1. $E_D(w) = \frac{1}{2}\sum_{n=1}^{N} r_n (t_n - w^T\phi(x_n))^2$

i) $\frac{d}{dw} E_D(w) = \sum_{n=1}^{N} r_n \{t_n - w^T\phi(x_n)\}\phi(x_n)^T = 0$

$$\sum_{n=1}^{N} r_n t_n \phi(x_n)^T = \sum_{n=1}^{N} r_n \phi(x_n)\phi(x_n)^T w$$

$$w = \frac{\sum_{n=1}^{N} r_n t_n \phi(x_n)^T}{\sum_{n=1}^{N} r_n \phi(x_n)\phi(x_n)^T}$$

ii) $E_D(w) = \frac{1}{2}\sum_{n=1}^{N} r_n (t_n - w^T\phi(x_n))^2$

Rewrite as $= \frac{1}{2}(\phi w - t)^T R(\phi w - t)$

$$= \frac{1}{2}(w^T\phi^T R\phi w - w^T\phi^T R t - t^T R\phi w + t^T R t)$$

$$= \frac{1}{2}(w^T\phi^T R\phi w - 2t^T R\phi w + t^T R t)$$

where $R = \mathrm{diag}(r_1 \dots r_n)$

$$\Delta E_D(w) = \phi^T R\phi w - t^T R\phi$$

$$w = (\phi^T R\phi)^{-1} t^T R\phi$$

$$= \phi^T R\phi^{-1}\phi^T R t.$$

2. $p(w,\beta) = \mathcal{N}(w|m_0, \beta^{-1}S_0)\,\mathrm{Gam}(\beta|a_0, b_0)$

$$\propto \left(\frac{\beta}{|S_0|}\right)^{\frac{1}{2}} \exp\left(-\frac{1}{2}(w-m_0)^T\beta S_0^{-1}(w-m_0)\right) b_0^{a_0}\beta^{a_0-1}\exp(-b_0\beta)$$

$p(t|x,w,\beta) = \prod_{n=1}^{N}\mathcal{N}(t_n|w^T\phi(x_n), \beta^{-1})$

$$\propto \prod_{n=1}^{N}\beta^{1/2}\exp\left[-\frac{\beta}{2}(t_n - w^T\phi(x_n))^2\right]$$

Since $p(w,\beta|t) \propto p(t|X,w,\beta) \times p(w,\beta)$

We have quadratic term $= -\frac{\beta}{2} w^T S_0^{-1} w + \sum_{n=1}^{N} -\frac{\beta}{2} w^T \phi(x_n)\phi(x_n)^T w$

$$= -\frac{\beta}{2} w^T \underbrace{\left(S_0^{-1} + \sum_{n=1}^{N} \phi(x_n)\phi(x_n)^T \right)}_{S_N^{-1}} w$$

$$S_N = \left[S_0^{-1} + \sum_{n=1}^{N} \phi(x_n)\phi(x_n)^T \right]^{-1}$$

linear term $= \beta m_0^T S_0^{-1} w + \sum_{n=1}^{N} \beta t_n \phi(x_n)^T w$

$$= \beta \underbrace{\left[m_0^T S_0^{-1} + \sum_{n=1}^{N} t_n \phi(x_n)^T \right]}_{m_N^T S_N^{-1}} w$$

Then $m_N = S_N \left[S_0^{-1} m_0 + \sum_{n=1}^{N} t_n \phi(x_n) \right]$

Constant term $= \left(-\frac{\beta}{2} m_0^T S_0^{-1} m_0 - b_0 \beta \right) - \frac{\beta}{2} \sum_{n=1}^{N} t_n^2$

$$= -\beta \left[\frac{1}{2} m_0^T S_0^{-1} m_0 + b_0 + \frac{1}{2} \sum_{n=1}^{N} t_n^2 \right)$$

Then $\frac{1}{2} m_N^T S_N^{-1} m_N + b_N = \frac{1}{2} m_0^T S_0^{-1} m_0 + b_0 + \frac{1}{2} \sum_{n=1}^{N} t_n^2$

$$b_N = \frac{1}{2} m_0^T S_0^{-1} m_0 + b_0 + \frac{1}{2} \sum_{n=1}^{N} t_n^2 - \frac{1}{2} m_N^T S_N^{-1} m_N$$

