Logistic Regression Take 01

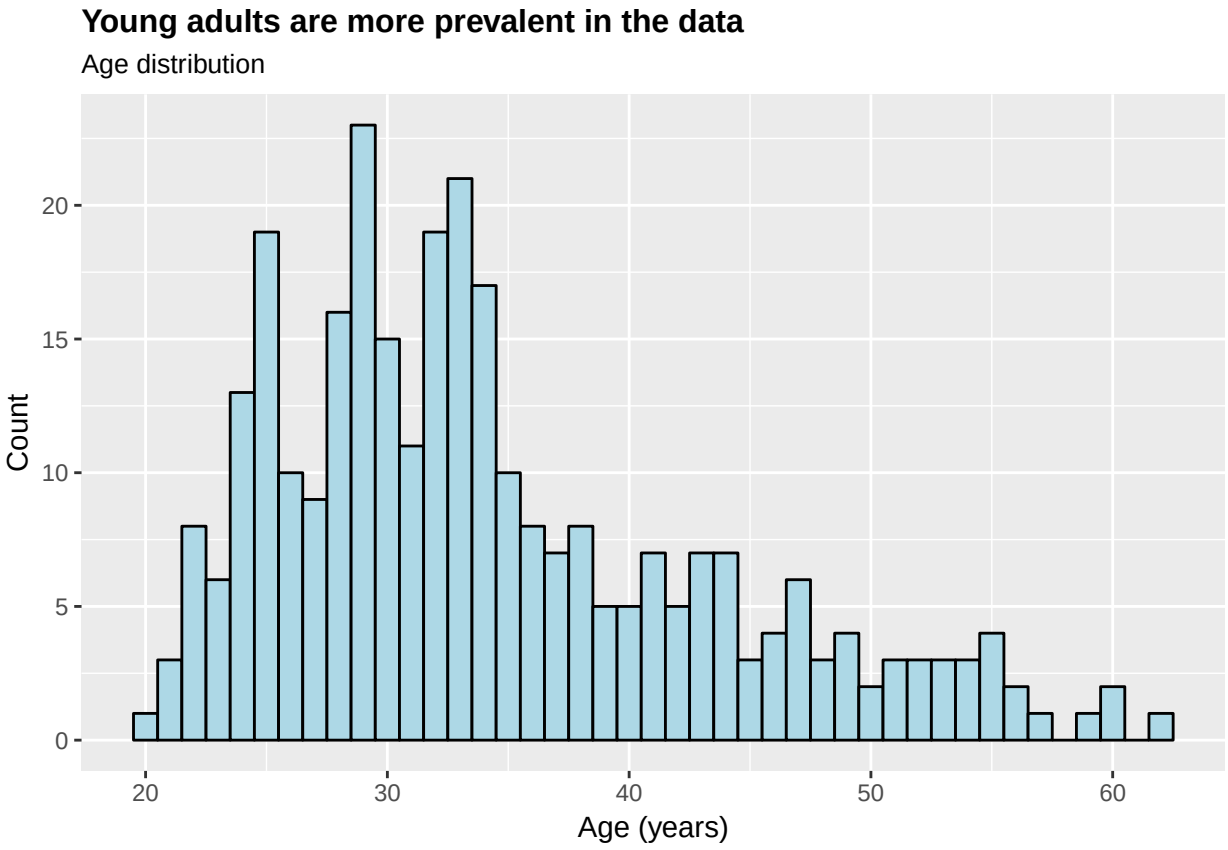
Andres Potapczynski (ap3635)

