Logistic Regression with Kickstarter Data

Marcelo Sanches

August 23, 2018

Project Overview and Limitations

This project is a basic introduction to logistic regression using a simple kickstarter dataset with a few variables. The scope is limited and for actual predictions, a full assessment of confounding variables and better datasets should be considered, as well as other, more advanced techniques. One of the main limitations is that predictions made here only work within this dataset, so the next step is to partition the dataset into training and validation sets so as to test how well predictions hold out of sample.

The Dataset

The data is freely available in Kaggle after registration: **Kickstarter Data.**

The 2018 dataset consists of data from 378,661 kickstarter projects such as amount pledged and goal amount, currency for those figures, number of backers, final project outcome (i.e. 'state'), country, deadline and launch date for a project.

1. Data Cleaning and Preparation

Downloading, Loading, Cleaning Dataset

Only the 2018 dataset is considered. The following variables were removed:

- 1:2: ID and name, unnecessary
- 3: category, too detailed
- 5: currency, unnecessary since analysis focuses on US projects
- 9: pledged, a data leakage problem: cannot predict on information that is unavailable at the start of a project
- 11: backers, a data leakage problem: as above
- 13-15: usd.pledged, etc, data leakage problems

First we cleanup the workspace, download the data, load it into R, and look at the first few rows:

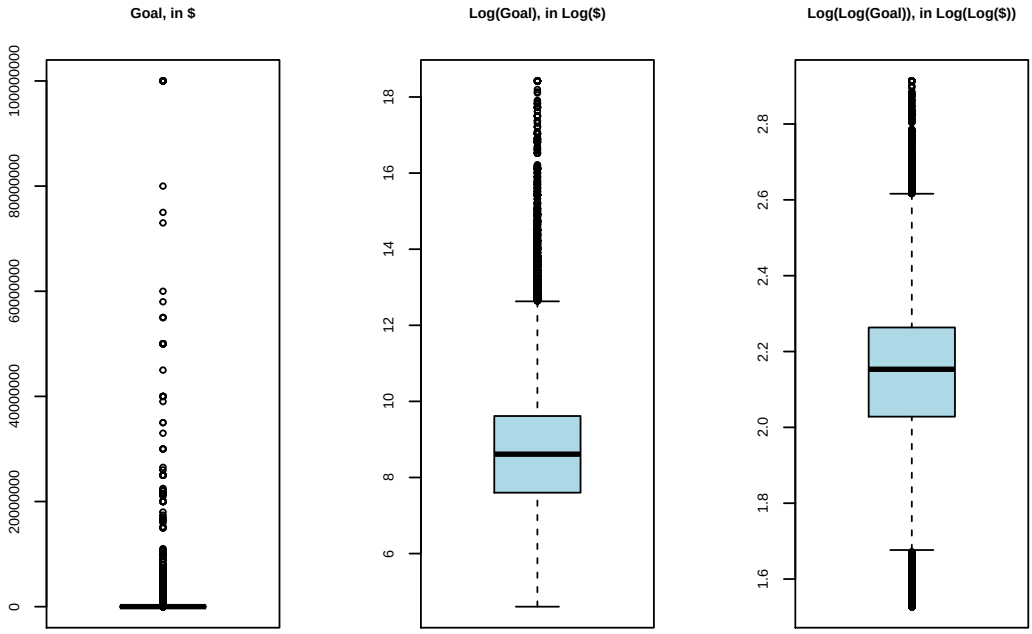
```
##   main_category   deadline  goal            launched      state country
## 1    Publishing 2015-10-09  1000 2015-08-11 12:12:28     failed      GB
## 2  Film & Video 2017-11-01 30000 2017-09-02 04:43:57     failed      US
## 3  Film & Video 2013-02-26 45000 2013-01-12 00:20:50     failed      US
## 4         Music 2012-04-16  5000 2012-03-17 03:24:11     failed      US
## 5  Film & Video 2015-08-29 19500 2015-07-04 08:35:03   canceled      US
## 6          Food 2016-04-01 50000 2016-02-26 13:38:27 successful      US
```

We focus on US projects, remove "live" projects since we don't know the outcome of those yet, convert factor variables to date ones, compute a new "duration" variable (i.e. project length) by subtracting "launched" date from "deadline", and re-order variables.

