Weight Bias in Health Sciences

Calyope Ortega

2024-10-06

weight\_ed <- read.csv('/Users/calyopeortega/Documents/GA Files/Survey of weight-based curriculum in healthcare education\_September 18, 2023\_09.39.csv')

## Checking Reliability of Items through Alpha—-

#Checking alpha for the items in the UMBFAT scale  
psych:: alpha(weight\_ed[,c('UMBFAT1','UMBFAT2','UMBFAT5','UMBFAT7','UMB.FAT.8','UMB.FAT.11','UMB.FAT.12','UMB.FAT.13',  
 'UMB.FAT.14','UMB.FAT.15','UMB.FAT.16','UMB.FAT.17','UMB.FAT.18','UMB.FAT.19','UMB.FAT.20')])

## Reliability analysis   
## Call: psych::alpha(x = weight\_ed[, c("UMBFAT1", "UMBFAT2", "UMBFAT5",   
## "UMBFAT7", "UMB.FAT.8", "UMB.FAT.11", "UMB.FAT.12", "UMB.FAT.13",   
## "UMB.FAT.14", "UMB.FAT.15", "UMB.FAT.16", "UMB.FAT.17", "UMB.FAT.18",   
## "UMB.FAT.19", "UMB.FAT.20")])  
  
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r  
## 0.5 0.45 0.75 0.052 0.82 0.068 4 0.46 -0.18  
   
## 95% confidence boundaries   
## lower alpha upper  
## Feldt 0.36 0.5 0.62  
## Duhachek 0.36 0.5 0.63

###Creating a composite variable in R ##Reverse Scoring of UMBFAT: 5,7,8,12,15,16,19. —-

#Reverse Scoring Necessary Items  
weight\_ed$UMBFAT5R <- 8 - weight\_ed$UMBFAT5  
#Double checking recode  
table(weight\_ed$UMBFAT5,weight\_ed$UMBFAT5R)

##   
## 1 2 3 4 5  
## 3 0 0 0 0 4  
## 4 0 0 0 13 0  
## 5 0 0 3 0 0  
## 6 0 16 0 0 0  
## 7 86 0 0 0 0

weight\_ed$UMBFAT7R <- 8 - weight\_ed$UMBFAT7  
table(weight\_ed$UMBFAT7,weight\_ed$UMBFAT7R)

weight\_ed$UMBFAT8R <- 8 - weight\_ed$UMB.FAT.8  
table(weight\_ed$UMB.FAT.8,weight\_ed$UMBFAT8R)

weight\_ed$UMBFAT12R <- 8 - weight\_ed$UMB.FAT.12  
table(weight\_ed$UMB.FAT.12,weight\_ed$UMBFAT12R)

weight\_ed$UMBFAT15R <- 8 - weight\_ed$UMB.FAT.15  
table(weight\_ed$UMB.FAT.15,weight\_ed$UMBFAT15R)

weight\_ed$UMBFAT16R <- 8 - weight\_ed$UMB.FAT.16  
table(weight\_ed$UMB.FAT.16,weight\_ed$UMBFAT16R)

weight\_ed$UMBFAT19R <- 8 - weight\_ed$UMB.FAT.19  
table(weight\_ed$UMB.FAT.19,weight\_ed$UMBFAT19R)

#Alpha for UMBFAT with Reverse Scoring  
psych:: alpha(weight\_ed[,c('UMBFAT1','UMBFAT2','UMBFAT5R','UMBFAT7R','UMBFAT8R','UMB.FAT.11','UMBFAT12R',  
 'UMB.FAT.13','UMB.FAT.14','UMBFAT15R','UMBFAT16R','UMB.FAT.17',  
 'UMB.FAT.18','UMBFAT19R','UMB.FAT.20')], na.rm=TRUE)

## Reliability analysis   
## Call: psych::alpha(x = weight\_ed[, c("UMBFAT1", "UMBFAT2", "UMBFAT5R",   
## "UMBFAT7R", "UMBFAT8R", "UMB.FAT.11", "UMBFAT12R", "UMB.FAT.13",   
## "UMB.FAT.14", "UMBFAT15R", "UMBFAT16R", "UMB.FAT.17", "UMB.FAT.18",   
## "UMBFAT19R", "UMB.FAT.20")], na.rm = TRUE)  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r  
## 0.9 0.9 0.93 0.38 9 0.013 1.7 0.85 0.34  
  
