

Solution for the assignment of the sixth class

Kovacs Marton

9/29/2021

Importing data

```
processed <- read_tsv("data/boldog_processed.tsv")

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

Data exploration

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

skim_type	skim_variable	n_missing	complete_rate	character.min	character.max	character.empty	char
character	neme	0	1.000	2	5	0	
character	isk	0	1.000	7	11	0	
numeric	index	0	1.000	NA	NA	NA	
numeric	eletkora	0	1.000	NA	NA	NA	
numeric	gyermeke	0	1.000	NA	NA	NA	
numeric	anyagi	0	1.000	NA	NA	NA	
numeric	p_elmeny_percent	0	1.000	NA	NA	NA	
numeric	testi_fi	3	0.994	NA	NA	NA	
numeric	alt_lelki	4	0.992	NA	NA	NA	
numeric	alt_eg_all	7	0.986	NA	NA	NA	
numeric	fizero	6	0.988	NA	NA	NA	
numeric	arcocska	0	1.000	NA	NA	NA	
numeric	aggodalo	0	1.000	NA	NA	NA	
numeric	ideges	0	1.000	NA	NA	NA	
numeric	feszult	0	1.000	NA	NA	NA	
numeric	nyugtala	0	1.000	NA	NA	NA	
numeric	diener1	0	1.000	NA	NA	NA	
numeric	diener2	0	1.000	NA	NA	NA	
numeric	diener3	0	1.000	NA	NA	NA	
numeric	diener4	0	1.000	NA	NA	NA	
numeric	diener5	0	1.000	NA	NA	NA	
numeric	diener6	0	1.000	NA	NA	NA	
numeric	diener7	0	1.000	NA	NA	NA	
numeric	diener8	0	1.000	NA	NA	NA	
numeric	jollet	0	1.000	NA	NA	NA	
numeric	savor	0	1.000	NA	NA	NA	
numeric	a_vhat	0	1.000	NA	NA	NA	
numeric	onreg	0	1.000	NA	NA	NA	
numeric	rezil	0	1.000	NA	NA	NA	
numeric	m_flow	0	1.000	NA	NA	NA	
numeric	g_jerz	0	1.000	NA	NA	NA	
numeric	g_jpszi	0	1.000	NA	NA	NA	
numeric	g_jszoc	0	1.000	NA	NA	NA	
numeric	g_jspir	0	1.000	NA	NA	NA	
numeric	g_jerzpsz	0	1.000	NA	NA	NA	
numeric	g_jspeszoc	0	1.000	NA	NA	NA	
numeric	pik_mm	0	1.000	NA	NA	NA	
numeric	pik_av	0	1.000	NA	NA	NA	
numeric	pik_onr	0	1.000	NA	NA	NA	
numeric	pik_rez	0	1.000	NA	NA	NA	
numeric	p_poz_erz	0	1.000	NA	NA	NA	
numeric	p_elmely	0	1.000	NA	NA	NA	
numeric	p_poz_kapc	0	1.000	NA	NA	NA	
numeric	p_ert_cel	0	1.000	NA	NA	NA	
numeric	p_telj	0	1.000	NA	NA	NA	
numeric	p_boldog	0	1.000	NA	NA	NA	
numeric	p_egeszs	0	1.000	NA	NA	NA	
numeric	p_neg_erz	0	1.000	NA	NA	NA	
numeric	p_magany	0	1.000	NA	NA	NA	
numeric	perma	0	1.000	NA	NA	NA	

1. What are the correlations and rank correlations between Globalis Jollet Scale and MET?

First I create a table for all the combinations between the variables.

```
global <- dplyr::select(processed, 31:36) %>% names()
met <- dplyr::select(processed, 25:29) %>% names
correlation_res <- tidyr::crossing(global, met)
```

Then I calculate the parametric correlation for each combination with a Pearson correlation, and the rank correlation with the Spearman method.

