Emotion dynamics is the study of how emotions change over time. Sometimes our feelings are quite stable, but other times capricious. Measuring and predicting these patterns for different people is somewhat of a Holy Grail for emotion researchers. In particular, some researchers are aspiring to discover mathematical laws that capture the complexity of our inner emotional experiences – much like physicists divining the laws that govern objects in the natural environment. These discoveries would revolutionize our understanding of our everyday feelings and when our emotions can go awry.

**Affective State Space**

We start with the affective state space – the theoretical landscape on which our conscious feelings roam free. This space is represented as *two-dimensional*, although we acknowledge that this fails to capture all aspects of conscious feeling. The first dimension, represented along the x-axis, is *valence* and this refers to how unpleasant vs. pleasant we feel. The second dimension, represented along the y-axis, is *arousal*. Somewhat less intuitive, arousal refers to how deactivated/sluggish/sleepy vs. activated/energized/awake we feel. At any time, our emotional state can be defined in terms of valence and arousal. So if you’re feeling stressed you would be low in valence and high in arousal. Let’s say you’re serene and calm, then you would be high in valence and low in arousal. Most of the time, we feel moderately high valence and moderate arousal (i.e., content), but if you’re the type of person who is chronically stressed, this would be different.

This is all well and good when we think about how we’re feeling right now, but it’s also worth considering how our emotions are changing. On a regular day, our emotions undergo minor fluctuations – sometimes in response to minor hassles or victories, and sometimes for no discernible reason. In this small paragraph, I’ve laid out a number of parameters, all of which vary between different people:

* **Attractor**: Our typical emotional state. At any given moment, our feelings are pulled toward this state. Some people tend to be happier, whereas others are less happy.
* **Stability**: How emotionally stable one is. Some people are more emotionally stable than others. Even in the face of adversity, an emotionally stable person keeps their cool.
* **Dispersion**: The range of our emotional landscape. Some people experience intense highs and lows, whereas others persist somewhere in the middle.

We’ll keep all of this in mind for the simulation. We’ll start with a fairly simple simulation with 100 hypothetical people. We’ll need the following packages.

library(psych)

library(tidyverse)

library(sn)

And then we’ll create a function that performs the simulation. Note that each person *i* has their own attractor, recovery rate, stability, and dispersion. For now we’ll just model random fluctuations in emotions, a sort of Brownian motion. You can imagine our little **simulatons** (fun name for the hypothetical people in the simulation) sitting around on an average day doing nothing in particular.

simulate\_affect <- function(n = 2, time = 250, negative\_event\_time = NULL) {

dt <- data.frame(matrix(nrow = time, ncol = 1))

colnames(dt) <- "time"

dt$time <- 1:time

valence <- data.frame(matrix(nrow = time, ncol = 0))

arousal <- data.frame(matrix(nrow = time, ncol = 0))

for(i in 1:n) {

attractor\_v <- rnorm(1, mean = 3.35, sd = .75)

instability\_v <- sample(3:12, 1, replace = TRUE, prob = c(.18, .22, .18, .15, .8, .6, .5, .4, .2, .1))

dispersion\_v <- abs(rsn(1, xi = .15, omega = .02, alpha = -6) \* instability\_v) #rsn simulates a skewed distribution.

if(!is.null(negative\_event\_time)) {

recovery\_rate <- sample(1:50, 1, replace = TRUE) + negative\_event\_time

negative\_event <- (dt$time %in% negative\_event\_time:recovery\_rate) \* seq.int(50, 1, -1)

}

else {

negative\_event <- 0

}

valence[[i]] <- ksmooth(x = dt$time,

y = (negative\_event \* -.10) + arima.sim(list(order = c(1, 0, 0),

ar = .50),

n = time),

bandwidth = time/instability\_v, kernel = "normal")$y \* dispersion\_v + attractor\_v

#instability is modelled in the bandwidth term of ksmooth, such that higher instability results in higher bandwidth (greater fluctuation).

#dispersion scales the white noise (arima) parameter, such that there are higher peaks and troughs at higher dispersion.

attractor\_a <- rnorm(1, mean = .50, sd = .75) + sqrt(instability\_v) #arousal attractor is dependent on instability. This is because high instability is associated with higher arousal states.

instability\_a <- instability\_v + sample(-1:1, 1, replace = TRUE)

dispersion\_a <- abs(rsn(1, xi = .15, omega = .02, alpha = -6) \* instability\_a)

arousal[[i]] <- ksmooth(x = dt$time,

y = (negative\_event \* .075) + arima.sim(list(order = c(1, 0, 0),

ar = .50),

n = time),

bandwidth = time/instability\_a, kernel = "normal")$y \* dispersion\_a + attractor\_a

}

valence[valence > 6] <- 6

valence[valence < 0] <- 0

arousal[arousal > 6] <- 6

arousal[arousal < 0] <- 0

colnames(valence) <- paste0("valence\_", 1:n)

colnames(arousal) <- paste0("arousal\_", 1:n)

dt <- cbind(dt, valence, arousal)

return(dt)

