WSP setup and cleaning code

Lizzie Jones*

02/05/2021

WSP Data cleaning

About this rMarkdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com. To generate the document of all content, click the **Knit** button.

This rMarkdown document will be periodically updated and uploaded to the OneDrive folder and pushed to the WSP GitHub code repository. The primary format of this document is HTML, but this can be easily changed by changing the output (e.g. PDF, GitHub) using the 'output' section at the top of the document. The possible output formats are listed here: https://rmarkdown.rstudio.com/lesson-9.html.

Data cleaning walk-through

This rMarkdown document has been written to take the reader through the data cleaning process for the White Stork Survey dataset.

The key aims of this rMarkdown are as follows:

1. View the data and familiarise the reader with the overall dataset
2. Format any data/questions into the appropritae format (e.g. factors or numerical responses)
3. Convert any raw data into more useable fomrats (e.g. seconds, rather than sec/min/hr)
4. Check for straightlining, even-odd consistencies and non-serious responses and consider for removal
5. Check open-ended questions and remove any non-serious/joke responses
6. Check for internal consistency of scores using Cronbach's Alpha

Initial formatting

To easily view which respondents had seen white Storks inside or outside the UK, and I have created a new composite value column with which we can sort or subset respondents (column = "Q8.WhereSeen, values = UK, Outside UK, Both, Neither, NA)

I have created a new age column to create matching age groups for both surveys (new column = 'Age_group_match'). The oldest age group for both surveys is now 65+. I converted the 'TimeTaken' column to a total number of seconds (SecsTaken) for easier to more easily investigate means and quantiles.

*University of Brighton, l.jones4@brighton.ac.uk

```r
# Cleaning full dataset to prevent having to do code for all samples


## Create a composite columns of where respondents had seen White Storks (UK, Outside UK, or Both)
# Multiple conditions when adding new column to dataframe:
str(all_data$Q8.1_UK) # Column is integer so need to format case_when accordingly
```

```
##  int [1:3560] NA 0 0 0 1 NA NA 0 NA 0 ...
```

```r
all_data <- all_data %>% mutate(Q8.WhereSeen =
                    case_when(Q8.1_UK == 1L & Q8.1_OutsideUK == 0L ~ "UK",
                              Q8.1_UK == 1L & Q8.1_OutsideUK == NA_integer_ ~ "UK",
                              Q8.1_UK == 0L & Q8.1_OutsideUK == 1L ~ "OutsideUK",
                              Q8.1_UK == NA_integer_ & Q8.1_OutsideUK == 1L ~ "OutsideUK",
                              Q8.1_UK == 1L & Q8.1_OutsideUK == 1L ~ "Both",
                              Q8.1_UK == 0L & Q8.1_OutsideUK == 0L ~ "Neither",
                              Q8.1_UK == NA_integer_ & Q8.1_OutsideUK == NA_integer_ ~ "NA",))
# Move new column next to existing Q8 columns and view new column
all_data <- all_data %>%  sjmisc::move_columns(Q8.WhereSeen, .after = "Q8.1_OutsideUK")


## Age columns
all_data$Age_group_match <- all_data$Age_group # Create new column with matching age-group formats
all_data <- all_data  %>%
  dplyr::mutate(Age_group_match = recode(Age_group_match, "c('65-74', '75 and over')='65+'"))
summary(all_data$Age_group_match)
```

```
##               18-24                25-34                35-44
##                 260                  510                  585
##               45-54                55-64                  65+
##                 700                  774                  719
## Prefer not to answer
##                  12
```

```r
## Formatting date and time columns
# Create numeric column of time taken (seconds)
all_data$SecsTaken <- as.numeric(lubridate::seconds(all_data$TimeTaken))
all_data$StartDate <- as.Date(all_data$StartDate, format = "%d/%m/%Y")
all_data$CompletionDate <- as.Date(all_data$CompletionDate, format = "%d/%m/%Y")
```