exp term $= (2 + a_0 - 1) + \frac{N}{2}$

$$2 + a_N - 1 = (2 + a_0 - 1) + \frac{N}{2}$$

$$a_N = a_0 + \frac{N}{2}$$

3. let $w^T\phi(x_n) > 0$ and let $w^T\phi(x_m) < 0$,

Since $P(C_1|\phi) = y(\phi) = \sigma(w^T\phi)$

when $|w| \to \infty$

$$p(C_1|\phi(x_n)) = \sigma(w^T\phi(x_n)) \to 1$$

$$p(C_2|\phi(x_m)) = 1 - p(C_1|\phi(x_m)) = 1 - \sigma(w^T\phi(x_m)) \to 1$$

4. Since y_n is the output of logistic regression model

Then $0 < y_n < 1 \Rightarrow y_n(1-y_n) > 0$

By $H = \nabla\nabla E(w) = \sum_{n=1}^{N} y_n(1-y_n)\phi_n\phi_n^T = \Phi^T R \Phi$ (4.97)

for an vector $a \neq 0$

$$a^T H a = a^T \left[\sum_{n=1}^{N} y_n(1-y_n)\phi_n\phi_n^T \right] a$$

$$= \sum_{n=1}^{N} y_n(1-y_n)(\phi_n^T a)^T(\phi_n^T a)$$

$$= \sum_{n=1}^{N} y_n(1-y_n) b_n^2 \quad \text{where denote } b_n = \phi_n^T a$$

Then There $\exists\, b_n \in \{b_1 \dots b_N\} \neq 0$.

Then $a^T H a > 0$, H is positive definite.

S.a) $f(x;\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$

$$= \frac{\beta^\alpha}{\Gamma(\alpha)} \exp\{-\beta x + (\alpha-1)\ln x\}$$

$\eta = \begin{bmatrix} -\beta \\ \alpha-1 \end{bmatrix}$, $u(x) = \begin{bmatrix} x \\ \ln x \end{bmatrix}$, $h(x)=1$ $g(\eta) = \ln \frac{\beta^\alpha}{\Gamma(\alpha)}$

b) $L(w)$: $f(t_k|x_k) = \frac{1}{\Gamma(v)} \left(\frac{vt_k}{y_k}\right)^v \frac{1}{y_k} e^{-\frac{vt_k}{y_k}}$

Then $\ln f(t_k|x_k) = v\ln(vt_k) - v\ln(y_k) - \frac{vt_k}{y_k} - \ln(\Gamma(v)) - \ln(y)$

$$= v\ln(vt_k) - \ln(\Gamma(v)) - v(w_0+w_1x_k) - \frac{vt_k}{w_0+w_1x_k} - (w_0+w_1x_k)$$

$$\frac{\partial \ln f(t_k|x_k)}{\partial w_0} = -v - \frac{vt_k}{(w_0+w_1x_k)^2} - 1$$

$$\frac{\partial^2 \ln f(t_k|x_k)}{\partial^2 w_0} = \frac{2vt_k}{(w_0+w_1x_k)^3} \geq 0$$

$$\frac{\partial \ln f(t_k|x_k)}{\partial^2 w_1} = vx_k - \frac{vt_kx_k}{(w_0+w_1x_k)^2} - x_k$$

$$\frac{\partial^2 \ln f(t_k|x_k)}{\partial^2 w_1} = \frac{2vt_kx_k(1+x_k)}{(w_0+w_1x_k)^3} \geq 0$$

Then the likelihood function is a convex function.

6 a) $p(w) = \frac{1}{2b} \exp\left(-\frac{|w|}{b}\right)$

$$L(w) = \prod_{i=1}^{n} \frac{1}{2b} \exp\left(-\frac{|w_i|}{b}\right)$$

$$= \left(\frac{1}{2b}\right)^n \cancel{\exp} \prod_{i=1}^{n} \exp\left(-\frac{|w_i|}{b}\right)$$

$$\log L(w) = -n \log(2b) - \frac{1}{b} \sum_{i=1}^{n} |w_i|$$

The Method of moments is

$E_n(x) = x^{n-1} \Gamma(1-n, x)$, which is a Lasso regression.

