Digging for Gold

Perry Stephenson

2018-04-21

```
library(tidyverse)
library(goldminer) # https://github.com/perrystephenson/goldminer
```

```
config <- list()
params <- list()
env1 <- list()
env2 <- list()
config$rounds <- 1e5L # 1 hundred thousand rounds
config$years <- 5
```

Approach

This analysis aims to simulate the environment independently from the decision making, so that the impacts of the decision making can be assessed in isolation from the assumed parameters driving the simulation. Accordingly, each of the environment variables will be modelled based on the provided parameters regardless of whether or not they are required in any individual simulation. Once the environment has been modelled, the impacts of decisions made by the business can be reasoned about without altering the modelled environment.

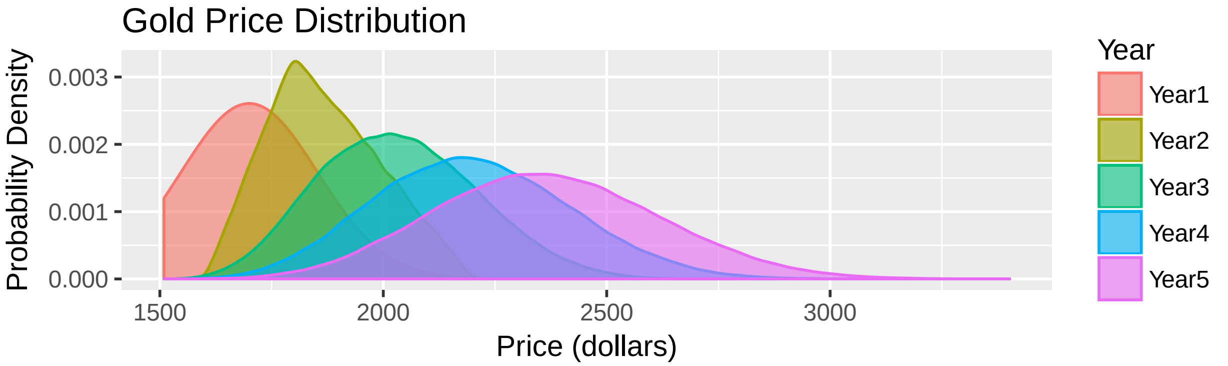
Environment

Gold Price

As specified, the initial price of gold in the first year is $1700, and the annual (absolute) change in the price is between -100 and 500, with an expected value of 100.

```
params$gold_price_init <- 1700
params$gold_price_delta <- list(a = -100, b = 500, c = 100)
```

```
env1$gold_price <- gen_gold_price(rounds = config$rounds,
                                  years = config$years,
                                  gold_price_init = params$gold_price_init,
                                  gold_price_delta = params$gold_price_delta)
```



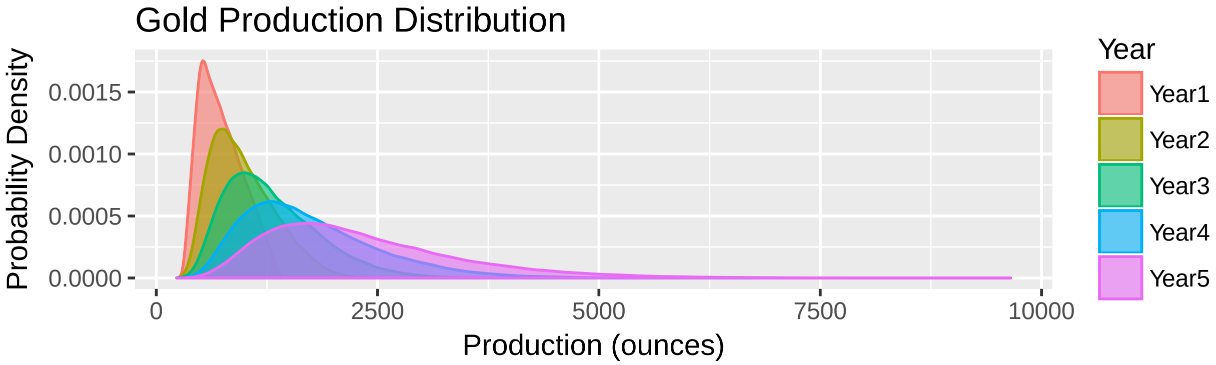
The distribution for Year 1 above is not an accurate depiction of the distribution, and is an artefact of the kernel estimator being used by the PDF function; a more accurate depiction of the PDF for Year 1 would be a single line at $1700.

