Video evaluation via SoSci

Irene Sophia Plank

2022-10-05

Setup

The data is loaded and relevant columns are selected. Then, it is transformed to the long format and the Video names are added. In the end, all information is contained in one data frame that is in the long format.

Choosing videos for the FER task

For the FER task, we want to choose the most natural video groups. Two things are important: that the videos are consistently rated as natural (median above 50) and that this is not due to a few very high ratings. First, outliers are removed that are lower than or equal to the 25% quantile minus 1.5 times the interquartile range. Then, face IDs are chosen who have the highest minima and a median per emotion of at least 50. We chose the median instead of the mean because the distributions are not at all normally distributed. The comments concerning the selected videos are saved to further improve them.

```
# get rid of extreme outliers for each video
df_fil = df %>% group_by(VID, FID, GEN) %>%
  filter(NAT > quantile(NAT, na.rm = T)[2] - 1.5*IQR(NAT, na.rm = T))

# aggregate per face ID and emotion
df_agg = df_fil %>% group_by(VID, GEN, EMO) %>%
  summarise(
    NAT_med = median(NAT, na.rm = T),
    NAT_min = min(NAT, na.rm = T),
    NAT_val = n()
  ) %>% filter(NAT_med >= 50 & NAT_val > 10)
```

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

```
# choose the videos with the highest minima
n = 8 # number of videos per gender and emotion
vid_sel = df_agg %>%
  arrange(desc(NAT_min)) %>%
  group_by(GEN, EMO) %>%
  slice(1:n) %>%
  select(GEN, EMO, VID)

# filter the full data frame to only include selected videos
df_FER_sel = df_fil %>% filter(VID %in% vid_sel$VID)

# aggregate info for the selected videos
df_FER_sel_agg = df_FER_sel %>% group_by(FID, VID, EMO, GEN) %>%
  summarise(
    NAT_avg = mean(NAT, na.rm = T),
    NAT_ste = sd(NAT, na.rm = T) / sqrt(n()),
    NAT_med = median(NAT, na.rm = T),
    NAT_min = min(NAT, na.rm = T)
  ) %>%
  arrange(FID, EMO, VID) %>%
  mutate(
    EMO     = as.factor(EMO),
    GEN     = as.factor(GEN)
  )
```

```
## `summarise()` has grouped output by 'FID', 'VID', 'EMO'. You can override using
## the `.groups` argument.
```

```
kable(df_FER_sel_agg)
```

FID	VID	EMO	GEN	NAT_avg	NAT_ste	NAT_med	NAT_min
F01	AF01AFS_gimped	AF	F	65.48000	4.679430	65.0	30
F01	AF01ANS_gimped	AN	F	80.08696	4.076789	85.0	37
F01	AF01HAS_gimped	HA	F	70.14286	4.723440	74.0	27
F01	AF01SAS_gimped	SA	F	65.40000	5.216640	66.0	21
F02	AF02AFS_gimped	AF	F	78.96000	3.592344	81.0	41
F02	BF02HAS_gimped	HA	F	82.81250	4.041677	84.5	46
F02	AF02SAS_gimped	SA	F	60.78947	5.372977	65.0	27
F05	AF05AFS_gimped	AF	F	74.84000	4.412361	81.0	25
F05	AF05ANS_gimped	AN	F	65.60000	4.720876	63.0	21
F05	AF05HAS_gimped	HA	F	72.09091	3.921254	74.5	21
F05	AF05SAS_gimped	SA	F	69.69565	4.852950	72.0	21
F07	BF07ANS_gimped	AN	F	65.56000	4.519248	68.0	21
F07	BF07SAS_gimped	SA	F	64.12000	4.836776	61.0	25
F08	BF08HAS_gimped	HA	F	73.52381	3.566791	76.0	31
F09	AF09AFS_gimped	AF	F	74.92308	7.346254	83.0	23
F09	AF09HAS_gimped	HA	F	70.04000	5.441960	81.0	10
F09	AF09SAS_gimped	SA	F	79.50000	3.979112	86.0	38
F11	AF11SAS_gimped	SA	F	65.32000	5.179356	68.0	25
F14	AF14ANS_gimped	AN	F	63.52000	5.400654	63.0	26
F14	AF14SAS_gimped	SA	F	72.68000	4.248027	74.0	37
F16	AF16AFS_gimped	AF	F	63.76000	5.407489	75.0	26
F16	AF16HAS_gimped	HA	F	58.87500	5.785199	63.5	11
F19	AF19HAS_gimped	HA	F	63.40000	6.081940	73.0	10
F20	AF20AFS_gimped	AF	F	70.50000	5.208341	77.5	22
F20	AF20ANS_gimped	AN	F	71.36000	4.116989	69.0	30
F20	AF20SAS_gimped	SA	F	66.12000	4.294306	70.0	25
F24	AF24AFS_gimped	AF	F	61.04167	5.294747	60.0	25
F24	AF24ANS_gimped	AN	F	73.52000	4.004214	78.0	31
F29	AF29AFS_gimped	AF	F	68.04000	4.747448	69.0	28
F29	AF29ANS_gimped	AN	F	64.76000	5.172775	66.0	18
F29	AF29HAS_gimped	HA	F	66.24000	4.966313	75.0	17
F32	AF32ANS_gimped	AN	F	69.92000	4.535754	77.0	25
M01	AM01AFS_gimped	AF	M	73.40000	4.277481	72.5	34
M01	AM01ANS_gimped	AN	M	79.05882	3.203250	76.0	56
M01	AM01SAS_gimped	SA	M	67.27778	5.642281	71.5	21
M02	AM02ANS_gimped	AN	M	64.80952	5.251584	63.0	22
M02	AM02HAS_gimped	HA	M	76.05263	3.561879	75.0	55
M03	AM03AFS_gimped	AF	M	71.33333	4.703764	71.0	29
M03	AM03SAS_gimped	SA	M	69.75000	4.561207	68.0	30