We dummy-code the 'state' variable to predict success when 'state' = 1, versus failure, when 'state' = 0. We discard 6 senseless date outliers (projects from 1970) and discard 2,909 projects under $100 in goal, which are probably gaming the system using kickstarter's promotion of projects with a high pledged-to-goal ratio.

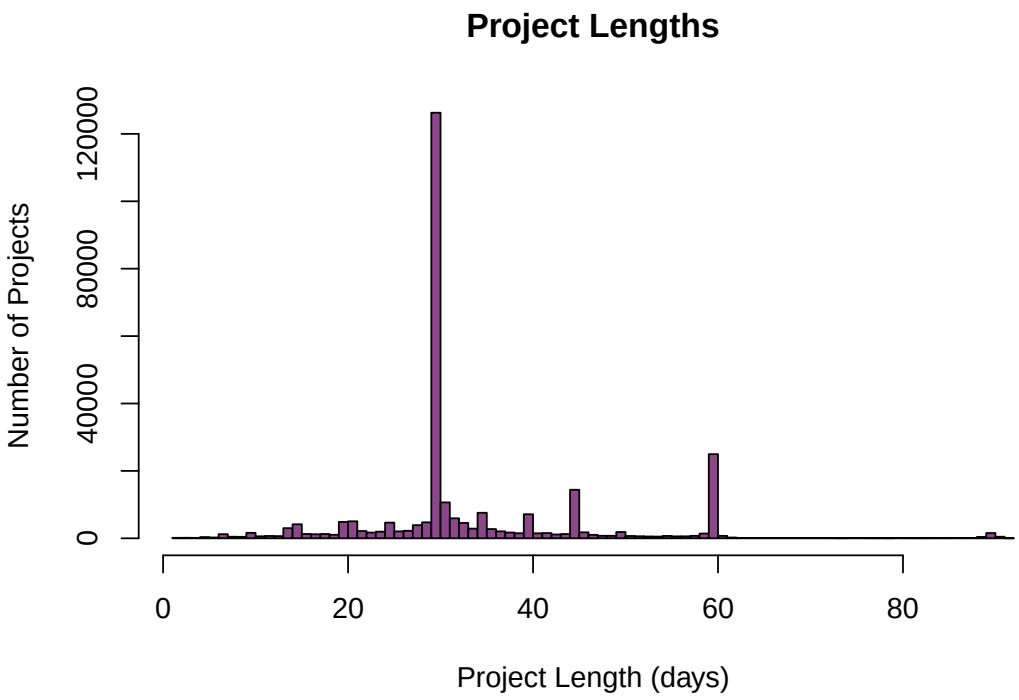
```
##   state   launched   deadline duration  goal main_category
## 2     0 2017-09-02 2017-11-01       60 30000  Film & Video
## 3     0 2013-01-12 2013-02-26       45 45000  Film & Video
## 4     0 2012-03-17 2012-04-16       30  5000         Music
## 5     0 2015-07-04 2015-08-29       56 19500  Film & Video
## 6     1 2016-02-26 2016-04-01       35 50000          Food
## 7     1 2014-12-01 2014-12-21       20  1000          Food
```

Next, we transform the 'goal' variable since it has a very skewed distribution:



Then we dummy-code the main categories, which were 15 total but since there are many trailing categories with little representation, we create 7 dummies total and bin the trailing categories into an 'other' category. Details of how this is done can be found in the code appendix.

Duration has an uneven, modal distribution as 30 is the default number of days for a project in Kickstarter, so it cannot be used in logistic regression which expects normality of the data.



