DGP Description

S. Vasserman

4/28/2017

There are N individuals, each of whom is sampled once.

Parameter DGP Specification

Each individual has:

Data: * Demographic characteristics X * Accident occurence: claim ~ Bernoulli(μ)

Parameters: * Bernoulli accident probability μ * Risk distortion parameter (factor by which the individual overestimates his risk) δ * Risk aversion parameter γ

A sample DGP for these parameters as follows:

$$\mu = \text{invlogit}(X\beta) \text{ where } \beta \sim \mathcal{N}(0,1)$$

$$\gamma \sim \mathcal{N}(\mu_\gamma, \sigma_\gamma) \text{ and } \gamma \in [0,\infty)$$
$$\mu_\gamma \sim \mathcal{N}^+(0,1) \text{ and } \sigma_\gamma \sim \mathcal{N}^+(0,2)$$

In simulation, I kept getting large values of μ_γ (not sure why), which made getting enough variance hard. I limited $\mu_\gamma \in [0,.5]$ as a hack to help with that.

To reduce the degrees of freedom, I've set $\delta = 10$ for everyone for now.

Identification

Identification is based off of *plan choice*. Plans y_j consist of tuples of (premium,deductible).

For development, I'm using these sample numbers (that have some basis in reality from a related paper):

	premium	Deductible
Low	1,324.71	374.92
High	1,093.51	1,124.75

Plans are evaluated based on expected utility with the following formulation:

$$EU(y_j) = (\mu \cdot \delta)u(-p_j - d_j) + (1 - \mu \cdot \delta)u(-p_j)$$

where p_j and d_j are the premium and deductible for plan y_j and

$$u(\Delta) = \Delta - \frac{\gamma}{2}(\Delta)^2.$$

Individuals are assumed to observe their expected utility for each plan, together with a Gumbel error, and choose the plan that maximizes utiltiy. Equivalently, individuals choose plan y_j according to a Categorical distribution with probability given by the softmax of each plan's expected utility:

$$\hat{y}_i \sim \text{Categorical}(\theta_{i,j})$$

$$\theta_{i,j} = Pr(\hat{y}_i = y_j) = \frac{exp(EU(y_j))}{\sum_k exp(EU(y_k))}$$

Normalizations

In order to identify the softmax probabilities without an outside good, I normalize the premiums and deductible with respect to those of the first plan:

$$p_j = \frac{p_j - p_1}{p_1} \text{ and } d_j = \frac{d_j - d_1}{d_1}$$

Data Simulation

The idea here is that lower gamma should map to a higher probability of choosing the high deductible plan. Given these premium/deductible numbers, I've found that I need a fairly wide distribution of gammas for this (e.g. low gamma needs to be fairly close to 0, high gamma needs to be closer to 1 at least, in order to produce enough variance). With a truncated normal, I can kind of get at this, but it's not super great.

```r
library(dplyr); library(ggplot2); library(ggthemes); library(fExtremes); library(rmutil)
library(scales);library(lme4); library(MASS);
source('fte_theme.R')
source('helperFunctions.R')

set.seed(424)

# N obs
num_inds <- 1000

#Group Categories
age_groups <- 0:1
male <- 0:1
home_owner <- 0:1

# Plan Definitions
p = c(1324.71,1093.51)
Z = c(374.92,1124.75)

p = (p - p[1])/p[1]
Z = (Z - Z[1])/Z[1]

## Draw random population sample
pop_indices <- expand.grid(age = 0:1, male = 0:1, home_owner = 0:1)

sample_panel <- pop_indices %>%
  sample_n(size = num_inds, replace=TRUE) %>%
  mutate(
    individual_id = row_number()
  )

### Convert to matrices for Stan ###

## Characteristics (factors)
X = as.matrix(data.frame(sample_panel$age, sample_panel$male, sample_panel$home_owner))
## Claim outcomes
```

```r
claim <- sample_panel$outcome

## matrix sizes
N <- nrow(X)
M <- ncol(X)
J <- length(p)

## plan characteristics
rep.row<-function(x,n){
  matrix(rep(x,each=n),nrow=n)
}

premiums <- rep.row(p,N)
deductibles <- rep.row(Z,N)

sample_panel$delta <- 12

# ## Put data together for stan
raw_data_list <- list(N = N,
                  M = M,
                  J = J,
                  X = X,
                  premiums = premiums,
                  deductibles = deductibles,
                  delta = as.vector(sample_panel$delta)
)

library(rstan)
options(mc.cores = parallel::detectCores())
dgp_model <- stan_model("gamma_nodelta_truncnorm_DGP.stan")
```

```
## In file included from file3853717436fb.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/BH/include/boos
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/BH/include/boos
## /Library/Frameworks/R.framework/Versions/3.3/Resources/library/BH/include/boost/config/compiler/clang
## #  define BOOST_NO_CXX11_RVALUE_REFERENCES
##          ^
## <command line>:6:9: note: previous definition is here
## #define BOOST_NO_CXX11_RVALUE_REFERENCES 1
##         ^
## In file included from file3853717436fb.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/include/stan/math/rev/core
##     static void set_zero_all_adjoints() {
##                 ^
## In file included from file3853717436fb.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/include/stan/math/rev/core
##     static void set_zero_all_adjoints_nested() {
##                 ^
## In file included from file3853717436fb.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/include/stan/math/prim/mat
##       size_t fft_next_good_size(size_t N) {
##              ^
## In file included from file3853717436fb.cpp:8:
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/StanHeaders/incl
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/BH/include/boost
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/BH/include/boost
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/BH/include/boost
## In file included from /Library/Frameworks/R.framework/Versions/3.3/Resources/library/BH/include/boost
## /Library/Frameworks/R.framework/Versions/3.3/Resources/library/BH/include/boost/multi_array/concept_c
##       typedef typename Array::index_range index_range;
##                                           ^
## /Library/Frameworks/R.framework/Versions/3.3/Resources/library/BH/include/boost/multi_array/concept_c
##       typedef typename Array::index index;
##                                     ^
## /Library/Frameworks/R.framework/Versions/3.3/Resources/library/BH/include/boost/multi_array/concept_c
##       typedef typename Array::index_range index_range;
##                                           ^
## /Library/Frameworks/R.framework/Versions/3.3/Resources/library/BH/include/boost/multi_array/concept_c
##       typedef typename Array::index index;
##                                     ^
## 8 warnings generated.
```