```
correlation_res <-
  correlation_res %>%
  mutate(
    pearson_res = map2(global, met,
      ~ my_cor(
        data = processed,
        x = .x,
        y = .y,
        method = "pearson")),
    pearson_r = map_dbl(pearson_res, ~ pluck(.x, "estimate", "cor")),
    pearson_p = map_dbl(pearson_res, ~ pluck(.x, "p.value")),
    spearman_res = map2(global, met,
      ~ my_cor(
        data = processed,
        x = .x,
        y = .y,
        method = "spearman")),
    spearman_r = map_dbl(spearman_res, ~ pluck(.x, "estimate", "rho")),
    spearman_p = map_dbl(spearman_res, ~ pluck(.x, "p.value")),
  )
```

```
## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties
```

```
## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties
```

```
## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties
```

```
## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties
```

```
## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties
```

```
## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties
```

```
## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
```



```
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties

## Warning in cor.test.default(data[[x]], data[[y]], method = method): Cannot
## compute exact p-value with ties
```

```
correlation_res %>%
  dplyr::select(-pearson_res, -spearman_res) %>%
  arrange(desc(pearson_r))
```

```
## # A tibble: 30 x 6
##   global    met  pearson_r pearson_p spearman_r spearman_p
##   <chr>    <chr>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 g_jerz   jollet    0.804 1.25e-114    0.781 6.27e-104
## 2 g_jerzpsz jollet    0.794 1.31e-109    0.761 8.53e- 96
## 3 g_jpszi   jollet    0.721 1.76e- 81    0.683 4.66e- 70
## 4 g_jpszi   a_vhat    0.705 2.15e- 76    0.658 2.80e- 63
## 5 g_jerzpsz a_vhat    0.696 9.46e- 74    0.636 5.60e- 58
## 6 g_jerz    a_vhat    0.636 4.68e- 58    0.573 4.73e- 45
## 7 g_jspszoc a_vhat    0.613 6.24e- 53    0.583 7.02e- 47
## 8 g_jszoc   jollet    0.608 6.90e- 52    0.574 3.07e- 45
## 9 g_jspszoc jollet    0.607 1.35e- 51    0.570 2.02e- 44
## 10 g_jszoc   a_vhat    0.594 4.73e- 49    0.565 2.00e- 43
## # ... with 20 more rows
```

The highest correlation was between *gjerz* and the *jollet* scales.

2. Find scales that are in a non-linear relationship with any of the subscales of MET.

3. What is the proportion of the negative and positive relationships between the resilience scale and the total score of the Diener scale.

I am not sure that I understand the question correctly, but I will calculate the correlation between the *resil* variable and each item of the Diener flourishing scale. Then, I will look at the proportion of the positive and negative correlations based on the Pearson correlation coefficient.

```

diener <- dplyr::select(processed, contains("diener")) %>% names()
rezil <- c("rezil")

diener_rezil_res <- tidyr::crossing(rezil, diener)

diener_rezil_res <-
  diener_rezil_res %>%
  mutate(
    res = map2(rezil, diener,
               ~ my_cor(
                 data = processed,
                 x = .x,
                 y = .y,
                 method = "pearson")),
    r = map_dbl(res, ~ pluck(.x, "estimate", "cor")),
    r = round(r, 2),
    valence = case_when(r > 0 ~ "positive",
                        r < 0 ~ "negative")
  ) %>%
  dplyr::select(-res) %>%
  arrange(desc(r))

diener_rezil_res %>%
  count(valence) %>%
  mutate(prop = n / 8 * 100)

```

```

## # A tibble: 1 x 3
##   valence      n prop
##   <chr>    <int> <dbl>
## 1 positive      8  100

```

4. How big is the Cronbachs alpha for the total score of the Diener scale?

```

diener_data <-
  processed %>%
  dplyr::select(contains("diener"))

cronbach.alpha(diener_data, CI = TRUE)

##
## Cronbach's alpha for the 'diener_data' data-set
##
## Items: 8
## Sample units: 500
## alpha: 0.94
##
## Bootstrap 95% CI based on 1000 samples
## 2.5% 97.5%
## 0.926 0.950

```

5. Same for each gender group.