FID	VID	EMO	GEN	NAT_avg	NAT_ste	NAT_med	NAT_min
M05	AM05AFS_gimped	AF	M	70.61905	4.419910	75.0	34
M05	AM05ANS_gimped	AN	M	59.80952	5.367281	56.0	16
M06	AM06SAS_gimped	SA	M	65.95238	5.105308	73.0	23
M07	AM07AFS_gimped	AF	M	66.38095	5.654108	69.0	21
M07	AM07SAS_gimped	SA	M	63.52381	4.332391	65.0	27
M09	AM09HAS_gimped	HA	M	85.35294	2.400890	84.0	64
M09	AM09SAS_gimped	SA	M	77.70000	3.935065	80.5	40
M10	AM10AFS_gimped	AF	M	74.76190	5.085415	86.0	36
M10	AM10HAS_gimped	HA	M	82.72222	3.545968	83.5	46
M11	AM11AFS_gimped	AF	M	74.47619	4.552142	78.0	33
M11	AM11ANS_gimped	AN	M	79.00000	3.277515	80.5	46
M11	AM11HAS_gimped	HA	M	79.94444	3.257442	79.0	60
M13	AM13AFS_gimped	AF	M	73.60000	4.660811	81.0	30
M13	AM13HAS_gimped	HA	M	79.42857	4.060009	82.0	31
M13	AM13SAS_gimped	SA	M	67.14286	5.033358	78.0	23
M14	AM14ANS_gimped	AN	M	76.85714	4.398747	79.0	35
M14	AM14HAS_gimped	HA	M	76.04762	4.027569	76.0	42
M14	AM14SAS_gimped	SA	M	72.57143	5.222466	74.0	21
M23	BM23ANS_gimped	AN	M	55.85714	6.508707	63.0	12
M23	BM23SAS_gimped	SA	M	57.28571	5.424435	63.0	19
M27	BM27HAS_gimped	HA	M	66.33333	4.939796	65.0	22
M28	BM28ANS_gimped	AN	M	69.25000	4.566396	73.5	25
M28	BM28HAS_gimped	HA	M	78.33333	3.699496	79.0	41
M31	BM31AFS_gimped	AF	M	73.60000	4.718608	73.5	32
M31	BM31ANS_gimped	AN	M	71.00000	5.366127	78.0	23

