Inventing Life Insurance Experience Data

Philip Adams

Dabbling in the Dark Arts for Clicks

Knowing how to create fake believable data is at once valuable and suspicious. On the one hand, it shows a deep understanding of the processes and nuances of the data you are simulating. On the other hand, it shows an understanding for evil. For my own sake, I would like to think that I am too honest for the latter. If I ever were to fake results for a real situation, it probably means that I am suffering mental illness and need to be committed.

But, life insurance datasets are impossible to find. So I will make one up. Turns out, it isn't easy getting something believable off the ground. Thus, the risk of inspiring another Billion Dollar Bubble movie is low.

Approach

Here I use R. I will generate a census of lives including deaths and lapse information. Many of the features of this data are based on my experiences working with life insurance data over the years.

I have done this in Python for the sake of my own education and to compare methods. I discuss the issues of using Python in my blog.

Preparation

```
library(data.table)
library(data.table)
library(plyr)
library(tidyverse)
library(arrow)
library(parallel)
```

```r
library(rvinecopulib)
library(patchwork)
library(GGally)

RNGkind("L'Ecuyer-CMRG")



sILECPath <- "/workspace/Projects/ILEC/VBT/Data/ilecdata_20240119"
nPolicyCensusSize <- 2000000
arrIssueYearRange <- 2041:2054

set.seed(0xBEEF)

# Low Face is 90% of these, Males are 120% of these
fLowFaceFactorTrue <- 0.9
fMaleFactorTrue <- 1.2
```

Building a Source Distribution

I build a source distribution based on the ILEC data from 2011-2017. This uses the distribution of risks for term business in duration 1 issued under a four-class preferred system, by issue age, sex, and face amount. I weight by policies exposed. To expand by issue year, I create a fake issue year table covering our futuristic issue year along with relative proportion by issue year. The relative proportion is used to weight the policies exposed from the prior step. This is meant to simulate sales growth.

```r
ilec_dataset <- arrow::open_dataset(
  sources=sILECPath,
  format="parquet"
)

# Extract the data
ilec_dataset %>%
  filter(Insurance_Plan == "Term" &
           Duration == 1 &
           Issue_Age >= 18 &
           Issue_Age <= 70 &
           Number_of_Pfd_Classes == "4" &
           Observation_Year >= 2011 &
```

```
            Observation_Year <= 2017) %>%
  group_by(Sex,Issue_Age,Face_Amount_Band) %>%
  summarize(Policies_Exposed=sum(Policies_Exposed)) %>%
  collect() %>%
  data.table() ->
  src_distribution

# Regroup the face amounts to low and high face
src_distribution %>%
  mutate(Face_Group=fct_collapse(Face_Amount_Band,
                                 Low_Face=c("01: 0 - 9,999",
                                            "02: 10,000 - 24,999",
                                            "03: 25,000 - 49,999",
                                            "04: 50,000 - 99,999",
                                            "05: 100,000 - 249,999",
                                            "06: 250,000 - 499,999"),
                                 other_level="High_Face")
  ) %>%
  group_by(Sex,Issue_Age,Face_Group) %>%
  summarize(Policies_Exposed=sum(Policies_Exposed)) %>%
  arrange(Sex,Issue_Age,Face_Group) %>%
  data.table() ->
  src_distribution

# Expand the issue year dimension and adjust the exposure per year
src_distribution %>%
  cross_join(
    data.table( Issue_Year=arrIssueYearRange,
                Proportion=seq(.8,1.3,length=length(arrIssueYearRange)))
  ) %>%
  mutate(Policies_Exposed=Policies_Exposed*Proportion) %>%
  select(-Proportion) ->
  src_distribution

head(src_distribution)
```

```
   Sex Issue_Age Face_Group Policies_Exposed Issue_Year
 <char>     <int>     <fctr>            <num>      <int>
1:      F        18   Low_Face         1662.965       2041
2:      F        18   Low_Face         1742.915       2042
3:      F        18   Low_Face         1822.865       2043
4:      F        18   Low_Face         1902.815       2044
```

```
5:      F        18   Low_Face         1982.766       2045
6:      F        18   Low_Face         2062.716       2046
```

