Finding a suitable expression for the probability of failing a weight

2022-04-03

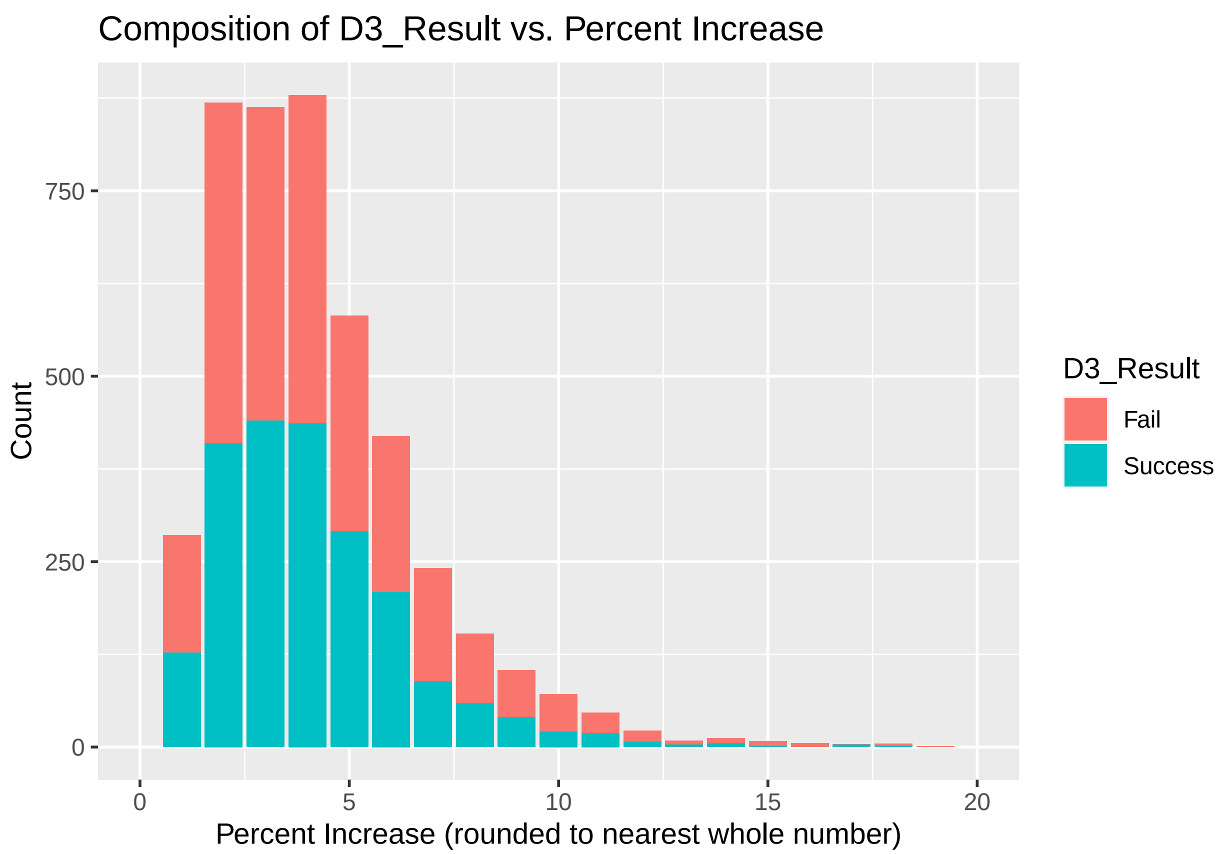
Reading in the data

```r
# read in a csv with file path "path", keep only necessary columns and rename
#   for readability, add column "Subtotal" which is equal to the sum of
#   BestSquat, BestBench, and max(0, Deadlift1, Deadlift2)
read_results <- function(path) {
  read_csv(path, col_names = TRUE) %>%
    select(Name, Sex, Division,
           WeightClass = "WeightClassKg", Bodyweight = "BodyweightKg",
           Squat1 = "Squat1Kg", Squat2 = "Squat2Kg", Squat3 = "Squat3Kg",
           BestSquat = "Best3SquatKg",
           Bench1 = "Bench1Kg", Bench2 = "Bench2Kg", Bench3 = "Bench3Kg",
           BestBench = "Best3BenchKg",
           Deadlift1 = "Deadlift1Kg", Deadlift2 = "Deadlift2Kg",
           Deadlift3 = "Deadlift3Kg", BestDeadlift = "Best3DeadliftKg",
           Total = "TotalKg", Place) %>%
    mutate(Subtotal = BestSquat + BestBench +
             ifelse(Deadlift1 > 0 | Deadlift2 > 0,
                    ifelse(Deadlift2 > Deadlift1, Deadlift2, Deadlift1),
                    0)) %>%
    relocate(Subtotal, .before = Deadlift3)
}

# read in data
data_21 <- read_results("2021.csv")
data_19 <- read_results("2019.csv")
data_18 <- read_results("2018.csv")
data_17 <- read_results("2017.csv")
data_16 <- read_results("2016.csv")
data_15 <- read_results("2015.csv")
data_14 <- read_results("2014.csv")
data_13 <- read_results("2013.csv")
data_12 <- read_results("2012.csv")

# merge all datasets so that we can run regressions with all of the data
  # note that when a weight n is attempted and failed, it is entered as -n.
  # that's why we calculate attempted lifts as the absolute value of the entry.
all_data <- rbind(data_21, data_19, data_18, data_17, data_16, data_15,
                  data_14, data_13, data_12) %>%
  select(Name, WeightClass, Bodyweight, Squat3, Deadlift2, Deadlift3) %>%
  mutate(AttSquat3 = abs(Squat3), # Att is short for attempted.
         AttDeadlift2 = abs(Deadlift2),
         AttDeadlift3 = abs(Deadlift3),
         Percent_Increase =
```

```
           round(((AttDeadlift3 - AttDeadlift2) / AttDeadlift2) * 100),
         D3_Success = ifelse(Deadlift3 > 0, 1, 0),
         D3_Result = ifelse(Deadlift3 > 0, "Success", "Fail")) %>%
  filter((!is.na(Percent_Increase)))
```