Gold Production

As specified, the initial gold production in the first year is between 300 and 1400 ounces, with an expected value of 500 ounces. The production in each subsequent year is determined using the previous year's output and multiplying the output by a value with a range of 0.8 to 1.8 with an expected value of 1.4.

```
params$prod_init <- list(a = 300, b = 1400, c = 500)
params$prod_delta <- list(a = 0.8, b = 1.8, c = 1.4)
```

```
env1$gold_prod <- gen_gold_prod(rounds = config$rounds,
                                years = config$years,
                                prod_init = params$prod_init,
                                prod_delta = params$prod_delta)
```



The production of gold is trending up over time, with the length of the positive tail increasing each year. The uncertainty also increases over time.

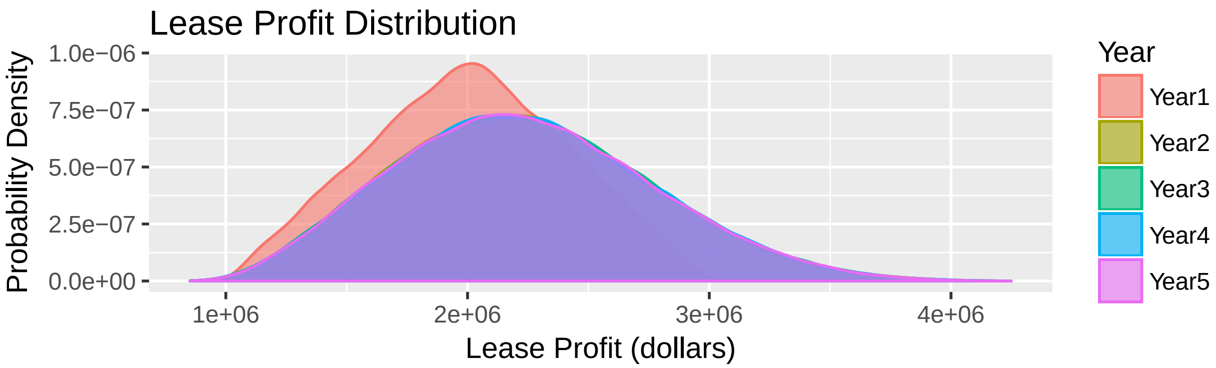
Leased Mine Income Share

The company owns a second mine that they lease to a third party. The contract is structured in such a way that the income received includes a profit share component. The range of income from leasing this second mine is between 1 million and 3 million dollars, with an expected value of 2 million dollars.

Each subsequent year the income from the lease is determined by modifying the amount of the lease that was received in the **first year**. The income from the mine is in the range of -20% (i.e. 0.8x multiplier) and 45% (i.e. 1.45x multiplier) with an expected value of 10% (i.e. 1.1x multiplier).

```
params$lease_profit_init <- list(a = 1000000, b = 3000000, c = 2000000)
params$lease_profit_delta <- list(a = 0.8, b = 1.45, c = 1.1)
```

```
env1$lease_profit <- gen_lease_profit(rounds = config$rounds,
                                      years = config$years,
                                      profit_init = params$lease_profit_init,
                                      profit_delta = params$lease_profit_delta)
```



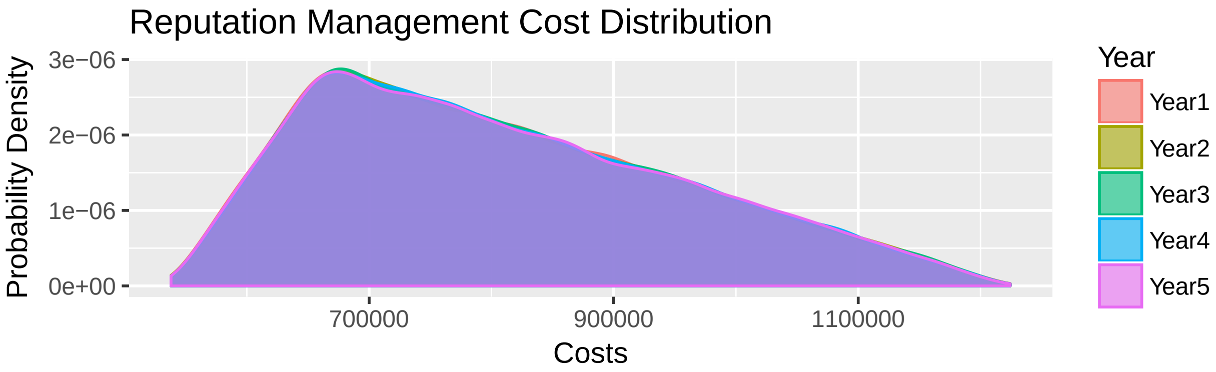
The probability distribution for each of the subsequent years is equal for all intents and purposes due to the fact that years 2 through 5 are based on year 1 rather than the previous year.