We transform this continuous variable into a categorical variable with 4 levels: 1-29 days, 30-39 days, 40-59 days, and 60-92 days in project length. This is what the data looks like just prior to fitting a regression model:

```
##    state logloggoal dur30_39 dur40_59 dur_60_92 music publishing games art
## 2      0   2.333013        0        0         1     0          0     0   0
## 3      0   2.371590        0        1         0     0          0     0   0
## 4      0   2.142087        1        0         0     1          0     0   0
## 5      0   2.290327        0        1         0     0          0     0   0
## 6      1   2.381376        1        0         0     0          0     0   0
## 7      1   1.932645        0        0         0     0          0     0   0
## 8      0   2.315169        0        1         0     0          0     0   0
## 9      0   2.462667        1        0         0     0          0     0   0
## 10     0   2.405335        1        0         0     0          0     0   0
## 12     1   2.244265        1        0         0     1          0     0   0
##    design technology other
## 2       0          0     0
## 3       0          0     0
## 4       0          0     0
## 5       0          0     0
## 6       0          0     1
## 7       0          0     1
## 8       0          0     1
## 9       1          0     0
## 10      0          0     0
## 12      0          0     0
```

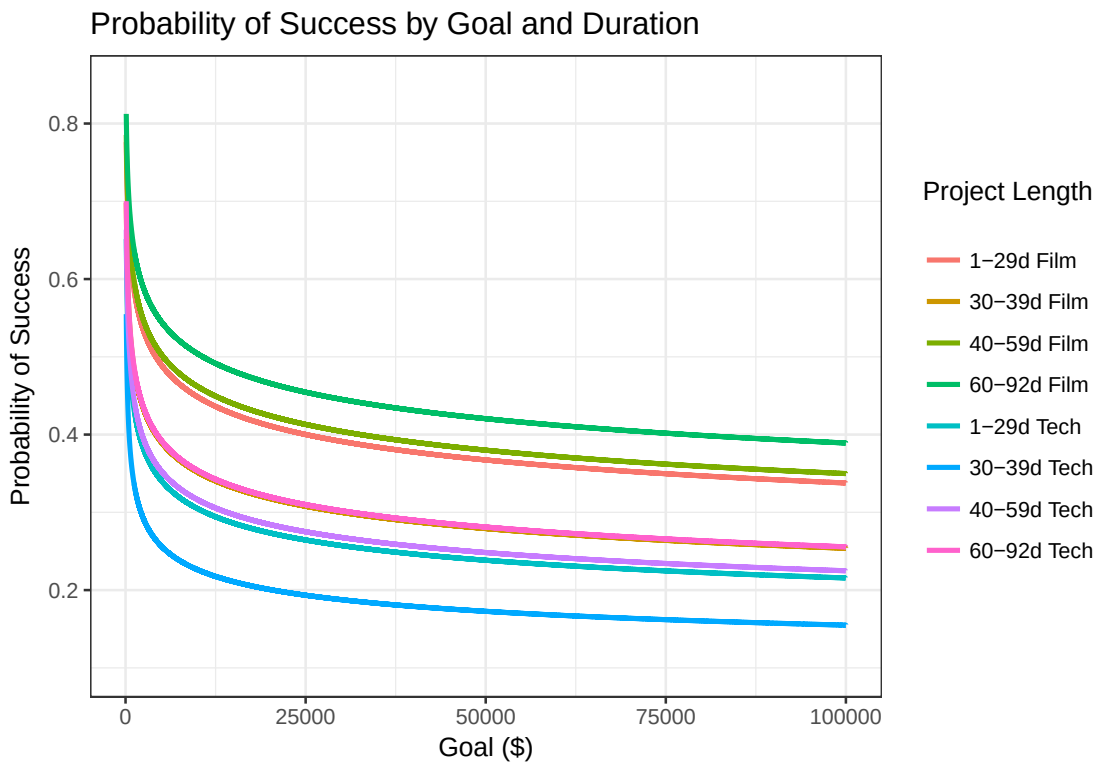
2. Data Analysis

We use a logistic regression model with an interaction between duration and goal to predict project success.

```
##
## Call:
## glm(formula = state ~ dur30_39 * logloggoal + dur40_59 * logloggoal +
##     dur_60_92 * logloggoal + music + publishing + games + art +
##     design + technology + other, family = binomial, data = kickstarter)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8906  -0.9555  -0.7408   1.2173   2.4454
