Thoughts on Storks - setup and cleaning code

Lizzie Jones[1]

26/07/2021

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 (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 (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.

```
## Create a composite columns of where respondents had seen White Storks (UK, Outside UK, or Bo
th)
# colnames(all_data)
all_data <- all_data %>% mutate(Q8_Seen =
                     case_when(Q8_wild_seen == 1L ~ "Wild",
                               Q8_wild_seen == 1L & Q8_captivity_seen == 1L ~ "Wild",
                               Q8_wild_seen == 0L & Q8_captivity_seen == 1L ~ "Captivity",
                               Q8_wild_seen == 0L & Q8_captivity_seen == 1L & Q8_pictures_video
== 1L ~ "Captivity",
                               Q8_wild_seen == 1L & Q8_captivity_seen == 1L & Q8_No == 1L ~ "N
o/Not sure",
                               Q8_wild_seen == 0L & Q8_captivity_seen == 1L & Q8_No == 1L ~ "N
o/Not sure",
                               Q8_wild_seen == 1L & Q8_captivity_seen == 1L & Q8_NotSure == 1L
 ~ "No/Not sure",
                               Q8_wild_seen == 0L & Q8_captivity_seen == 1L & Q8_NotSure == 1L
 ~ "No/Not sure",
                               Q8_wild_seen == 0L & Q8_captivity_seen == 1L & Q8_NotSure == 1L
 & Q8_NotSure == 1L ~ "No/Not sure",
                               Q8_wild_seen == 0L & Q8_captivity_seen == 0L & Q8_pictures_video
== 1L ~ "No/Not sure",
                               Q8_wild_seen == 0L & Q8_captivity_seen == 0L & Q8_pictures_video
== 0L ~ "No/Not sure"))
all_Q8_colnames <- select(all_data, starts_with("Q8_"))

# 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 ...
```

```
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 %>%
  sjmisc::move_columns(Q8.WhereSeen, .after = "Q8.1_OutsideUK") %>%
  select(., starts_with("Q8.")) %>%
  head(., n=5)
```

```
##   Q8.1_UK Q8.1_OutsideUK Q8.WhereSeen
## 1      NA             NA         <NA>
## 2       0              1    OutsideUK
## 3       0              1    OutsideUK
## 4       0              1    OutsideUK
## 5       1              1         Both
##
Q8.2_feelings
## 1
<NA>
## 2
I saw them nesting on cliff tops and rocks surrounded by the sea in Portugal - it was really co
ol!
## 3
always a pleasing sight, no matter how many you've seen already
## 4
They are common throughout many parts of Europe so didn't feel anything in particular but would
be ecstatic to see one over London.
## 5 Fascinated, and in awe. They're size when flying over head was outstanding (made all the m
ore incredible with a huge feather dropping by my feet!) not something I imagined experiencing
in the UK. My first experience of them was in Hungary, when I heard their bizarre clacking and
wondered what on earth it was. Soon I saw them nesting on the chimneys and poles in towns and o
n roads, their clacking gave the soundtrack to the area a 'wild' sense.
```

```
## Cleaning full dataset to prevent having to do code for all samples
all_data$Age_group_match <- all_data$Age_group # Create new column with matching age-group form
ats
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
```

```
# 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.

```
# 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)

### 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
```

```
# 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)
summary(which(grepl(6, nci_straight)))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     2.0   831.5  1624.0  1672.1  2424.5  3560.0
```

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

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

```
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
```

```
# 245 rows with 10 consecutive answers (possible candidates for removal,
# but only 4 questions so unintentional straightlining would be likely for this question)
summary(which(grepl(4, ProCoBS_straight)))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       9    1251    2459    2105    2935    3553
```

```
# Comparing overall straightlining row numbers to attitude row numbers
both_straightline_rownames <- intersect(all_possible_st,attitude_possible_st) # rows in both
rows_straightlined <- all_data[c(2678, 2713, 2723, 2738, 2772,
             3121, 3209, 3287, 3307, 3455, 3503), ] # Create df to view
summary(rows_straightlined$SecsTaken) # Most took survey quickly and have skipped the open ques
tions
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    58.0   380.5   437.0   477.1   574.0  1133.0
```