Reputation Management

The company incurs reputation management costs while they own the second (leased) mine. These costs have a fixed component ($350,000 per year whilst the leased mine is still in operation) and a variable component ($2,500 per day) with an estimated range of 75 to 350 days per year (expected value 125 days).

```
params$rep_mgmt_fixed <- 350000
params$rep_mgmt_rate <- 2500
params$rep_mgmt_var <- list(a = 75, b = 350, c = 125)
```

```
env1$rep_mgmt <- gen_rep_mgmt(rounds = config$rounds,
                          years = config$years,
                          fixed = params$rep_mgmt_fixed,
                          rate = params$rep_mgmt_rate,
                          hours = params$rep_mgmt_var)
```



The probability distribution for each year of reputation management is equal for all intents and purposes due to the fact that the assumptions are independent from year to year.

Mine Sale

The leased mine can be sold at the start of any year, and the price offered will be fixed as follows:

- Year 1 - $3,000,000
- Year 2 - $1,700,000
- Year 3 - $1,250,000
- Year 4 - $750,000

- Year 5 - $500,000

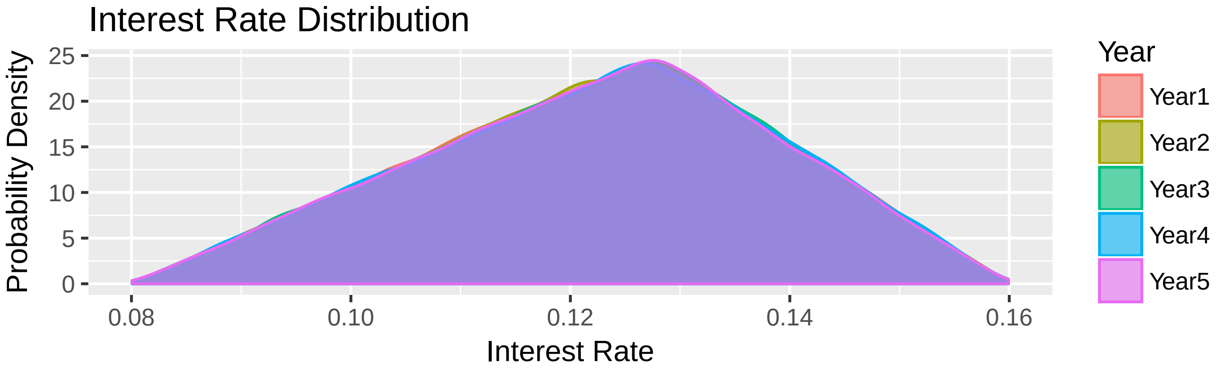
```
params$sale_price <- list(
  "Year1" = 3000000,
  "Year2" = 1700000,
  "Year3" = 1250000,
  "Year4" =  750000,
  "Year5" =  500000)
```

Interest Rate

The interest rate is variable, but fixed for each 12 month period. The interest rate is not known in advance, but is understood to be in the range of 8% to 16% with an expected value of 12.75%.

```
params$interest_rate_range <- list(a = 0.08, b = 0.16, c = 0.1275)
```

```
env1$interest_rate <- gen_independent(rounds = config$rounds,
                                      years = config$years,
                                      t = params$interest_rate_range)
```



The probability distribution for the interest rate for each year is equal for all intents and purposes due to the fact that the assumptions are independent from year to year.

