Exploratory Data Analysis- Markdown Homework

Tina Vo

3/3/2021

This project has two raw data files at different scales from a study of infants, children, and adults watching a series of 7 video clips. I wrote Steps 1 and 2 to import and merge the data, and kept them there for your reference. Skip down to Step 3 to work on EDA

### SOURCE DESCRIPTION

## FILE 1: auc.csv

Columns:

* stim (stimulus video, levels/labels provided below)
* id (unique participant identifier)
* age (in days)
* AUC\_sal (area-under-the-curve for a saliency model)
* AUC\_dist (area-under-the-curve for a distance model)

AUC values indicate how well each model predicted where participants looked when watching a video.AUC values can range from 0-1 where .5 is chance and 1 is perfect prediction

## FILE 2: participants\_info\_full\_headers.csv

Columns:

* id (unique participant identifier, matches auc.csv)
* age\_group (a categorical age variable with levels:
* “.5-1 y” “1-1.5 y” “1.5-2 y” “2-4 y” “4-6 y” “8-10 y” “adult” \* precision (a quality measure of the eye data, smaller is better)
* 7 columns of “Seen X” the stimulus video before the study coded as SEEN (1), NOT SEEN (2), NOT SURE (3)

### STEP 1: Read in AUC data

Code stim as a factor

library(tidyverse)  
library(here)  
library(knitr)  
auc <- read\_csv(here("data\_raw", "auc\_bystim.csv"))  
stim\_levels <- 1:7  
stim\_labels <- c("Fallon","Feist","Pentatonix","Science","Rube","Plane","Dogs")  
auc <- auc %>% mutate(stim = factor(stim, levels = stim\_levels, labels = stim\_labels))

### STEP 2: Read in participant info data

Wrangle the ppt info data so that you can merge it into the auc data. Drop any data where the AUC values are missing. In the final, merged data, make the watched variable is coded as a factor with levels “seen” (1), “not seen” (2), “not sure” (3).  
Write the cleaned file to data\_cleaned/

Read in the ppt data and rename columns to be easier to work with

ppt <- read\_csv(here("data\_raw","participants\_info\_full\_headers.csv")) %>%   
 rename(id = `participant ID`,  
 age\_group = `Age group`,  
 precision = "Precision")

Each question about watching each video is a column, so pivot\_longer. Use separate to get just the video name into it’s own column.

ppt\_long <- ppt %>% pivot\_longer(cols = starts\_with("Seen"), names\_to = "stim", values\_to = "watched")  
ppt\_long <- ppt\_long %>% separate(stim, into = c(NA, "stim"))

Code stim and watched as factors

ppt\_long <- ppt\_long %>% mutate(  
 stim = factor(stim, levels = stim\_labels, labels = stim\_labels),  
 watched = factor(watched, levels = 1:3, labels = c("Yes","No","Not Sure")))

Join the ppt data to the AUC data (by id and by stim since each participant has observations for each stim)

ds <- left\_join(auc, ppt\_long, by = c("id", "stim"))  
ds <- ds %>% drop\_na(AUC\_sal:AUC\_dist) #Drop participants for whom we don't have data for the DV

Write the data to file

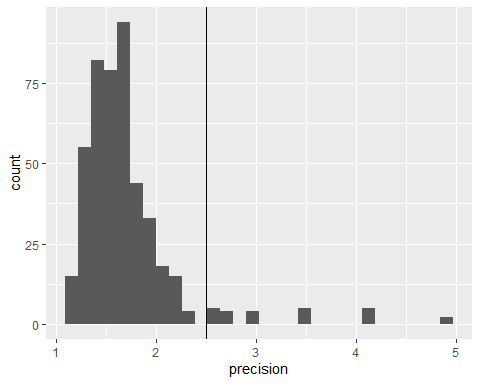
ds %>% write\_csv(here("data\_cleaned","cleaned.csv"))

### STEP 3: EXPLORATORY DATA ANALYSIS

## 3A PRECISION

Visualize the distribution of precision to see if there are values above 2.5

ds %>% ggplot(aes(x = precision)) + geom\_histogram() + geom\_vline(xintercept = 2.5)



##`stat\_bin()` using `bins=30`. Pick better value with `binwidth`.