```
diener_male_data <-  
  processed %>%  
  filter(neme == "ferfi") %>%  
  dplyr::select(contains("diener"))  
  
cronbach.alpha(diener_male_data, CI = TRUE)  
  
##  
## Cronbach's alpha for the 'diener_male_data' data-set  
##  
## Items: 8  
## Sample units: 250  
## alpha: 0.942  
##  
## Bootstrap 95% CI based on 1000 samples  
## 2.5% 97.5%  
## 0.921 0.957
```

```
diener_female_data <-  
  processed %>%  
  filter(neme == "no") %>%  
  dplyr::select(contains("diener"))  
  
cronbach.alpha(diener_female_data, CI = TRUE)
```

```
##  
## Cronbach's alpha for the 'diener_female_data' data-set  
##  
## Items: 8  
## Sample units: 250  
## alpha: 0.937  
##  
## Bootstrap 95% CI based on 1000 samples  
## 2.5% 97.5%  
## 0.915 0.951
```

6. How big is the Mcdonalds' omega for the Diener flourishing scale?

```
psych::omega(diener_data, nfactors = 1)
```

```
## Omega_h for 1 factor is not meaningful, just omega_t
```

```
## Warning in schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :  
## Omega_h and Omega_asymptotic are not meaningful with one factor
```

```

## Warning in cov2cor(t(w) %*% r %*% w): diag(.) had 0 or NA entries; non-finite
## result is doubtful

## Omega
## Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
##   digits = digits, title = title, sl = sl, labels = labels,
##   plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
##   covar = covar)
## Alpha:                0.94
## G.6:                  0.94
## Omega Hierarchical:   0.94
## Omega H asymptotic:   1
## Omega Total           0.94
##
## Schmid Leiman Factor loadings greater than 0.2
##      g  F1*   h2   u2 p2
## diener1 0.83    0.69 0.31 1
## diener2 0.79    0.63 0.37 1
## diener3 0.84    0.70 0.30 1
## diener4 0.78    0.60 0.40 1
## diener5 0.78    0.61 0.39 1
## diener6 0.89    0.79 0.21 1
## diener7 0.82    0.67 0.33 1
## diener8 0.81    0.66 0.34 1
##
## With eigenvalues of:
##      g  F1*
## 5.4 0.0
##
## general/max  Inf  max/min =  NaN
## mean percent general = 1 with sd = 0 and cv of 0
## Explained Common Variance of the general factor = 1
##
## The degrees of freedom are 20 and the fit is 0.32
## The number of observations was 500 with Chi Square = 156.79 with prob < 3.1e-23
## The root mean square of the residuals is 0.03
## The df corrected root mean square of the residuals is 0.04
## RMSEA index = 0.117 and the 10 % confidence intervals are 0.1 0.134
## BIC = 32.5
##
## Compare this with the adequacy of just a general factor and no group factors
## The degrees of freedom for just the general factor are 20 and the fit is 0.32
## The number of observations was 500 with Chi Square = 156.79 with prob < 3.1e-23
## The root mean square of the residuals is 0.03
## The df corrected root mean square of the residuals is 0.04
##
## RMSEA index = 0.117 and the 10 % confidence intervals are 0.1 0.134
## BIC = 32.5
##
## Measures of factor score adequacy
##
##      g  F1*
## Correlation of scores with factors    0.97  0
## Multiple R square of scores with factors    0.95  0
## Minimum correlation of factor score estimates 0.89 -1

```



```
##
## Total, General and Subset omega for each subset
##
##              g    F1*
## Omega total for total scores and subscales    0.94 0.94
## Omega general for total scores and subscales  0.94 0.94
## Omega group for total scores and subscales    0.00 0.00
```

7. How big is the Cronbachs alpha and Mcdonalds' omega for the scale consisting of the Testi_fi, Áltlelki, Áltegáll, Fizerö, Arcocska, Aggodalo, Ideges, Feszült, Nyugtalan items?

Some of the items are on reversed scales so we have to reverse them. To run the analyses I also have to drop missing items.