##
## Coefficients:
##                       Estimate Std. Error z value        Pr(>|z|)
## (Intercept)            4.43768    0.10303  43.072         < 2e-16 ***
## dur30_39              -0.44645    0.12054  -3.704        0.000212 ***
## logloggoal            -2.09175    0.04930 -42.426         < 2e-16 ***
## dur40_59               0.46695    0.17604   2.652        0.007990 **
## dur_60_92              1.34716    0.20277   6.644 0.0000000000305 ***
## music                  0.36074    0.01374  26.255         < 2e-16 ***
## publishing            -0.45884    0.01569 -29.250         < 2e-16 ***
## games                 -0.04811    0.01658  -2.901        0.003720 **
## art                   -0.10699    0.01718  -6.229 0.0000000004700 ***
## design                -0.05912    0.01746  -3.386        0.000710 ***
## technology            -0.61791    0.01945 -31.776         < 2e-16 ***
## other                 -0.26228    0.01236 -21.229         < 2e-16 ***
## dur30_39:logloggoal    0.04141    0.05751   0.720        0.471521
## logloggoal:dur40_59   -0.41321    0.08233  -5.019 0.0000005195989 ***
## logloggoal:dur_60_92  -1.12461    0.09543 -11.784         < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 380748  on 287971  degrees of freedom
## Residual deviance: 358364  on 287957  degrees of freedom
## AIC: 358394
##
## Number of Fisher Scoring iterations: 4
```

Calculating prediction probabilities for plotting required many lines of code, found in the appendix.

We focus on Film and Video versus Technology categories, since the Technology coefficient was the furthest from the base case (Film and Video). This helps the visualization considering we have four duration probability lines plotted per category.

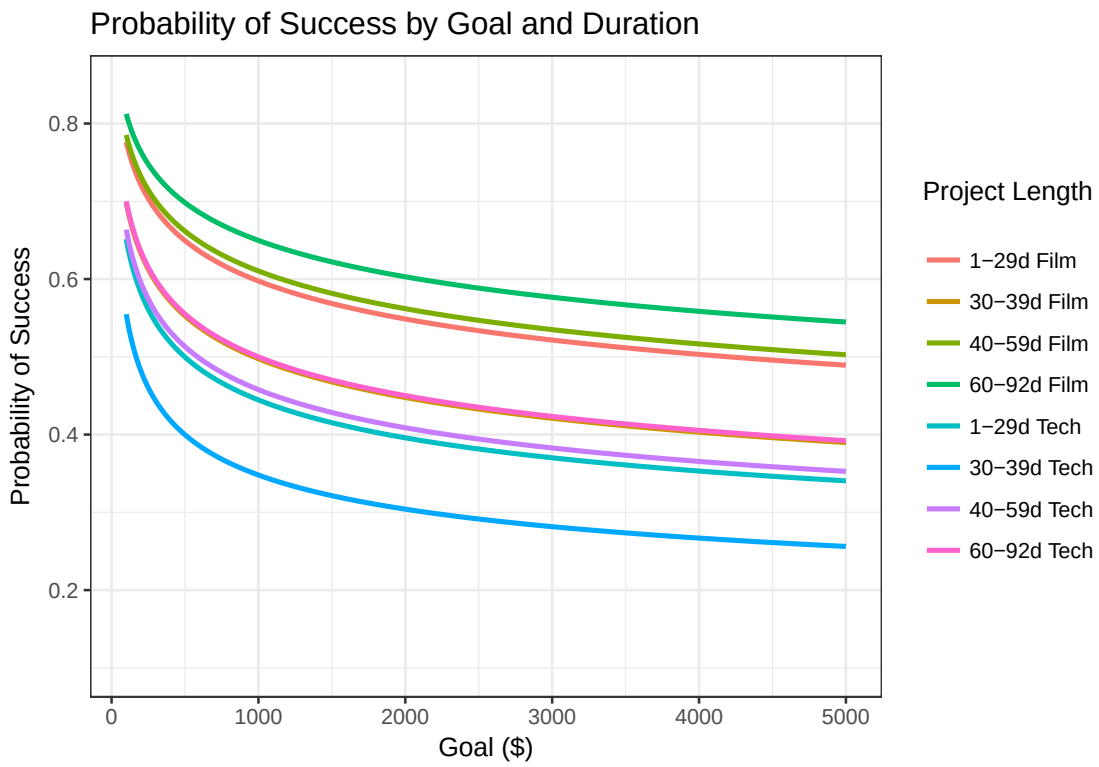


Film and Video projects have higher probability of success in general compared to Tech projects by approx. 13% according to this model.