After the WSP group meeting on 17/05/21 I removed the 3 Northern Irish respondents from the Proactive sample and merged the respondents thta selected Wadhurst and Wadhurst Park as the nearest release site.

```r
### Removing the N.Ireland respondents
summary(all_data$Region)
```

```
##            East Midlands          East of England          Greater London
##                      127                      232                      331
##               North East               North West        Northern Ireland
##                       76                      175                        3
##                 Scotland               South East              South West
##                        152                       1729                        313
##                      Wales              West Midlands Yorkshire and the Humber
##                         98                        160                        164
```

```
# Remove rows where Region = "Northern Ireland"
all_data <- subset(all_data, all_data$Region != "Northern Ireland")
 # Drop the N.Ireland factor level
all_data$Region <- droplevels(all_data$Region)
summary(all_data$Region)
```

```
##             East Midlands          East of England           Greater London
##                       127                      232                      331
##                North East               North West                 Scotland
##                        76                      175                      152
##                South East               South West                    Wales
##                      1729                      313                       98
##             West Midlands Yorkshire and the Humber
##                       160                      164
```

```
### Merging the Wadhurst and Wadhurst Park respondents
all_data <- transform(all_data,
          ReleaseSite=plyr::revalue(ReleaseSite,c("Wadhurst"="Wadhurst Park")))
summary(all_data$ReleaseSite)
```

```
##              Knepp Knepp-Wintershall                No     Wadhurst Park
##                437               270              2524               198
##        Wintershall
##                128
```

Full dataset checks

I initially went through the full dataset manually and checked for any respondents that were clearly straightlining and/or not taking the questionnaire seriously (e.g. open answers such as “jkjkjkjk”). I removed the entire row for respondents that were both non-serious and straightlining, but I removed the open answers only for those who appreared to take the close questions seriously and put junk answers for the open questions.

```
##### Data cleaning using the 'careless' package

# Overall straightlining (whole survey)
 # Identifies the longest string of identical consecutive responses for each observation
all_straightline <- longstring(all_data, avg = FALSE)
summary(all_straightline) # Mean number of consecutive attitude answers = 14, max = 14
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00   11.00   11.00   11.22   13.00   14.00
```

```
 # 127 rows with 14 consecutive answers (possible candidates for removal)
all_possible_st <- which(grepl(14, all_straightline))


### Checking straightlining for all Likert style questions with over 3 columns
# Checking the attitudes to WS columns (Q12, 13 and 14)
ncol(all_attitude_colnames) # Max possible number of consecutive answers is 10
```

```
## [1] 10
```

```
# Identifies the longest string of identical consecutive
attitudes_straight <- longstring(all_attitude_colnames, avg = FALSE)
summary(attitudes_straight) # Mean number of consecutive attitude answers = 3
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.000   2.995   4.000  10.000
```

```
# Find rows with 10 consecutive answers (possible candidates for removal)
attitude_possible_st <- which(grepl(10, attitudes_straight))

# Checking the NCI columns
ncol(Q19_NCI_colnames) # Max possible number of consecutive answers is 6
```

```
## [1] 6
```

```
nci_straight <- longstring(Q19_NCI_colnames, avg = FALSE)
summary(nci_straight) # Mean number of consecutive attitude answers = 3
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   5.000   4.611   6.000   6.000
```

```r
# Find rows with 6 consecutive answers (~1700 gave max consecutive for NCI
# across both surveys, which makes sense especially for proactive sample,
# as sample will have a high interest and connection to nature)
# Therefore will not remove suspected participants based on this question
nci_possible_st <-which(grepl(6, nci_straight))

# Checking the ProCoBS columns
ncol(Q21_ProCoBS_colnames) # Max possible number of consecutive answers is 4
```

```
## [1] 4
```

```r
ProCoBS_straight <- longstring(Q21_ProCoBS_colnames, avg = FALSE)
summary(ProCoBS_straight) # Mean number of consecutive attitude answers = 3
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   1.827   2.000   4.000
```

```r
# 245 rows with 10 consecutive answers (possible candidates for removal,
# but only 4 questions so unintentional straightlining would be likely for this question)
# Therefore will not remove suspected participants based on this question
pro_possible_st <-which(grepl(4, ProCoBS_straight))



# Comparing overall suspected straightlining row numbers to suspected straightlining attitude row numbe
both_straightline_rownames <- intersect(all_possible_st, attitude_possible_st) # rows in both
both_straightline_rownames
```

```
## [1] 2600 2908 3035 3203 3519
```

```r
rows_straightlined <- all_data[c(2600, 2908, 3035, 3203, 3519), ] # Create df to view
summary(rows_straightlined$SecsTaken) # Mean survey time = 235 seconds or ~4 mins.
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   125.0   195.0   257.0   235.6   273.0   328.0
```

```r
# Most have skipped the open questions

# Removing straightlined participants
row_straightlined <- c(2600, 2908, 3035, 3203, 3519) # 5 respondents
## Create new dataset for further analysis and remove rows with straightlining etc.
data_clean <- all_data[!all_data$UniqueID_all %in% row_straightlined,]

# Writing this new dataframe of the cleaned dataframe as a dataframe for visual inspection in Excel
write.csv(data_clean, "WSP_R_cleaned_dataset.csv")  ### Once cleaned save as WSP_R_cleaned_dataset1.csv
```

