Exp 2 R Markdown (aka Leverhulme Study 7)

Matthew Haigh & Hope Birch

25/11/2020

Load packages

```
library("qualtRics")
```

```
## Warning: package 'qualtRics' was built under R version 4.0.3
```

```
library("httr")
```

```
## Warning: package 'httr' was built under R version 4.0.3
```

```
library("tidyr")
```

```
## Warning: package 'tidyr' was built under R version 4.0.3
```

```
library("reshape2")
```

```
## Warning: package 'reshape2' was built under R version 4.0.3
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library("psych")
```

```
## Warning: package 'psych' was built under R version 4.0.3
```

```
library("yarrr")
```

```
## Warning: package 'yarrr' was built under R version 4.0.3
```

```
## Loading required package: jpeg
```

```
## Warning: package 'jpeg' was built under R version 4.0.3
```

```
## Loading required package: BayesFactor
```

```
## Warning: package 'BayesFactor' was built under R version 4.0.3
```

```
## Loading required package: coda
```

```
## Warning: package 'coda' was built under R version 4.0.3
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## ************
## Welcome to BayesFactor 0.9.12-4.2. If you have questions, please contact Richard Morey (ri
charddmorey@gmail.com).
##
## Type BFManual() to open the manual.
## ************
```

```
## Loading required package: circlize
```

```
## Warning: package 'circlize' was built under R version 4.0.3
```

```
## ========================================
## circlize version 0.4.11
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: https://jokergoo.github.io/circlize_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
##   in R. Bioinformatics 2014.
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(circlize))
## ========================================
```

```
## yarrr v0.1.5. Citation info at citation('yarrr'). Package guide at yarrr.guide()
```

```
## Email me at Nathaniel.D.Phillips.is@gmail.com
```

```
library("car")
```

```
## Warning: package 'car' was built under R version 4.0.3
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.0.3
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:psych':
##
##     logit
```

```
library("effectsize")
```

```
## Warning: package 'effectsize' was built under R version 4.0.3
```

```
##
## Attaching package: 'effectsize'
```

```
## The following object is masked from 'package:psych':
##
##     phi
```

```
library("BayesFactor")
```

FETCH THE DATA FROM OSF

The line below downloads the data csv file from oSF to your working directory, saved as “study2.csv” credit: https://twitter.com/lakens/status/839469115253325824 (https://twitter.com/lakens/status/839469115253325824)

```
data <- GET('https://osf.io/fteub/?action=download', write_disk('study2.csv', overwrite = TRU
E))
```

The line below reads the data from your working directory using the qualtRics ‘readSurvey’ function. This function strips out the text in rows 2 and 3 (among other useful things)

```
mydata=readSurvey("study2.csv")
```

```
## Warning: 'readSurvey' is deprecated.
## Use 'read_survey' instead.
## See help("Deprecated")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   StartDate = col_datetime(format = ""),
##   EndDate = col_datetime(format = ""),
##   IPAddress = col_character(),
##   RecordedDate = col_datetime(format = ""),
##   ResponseId = col_character(),
##   RecipientLastName = col_character(),
##   RecipientFirstName = col_character(),
##   RecipientEmail = col_character(),
##   ExternalReference = col_character(),
##   LocationLatitude = col_character(),
##   LocationLongitude = col_character(),
##   DistributionChannel = col_character(),
##   UserLanguage = col_character(),
##   PID = col_character(),
##   Memory_task = col_character(),
##   Memory_task_DO = col_character(),
##   Prolific_PID = col_character(),
##   FL_12_DO = col_character()
## )
## i Use `spec()` for the full column specifications.
```

CONVERT mydata to a data.frame

'mydata' is a tibble rather than a data.frame (this is due to the way that the qualtRics package Formats the data)

The code below converts mydata into a data.frame (**doing this this avoids problems further down the line)

```
mydata<- as.data.frame(mydata)
```

Count how many particpants consented to take part (this is BEFORE any exclusions for non-completion / non-serious responding)

```
length(which(mydata$Consent==1))
```

```
## [1] 412
```

OPTIONAL: the line below identifies the types of variable we have (e.g., numeric, string, character etc.) - Just to check that qualtRics readSurvey has worked properly (i.e. to check whether it has set all relevant columns to 'numeric') Remove # to run

```
#sapply(mydata, class)
```

Rename variables that have odd names

```
colnames(mydata)[colnames(mydata)=="SC0"] <- "recall_score"
colnames(mydata)[colnames(mydata)=="FL_12_DO"] <- "condition"
```

EXCLUDE PARTICIPANTS IF THEY DON'T MEET PREREGISTERED INCLUSION CRITERIA

Create a new data.frame that is a SUBSET of the original. This data.frame…

1. only includes SUBSET of participants who finished the study (Finished==1) AND declared that they answered seriously (seriousness_check==1) AND scored 4 or above on recall - i.e., excluding non-completers, non-serious responses and those who did not recall 4 or more items
2. only includes SUBSET of relevant columns required for analysis

```
mydata <- subset(mydata, Finished==1 & Serious_check==1 & recall_score>=4, select=c(Finished,
`Duration (in seconds)`, Gender, Age, Serious_check, recall_score, condition, NC_1:Developmen
t_sci_know_6))
```

Count how many participants remain after exclusions (final sample size)

```
length(which(mydata$Serious_check==1))
```

```
## [1] 400
```

Calculate mean completion time in seconds

```
mean(mydata$`Duration (in seconds)`) #653.68
```

```
## [1] 653.68
```

```
#Convert to minutes
653.68/60 #10.89467
```

```
## [1] 10.89467
```

OPTIONAL: export the data to a .csv to check it all looks in order (i.e. correct rows and columns retained? / only those who completed seriously retained?)

```
write.csv(mydata, file = "MyDataTidied.csv")
```

DEMOGRAPHICS

Count up how many pps assigned to each of the four conditions

```
length(which(mydata$condition=="Block_1_Generic_Conflict"))   #98
```

```
## [1] 98
```

```
length(which(mydata$condition=="Block_2_Generic_Consistent"))   #101
```

```
## [1] 101
```

```
length(which(mydata$condition=="Block_3_Qualified_Conflict"))    #101
```

```
## [1] 101
```

```
length(which(mydata$condition=="Block_4_Qualified_Consistent"))   #100
```

```
## [1] 100
```

Total n (after exclusions)

```
nrow(mydata)
```

```
## [1] 400
```

```
Total_n <- nrow(mydata)
```

Age

```
mean(mydata$Age,na.rm=TRUE)
```

```
## [1] 33.465
```

```
sd(mydata$Age,na.rm=TRUE)
```

```
## [1] 12.03415
```

```
min(mydata$Age,na.rm=TRUE)
```

```
## [1] 18
```

```
max(mydata$Age,na.rm=TRUE)
```

```
## [1] 73
```