hw2

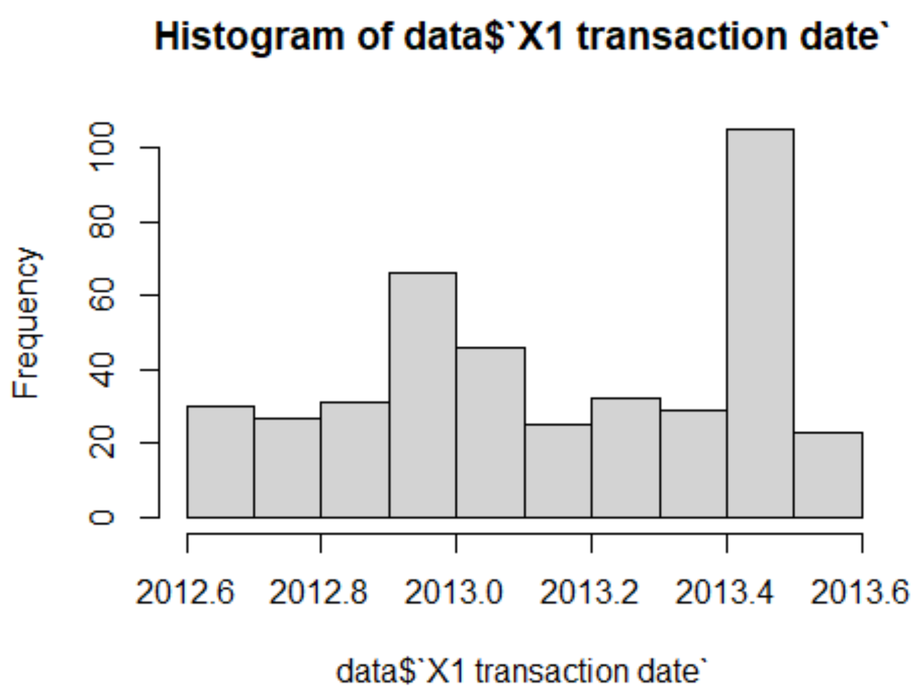
Enbo Tian

2/14/2022

Linear Regression Problem

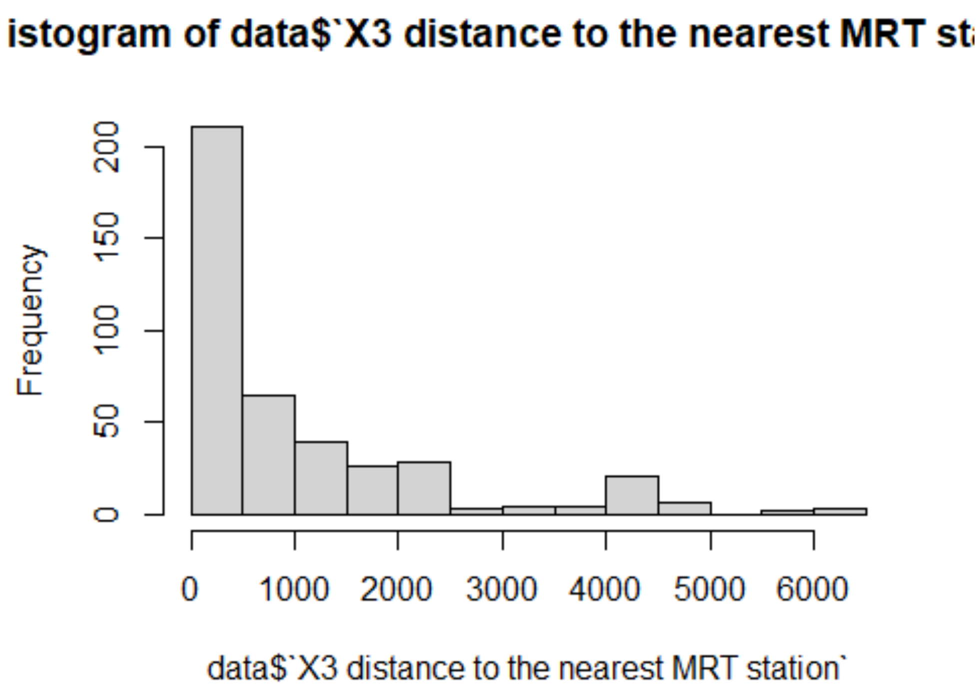
1)

```
rm(list=ls())
library("readxl")
data<-read_excel("Real estate valuation data set.xlsx")
## a
hist(data$`X1 transaction date`)
```