Checking the fastest responses

I then focussed on the fastest 5% of respondents across both surveys as they are most likely to have straight-lined through the survey. I visually inspected the data, then used the 'careless' package to find evidence of straightlining 'even-odd' consistencies, and intra-individual response variability (IRV), across the whole survey and within the multiple choice questions (particularly questions 4, 5, 13, 15, 16, 17, 22, 23, 24).

```
### Explore average time taken to complete questionnaire and check for straightlining
quantile(data_clean$SecsTaken, 0.1) # Fastest 10% of all respondents = completion in 188.9 seconds/ abo
```

```
## 10%
## 190
```

```
quantile(data_clean$SecsTaken, 0.05) # Fastest 5% of all respondents = completion in 117.95 seconds/ ab
```

```
##     5%
## 118.55
```

```
quantile(data_clean$SecsTaken, 0.025) # Fastest 2.5% of all respondents = completion in 70.975 seconds/
```

```
## 2.5%
##   71
```

```
fastest_10 <- subset(data_clean, SecsTaken < 191) # Sample of fastest 10% of all respondents
fastest_5 <- subset(data_clean, SecsTaken < 121) # Sample of fastest 5% of all respondents
fastest_2.5 <- subset(data_clean, SecsTaken < 72) # Sample of fastest 2.5% of all respondents
summary(fastest_5$SurveyType) # 96% of respondents in fastest 5% are from the NatRep sample
```

```
##    NatRep Proactive
##       174         7
```

```
summary(fastest_2.5$SurveyType) # 100% of respondents in fastest 2.5% are from the NatRep sample
```

```
##    NatRep Proactive
##        91         0
```

Focussing on the the fastest 5% of responses

Here I have checked the responses of the fastest 5% of the dataframe (after straightlined responses had been removed). I compare the mean values of the numeric/score columns between the full cleaned dataset and the fastest 5% but as there don't seem to be significant differences between the full dataset and the fastest 5% sample, I manually checked the full dataset.

```
### Checking the fastest 5% of respondents for straightlining across whole survey
 # Identifies the longest string of identical consecutive responses for each respondent
long_fastest_5 <- longstring(fastest_5, avg = FALSE)
evenodd_fastest_5 <- evenodd(fastest_5, rep(5,10))

# Checking the fastest 5% for straightlining within each set of mutliple choice questions
# summary(data_clean$Q5_overallscore_diet) # e.g. Q5 diet
# summary(fastest_5$Q5_overallscore_diet) ### Not a significant difference in Q5 diet score

### Full cleaned dataset
careless_all <- evenodd(data_clean, rep(5,10))
# Calculates the intra-individual response variability (IRV)
irv_total <- irv(data_clean)

### Fastest 5%
careless_fast <- evenodd(fastest_5, rep(5,10))
irv_fast <- irv(fastest_5)

# Writing the fastest 5% subset of the cleaned dataframe as a dataframe for visual inspection in Excel
write.csv(fastest_5, "WSP_fastest5.csv")
```

Creating a final, cleaned dataset

The full dataset was checked for overall straightlining again and then manually checked the dataset for any irregularities and non-serious answers. Many respondents who wrote non-serious/joke answers in the open questions also shows evidence of straighlining, therefore these respondents were removed. If the respondent did not appreat to straightline, but had written random letters (e.g. "jadbfsjdbg") in the open questions, only the open question answers were removed.