Gender

```
#count of Males
length(which(mydata$Gender==1))
```

```
## [1] 150
```

```
Males<- length(which(mydata$Gender==1))
PercentMale <- (Males/Total_n)*100

#count of Females
length(which(mydata$Gender==2))
```

```
## [1] 248
```

```
Females<- length(which(mydata$Gender==2))
PercentFemale <- (Females/Total_n)*100

#count of 'Other'
length(which(mydata$Gender==3))
```

```
## [1] 2
```

```
Other<- length(which(mydata$Gender==3))
PercentOther <- (Other/Total_n)*100

#count of 'Prefer not to say'
length(which(mydata$Gender==4))
```

```
## [1] 0
```

```
Prefer_not_say <-length(which(mydata$Gender==4))
PercentPreferNot <- (Prefer_not_say/Total_n)*100

#print counts
Total_n
```

```
## [1] 400
```

```
Males
```

```
## [1] 150
```

```
Females
```

```
## [1] 248
```

```
Other
```

```
## [1] 2
```

```
Prefer_not_say
```

```
## [1] 0
```

```
#print percentages
PercentMale
```

```
## [1] 37.5
```

```
PercentFemale
```

```
## [1] 62
```

```
PercentOther
```

```
## [1] 0.5
```

```
PercentPreferNot
```

```
## [1] 0
```

Create sepearte columns to identify levels of each IV

(original 'condition' Format is "Block_1_Generic_Conflict" - this just splits into four speerate columns using underscore as delimiter - 'block' and 'number' are meaningless - Format and Conflict are the IVs)

```
mydata<- separate(data = mydata, col = condition, into = c("block", "number", "Format", "Conf
lict"))

#look at the results of this command
write.csv(mydata, file = "MyDataTidied.csv")
```

Rename Generic=Gen., Qualified=Qual. (this is for the benefit of the pirate plot below to avoid overwriting)

```
mydata$Conflict <- gsub("Conflict", "Conf.", mydata$Conflict)
mydata$Conflict <- gsub("Consistent", "Non-Conf.", mydata$Conflict)
```

Set these new IV columns as factors

```
mydata$Format <- as.factor(mydata$Format)
levels(mydata$Format)
```

```
## [1] "Generic"   "Qualified"
```

```
mydata$Conflict <- as.factor(mydata$Conflict)
levels(mydata$Conflict)
```

```
## [1] "Conf."     "Non-Conf."
```

Score the scales (reverse scoring not required as items were reverse coded within Qualtrics)

Calculated the average for each scale

```
#nutrional confusion mean
mydata$confusion <- ((mydata$NC_1 + mydata$NC_2 + mydata$NC_3 + mydata$NC_4 + mydata$NC_5 + m
ydata$NC_6)/6)

#nutrional backlash mean
mydata$backlash <- ((mydata$NBS_1 + mydata$NBS_2 + mydata$NBS_3 + mydata$NBS_4 + mydata$NBS_5
+ mydata$NBS_6)/6)

#Mistrust of expertise mean
mydata$mistrust <- ((mydata$Mistrust_expertise_1 + mydata$Mistrust_expertise_2 + mydata$Mistr
ust_expertise_3)/3)

#Certainty of knowledge mean
mydata$certainty <- ((mydata$Certainty_sci_know_1 + mydata$Certainty_sci_know_2 + mydata$Cert
ainty_sci_know_3 + mydata$Certainty_sci_know_4 + mydata$Certainty_sci_know_5 + mydata$Certain
ty_sci_know_6)/6)

#Development of knowledge mean
mydata$development <- ((mydata$Development_sci_know_1 + mydata$Development_sci_know_2 +mydata
$Development_sci_know_3 +mydata$Development_sci_know_4 +mydata$Development_sci_know_5 +mydata
$Development_sci_know_6)/6)
```

OPTIONAL: the line below identifies the types of variable we have (e.g., numeric, string, character etc.) - Just to check that qualtRics readSurvey has worked properly (i.e. to check whether it has set all relevant columns to ‘numeric’)

```
sapply(mydata, class)
```

```
##                Finished  Duration (in seconds)                 Gender 
##               "numeric"              "numeric"              "numeric" 
##                     Age          Serious_check           recall_score 
##               "numeric"              "numeric"              "numeric" 
##                   block                 number                 Format 
##             "character"            "character"               "factor" 
##                Conflict                   NC_1                   NC_2 
##                "factor"              "numeric"              "numeric" 
##                    NC_3                   NC_4                   NC_5 
##               "numeric"              "numeric"              "numeric" 
##                    NC_6                  NBS_1                  NBS_2 
##               "numeric"              "numeric"              "numeric" 
##                   NBS_3                  NBS_4                  NBS_5 
##               "numeric"              "numeric"              "numeric" 
##                   NBS_6   Mistrust_expertise_1   Mistrust_expertise_2 
##               "numeric"              "numeric"              "numeric" 
##    Mistrust_expertise_3                    GSS   Certainty_sci_know_1 
##               "numeric"              "numeric"              "numeric" 
##    Certainty_sci_know_2   Certainty_sci_know_3   Certainty_sci_know_4 
##               "numeric"              "numeric"              "numeric" 
##    Certainty_sci_know_5   Certainty_sci_know_6 Development_sci_know_1 
##               "numeric"              "numeric"              "numeric" 
## Development_sci_know_2 Development_sci_know_3 Development_sci_know_4 
##               "numeric"              "numeric"              "numeric" 
## Development_sci_know_5 Development_sci_know_6              confusion 
##               "numeric"              "numeric"              "numeric" 
##                backlash               mistrust              certainty 
##               "numeric"              "numeric"              "numeric" 
##             development 
##               "numeric"
```

Look at the results of these command (5 new columns)

```
write.csv(mydata, file = "MyDataTidied.csv")
```