Composition of D3_Result vs. Percent Increase

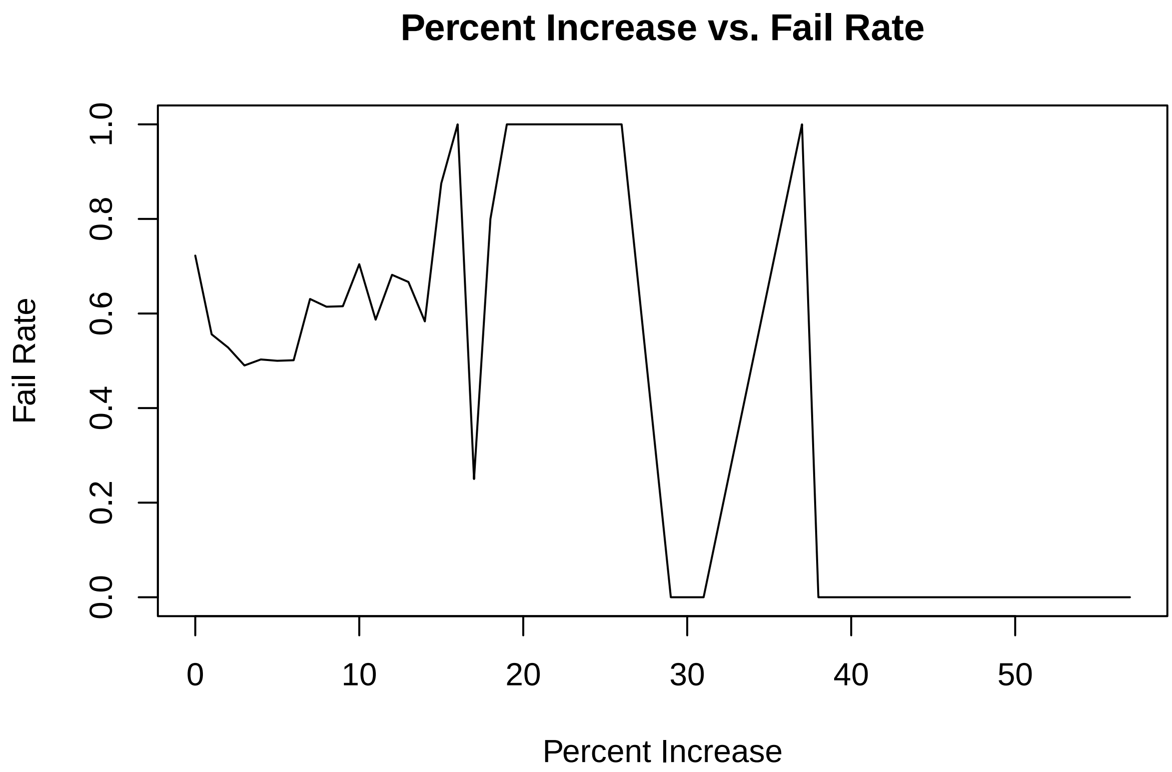
```
all_data %>%
  ggplot() +
  geom_bar(aes(x = Percent_Increase, fill = D3_Result)) +
  xlim(c(0, 20)) +
  xlab("Percent Increase (rounded to nearest whole number)") +
  ylab("Count") +
  labs(title = "Composition of D3_Result vs. Percent Increase")
```




```
composition <-
  all_data %>%
  group_by(Percent_Increase) %>%
  summarize(Success_Rate = mean(D3_Success),
            Fail_Rate = 1 - Success_Rate)

kable(composition)
```