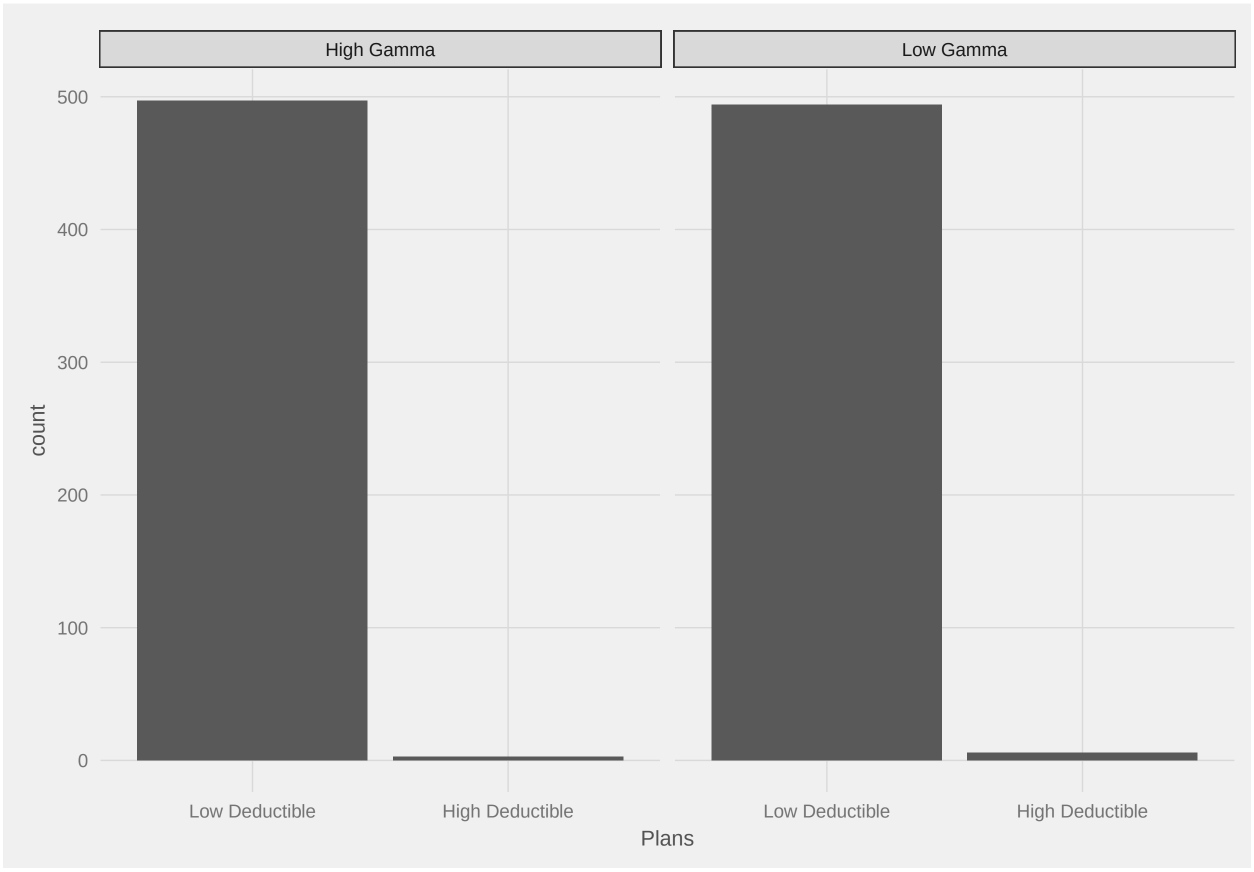
```r
dgp_test <- sampling(dgp_model, data = raw_data_list, iter = 24, chains = 1)
```

```
##
## SAMPLING FOR MODEL 'gamma_nodelta_truncnorm_DGP' NOW (CHAIN 1).
##
## Rejecting initial value:
##   Error evaluating the log probability at the initial value.
##
## Rejecting initial value:
##   Error evaluating the log probability at the initial value.
##
## Rejecting initial value:
##   Error evaluating the log probability at the initial value.
##
```