Longer projects (60 to 92 days) have a higher probability of success, followed by 40-to-59-day projects and 1-to-29-day projects. The modal project length category from 30 to 39 days has the lowest probability of success.

The longest project category in Technology has slightly better (0.2% difference) probabilities of success compared to the modal project length in Film and Video.

As the goal increases, the probability of success decreases. The rate of decrease is steeper for the first $2,000 or so, and levels off after that, as we can see in this final plot.



Code Appendix

Data Cleaning and Preparation

```
# cleanup workspace
rm(list=ls())

# download data
download.file(
"https://www.kaggle.com/kemical/kickstarter-projects/downloads/kickstarter-projects.zip/7",
destfile = "ks-projects-201801.csv")

# load pertinent variables
ks18 <- read.csv(
file = "./kickstarter/ks-projects-201801.csv", header = TRUE)[,-c(1:3, 5, 9, 11, 13:15)]
head(ks18)
```

```
# subset US projects, remove country variable
us18 <- ks18[ks18$country == "US", -6]

# remove "live" projects (no outcome to base prediction on)
us18 <- us18[us18$state != "live", ]

# convert factors to date variables
us18$deadline <- as.Date(us18$deadline)
```

```r
us18$launched <- as.Date(us18$launched)

# compute duration (project length) variable
us18$duration <- as.numeric(us18$deadline - us18$launched)

# re-order data
us18 <- us18[,c(5,4,2,6,3,1)]

# dummy-code outcome variable 'state' to predict success
us18$state <- ifelse(us18$state == "successful", 1, 0)

# discard senseless date outliers (projects from 1970)
us18 <- us18[us18$duration < 100, ]

# discard 2909 projects under $100 (likely gaming the system)
us18 <- us18[us18$goal > 99, ]
head(us18)
```

```r
# boxplots for tranformation of goal
par(mfrow=c(1,3), cex.lab=.8, cex.axis=.8, cex.main=.8)
options(scipen=8)
boxplot(us18$goal,
    main = "Goal, in $", col ="magenta")
boxplot(log(us18$goal),
    main = "Log(Goal), in Log($)", col ="lightblue")
boxplot(log(log(us18$goal)),
    main = "Log(Log(Goal)), in Log(Log($))", col ="lightblue")
```

```r
# transform goal distribution to log(log(goal))
# so as to normalize it for regression
us18$logloggoal <- log(log(us18$goal))

# dump goal and re-order dataset
us18 <- us18[, -5]; us18 <- us18[,c(1,2,3,4,6, 5)]
```

```r
# dummy-code main categories (15 total, but 7 dummies)
# base case: Film and Video, + other category
us18$music  <- ifelse(us18$main_category == "Music", 1,0)
us18$publishing <- ifelse(us18$main_category == "Publishing", 1,0)
us18$games <- ifelse(us18$main_category == "Games", 1,0)
us18$art <- ifelse(us18$main_category == "Art", 1,0)
us18$design <- ifelse(us18$main_category == "Design", 1,0)
us18$technology <- ifelse(us18$main_category == "Technology", 1,0)

# other category
us18$other <- ifelse(
    us18$main_category == "Food" | us18$main_category == "Fashion" |
    us18$main_category == "Comics" | us18$main_category == "Theater" |
    us18$main_category == "Photography" | us18$main_category == "Crafts" |
    us18$main_category == "Journalism" | us18$main_category == "Dance",
    1, 0)

us18 <- us18[, -c(2:3,6)]
```

```r
# duration's uneven, modal distribution
hist(us18$duration, 100, col="orchid4",
    main="Project Lengths",
    ylab="Number of Projects",
    xlab="Project Length (days)")
```

```r
# re-code and bin it into 4 categories (base case: 1 to 29 days)
us18$dur30_39 <- ifelse(us18$duration > 29 & us18$duration <= 39, 1, 0)
us18$dur40_59 <- ifelse(us18$duration > 39 & us18$duration <= 59, 1, 0)
us18$dur_60_92 <- ifelse(us18$duration > 59 & us18$duration <= 92, 1, 0)

# dump old duration variable, rename, reorder
kickstarter <- us18[, -2]; kickstarter <- kickstarter[,c(1,2,10:12,3:9)]
kickstarter[1:10, ]
```