11/9/2018

Results from first logistic regression

Plot variable distributions

Now we have the following age distribution



Also, the distribution of the number of elements per zip code is the following

```
zip_summary = data %>%
  group_by(postal_code) %>%
  summarize(mort_no = n()) %>%
  arrange(desc(mort_no))

reversal = zip_summary %>%
  group_by(mort_no) %>%
  summarize(count = n())
ggplot(data = reversal, mapping = aes(x = count)) +
  geom_histogram(binwidth = 10, fill='lightblue', color='black') +
  xlab('No of mortages') +
  ylab('Count') +
```

```
  xlim(0, 100) +
  labs(title='...',
       subtitle = 'Number of mortgages per zip code') +
  theme(plot.title = element_text(size = 12, face='bold')) +
  theme(plot.subtitle = element_text(size = 10))
```

Run model

```
y = data$y
data_sub <- data %>% select(client_income,
                        ratio,
                        age,
                        asset_market_value,
                        lender_score,
                        factor_employed,
                        risk_index)
summary(data_sub)
```

```
##  client_income        ratio              age        asset_market_value
##  Min.   : 246.5   Min.   :0.0720   Min.   :20.00   Min.   : 261000
##  1st Qu.: 282.5   1st Qu.:0.2600   1st Qu.:28.00   1st Qu.: 386000
##  Median : 309.6   Median :0.2649   Median :32.00   Median : 432000
##  Mean   : 417.6   Mean   :0.2689   Mean   :34.34   Mean   : 551544
##  3rd Qu.: 342.1   3rd Qu.:0.2797   3rd Qu.:39.00   3rd Qu.: 528000
##  Max.   :1812.7   Max.   :0.3085   Max.   :62.00   Max.   :3441200
##   lender_score   factor_employed    risk_index
##  Min.   :  0.0   Min.   : 0.885   Min.   :2040
##  1st Qu.:130.0   1st Qu.: 1.107   1st Qu.:2183
##  Median :137.0   Median : 1.193   Median :2209
##  Mean   :134.3   Mean   : 1.977   Mean   :2209
##  3rd Qu.:140.0   3rd Qu.: 1.336   3rd Qu.:2238
##  Max.   :153.0   Max.   :68.787   Max.   :2326
```

Evaluate the model

I compile the model to access it later and then run it in a separate code chunk.

```
sm <- stan_model('./logistic_reg_v01.stan')
```

```
## recompiling to avoid crashing R session
```

```
### Debug this function
single_analysis <- function(var, data, y, sm){
  input_data = data %>% select(var)
  inputs = list(N = nrow(input_data), D=ncol(input_data), X=input_data, y=y)
  model.fit = sampling(sm, data=inputs, verbose=F)
  return(model.fit)
}
```

For age only

```
input_data = data_sub %>% select(age)
inputs = list(N = nrow(input_data), D=ncol(input_data), X=input_data, y=y)
```

The results are the following

```
print(model_v01, digits = 2)
```

```
## Inference for Stan model: logistic_reg_v01.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## alpha    -3.17    0.03 0.93  -5.00  -3.79  -3.16  -2.55  -1.33   829    1
## beta[1]   0.01    0.00 0.03  -0.04  -0.01   0.01   0.03   0.06   854    1
## lp__    -72.17    0.03 0.99 -74.78 -72.59 -71.87 -71.44 -71.18  1121    1
##
## Samples were drawn using NUTS(diag_e) at Fri Nov  9 20:41:20 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

For risk index only

```
input_data = data_sub %>% select(risk_index)
inputs = list(N = nrow(input_data), D=ncol(input_data), X=input_data, y=y)
```

The results are the following

```
print(model_v01, digits = 2)
```

```
## Inference for Stan model: logistic_reg_v01.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## alpha     2.80    0.20 4.61  -6.16  -0.37   2.83   6.08  11.45   534 1.01
## beta[1]   0.00    0.00 0.00  -0.01   0.00   0.00   0.00   0.00   534 1.01
## lp__    -71.64    0.04 1.01 -74.45 -72.04 -71.32 -70.93 -70.66   807 1.00
##
## Samples were drawn using NUTS(diag_e) at Fri Nov  9 20:41:24 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