```
scale_data <-
  processed %>%
  dplyr::select(8:16) %>%
  filter_all(all_vars(!is.na(.)))

# Looking at the maximum values for each scale
scale_data %>%
  summarise_all(max, na.rm = TRUE)
```

```
## # A tibble: 1 x 9
##   testi_fi alt_lelki alt_eg_all fizero arcocska aggodalo ideges feszult nyugtala
##   <dbl>    <dbl>    <dbl>  <dbl>    <dbl>    <dbl>  <dbl>  <dbl>    <dbl>
## 1      6      6      6      6      7      6      6      6      6
```

```
# I only suspect that if these items belong to the same scale the maximum value for all of them is 7
reverse <- function(x) {
  7 - x
}

scale_data <-
  scale_data %>%
  mutate_at(vars(arcocska, aggodalo, ideges, feszult, nyugtala), reverse)
```

We can now calculate the Cronbachs alpha and Mcdonalds' omega. For calculating the omega, I propose that the items are grouped into one factor.

```
cronbach.alpha(scale_data, CI = TRUE)
```

```
##
## Cronbach's alpha for the 'scale_data' data-set
##
## Items: 9
## Sample units: 483
## alpha: 0.879
##
## Bootstrap 95% CI based on 1000 samples
```

```
## 2.5% 97.5%
## 0.859 0.895
```

```
psych::omega(scale_data, nfactors = 1)
```

```
## Omega_h for 1 factor is not meaningful, just omega_t
```

```
## Warning in schmid(m, nfactors, fm, digits, rotate = rotate, n.obs = n.obs, :
## Omega_h and Omega_asymptotic are not meaningful with one factor
```

```
## Omega
## Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
##   digits = digits, title = title, sl = sl, labels = labels,
##   plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
##   covar = covar)
## Alpha:                0.88
## G.6:                  0.92
## Omega Hierarchical:   0.88
## Omega H asymptotic:   0.99
## Omega Total           0.88
##
## Schmid Leiman Factor loadings greater than 0.2
##      g F1* h2 u2 p2
## testi_fi 0.54 0.29 0.71 1
## alt_lelki 0.68 0.46 0.54 1
## alt_eg_all 0.58 0.33 0.67 1
## fizero 0.54 0.30 0.70 1
## arcocska 0.72 0.52 0.48 1
## aggodalo 0.72 0.52 0.48 1
## ideges 0.79 0.62 0.38 1
## feszult 0.81 0.65 0.35 1
## nyugtala 0.65 0.42 0.58 1
##
## With eigenvalues of:
##      g F1*
## 4.1 0.0
##
## general/max 7.387003e+16 max/min = 1
## mean percent general = 1 with sd = 0 and cv of 0
## Explained Common Variance of the general factor = 1
##
## The degrees of freedom are 27 and the fit is 3.03
## The number of observations was 483 with Chi Square = 1448.31 with prob < 1.1e-288
## The root mean square of the residuals is 0.2
## The df corrected root mean square of the residuals is 0.23
## RMSEA index = 0.33 and the 10 % confidence intervals are 0.316 0.345
## BIC = 1281.45
##
## Compare this with the adequacy of just a general factor and no group factors
## The degrees of freedom for just the general factor are 27 and the fit is 3.03
## The number of observations was 483 with Chi Square = 1448.31 with prob < 1.1e-288
## The root mean square of the residuals is 0.2
## The df corrected root mean square of the residuals is 0.23
```

```

##
## RMSEA index = 0.33 and the 10 % confidence intervals are 0.316 0.345
## BIC = 1281.45
##
## Measures of factor score adequacy
##
## Correlation of scores with factors      g F1*
## Multiple R square of scores with factors 0.95 0
## Minimum correlation of factor score estimates 0.79 -1
##
## Total, General and Subset omega for each subset
##
## Omega total for total scores and subscales 0.88 0.88
## Omega general for total scores and subscales 0.88 0.88
## Omega group for total scores and subscales 0.00 0.00

```