```
# Removing straightlined participants
ID_straightlined <- c(2678, 2713, 2723, 2738, 2772,
             3121, 3209, 3287, 3307, 3455, 3503) # 11 respondents
## Create new dataset for further analysis and remove rows with straightlining etc.
data_clean <- all_data[!all_data$UniqueID_all %in% ID_straightlined,]

# Removing non-serious (and often also straightlined through most questions) participants
ID_notserious <- c(2607, 2630, 3297, 3285, 3340, 3441, 3439, 3474) # 8 respondents
## Create new dataset for further analysis and remove rows with straightlining etc.
data_clean <- data_clean[!data_clean$UniqueID_all %in% ID_notserious,]

nrow(all_data)
```

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

```
nrow(data_clean)
```

```
## [1] 3538
```

Checking the fastest responses

I then focussed on the fastest 5% of respondents across both surveys as they are most likely to have straightlined 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/ about 3 mins
```

```
## 10%
## 191
```

```
quantile(data_clean$SecsTaken, 0.05) # Fastest 5% of all respondents = completion in 117.95 seconds/ about 2 mins
```

```
##     5%
## 120.85
```

```
quantile(data_clean$SecsTaken, 0.025) # Fastest 2.5% of all respondents = completion in 70.975 seconds/ about 1.2 mins
```

```
##   2.5%
## 71.425
```

```
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
##       170         7
```

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

```
##    NatRep Proactive
##        89         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%, checked for overall straightlining again and then manually checked the dataset for any irregularities.

I have then created a 'final' dataset for further data checking, stats and analysis called 'final_data'.

```
### 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)
 # Calculates the even-odd consistency score
evenodd_fastest_5 <- evenodd(fastest_5, rep(5,10))

# Checking the fastest 5% for straightlining within each set of mutliple choice questions
# e.g. Q5 diet
# summary(data_clean$Q5_overallscore_diet)
# summary(fastest_5$Q5_overallscore_diet) ### Not a significant difference in Q5 diet score
# between all_data, fastest 5% and 2.5% samples

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

### Fastest 5%
# Calculates the even-odd consistency score
careless_fast <- evenodd(fastest_5, rep(5,10))
# Calculates the intra-individual response variability (IRV)
irv_fast <- irv(fastest_5)


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

# Manually check the data
# Removed as comments suggested not taking the survey seriously (e.g. "lololol")
manualcheckID_to_remove <- c(3321, 2643, 566, 916)
## Create new dataset for further analysis and remove rows with straightlining etc.
data_clean <- data_clean[!data_clean$UniqueID_all %in% manualcheckID_to_remove,]
```

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.

```
# Load in the FINAL dataset for publication
final_data <- read.csv("WSP_R_FINAL_dataset2.csv", header = TRUE, stringsAsFactors=TRUE)