}

set.seed(190625)

emotions <- simulate\_affect(n = 100, time = 300)

emotions %>%

select(valence\_1, arousal\_1) %>%

head()

## valence\_1 arousal\_1

## 1 1.328024 5.380643

## 2 1.365657 5.385633

## 3 1.401849 5.390470

## 4 1.436284 5.395051

## 5 1.468765 5.399162

## 6 1.499062 5.402752

So we see the first six rows for participant 1’s valence and arousal. But if we want to plot these across multiple simulatons, we need to wrangle the data into long form. We’ll also compute some measures of within-person deviation. The Root Mean Square Successive Difference (RMSSD) takes into account gradual shifts in the mean. Those who are more emotionally unstable will have a higher RMSSD. For two dimensions (valence and arousal) we’ll just compute the mean RMSSD.

emotions\_long <- emotions %>%

gather(key, value, -time) %>%

separate(key, into = c("dimension", "person"), sep = "\_") %>%

spread(dimension, value) %>%

group\_by(person) %>%

mutate(rmssd\_v = rmssd(valence),

rmssd\_a = rmssd(arousal),

rmssd\_total = mean(rmssd\_v + rmssd\_a)) %>%

ungroup()

Let’s see what this looks like for valence and arousal individually.

emotions\_long %>%

ggplot(aes(x = time, y = valence, group = person, color = rmssd\_v)) +

geom\_line(size = .75, alpha = .75) +

scale\_color\_gradient2(low = "black", mid = "grey", high = "red", midpoint = median(emotions\_long$rmssd\_v)) +

labs(x = "Time",

y = "Valence",

color = "Instability",

title = "Simulated Valence Scores over Time for 100 People") +

theme\_minimal(base\_size = 16)

emotions\_long %>%

ggplot(aes(x = time, y = arousal, group = person, color = rmssd\_a)) +

geom\_line(size = .75, alpha = .75) +

scale\_color\_gradient2(low = "black", mid = "grey", high = "red", midpoint = median(emotions\_long$rmssd\_a)) +

labs(x = "Time",

y = "Arousal",

color = "Instability",

title = "Simulated Arousal Scores over Time for 100 People") +

theme\_minimal(base\_size = 16)

We see that some lines are fairly flat and others fluctuate more widely. More importantly, most people are somewhere in the middle.

We can get a sense of one simulated person’s affective state space as well. The goal here is to mimic the kinds of models shown in below:

**Abstract**

People display a remarkable variability in the patterns and trajectories with which their feelings change over time. In this article, we present a theoretical account for the dynamics of affect (DynAffect) that identifies the major processes underlying individual differences in the temporal dynamics of affective experiences. It is hypothesized that individuals are characterized by an affective home base, a baseline attractor state around which affect fluctuates. These fluctuations vary as the result of internal or external processes to which an individual is more or less sensitive and are regulated and tied back to the home base by the attractor strength. Individual differences in these 3 processes--affective home base, variability, and attractor strength--are proposed to underlie individual differences in affect dynamics. The DynAffect account is empirically evaluated by means of a diffusion modeling approach in 2 extensive experience-sampling studies on people's core affective experiences. The findings show that the model is capable of adequately capturing the observed dynamics in core affect across both large (Study 1) and shorter time scales (Study 2) and illuminate how the key processes are related to personality and emotion dispositions. Implications for the understanding of affect dynamics and affective dysfunctioning in psychopathology are also discussed.

emotions\_long %>%

filter(person %in% sample(1:100, 6, replace = FALSE)) %>%

ggplot(aes(x = valence, y = arousal, group = person)) +

geom\_path(size = .75) +

scale\_x\_continuous(limits = c(0, 6)) +

scale\_y\_continuous(limits = c(0, 6)) +

labs(x = "Valence",

y = "Arousal",

title = "Affective State Space for Six Randomly Simulated People") +

facet\_wrap(~person) +

theme\_minimal(base\_size = 18) +

theme(plot.title = element\_text(size = 18, hjust = .5))

**Animating the Affective State Space**

To really appreciate what’s going on, we need to animate this over time. I’ll add some labels to the affective state space so that it’s easier to interpret what one might be feeling at that time. I’ll also add color to show which individuals are more unstable according to RMSSD.

library(gganimate)

p <- emotions\_long %>%

ggplot(aes(x = valence, y = arousal, color = rmssd\_total)) +

annotate("text", x = c(1.5, 4.5, 1.5, 4.5), y = c(1.5, 1.5, 4.5, 4.5), label = c("Gloomy", "Calm", "Anxious", "Happy"),

size = 10, alpha = .50) +

annotate("rect", xmin = 0, xmax = 3, ymin = 0, ymax = 3, alpha = 0.25, color = "black", fill = "white") +

annotate("rect", xmin = 3, xmax = 6, ymin = 0, ymax = 3, alpha = 0.25, color = "black", fill = "white") +

annotate("rect", xmin = 0, xmax = 3, ymin = 3, ymax = 6, alpha = 0.25, color = "black", fill = "white") +

annotate("rect", xmin = 3, xmax = 6, ymin = 3, ymax = 6, alpha = 0.25, color = "black", fill = "white") +