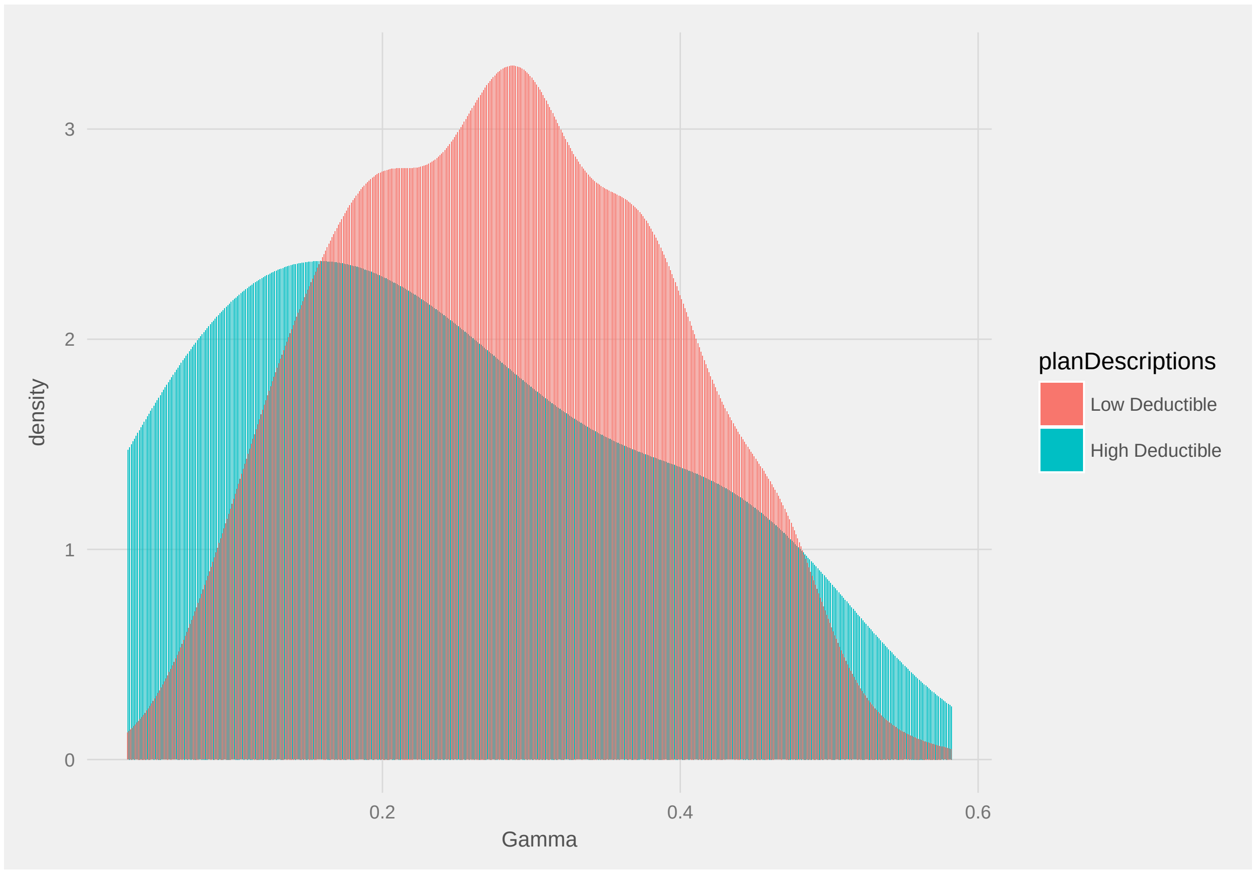
```
## Rejecting initial value:
##   Error evaluating the log probability at the initial value.
##
## Rejecting initial value:
##   Error evaluating the log probability at the initial value.
##
## Rejecting initial value:
##   Error evaluating the log probability at the initial value.
##
## Rejecting initial value:
##   Error evaluating the log probability at the initial value.
##
## Rejecting initial value:
##   Error evaluating the log probability at the initial value.
##
## Gradient evaluation took 0.002015 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 20.15 seconds.
## Adjust your expectations accordingly!
##
##
## WARNING: No variance estimation is
##          performed for num_warmup < 20
##
## Iteration:  1 / 24 [  4%]  (Warmup)
## Iteration:  2 / 24 [  8%]  (Warmup)
## Iteration:  4 / 24 [ 16%]  (Warmup)
## Iteration:  6 / 24 [ 25%]  (Warmup)
## Iteration:  8 / 24 [ 33%]  (Warmup)
## Iteration: 10 / 24 [ 41%]  (Warmup)
## Iteration: 12 / 24 [ 50%]  (Warmup)
## Iteration: 13 / 24 [ 54%]  (Sampling)
## Iteration: 14 / 24 [ 58%]  (Sampling)
## Iteration: 16 / 24 [ 66%]  (Sampling)
## Iteration: 18 / 24 [ 75%]  (Sampling)
## Iteration: 20 / 24 [ 83%]  (Sampling)
## Iteration: 22 / 24 [ 91%]  (Sampling)
## Iteration: 24 / 24 [100%]  (Sampling)
##
##  Elapsed Time: 0.111372 seconds (Warm-up)
##                0.124911 seconds (Sampling)
##                0.236283 seconds (Total)
```

```r
dgp_claim_samples <- as.data.frame(dgp_test,pars = "claim_sim")
dgp_claim_samples <- t(dgp_claim_samples[1,])

dgp_mu_gamma_samples <- as.data.frame(dgp_test,pars = "mu_gamma")
dgp_sigma_gamma_samples <- as.data.frame(dgp_test,pars = "sigma_gamma")

dgp_mu_gamma_samples <- t(dgp_mu_gamma_samples[1,])
dgp_sigma_gamma_samples <- t(dgp_sigma_gamma_samples[1,])