One cleaned I wrote a new 'final' dataset as a CSV file for further stats and analysis called 'final_data'.

```
##### Manually check the full dataset
# (write down UniqueID_all numbers for removal in R)
# Manually remove non-serious open-ended question responses
# ---------------------------------------------------------------------

# Load in manually cleaned dataset
data_clean1 <- read.csv("WSP_R_cleaned_dataset1.csv", header = TRUE, stringsAsFactors=TRUE)

# Removed as comments suggested not taking the survey seriously (e.g. UniqueID_all = 3309)
# Or straightlining that hasn't picked up in previous analysis
manualcheckID_to_remove <- c(566, 916, 2608, 2643, 2738, 2746, 2758, 2777, 2869, 2889, 3022,
                            3201, 3209, 3308, 3309, 3321, 3352,3363, 3441, 3464, 3466)

## Create new dataset for further analysis and remove rows with straightlining etc.
data_clean2 <- data_clean1[!data_clean1$UniqueID_all %in% manualcheckID_to_remove,]
```

```r
#### Checking timeline of respondent removal
nrow(original_data) # 3560 - Original dataset sample size
```

```
## [1] 3560
```

```r
nrow(all_data) ### 3557 - After removal of the 3 N.Ireland respondents
```

```
## [1] 3557
```

```r
nrow(data_clean) # 3552 - After removal of 5 straightlined respondents
```

```
## [1] 3552
```

```r
nrow(data_clean2) # 3531
```

```
## [1] 3531
```

```
## 'data.frame':    3531 obs. of  226 variables:
##  $ X
##  $ SurveyType
##  $ UniqueID_long
##  $ UniqueID_short
##  $ UniqueID_all
##  $ TimeTaken
##  $ StartDate
##  $ StartTime
##  $ CompletionDate
##  $ CompletionTime
##  $ Q1_aware_stork
##  $ Q2_photo_recog
##  $ Q2_photo_recog_score
##  $ Q3_is_native
##  $ Q3_is_native_explain
##  $ Q4.1_migrate
##  $ Q4.1_migrate_score
##  $ Q4.2_wingspan
##  $ Q4.2_wingspan_score
##  $ Q4.3_globallyrare
##  $ Q4.3_globallyrare_score
##  $ Q4_overallscore
##  $ Q5a_amphibians_diet
##  $ Q5b_birdeggs.chicks_diet
##  $ Q5c_carrion_diet
##  $ Q5d_fish_diet
##  $ Q5e_foodwaste_diet
##  $ Q5f_fruit_diet
##  $ Q5g_inverts_diet
##  $ Q5h_reptiles_diet
##  $ Q5i_seeds_diet
##  $ Q5j_smallmammals_diet
##  $ Q5k_vegetation_diet
##  $ Q5l_Don.tKnow_diet
##  $ Q5_rawscore_diet
##  $ Q5_diet_overallscore
##  $ Q6a_farmland_habitat
##  $ Q6b_grassland_habitat
##  $ Q6c_wetlands_habitat
##  $ Q6d_woodland_habitat
##  $ Q6e_urban_habitat
##  $ Q6f_Don.tKnow_habitat
##  $ Q6_habitat_rawscore
##  $ Q6_habitat_overallscore
##  $ Q7a_chimneys_nesting
##  $ Q7b_ground_nesting
##  $ Q7c_roofs_nesting
##  $ Q7d_telegraphpoles_nesting
##  $ Q7e_trees_nesting
##  $ Q7f_Don.tKnow_nesting
##  $ Q7_nesting_rawscore
##  $ Q7_nesting_overallscore
##  $ KnowledgeScore
##  $ Q8_wild_seen
##  $ Q8_captivity_seen
##  $ Q8_pictures_video
##  $ Q8_No
##  $ Q8_NotSure
##  $ Q8.1_UK
##  $ Q8.1_OutsideUK
##  $ Q8.WhereSeen
##  $ Q8.2_feelings
##  $ Q9_heard
##  $ Q9a_what_heard
##  $ Q10_project_knowledge
##  $ Q10a_WSPwebsite
##  $ Q10a_Socialmedia
##  $ Q10a_TV.Radio
##  $ Q10a_Newspaper
##  $ Q10a_Email
##  $ Q10a_Magazine
##  $ Q10a_Leaflet
##  $ Q10a_spokesperson
##  $ Q10a_VisitingKnepp
##  $ Q10a_Wordofmouth
##  $ Q10a_Other
##  $ Q10a_Other_open
##  $ Q10b_WSPwebsite
##  $ Q10b_Socialmedia
##  $ Q10b_TV.Radio
##  $ Q10b_Newspaper
##  $ Q10b_Email
##  $ Q10b_Magazine
##  $ Q10b_Leaflet
##  $ Q10b_spokesperson
##  $ Q10b_NotInterested
##  $ Q10b_Other
##  $ Q10b_Other_open
##  $ Q11_word1
##  $ Q11_word2
##  $ Q11_word3
##  $ Q12.1..White.storks.symbolise.the.beauty.of.nature.
##  $ Q12.1_agreement_score
##  $ Q12.2..White.storks.play.an.important.role.in.their.environment.
##  $ Q12.2_agreement_score
##  $ Q12.3..Reintroduced.white.storks.may.have.a.negative.impact.on.my.life.
##  $ Q12.3_agreement_score
##  $ Q12.4..I.do.not.want.white.storks.living.near.me.
##  $ Q12.4_agreement_score
##   [list output truncated]
```

```
## [1] 3560
```

```
##    NatRep Proactive
##      1167      2393
```

```
## [1] 3531
```

```
##    NatRep Proactive
##      1143      2388
```