## 95% confidence boundaries   
## lower alpha upper  
## Feldt 0.87 0.9 0.92  
## Duhachek 0.87 0.9 0.93

**As can be seen, after reverse scoring the correct items, the alpha level is high.**

**This helps me know that the items are measuring the same construct of weight prejudice**

**A high alpha level suggests that I can make this a composite variable.**

**###Overall scale score variable**

#Composite Variable with Reverse Scoring  
weight\_ed$UMBFAT\_SCALE <- rowMeans(weight\_ed[,c('UMBFAT1','UMBFAT2','UMBFAT5R','UMBFAT7R','UMBFAT8R','UMB.FAT.11',  
 'UMBFAT12R','UMB.FAT.13','UMB.FAT.14','UMBFAT15R','UMBFAT16R','UMB.FAT.17',  
 'UMB.FAT.18','UMBFAT19R','UMB.FAT.20')], na.rm=TRUE)

**# Weight-based stereotypes were measured using the BAOP**

weight\_ed$BAOP2R <- 7 - weight\_ed$BAOP2  
table(weight\_ed$BAOP2,weight\_ed$BAOP2R)

##   
## 1 2 3  
## 4 0 0 52  
## 5 0 44 0  
## 6 26 0 0

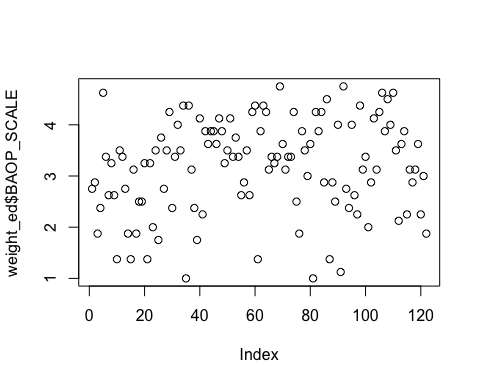
weight\_ed$BAOP7R <- 7 - weight\_ed$BAOP7  
table(weight\_ed$BAOP7,weight\_ed$BAOP7R)

##   
## 1 2 3  
## 4 0 0 42  
## 5 0 39 0  
## 6 41 0 0

psych:: alpha(weight\_ed[,c('BAOP1','BAOP2R','BAOP3','BAOP4','BAOP5','BAOP6','BAOP7R','BAOP8')])

## Reliability analysis   
## Call: psych::alpha(x = weight\_ed[, c("BAOP1", "BAOP2R", "BAOP3", "BAOP4",   
## "BAOP5", "BAOP6", "BAOP7R", "BAOP8")])  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r  
## 0.83 0.83 0.84 0.37 4.8 0.021 3.2 0.91 0.36  
##   
## 95% confidence boundaries   
## lower alpha upper  
## Feldt 0.78 0.83 0.87  
## Duhachek 0.79 0.83 0.87

##Creating scale score for BAOP items ----  
weight\_ed$BAOP\_SCALE <-   
 rowMeans(weight\_ed[,c('BAOP1','BAOP2R','BAOP3','BAOP4','BAOP5','BAOP6','BAOP7R','BAOP8')],   
 na.rm=TRUE)  
  
##Checking my scale score for accuracy ----  
plot(weight\_ed$BAOP\_SCALE)



#Aplhas for Curriculum Variables: 1-3; 4-9; every item together  
  
psych:: alpha(weight\_ed[,c("curr1","curr2","curr3")], na.rm = TRUE)

## Reliability analysis   
## Call: psych::alpha(x = weight\_ed[, c("curr1", "curr2", "curr3")], na.rm = TRUE)  
  
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r  
## 0.9 0.9 0.85 0.74 8.7 0.016 3.8 1.1 0.74  
##   
## 95% confidence boundaries   
## lower alpha upper  
## Feldt 0.86 0.9 0.92  
## Duhachek 0.86 0.9 0.93  
  
