

WMO Analysis

Theodros H.

3/6/2019

Communicaiton efficiency for images (color naming theory)

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --
## v ggplot2 3.3.6      v purrr  0.3.4
## v tibble  3.1.7      v dplyr  1.0.9
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   2.1.2      v forcats 0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(matlab)

##
## Attaching package: 'matlab'

## The following object is masked from 'package:stats':
##
##   reshape

## The following objects are masked from 'package:utils':
##
##   find, fix

## The following object is masked from 'package:base':
##
##   sum

dat.names <- paste0('./CollinsNaming/Data/', dir('CollinsNaming/Data/', pattern = 'CollinsNaming'))
subjects =
  dat.names %>%
  str_extract(pattern = '.....txt') %>%
  str_remove_all(pattern = '.txt')

all_dat <-
  read_delim(dat.names, col_names = F) %>%
  set_names(c('condition', 'category', 'item', 'response')) %>%
  filter(condition != 'namingprac') %>%
  dplyr::mutate(names = case_when(str_detect(response, pattern = "[:alpha:]" ) ~ 1,
                                TRUE ~ 0
                                )) %>%
```

```

filter(names==1) %>%
mutate(response = str_remove_all(response, pattern = "[0-9:\\\" ]"),
       .keep='unused') %>%
select(-names)

```

```

## Warning: One or more parsing issues, see `problems()` for details

## Rows: 43434 Columns: 4
## -- Column specification -----
## Delimiter: " "
## chr (4): X1, X2, X3, X4
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

```

```

optimalresp_freq <-
  all_dat %>%
  group_by(category, item) %>%
  #filter(item=='camel')
  count(response) %>%
  filter( response==item) %>%

  group_by(category, item) %>%
  summarize(proportion=n/(length(subjects)*2))

```

```

## `summarise()` has grouped output by 'category'. You can override using the
## `.groups` argument.

```

```

optimalresp_freq %>%
  group_by(category) %>%
  summarise(m=mean(proportion))

```

```

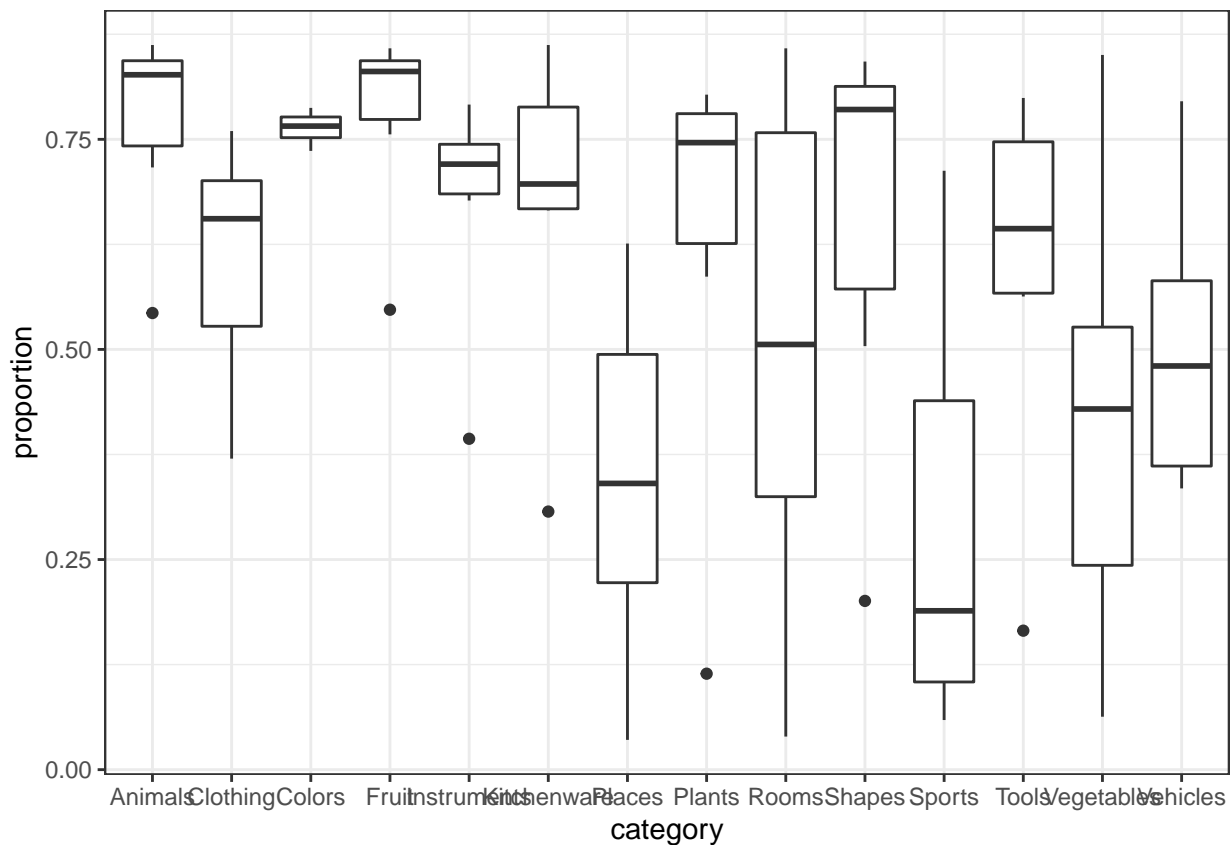
## # A tibble: 14 x 2
##   category      m
##   <chr>      <dbl>
## 1 Animals    0.770
## 2 Clothing   0.607
## 3 Colors     0.764
## 4 Fruit      0.778
## 5 Instruments 0.675
## 6 Kitchenware 0.673
## 7 Places     0.345
## 8 Plants     0.631
## 9 Rooms      0.503
## 10 Shapes    0.656
## 11 Sports    0.291
## 12 Tools     0.596
## 13 Vegetables 0.419
## 14 Vehicles  0.505

```

```

optimalresp_freq %>%
  ggplot(aes(y=proportion, x=category))+
  geom_boxplot() +
  theme_bw()

```



```
response_diversity <-
all_dat %>%
  group_by(category, item) %>%
  count(response) %>%
  #filter(n>1) %>%
  group_by(category, item) %>%
  summarise(resp_unique = numel(unique(response)),
            prop = resp_unique/(length(subjects)*2)
  )
```

`summarise()` has grouped output by 'category'. You can override using the
`.groups` argument.

```
response_diversity %>%
  group_by(category) %>%
  summarise(m=mean(scale(resp_unique)))
```

```
## # A tibble: 14 x 2
##   category      m
##   <chr>      <dbl>
## 1 Animals  1.57e-16
## 2 Clothing  3.15e-16
## 3 Colors   7.24e-16
## 4 Fruit    2.23e-16
## 5 Instruments 1.57e-16
## 6 Kitchenware -2.41e-16
## 7 Places   -2.36e-16
## 8 Plants   -5.00e-16
```

```
## 9 Rooms      3.61e-16
## 10 Shapes    -1.85e-17
## 11 Sports    -2.59e-16
## 12 Tools     -9.26e-18
## 13 Vegetables -1.50e-16
## 14 Vehicles   2.04e-16
```

```
response_diversity %>%
  ggplot(aes(y=resp_unique,x=item)) +
  geom_boxplot()
```