Cronbachs alphas (use psych package)

```
confusion_alpha <- subset(mydata, select=c(NC_1:NC_6))
psych::alpha(confusion_alpha)
```

```
##
## Reliability analysis
## Call: psych::alpha(x = confusion_alpha)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean   sd median_r
##       0.82      0.82    0.81      0.43 4.5 0.014  2.9 0.76     0.42
##
##  lower alpha upper     95% confidence boundaries
## 0.79 0.82 0.84
##
##  Reliability if an item is dropped:
##      raw_alpha std.alpha G6(smc) average_r S/N alpha se  var.r med.r
## NC_1      0.79      0.79    0.78      0.43 3.8    0.017 0.0106  0.42
## NC_2      0.78      0.79    0.78      0.43 3.7    0.017 0.0104  0.42
## NC_3      0.79      0.79    0.78      0.43 3.8    0.017 0.0133  0.44
## NC_4      0.77      0.77    0.76      0.40 3.4    0.018 0.0121  0.36
## NC_5      0.80      0.79    0.77      0.44 3.9    0.016 0.0060  0.43
## NC_6      0.79      0.79    0.77      0.43 3.8    0.016 0.0093  0.45
##
##  Item statistics
##        n raw.r std.r r.cor r.drop mean   sd
## NC_1 400  0.73  0.71  0.63   0.58  3.2 1.18
## NC_2 400  0.74  0.73  0.65   0.59  3.2 1.09
## NC_3 400  0.71  0.71  0.62   0.57  2.3 1.04
## NC_4 400  0.78  0.78  0.73   0.66  3.2 1.09
## NC_5 400  0.68  0.70  0.64   0.54  2.8 0.97
## NC_6 400  0.69  0.71  0.64   0.55  2.6 0.94
##
## Non missing response frequency for each item
##         1    2    3    4    5 miss
## NC_1 0.08 0.31 0.07 0.46 0.08    0
## NC_2 0.06 0.25 0.19 0.42 0.09    0
## NC_3 0.20 0.51 0.12 0.14 0.03    0
## NC_4 0.05 0.24 0.23 0.36 0.11    0
## NC_5 0.05 0.42 0.28 0.22 0.04    0
## NC_6 0.07 0.51 0.22 0.17 0.02    0
```

```
backlash_alpha <- subset(mydata, select=c(NBS_1:NBS_6))
psych::alpha(backlash_alpha)
```

```
##
## Reliability analysis
## Call: psych::alpha(x = backlash_alpha)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean   sd median_r
##       0.71      0.72    0.73       0.3 2.6 0.022  2.7 0.64     0.31
##
##  lower alpha upper     95% confidence boundaries
## 0.67 0.71 0.76
##
##  Reliability if an item is dropped:
##       raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r med.r
## NBS_1      0.70      0.71    0.70      0.33 2.5    0.024 0.018  0.32
## NBS_2      0.66      0.68    0.68      0.30 2.1    0.027 0.023  0.31
## NBS_3      0.64      0.66    0.65      0.28 1.9    0.028 0.019  0.28
## NBS_4      0.69      0.70    0.68      0.31 2.3    0.025 0.015  0.33
## NBS_5      0.65      0.65    0.64      0.27 1.9    0.027 0.017  0.28
## NBS_6      0.71      0.72    0.71      0.34 2.6    0.023 0.022  0.34
##
##  Item statistics
##         n raw.r std.r r.cor r.drop mean   sd
## NBS_1 400  0.63  0.58  0.46   0.39  3.4 1.16
## NBS_2 400  0.69  0.67  0.58   0.50  2.7 1.03
## NBS_3 400  0.72  0.72  0.65   0.54  2.5 1.05
## NBS_4 400  0.57  0.63  0.53   0.40  2.1 0.81
## NBS_5 400  0.70  0.73  0.69   0.55  2.4 0.84
## NBS_6 400  0.58  0.56  0.42   0.34  3.0 1.08
##
## Non missing response frequency for each item
##          1    2    3    4    5 miss
## NBS_1 0.06 0.23 0.16 0.40 0.15    0
## NBS_2 0.10 0.43 0.23 0.20 0.04    0
## NBS_3 0.16 0.42 0.20 0.18 0.03    0
## NBS_4 0.18 0.60 0.14 0.06 0.01    0
## NBS_5 0.10 0.56 0.23 0.09 0.01    0
## NBS_6 0.07 0.27 0.26 0.33 0.07    0
```

```
mistrust_alpha <-subset(mydata, select=c(Mistrust_expertise_1:Mistrust_expertise_3))
psych::alpha(mistrust_alpha)
```

```
##
## Reliability analysis
## Call: psych::alpha(x = mistrust_alpha)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean   sd median_r
##       0.72      0.73    0.65      0.47 2.6 0.024    2 0.65     0.42
##
##  lower alpha upper     95% confidence boundaries
## 0.68 0.72 0.77
##
##  Reliability if an item is dropped:
##                      raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r
## Mistrust_expertise_1      0.59      0.59    0.42      0.42 1.5    0.041    NA
## Mistrust_expertise_2      0.59      0.59    0.42      0.42 1.5    0.041    NA
## Mistrust_expertise_3      0.71      0.72    0.56      0.56 2.6    0.028    NA
##                      med.r
## Mistrust_expertise_1  0.42
## Mistrust_expertise_2  0.42
## Mistrust_expertise_3  0.56
##
##  Item statistics
##                        n raw.r std.r r.cor r.drop mean   sd
## Mistrust_expertise_1 400  0.83  0.82  0.69   0.58  2.0 0.86
## Mistrust_expertise_2 400  0.81  0.82  0.69   0.58  1.9 0.75
## Mistrust_expertise_3 400  0.77  0.76  0.55   0.48  2.0 0.81
##
## Non missing response frequency for each item
##                         1    2    3    4    5 miss
## Mistrust_expertise_1 0.29 0.42 0.28 0.00 0.02    0
## Mistrust_expertise_2 0.34 0.47 0.19 0.00 0.00    0
## Mistrust_expertise_3 0.25 0.58 0.10 0.05 0.01    0
```

```
certainty_alpha <-subset(mydata, select=c(Certainty_sci_know_1:Certainty_sci_know_6))
psych::alpha(certainty_alpha)
```