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se var.r med.r  
## curr1 0.85 0.85 0.74 0.74 5.7 0.027 NA 0.74  
## curr2 0.85 0.85 0.73 0.73 5.5 0.028 NA 0.73  
## curr3 0.86 0.86 0.76 0.76 6.2 0.025 NA 0.76  
  
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## curr1 122 0.91 0.91 0.84 0.80 4.1 1.2  
## curr2 122 0.92 0.91 0.85 0.80 3.8 1.3  
## curr3 121 0.91 0.91 0.83 0.79 3.7 1.2  
  
## Non missing response frequency for each item  
## 1 2 3 4 5 miss  
## curr1 0.04 0.11 0.06 0.31 0.48 0.00  
## curr2 0.09 0.11 0.07 0.43 0.31 0.00  
## curr3 0.11 0.07 0.16 0.40 0.27 0.01

psych:: alpha(weight\_ed[,c("curr4","curr5","curr6","curr7","curr8","curr9")], na.rm = TRUE)

## Reliability analysis   
## Call: psych::alpha(x = weight\_ed[, c("curr4", "curr5", "curr6", "curr7",   
## "curr8", "curr9")], na.rm = TRUE)  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r  
## 0.86 0.86 0.88 0.5 6.1 0.02 4.2 0.87 0.48  
##   
## 95% confidence boundaries   
## lower alpha upper  
## Feldt 0.81 0.86 0.89  
## Duhachek 0.82 0.86 0.90  
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se var.r med.r  
## curr4 0.86 0.86 0.86 0.55 6.1 0.018 0.030 0.58  
## curr5 0.82 0.82 0.83 0.48 4.6 0.025 0.020 0.47  
## curr6 0.82 0.82 0.83 0.47 4.5 0.026 0.017 0.47  
## curr7 0.82 0.82 0.85 0.48 4.7 0.025 0.038 0.44  
## curr8 0.86 0.86 0.87 0.56 6.2 0.021 0.029 0.58  
## curr9 0.81 0.82 0.83 0.47 4.5 0.026 0.025 0.46  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## curr4 122 0.67 0.66 0.58 0.50 3.8 1.27  
## curr5 122 0.82 0.82 0.80 0.73 4.3 1.11  
## curr6 122 0.83 0.83 0.81 0.74 4.2 1.21  
## curr7 122 0.81 0.81 0.75 0.70 3.9 1.21  
## curr8 122 0.61 0.65 0.54 0.50 4.5 0.83  
## curr9 122 0.84 0.83 0.81 0.75 4.2 1.16  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 miss  
## curr4 0.08 0.10 0.13 0.32 0.37 0  
## curr5 0.04 0.07 0.06 0.18 0.66 0  
## curr6 0.06 0.08 0.05 0.20 0.61 0  
## curr7 0.07 0.06 0.17 0.30 0.39 0  
## curr8 0.02 0.00 0.07 0.28 0.63 0  
## curr9 0.06 0.05 0.09 0.23 0.57 0

psych:: alpha(weight\_ed[,c("curr1","curr2","curr3","curr4","curr5","curr6","curr7","curr8","curr9")], na.rm = TRUE)

## Warning in psych::alpha(weight\_ed[, c("curr1", "curr2", "curr3", "curr4", : Some items were negatively correlated with the first principal component and probably   
## should be reversed.   
## To do this, run the function again with the 'check.keys=TRUE' option

## Some items ( curr1 curr2 curr3 ) were negatively correlated with the first principal component and   
## probably should be reversed.   
## To do this, run the function again with the 'check.keys=TRUE' option