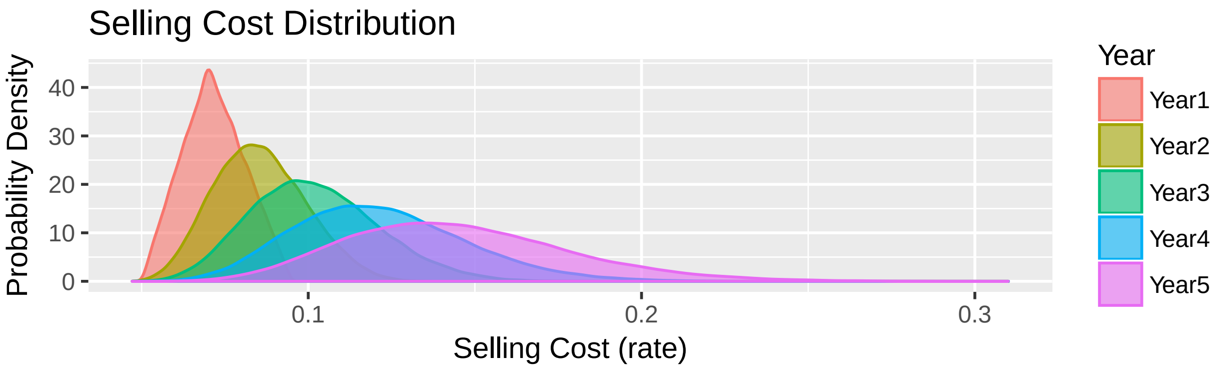
Selling Cost

There is a cost associated with selling the gold such as transportation, broker fees, taxes and other charges. This is simplified for this model by taking a percentage of the total amount of money generated from the ore. This amount is initially in the range of 5% to 9.5% with an expected value of 7%.

The selling cost in each subsequent year is determined by applying a modifier to the previous year's value. This modifier is a % change that is added to the previous year's cost using the formula $prev * (1 + mod)$. The selling cost modifier is in the range -0.1 to 0.45, with an expected value of 0.2275.

```
params$selling_cost_init <- list(a = 0.05, b = 0.095, c = 0.07)
params$selling_cost_delta <- list(a = -0.1, b = 0.45, c = 0.2275)
```

```
env1$selling_cost <- gen_selling_rate(rounds = config$rounds,
                                    years = config$years,
                                    cost_init = params$selling_cost_init,
                                    cost_delta = params$selling_cost_delta)
```



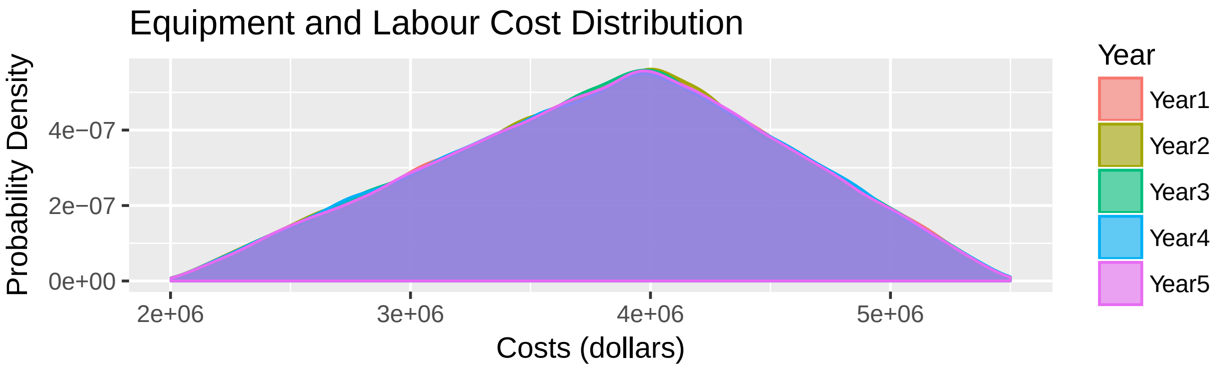
Selling costs are increasing over time, with increasing uncertainty.

Equipment and Labour Costs

The mine requires fixed plant and employees to run. To simplify the model, all these costs are rolled up into the range of $2,000,000 to $5,500,000 with an expected value of $4,000,000. As equipment can break down and even need replacing this value is calculated each year and is independent of the previous year.

```
params$equip_labour_range <- list(a = 2000000, b = 5500000, c = 4000000)
```

```
env1$equip_labour <- gen_independent(rounds = config$rounds,
                                     years = config$years,
                                     t = params$equip_labour_range)
```



The probability distribution for the equipment and labour costs for each year is equal for all intents and purposes due to the fact that the assumptions are independent from year to year.