Cronbach's alpha

Now we have a cleaned dataset I have gone through the grouped columns are numeric scores of Likert or multiple choice questions, including: AttitudeScore, NCI, EnvConcern.score, ProCoBS and BirdInterestScore.

Based on the 0.7 threshold, all groups have an acceptable Cronbach's alpha score.

```
### Reminding myself of the column names again!
# colnames(final_data)
library("psych")

# Using Cronbach's alpha on the score columns using the psych package (alpha::psych)
# Questions 13 & 14 attitudes
final_data %>%
  select(., starts_with("Q12"), starts_with("Q13"), starts_with("Q14")) %>%
  select(., ends_with('score')) %>%
  psych::alpha(title = "Attitudes")
```

```
##
## Reliability analysis  Attitudes
## Call: psych::alpha(x = ., title = "Attitudes")
##
##   raw_alpha std.alpha G6(smc) average_r S/N    ase mean   sd median_r
##       0.91      0.92    0.93      0.42  11 0.0021  4.1 0.62     0.43
##
##  lower alpha upper     95% confidence boundaries
## 0.91 0.91 0.92
##
##  Reliability if an item is dropped:
##                       raw_alpha std.alpha G6(smc) average_r  S/N alpha se
## Q12.1_agreement_score      0.91      0.91    0.92      0.42 10.1   0.0023
## Q12.2_agreement_score      0.91      0.91    0.92      0.42 10.2   0.0023
## Q12.3_agreement_score      0.91      0.91    0.92      0.43 10.4   0.0022
## Q12.4_agreement_score      0.91      0.91    0.92      0.42 10.1   0.0023
## Q12.5_agreement_score      0.91      0.91    0.92      0.43 10.6   0.0022
## Q13.1_agreement_score      0.91      0.91    0.92      0.41  9.8   0.0023
## Q13.2_agreement_score      0.91      0.91    0.92      0.43 10.4   0.0022
## Q13.3_agreement_score      0.91      0.91    0.92      0.42 10.0   0.0023
## Q13.4_agreement_score      0.91      0.91    0.92      0.42 10.3   0.0023
## Q13.5_agreement_score      0.91      0.91    0.92      0.41  9.9   0.0023
## Q14.1_agreement_score      0.91      0.91    0.92      0.42 10.3   0.0022
## Q14.2_agreement_score      0.91      0.91    0.92      0.42 10.2   0.0023
## Q14.3_agreement_score      0.91      0.91    0.92      0.41  9.9   0.0023
## Q14.4_agreement_score      0.91      0.91    0.92      0.43 10.7   0.0022
## Q14.5_agreement_score      0.91      0.91    0.92      0.42 10.1   0.0023
##                        var.r med.r
## Q12.1_agreement_score 0.0102  0.43
## Q12.2_agreement_score 0.0106  0.44
## Q12.3_agreement_score 0.0099  0.44
## Q12.4_agreement_score 0.0102  0.43
## Q12.5_agreement_score 0.0101  0.44
## Q13.1_agreement_score 0.0104  0.41
## Q13.2_agreement_score 