Building a Policy Census

To sample a census, I randomly sample rows from the source distribution table. Then I construct random issue dates within a calendar year. While the day of the month is random, the month is weighted toward the end of the year and away from the beginning of the year. Many companies will create sales incentives that increase sales at the end of the year, often at the expense of sales in the first quarter.

```
# Sample the source distribution to build a policy listing
policy_pop <- src_distribution[
  sample.int(n=nrow(src_distribution),
             size=nPolicyCensusSize,
             replace=T,
             prob=src_distribution$Policies_Exposed),
  .(Sex,Issue_Age,Face_Group,Issue_Year)
]

# Set an ID
policy_pop[,PolID:=1:nrow(.SD)]

# Issue Date
# Create a table of months and days in a standard year, and then add a proportion
data.table(DayOfYear=1:365) %>%
  mutate(DateTemp = ymd(20101231) %m+% days(DayOfYear),
         Month=month(DateTemp),
         Day=day(DateTemp),
         Proportion=1/365) %>%
  select(Month,Day,Proportion) ->
  issue_days

issue_days[Month==1,Proportion:=Proportion*.5]
issue_days[Month==2,Proportion:=Proportion/.7]
issue_days[Month==11,Proportion:=Proportion/.7]
issue_days[Month==12,Proportion:=Proportion/.5]

# Sample a random day
policy_pop <- cbind(policy_pop,
                            issue_days[
```

```
                              sample.int(n=nrow(issue_days),
                                         size=nPolicyCensusSize,
                                         replace=T,
                                         prob=issue_days$Proportion),
                              .(Month,Day)
                            ]
)

# ... and create the issue date
policy_pop %>%
  mutate(Issue_Date=ymd(paste0(Issue_Year," ",Month," ",Day))) %>%
  select(-Month,-Day) ->
  policy_pop

# Face Amounts
# Low face is 100-450K spaced at 50K intervals
# High Face is 500K - 2M spaced at 100K intervals
policy_pop[Face_Group=="Low_Face",
           Face_Amount:=1000*sample(x=seq(100,450,50),
                                    size=nrow(.SD),
                                    replace=T)
]

policy_pop[Face_Group=="High_Face",
           Face_Amount:=1000*sample(x=seq(500,2000,100),
                                       size=nrow(.SD),
                                       replace=T)
]

# Set a premium mode based on reasonable real world proportions
policy_pop[,prem_mode:= sample(c("A","Q","M"),
                               size=nrow(.SD),
                               prob=c(.04,.01,.95),
                               replace=T)]

head(policy_pop)
```

```
      Sex Issue_Age Face_Group Issue_Year PolID Issue_Date Face_Amount
   <char>     <int>     <fctr>      <int> <int>     <Date>       <num>
1:      F        37   Low_Face       2054     1 2054-08-28      200000
2:      M        34   Low_Face       2049     2 2049-03-24      100000
3:      F        38   Low_Face       2052     3 2052-02-16      100000
```

```
4:      M        34  High_Face       2052     4 2052-09-22     1700000
5:      M        47  High_Face       2043     5 2043-10-18      900000
6:      M        62   Low_Face       2043     6 2043-11-14      300000
   prem_mode
      <char>
1:         M
2:         M
3:         M
4:         M
5:         M
6:         M
```

Simulate Preferred Attributes Using Vine Copulas

To simulate a believable distribution of preferred criteria, I use vine copulas. The associated blog post explains these at a high level. Here, we model Bot Mass Index (BMI), Oil Pressure, and Oil Viscosity using a copula where BMI is dependent on each of oil pressure and oil viscosity, and pressure and viscosity are dependent given BMI. The analogy to Body Mass Index, blood pressure, and cholesterol might be coincidental.

The dependence of BMI to the others is a BB7 copula with parameters 1.5 and 3. The dependence of oil pressure and viscosity given BMI is Gaussian with correlatio 0.4.