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

```
##  [1] "X.3"
##  [2] "X.2"
##  [3] "X.1"
##  [4] "X"
##  [5] "SurveyType"
##  [6] "UniqueID_long"
##  [7] "UniqueID_short"
##  [8] "UniqueID_all"
##  [9] "TimeTaken"
## [10] "StartDate"
## [11] "StartTime"
## [12] "CompletionDate"
## [13] "CompletionTime"
## [14] "Q1_aware_stork"
## [15] "Q2_photo_recog"
## [16] "Q2_photo_recog_score"
## [17] "Q3_is_native"
## [18] "Q3_is_native_explain"
## [19] "Q4.1_migrate"
## [20] "Q4.1_migrate_score"
## [21] "Q4.2_wingspan"
## [22] "Q4.2_wingspan_score"
## [23] "Q4.3_globallyrare"
## [24] "Q4.3_globallyrare_score"
## [25] "Q4_overallscore"
## [26] "Q5a_amphibians_diet"
## [27] "Q5b_birdeggs.chicks_diet"
## [28] "Q5c_carrion_diet"
## [29] "Q5d_fish_diet"
## [30] "Q5e_foodwaste_diet"
## [31] "Q5f_fruit_diet"
## [32] "Q5g_inverts_diet"
## [33] "Q5h_reptiles_diet"
## [34] "Q5i_seeds_diet"
## [35] "Q5j_smallmammals_diet"
## [36] "Q5k_vegetation_diet"
## [37] "Q5l_Don.tKnow_diet"
## [38] "Q5_rawscore_diet"
## [39] "Q5_diet_overallscore"
## [40] "Q6a_farmland_habitat"
## [41] "Q6b_grassland_habitat"
## [42] "Q6c_wetlands_habitat"
## [43] "Q6d_woodland_habitat"
## [44] "Q6e_urban_habitat"
## [45] "Q6f_Don.tKnow_habitat"
## [46] "Q6_habitat_rawscore"
## [47] "Q6_habitat_overallscore"
## [48] "Q7a_chimneys_nesting"
## [49] "Q7b_ground_nesting"
## [50] "Q7c_roofs_nesting"
## [51] "Q7d_telegraphpoles_nesting"
## [52] "Q7e_trees_nesting"
## [53] "Q7f_Don.tKnow_nesting"
## [54] "Q7_nesting_rawscore"
## [55] "Q7_nesting_overallscore"
## [56] "KnowledgeScore"
## [57] "Q8_wild_seen"
## [58] "Q8_captivity_seen"
## [59] "Q8_pictures_video"
## [60] "Q8_No"
## [61] "Q8_NotSure"
##  [62] "Q8.1_UK"
##  [63] "Q8.1_OutsideUK"
##  [64] "Q8.WhereSeen"
##  [65] "Q8.2_feelings"
##  [66] "Q9_heard"
##  [67] "Q9a_what_heard"
##  [68] "Q10_project_knowledge"
##  [69] "Q10a_WSPwebsite"
##  [70] "Q10a_Socialmedia"
##  [71] "Q10a_TV.Radio"
##  [72] "Q10a_Newspaper"
##  [73] "Q10a_Email"
##  [74] "Q10a_Magazine"
##  [75] "Q10a_Leaflet"
##  [76] "Q10a_spokesperson"
##  [77] "Q10a_VisitingKnepp"
##  [78] "Q10a_Wordofmouth"
##  [79] "Q10a_Other"
##  [80] "Q10a_Other_open"
##  [81] "Q10b_WSPwebsite"
##  [82] "Q10b_Socialmedia"
##  [83] "Q10b_TV.Radio"
##  [84] "Q10b_Newspaper"
##  [85] "Q10b_Email"
##  [86] "Q10b_Magazine"
##  [87] "Q10b_Leaflet"
##  [88] "Q10b_spokesperson"
##  [89] "Q10b_NotInterested"
##  [90] "Q10b_Other"
##  [91] "Q10b_Other_open"
##  [92] "Q11_word1"
##  [93] "Q11_word2"
##  [94] "Q11_word3"
##  [95] "Q12.1..White.storks.symbolise.the.beauty.of.nature."
##  [96] "Q12.1_agreement_score"
##  [97] "Q12.2..White.storks.play.an.important.role.in.their.environment."
##  [98] "Q12.2_agreement_score"
##  [99] "Q12.3..Reintroduced.white.storks.may.have.a.negative.impact.on.my.life."
## [100] "Q12.3_agreement_score"
## [101] "Q12.4..I.do.not.want.white.storks.living.near.me."
## [102] "Q12.4_agreement_score"
## [103] "Q12.5..White.storks.in.England.could.benefit.the.tourism.industry.where.they.re.foun
d."
## [104] "Q12.5_agreement_score"
## [105] "Q13.1..I.would.find.it.exciting.to.see.white.storks.in.the.wild.in.England."
## [106] "Q13.1_agreement_score"
## [107] "Q13.2..White.storks.symbolise.hope..rebirth.and.new.life."
## [108] "Q13.2_agreement_score"
## [109] "Q13.3..Money.spent.reintroducing.white.storks.would.be.better.spent.elsewhere."
## [110] "Q13.3_agreement_score"
## [111] "Q13.4..White.storks.might.be.detrimental.to.local.wildlife."
## [112] "Q13.4_agreement_score"
## [113] "Q13.5..There.is.no.need.to.reintroduce.the.white.stork.to.England.as.it.is.a.common.s
pecies.throughout.mainland.Europe."