0.0096  0.44
## Q13.3_agreement_score 0.0103  0.43
## Q13.4_agreement_score 0.0102  0.44
## Q13.5_agreement_score 0.0103  0.42
## Q14.1_agreement_score 0.0111  0.44
## Q14.2_agreement_score 0.0103  0.43
## Q14.3_agreement_score 0.0102  0.42
## Q14.4_agreement_score 0.0096  0.44
## Q14.5_agreement_score 0.0111  0.43
##
##  Item statistics
##                          n raw.r std.r r.cor r.drop mean   sd
## Q12.1_agreement_score 3471  0.68  0.69  0.67   0.63  4.2 0.84
## Q12.2_agreement_score 3087  0.66  0.68  0.65   0.61  4.1 0.80
## Q12.3_agreement_score 3376  0.65  0.64  0.62   0.58  4.5 0.88
## Q12.4_agreement_score 3450  0.71  0.70  0.68   0.65  4.5 0.87
## Q12.5_agreement_score 3324  0.60  0.60  0.56   0.53  4.2 0.88
## Q13.1_agreement_score 3491  0.76  0.76  0.75   0.71  4.5 0.81
## Q13.2_agreement_score 3415  0.63  0.63  0.60   0.56  3.9 0.96
## Q13.3_agreement_score 3387  0.73  0.72  0.70   0.67  3.8 1.04
## Q13.4_agreement_score 2910  0.66  0.66  0.63   0.60  3.7 1.05
## Q13.5_agreement_score 3249  0.75  0.74  0.73   0.70  4.0 1.00
## Q14.1_agreement_score 3419  0.66  0.65  0.62   0.59  4.4 0.78
## Q14.2_agreement_score 3216  0.68  0.68  0.66   0.62  3.8 0.97
## Q14.3_agreement_score 3421  0.74  0.74  0.73   0.69  4.2 0.84
## Q14.4_agreement_score 3261  0.59  0.57  0.53   0.50  4.1 1.06
## Q14.5_agreement_score 3416  0.69  0.70  0.67   0.64  4.4 0.88
##
## Non missing response frequency for each item
##                          1    2    3    4    5 miss
## Q12.1_agreement_score 0.01 0.01 0.15 0.41 0.41 0.02
## Q12.2_agreement_score 0.01 0.01 0.17 0.46 0.34 0.13
## Q12.3_agreement_score 0.02 0.03 0.07 0.20 0.68 0.04
## Q12.4_agreement_score 0.02 0.02 0.09 0.17 0.70 0.02
## Q12.5_agreement_score 0.02 0.02 0.13 0.40 0.43 0.06
## Q13.1_agreement_score 0.02 0.01 0.07 0.25 0.65 0.01
## Q13.2_agreement_score 0.02 0.04 0.28 0.34 0.32 0.03
## Q13.3_agreement_score 0.04 0.06 0.26 0.38 0.26 0.04
## Q13.4_agreement_score 0.04 0.09 0.28 0.36 0.24 0.18
## Q13.5_agreement_score 0.03 0.05 0.18 0.41 0.33 0.08
## Q14.1_agreement_score 0.01 0.02 0.08 0.32 0.58 0.03
## Q14.2_agreement_score 0.02 0.05 0.27 0.38 0.27 0.09
## Q14.3_agreement_score 0.01 0.02 0.14 0.45 0.38 0.03
## Q14.4_agreement_score 0.04 0.05 0.13 0.32 0.47 0.08
## Q14.5_agreement_score 0.02 0.02 0.09 0.27 0.61 0.03
```

```r
# Question 21 ProCoBS
final_data %>%
  select(., starts_with("Q21") & ends_with('score')) %>%
  psych::alpha(title = "ProCoBS")
```

```
