If you’re anything like me, you probably set a lot of goals. Whether it’s to finish a paper by the end of the summer or to spend more time with friends and family, goals are what help motivate us to *do something*. Goals are also intimately tied to our feelings. You may have had the experience of falling behind in your goals, which made you upset, but ultimately motivated you to step up your efforts. Conversely, you could be comfortably ahead of your goals and be tempted to coast for a while (much like the Hare in the Tortoise and the Hare who takes a nap when he is far ahead of the Tortoise).

Charles Carver presented an elegant model of emotion and behavior as intertwined in nested *feedback loops*. The idea behind a feedback loop is that you have a system whose output feeds into its input (the typical example being a thermostat). This idea isn’t new (see Powers, 1973), but Carver brilliantly formalized these ideas in this [2015 paper](https://journals.sagepub.com/doi/abs/10.1177/1754073915590616?journalCode=emra), using process diagrams to illustrate his model. In honor of Dr. Carver who passed away this year, I’m bringing his model to life in the form of a simulation.

**Simple Feedback Loop**

Let’s start with a simple type of feedback loop called a *discrepancy-reducing loop*. As we’ll see, the term implies that the object of the loop is to reduce a discrepancy between the system’s current state and an optimal, desired state. Imagine we have a goal of running a marathon and we’re tracking our progress toward that goal. Our desired outcome is to be 100% prepared for the marathon (note that we’re considering the act of running the marathon separate). Every day, we’ll compare our current progress to our desired outcome and this comparison will dictate how much effort we put into training that day. It’s easier to see with a process diagram:

This diagram describes a simple feedback loop whereby a Comparison between Current and Desired Progress affects behavior which, in turn, affects progress. We can emulate this in a simulation using R code.

Let’s return to our example of wannabe marathon runners. We’ll simulate 10 runners whose marathon preparedness progress starts at 0 and maxes out at 1. The code consists of nested loops. In the first loop (denoted by \(i\)) we have each of the 10 runners. We initialize their desired condition, which is 1 because they desire to be 100% prepared for the marathon. We then initialize the current condition to 0 to reflect their progress before the simulation starts.

In the second loop (denoted by \(j\)) we have each of the 100 days nested within each of the 10 runners. Each day, the runner compares their current progress with their desired condition. The larger this difference, the more *effort* they’ll put into training, which increases their chances of making progress. The closer they are to their goal, the less effort they put in. If one’s effort is successful, they make 2.5% progress, if they are not successful, there is a 50/50 chance that progress will decline by 2.5% or stay at it’s current state.

progress <- data.frame(matrix(nrow = 100, ncol = 10)) # initialize an empty dataframe in which to store progress

set.seed(1678) # for reproducibility

for(i in 1:10){

desired\_condition <- 1 # person desires to complete the task (100% completion)

current\_condition <- 0 # person starts with 0% completion

for(j in 1:100){

behavior <- desired\_condition - current\_condition # distance between desire and reality

environment <- sample(x = c(-.025, 0, .025), size = 1, prob = c(abs(1 - behavior) \* .50, abs(1 - behavior) \* .50, abs(behavior)))

current\_condition <- current\_condition + environment

progress[j, i] <- current\_condition

}

}

If we track progress over each day, here’s what it looks like:

library(tidyverse)

library(RColorBrewer)

progress <- progress %>%

mutate(time = 1:100)

progress\_gathered <- gather(progress, key = "person", value = "progress", -time)

progress\_gathered %>%

ggplot(aes(x = time, y = progress, group = person)) +

geom\_line(size = 1.25, color = "dodgerblue2", alpha = .65) +

labs(x = "Day",

y = "Progress toward Goal",

title = "Simple Feedback Loop") +

theme\_minimal(base\_size = 24) +

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

Each line represents a person in the simulation. We see a sharp spike in progress during the first 25 days, then it levels off as the runners approach their goal. Admittedly, the example might not perfectly apply to marathon runners, who may actually see less progress early on when they are in poorer physical shape. A more appropriate example might be weight loss, where people are initially highly motivated to lose weight, but then the person reaches an asymptote and less progress is made.

**Simple Feedback with an Outer Monitoring Loop**

So, our first example doesn’t do a great job of simulating *human* behavior. One major drawback is that progress is only affected by the distance between one’s current state and their desired state. People don’t always use such a wide perspective when contemplating their goals. Recent progress is perhaps more important. In the rhetoric of motivational speakers and self help books: take it one day at a time.