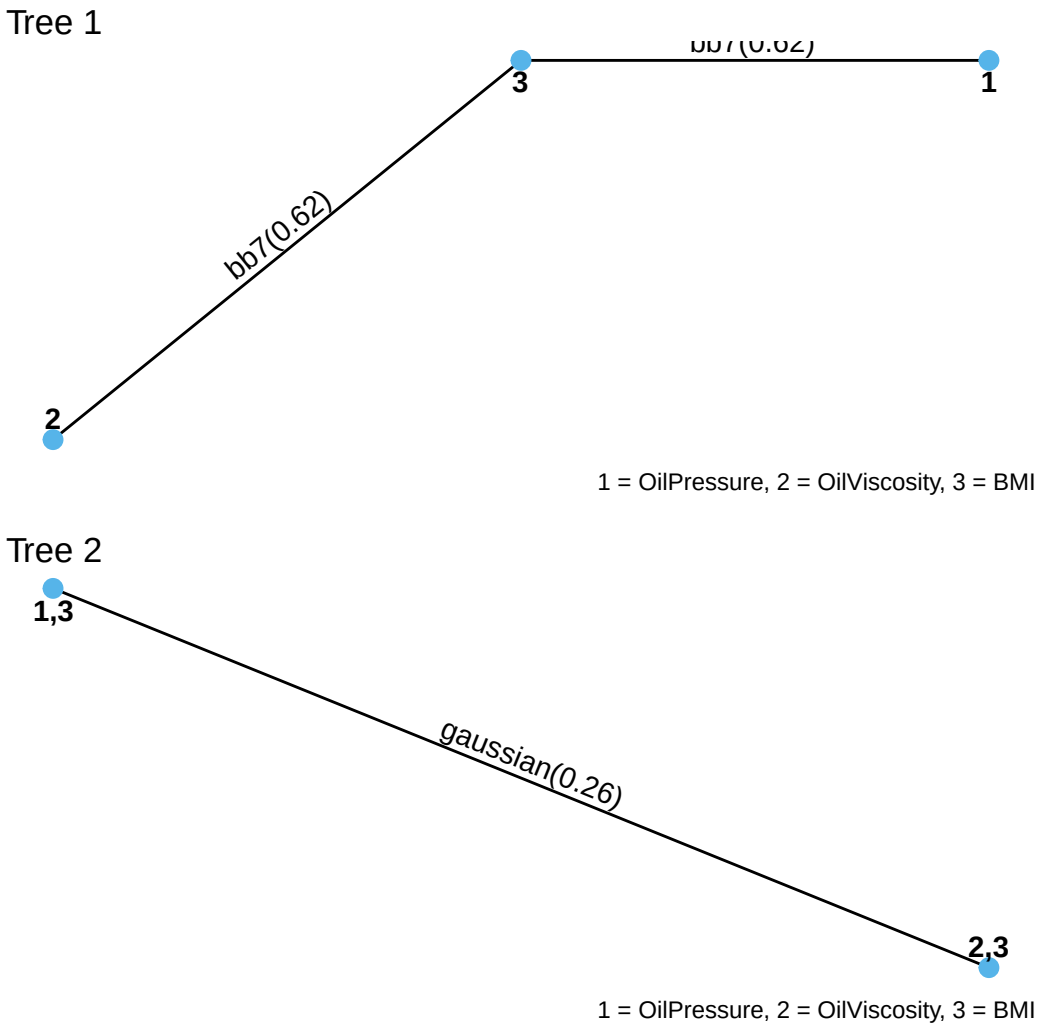
```
# Create synthetic preferred distributions
# BMI, oil pressure, and oil viscosity are modeled with a copula, which is
# necessarily a cvine (3 variables)
pref.vine <- vinecop_dist(
  pair_copulas = list(
    list(
      bicop_dist("bb7",180,c(1.5,3)),
      bicop_dist("bb7",180,c(1.5,3))
    ),
    list(bicop_dist("gaussian",0,.4))
  ),
  structure = cvine_structure(1:3)
)

pref.vine$names <- c("OilPressure","OilViscosity","BMI")

# Simulate the collectino of dependent uniform random variates
rvinecop(n=nPolicyCensusSize,
         pref.vine,
```

```
          cores=8) ->
  pref.samples

pref.samples %>%
  as.data.table() %>%
  rename(
    BMI_u=BMI,
    OilViscosity_u=OilViscosity,
    OilPressure_u=OilPressure
  ) -> pref.samples

policy_pop %>%
  cbind(pref.samples) ->
  policy_pop
```

It helps to visualize the vine and its distributions, and they can be plotted.