## [114] "Q13.5_agreement_score"
## [115] "Q14.1..I.think.white.storks.are.useless.birds."
## [116] "Q14.1_agreement_score"
## [117] "Q14.2..White.storks.are.part.of.our.cultural.and.natural.heritage."
## [118] "Q14.2_agreement_score"
## [119] "Q14.3..The.reintroduced.white.stork.can.help.people..re.connect.with.the.natural.worl
d."
## [120] "Q14.3_agreement_score"
## [121] "Q14.4..The.countryside.will.be.worse.off.with.white.storks.around."
## [122] "Q14.4_agreement_score"
## [123] "OverallAttitudeScore"
## [124] "Q14.5..Overall..I.support.efforts.that.aim.to.reintroduce.the.UK.s.lost.species.and.r
estore.its.natural.systems."
## [125] "Q14.5_agreement_score"
## [126] "Q15_WSP_support"
## [127] "Q15_WSP_support_open"
## [128] "Q16_views_management"
## [129] "Q16_views_management_open"
## [130] "Q17.1_Nest_monitoring"
## [131] "Q17.2_Nesting_platforms"
## [132] "Q17.3_Discouragenestbuilding"
## [133] "Q17.4_Nest_removal"
## [134] "Q17.5_Tracking"
## [135] "Q17.6_Public_engagement"
## [136] "Q17.7_Supplementary_food"
## [137] "Q17.8_compensation_storkdamage"
## [138] "Q17.9_Stork_relocation"
## [139] "Q17.10_Culling"
## [140] "Q17_11.management.not.needed"
## [141] "Q17.12_Don.tknow"
## [142] "Q17.13_other"
## [143] "Q17.13a_other_open"
## [144] "Q18_exp_nature"
## [145] "Q18a_dogwalking"
## [146] "Q18a_walking"
## [147] "Q18a_running.cycling"
## [148] "Q18a.golf"
## [149] "Q18a.picnic"
## [150] "Q18a.horse.riding"
## [151] "Q18a.bird.wildlife.watching"
## [152] "Q18a.photography"
## [153] "Q18a.camping"
## [154] "Q18a.fishing"
## [155] "Q18a.shooting.hunting"
## [156] "Q18a.water.sports.swimming"
## [157] "Q18a.gardening"
## [158] "Q18a.don.t.spend.free.time.in.green.natural.spaces"
## [159] "Q18a_other"
## [160] "Q18a_other_open"
## [161] "Q19.1..I.find.being.in.nature.really.amazing"
## [162] "Q19.1.score"
## [163] "Q19.2..Spending.time.in.nature.is.very.important.to.me"
## [164] "Q19.2.score"
## [165] "Q19.3..Being.in.nature.makes.me.very.happy"
## [166] "Q19.3.score"
## [167] "Q19.4..I.always.find.beauty.in.nature"
## [168] "Q19.4.score"
## [169] "Q19.5..I.always.treat.nature.with.respect"
## [170] "Q19.5.score"
## [171] "Q19.6..I.feel.part.of.nature"
## [172] "Q19.6.score"
## [173] "NCI"
## [174] "Q20.1..Damage.to.the.natural.environment"
## [175] "Q20.2..The.consequences.of.a.loss.of.variety.of.wildlife"
## [176] "EnvConcern.score"
## [177] "Q21.1..When.I.see.litter..I.pick.it.up"
## [178] "Q21.1.score"
## [179] "Q21.2..I.vote.for.nature.or.wildlife.conservation.friendly.legislation.in.local.or.na
tional.referendums.votes.etc."
## [180] "Q21.2.score"
## [181] "Q21.3..I.get.in.touch.with.local.authorities.on.nature.conservation.issues"
## [182] "Q21.3.score"
## [183] "Q21.4..I.vote.for.parties..candidates.with.strong.pro.nature.conservation.policies.i
n.elections"
## [184] "Q21.4.score"
## [185] "ProCoBS"
## [186] "Q22....Are.you.a.member.of.any.environmental..wildlife.or.conservation.organisation
s."