Exploration

The majority of the exploration parameters are related to decision making rather than the environment, so they will be assessed in the next section of this report. As the second mine (if discovered) has a similar environment to the first mine, we can model it's environment here using the same approach as above.

```
# Start by copying the things that are equal for both mines
env2 <- env1[c("gold_price", "interest_rate")]

# Gold production in this mine is independent, but with the same parameters
env2$gold_prod <- gen_gold_prod(rounds = config$rounds,
                                years = config$years,
                                prod_init = params$prod_init,
                                        prod_delta = params$prod_delta)

# Selling cost in this mine is independent, but with the same parameters
env2$selling_cost <- gen_selling_rate(rounds = config$rounds,
                                      years = config$years,
                                      cost_init = params$selling_cost_init,
                                      cost_delta = params$selling_cost_delta)

# Equipment and labour costs are half that of the other mine, but not
# independent. This means that they will depend on the simulation for mine 1.
env2$equip_labour <- env1$equip_labour / 2
```

For these values, the only one that is time-dependent is the gold production value, which effectively has a "time-shift" component based on the year the mine is found. We can deal with this when we need to. All other values are considered to be "fixed", i.e. the costs for year 3 do not depend on the year in which the mine was discovered.

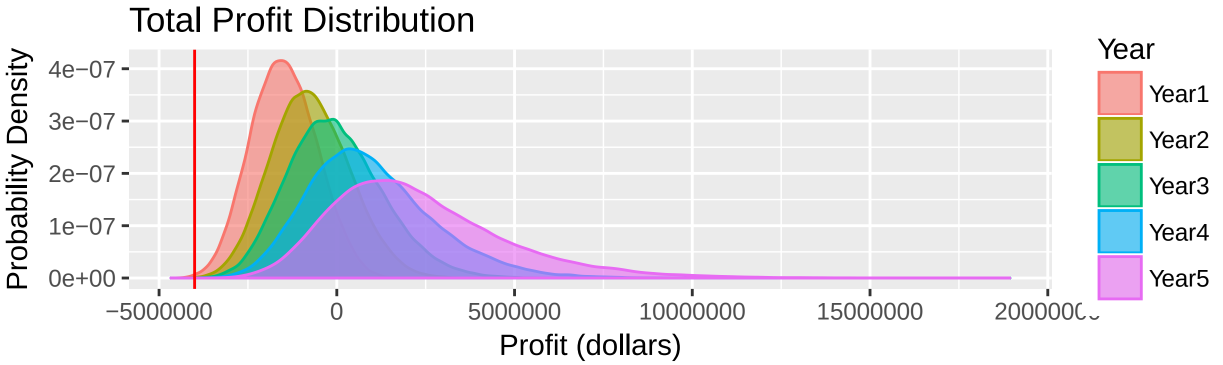
Simulation and Decision Making

With the environment modelled appropriately, we can now commence simulation and evaluation of possible decisions. Firstly, let's look at some base-case assumptions for a first pass simulation:

- No attempt to find the second mine
- No mine sale activity

```
profit_naive <- do_nothing_profits(env1)
```

Overall, how are the total profits in each year distributed?



Without taking any action, the mining operation becomes more profitable over time. Unfortunately there are 92 cases where the profits are so far in the red that the bank will foreclose on the mine - we will need to work out how to avoid these cases in future.

There are also 94,220 cases where the profit was negative, which is concerning but ultimately not a dealbreaker unless those cases don't provide a positive return over the 5 year estimates. Let's take a look at the 5 year estimates in detail:

```
##        Status
## Year    Foreclosure  Loss Profit
##   Year1          92 94128   5780
##   Year2       18805 71928   9267
##   Year3       33637 48267  18096
##   Year4       38100 30417  31483
##   Year5       39122 15298  45580
```

Reproducability

This analysis was performed using R, and the environment is described below to allow the analysis to be reproduced faithfully.