So now we’ll incorporate an *outer loop* that monitors activity in the *inner loop* that we built earlier. Importantly though, this monitoring will affect stuff in the inner loop, so it’s not just a passive observer. We’re going to add a *rate* parameter that measures the difference between one’s current progress and their progress of the previous day. This will be compared to their desired rate of progress, which we’ll choose to be 5% across the board. For greater clarity, here’s Carver’s take on how the outer loop functions:

“This second loop essentially checks on how well the first one is doing. Thus, the input for the second loop is a representation of the *rate of discrepancy reduction in the action system over time*” (emphasis in original).

— Carver (2015, p. 302)

We’ll use the word *approach* to define behavior directed toward a goal. Carver proposed that approach is essentially a *discrepancy-reducing* behavior (as we’ll see soon, there’s a counterpart to approach that is instrumental to understanding more complex behaviors). So the higher one’s approach, the greater chances are of success. Diagramatically, it looks like this:

Adding another feedback loop complicates things, but what’s important to see is that the organism is comparing both the overall progress and the rate of progress. We can think of it as reflecting on how well it’s doing – not just how far it is from its goal. Now we can add these parameters to the simulation.

set.seed(1272)

progress <- data.frame(matrix(nrow = 100, ncol = 10))

for(i in 1:10){

desired\_condition <- 1

desired\_rate <- .025 # desirable rate of progress (2.5% per instance)

current\_condition <- 0

rate\_comparison <- 1 # person starts with rate comparison of 1

for(j in 1:100){

# Inner Loop

comparison <- desired\_condition - current\_condition

approach <- comparison \* .50 + rate\_comparison \* .50

environment <- sample(x = c(-.025, 0, .025), size = 1, prob = c(abs(1 - approach) \* .50, abs(1 - approach) \* .50, abs(approach)))

current\_condition <- current\_condition + environment

progress[j, i] <- current\_condition

# Outer Loop

if(j == 1) {

current\_rate <- 0

} else {

current\_rate <- progress[j, i] - progress[j - 1, i]

}

rate\_comparison <- desired\_rate - current\_rate

}

}

Again, we can look at progress over time. From now on, I’ll use different colors for each person in the simulation.

progress <- progress %>%

mutate(time = 1:100)

progress\_gathered <- gather(progress, key = "person", value = "progress", -time)

progress\_gathered %>%

ggplot(aes(x = time, y = progress, color = person)) +

geom\_line(size = 1.25, alpha = .65) +

scale\_color\_brewer(palette = "Spectral") +

labs(x = "Day",

y = "Progress toward Goal",

title = "Feedback Loop with Monitoring") +

theme\_minimal(base\_size = 24) +

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

legend.position = "none")

Compared to before, the rate of progress is slower on average. There is also more variation between people, as some achieve 50% progress and others are not much farther from where they started. What’s happening is *approach* behavior increases when the rate of progress is less than desirable. However, when the rate of progress is greater than desirable, approach behavior decreases. This leads to a fluctuating pattern of ups and downs, similar to what occurs in people who diet.

**Feedback Loops and Mood**

In the last example, we used an *Outer Loop* to monitor the rate of progress in the system. We can think of *mood* as a byproduct of this outer loop – our subjective feeling about our progress. When we make progress toward our goals we feel happy and we relax. When we fall behind in our goals we feel upset and increase our efforts.

Now we’ll add to the outer loop a variable called *affect*, which is just the psychologist’s way of saying ‘mood’. Unlike my previous emotion simulations, we’ll stick to only one dimension of affect: valence (pleasant vs. unpleasant). High valence means you’re feeling happy, while low valence could mean sadness or anger.

Affect will be neutral for the first 5 days. After that, affect will be a function of *autocorrelation*, one’s current rate of progress, and their overall *progress slope* (i.e., extant rate of progress). The progress slope forecasts one’s overall rate of progress based on how well they’ve done so far. In this way, if someone is progressing at a slow rate, they will feel more negative.