##   
## Reliability analysis   
## Call: psych::alpha(x = weight\_ed[, c("curr1", "curr2", "curr3", "curr4",   
## "curr5", "curr6", "curr7", "curr8", "curr9")], na.rm = TRUE)  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r  
## 0.69 0.7 0.85 0.21 2.3 0.045 4.1 0.63 0.17  
##   
## 95% confidence boundaries   
## lower alpha upper  
## Feldt 0.6 0.69 0.77  
## Duhachek 0.6 0.69 0.78  
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se var.r med.r  
## curr1 0.70 0.72 0.84 0.24 2.5 0.042 0.13 0.315  
## curr2 0.70 0.71 0.84 0.24 2.5 0.041 0.13 0.315  
## curr3 0.71 0.72 0.85 0.24 2.6 0.040 0.12 0.315  
## curr4 0.69 0.70 0.84 0.22 2.3 0.046 0.14 0.020  
## curr5 0.60 0.61 0.80 0.17 1.6 0.059 0.14 -0.063  
## curr6 0.62 0.63 0.80 0.18 1.7 0.057 0.14 0.020  
## curr7 0.65 0.66 0.83 0.19 1.9 0.052 0.13 0.020  
## curr8 0.67 0.67 0.84 0.21 2.1 0.050 0.16 0.020  
## curr9 0.61 0.63 0.80 0.17 1.7 0.057 0.14 0.020  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## curr1 122 0.38 0.35 0.30 0.18 4.1 1.16  
## curr2 122 0.40 0.37 0.33 0.19 3.8 1.25  
## curr3 121 0.35 0.33 0.27 0.14 3.7 1.25  
## curr4 122 0.44 0.45 0.37 0.23 3.8 1.27  
## curr5 122 0.77 0.77 0.77 0.66 4.3 1.11  
## curr6 122 0.71 0.72 0.72 0.58 4.2 1.21  
## curr7 122 0.60 0.62 0.57 0.44 3.9 1.21  
## curr8 122 0.50 0.55 0.45 0.38 4.5 0.83  
## curr9 122 0.73 0.73 0.72 0.60 4.2 1.16  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 miss  
## curr1 0.04 0.11 0.06 0.31 0.48 0.00  
## curr2 0.09 0.11 0.07 0.43 0.31 0.00  
## curr3 0.11 0.07 0.16 0.40 0.27 0.01  
## curr4 0.08 0.10 0.13 0.32 0.37 0.00  
## curr5 0.04 0.07 0.06 0.18 0.66 0.00  
## curr6 0.06 0.08 0.05 0.20 0.61 0.00  
## curr7 0.07 0.06 0.17 0.30 0.39 0.00  
## curr8 0.02 0.00 0.07 0.28 0.63 0.00  
## curr9 0.06 0.05 0.09 0.23 0.57 0.00

#Reverse Scoring Curr Variables 1-3  
weight\_ed$curr1R <- 6 - weight\_ed$curr1  
table(weight\_ed$curr1,weight\_ed$curr1R)

##   
## 1 2 3 4 5  
## 1 0 0 0 0 5  
## 2 0 0 0 13 0  
## 3 0 0 7 0 0  
## 4 0 38 0 0 0  
## 5 59 0 0 0 0

weight\_ed$curr2R <- 6 - weight\_ed$curr2  
table(weight\_ed$curr2,weight\_ed$curr2R)

##   
## 1 2 3 4 5  
## 1 0 0 0 0 11  
## 2 0 0 0 13 0  
## 3 0 0 8 0 0  
## 4 0 52 0 0 0  
## 5 38 0 0 0 0

weight\_ed$curr3R <- 6 - weight\_ed$curr3  
table(weight\_ed$curr3,weight\_ed$curr3R)

##   
## 1 2 3 4 5  
## 1 0 0 0 0 13  
## 2 0 0 0 8 0  
## 3 0 0 19 0 0  
## 4 0 48 0 0 0  
## 5 33 0 0 0 0

#Alpha for Curr Complete items including reverse scoring  
  
psych:: alpha(weight\_ed[,c("curr1R","curr2R","curr3R","curr4","curr5","curr6","curr7","curr8","curr9")], na.rm = TRUE)