For asset market value only

```
input_data = data_sub %>% select(asset_market_value)
inputs = list(N = nrow(input_data), D=ncol(input_data), X=input_data, y=y)
```

```
## Warning: There were 2355 transitions after warmup that exceeded the maximum treedepth. Increase max_
## http://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

The results are the following

```
print(model_v01, digits = 2)
```

```
## Inference for Stan model: logistic_reg_v01.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff
## alpha    -1.03    0.55 0.78  -1.81  -1.48  -1.29  -0.86   0.28     2
## beta[1]   0.00    0.00 0.00   0.00   0.00   0.00   0.00   0.00     2
## lp__    -70.16    0.49 0.93 -72.31 -70.90 -69.78 -69.39 -69.25     4
##           Rhat
## alpha   243.63
## beta[1]   3.74
## lp__      1.49
##
## Samples were drawn using NUTS(diag_e) at Fri Nov  9 20:41:52 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

were we see that there was a lot of problem with convergence in this case.

For client income

```
input_data = data_sub %>% select(client_income)
inputs = list(N = nrow(input_data), D=ncol(input_data), X=input_data, y=y)
```

The results are the following

```
print(model_v01, digits = 2)
```

```
## Inference for Stan model: logistic_reg_v01.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## alpha    -0.95    0.05 1.08  -2.57  -1.73  -1.10  -0.33   1.50   462 1.01
## beta[1]  -0.01    0.00 0.00  -0.01  -0.01   0.00   0.00   0.00   456 1.01
## lp__    -70.26    0.04 1.04 -73.06 -70.69 -69.94 -69.51 -69.23   740 1.01
##
## Samples were drawn using NUTS(diag_e) at Fri Nov  9 20:41:54 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

were there is convergence but, yet again, one of the variables is not important again.

For ratio

```
input_data = data_sub %>% select(ratio)
inputs = list(N = nrow(input_data), D=ncol(input_data), X=input_data, y=y)
```

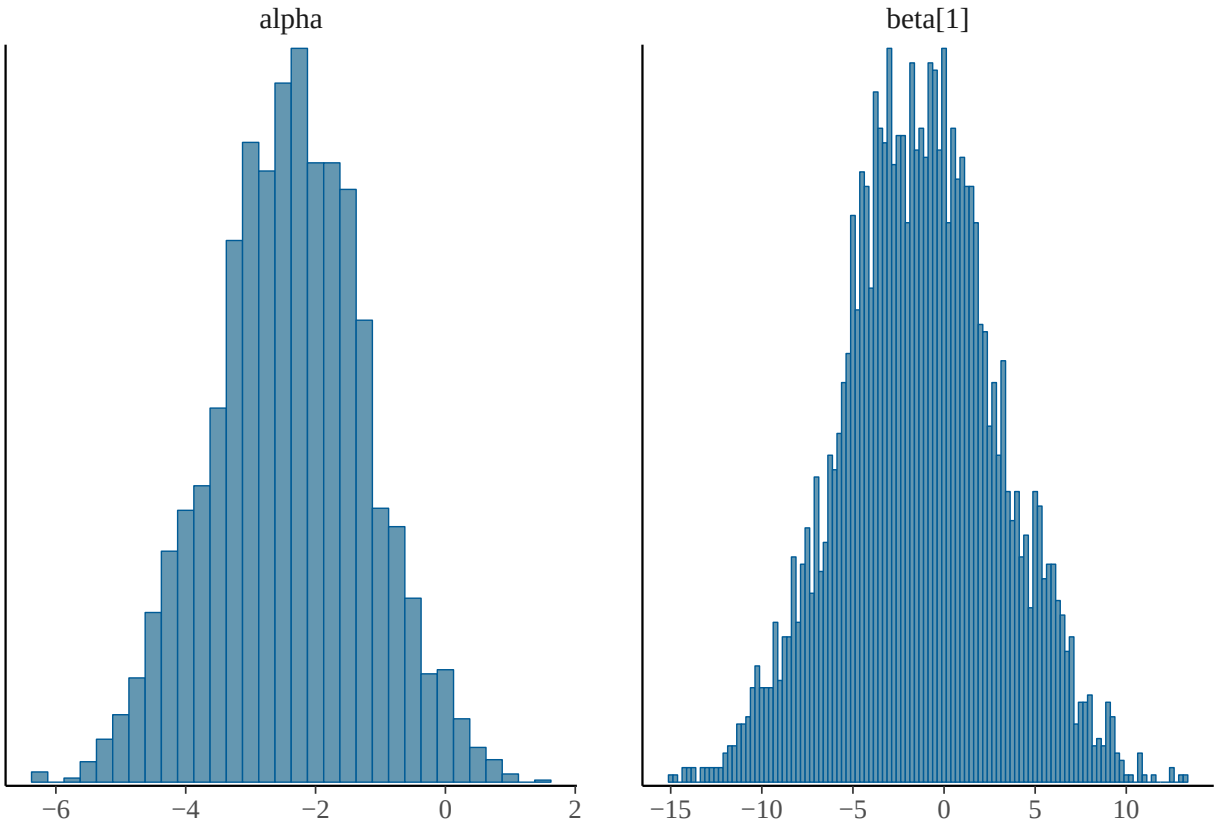
The results are the following