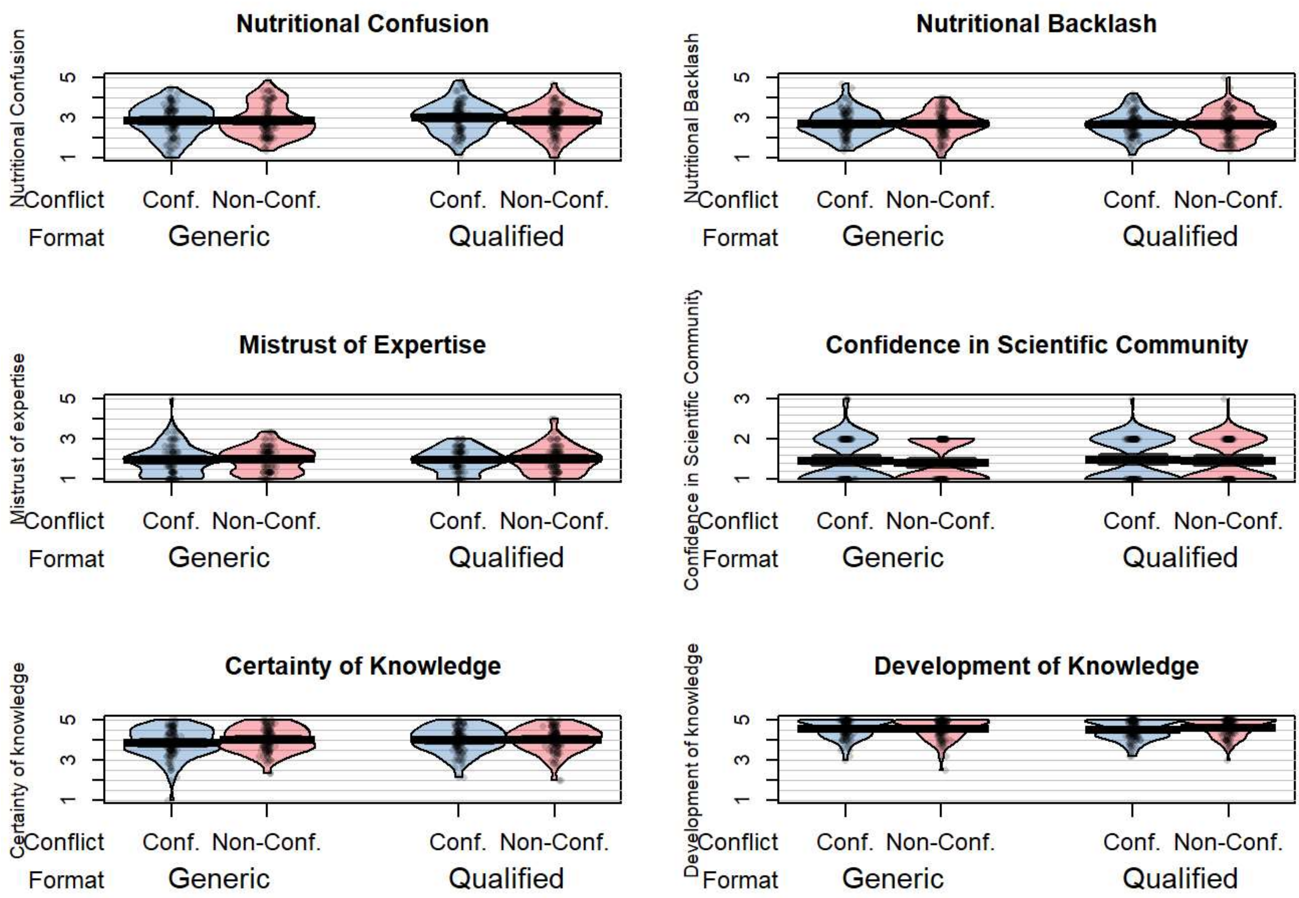
```
##
## Reliability analysis
## Call: psych::alpha(x = certainty_alpha)
##
##   raw_alpha std.alpha G6(smc) average_r S/N  ase mean   sd median_r
##       0.75      0.76    0.74      0.35 3.2 0.02    4 0.65     0.35
##
##  lower alpha upper     95% confidence boundaries
## 0.71 0.75 0.79
##
##  Reliability if an item is dropped:
##                     raw_alpha std.alpha G6(smc) average_r S/N alpha se  var.r
## Certainty_sci_know_1      0.70      0.73    0.70      0.35 2.7    0.024 0.0151
## Certainty_sci_know_2      0.76      0.76    0.73      0.39 3.2    0.019 0.0079
## Certainty_sci_know_3      0.71      0.73    0.69      0.35 2.7    0.023 0.0079
## Certainty_sci_know_4      0.71      0.73    0.70      0.35 2.7    0.023 0.0129
## Certainty_sci_know_5      0.68      0.69    0.66      0.31 2.2    0.025 0.0072
## Certainty_sci_know_6      0.70      0.72    0.69      0.34 2.6    0.023 0.0104
##                     med.r
## Certainty_sci_know_1  0.31
## Certainty_sci_know_2  0.39
## Certainty_sci_know_3  0.36
## Certainty_sci_know_4  0.33
## Certainty_sci_know_5  0.29
## Certainty_sci_know_6  0.32
##
##  Item statistics
##                       n raw.r std.r r.cor r.drop mean   sd
## Certainty_sci_know_1 400  0.68  0.67  0.57   0.51  3.9 0.99
## Certainty_sci_know_2 400  0.61  0.56  0.41   0.36  3.2 1.20
## Certainty_sci_know_3 400  0.63  0.68  0.60   0.49  4.6 0.76
## Certainty_sci_know_4 400  0.68  0.67  0.56   0.49  3.5 1.08
## Certainty_sci_know_5 400  0.75  0.77  0.74   0.62  4.3 0.86
## Certainty_sci_know_6 400  0.68  0.70  0.61   0.51  4.3 0.95
##
## Non missing response frequency for each item
##                         1    2    3    4    5 miss
## Certainty_sci_know_1 0.02 0.10 0.13 0.46 0.30    0
## Certainty_sci_know_2 0.07 0.28 0.18 0.32 0.15    0
## Certainty_sci_know_3 0.00 0.03 0.04 0.21 0.71    0
## Certainty_sci_know_4 0.02 0.19 0.24 0.35 0.19    0
## Certainty_sci_know_5 0.00 0.04 0.10 0.36 0.49    0
## Certainty_sci_know_6 0.01 0.06 0.10 0.32 0.51    0
```

```
development_alpha <-subset(mydata, select=c(Development_sci_know_1:Development_sci_know_6))
psych::alpha(development_alpha)
```

```
##
## Reliability analysis
## Call: psych::alpha(x = development_alpha)
##
##   raw_alpha std.alpha G6(smc) average_r S/N   ase mean   sd median_r
##        0.8      0.82    0.81      0.42 4.4 0.016  4.5 0.47     0.49
##
##  lower alpha upper     95% confidence boundaries
## 0.76 0.8 0.83
##
##  Reliability if an item is dropped:
##                       raw_alpha std.alpha G6(smc) average_r S/N alpha se var.r
## Development_sci_know_1      0.77      0.79    0.78      0.43 3.8    0.019 0.028
## Development_sci_know_2      0.74      0.76    0.75      0.39 3.2    0.021 0.028
## Development_sci_know_3      0.84      0.84    0.83      0.52 5.4    0.013 0.010
## Development_sci_know_4      0.73      0.75    0.74      0.38 3.0    0.022 0.024
## Development_sci_know_5      0.75      0.76    0.75      0.39 3.2    0.021 0.027
## Development_sci_know_6      0.77      0.79    0.78      0.43 3.8    0.019 0.032
##                       med.r
## Development_sci_know_1  0.48
## Development_sci_know_2  0.36
## Development_sci_know_3  0.54
## Development_sci_know_4  0.36
## Development_sci_know_5  0.40
## Development_sci_know_6  0.51
##
##  Item statistics
##                          n raw.r std.r r.cor r.drop mean   sd
## Development_sci_know_1 400  0.67  0.70  0.62   0.53  4.7 0.60
## Development_sci_know_2 400  0.79  0.80  0.76   0.68  4.5 0.63
## Development_sci_know_3 400  0.55  0.50  0.33   0.30  4.4 0.82
## Development_sci_know_4 400  0.82  0.83  0.82   0.72  4.5 0.64
## Development_sci_know_5 400  0.77  0.80  0.76   0.67  4.7 0.54
## Development_sci_know_6 400  0.71  0.70  0.62   0.54  4.4 0.73
##
## Non missing response frequency for each item
##                           1    2    3    4    5 miss
## Development_sci_know_1 0.00 0.01 0.04 0.24 0.71    0
## Development_sci_know_2 0.00 0.01 0.05 0.38 0.56    0
## Development_sci_know_3 0.01 0.03 0.06 0.30 0.60    0
## Development_sci_know_4 0.00 0.01 0.03 0.34 0.61    0
## Development_sci_know_5 0.00 0.00 0.03 0.23 0.73    0
## Development_sci_know_6 0.00 0.02 0.07 0.36 0.54    0
```