Create a summary to figure out which participants would we need to exclude if > 2.5 meant the data are unuseable?

ds %>% group\_by(id, age\_group) %>%   
 summarize(precision = mean(precision, na.RM = T)) %>%   
 filter(precision > 2.5) %>% kable()

|  |  |  |
| --- | --- | --- |
| id | age\_group | precision |
| 52 | 1-1.5 y | 3.025000 |
| 78 | .5-1 y | 3.450000 |
| 79 | .5-1 y | 4.890000 |
| 81 | .5-1 y | 2.680000 |
| 84 | 4-6 y | 2.635714 |
| 108 | .5-1 y | 4.100000 |

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

ds %>% filter(precision > 2.5) %>% kable()

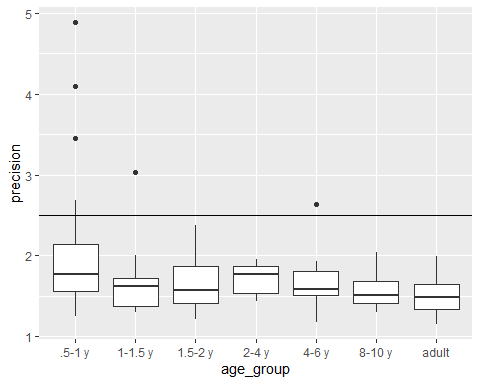
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| stim | id | age | AUC\_sal | AUC\_dist | age\_group | precision | watched |
| Feist | 52 | 539 | 0.54673 | 0.50661 | 1-1.5 y | 3.025000 | No |
| Science | 52 | 539 | 0.44698 | 0.35551 | 1-1.5 y | 3.025000 | No |
| Plane | 52 | 539 | 0.49770 | 0.48446 | 1-1.5 y | 3.025000 | No |
| Dogs | 52 | 539 | 0.66759 | 0.47044 | 1-1.5 y | 3.025000 | No |
| Feist | 78 | 198 | 0.50675 | 0.44941 | .5-1 y | 3.450000 | No |
| Pentatonix | 78 | 198 | 0.57668 | 0.57997 | .5-1 y | 3.450000 | No |
| Science | 78 | 198 | 0.75457 | 0.82130 | .5-1 y | 3.450000 | No |
| Plane | 78 | 198 | 0.61133 | 0.55452 | .5-1 y | 3.450000 | No |
| Dogs | 78 | 198 | 0.49372 | 0.48506 | .5-1 y | 3.450000 | No |
| Feist | 79 | 226 | 0.60023 | 0.59731 | .5-1 y | 4.890000 | No |
| Pentatonix | 79 | 226 | 0.60949 | 0.64914 | .5-1 y | 4.890000 | No |
| Feist | 81 | 285 | 0.61159 | 0.58766 | .5-1 y | 2.680000 | No |
| Pentatonix | 81 | 285 | 0.58287 | 0.48083 | .5-1 y | 2.680000 | Yes |
| Plane | 81 | 285 | 0.53034 | 0.54342 | .5-1 y | 2.680000 | No |
| Dogs | 81 | 285 | 0.56804 | 0.56273 | .5-1 y | 2.680000 | No |
| Feist | 84 | 1757 | 0.57606 | 0.77110 | 4-6 y | 2.635714 | Yes |
| Pentatonix | 84 | 1757 | 0.54066 | 0.63486 | 4-6 y | 2.635714 | No |
| Science | 84 | 1757 | 0.65263 | 0.64523 | 4-6 y | 2.635714 | No |
| Plane | 84 | 1757 | 0.58816 | 0.57367 | 4-6 y | 2.635714 | No |
| Dogs | 84 | 1757 | 0.55541 | 0.58090 | 4-6 y | 2.635714 | No |
| Feist | 108 | 194 | 0.62370 | 0.58747 | .5-1 y | 4.100000 | No |
| Pentatonix | 108 | 194 | 0.57016 | 0.61872 | .5-1 y | 4.100000 | No |
| Science | 108 | 194 | 0.68565 | 0.76453 | .5-1 y | 4.100000 | No |
| Plane | 108 | 194 | 0.55109 | 0.57446 | .5-1 y | 4.100000 | No |
| Dogs | 108 | 194 | 0.58014 | 0.65896 | .5-1 y | 4.100000 | No |

Use a summary table and plots to investigate whether data equally precise for participants of different ages