```
print(model_v01, digits = 2)
```

```
## Inference for Stan model: logistic_reg_v01.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd   2.5%    25%    50%   75%  97.5% n_eff Rhat
## alpha    -2.40    0.05 1.18  -4.73  -3.16  -2.37  -1.6  -0.03   625 1.01
## beta[1]  -1.28    0.17 4.29  -9.81  -4.15  -1.31   1.5   7.11   653 1.01
## lp__    -72.23    0.03 1.02 -74.99 -72.64 -71.91 -71.5 -71.24  1017 1.00
##
## Samples were drawn using NUTS(diag_e) at Fri Nov  9 20:41:55 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

were, so far, this is the only variable that has shown some significance.

```
sims = rstan::extract(model_v01)
alpha_mean = apply(X = sims$alpha, MARGIN = 1, FUN = mean)
beta_mean = apply(X = sims$beta, MARGIN = 1, FUN = mean)
true_params = c(alpha_mean, beta_mean)
posterior_params = as.matrix(model_v01, pars=c('alpha', 'beta'))
# mcmc_recover_hist(posterior_params)
mcmc_hist(posterior_params, binwidth = 0.25)
```



For lender score

```
input_data = data_sub %>% select(lender_score)
inputs = list(N = nrow(input_data), D=ncol(input_data), X=input_data, y=y)
```