##   
## Reliability analysis   
## Call: psych::alpha(x = weight\_ed[, c("curr1R", "curr2R", "curr3R",   
## "curr4", "curr5", "curr6", "curr7", "curr8", "curr9")], na.rm = TRUE)  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r  
## 0.82 0.82 0.89 0.34 4.6 0.026 3.5 0.75 0.29  
##   
## 95% confidence boundaries   
## lower alpha upper  
## Feldt 0.77 0.82 0.86  
## Duhachek 0.77 0.82 0.87  
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se var.r med.r  
## curr1R 0.81 0.81 0.88 0.35 4.3 0.027 0.059 0.31  
## curr2R 0.81 0.81 0.88 0.35 4.4 0.026 0.056 0.31  
## curr3R 0.81 0.81 0.88 0.35 4.3 0.027 0.061 0.31  
## curr4 0.80 0.80 0.88 0.33 3.9 0.029 0.080 0.24  
## curr5 0.80 0.80 0.87 0.33 4.0 0.028 0.054 0.29  
## curr6 0.79 0.79 0.87 0.32 3.8 0.030 0.059 0.28  
## curr7 0.78 0.78 0.87 0.31 3.6 0.031 0.072 0.28  
## curr8 0.81 0.81 0.89 0.35 4.3 0.028 0.075 0.28  
## curr9 0.79 0.79 0.87 0.32 3.8 0.030 0.061 0.28  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## curr1R 122 0.58 0.57 0.53 0.45 1.9 1.16  
## curr2R 122 0.57 0.55 0.52 0.43 2.2 1.25  
## curr3R 121 0.60 0.58 0.55 0.46 2.3 1.25  
## curr4 122 0.67 0.68 0.62 0.55 3.8 1.27  
## curr5 122 0.64 0.65 0.63 0.52 4.3 1.11  
## curr6 122 0.70 0.71 0.69 0.59 4.2 1.21  
## curr7 122 0.76 0.76 0.72 0.66 3.9 1.21  
## curr8 122 0.53 0.57 0.48 0.43 4.5 0.83  
## curr9 122 0.70 0.70 0.68 0.59 4.2 1.16  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 miss  
## curr1R 0.48 0.31 0.06 0.11 0.04 0.00  
## curr2R 0.31 0.43 0.07 0.11 0.09 0.00  
## curr3R 0.27 0.40 0.16 0.07 0.11 0.01  
## curr4 0.08 0.10 0.13 0.32 0.37 0.00  
## curr5 0.04 0.07 0.06 0.18 0.66 0.00  
## curr6 0.06 0.08 0.05 0.20 0.61 0.00  
## curr7 0.07 0.06 0.17 0.30 0.39 0.00  
## curr8 0.02 0.00 0.07 0.28 0.63 0.00  
## curr9 0.06 0.05 0.09 0.23 0.57 0.00

#Composite Variable for Curr  
  
weight\_ed$CURR\_SCALE <- rowMeans(weight\_ed[,c("curr1R","curr2R","curr3R","curr4","curr5","curr6","curr7","curr8","curr9")], na.rm = TRUE)

#Creating Sub-Scale Scores for UMBFAT Scale—

#Equal Rights UMB Sub-Scale  
weight\_ed$ER\_Subscale <- rowMeans(weight\_ed[,c("UMBFAT1","UMB.FAT.11","UMB.FAT.13","UMB.FAT.14","UMB.FAT.18")], na.rm = TRUE)  
#Distance UMB Sub-Scale  
weight\_ed$DIS\_Subscale <- rowMeans(weight\_ed[,c("UMBFAT2","UMBFAT8R","UMB.FAT.17","UMBFAT19R","UMB.FAT.20")], na.rm = TRUE)  
#Negative Judgment UMB Sub-Scale  
weight\_ed$NJ\_Subscale <- rowMeans(weight\_ed[,c("UMBFAT5R","UMBFAT7R","UMBFAT12R","UMBFAT15R","UMBFAT16R")], na.rm = TRUE)  
  
##Double checking Subscales  
psych:: describe(weight\_ed[,c('ER\_Subscale','DIS\_Subscale','NJ\_Subscale')])

## vars n mean sd median trimmed mad min max range skew kurtosis  
## ER\_Subscale 1 122 1.85 1.23 1.2 1.61 0.3 1 6.4 5.4 1.66 2.33  
## DIS\_Subscale 2 122 1.73 0.92 1.2 1.56 0.3 1 4.8 3.8 1.24 0.80  
## NJ\_Subscale 3 122 1.55 0.91 1.0 1.34 0.0 1 4.6 3.6 1.67 1.70  
## se  
## ER\_Subscale 0.11  
## DIS\_Subscale 0.08  
## NJ\_Subscale 0.08