Dependency Structure

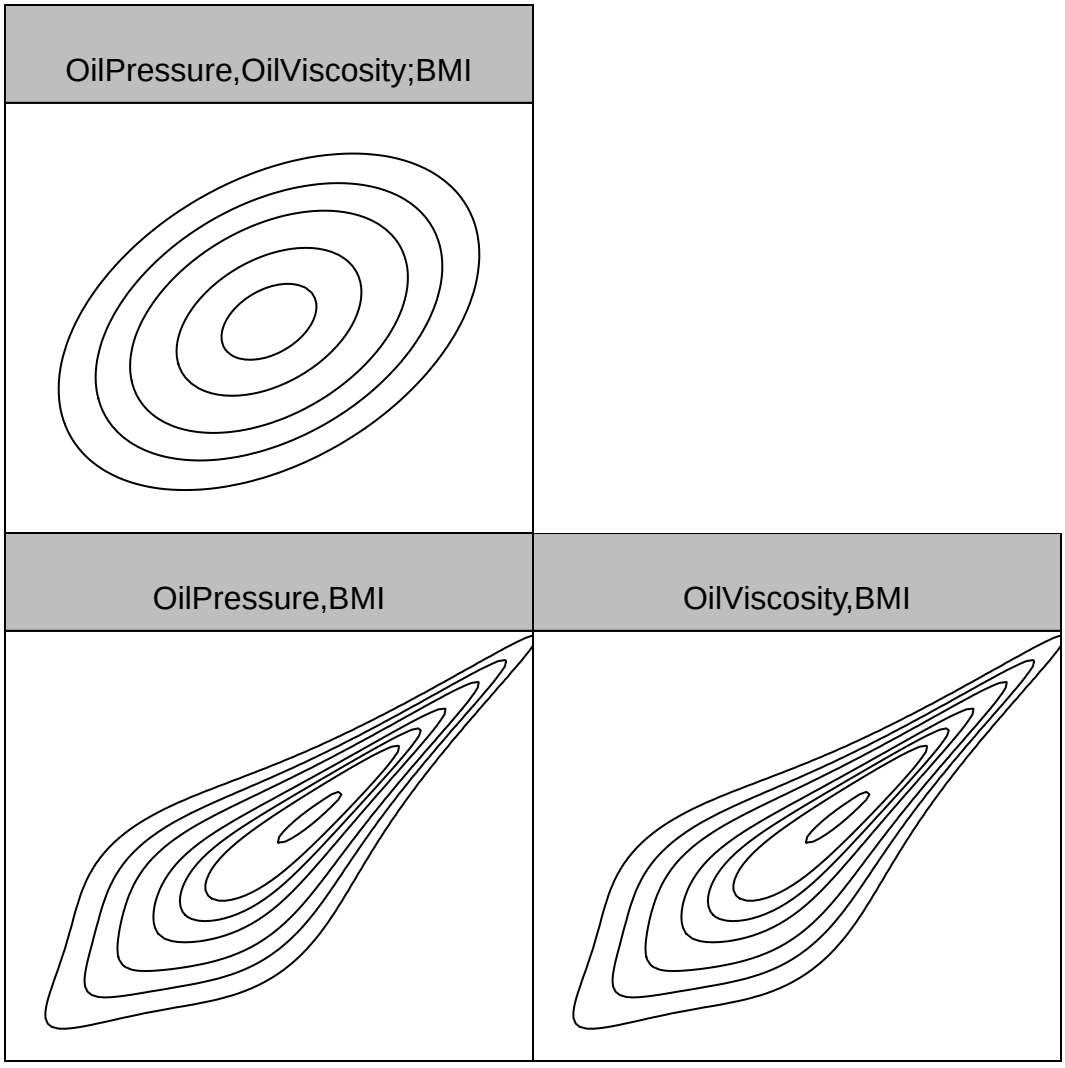
```
plot(pref.vine,tree=1:2, edge_labels = "family_tau",var_names = "legend")
```



Contour plots of bivariate copulas

It is easy to see here the strong upper tail dependence. The second level has some correlation, but the expectation is that most of the dependence will arise from BMI. Note that the Gaussian has no tail dependence. Also note that the contour plots assume normal margins.

```
contour(pref.vine)
```



Convert Simulated Uniform Marginals to Native Distribution

The simulator function for the vine copula generates values with uniform margins, so they are converted to the distributional scale we need.

- Each margin uses a gamma distribution.
- BMI for high face policies initially has mean 25 and variance 50.
- Oil pressure for high face policies initially has mean 30 and variance 20.
- Oil viscosity for high face policies initially has mean 20 and variance 20.
- Low face policies have mean 90% of high face, and males have mean 120% over females (at 100%).
- Starting in 2046, the means increase by 1% per issue year.