Data Analysis

```r
# logistic regression
mod1 <- glm(state ~ dur30_39 * logloggoal +
                    dur40_59 * logloggoal +
                    dur_60_92 * logloggoal +
                    music + publishing + games + art + design + technology + other,
                    family = binomial, data=kickstarter)
summary(mod1)
```

```r
# name/assign coefficients
# a0 coef means 1_29-day project in Film/Video with $0 logloggoal
a0 <- coef(mod1)[1]
b_dur30_39 <- coef(mod1)[2]
b_logloggoal <- coef(mod1)[3]
b_dur40_59 <- coef(mod1)[4]
b_dur60_92 <- coef(mod1)[5]
b_music <- coef(mod1)[6]
b_publishing <- coef(mod1)[7]
b_games <- coef(mod1)[8]
b_art <- coef(mod1)[9]
b_design <- coef(mod1)[10]
b_technology <- coef(mod1)[11]
b_other <- coef(mod1)[12]
b_dur30_39int <- coef(mod1)[13]
b_dur40_59int <- coef(mod1)[14]
b_dur60_92int <- coef(mod1)[15]

## initialize probability vectors
Fprobs29 <- NA; Fprobs39 <- NA; Fprobs59 <- NA; Fprobs92 <- NA
Mprobs29 <- NA; Mprobs39 <- NA; Mprobs59 <- NA; Mprobs92 <- NA
Pprobs29 <- NA; Pprobs39 <- NA; Pprobs59 <- NA; Pprobs92 <- NA
Gprobs29 <- NA; Gprobs39 <- NA; Gprobs59 <- NA; Gprobs92 <- NA
Aprobs29 <- NA; Aprobs39 <- NA; Aprobs59 <- NA; Aprobs92 <- NA
Dprobs29 <- NA; Dprobs39 <- NA; Dprobs59 <- NA; Dprobs92 <- NA
Tprobs29 <- NA; Tprobs39 <- NA; Tprobs59 <- NA; Tprobs92 <- NA
Oprobs29 <- NA; Oprobs39 <- NA; Oprobs59 <- NA; Oprobs92 <- NA
```

```r
# Calculating probabilities from $100 to $100,000 (in goal)
# for various categories and durations

# Fprobs = Film  & Video
# 1-29 days
for (i in 100:100000) {
    regr <- a0 +
            b_logloggoal * log(log(i))
    Fprobs29[i] <- unname(exp(regr)/(1+exp(regr)))
}
Fprobs29 <- Fprobs29[-c(1:99)]

# 30-39 days
for (i in 100:100000) {
    regr <- a0 +
            b_logloggoal * log(log(i)) +
            b_dur30_39 * 1 +
            b_dur30_39int * 1
    Fprobs39[i] <- unname(exp(regr)/(1+exp(regr)))
}
Fprobs39 <- Fprobs39[-c(1:99)]

# 40-59 days
for (i in 100:100000) {
    regr <- a0 +
            b_logloggoal * log(log(i)) +
            b_dur40_59 * 1 +
            b_dur40_59int * 1
    Fprobs59[i] <- unname(exp(regr)/(1+exp(regr)))
}
Fprobs59 <- Fprobs59[-c(1:99)]

# 60-92 days
for (i in 100:100000) {
    regr <- a0 +
            b_logloggoal * log(log(i)) +
            b_dur60_92 * 1 +
            b_dur60_92int * 1
    Fprobs92[i] <- unname(exp(regr)/(1+exp(regr)))
}
Fprobs92 <- Fprobs92[-c(1:99)]

## Tprobs = Technology
# 1-29 days
for (i in 100:100000) {
    regr <- a0 +
            b_logloggoal * log(log(i)) +
            b_technology
    Tprobs29[i] <- unname(exp(regr)/(1+exp(regr)))
}
Tprobs29 <- Tprobs29[-c(1:99)]

# 30-39 days
```

```
for (i in 100:100000) {
    regr <- a0 +
            b_logloggoal * log(log(i)) +
            b_dur30_39 * 1 +