## [187] "Q22.a..Which.ones...Optional.."
## [188] "Q23.1..I.pay.attention.to.birds.wherever.I.go."
## [189] "Q23.1..Score"
## [190] "Q23.2..I.can.identify.common.birds.in.my.area."
## [191] "Q23.2.Score"
## [192] "Q23.3..Seeing.a.new.bird.fills.me.with.excitement."
## [193] "Q23.3.Score"
## [194] "Q23.4..I.am.not.interested.in.birds."
## [195] "Q24.4.score"
## [196] "BirdInterestScore"
## [197] "Age_group"
## [198] "Gender"
## [199] "Gender_other"
## [200] "Region"
## [201] "County"
## [202] "Area_type"
## [203] "Postcode"
## [204] "ReleaseSite"
## [205] "SiteProximity"
## [206] "SiteLocal"
## [207] "Q27_Knepp_visit"
## [208] "Q27.a_Knepp_activity"
## [209] "Q27.a_Knepp_activity_other"
## [210] "Education"
## [211] "Education_other"
## [212] "Occupation"
## [213] "Occupation_short"
## [214] "Occupation_short_clean"
## [215] "Occupation_other"
## [216] "Q30_Press"
## [217] "Q30_TV.Radio"
## [218] "Q30_Facebook"
## [219] "Q30_Twitter"
## [220] "Q30_Social_media"
## [221] "Q30_Durrell.WSP"
## [222] "Q30_Other_wildlife.nature.org."
## [223] "Q30_Farming_org."
## [224] "Q30_Business_org."
## [225] "Q30_Tourism_org."
## [226] "Q30_Local_council"
## [227] "Q30_Friend.family"
## [228] "Q30_Researcher"
## [229] "Q30_other"
## [230] "Q30.a_Other_open"
## [231] "Q31_comments"
## [232] "Age_group_match"
## [233] "SecsTaken"
## [234] "Age_short"
## [235] "Education_short"
```

```r
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
```

```
# Question 19 NCI
final_data %>%
  select(., starts_with("Q19") & ends_with('score')) %>%
  psych::alpha(title = "NCI")
```

```
##
## Reliability analysis  NCI
## Call: psych::alpha(x = ., title = "NCI")
##
##   raw_alpha std.alpha G6(smc) average_r S/N    ase mean   sd median_r
##       0.95      0.95    0.95      0.76  19 0.0013  5.8 0.97     0.76
##
##  lower alpha upper     95% confidence boundaries
## 0.95 0.95 0.95
##
##  Reliability if an item is dropped:
##             raw_alpha std.alpha G6(smc) average_r S/N alpha se  var.r med.r
## Q19.1.score      0.95      0.95    0.94      0.78  18    0.0015 0.0032  0.78
## Q19.2.score      0.94      0.94    0.93      0.75  15    0.0017 0.0039  0.75
## Q19.3.score      0.93      0.93    0.92      0.74  14    0.0018 0.0026  0.74
## Q19.4.score      0.94      0.94    0.93      0.76  15    0.0017 0.0045  0.76
## Q19.5.score      0.94      0.94    0.94      0.77  17    0.0015 0.0042  0.77
## Q19.6.score      0.95      0.95    0.94      0.79  19    0.0014 0.0032  0.79
##
##  Item statistics
##                n raw.r std.r r.cor r.drop mean   sd
## Q19.1.score 3531  0.86  0.86  0.82   0.80  5.8 1.07
## Q19.2.score 3531  0.92  0.92  0.91   0.89  5.8 1.12
## Q19.3.score 3531  0.94  0.94  0.94   0.92  5.8 1.06
## Q19.4.score 3531  0.91  0.91  0.90   0.87  5.8 1.07
## Q19.5.score 3531  0.88  0.88  0.85   0.83  6.0 0.99
## Q19.6.score 3531  0.86  0.85  0.80   0.79  5.5 1.21
##
## Non missing response frequency for each item
##                1    2    3    4    5    6    7 miss
## Q19.1.score 0.01 0.01 0.01 0.06 0.25 0.38 0.29    0
## Q19.2.score 0.01 0.01 0.02 0.06 0.21 0.38 0.31    0
## Q19.3.score 0.01 0.01 0.01 0.05 0.25 0.39 0.29    0
## Q19.4.score 0.01 0.01 0.01 0.05 0.25 0.39 0.29    0
## Q19.5.score 0.01 0.01 0.01 0.03 0.20 0.43 0.32    0
## Q19.6.score 0.01 0.01 0.03 0.14 0.25 0.33 0.22    0
```

```
# 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
```

```
# 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
```

[1] University of Brighton, l.jones4@brighton.ac.uk (mailto:l.jones4@brighton.ac.uk)↩