anova_res, ~ pluck(.x, "ANOVA", "F")),
```

```r
    p_value = map_dbl(anova_res, ~ pluck(.x, "ANOVA", "p")),
    generalized_eta_square = map_dbl(anova_res, ~ pluck(.x, "ANOVA", "ges"))
    ) %>%
  select(-anova_res) %>%
  arrange(desc(generalized_eta_square))
```

```
## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()
```

```
## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()
```

```
## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()
```

```
## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Converting "index" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()
```

```r
isk_scale_merged_res %>%
  kable(
    format = "latex",
    col.names = c("Variable", "F value", "p value", "Generalized Eta-squared"),
    align = c("l", "c", "c", "c"),
    caption = "One-way ANOVA Results for Each Scale Variable by Education Level"
  ) %>%
  row_spec(row = 0, align = "c") %>%
  kable_styling(full_width = TRUE) %>%
  add_footnote("Results are arranged in descending order by the size of the effec size.")
```

Run a two-way ANOVA with educational level and gender on the scale type variables. Is there a scale where the interaction significant? How big is the eta-squared here?

Table 2: One-way ANOVA Results for Each Scale Variable by Education Level

Variable	F value	p value	Generalized Eta-squared
m__flow	8.7183317	0.0001899	0.0338947
onreg	8.7044261	0.0001925	0.0338424
p__ert__cel	8.3464017	0.0002721	0.0324957
a__vhat	8.0119699	0.0003762	0.0312343
p__telj	7.4216047	0.0006669	0.0289995
pik__mm	7.0311525	0.0009746	0.0275158
perma	6.7933451	0.0012283	0.0266100
pik__av	6.2774193	0.0020306	0.0246388
pik__onr	5.5604026	0.0040904	0.0218861
p__neg__erz	5.4508846	0.0045530	0.0214643
pik__rez	5.2219310	0.0056970	0.0205813
p__poz__erz	4.9598958	0.0073648	0.0195688
p__boldog	4.4416829	0.0122472	0.0175601
p__elmely	4.0589266	0.0178432	0.0160712
g__jerz	3.9933334	0.0190329	0.0158156
p__poz__kapc	3.7903888	0.0232425	0.0150239
savor	3.7593595	0.0239639	0.0149028
g__jerzpsz	3.2003591	0.0415887	0.0127150
g__jszoc	3.1986510	0.0416589	0.0127083
g__jspszoc	2.6622574	0.0707857	0.0105998
g__jspir	2.1824140	0.1138485	0.0087059
g__jpszi	2.0510451	0.1296876	0.0081861
rezil	1.6009931	0.2027345	0.0064014
p__egeszs	1.4480227	0.2360244	0.0057933
p__magany	0.9021394	0.4063641	0.0036172

[a] Results are arranged in descending order by the size of the effec size.

```r
# Converting neme to a factor variable
isk_scale_merged_data <-
  isk_scale_merged_data %>%
  mutate(neme = as.factor(neme),
         index = as.factor(index))

# Function to run the two-way ANOVA
isk_neme_anova <- function(data, dv_var) {
  eval(
    substitute(
      # The ANOVA
      ezANOVA(data = data, dv = dv_var, wid = index, between = .(isk, neme), detailed = TRUE),
      list(dv_var = dv_var)
      )
    )
}

# Running the ANOVA tests
isk_neme_scale_res <-
  tibble(
    variable = interval_vars,
    anova_res = map(variable,
                    ~ isk_neme_anova(data = isk_scale_merged_data, dv_var = .x)[["ANOVA"]]),
    interaction_res = map(anova_res, . %>% filter(Effect == "isk:neme")),
    f_value = map_dbl(interaction_res, . %>% pull(F)),
    p_value = map_dbl(interaction_res, . %>% pull(p)),
    generalized_eta_square = map_dbl(interaction_res, . %>% pull(ges))
    ) %>%
  select(-anova_res, -interaction_res) %>%
  arrange(desc(generalized_eta_square))
```