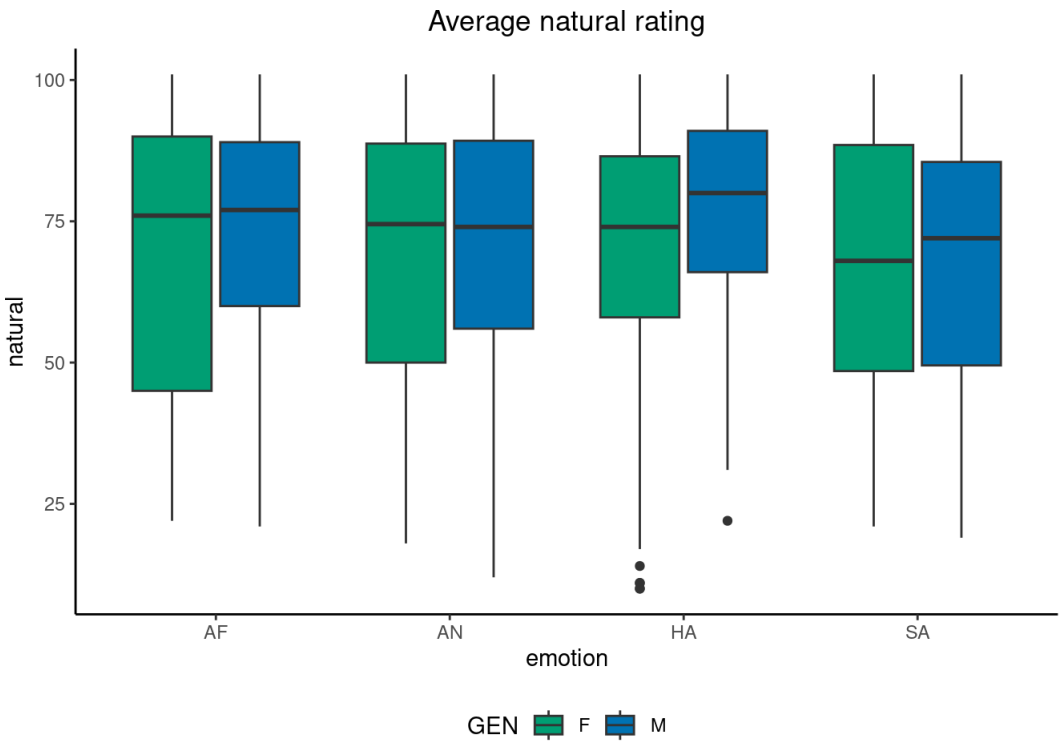
```
# # get all comments
# df_FER_sel_COM = df_FER_sel %>% filter(!is.na(COM)) %>% select(GEN, FID, VID, COM) %>%
#   arrange(FID, VID)
# kable(df_FER_sel_COM)

# drop all unnecessary variables
rm(list = setdiff(ls(), c("df", "df_FER_sel", "df_FER_sel_agg", "df_FER_sel_COM")))
```

Visualisation

```
# set colour scheme
cls = c("#0072B2", "#009E73")
names(cls) = c("M", "F")

# create boxplots
ggplot(df_FER_sel, aes(x = EMO, y = NAT, fill = GEN)) +
  geom_boxplot() +
  labs (x = "emotion", y = "natural") +
  ggtitle("Average natural rating") +
  theme_classic() + scale_fill_manual(values = cls) +
  theme(legend.position = "bottom", plot.title = element_text(hjust = 0.5))
```



Gender and emotion effects

To stay in line with our main analysis, we compute a Bayesian ANOVA to assess possible differences in naturalness between emotions and gender expression.

```
# check normal distribution
kable(df_FER_sel_agg %>%
        group_by(EMO, GEN) %>%
        rstatix::shapiro_test(NAT_avg) %>%
        mutate(
sig = if_else(p < 0.05, "*", "")
        ))
```

EMO	GEN	variable	statistic	p	sig
AF	F	NAT_avg	0.9628623	0.8368830	
AF	M	NAT_avg	0.8284125	0.0571510	
AN	F	NAT_avg	0.8958537	0.2649964	
AN	M	NAT_avg	0.9221722	0.4477364	
HA	F	NAT_avg	0.9692711	0.8922278	
HA	M	NAT_avg	0.9127296	0.3736861	
SA	F	NAT_avg	0.9112234	0.3627591	
SA	M	NAT_avg	0.9806486	0.9660427	

```
# now we can compute our ANOVA
BayesFactor::anovaBF(NAT_avg ~ GEN * EMO, data = df_FER_sel_agg)
```

```
## Warning: data coerced from tibble to data frame
```

```
## Bayes factor analysis
## --------------
## [1] EMO                 : 1.079402  ±0%
## [2] GEN                 : 0.7966014 ±0.01%
## [3] EMO + GEN           : 0.9400547 ±1.06%
## [4] EMO + GEN + EMO:GEN : 0.5766506 ±1.16%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

None of the models credibly outperform the intercept-only model for this data, with anecdotal evidence against all models except the one including emotion as a predictor for which there was anecdotal evidence in favour. Therefore, the stimuli are well matched in terms of naturalness.

Choosing videos for the PAL task

Since in the PAL task, we do not use the videos themselves but three stills of different emotion strengths, we can simply choose the videos with the best median natural ratings. However, to keep it consistent with the Lawson et al. (2017) paper, we will use six identities which are shown in each emotion.