Pirate plots/descriptives

```
par(mfrow = c(3, 2)) #will pace plots in a grid with 3 rows and 2 columns


pirateplot(formula = confusion ~ Conflict*Format, data = mydata, yaxt = "n", theme=1,main =
"Nutritional Confusion", ylab = "Nutritional Confusion", cex.names = 0.75, cex.lab = 0.9, in
f.method = 'ci')
axis(2, at = seq(from = 1, to = 5, by = 1))

pirateplot(formula = backlash ~ Conflict*Format, data = mydata, yaxt = "n", theme=1,main = "N
utritional Backlash", ylab = "Nutritional Backlash", cex.names = 0.75, cex.lab = 0.9, inf.met
hod = 'ci')
axis(2, at = seq(from = 1, to = 5, by = 1))

pirateplot(formula = mistrust ~ Conflict*Format, data = mydata, yaxt = "n", theme=1,main = "M
istrust of Expertise", ylab = "Mistrust of expertise", cex.names = 0.75, cex.lab = 0.9, inf.m
ethod = 'ci')
axis(2, at = seq(from = 1, to = 5, by = 1))

pirateplot(formula = GSS ~ Conflict*Format, data = mydata, yaxt = "n", theme=1,main = "Confid
ence in Scientific Community", ylab = "Confidence in Scientific Community", cex.names = 0.75,
cex.lab = 0.9, inf.method = 'ci')
axis(2, at = seq(from = 1, to = 3, by = 1))

pirateplot(formula = certainty ~ Conflict*Format, data = mydata, yaxt = "n", theme=1,main =
"Certainty of Knowledge", ylab = "Certainty of knowledge", cex.names = 0.75, cex.lab = 0.9, i
nf.method = 'ci')
axis(2, at = seq(from = 1, to = 5, by = 1))

pirateplot(formula = development ~ Conflict*Format, data = mydata,yaxt = "n", ylim =c(1,5), t
heme=1,main = "Development of Knowledge", ylab = "Development of knowledge", cex.names = 0.75
, cex.lab = 0.9, inf.method = 'ci')
axis(2, at = seq(from = 1, to = 5, by = 1))
```




```
#Note that I added  ylim =c(1,5)  here because the default scale went from 3-5 rather than
 1-5 (presumably because there were no scores of 1 or 2)

#remember to set grid back to default (just one plot)
par(mfrow = c(1, 1))
```

Print descriptives that pirate plots are based on

```
confusion.pp <- pirateplot(formula = confusion ~ Conflict*Format,data = mydata,plot = FALSE)
confusion.pp
```

```
## $summary
##    Conflict    Format bean.num   n      avg   inf.lb   inf.ub
## 1     Conf.   Generic        1  98 2.852041 2.684187 3.015607
## 2 Non-Conf.   Generic        2 101 2.823432 2.663834 2.977843
## 3     Conf. Qualified        3 101 2.985149 2.835265 3.128341
## 4 Non-Conf. Qualified        4 100 2.843333 2.699467 2.983947
##
## $avg.line.fun
## [1] "mean"
##
## $inf.method
## [1] "hdi"
##
## $inf.p
## [1] 0.95
```

```
backlash.pp <- pirateplot(formula = backlash ~ Conflict*Format,data = mydata,plot = FALSE)
backlash.pp
```

```
## $summary
##    Conflict    Format bean.num   n      avg   inf.lb   inf.ub
## 1     Conf.   Generic        1  98 2.685374 2.558763 2.823475
## 2 Non-Conf.   Generic        2 101 2.676568 2.545647 2.793039
## 3     Conf. Qualified        3 101 2.660066 2.560322 2.791128
## 4 Non-Conf. Qualified        4 100 2.653333 2.505677 2.795170
##
## $avg.line.fun
## [1] "mean"
##
## $inf.method
## [1] "hdi"
##
## $inf.p
## [1] 0.95
```

```
mistrust.pp <- pirateplot(formula = mistrust ~ Conflict*Format,data = mydata,plot = FALSE)
mistrust.pp
```

```
## $summary
##    Conflict    Format bean.num   n      avg   inf.lb   inf.ub
## 1     Conf.   Generic        1  98 1.945578 1.787618 2.083879
## 2 Non-Conf.   Generic        2 101 1.980198 1.860401 2.111055
## 3     Conf. Qualified        3 101 1.957096 1.850338 2.064790
## 4 Non-Conf. Qualified        4 100 1.990000 1.852332 2.124149
##
## $avg.line.fun
## [1] "mean"
##
## $inf.method
## [1] "hdi"
##
## $inf.p
## [1] 0.95
```

```
GSS.pp <- pirateplot(formula = GSS ~ Conflict*Format,data = mydata,plot = FALSE)
GSS.pp
```

```
## $summary
##    Conflict    Format bean.num   n      avg   inf.lb   inf.ub
## 1     Conf.   Generic        1  98 1.448980 1.346082 1.566233
## 2 Non-Conf.   Generic        2 101 1.405941 1.307677 1.502366
## 3     Conf. Qualified        3 101 1.485149 1.389257 1.589146
## 4 Non-Conf. Qualified        4 100 1.460000 1.355277 1.556081
##
## $avg.line.fun
## [1] "mean"
##
## $inf.method
## [1] "hdi"
##
## $inf.p
## [1] 0.95
```

```
certainty.pp <- pirateplot(formula = certainty ~ Conflict*Format,data = mydata,plot = FALSE)
certainty.pp
```

```
## $summary
##    Conflict    Format bean.num   n      avg   inf.lb   inf.ub
## 1     Conf.   Generic        1  98 3.836735 3.687311 3.977783
## 2 Non-Conf.   Generic        2 101 3.993399 3.866573 4.110793
## 3     Conf. Qualified        3 101 3.985149 3.863675 4.114646
## 4 Non-Conf. Qualified        4 100 4.010000 3.878716 4.116514
##
## $avg.line.fun
## [1] "mean"
##
## $inf.method
## [1] "hdi"
##
## $inf.p
## [1] 0.95
```

```
development.pp <- pirateplot(formula = development ~ Conflict*Format,data = mydata,plot = FAL
SE)
development.pp
```

```
## $summary
##    Conflict    Format bean.num   n      avg   inf.lb   inf.ub
## 1     Conf.   Generic        1  98 4.554422 4.459310 4.638713
## 2 Non-Conf.   Generic        2 101 4.534653 4.445220 4.631630
## 3     Conf. Qualified        3 101 4.486799 4.393987 4.581234
## 4 Non-Conf. Qualified        4 100 4.601667 4.510769 4.685083
##
## $avg.line.fun
## [1] "mean"
##
## $inf.method
## [1] "hdi"
##
## $inf.p
## [1] 0.95
```