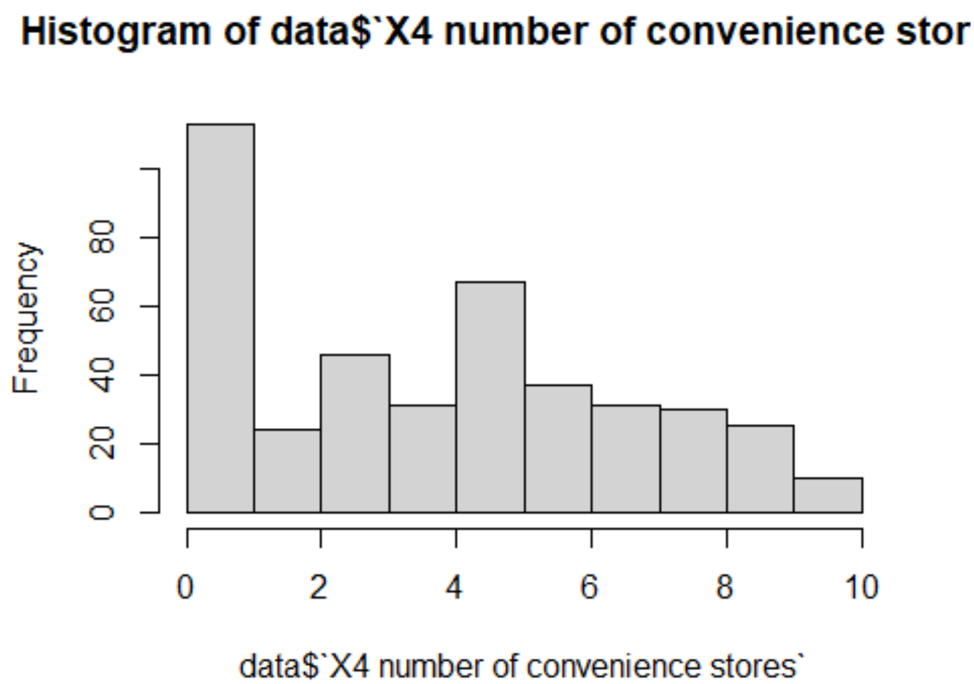



```
hist(data$`X2 house age`)
```

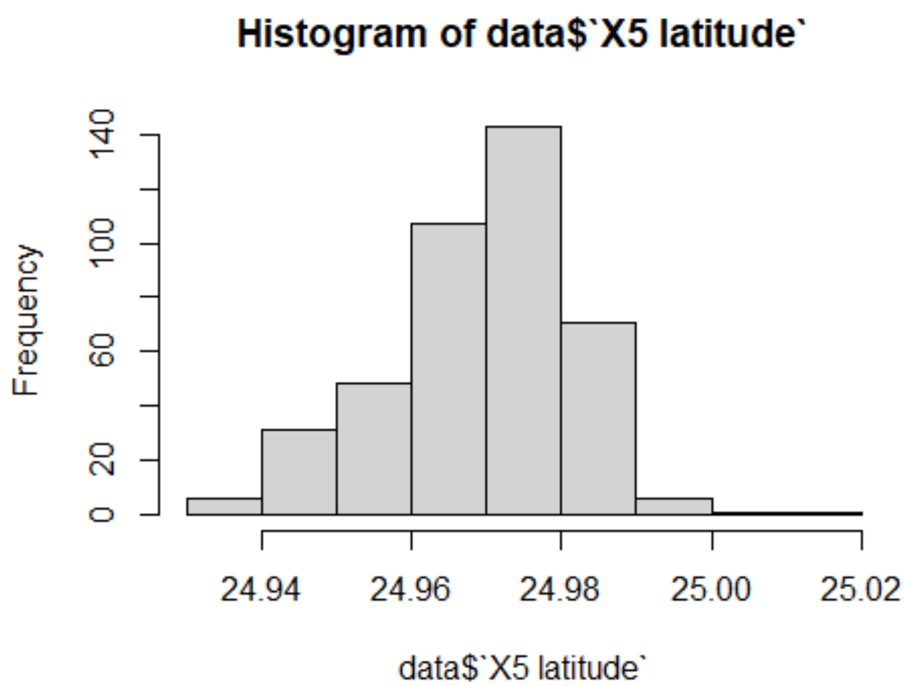

```
hist(data$`X3 distance to the nearest MRT station`)
```



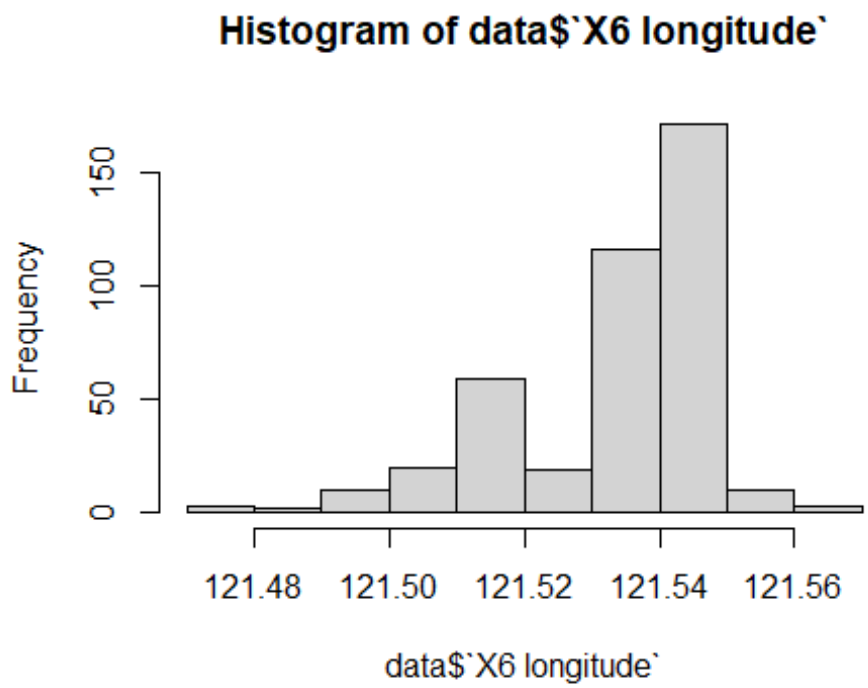

```
hist(data$`X4 number of convenience stores`)
```