```
# get rid of extreme outliers for each video
df_fil = df %>% group_by(VID, FID, GEN) %>%
  filter(NAT > quantile(NAT, na.rm = T)[2] - 1.5*IQR(NAT, na.rm = T))

# aggregate per face ID and emotion
df_agg = df_fil %>% group_by(VID, FID, GEN, EMO) %>%
  summarise(
    NAT_med = median(NAT, na.rm = T)
  ) %>% filter(VID %in% df_FER_sel_agg$VID & (EMO == "AF" | EMO == "HA"))
```

```
## `summarise()` has grouped output by 'VID', 'FID', 'GEN'. You can override using
## the `.groups` argument.
```

```
# find out of which IDs both emotions are left
fids = df_agg$FID[duplicated(df_agg$FID)]
df_agg = df_agg %>% filter(FID %in% fids)

# choose the videos with the highest median
n = 3 # number of videos per gender and emotion
fid_sel = df_agg %>% group_by(GEN, FID) %>%
  summarise(NAT_med = mean(NAT_med)) %>%
  arrange(desc(NAT_med)) %>%
  group_by(GEN) %>%
  slice(1:n) %>%
  select(GEN, FID)
```

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

```
# filter the full data frame to only include selected videos
df_PAL_sel = df_fil %>% filter(FID %in% fid_sel$FID & (EMO == "AF" | EMO == "HA"))

# aggregate info for the selected videos
df_PAL_sel_agg = df_PAL_sel %>% group_by(FID, VID, EMO, GEN) %>%
  summarise(
    NAT_avg = mean(NAT, na.rm = T),
    NAT_ste = sd(NAT, na.rm = T) / sqrt(n()),
    NAT_med = median(NAT, na.rm = T),
    NAT_min = min(NAT, na.rm = T)
  ) %>%
  arrange(FID, EMO, VID) %>%
  mutate(
    EMO = as.factor(EMO),
    GEN = as.factor(GEN)
  )
```