dgp_planChoice_samples <- as.data.frame(dgp_test,pars = "plan_choice_sim")
dgp_planChoice_samples <- t(dgp_planChoice_samples[1,])
```

```r
dgpData <- data.frame(planChoice = dgp_planChoice_samples[,1],  claims = dgp_claim_samples[,1])

plan_labels <- c("Low Deductible","High Deductible")
plan_levels <- c(1,2)
dgpData$planDescriptions <- factor(dgpData$planChoice,
                                              levels = plan_levels,
                                              labels = plan_labels)

gamma_sims <- get_posterior_mean(dgp_test, pars = "gamma")
med_gamma_est <- median(gamma_sims[,1])
dgpData$gamma <- gamma_sims[,1]
dgpData$gamma_group <- ifelse(dgpData$gamma < med_gamma_est,"Low Gamma","High Gamma")
dgpData$planDescriptions <- dgpData$planDescriptions


ggplot(data=dgpData, aes(x=planDescriptions)) +
  geom_bar(position="dodge") +
  guides(fill=guide_legend(title="Effort Cost Type")) +
  facet_grid(~gamma_group) +
  scale_fill_manual(values = gamma_group) +
  scale_x_discrete(name ="Plans") +
  fte_theme()
```



```r
ggplot(data=dgpData, aes(x=gamma, fill=planDescriptions)) +
  geom_histogram(position="dodge", stat="density")  +
  scale_x_continuous(name ="Gamma") +
fte_theme()
```



When I try to back out my parameters, I'm able to get something reasonable-ish for the hyper parameters, but have not had much luck in estimating gammas for individuals. It seems to me that this is an issue with choosing the right prior. When I simulate this with two types: $\gamma_L = 0.0001$ and $\gamma_H = 0.95$, I get a much cleaner sorting of low gamma types into high deductible plans. Similarly, when I simulated data with a beta distribution, I got a cleaner distribution (happy to show you if you're interested), but estimating individual gammas was still quite bad.