```
## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()
```

```
## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()
```

```
## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()
```

Table 3: Results of the Interaction Terms of the Two-way ANOVA for Each Scale Variable by Education Level

Variable	F value	p value	Generalized Eta-squared
p_neg_erz	6.5937844	0.0014925	0.0260014
p_elmely	6.1977004	0.0021958	0.0244777
m_flow	5.0882903	0.0064959	0.0201846
p_poz_erz	4.4520338	0.0121262	0.0177053
perma	4.2788559	0.0143757	0.0170283
p_telj	3.5018631	0.0308916	0.0139794
g_jpszi	3.4871377	0.0313434	0.0139214
g_jerzpsz	3.3440698	0.0360938	0.0133579
g_jspir	2.8624591	0.0580762	0.0114561
p_egeszs	2.8524759	0.0586522	0.0114166
g_jerz	2.7619516	0.0641441	0.0110583
rezil	2.1931892	0.1126455	0.0088012
g_jspszoc	1.6516000	0.1927998	0.0066422
p_ert_cel	1.5769930	0.2076336	0.0063441
pik_onr	1.2194553	0.2962786	0.0049128
a_vhat	1.1076563	0.3311512	0.0044644
savor	0.8243678	0.4391145	0.0033264
g_jszoc	0.8033598	0.4484060	0.0032419
pik_mm	0.7306656	0.4821080	0.0029494
p_boldog	0.7011803	0.4964925	0.0028308
p_poz_kapc	0.4606435	0.6311483	0.0018615
pik_rez	0.3260950	0.7218918	0.0013185
p_magany	0.2208301	0.8019320	0.0008933
pik_av	0.1930446	0.8245074	0.0007809
onreg	0.1149874	0.8914012	0.0004653

[a] Results are arranged in descending order by the size of the effec size.

```
isk_neme_scale_res %>%
  kable(
    format = "latex",
    col.names = c("Variable", "F value", "p value", "Generalized Eta-squared"),
    align = c("l", "c", "c", "c"),
    caption = "Results of the Interaction Terms of the Two-way ANOVA for Each Scale Variable by Educati
  ) %>%
  row_spec(row = 0, align = "c") %>%
  kable_styling(full_width = TRUE) %>%
  add_footnote("Results are arranged in descending order by the size of the effec size.")
```

4. Run the same as before but with a robust equivalent of the ANOVA.

As a robust non-parametric alternative for the two-way ANOVA I will use the medians instead of the means for the calculation.

```r
# Function to run the robust two-way ANOVA
isk_neme_robust_anova <- function(dv_var) {
  pbad2way(as.formula(paste(dv_var, "~ isk:neme")), data = isk_scale_merged_data, est = "median")
}

# Running the ANOVA tests
isk_neme_scale_robust_res <-
  tibble(
    variable = interval_vars,
    anova_res = map(variable,
                    ~ isk_neme_robust_anova(dv_var = .x)),
    p_value = map_dbl(anova_res, ~pluck(.x, "AB.p.value"))
    ) %>%
  select(-anova_res) %>%
  mutate(significance = case_when(p_value <= 0.05 ~ "Significant",
                                  p_value > 0.05 ~ "Nonsignificant"))
```

```
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
## Aggregation requires fun.aggregate: length used as default
```

