Paper 4: Visual analytics and well-being

The objective of the paper is to link consumer well-being (measured by PERMA scale) to instagram activities. Can we predict well-being from consumer generated visual data?

Below you'll find an overview of the data we gave available to do that.

Dependent Variable: PERMA

P__EMO and N__EMO = Positive and Negative emotions

Emotions are an important part of our well-being. Emotions can range from very negative to very positive, and range from high arousal (e.g., excitement, explosive) to low arousal (e.g., calm, relaxed, sad). For Positive emotion, the PERMA-Profiler measures general tendencies toward feeling contentment and joy. For Negative emotion, the Profiler measures tendencies toward feeling, sad, anxious, and angry.

E = Engagement

Engagement refers to being absorbed, interested, and involved in an activity or the world itself. Very high levels of engagement are known as a state called "flow", in which you are so completely absorbed in an activity that you lose all sense of time.

R = Relationships

Relationships refer to feeling loved, supported, and valued by others. Having positive relationships with others is an important part of life feeling good and going well. Other people matter!

M = Meaning

Meaning refers to having a sense of purpose in life, a direction where life is going, feeling that life is valuable and worth living, or connecting to something greater than ourselves, such as religious faith, a charity or a personally meaningful goal. Meaning provides a sense that life matters.

A = Accomplishment

Accomplishment can be objective, marked by honors and awards received, but feelings of mastery and achievement is also important. The Profiler measures subjective feelings of accomplishment and staying on top of daily responsibilities. It involves working toward and reaching goals, and feeling able to complete tasks and daily responsibilities.

H = Health

Although not part of the PERMA model itself, physical health and vitality is another important part of well-being. The Profiler measures a subjective sense of health – feeling good and healthy each day.

Aggreate to scores

```
library(tidyverse)


q_csv_results <- read.csv('well_being_data.csv') %>%
                 rename( id = X.,
                         gender = `What.is.your.gender.`,
                         born= `In.what.year.were.you.born.`,
                         education = `What.is.the.highest.level.of.education.you.have.completed.`,
                         employed = `Are.you.currently...`,
                         income = `What.is.your.total.household.income.`,
                         A_2 = `How.often.do.you.achieve.the.important.goals.you.have.set.for.yourself.
                         N_1 = `How.often.do.you.feel.anxious.`,
                         P_1 = `How.often.do.you.feel.joyful.`,
                         E_1 = `How.often.do.you.become.absorbed.in.what.you.are.doing.`,
                         A_1 = `How.much.of.the.time.do.you.feel.you.are.making.progress.towards.accompl
                         H_1 = `How.would.you.say.your.health.is.`,
                         M_1 = `To.what.extent.do.you.lead.a.purposeful.and.meaningful.life.`,
                         R_1 = `To.what.extent.do.you.receive.help.and.support.from.others.when.you.nee
                         M_2 = `In.general..to.what.extent.do.you.feel.that.what.you.do.in.your.life.is
                         E_2 = `To.what.extent.do.you.feel.excited.and.interested.in.things.`,
                         LON = `How.lonely.do.you.feel.in.your.daily.life.`,
                         H_2 = `How.satisfied.are.you.with.your.current.physical.health.`,
                         P_2 = `How.often.do.you.feel.positive.`,
                         N_2 = `How.often.do.you.feel.angry.`,
                         A_3 = `How.often.are.you.able.to.handle.your.responsibilities.`,
                         N_3 = `How.often.do.you.feel.sad.`,
                         E_3 = `How.often.do.you.lose.track.of.time.while.doing.something.you.enjoy.`,
                         H_3 = `Compared.to.others.of.your.same.age.and.sex..how.is.your.health.`,
                         R_2 = `To.what.extent.do.you.feel.loved.`,
                         M_3 = `To.what.extent.do.you.generally.feel.you.have.a.sense.of.direction.in.y
                         R_3 = `How.satisfied.are.you.with.your.personal.relationships.`,
                         P_3 = `To.what.extent.do.you.feel.contented.`,
                         HAP = `Taking.all.things.together..how.happy.would.you.say.you.are.`,
                         participate = `Would.you.be.willing.to.participate.in.a.follow.up.research.`,
                         insta_user_id = code_complete,
                         start_q = `Start.Date..UTC.`,
                         end_q = `Submit.Date..UTC.`,
                         network_id = `Network.ID`
                         )
```

Aggregate scores

The individual scores are summarized in 5 dimensions, which in turn is summarized in a single PERMA score.

```
  aggregate_scores <- q_csv_results %>%
                      dplyr::rowwise() %>%
                      dplyr::mutate( P = mean(P_1,P_2,P_3) ,
                                     E = mean(E_1,E_2,E_3),
                                     R = mean(R_1,R_2,R_3),
                                     M = mean(M_1,M_2,M_3),
                                     A = mean(A_1,A_2,A_3),
                                     PERMA= mean(P_1,P_2,P_3,E_1,E_2,E_3, R_1,R_2,R_3, M_1,M_2,M_3,A_1,
                                     N_EMO = mean(N_1,N_2,N_3),
                                     P_EMO = mean(P_1,P_2,P_3))
```

Columns:

id, gender, born, education, employed, income, A__2, N__1, P__1, E__1, A__1, H__1, M__1, R__1, M__2, E__2, LON, H__2, P__2, N__2, A__3, N__3, E__3, H__3, R__2, M__3, R__3, P__3, HAP, participate, insta__user__id, completed, start__q, end__q, network__id, P, E, R, M, A, PERMA, N__EMO, P__EMO

Independent variable

I created six files each containing a subset of the variables we can use. The data needs to be aggregated so that we have a single variable for each respondent in the sample.