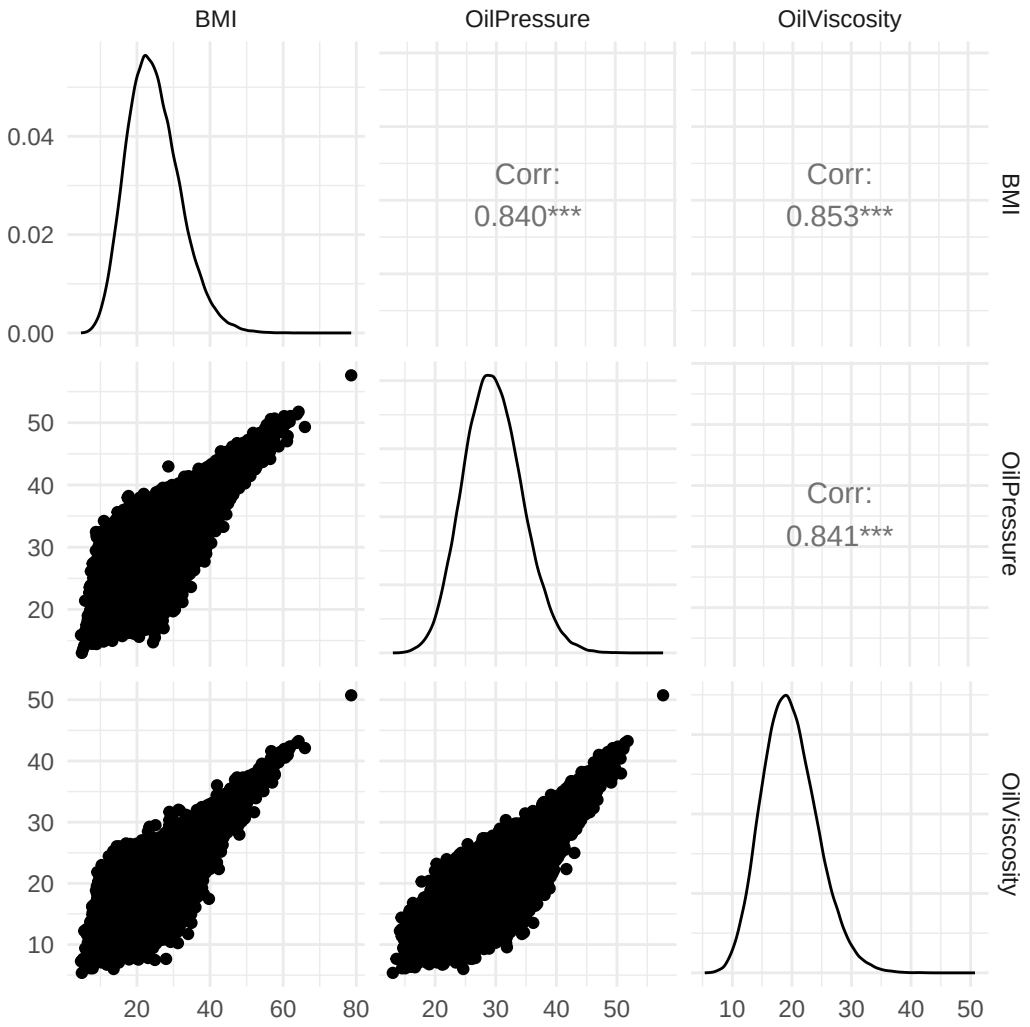
```r
# High Face
# BMI is gamma with mean 25 and variance 50
# Oil Pressure is gamma with mean 30 and variance 20
# Oil viscosity is gamma with mean 20 and variance 20

# Low Face is 90% of these, Males are 120% of these
# Starting in year 2046, apply 1% per issue year increase to all means
low_face_factor_true <- 0.9
male_factor_true <- 1.2

# Essentiall the inverse of the gamma CDF, but since we are specifying
# everything using the mean and variance, these have to be translated
# to shape and scale
convert_marginal_to_gamma <- function(x, mean=1, variance=1, rounding=2) {
  shape <- mean*mean/variance
  scale <- variance/mean

  round(qgamma(x,shape=shape,scale=scale),rounding)
}

policy_pop[,PrefMeanFactor:=ifelse(Sex=="Male",male_factor_true,1)*
             ifelse(Face_Group=="Low_Face",low_face_factor_true,1)*
             (1+pmax(0,Issue_Year-2045)/100)]

policy_pop[,
           `:=`(
             BMI=mapply(FUN=convert_marginal_to_gamma,
                        BMI_u,
                        25*PrefMeanFactor,
                        variance=50),
             OilPressure=mapply(FUN=convert_marginal_to_gamma,
                                OilPressure_u,
                                30*PrefMeanFactor,
                                variance=20),
             OilViscosity=mapply(FUN=convert_marginal_to_gamma,
                                 OilViscosity_u,
                                 20*PrefMeanFactor,
                                 variance=20)
           )]
```

```
policy_pop[,`:=`(OilViscosity_u=NULL,
                 OilPressure_u=NULL,
                 BMI_u=NULL,
                 PrefMeanFactor=NULL)]

head(policy_pop)
```

```
      Sex Issue_Age Face_Group Issue_Year PolID Issue_Date Face_Amount
   <char>     <int>     <fctr>      <int> <int>     <Date>       <num>
1:      F        37   Low_Face       2054     1 2054-08-28      200000
2:      M        34   Low_Face       2049     2 2049-03-24      100000
3:      F        38   Low_Face       2052     3 2052-02-16      100000
4:      M        34  High_Face       2052     4 2052-09-22     1700000
5:      M        47  High_Face       2043     5 2043-10-18      900000
6:      M        62   Low_Face       2043     6 2043-11-14      300000
   prem_mode   BMI OilPressure OilViscosity
      <char> <num>       <num>        <num>
1:         M 23.46       29.74        13.58
2:         M 23.71       26.15        16.28
3:         M 26.27       31.65        21.88
4:         M 31.88       35.67        23.90
5:         M 18.64       27.39        13.62
6:         M 13.64       25.06        15.13
```

