It’s Valentine’s day, making this the most romantic time of the year. But actually, already 2018 was a year full of love here at [STATWORX](https://www.statworx.com/): many of my STATWORX colleagues got engaged. And so we began to wonder – some fearful, some hopeful – who will be next?  
Therefore, today we’re going to tackle this question in the only true way: with data science!

**Gathering the Data**

To get my data, I surveyed my colleagues. I asked my (to be) married colleagues to answer my questions based on the very day they got engaged. My single colleagues answered my questions with respect to their current situation. I asked them about some factors that I’ve always suspected to influence someone’s likeliness to get married. For example, I’m sure that in comparison to Python users, R users are much more romantic. The indiscreet questions I badgered my coworkers with were:

* Are you married or engaged?
* How long have you been in your relationship?
* Is your employment permanent?
* How long have you been working at STATWORX?
* What’s your age?
* Are you living together with your partner?
* Are you co-owning a pet with your partner?
* What’s your preferred programming language: R, Python or none of both.

I’m going to treat the relationship status as dichotomous variable: Married or engaged vs. single or “only” dating. To maintain some of the privacy of my colleagues I gave them all some randomly (!!) chosen pet names. (Side note: There really is a subreddit for everything.)

**Descriptive Exploration**

Since the first step in generating data driven answer should always be a descriptive exploration of the data at hand, I made some plots.

First, I took a look at the absolute frequencies of preferred programming languages in the groups of singles vs. married or engaged STATWORX employees. I fear, the romantic nature of R users is not the explanation we’re looking for:

# reformatting the target variable

df1 <- df %>%

dplyr::mutate(engaged = ifelse(engaged == "yes",

"Engaged or Married",

"Single")) %>%

dplyr::group\_by(`primary programming language`, engaged) %>%

dplyr::summarise(freq = n(),

image = "~/Desktop/heart\_red.png")

# since in geom\_image size cannot be mapped to variable

# multiple layers of data subsets

ggplot() +

geom\_image(data = filter(df1, freq == 1),

aes(y = `primary programming language`,

x = engaged,

image = image),

size = 0.1) +

geom\_image(data = filter(df1, freq > 1 & freq <= 5),

aes(y = `primary programming language`,

x = engaged,

image = image),

size = 0.2) +

geom\_image(data = filter(df1, freq >= 13),

aes(y= `primary programming language`,

x = engaged,

image = image),

size = 0.3) +

geom\_text(data = df1,

aes(y =`primary programming language`,

x = engaged,

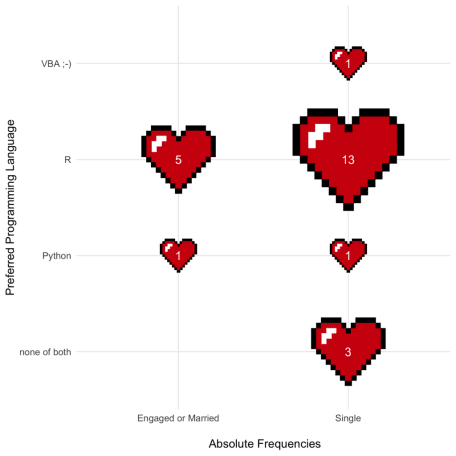
label = freq),

color = "white", size = 4) +

ylab("Preferred programming language") +

xlab("\n Absolute frequencies") +

theme\_minimal()



I also explored the association of relationship status and the more conventional factors of age and relationship duration. And indeed, those of my colleagues who are in their late twenties or older and have been partnered for a while now, are mostly married or engaged.

# plotting age and relationship duration vs. relationship status

ggplot() +

# doing this only to get a legend:

geom\_point(data = df,

aes(x = age, y = `relationship duration`,

color = engaged), shape = 15) +

geom\_image(data = filter(df, engaged == "yes"),

aes(x = age, y = `relationship duration`,

image = "~/Desktop/heart\_red.png")) +

geom\_image(data = filter(df, engaged == "no"),

aes(x = age, y = `relationship duration`,

image = "~/Desktop/heart\_black.png")) +

ylab("Relationship duration \n") +

xlab("\n Age") +

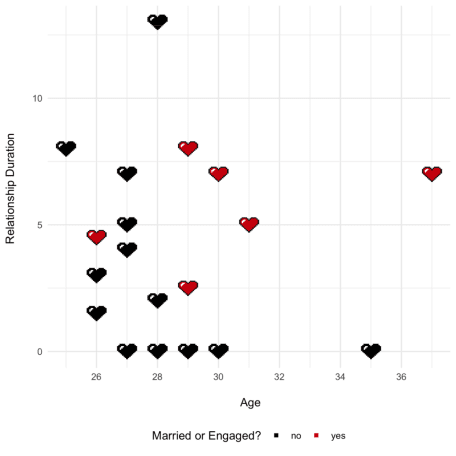
scale\_color\_manual(name = "Married or engaged?",

values = c("#000000", "#D00B0B")) +

scale\_x\_continuous(breaks = pretty\_breaks()) +

theme\_minimal() +

theme(legend.position = "bottom")