#Alpha for ER Subscale  
psych:: alpha(weight\_ed[,c("UMBFAT1","UMB.FAT.11","UMB.FAT.13","UMB.FAT.14","UMB.FAT.18")], na.rm = TRUE)

##   
## Reliability analysis   
## Call: psych::alpha(x = weight\_ed[, c("UMBFAT1", "UMB.FAT.11", "UMB.FAT.13",   
## "UMB.FAT.14", "UMB.FAT.18")], na.rm = TRUE)  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r  
## 0.9 0.9 0.91 0.64 9 0.014 1.9 1.2 0.65  
##   
## 95% confidence boundaries   
## lower alpha upper  
## Feldt 0.87 0.9 0.93  
## Duhachek 0.87 0.9 0.93  
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se var.r med.r  
## UMBFAT1 0.88 0.88 0.90 0.65 7.4 0.017 0.0369 0.70  
## UMB.FAT.11 0.87 0.87 0.86 0.63 6.7 0.019 0.0150 0.64  
## UMB.FAT.13 0.92 0.92 0.91 0.73 10.9 0.012 0.0089 0.74  
## UMB.FAT.14 0.85 0.85 0.86 0.59 5.7 0.021 0.0308 0.58  
## UMB.FAT.18 0.87 0.87 0.86 0.62 6.5 0.019 0.0170 0.65  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## UMBFAT1 122 0.83 0.84 0.77 0.74 1.7 1.3  
## UMB.FAT.11 122 0.88 0.87 0.86 0.80 1.9 1.5  
## UMB.FAT.13 122 0.70 0.72 0.62 0.57 1.7 1.3  
## UMB.FAT.14 120 0.93 0.92 0.91 0.87 1.9 1.6  
## UMB.FAT.18 122 0.89 0.88 0.87 0.81 2.0 1.6  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 6 7 miss  
## UMBFAT1 0.68 0.15 0.04 0.09 0.02 0.00 0.02 0.00  
## UMB.FAT.11 0.64 0.08 0.06 0.19 0.01 0.00 0.02 0.00  
## UMB.FAT.13 0.64 0.19 0.06 0.08 0.01 0.01 0.02 0.00  
## UMB.FAT.14 0.68 0.10 0.03 0.13 0.00 0.00 0.05 0.02  
## UMB.FAT.18 0.64 0.09 0.04 0.18 0.00 0.01 0.04 0.00

#Alpha for DIS Subscale  
psych:: alpha(weight\_ed[,c("UMBFAT2","UMBFAT8R","UMB.FAT.17","UMBFAT19R","UMB.FAT.20")], na.rm = TRUE)

##   
## Reliability analysis   
## Call: psych::alpha(x = weight\_ed[, c("UMBFAT2", "UMBFAT8R", "UMB.FAT.17",   
## "UMBFAT19R", "UMB.FAT.20")], na.rm = TRUE)  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r  
## 0.74 0.74 0.71 0.37 2.9 0.035 1.7 0.92 0.39  
##   
## 95% confidence boundaries   
## lower alpha upper  
## Feldt 0.66 0.74 0.81  
## Duhachek 0.67 0.74 0.81  
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se var.r med.r  
## UMBFAT2 0.73 0.74 0.69 0.41 2.8 0.039 0.0035 0.39  
## UMBFAT8R 0.68 0.68 0.63 0.35 2.1 0.045 0.0109 0.33  
## UMB.FAT.17 0.70 0.70 0.64 0.37 2.4 0.043 0.0029 0.39  
## UMBFAT19R 0.70 0.70 0.65 0.37 2.3 0.043 0.0109 0.39  
## UMB.FAT.20 0.68 0.68 0.63 0.35 2.1 0.046 0.0068 0.37  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## UMBFAT2 122 0.58 0.63 0.47 0.40 1.3 1.0  
## UMBFAT8R 122 0.76 0.74 0.65 0.56 1.8 1.5  
## UMB.FAT.17 122 0.73 0.70 0.60 0.52 2.2 1.4  
## UMBFAT19R 122 0.68 0.71 0.59 0.52 1.4 1.1  
## UMB.FAT.20 122 0.76 0.74 0.65 0.57 1.9 1.4  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 6 7 miss  
## UMBFAT2 0.87 0.07 0.00 0.03 0.01 0.00 0.02 0  
## UMBFAT8R 0.72 0.09 0.02 0.08 0.04 0.03 0.01 0  
## UMB.FAT.17 0.52 0.12 0.01 0.33 0.01 0.00 0.01 0  
## UMBFAT19R 0.82 0.07 0.01 0.09 0.00 0.01 0.01 0  
## UMB.FAT.20 0.68 0.02 0.01 0.28 0.00 0.00 0.01 0