A pairs plot shows off the result. There is quite a bit of correlation among criteria. In this case, it's linear correlation on the original scale.

```
ggpairs(policy_pop[sample.int(n=nPolicyCensusSize,size=100000,replace=T),
                   .(BMI,OilPressure,OilViscosity)]) +
  theme_minimal()
```



Assign Preferred Factors

Based on made-up parameterizations based on real phenomena, I modeled the relative mortality risk for each of the preferred variables as

- BMI follows a j-curve where every 5 points of BMI increases mortality by 25%
- Oil viscosity and oil pressure follow u-curves modeled as overlaid j-curves

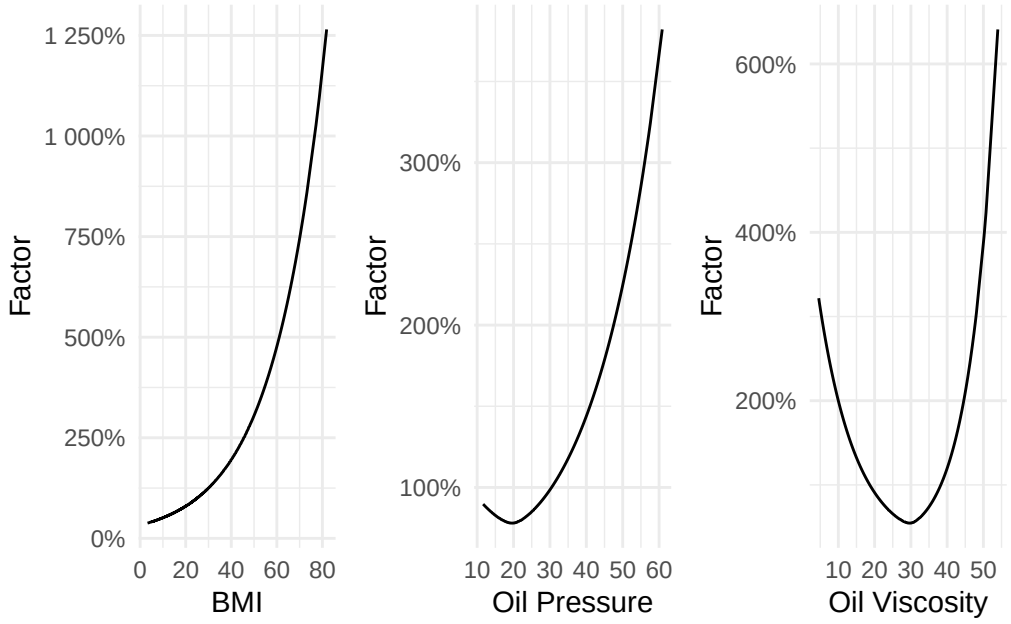
```
softplus <- function(x) {
  log(1+exp(x))
}
```

```
# Mortality factors for preferreds
policy_pop[,
           `:=`(PrefFactor_BMI=exp(log(1.25)*(BMI-25)/5),
                PrefFactor_Pressure=(exp(log(1.25)*
                                           softplus( .25*(OilPressure-20)/1 ))+
                                       exp(log(1.05)*
                                             softplus( -1*(OilPressure-20)/1 )))/
                  (1.05+1.25),
                PrefFactor_Viscosity=(exp(log(1.25)*
                                            softplus( 0.6*(OilViscosity-30)/1 ))+
                                        exp(log(1.05)*
                                              softplus( -2*(OilViscosity-30)/1 )))/
                  (1.05+1.25)
                )]

# The softplus unfortunately isn't automatically normalized
policy_pop[,
           `:=`(PrefFactor_Viscosity=PrefFactor_Viscosity/mean(PrefFactor_Viscosity),
                PrefFactor_Pressure=PrefFactor_Pressure/mean(PrefFactor_Pressure))]

policy_pop[,PrefFactor:=PrefFactor_BMI*PrefFactor_Pressure*PrefFactor_Viscosity]
```