**Statistical Models**

I’ll employ some statistical models, but the data base is rather small. Therefore, our model options are somewhat limited (and of course only suitable for entertainment).  
But it’s still possible to fit a decision tree, which might help to pinpoint due to which circumstance some of us are still waiting for that special someone to put a ring on (it).

# recoding target to get more understandable labels

df <- df %>%

dplyr::mutate(engaged = ifelse(engaged == "yes",

"(to be) married",

"single"))

# growing a decision three with a ridiculous low minsplit

fit <- rpart(engaged ~ `relationship duration` + age +

`shared pet` + `permanent employment` +

cohabitating + `years at STATWORX`,

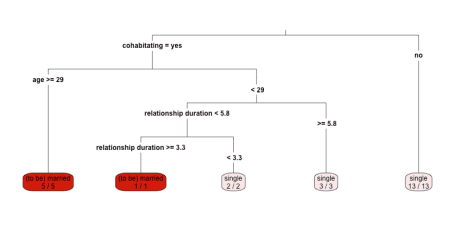
control = rpart.control(minsplit = 2), # overfitting ftw

method = "class", data = df)

# plotting the three

rpart.plot(fit, type = 3, extra = 2,

box.palette = c("#D00B0B", "#fae6e6"))



Our decision tree implies, that the unintentionally unmarried of us maybe should consider to move in with their partner, since cohabitating seems to be the most important factor.

But that still doesn’t exactly answer the question, who of us will be next. To predict our chances to get engaged, I estimated a logistic regression.

We see that cohabiting, one’s age and the time we’ve been working at STATWORX are accompanied by a higher probability to (soon to) be married. However, simply having been together for a long time or owning a pet together with our partner, does not help. (Although, I assume that this rather unintuitive interrelation is caused by a certain outlier in the data – “Honey”, I’m looking at you!)

Finally, I got the logistic regression’s predicted probabilities for all of us to be married or engaged. As you can see down below, the single days of “Teddy Bear”, “Honey”, “Sweet Pea” and “Babe” seem to be numbered.

# reformatting the target variable

df <- df %>%

dplyr::mutate(engaged = ifelse(engaged == "(to be) married", 1, 0))

# in-sample fitting: estimating the model

log\_reg <- glm(engaged ~ `relationship duration` + age +

`shared pet` + `permanent employment` +

cohabitating + `years at STATWORX`,

family = binomial, data = df)

df$probability <- predict(log\_reg, df, type = "response")

# plotting the predicted probabilities

ggplot() +

# again, doing this only to get a legend:

geom\_point(data = df,

aes(x = probability, y = nickname,

color = as.factor(engaged)), shape = 15) +

geom\_image(data = filter(df, engaged == 1),

aes(x = probability, y = nickname,

image = "~/Desktop/heart\_red.png")) +

geom\_image(data = filter(df, engaged == 0),

aes(x = probability, y = nickname,

image = "~/Desktop/heart\_black.png")) +

ylab(" ") +

xlab("Predicted Probability") +

scale\_color\_manual(name = "Married or engaged?",

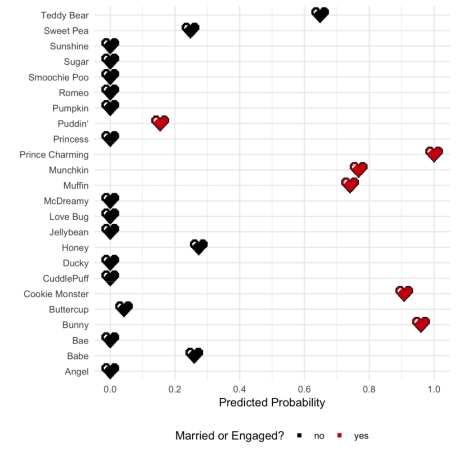
values = c("#000000", "#D00B0B"),

labels = c("no", "yes")) +

scale\_x\_continuous(breaks = pretty\_breaks()) +

theme\_minimal() +

theme(legend.position = "bottom")



I hope this was as insightful for you as it was for me. And to all of us, whose hopes have been shattered by cold, hard facts, let’s remember: there are tons of discounted chocolates waiting for us on February 15th.