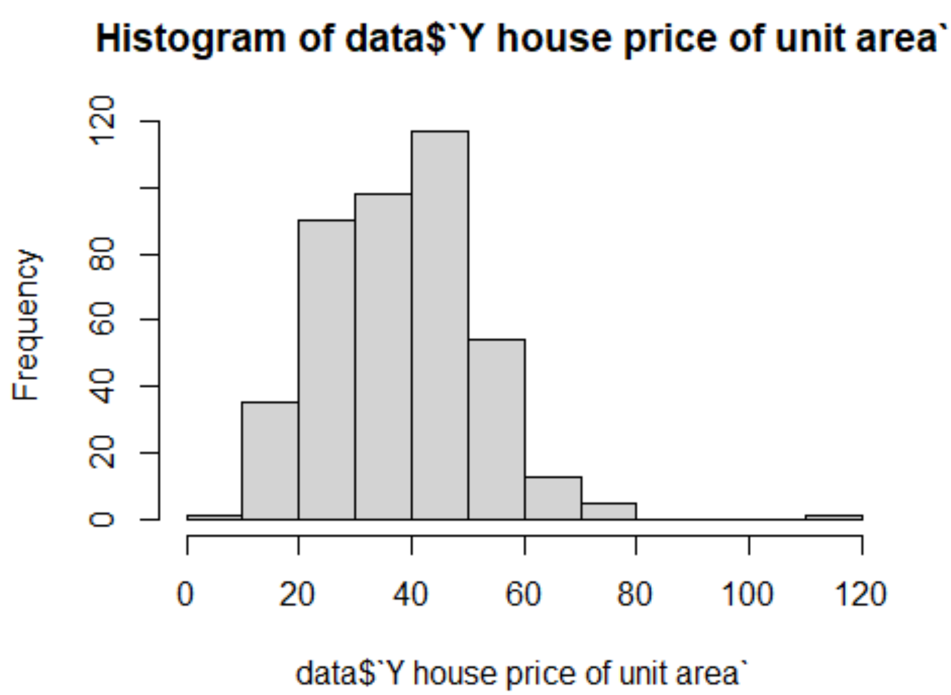

```r
hist(data$`X5 latitude`)
```



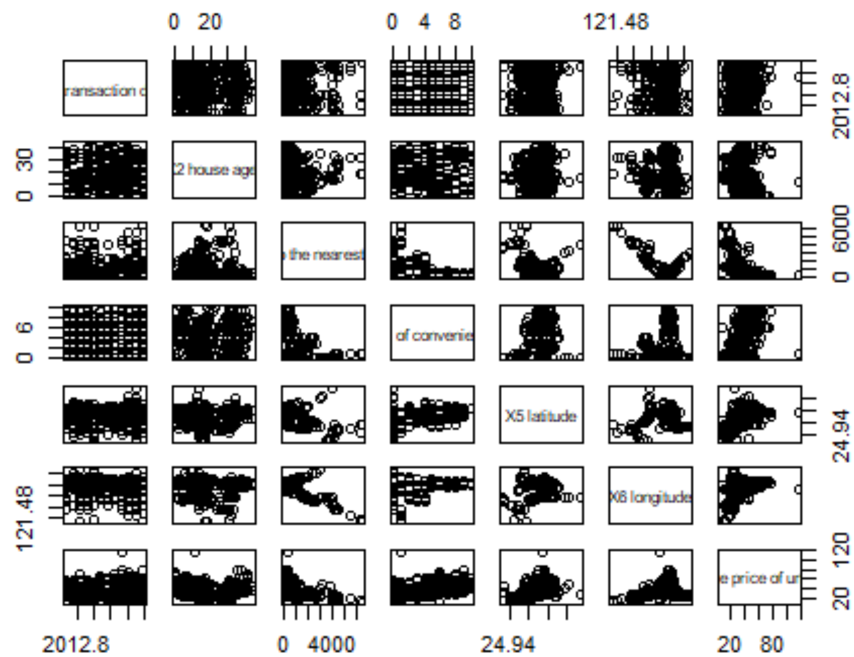

```r
hist(data$`X6 longitude`)
```



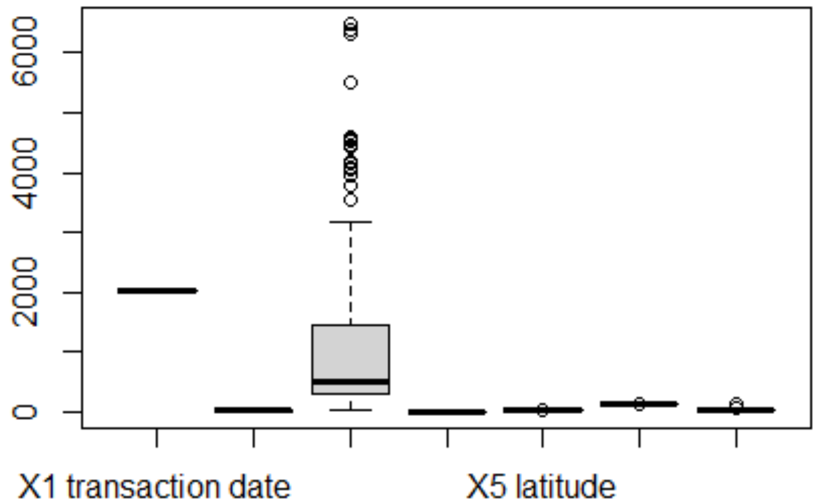

```
hist(data$`Y house price of unit area`)
```



```r
## b
plot(data[,2:8])
```




```r
boxplot(data[,2:8])
```




```
## There is significant outlier for X3: distance to the nearest MRT sta
tion
cor(data[,2:8])

##                                          X1 transaction date X2 house
age
## X1 transaction date                              1.000000000   0.01754
234
## X2 house age                                     0.017542341   1.00000
000
## X3 distance to the nearest MRT station           0.060880095   0.02562
205
## X4 number of convenience stores                  0.009544199   0.04959
251
## X5 latitude                                      0.035016305   0.05441
990
## X6 longitude                                    -0.041065078  -0.04852
005
## Y house price of unit area                       0.087529272  -0.21056
705
##                                          X3 distance to the nearest MR
T station
## X1 transaction date                                                  0.
06088009
## X2 house age                                                         0.
02562205
## X3 distance to the nearest MRT station                               1.
```

```
00000000
## X4 number of convenience stores                                     -0.
60251914
## X5 latitude                                                         -0.
59106657
## X6 longitude                                                        -0.
80631677
## Y house price of unit area                                          -0.
67361286
##                                        X4 number of convenience stor
es
## X1 transaction date                                         0.0095441
99
## X2 house age                                                0.0495925
13
## X3 distance to the nearest MRT station                     -0.6025191
45
## X4 number of convenience stores                             1.0000000
00
## X5 latitude                                                 0.4441433
06
## X6 longitude                                                0.4490990
07
## Y house price of unit area                                  0.5710049
11
##                                        X5 latitude X6 longitude
## X1 transaction date                     0.03501631  -0.04106508
## X2 house age                            0.05441990  -0.04852005
## X3 distance to the nearest MRT station -0.59106657  -0.80631677
## X4 number of convenience stores         0.44414331   0.44909901
## X5 latitude                             1.00000000   0.41292394
## X6 longitude                            0.41292394   1.00000000
## Y house price of unit area              0.54630665   0.52328651
##                                        Y house price of unit area
## X1 transaction date                                    0.08752927
## X2 house age                                          -0.21056705
## X3 distance to the nearest MRT station                -0.67361286
## X4 number of convenience stores                        0.57100491
## X5 latitude                                            0.54630665
## X6 longitude                                           0.52328651
## Y house price of unit area                             1.00000000
```