Percent_Increase	Success_Rate	Fail_Rate
0	0.2774648	0.7225352
1	0.4440559	0.5559441
2	0.4718067	0.5281933
3	0.5098494	0.4901506
4	0.4971559	0.5028441
5	0.5000000	0.5000000
6	0.4988067	0.5011933
7	0.3692946	0.6307054
8	0.3856209	0.6143791
9	0.3846154	0.6153846
10	0.2957746	0.7042254
11	0.4130435	0.5869565
12	0.3181818	0.6818182
13	0.3333333	0.6666667
14	0.4166667	0.5833333
15	0.1250000	0.8750000
16	0.0000000	1.0000000
17	0.7500000	0.2500000
18	0.2000000	0.8000000
19	0.0000000	1.0000000
20	0.0000000	1.0000000
21	0.0000000	1.0000000
22	0.0000000	1.0000000
24	0.0000000	1.0000000
26	0.0000000	1.0000000
29	1.0000000	0.0000000
31	1.0000000	0.0000000
37	0.0000000	1.0000000
38	1.0000000	0.0000000
57	1.0000000	0.0000000

```r
plot(x = composition$Percent_Increase,
     y = composition$Fail_Rate,
     type = "l",
     xlab = "Percent Increase",
     ylab = "Fail Rate",
     main = "Percent Increase vs. Fail Rate")
```



Let's see if we can fit an existing CDF to this data, since we had problems with our made-up probability function. We'll try the CDF of a $\text{Exp}(\lambda)$ random variable which is

$$1 - e^{-\lambda x}.$$

But before we do that, to make this easier I will "cheat" a little bit by hiding cases where I believe the data is deceiving.