Image data

We have the 'usual' Instagram data fields.

Columns:

image__id, image__link, image__url, image__height, image__width, image__filter, image__posted__time__unix, image__posted__time, data__memorability, user__id, user__full__name, user__name, user__website, user__profile__pic, user__bio, user__followed__by, user__follows, user__posted__photos

Sentiment, Anp, Emotion

Each image has 5 ANP's. Each ANP has a sentiment score and is scored along the 24 emotions of Putchnik. I selected the top emotions (most associated with an ANP) , to be represented. Using all emotions would result in 4 million rows. I notice that most ANP's per image tend to have similar emotions associated with the image, so we can have a single emotion per image.

```
anp_data <- feather::read_feather('export_data/anp.feather')

head(anp_data)
```

```
## # A tibble: 6 x 5
##                         image_id     anp_label anp_sentiment emotion_score
##                            <chr>         <chr>         <dbl>         <dbl>
## 1 951727030670259635_143763900      hot_boys         0.017        0.1760
## 2 951727030670259635_143763900  young_couple         0.019        0.2113
## 3 951727030670259635_143763900 dirty_laundry        -0.263        0.0929
## 4 951727030670259635_143763900   global_mall        -0.031        0.1304
## 5 951728575726873168_289794729    high_boots         0.025        0.1394
## 6 951728575726873168_289794729    funny_pets         0.078        0.1924
## # ... with 1 more variables: emotion_label <chr>
```

Columns:

image__id, anp__label, anp__sentiment, emotion__score, emotion__label

Object Labels

Each image has an array of objects (max 10). Each label has a minimal confidence measure of > 0.7.

```
object_label <- feather::read_feather('export_data/object_labels.feather')

head(object_label)
```

```
## # A tibble: 6 x 3
##                         image_id data_amz_label data_amz_label_confidence
##                            <chr>          <chr>                     <dbl>
## 1  863479386465416946_545497348         Animal                  90.16310
## 2  916939688871507178_545497348         Animal                  83.51867
## 3  551681403589539797_545497348         Animal                  74.83721
## 4 1189285646274180856_545497348         Animal                  76.92097
## 5  962361211517974133_545497348         Animal                  71.22387
## 6  986200749440808002_545497348         Animal                  80.85960
```

Columns:

image_id, data_amz_label, data_amz_label_confidence

Face

Each image face detection was applied. we have some nice new params included in this set smile, sunglass, beard etc. Each variable has a TRUE / FALSE indicator and a confidence score. Also each detected face has an emotion and emotion score attached.

```
face <- feather::read_feather('export_data/face.feather')

head(face)
```

```
## # A tibble: 6 x 17
##                         image_id face_id face_gender face_gender_confidence
##                            <chr>   <dbl>       <chr>                  <dbl>
## 1 1003944279371027183_703978203       6      Female               98.74142
## 2 1003944279371027183_703978203       6      Female               98.74142
## 3 1003944279371027183_703978203       6      Female               98.74142
## 4 1003944279371027183_703978203      68        Male               99.92752
## 5 1003944279371027183_703978203      68        Male               99.92752
## 6 1003944279371027183_703978203      68        Male               99.92752
## # ... with 13 more variables: face_age_range_high <dbl>,
## #   face_age_range_low <dbl>, face_sunglasses <lgl>, face_beard <lgl>,
## #   face_beard_confidence <dbl>, face_mustache <lgl>,
## #   face_mustache_confidence <dbl>, face_smile <lgl>,
## #   face_smile_confidence <dbl>, eyeglasses <lgl>,
## #   eyeglasses_confidence <dbl>, face_emo <chr>, emo_confidence <dbl>
```

Columns:

image_id, face_id, face_gender, face_gender_confidence, face_age_range_high, face_age_range_low, face_sunglasses, face_beard, face_beard_confidence, face_mustache, face_mustache_confidence, face_smile, face_smile_confidence, eyeglasses, eyeglasses_confidence, face_emo, emo_confidence

Celebrity

Each image celebrity detection was applied. It's a novel feature so I am not sure if we want to use it. But I could see potential for this in advertising and endorsements etc.

```r
celebrity <- feather::read_feather('export_data/celebrity.feather')

head(celebrity)
```

```
## # A tibble: 6 x 4
##                         image_id face_celebrity_name face_celebrity_id
##                            <chr>               <chr>             <chr>
## 1 1305879424679287435_3041716852        Memos Begnis          2fe8aC5a
## 2  1474727143443660143_263042348     Victoria Dyring            11G9b2
## 3   1376342482815974929_53918317         Susan Boyle          3cI5xV1n
## 4   1373407612938533591_53918317       Peter Capaldi          41K0na40
## 5  1021320177296478352_703978203               Drake           4g7LX8j
## 6  1041654001515500189_703978203               Drake           4g7LX8j
## # ... with 1 more variables: face_celebrity_match_confidence <dbl>
```

Columns:

image_id, face_celebrity_name, face_celebrity_id, face_celebrity_match_confidence