Here are the curves for the relative risks for each of the preferred characteristics. The gamma yields a heavy tail. These handful of cases will be declined.

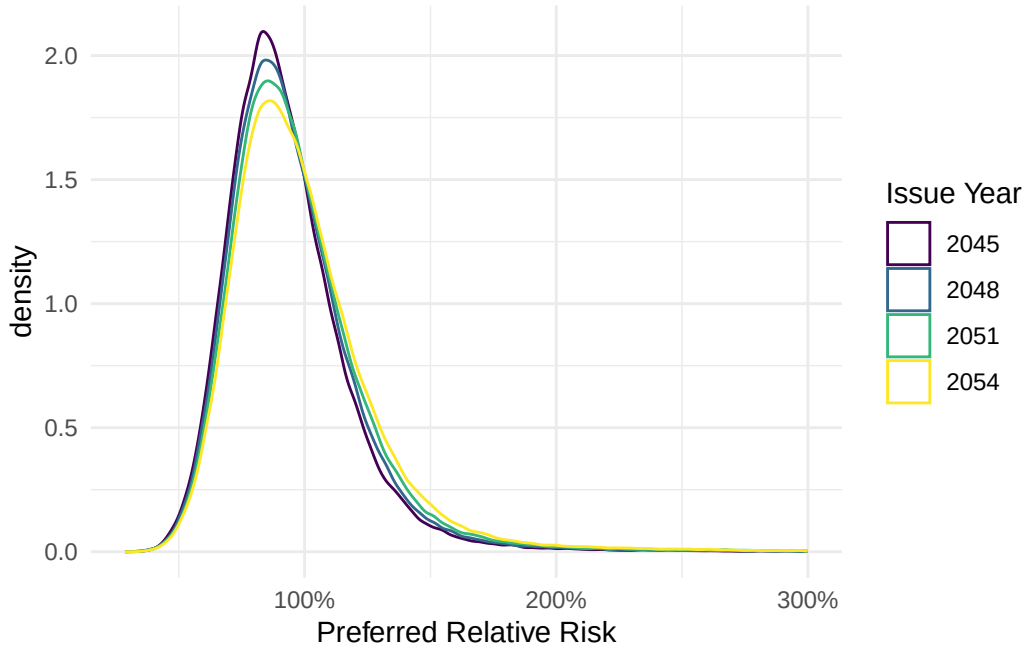
```
policy_pop %>%
  select(BMI, PrefFactor_BMI) %>%
  distinct() %>%
  ggplot(aes(x=BMI,y=PrefFactor_BMI)) +
  geom_line() +
  scale_color_viridis_d(name="Issue Year") +
  scale_x_continuous(name="BMI") +
  scale_y_continuous(name="Factor", labels=scales::percent) +
  theme_minimal() -> p1

policy_pop %>%
  select(OilPressure, PrefFactor_Pressure) %>%
  distinct() %>%
  ggplot(aes(x=OilPressure,y=PrefFactor_Pressure)) +
  geom_line() +
  scale_color_viridis_d(name="Issue Year") +
```

```
  scale_x_continuous(name="Oil Pressure") +
  scale_y_continuous(name="Factor", labels=scales::percent) +
  theme_minimal() -> p2


policy_pop %>%
  select(OilViscosity, PrefFactor_Viscosity) %>%
  distinct() %>%
  ggplot(aes(x=OilViscosity,y=PrefFactor_Viscosity)) +
  geom_line() +
  scale_color_viridis_d(name="Issue Year") +
  scale_x_continuous(name="Oil Viscosity") +
  scale_y_continuous(name="Factor", labels=scales::percent) +
  theme_minimal() -> p3

p1 + p2 + p3
```