#Alpha NJ\_Subscale  
psych:: alpha(weight\_ed[,c("UMBFAT5R","UMBFAT7R","UMBFAT12R","UMBFAT15R","UMBFAT16R")], na.rm = TRUE)

##   
## Reliability analysis   
## Call: psych::alpha(x = weight\_ed[, c("UMBFAT5R", "UMBFAT7R", "UMBFAT12R",   
## "UMBFAT15R", "UMBFAT16R")], na.rm = TRUE)  
##   
## raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd median\_r  
## 0.85 0.85 0.84 0.53 5.6 0.022 1.6 0.91 0.54  
##   
## 95% confidence boundaries   
## lower alpha upper  
## Feldt 0.8 0.85 0.88  
## Duhachek 0.8 0.85 0.89  
##   
## Reliability if an item is dropped:  
## raw\_alpha std.alpha G6(smc) average\_r S/N alpha se var.r med.r  
## UMBFAT5R 0.80 0.81 0.77 0.51 4.1 0.030 0.011 0.53  
## UMBFAT7R 0.80 0.81 0.79 0.51 4.2 0.029 0.017 0.54  
## UMBFAT12R 0.86 0.86 0.84 0.60 6.1 0.021 0.004 0.61  
## UMBFAT15R 0.80 0.80 0.77 0.49 3.9 0.030 0.022 0.46  
## UMBFAT16R 0.81 0.81 0.79 0.52 4.3 0.029 0.016 0.54  
##   
## Item statistics   
## n raw.r std.r r.cor r.drop mean sd  
## UMBFAT5R 122 0.82 0.82 0.78 0.71 1.6 1.15  
## UMBFAT7R 122 0.82 0.81 0.76 0.69 1.7 1.22  
## UMBFAT12R 122 0.66 0.67 0.54 0.48 1.4 1.13  
## UMBFAT15R 122 0.83 0.84 0.79 0.73 1.4 0.99  
## UMBFAT16R 122 0.81 0.80 0.75 0.68 1.7 1.27  
##   
## Non missing response frequency for each item  
## 1 2 3 4 5 6 7 miss  
## UMBFAT5R 0.70 0.13 0.02 0.11 0.03 0.00 0.00 0  
## UMBFAT7R 0.70 0.13 0.02 0.10 0.02 0.02 0.00 0  
## UMBFAT12R 0.84 0.05 0.01 0.08 0.01 0.01 0.01 0  
## UMBFAT15R 0.85 0.03 0.02 0.09 0.00 0.01 0.00 0  
## UMBFAT16R 0.72 0.10 0.02 0.14 0.01 0.01 0.01 0

#Checking descriptives for Curriculum scale created

psych:: describe(weight\_ed[,c("CURR\_SCALE")])

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 122 3.49 0.75 3.56 3.51 0.66 1.33 5 3.67 -0.22 0.04 0.07

#Checking BAOP Scale

psych:: describe(weight\_ed[,c("BAOP\_SCALE")])

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 122 3.18 0.91 3.38 3.23 0.93 1 4.75 3.75 -0.44 -0.53 0.08

#Descriptives Curr Subscales —-

##Weight Management Curriculum Subscale  
weight\_ed$CURR\_WM <- rowMeans(weight\_ed[,c("curr1","curr2","curr3")], na.rm = TRUE)  
psych::describe(weight\_ed[,c("CURR\_WM")])

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 122 3.84 1.11 4 4 0.99 1 5 4 -1.09 0.3 0.1