We won’t bother with the diagrams from hereout because they become needlessly complicated. What’s important to know is that affect emerges as a byproduct of multiple comparisons between progress and desired progress. Here’s the revised R code for the simulation with affect.

set.seed(12581)

progress <- data.frame(matrix(nrow = 100, ncol = 10))

affect <- data.frame(matrix(nrow = 100, ncol = 10))

for(i in 1:10){

desired\_condition <- 1

desired\_rate <- .025

current\_condition <- 0

rate\_comparison <- 1

valence\_attractor <- rnorm(n = 1, mean = 0.3, sd = .01) # average emotional state

for(j in 1:100){

# Inner Loop

comparison <- desired\_condition - current\_condition

approach <- comparison \* .50 + rate\_comparison \* .50

environment <- sample(x = c(-.025, 0, .025), size = 1, prob = c(abs(1 - approach) \* .50, abs(1 - approach) \* .50, abs(approach)))

current\_condition <- current\_condition + environment

progress[j, i] <- current\_condition

# Outer Loop

if(j == 1) {

current\_rate <- 0

} else {

current\_rate <- progress[j, i] - progress[j - 1, i]

}

rate\_comparison <- desired\_rate - current\_rate

progress\_slope <- progress[j, i] - progress[1, i]/(j - 1)

if(j <= 5) {

affect[j, i] <- valence\_attractor + rnorm(n = 1, mean = valence\_attractor, sd = 1) \* .01

} else {

affect[j, i] <- affect[j - 1, i] \* .25 + progress\_slope \* .50 + (1 - rate\_comparison) \* .24 + rnorm(n = 1, mean = valence\_attractor, sd = 1) \* .01

}

}

}

Now that we have progress and affect, let’s look at them together. We can add two plots together using the *ggpubr* package.

library(ggpubr)

progress <- progress %>%

mutate(time = 1:100)

progress\_gathered <- gather(progress, key = "person", value = "progress", -time)

progress\_plot <- progress\_gathered %>%

ggplot(aes(x = time, y = progress, color = person)) +

geom\_line(size = 1.25, alpha = .65) +

scale\_color\_brewer(palette = "Spectral") +

labs(x = "Day",

y = "Progress toward Goal",

title = "Progress") +

theme\_minimal(base\_size = 20) +

theme(legend.position = "none",

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

axis.title.y = element\_text(size = 14))

affect <- affect %>%

mutate(time = 1:100)

affect\_gathered <- gather(affect, key = "person", value = "affect", -time)

affect\_plot <- affect\_gathered %>%

ggplot(aes(x = time, y = affect, color = person)) +

geom\_line(size = 1.25, alpha = .65) +

scale\_color\_brewer(palette = "Spectral") +

labs(x = "Day",

y = "Affect (higher = happier)",

title = "Affect") +

theme\_minimal(base\_size = 20) +

theme(legend.position = "none",

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

axis.title.y = element\_text(size = 14))

plots1 <- ggarrange(progress\_plot, affect\_plot, nrow = 2)

plots1

Affect is largely in-step with progress. Again, this is reflecting that people feel happier when they are making progress toward their goals. But we know that our mood is more complicated than this. Say we’re competing against others in a diet challenge to see who can lose the most weight over 100 days. If we see that we’re way behind in our progress, with no hope of catching up, we may throw in the towel and quit trying altogether. Indeed, humans are highly attuned to *social comparisons*, especially in competitive environments.

**Feedback Loops, Mood, and Social Comparison**

Our last and most complex simulation will take into account how people’s behavior toward a goal changes when they receive social comparison feedback. If you have no chance winning the competition, why expend effort at all? Things get even more complicated from here, as we now have to create dependence among organisms in the simulation. Before, everyone’s performance was independent; now each new person who enters the simulation becomes aware of their performance relative to their peers’.

We add two new parameters. The first, *average progress rate*, is the average slope of prior runs in the simulation (akin to watching your competitor’s performance and then yourself performing). Indeed, these are not run in parallel, so the first simulated person has no insight into the performance of others, while the second person is only aware of how well the first person did, and so on. The second parameter is *relative standing* and it is simply the difference between one’s current progress rate and the average progress rate. In general, as relative standing increases (as the distance becomes larger) avoidance behavior increases. This works in two ways: (1) performance exceeds that of others and effort goes down (e.g., coasting) or (2) performance pales relative to others and hopelessness ensues. To model this, at the halfway point, if one’s relative standing is 35% less than the average, that person will effectively give up, saving themselves from severe upset. Indeed, some have suggested that this is an adaptive response because it allows people to reprioritize their efforts (Simon, 1967).