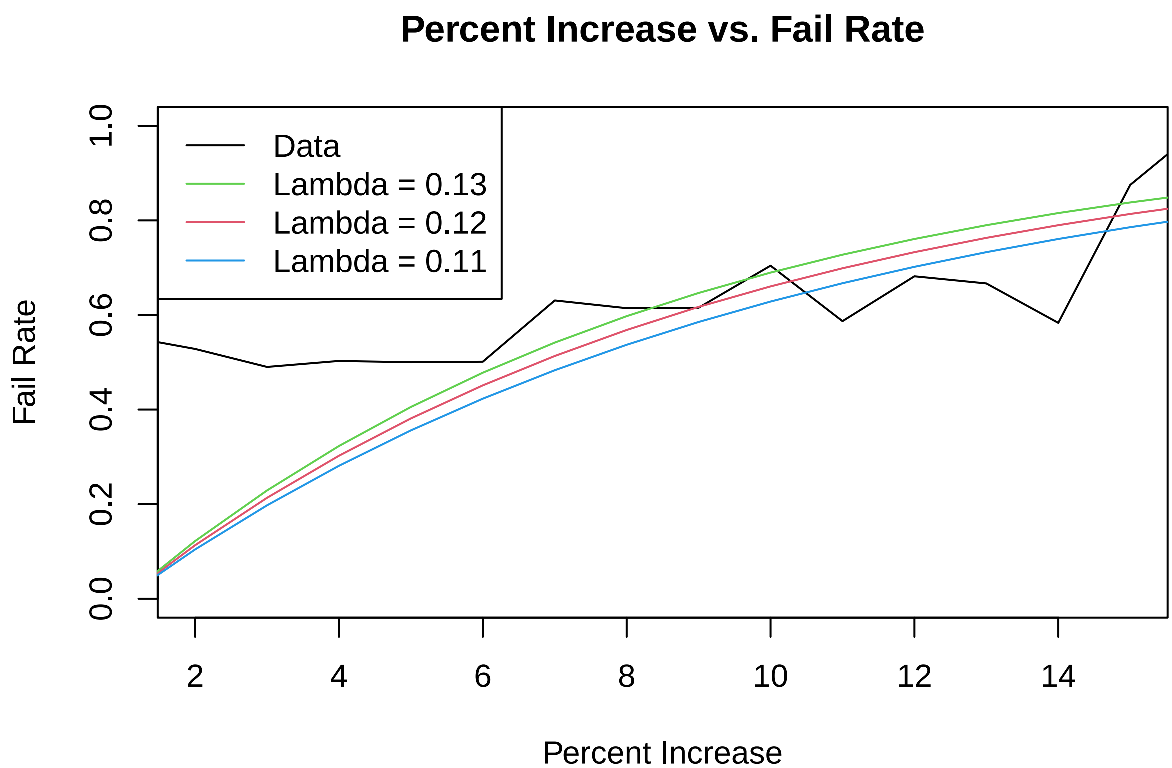
Specifically, when the percent increase is in the interval $[0, 2]$, that often happens because the competitor knows they can't do anything higher, they are tired, they are re-trying their second attempt which they were not strong enough to do, etc… this causes the failure rate to be inflated, in my opinion. After all, it should be safe to assume that *ceteris paribus*, increasing the weight increases risk of failure, right?

Secondly, for every percent increase that occurs that is greater than 15, there are at most 4 observations each. And since there are 5303 total observations, I don't think the sample can represent reality. They were most likely unique situations in which competitors were compelled to make bizarre attempt selections, so we shouldn't let these datapoints affect our CDF. Another reason why we won't lose much by hiding this interval is we simply don't care, since as we can see from the sample size, no one really ever jumps by more than 16 percent.

Let's plot a few CDF's with various values of λ and see which one fits the data the best.

```
lambda = 0.12

cdf <- function(lambda) {
  1 - exp(-1 * lambda * composition$Percent_Increase)
}
```

```
plot(x = composition$Percent_Increase,
     y = composition$Fail_Rate,
     type = "l",
     xlim = c(2, 15),
     xlab = "Percent Increase",
     ylab = "Fail Rate",
     main = "Percent Increase vs. Fail Rate")
lines(cdf(lambda), col = 2)
lines(cdf(lambda + 0.01), col = 3)
lines(cdf(lambda - 0.01), col = 4)
legend("topleft",
       col = c(1, 3, 2, 4),
       lwd = 1,
       legend = c("Data", "Lambda = 0.13", "Lambda = 0.12", "Lambda = 0.11"))
```




```
kable(data.frame(
  MSE_lambda_0.12 = mean((composition$Fail_Rate[3:16] - cdf(lambda)[3:16])^2),
  MSE_lambda_0.13 = mean((composition$Fail_Rate[3:16] -
                            cdf(lambda + 0.01)[3:16])^2),
  MSE_lambda_0.11 = mean((composition$Fail_Rate[3:16] -
                            cdf(lambda - 0.01)[3:16])^2)))
```

MSE_lambda_0.12	MSE_lambda_0.13	MSE_lambda_0.11
0.0182428	0.0190314	0.0187239

We see that the mean squared error is minimized when $\lambda = 0.12$, so I suggest the CDF

$$P(Fail|PercentIncrease = x) = 1 - e^{-0.12x}.$$