geom\_point(size = 3.5) +

scale\_color\_gradient2(low = "black", mid = "grey", high = "red", midpoint = median(emotions\_long$rmssd\_total)) +

scale\_x\_continuous(limits = c(0, 6)) +

scale\_y\_continuous(limits = c(0, 6)) +

labs(x = "Valence",

y = "Arousal",

color = "Instability",

title = 'Time: {round(frame\_time)}') +

transition\_time(time) +

theme\_minimal(base\_size = 18)

ani\_p <- animate(p, nframes = 320, end\_pause = 20, fps = 16, width = 550, height = 500)

ani\_p

**There’s a Storm Coming…**

Our simulation does a pretty good job at emulating the natural ebb and flow of emotions, but we know that emotions can be far more volatile. Let’s subject our simulation to a negative event. Perhaps all 100 **simulatons** co-authored a paper that just got rejected. In the function *simulate\_affect*, there’s an optional argument *negative\_event\_time* that causes a negative event to occur at the specified time. For this, we need to consider one more emotion dynamics parameter:

* **Recovery rate**: How quickly one recovers from an emotional event. If something bad happens, how long does it take to return to the attractor. You can see how I’ve modelled this parameter in the function above.

So we’ll run the simulation with a negative event arising at *t* = 150. The negative event will cause a downward spike in valence and an upward spike in arousal.

emotions\_event <- simulate\_affect(n = 100, time = 300, negative\_event\_time = 150)

emotions\_event\_long <- emotions\_event %>%

gather(key, value, -time) %>%

separate(key, into = c("dimension", "person"), sep = "\_") %>%

spread(dimension, value) %>%

group\_by(person) %>%

mutate(rmssd\_v = rmssd(valence),

rmssd\_a = rmssd(arousal),

rmssd\_total = mean(rmssd\_v + rmssd\_a)) %>%

ungroup()

emotions\_event\_long %>%

ggplot(aes(x = time, y = valence, group = person, color = rmssd\_v)) +

geom\_line(size = .75, alpha = .75) +

scale\_color\_gradient2(low = "black", mid = "grey", high = "red", midpoint = median(emotions\_event\_long$rmssd\_v)) +

labs(x = "Time",

y = "Valence",

color = "Instability",

title = "Simulated Valence Scores over Time for 100 People") +

theme\_minimal(base\_size = 16)

emotions\_event\_long %>%

ggplot(aes(x = time, y = arousal, group = person, color = rmssd\_a)) +

geom\_line(size = .75, alpha = .75) +

scale\_color\_gradient2(low = "black", mid = "grey", high = "red", midpoint = median(emotions\_event\_long$rmssd\_a)) +

labs(x = "Time",

y = "Arousal",

color = "Instability",

title = "Simulated Arousal Scores over Time for 100 People") +

theme\_minimal(base\_size = 16)

It’s pretty clear that something bad happened. Of course, some of our **simulatons** are unflappable, but most experienced a drop in valence and spike in arousal that we might identify as anxiety. Again, let’s visualize this evolving over time. Pay close attention to when the timer hits 150.

p2 <- emotions\_event\_long %>%

ggplot(aes(x = valence, y = arousal, color = rmssd\_total)) +

annotate("text", x = c(1.5, 4.5, 1.5, 4.5), y = c(1.5, 1.5, 4.5, 4.5), label = c("Gloomy", "Calm", "Anxious", "Happy"),

size = 10, alpha = .50) +

annotate("rect", xmin = 0, xmax = 3, ymin = 0, ymax = 3, alpha = 0.25, color = "black", fill = "white") +

annotate("rect", xmin = 3, xmax = 6, ymin = 0, ymax = 3, alpha = 0.25, color = "black", fill = "white") +

annotate("rect", xmin = 0, xmax = 3, ymin = 3, ymax = 6, alpha = 0.25, color = "black", fill = "white") +

annotate("rect", xmin = 3, xmax = 6, ymin = 3, ymax = 6, alpha = 0.25, color = "black", fill = "white") +

geom\_point(size = 3.5) +

scale\_color\_gradient2(low = "black", mid = "grey", high = "red", midpoint = median(emotions\_event\_long$rmssd\_total)) +

scale\_x\_continuous(limits = c(0, 6)) +

scale\_y\_continuous(limits = c(0, 6)) +

labs(x = "Valence",

y = "Arousal",

color = "Instability",

title = 'Time: {round(frame\_time)}') +

transition\_time(time) +

theme\_minimal(base\_size = 18)

ani\_p2 <- animate(p2, nframes = 320, end\_pause = 20, fps = 16, width = 550, height = 500)

ani\_p2

The overall picture is that some are more emotionally resilient than others. As of now, all the **simulatons** return to their baseline attractor, but we would realistically expect some to stay stressed or gloomy following bad news. In the coming months I’ll be looking into how to incorporate emotion regulation into the simulation. For example, maybe some of the **simulatons** use better coping strategies than others? I’m also interested in incorporating *appraisal* mechanisms that allow for different reactions depending on the type of emotional stimulus.