            b_dur30_39int * 1 +
            b_technology
    Tprobs39[i] <- unname(exp(regr)/(1+exp(regr)))
}
Tprobs39 <- Tprobs39[-c(1:99)]

# 40-59 days
for (i in 100:100000) {
    regr <- a0 +
            b_logloggoal * log(log(i)) +
            b_dur40_59 * 1 +
            b_dur40_59int * 1 +
            b_technology
    Tprobs59[i] <- unname(exp(regr)/(1+exp(regr)))
}
Tprobs59 <- Tprobs59[-c(1:99)]

# 60-92 days
for (i in 100:100000) {
    regr <- a0 +
            b_logloggoal * log(log(i)) +
            b_dur60_92 * 1 +
            b_dur60_92int * 1 +
            b_technology
    Tprobs92[i] <- unname(exp(regr)/(1+exp(regr)))
}
Tprobs92 <- Tprobs92[-c(1:99)]
```

```
# Visualization - Film & Video vs. Technology
# install-load required packages
if("ggplot2" %in% rownames(installed.packages()) == FALSE) {
    suppressWarnings(install.packages("ggplot2"))
}
suppressMessages(require(ggplot2))
if("reshape2" %in% rownames(installed.packages()) == FALSE) {
    suppressWarnings(install.packages("reshape2"))
}
suppressMessages(require(reshape2))

# data frame to hold probabilities
dfm <- data.frame(
        "Goal" = 100:100000,
        "Fprobs29" = Fprobs29,
        "Fprobs39" = Fprobs39,
        "Fprobs59" = Fprobs59,
        "Fprobs92" = Fprobs92,
        "Tprobs29" = Tprobs29,
        "Tprobs39" = Tprobs39,
        "Tprobs59" = Tprobs59,
        "Tprobs92" = Tprobs92
```

```
        )

# tidy data frame with factor variable for prob type and numeric variable of probs
dfm.melt <- melt(dfm, id = "Goal")

ggplot(data=dfm.melt, aes(x=Goal,y=value, color=variable)) +
    geom_line(size=1) +
    ylim(0.1,0.85) +
    labs(title = "Probability of Success by Goal and Duration",
        x = "Goal ($)", y = "Probability of Success", color = "Project Length\n") +
    scale_color_hue(labels = c("1-29d Film", "30-39d Film", "40-59d Film", "60-92d Film",
                               "1-29d Tech", "30-39d Tech", "40-59d Tech", "60-92d Tech")) +
    theme_bw()
```

```
# Film & Video vs. Tech mean prob of success
mean(c(Fprobs29,Fprobs39,Fprobs59,Fprobs92))-mean(c(Tprobs29,Tprobs39,Tprobs59,Tprobs92))

# 0.2% difference in highest prob for Tech vs lowest prob for Film & Video
mean(Tprobs92) - mean(Fprobs39)
```

```
# final plot = up to $5,000 goal
dfm <- dfm[1:4901,]
dfm.melt <- melt(dfm, id = "Goal")
ggplot(data=dfm.melt, aes(x=Goal,y=value, color=variable)) +
    geom_line(size=1) +
    ylim(0.1,0.85) +
    labs(title = "Probability of Success by Goal and Duration",
        x = "Goal ($)", y = "Probability of Success", color = "Project Length\n") +
    scale_color_hue(labels = c("1-29d Film", "30-39d Film", "40-59d Film", "60-92d Film",
                               "1-29d Tech", "30-39d Tech", "40-59d Tech", "60-92d Tech")) +
    theme_bw()
```