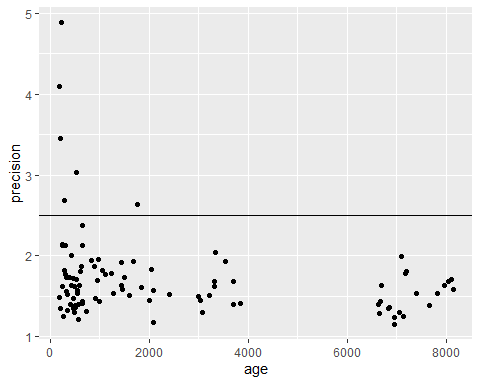
ds %>% group\_by(age\_group) %>% summarize(across(precision, list(M = mean, MIN = min, MAX = max))) %>% kable()

|  |  |  |  |
| --- | --- | --- | --- |
| age\_group | precision\_M | precision\_MIN | precision\_MAX |
| .5-1 y | 2.109234 | 1.250000 | 4.890000 |
| 1-1.5 y | 1.658714 | 1.292857 | 3.025000 |
| 1.5-2 y | 1.644224 | 1.207143 | 2.375000 |
| 2-4 y | 1.715520 | 1.428571 | 1.957143 |
| 4-6 y | 1.680247 | 1.178571 | 2.635714 |
| 8-10 y | 1.586332 | 1.300000 | 2.042857 |
| adult | 1.498571 | 1.150000 | 1.985714 |

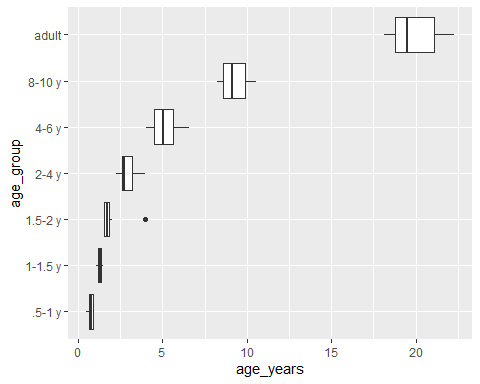
ds %>% ggplot(aes(x = age\_group, y = precision)) + geom\_boxplot() + geom\_hline(yintercept = 2.5)



ds %>% ggplot(aes(x = age, y = precision)) + geom\_point() + geom\_hline(yintercept = 2.5)

 ### 3B Age Convert age to years so that it can be more easily compared to age\_group. Visualize age in years by age\_group to see whether participants are the correct age for their group

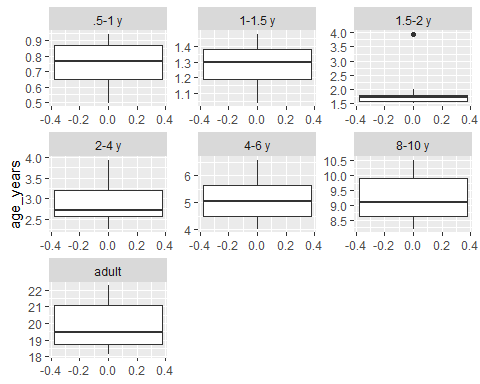
ds <- ds %>% mutate(age\_years = age/365.25)  
ds %>% group\_by(id, age\_group) %>%   
 summarize(age\_years = mean(age\_years)) %>%   
 ggplot(aes(y = age\_group, x = age\_years)) + geom\_boxplot()



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

Another option would be to facet by age group and to let the scales be “free” to get a better look

ds %>% group\_by(id, age\_group) %>%   
 summarize(age\_years = mean(age\_years)) %>%   
 ggplot(aes(y = age\_years)) +   
 geom\_boxplot() +   
 facet\_wrap("age\_group", scales = "free")



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

Make a summary table of age in years by age group to check whether all participants’ ages are correct

ds %>% group\_by(age\_group) %>% summarize(min\_age = min(age\_years), max\_age = max(age\_years))

## # A tibble: 7 x 3  
## age\_group min\_age max\_age  
## <chr> <dbl> <dbl>  
## 1 .5-1 y 0.493 0.942  
## 2 1-1.5 y 1.04 1.48   
## 3 1.5-2 y 1.51 3.94   
## 4 2-4 y 2.26 3.94   
## 5 4-6 y 4.02 6.57   
## 6 8-10 y 8.21 10.5   
## 7 adult 18.1 22.3