Inferential stats 2x2 ANOVA (using car package: car uses Type 3 SS to be consistent with SPSS)

https://www.rdocumentation.org/packages/car/versions/1.0-7/topics/Anova (https://www.rdocumentation.org/packages/car/versions/1.0-7/topics/Anova)
Conflict and Format were set as.factors earlier

```
#this line turns off scientific notation for p values
options(scipen=999)
```

This guide tells us how to get the same output values as SPSS. It involves changing the default contrast settings and using type III sum of squares http://www.statscanbefun.com/rblog/2015/8/27/ensuring-r-generates-the-same-anova-f-values-as-spss (http://www.statscanbefun.com/rblog/2015/8/27/ensuring-r-generates-the-same-anova-f-values-as-spss) It tells us change default contrasts using the command below

```
options(contrasts = c("contr.helmert", "contr.poly"))
```

Nutritional confusion ANOVA

```
confusionlm <- lm(confusion ~ Conflict*Format, data = mydata)   #we first create a linear mod
el
Anova(confusionlm,type=3)    #we then use the car package calculates Anova (with type 3 SS) o
n the linear model we have just created
```

```
## Anova Table (Type III tests)
##
## Response: confusion
##                 Sum Sq  Df   F value              Pr(>F)
## (Intercept)     3308.0   1 5711.2473 <0.0000000000000002 ***
## Conflict           0.7   1    1.2534              0.2636
## Format             0.6   1    1.0103              0.3154
## Conflict:Format    0.3   1    0.5531              0.4575
## Residuals        229.4 396
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
cohens_f(confusionlm, partial = TRUE, ci = 0.95, squared = FALSE, model2 = NULL)
```

```
## Parameter       | Cohen's f (partial) |       95% CI
## ---------------------------------------------------
## Conflict        |                0.06 | [0.00, 0.16]
## Format          |                0.05 | [0.00, 0.15]
## Conflict:Format |                0.04 | [0.00, 0.14]
```

Bayesian nutritional confusion ANOVA

```
confusion_bayes <- anovaBF(confusion ~ Conflict*Format, data = mydata)
confusion_bayes # lists the Bayes factor for each model against the null model
```

```
## Bayes factor analysis
## --------------
## [1] Format                                : 0.1816163   ±0%
## [2] Conflict                              : 0.2058719   ±0%
## [3] Format + Conflict                     : 0.0381473   ±8.01%
## [4] Format + Conflict + Format:Conflict : 0.007427427 ±1.55%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
#whichModels = "all" computes Bayes factors for all models
confusion_bayes_all <- anovaBF(confusion ~ Conflict*Format, whichModels = "all", data = mydat
a)
confusion_bayes_all
```

```
## Bayes factor analysis
## --------------
## [1] Format                                : 0.1816163   ±0%
## [2] Conflict                              : 0.2058719   ±0%
## [3] Format:Conflict                       : 0.1446205   ±0%
## [4] Format + Conflict                     : 0.03675913  ±1.13%
## [5] Format + Format:Conflict              : 0.03500037  ±2.04%
## [6] Conflict + Format:Conflict            : 0.04089063  ±1.13%
## [7] Format + Conflict + Format:Conflict : 0.007275424 ±1.55%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

Nutritional backlash ANOVA

```
backlashlm <- lm(backlash ~ Conflict*Format, data = mydata)
Anova(backlashlm,type=3)
```

```
## Anova Table (Type III tests)
##
## Response: backlash
##                  Sum Sq  Df   F value              Pr(>F)
## (Intercept)     2848.64   1 6838.3381 <0.0000000000000002 ***
## Conflict           0.01   1    0.0145              0.9043
## Format             0.06   1    0.1414              0.7071
## Conflict:Format    0.00   1    0.0003              0.9872
## Residuals        164.96 396
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
cohens_f(backlashlm, partial = TRUE, ci = 0.95, squared = FALSE, model2 = NULL)
```

```
## Parameter       | Cohen's f (partial) |       95% CI
## ---------------------------------------------------
## Conflict        |            5.86e-03 | [0.00, 0.08]
## Format          |                0.02 | [0.00, 0.11]
## Conflict:Format |            8.07e-04 | [0.00, 0.00]
```

Bayesian nutritional backlash ANOVA

```
backlash_bayes <- anovaBF(backlash ~ Conflict*Format, data = mydata)
backlash_bayes # lists the Bayes factor for each model against the null model
```

```
## Bayes factor analysis
## --------------
## [1] Format                              : 0.1185148   ±0%
## [2] Conflict                            : 0.1114378   ±0%
## [3] Format + Conflict                   : 0.01261819  ±1.6%
## [4] Format + Conflict + Format:Conflict : 0.001907912 ±1.91%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
#whichModels = "all" computes Bayes factors for all models
backlash_bayes_all <- anovaBF(backlash ~ Conflict*Format, whichModels = "all", data = mydata)
backlash_bayes_all
```

```
## Bayes factor analysis
## --------------
## [1] Format                              : 0.1185148   ±0%
## [2] Conflict                            : 0.1114378   ±0%
## [3] Format:Conflict                     : 0.1107198   ±0%
## [4] Format + Conflict                   : 0.01286368  ±1.85%
## [5] Format + Format:Conflict            : 0.01824676  ±1.16%
## [6] Conflict + Format:Conflict          : 0.01710964  ±1.16%
## [7] Format + Conflict + Format:Conflict : 0.001978458 ±2.59%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