The results are the following

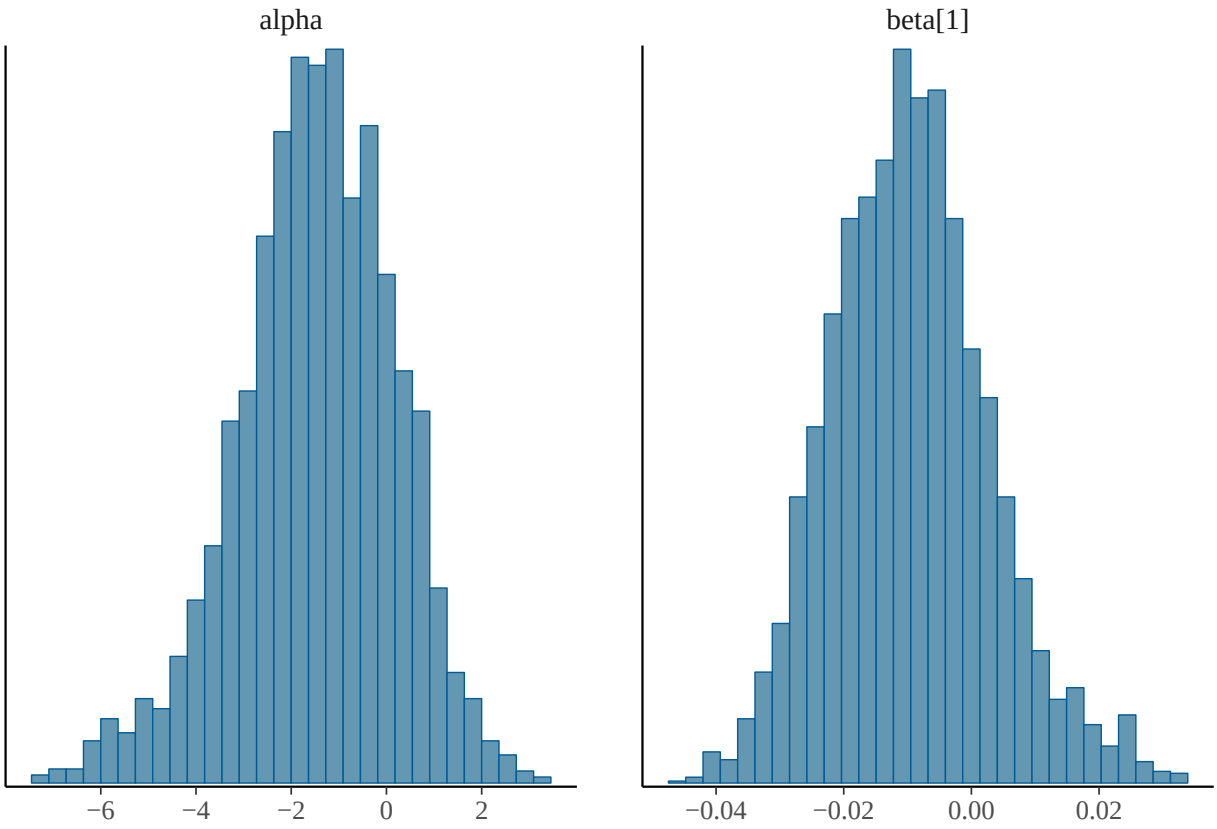
```
print(model_v01, digits = 2)
```

```
## Inference for Stan model: logistic_reg_v01.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## alpha    -1.48    0.06 1.66  -5.20  -2.49  -1.40  -0.32   1.49   760    1
## beta[1]  -0.01    0.00 0.01  -0.03  -0.02  -0.01   0.00   0.02   765    1
## lp__    -71.84    0.04 1.03 -74.65 -72.26 -71.54 -71.09 -70.79   695    1
##
## Samples were drawn using NUTS(diag_e) at Fri Nov  9 20:41:57 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

were apparently there is a little effect.

```
sims = rstan::extract(model_v01)
alpha_mean = apply(X = sims$alpha, MARGIN = 1, FUN = mean)
beta_mean = apply(X = sims$beta, MARGIN = 1, FUN = mean)
true_params = c(alpha_mean, beta_mean)
posterior_params = as.matrix(model_v01, pars=c('alpha', 'beta'))
# mcmc_recover_hist(posterior_params)
mcmc_hist(posterior_params)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



For factor employed

```
input_data = data_sub %>% select(factor_employed)
inputs = list(N = nrow(input_data), D=ncol(input_data), X=input_data, y=y)
```

The results are the following

```
print(model_v01, digits = 2)
```

```
## Inference for Stan model: logistic_reg_v01.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
## alpha    -0.53    0.05 1.19  -2.39  -1.47  -0.68   0.25   2.06   532 1.01
## beta[1]  -1.78    0.04 1.01  -4.02  -2.43  -1.61  -0.99  -0.30   531 1.01
## lp__    -69.94    0.04 1.03 -72.77 -70.35 -69.62 -69.21 -68.92   818 1.00
##
## Samples were drawn using NUTS(diag_e) at Fri Nov  9 20:41:59 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

were in the previous variable the beta never went past 0 but here we do have this effect.