set.seed(1249)

progress <- data.frame(matrix(nrow = 100, ncol = 10))

affect <- data.frame(matrix(nrow = 100, ncol = 10))

for(i in 1:10){

desired\_condition <- 1

desired\_rate <- .050

current\_condition <- 0

rate\_comparison <- 1

valence\_attractor <- rnorm(n = 1, mean = 0.3, sd = .01)

average\_progress\_rate <- 0

relative\_standing <- 1

for(j in 1:100){

# Inner Loop

comparison <- desired\_condition - current\_condition

approach <- (abs(comparison \* .50 + rate\_comparison \* .50))/(abs(relative\_standing) + (comparison \* .75 + abs(rate\_comparison) \* .25))

avoidance <- (abs(relative\_standing))/(abs(relative\_standing) + (comparison \* .75 + abs(rate\_comparison) \* .25))

environment <- sample(x = c(-.025, 0, .025), size = 1, prob = c(avoidance \* .50, avoidance \* .50, approach))

current\_condition <- current\_condition + environment

progress[j, i] <- ifelse(current\_condition <= 0, 0, current\_condition)

# Outer Loop

if(j == 1) {

current\_rate <- 0

} else {

current\_rate <- progress[j, i] - progress[j - 1, i]

progress\_slope <- progress[j, i] - progress[1, i]/(j - 1)

relative\_standing <- progress\_slope - average\_progress\_rate

}

rate\_comparison <- desired\_rate - current\_rate

if(j <= 5) {

affect[j, i] <- valence\_attractor + rnorm(n = 1, mean = valence\_attractor, sd = 1) \* .01

} else if (j >= 50 && relative\_standing <= .35) {

desired\_condition <- current\_condition

rate\_comparison <- 0

affect[j, i] <- affect[j - 1, i] \* .99 + rnorm(n = 1, mean = valence\_attractor, sd = 1) \* .01

} else {

affect[j, i] <- affect[j - 1, i] \* .25 + progress\_slope \* .50 + relative\_standing \* .24 + rnorm(n = 1, mean = valence\_attractor, sd = 1) \* .01

}

}

if(i > 1){

average\_progress\_rate <- mean(t(progress[100, (i - 1):1]))

} else {

average\_progress\_rate <- 0

}

}

Running the code above, we can enter it into the next code chunk to get the plots. Again, we’ll plot them together to compare progress and affect.

progress <- progress %>%

mutate(time = 1:100)

progress\_gathered <- gather(progress, key = "person", value = "progress", -time)

progress\_plot2 <- progress\_gathered %>%

ggplot(aes(x = time, y = progress, color = person)) +

geom\_line(size = 1.25, alpha = .65) +

scale\_color\_brewer(palette = "Spectral") +

labs(x = "Day",

y = "Progress toward Goal",

title = "Progress") +

theme\_minimal(base\_size = 20) +

theme(legend.position = "none",

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

axis.title.y = element\_text(size = 14))

affect <- affect %>%

mutate(time = 1:100)

affect\_gathered <- gather(affect, key = "person", value = "affect", -time)

affect\_plot2 <- affect\_gathered %>%

ggplot(aes(x = time, y = affect, color = person)) +

geom\_line(size = 1.25, alpha = .65) +

scale\_color\_brewer(palette = "Spectral") +

labs(x = "Day",

y = "Affect (higher = happier)",

title = "Affect") +

theme\_minimal(base\_size = 20) +

theme(legend.position = "none",

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

axis.title.y = element\_text(size = 14))

plots2 <- ggarrange(progress\_plot2, affect\_plot2, nrow = 2)

plots2

We can clearly see where a few competitors bailed out and their progress dropped to zero. This is because their relative standing was weak and they relinquished. Importantly though, they didn’t suffer too heavy a blow in affect, and this is because they no longer valued progress toward this goal.

**Conclusion**

We set out to show that motivation and emotion can be simulated using fairly simple feedback loops. Of course, human behavior is exceedingly more complicated, but we can learn a lot by simulating ersatz humans with only a handful of parameters. As I continue to work toward my PhD, my appreciation for Carver’s computational models only grows stronger. My goal is to see psychological research and computer simulations partake in their own form of feedback loop, where knowledge from the former feeds into our construction of the latter.