Mistrust ANOVA

```
mistrustlm <- lm(mistrust ~ Conflict*Format, data = mydata)
Anova(mistrustlm,type=3)
```

```
## Anova Table (Type III tests)
##
## Response: mistrust
##                  Sum Sq  Df   F value              Pr(>F)
## (Intercept)     1549.32   1 3648.9037 <0.0000000000000002 ***
## Conflict           0.11   1    0.2684              0.6047
## Format             0.01   1    0.0268              0.8701
## Conflict:Format    0.00   1    0.0002              0.9895
## Residuals        168.14 396
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
cohens_f(mistrustlm, partial = TRUE, ci = 0.95, squared = FALSE, model2 = NULL)
```

```
## Parameter       | Cohen's f (partial) |       95% CI
## -------------------------------------------------------
## Conflict        |                0.03 | [0.00, 0.12]
## Format          |            8.22e-03 | [0.00, 0.09]
## Conflict:Format |            6.61e-04 | [0.00, 0.00]
```

Bayesian mistrust ANOVA

```
mistrust_bayes <- anovaBF(mistrust ~ Conflict*Format, data = mydata)
mistrust_bayes # lists the Bayes factor for each model against the null model
```

```
## Bayes factor analysis
## --------------
## [1] Format                            : 0.1120603   ±0%
## [2] Conflict                          : 0.1260003   ±0%
## [3] Format + Conflict                 : 0.01366692  ±1.5%
## [4] Format + Conflict + Format:Conflict : 0.002035087 ±2.26%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
#whichModels = "all" computes Bayes factors for all models
mistrust_bayes_all <- anovaBF(mistrust ~ Conflict*Format, whichModels = "all", data = mydata)
mistrust_bayes_all
```

```
## Bayes factor analysis
## --------------
## [1] Format                               : 0.1120603   ±0%
## [2] Conflict                             : 0.1260003   ±0%
## [3] Format:Conflict                      : 0.110715    ±0%
## [4] Format + Conflict                    : 0.0142403   ±3.75%
## [5] Format + Format:Conflict             : 0.01693375  ±2.28%
## [6] Conflict + Format:Conflict           : 0.02528776  ±21.01%
## [7] Format + Conflict + Format:Conflict : 0.002053152 ±2.34%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

Confidence ANOVA

```
GSSlm <- lm(GSS ~ Conflict*Format, data = mydata)
Anova(GSSlm,type=3)
```

```
## Anova Table (Type III tests)
##
## Response: GSS
##                 Sum Sq  Df   F value              Pr(>F)
## (Intercept)     840.89   1 3064.2893 <0.0000000000000002 ***
## Conflict          0.12   1    0.4235              0.5156
## Format            0.20   1    0.7416              0.3897
## Conflict:Format   0.01   1    0.0292              0.8645
## Residuals       108.67 396
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
cohens_f(GSSlm, partial = TRUE, ci = 0.95, squared = FALSE, model2 = NULL)
```

```
## Parameter       | Cohen's f (partial) |       95% CI
## ----------------------------------------------------
## Conflict        |                0.03 | [0.00, 0.13]
## Format          |                0.04 | [0.00, 0.14]
## Conflict:Format |            8.58e-03 | [0.00, 0.09]
```

Bayesian confidence ANOVA

```
GSS_bayes <- anovaBF(GSS ~ Conflict*Format, data = mydata)
GSS_bayes # lists the Bayes factor for each model against the null model
```

```
## Bayes factor analysis
## --------------
## [1] Format                               : 0.1595548   ±0%
## [2] Conflict                             : 0.136581    ±0%
## [3] Format + Conflict                    : 0.02134716  ±1.82%
## [4] Format + Conflict + Format:Conflict : 0.003320568 ±3.25%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
#whichModels = "all" computes Bayes factors for all models
GSS_bayes_all <- anovaBF(GSS ~ Conflict*Format, whichModels = "all", data = mydata)
GSS_bayes_all
```

```
## Bayes factor analysis
## --------------
## [1] Format                               : 0.1595548   ±0%
## [2] Conflict                             : 0.136581    ±0%
## [3] Format:Conflict                      : 0.1123029   ±0%
## [4] Format + Conflict                    : 0.02113355  ±3.17%
## [5] Format + Format:Conflict             : 0.02371264  ±1.37%
## [6] Conflict + Format:Conflict           : 0.02055974  ±1.43%
## [7] Format + Conflict + Format:Conflict : 0.003354887 ±4.22%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

Certainty ANOVA

```
certaintylm <- lm(certainty ~ Conflict*Format, data = mydata)
Anova(certaintylm,type=3)
```

```
## Anova Table (Type III tests)
##
## Response: certainty
##                 Sum Sq  Df    F value              Pr(>F)
## (Intercept)     6260.0   1 14727.4682 <0.0000000000000002 ***
## Conflict           0.8   1     1.9376              0.1647
## Format             0.7   1     1.6013              0.2065
## Conflict:Format    0.4   1     1.0217              0.3127
## Residuals        168.3 396
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
cohens_f(certaintylm, partial = TRUE, ci = 0.95, squared = FALSE, model2 = NULL)
```

```
## Parameter       | Cohen's f (partial) |       95% CI
## ---------------------------------------------------
## Conflict        |                0.07 | [0.00, 0.17]
## Format          |                0.06 | [0.00, 0.16]
## Conflict:Format |                0.05 | [0.00, 0.15]
```

Bayesian certainty ANOVA

```
certainty_bayes <- anovaBF(certainty ~ Conflict*Format, data = mydata)
certainty_bayes # lists the Bayes factor for each model against the null model
```

```
## Bayes factor analysis
## --------------
## [1] Format                                : 0.2337549  ±0%
## [2] Conflict                              : 0.2747547  ±0%
## [3] Format + Conflict                     : 0.06541597 ±1.1%
## [4] Format + Conflict + Format:Conflict : 0.01609029 ±3.08%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
#whichModels = "all" computes Bayes factors for all models
certainty_bayes_all <- anovaBF(certainty ~ Conflict*Format, whichModels = "all", data = mydat
a)
certainty_bayes_all
```

```
## Bayes factor analysis
## --------------
## [1] Format                                : 0.2337549  ±0%
## [2] Conflict                              : 0.2747547  ±0%
## [3] Format:Conflict                       : 0.1785914  ±0%
## [4] Format + Conflict                     : 0.06320864 ±1.77%
## [5] Format + Format:Conflict              : 0.05516146 ±1.45%
## [6] Conflict + Format:Conflict            : 0.06818002 ±1.1%
## [7] Format + Conflict + Format:Conflict : 0.01543118 ±2.75%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

Development ANOVA

```
developmentlm <- lm(development ~ Conflict*Format, data = mydata)
Anova(developmentlm,type=3)
```