2

```
df <- data.frame(data[2:8])
LR<-lm(data$`Y house price of unit area`~data$`X1 transaction date`
       +data$`X2 house age`
       +data$`X3 distance to the nearest MRT station`
       +data$`X4 number of convenience stores`
       +data$`X5 latitude`+data$`X6 longitude`)
summary(LR)
```

```
##
## Call:
## lm(formula = data$`Y house price of unit area` ~ data$`X1 transactio
n date` +
##     data$`X2 house age` + data$`X3 distance to the nearest MRT stati
on` +
##     data$`X4 number of convenience stores` + data$`X5 latitude` +
##     data$`X6 longitude`)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.667  -5.412  -0.967   4.217  75.190
##
## Coefficients:
##                                                  Estimate Std. Error
t value
## (Intercept)                                    -1.444e+04  6.775e+03
 -2.132
## data$`X1 transaction date`                      5.149e+00  1.557e+00
  3.307
## data$`X2 house age`                            -2.697e-01  3.853e-02
 -7.000
## data$`X3 distance to the nearest MRT station` -4.488e-03  7.180e-04
 -6.250
## data$`X4 number of convenience stores`          1.133e+00  1.882e-01
  6.023
## data$`X5 latitude`                              2.255e+02  4.457e+01
  5.059
## data$`X6 longitude`                            -1.243e+01  4.858e+01
 -0.256
##                                                Pr(>|t|)
## (Intercept)                                     0.03364 *
## data$`X1 transaction date`                      0.00103 **
## data$`X2 house age`                            1.06e-11 ***
## data$`X3 distance to the nearest MRT station` 1.04e-09 ***
## data$`X4 number of convenience stores`         3.83e-09 ***
## data$`X5 latitude`                             6.38e-07 ***
## data$`X6 longitude`                             0.79820
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.858 on 407 degrees of freedom
## Multiple R-squared:  0.5824, Adjusted R-squared:  0.5762
## F-statistic:  94.6 on 6 and 407 DF,  p-value: < 2.2e-16
```

```
# predicted weights
summary(LR)$coefficients
```

```
##                                                    Estimate   Std. E
rror
```

```
## (Intercept)                                   -1.444198e+04 6.775386
e+03
## data$`X1 transaction date`                     5.149017e+00 1.556876
e+00
## data$`X2 house age`                           -2.696967e-01 3.852998
e-02
## data$`X3 distance to the nearest MRT station` -4.487508e-03 7.180118
e-04
## data$`X4 number of convenience stores`         1.133325e+00 1.881597
e-01
## data$`X5 latitude`                             2.254701e+02 4.456578
e+01
## data$`X6 longitude`                           -1.242906e+01 4.858117
e+01
##                                                  t value     Pr(>|t|)
## (Intercept)                                   -2.1315365 3.364344e-0
2
## data$`X1 transaction date`                     3.3072743 1.025782e-0
3
## data$`X2 house age`                           -6.9996591 1.063915e-1
1
## data$`X3 distance to the nearest MRT station` -6.2499086 1.037344e-0
9
## data$`X4 number of convenience stores`         6.0232088 3.826895e-0
9
## data$`X5 latitude`                             5.0592667 6.382166e-0
7
## data$`X6 longitude`                           -0.2558411 7.982028e-0
1

# price as a function of time(X1)
## increase in 5.149
# RMSE
summary(LR)$sigma

## [1] 8.857515
```