##Health At Every Size Curriculum Subscale  
weight\_ed$CURR\_HAES <- rowMeans(weight\_ed[,c("curr4","curr5","curr6","curr7","curr8","curr9")], na.rm = TRUE)  
psych::describe(weight\_ed[,c("CURR\_HAES")])

## vars n mean sd median trimmed mad min max range skew kurtosis se  
## X1 1 122 4.16 0.87 4.33 4.31 0.74 1.17 5 3.83 -1.45 1.67 0.08

#Frequencies for Categorical Variables except Sect Size   
  
table(weight\_ed$gender)

##   
## Female Male Prefer not to say   
## 2 89 30 1

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

weight\_ed %>% count(gender)

## gender n  
## 1 2  
## 2 Female 89  
## 3 Male 30  
## 4 Prefer not to say 1

table(weight\_ed$numsect)

##   
## 1 1.5 2 2.5 3 4 4.5 5 6 7 8 9 10 11 12   
## 24 5 44 2 16 7 1 4 9 3 2 1 1 1 1

table(weight\_ed$degree)

##   
## Athletic Training Medicine (MD) Nursing   
## 3 7 33   
## Nutrition/Dietetics Occupational Therapy Other (please specify)   
## 34 2 39   
## Physical Therapy   
## 4

table(weight\_ed$numstudents)

##   
## Between 5,001-14,999 Greater than or equal to 15,000   
## 34 52   
## Less than or equal to 5,000   
## 36

table(weight\_ed$yearsofcoll)

##   
## 2 year, public 4 year, private non-religious   
## 2 7   
## 4 year, private religious 4 year, public   
## 21 91

table(weight\_ed$teachyears)

##   
## 0-5 years 11-15 years 16-20 years 6-10 years   
## 30 32 10 34   
## More than 20 years   
## 16

table(weight\_ed$degreeyear)

##   
##   
## 2   
## 1982   
## 1   
## 1985   
## 1   
## 1987   
## 1   
## 1989   
## 4   
## 1994   
## 2   
## 1996   
## 3   
## 1998   
## 4   
## 2000   
## 2   
## 2001   
## 3   
## 2003   
## 2   
## 2004   
## 3   
## 2004, defending for terminal degree in 2024   
## 1   
## 2005   
## 3   
## 2006   
## 3   
## 2007   
## 3   
## 2008   
## 7   
## 2009   
## 4   
## 2010   
## 3   
## 2011   
## 6   
## 2012   
## 1   
## 2013   
## 3   
## 2014   
## 9   
## 2015   
## 3   
## 2016   
## 6   
## 2017   
## 7   
## 2018   
## 5   
## 2018 earned a PhD   
## 1   
## 2019   
## 3   
## 2020   
## 7   
## 2021   
## 6   
## 2022   
## 9   
## 2023   
## 1   
## Bachelor   
## 1   
## Doctorate   
## 1   
## MSN   
## 1

table(weight\_ed$clinical)

##   
## Does not apply No Yes   
## 1 5 69 47

table(weight\_ed$futurepaths)

##   
## Allied Health   
## 9   
## Athletic Training   
## 4   
## General education/non-healthcare professionals   
## 5   
## Medicine   
## 11   
## Nursing   
## 32   
## Nutrition/Dietetics   
## 35   
## Occupational Therapy   
## 2   
## Other (please specify)   
## 14   
## Physical Therapy   
## 9   
## Speech Language Pathology   
## 1

table(weight\_ed$black)

##   
## Black or African American   
## 120 2

table(weight\_ed$white)

##   
## White or Caucasian   
## 13 109

table(weight\_ed$asian)

##   
## Asian or Asian American   
## 119 3

table(weight\_ed$hisp)

##   
## Hispanic, Latinx, Chicano, or Mexicano   
## 118 4

table(weight\_ed$native)

##   
##   
## 121   
## American Indian, Native American, or Alaskan Native   
## 1

table(weight\_ed$pacific)

## < table of extent 0 >

table(weight\_ed$other)

##   
## Other race not listed   
## 120 2

table(weight\_ed$pna)

##   
## Prefer not to answer   
## 118 4