```
## Anova Table (Type III tests)
##
## Response: development
##                 Sum Sq  Df    F value              Pr(>F)
## (Intercept)     8259.3   1 37458.5214 <0.0000000000000002 ***
## Conflict           0.2   1     1.0253              0.3119
## Format             0.0   1     0.0000              0.9948
## Conflict:Format    0.5   1     2.0550              0.1525
## Residuals         87.3 396
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
cohens_f(developmentlm, partial = TRUE, ci = 0.95, squared = FALSE, model2 = NULL)
```

```
## Parameter       | Cohen's f (partial) |       95% CI
## ---------------------------------------------------
## Conflict        |                0.05 | [0.00, 0.15]
## Format          |            4.12e-05 | [0.00, 0.00]
## Conflict:Format |                0.07 | [0.00, 0.17]
```

Bayesian development ANOVA

```
development_bayes <- anovaBF(development ~ Conflict*Format, data = mydata)
development_bayes # lists the Bayes factor for each model against the null model
```

```
## Bayes factor analysis
## --------------
## [1] Format                            : 0.1107107   ±0%
## [2] Conflict                          : 0.1828534   ±0%
## [3] Format + Conflict                 : 0.02020768  ±3.55%
## [4] Format + Conflict + Format:Conflict : 0.008224144 ±1.54%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

```
#whichModels = "all" computes Bayes factors for all models
development_bayes_all <- anovaBF(development ~ Conflict*Format, whichModels = "all", data = m
ydata)
development_bayes_all
```

```
## Bayes factor analysis
## --------------
## [1] Format                              : 0.1107107   ±0%
## [2] Conflict                            : 0.1828534   ±0%
## [3] Format:Conflict                     : 0.3009226   ±0%
## [4] Format + Conflict                   : 0.01953052  ±2.19%
## [5] Format + Format:Conflict            : 0.04541848  ±3.65%
## [6] Conflict + Format:Conflict          : 0.07194554  ±2.32%
## [7] Format + Conflict + Format:Conflict : 0.007847729 ±2.3%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

ORDINAL REGRESSION

Load packages

```
library("foreign")
```

```
## Warning: package 'foreign' was built under R version 4.0.3
```

```
library("ggplot2")
```

```
## Warning: package 'ggplot2' was built under R version 4.0.3
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:yarrr':
##
##     diamonds
```

```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
library("MASS")
```

```
## Warning: package 'MASS' was built under R version 4.0.3
```

```
library("Hmisc")
```

```
## Warning: package 'Hmisc' was built under R version 4.0.3
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Warning: package 'Formula' was built under R version 4.0.3
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following object is masked from 'package:psych':
##
##     describe
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library("reshape2")
```

Assign a label to the value

```
mydata$GSSvalue <- ordered(mydata$GSS,
                                 levels = c(1, 2, 3),
                                 labels = c("A great deal", "Only some", "Hardly any"))
```

Specify reference category

```
mydata$Format <- relevel(mydata$Format, ref = "Generic")
mydata$Conflict <- relevel(mydata$Conflict, ref = "Conf.")
```

GSS ordinal regression

```
GSS_or <- polr(as.factor(GSSvalue) ~ Conflict * Format, data = mydata, Hess = TRUE)
GSS_or #print summary
```

```
## Call:
## polr(formula = as.factor(GSSvalue) ~ Conflict * Format, data = mydata,
##     Hess = TRUE)
##
## Coefficients:
##         Conflict1           Format1 Conflict1:Format1
##     -0.0502735859      0.0953639876      0.0009908555
##
## Intercepts:
## A great deal|Only some   Only some|Hardly any
##              0.2522736              4.3754038
##
## Residual Deviance: 592.4919
## AIC: 602.4919
```

Compute confusion table and misclassification error The confusion matrix shows the performance of the ordinal logistic regression model

```
predict_GSS = predict(GSS_or, mydata)
table(mydata$GSSvalue, predict_GSS)
```

```
##               predict_GSS
##                A great deal Only some Hardly any
##   A great deal          225         0          0
##   Only some             170         0          0
##   Hardly any              5         0          0
```

```
mean(as.character(mydata$GSSvalue) != as.character(predict_GSS))
```

```
## [1] 0.4375
```

```
#store table
(GSS_table <- coef(summary(GSS_or)))
```

```
##                                Value Std. Error      t value
## Conflict1              -0.0502735859  0.1004958 -0.500255439
## Format1                 0.0953639876  0.1005019  0.948877914
## Conflict1:Format1       0.0009908555  0.1004935  0.009859894
## A great deal|Only some  0.2522736311  0.1009714  2.498466062
## Only some|Hardly any    4.3754037703  0.4501421  9.720049454
```

Calculate and store p values

```
GSS_p <- pnorm(abs(GSS_table[, "t value"]), lower.tail = FALSE) * 2

#combined table
(GSS_table <- cbind(GSS_table, "p value" = GSS_p))
```

```
##                                Value Std. Error      t value
## Conflict1              -0.0502735859  0.1004958 -0.500255439
## Format1                 0.0953639876  0.1005019  0.948877914
## Conflict1:Format1       0.0009908555  0.1004935  0.009859894
## A great deal|Only some  0.2522736311  0.1009714  2.498466062
## Only some|Hardly any    4.3754037703  0.4501421  9.720049454
##                                             p value
## Conflict1              0.6168952265843190563288089834
## Format1                0.3426827089662082825860522917
## Conflict1:Format1      0.9921330702994372030545378038
## A great deal|Only some 0.0124732085221226033505681485
## Only some|Hardly any   0.0000000000000000000002476612
```

```
(GSS_ci <- confint(GSS_or)) #default method gives profiled CIs
```

```
## Waiting for profiling to be done...
```

```
##                          2.5 %    97.5 %
## Conflict1         -0.2475524 0.1467252
## Format1           -0.1014448 0.2928544
## Conflict1:Format1 -0.1961688 0.1981001
```

```
confint.default(GSS_or) #CIs assuming normality
```

```
##                          2.5 %    97.5 %
## Conflict1         -0.2472418 0.1466946
## Format1           -0.1016160 0.2923440
## Conflict1:Format1 -0.1959728 0.1979546
```

Odds ratios

```
exp(coef(GSS_or))
```

```
##         Conflict1           Format1 Conflict1:Format1
##         0.9509692         1.1000592         1.0009913
```

OR and CI

```
exp(cbind(OR = coef(GSS_or), GSS_ci))
```

```
##                          OR     2.5 %   97.5 %
## Conflict1         0.9509692 0.7807093 1.158036
## Format1           1.1000592 0.9035311 1.340248
## Conflict1:Format1 1.0009913 0.8218735 1.219084
```