```r
isk_neme_scale_robust_res %>%
  kable(
    format = "latex",
    col.names = c("Variable", "p value", "Significance"),
    align = c("l", "c", "c"),
    caption = "Results of the Interaction Terms of the Robust Two-way ANOVA for Each Scale Variable by 
  ) %>%
  row_spec(row = 0, align = "c") %>%
  kable_styling(full_width = TRUE)
```

Table 4: Results of the Interaction Terms of the Robust Two-way ANOVA for Each Scale Variable by Education Level

Variable	p value	Significance
savor	0.0333890	Significant
a__vhat	0.0434057	Significant
onreg	0.0400668	Significant
rezil	0.0450751	Significant
m__flow	0.0467446	Significant
g__jerz	0.0450751	Significant
g__jpszi	0.0467446	Significant
g__jszoc	0.0484140	Significant
g__jspir	0.0367279	Significant
g__jerzpsz	0.0584307	Nonsignificant
g__jspszoc	0.0383973	Significant
pik__mm	0.0317195	Significant
pik__av	0.0434057	Significant
pik__onr	0.0367279	Significant
pik__rez	0.0434057	Significant
p__poz__erz	0.0467446	Significant
p__elmely	0.0317195	Significant
p__poz__kapc	0.0617696	Nonsignificant
p__ert__cel	0.0434057	Significant
p__telj	0.0617696	Nonsignificant
p__boldog	0.0567613	Nonsignificant
p__egeszs	0.0584307	Nonsignificant
p__neg__erz	0.0467446	Significant
p__magany	0.0417362	Significant
perma	0.0500835	Nonsignificant

5. Is there a difference between the means of the subscales of the Globalis Jollet scale?

To answer this question first I select the needed variables and trasnform them to long format.

```
global_comparison_data <-
  processed %>%
  select(index, g_jpszi, g_jszoc, g_jspir, g_jerzpsz) %>%
  gather(key = "subscale", value = "value", -index)

# Running the ANOVA
ezANOVA(dv = value, wid = index, within = subscale, data = global_comparison_data)
```

```
## Warning: Converting "index" to factor for ANOVA.
```

```
## Warning: Converting "subscale" to factor for ANOVA.
```

```
## $ANOVA
##     Effect DFn  DFd        F            p p<.05        ges
## 2 subscale   3 1497 39.99037 7.385872e-25     * 0.01434657
##
## $`Mauchly's Test for Sphericity`
##     Effect         W             p p<.05
## 2 subscale 0.2195681 7.917897e-161     *
##
## $`Sphericity Corrections`
##     Effect       GGe        p[GG] p[GG]<.05       HFe        p[HF] p[HF]<.05
## 2 subscale 0.6126106 3.162009e-16         * 0.6147664 2.830209e-16         *
```

According to the one-way repeated measures ANOVA there is a significant difference between the subscales.

6. Two-way mixed ANOVA with the four subscales and gender.

Again, we start with the data transformation.

```
global_comparison_gender_data <-
  processed %>%
  select(index, neme, g_jpszi, g_jszoc, g_jspir, g_jerzpsz) %>%
  gather(key = "subscale", value = "value", -index, -neme)

# Running the ANOVA
ezANOVA(dv = value, wid = index, within = subscale, between = neme, data = global_comparison_gender_data
```

```
## Warning: Converting "index" to factor for ANOVA.
```

```
## Warning: Converting "subscale" to factor for ANOVA.
```

```
## Warning: Converting "neme" to factor for ANOVA.
```

```
## $ANOVA
##          Effect DFn  DFd         F            p p<.05         ges
## 2          neme   1  498  3.578935 5.909653e-02       0.005846805
## 3      subscale   3 1494 40.190380 5.631951e-25     * 0.014447752
## 4 neme:subscale   3 1494  3.495712 1.507624e-02     * 0.001273447
##
## $`Mauchly's Test for Sphericity`
##          Effect        W            p p<.05
## 3      subscale 0.220302 3.845292e-160     *
## 4 neme:subscale 0.220302 3.845292e-160     *
##
## $`Sphericity Corrections`
##          Effect       GGe        p[GG] p[GG]<.05       HFe        p[HF]
## 3      subscale 0.6124615 2.694281e-16         * 0.6146203 2.409807e-16
## 4 neme:subscale 0.6124615 3.448531e-02         * 0.6146203 3.432600e-02
##   p[HF]<.05
## 3         *
## 4         *
```

According to the results there is a significant interaction between the subscales and the gender. Also, the subscales have a significant main effect, but not gender.

References

Vargha, András, Virág Zábó, Regina Török, and Attila Oláh. 2020. "A jóllét és a Mentális Egészség mérése: A Mentális Egészség Teszt." *Mentálhigiéné és Pszichoszomatika* 21 (3): 281–322.