As a result of the increasing mean values of preferred variables, the distribution of preferred relative risks increases and becomes more spread out over time.

```
policy_pop %>%
  filter(Issue_Year %in% c(2045,2048,2051,2054) & PrefFactor <= 3) %>%
  mutate(Issue_Year=factor(Issue_Year)) %>%
  ggplot(aes(x=PrefFactor,color=Issue_Year)) +
  geom_density(alpha=0.2) +
  scale_color_viridis_d(name="Issue Year") +
  scale_x_continuous(name="Preferred Relative Risk",labels=scales::percent) +
  theme_minimal()
```



Define Preferred Classes

Preferred classes are defined based on old style cutoffs. It is unlikely that that will still be used in the far future.

```
policy_pop[,
           UW_Decision:="SUB"]
policy_pop[BMI <= 41 & OilPressure <= 41 & OilViscosity <= 32,
           UW_Decision:="STD"]
policy_pop[UW_Decision == "SUB" & PrefFactor > 3,
           UW_Decision := "DEC"]
```

```
policy_pop[,PrefClass:=3]
policy_pop[BMI <= 30 & OilPressure <= 35 & OilViscosity <= 25 &
             UW_Decision == "STD",
           PrefClass:=1]

policy_pop[BMI <= 33 & OilPressure <= 38 & OilViscosity <= 28 & PrefClass > 1 &
             UW_Decision == "STD",
           PrefClass:=2]
```

Simulate Deaths and Lapses

Simulation relies heavily on multicore computation. Sixteen cores can get through the mortality simulation for 2,000,000 cases in approximately 10 minutes. Adjust your expectations accordingly if you have different resources.

```
source("LoadAssumptions.R")

policy_pop[,
           Death_Duration:=mcmapply(FUN=sample_death_duration,
                  Sex,
                  Issue_Age,
                  Issue_Date,
                  PrefFactor,
                  mc.cores = 16)]


policy_pop[,
           Lapse_Duration:=mcmapply(FUN=sample_lapse_duration,
                                    Face_Group,
                                    mc.cores = 16)]
```

Compute Dates and Status

Deaths do not occur uniformly in a given policy year. As time passes, risk of death increases. We simulated the policy year of death but not the day. We do so with a distribution of days tipped toward the later part of the year. This is designed so that the likelihood of death on day 365 is about 10% higher than on day 1.

Lapses occur uniformly through the year on a premium due date. This is not accurate in the real world, but it is enough for this case study.

```
death.skew <- 1:365
death.skew <- (1+death.skew*.1/364-.05-.1/364)/365

policy_pop[,death_skew_days:= sample.int(365,
                                         size=nrow(.SD),
                                         replace=T,
                                         prob = death.skew)]
policy_pop[,lapse_skew_months := sample.int(12,size=nrow(.SD),replace=T)]

policy_pop[,
           Death_Date:=Issue_Date %m+% years(Death_Duration-1) %m+%
             days(death_skew_days-1)]

policy_pop[prem_mode=="M",
           Lapse_Date:=Issue_Date %m+% years(Lapse_Duration-1) %m+%
             months(lapse_skew_months)]
policy_pop[prem_mode=="A",
           Lapse_Date:=Issue_Date %m+% years(Lapse_Duration-1)]
policy_pop[prem_mode=="Q",
           Lapse_Date:=Issue_Date %m+% years(Lapse_Duration-1) %m+%
             months(3*(lapse_skew_months %% 4 + 1 ) )]

policy_pop[,Status:="Active"]
policy_pop[Death_Date <= ymd(20531231) & Death_Date < Lapse_Date,
           `:=`(Status="Death",Term_Date=Death_Date)]
policy_pop[Lapse_Date <= ymd(20531231) & Lapse_Date < Death_Date,
           `:=`(Status="Lapsed",Term_Date=Lapse_Date)]

policy_pop[UW_Decision=="DEC",
           `:=`(
             Status="Not Issued",
             Term_Date=NA
           )]
```

Save the Work

```
arrow::write_parquet(x=policy_pop,
                     sink="policy_pop.parquet")
```