```
## `summarise()` has grouped output by 'FID', 'VID', 'EMO'. You can override using
## the `.groups` argument.
```

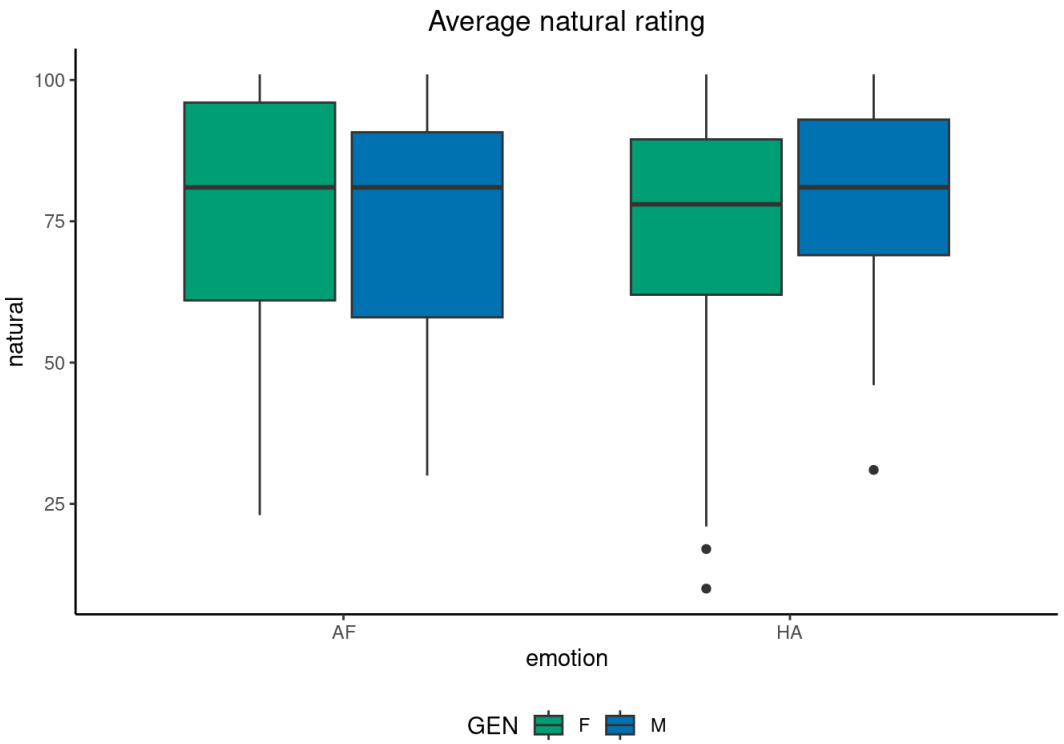
```
kable(df_PAL_sel_agg)
```

FID	VID	EMO	GEN	NAT_avg	NAT_ste	NAT_med	NAT_min
F02	AF02AFS_gimped	AF	F	78.96000	3.592344	81.0	41
F02	BF02HAS_gimped	HA	F	82.81250	4.041677	84.5	46
F05	AF05AFS_gimped	AF	F	74.84000	4.412361	81.0	25
F05	AF05HAS_gimped	HA	F	72.09091	3.921254	74.5	21
F09	AF09AFS_gimped	AF	F	74.92308	7.346254	83.0	23
F09	AF09HAS_gimped	HA	F	70.04000	5.441960	81.0	10
M10	AM10AFS_gimped	AF	M	74.76190	5.085415	86.0	36
M10	AM10HAS_gimped	HA	M	82.72222	3.545968	83.5	46
M11	AM11AFS_gimped	AF	M	74.47619	4.552142	78.0	33
M11	AM11HAS_gimped	HA	M	79.94444	3.257442	79.0	60
M13	AM13AFS_gimped	AF	M	73.60000	4.660811	81.0	30
M13	AM13HAS_gimped	HA	M	79.42857	4.060009	82.0	31

```
# # get all comments
# df_PAL_sel_COM = df_PAL_sel %>% filter(!is.na(COM)) %>% select(GEN, FID, VID, COM) %>%
#   arrange(FID, VID)
# kable(df_PAL_sel_COM)
```

Visualisation

```
# create boxplots
ggplot(df_PAL_sel, aes(x = EMO, y = NAT, fill = GEN)) +
  geom_boxplot() +
  labs (x = "emotion", y = "natural") +
  ggtitle("Average natural rating") +
  theme_classic() + scale_fill_manual(values = cls) +
  theme(legend.position = "bottom", plot.title = element_text(hjust = 0.5))
```



Gender and emotion effects

To stay in line with our main analysis, we compute a Bayesian ANOVA to assess possible differences in naturalness between emotions and gender expression.

```
# check normal distribution
kable(df_PAL_sel_agg %>%
        group_by(EMO, GEN) %>%
        rstatix::shapiro_test(NAT_avg) %>%
        mutate(
sig = if_else(p < 0.05, "*", "")
        ))
```

EMO	GEN	variable	statistic	p	sig
AF	F	NAT_avg	0.7651170	0.0336877	*
AF	M	NAT_avg	0.9207357	0.4549305	
HA	F	NAT_avg	0.8668404	0.2865976	
HA	M	NAT_avg	0.8641495	0.2790664	

```
# not normally distributed in female AF group, therefore, rank transform
df_PAL_sel_agg = df_PAL_sel_agg %>%
  ungroup() %>%
  mutate(
    rNAT_avg   = rank(NAT_avg)
  )

# now we can compute our ANOVA
BayesFactor::anovaBF(rNAT_avg ~ GEN * EMO, data = df_PAL_sel_agg)
```

```
## Warning: data coerced from tibble to data frame
```

```
## Bayes factor analysis
## --------------
## [1] EMO                 : 0.6226314 ±0%
## [2] GEN                 : 0.5005955 ±0%
## [3] EMO + GEN           : 0.3148277 ±0.74%
## [4] EMO + GEN + EMO:GEN : 0.6318817 ±0.82%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

None of the models credibly outperform the intercept-only model for this data, with moderate and anecdotal evidence against the models including any predictors. Therefore, the stimuli are well matched in terms of naturalness.