##
## Reliability analysis  ProCoBS
## Call: psych::alpha(x = ., title = "ProCoBS")
##
##   raw_alpha std.alpha G6(smc) average_r S/N    ase mean   sd median_r
##       0.82      0.81    0.79      0.52 4.4 0.0047    4 1.4      0.5
##
##  lower alpha upper     95% confidence boundaries
## 0.81 0.82 0.83
##
##  Reliability if an item is dropped:
##            raw_alpha std.alpha G6(smc) average_r S/N alpha se  var.r med.r
## Q21.1.score      0.84      0.84    0.79      0.64 5.4   0.0045 0.0094  0.61
## Q21.2.score      0.71      0.71    0.63      0.45 2.4   0.0082 0.0113  0.42
## Q21.3.score      0.77      0.76    0.73      0.51 3.2   0.0064 0.0434  0.44
## Q21.4.score      0.74      0.74    0.67      0.49 2.9   0.0072 0.0113  0.44
##
##  Item statistics
##                n raw.r std.r r.cor r.drop mean  sd
## Q21.1.score 3509  0.65  0.69  0.51   0.46  4.2 1.4
## Q21.2.score 3509  0.89  0.87  0.85   0.77  4.5 1.8
## Q21.3.score 3509  0.81  0.81  0.71   0.65  3.0 1.7
## Q21.4.score 3509  0.85  0.83  0.79   0.70  4.3 1.9
##
## Non missing response frequency for each item
##                1    2    3    4    5    6    7 miss
## Q21.1.score 0.03 0.08 0.17 0.30 0.23 0.13 0.05 0.01
## Q21.2.score 0.09 0.08 0.10 0.20 0.20 0.14 0.18 0.01
## Q21.3.score 0.25 0.22 0.14 0.21 0.09 0.05 0.04 0.01
## Q21.4.score 0.11 0.09 0.13 0.20 0.18 0.13 0.16 0.01
```

```r
# Question 22 BirdInterestScore
final_data %>%
  select(., starts_with("Q23") & ends_with('Score')) %>%
  psych::alpha(title = "BirdInterestScore")
```

```
##
## Reliability analysis  BirdInterestScore
## Call: psych::alpha(x = ., title = "BirdInterestScore")
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean   sd median_r
##       0.86      0.87    0.82      0.68 6.5 0.004  4.1 0.85     0.69
##
##  lower alpha upper     95% confidence boundaries
## 0.86 0.86 0.87
##
##  Reliability if an item is dropped:
##             raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## Q23.1..Score      0.77      0.77    0.63      0.63 3.4   0.0077    NA  0.63
## Q23.2.Score       0.85      0.85    0.74      0.74 5.6   0.0051    NA  0.74
## Q23.3.Score       0.81      0.81    0.69      0.69 4.4   0.0063    NA  0.69
##
##  Item statistics
##                 n raw.r std.r r.cor r.drop mean   sd
## Q23.1..Score 3531  0.90  0.91  0.85   0.79  4.2 0.90
## Q23.2.Score  3531  0.87  0.87  0.75   0.70  4.0 1.00
## Q23.3.Score  3531  0.89  0.89  0.80   0.74  4.1 0.96
##
## Non missing response frequency for each item
##                  1    2    3    4    5 miss
## Q23.1..Score 0.01 0.04 0.13 0.40 0.41    0
## Q23.2.Score  0.02 0.08 0.11 0.42 0.36    0
## Q23.3.Score  0.02 0.05 0.17 0.35 0.41    0
```