Session info

```
## Session info -------------------------------------------------------------
##  setting  value
##  version  R version 3.4.4 (2018-03-15)
##  system   x86_64, darwin15.6.0
##  ui       X11
##  language (EN)
##  collate  en_AU.UTF-8
##  tz       Australia/Sydney
##  date     2018-04-29
## Packages -----------------------------------------------------------------
##  package    * version    date       source
##  assertthat   0.2.0      2017-04-11 CRAN (R 3.4.0)
##  backports    1.1.0      2017-05-22 CRAN (R 3.4.0)
##  base       * 3.4.4      2018-03-15 local
##  bindr        0.1        2016-11-13 CRAN (R 3.4.0)
##  bindrcpp     0.2        2017-06-17 CRAN (R 3.4.0)
##  broom        0.4.2      2017-02-13 CRAN (R 3.4.0)
##  cellranger   1.1.0      2016-07-27 CRAN (R 3.4.0)
##  colorspace   1.3-2      2016-12-14 CRAN (R 3.4.0)
##  compiler     3.4.4      2018-03-15 local
##  datasets   * 3.4.4      2018-03-15 local
##  devtools     1.13.5     2018-02-18 CRAN (R 3.4.3)
##  digest       0.6.12     2017-01-27 CRAN (R 3.4.0)
##  dplyr      * 0.7.3      2017-09-09 CRAN (R 3.4.1)
##  evaluate     0.10.1     2017-06-24 CRAN (R 3.4.1)
##  forcats      0.2.0      2017-01-23 CRAN (R 3.4.0)
##  foreign      0.8-69     2017-06-22 CRAN (R 3.4.4)
##  ggplot2    * 2.2.1      2016-12-30 CRAN (R 3.4.0)
##  glue         1.1.1      2017-06-21 CRAN (R 3.4.1)
##  goldminer  * 0.0.0.9000 2018-04-29 local
##  graphics   * 3.4.4      2018-03-15 local
##  grDevices  * 3.4.4      2018-03-15 local
##  grid         3.4.4      2018-03-15 local
##  gtable       0.2.0      2016-02-26 CRAN (R 3.4.0)
##  haven        1.1.0      2017-07-09 CRAN (R 3.4.1)
##  hms          0.3        2016-11-22 CRAN (R 3.4.0)
##  htmltools    0.3.6      2017-04-28 CRAN (R 3.4.0)
##  httr         1.3.1      2017-08-20 CRAN (R 3.4.1)
##  jsonlite     1.5        2017-06-01 CRAN (R 3.4.0)
##  knitr        1.16       2017-05-18 CRAN (R 3.4.0)
##  labeling     0.3        2014-08-23 CRAN (R 3.4.0)
##  lattice      0.20-35    2017-03-25 CRAN (R 3.4.4)
##  lazyeval     0.2.0      2016-06-12 CRAN (R 3.4.0)
##  lubridate    1.6.0      2016-09-13 CRAN (R 3.4.0)
##  magrittr     1.5        2014-11-22 CRAN (R 3.4.0)
##  memoise      1.1.0      2017-04-21 CRAN (R 3.4.0)
##  methods    * 3.4.4      2018-03-15 local
##  mnormt       1.5-5      2016-10-15 CRAN (R 3.4.0)
##  modelr       0.1.1      2017-07-24 CRAN (R 3.4.1)
##  munsell      0.4.3      2016-02-13 CRAN (R 3.4.0)
##  nlme         3.1-131.1  2018-02-16 CRAN (R 3.4.4)
##  parallel     3.4.4      2018-03-15 local
##  pillar       1.1.0      2018-01-14 cran (@1.1.0)
##  pkgconfig    2.0.1      2017-03-21 CRAN (R 3.4.0)
##  plyr         1.8.4      2016-06-08 CRAN (R 3.4.0)
##  psych        1.7.5      2017-05-03 CRAN (R 3.4.1)
##  purrr      * 0.2.4      2017-10-18 cran (@0.2.4)
##  R6           2.2.2      2017-06-17 CRAN (R 3.4.0)
##  Rcpp         0.12.16    2018-03-13 CRAN (R 3.4.4)
##  readr      * 1.1.1      2017-05-16 CRAN (R 3.4.0)
##  readxl       1.0.0      2017-04-18 CRAN (R 3.4.0)
##  reshape2     1.4.2      2016-10-22 CRAN (R 3.4.0)
##  rlang        0.2.0      2018-02-20 cran (@0.2.0)
##  rmarkdown    1.6        2017-06-15 CRAN (R 3.4.0)
##  rprojroot    1.2        2017-01-16 CRAN (R 3.4.0)
##  rvest        0.3.2      2016-06-17 CRAN (R 3.4.0)
##  scales       0.4.1      2016-11-09 CRAN (R 3.4.0)
##  stats      * 3.4.4      2018-03-15 local
##  stringi      1.1.5      2017-04-07 CRAN (R 3.4.0)
##  stringr      1.2.0      2017-02-18 CRAN (R 3.4.0)
##  tibble     * 1.4.2      2018-01-22 cran (@1.4.2)
##  tidyr      * 0.7.1      2017-09-01 CRAN (R 3.4.1)
##  tidyverse  * 1.1.1      2017-01-27 CRAN (R 3.4.0)
##  tools        3.4.4      2018-03-15 local
##  utils      * 3.4.4      2018-03-15 local
##  withr        1.0.2      2016-06-20 CRAN (R 3.4.0)
##  xml2         1.1.1      2017-01-24 CRAN (R 3.4.0)
##  yaml         2.1.14     2016-11-12 CRAN (R 3.4.0)
```