3

```
library(rstanarm)

## Loading required package: Rcpp

## This is rstanarm version 2.21.1

## - See https://mc-stan.org/rstanarm/articles/priors for changes to de
fault priors!

## - Default priors may change, so it's safest to specify priors, even
if equivalent to the defaults.
```

```
## - For execution on a local, multicore CPU with excess RAM we recomme
nd calling

##   options(mc.cores = parallel::detectCores())
```

```
# Lambda = 1
bayesm<- stan_glm(df$Y.house.price.of.unit.area~.,data=df,prior_aux = e
xponential(1))
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.086 seconds (Warm-up)
## Chain 1:                0.117 seconds (Sampling)
## Chain 1:                0.203 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
```

```
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2: 
## Chain 2:  Elapsed Time: 0.085 seconds (Warm-up)
## Chain 2:                0.111 seconds (Sampling)
## Chain 2:                0.196 seconds (Total)
## Chain 2: 
## 
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3: 
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3: 
## Chain 3: 
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3: 
## Chain 3:  Elapsed Time: 0.095 seconds (Warm-up)
## Chain 3:                0.106 seconds (Sampling)
## Chain 3:                0.201 seconds (Total)
## Chain 3: 
## 
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4: 
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4: 
## Chain 4: 
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
```

```
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.079 seconds (Warm-up)
## Chain 4:                0.108 seconds (Sampling)
## Chain 4:                0.187 seconds (Total)
## Chain 4:
```

```
summary(bayesm)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      df$Y.house.price.of.unit.area ~ .
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 414
##  predictors:   7
##
## Estimates:
##                                            mean     sd       10%
50%
## (Intercept)                              -14500.0   6527.6 -22783.2 -1
4402.5
## X1.transaction.date                          5.2      1.5      3.2
   5.2
## X2.house.age                                -0.3      0.0     -0.3
  -0.3
## X3.distance.to.the.nearest.MRT.station      0.0      0.0      0.0
   0.0
## X4.number.of.convenience.stores              1.1      0.2      0.9
   1.1
## X5.latitude                                225.5     44.7    168.5
 225.4
## X6.longitude                               -12.0     47.4    -72.0
 -12.7
## sigma                                        8.8      0.3      8.4
   8.8
##                                            90%
## (Intercept)                               -6201.9
## X1.transaction.date                          7.1
## X2.house.age                                -0.2
```

```
## X3.distance.to.the.nearest.MRT.station      0.0
## X4.number.of.convenience.stores             1.4
## X5.latitude                               283.6
## X6.longitude                               48.5
## sigma                                       9.2
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 38.0    0.6 37.2  38.0  38.8
##
## The mean_ppd is the sample average posterior predictive distribution
 of the outcome variable (for details see help('summary.stanreg')).
##
## MCMC diagnostics
##                                           mcse  Rhat  n_eff
## (Intercept)                               103.8   1.0 3956
## X1.transaction.date                         0.0   1.0 5469
## X2.house.age                                0.0   1.0 6011
## X3.distance.to.the.nearest.MRT.station      0.0   1.0 3093
## X4.number.of.convenience.stores             0.0   1.0 5193
## X5.latitude                                 0.7   1.0 4665
## X6.longitude                                0.8   1.0 3712
## sigma                                       0.0   1.0 5577
## mean_PPD                                    0.0   1.0 4534
## log-posterior                               0.0   1.0 1698
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a c
rude measure of effective sample size, and Rhat is the potential scale
reduction factor on split chains (at convergence Rhat=1).
```

```r
# Lambda = 10
bayesm2<- stan_glm(df$Y.house.price.of.unit.area~.,data=df,prior_aux =
exponential(10))
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
```

```
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1: 
## Chain 1:  Elapsed Time: 0.1 seconds (Warm-up)
## Chain 1:                0.113 seconds (Sampling)
## Chain 1:                0.213 seconds (Total)
## Chain 1: 
## 
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2: 
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2: 
## Chain 2: 
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2: 
## Chain 2:  Elapsed Time: 0.102 seconds (Warm-up)
## Chain 2:                0.113 seconds (Sampling)
## Chain 2:                0.215 seconds (Total)
## Chain 2: 
## 
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3: 
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3: 
## Chain 3: 
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
```

```
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.09 seconds (Warm-up)
## Chain 3:                0.107 seconds (Sampling)
## Chain 3:                0.197 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.09 seconds (Warm-up)
## Chain 4:                0.117 seconds (Sampling)
## Chain 4:                0.207 seconds (Total)
## Chain 4:
```

```
summary(bayesm2)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      df$Y.house.price.of.unit.area ~ .
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 414
```

```
##  predictors:   7
##
## Estimates:
##                                            mean     sd       10%     
50%  
## (Intercept)                              -14463.2   6122.8 -22280.7 -1
4436.5
## X1.transaction.date                           5.1      1.4      3.3  
   5.1
## X2.house.age                                 -0.3      0.0     -0.3  
  -0.3
## X3.distance.to.the.nearest.MRT.station      0.0      0.0      0.0  
   0.0
## X4.number.of.convenience.stores             1.1      0.2      0.9  
   1.1
## X5.latitude                                224.8     41.2    172.7  
 225.3
## X6.longitude                               -12.1     43.8    -68.1  
 -12.8
## sigma                                         8.1      0.3      7.8  
   8.1
##                                            90%   
## (Intercept)                               -6690.0
## X1.transaction.date                           7.0
## X2.house.age                                 -0.2
## X3.distance.to.the.nearest.MRT.station      0.0
## X4.number.of.convenience.stores             1.4
## X5.latitude                                276.8
## X6.longitude                                 43.5
## sigma                                         8.4
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 38.0    0.6 37.3  38.0  38.7 
##
## The mean_ppd is the sample average posterior predictive distribution
 of the outcome variable (for details see help('summary.stanreg')).
##
## MCMC diagnostics
##                                        mcse  Rhat  n_eff
## (Intercept)                            100.4   1.0 3717 
## X1.transaction.date                      0.0   1.0 5862 
## X2.house.age                             0.0   1.0 5400 
## X3.distance.to.the.nearest.MRT.station   0.0   1.0 2896 
## X4.number.of.convenience.stores          0.0   1.0 4984 
## X5.latitude                              0.6   1.0 4421 
## X6.longitude                             0.8   1.0 3384 
## sigma                                    0.0   1.0 4929 
## mean_PPD                                 0.0   1.0 4577 
## log-posterior                            0.0   1.0 1847 
```

```
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a c
rude measure of effective sample size, and Rhat is the potential scale
reduction factor on split chains (at convergence Rhat=1).
```

```
# lambda = 100
bayesm3<- stan_glm(df$Y.house.price.of.unit.area~.,data=df,prior_aux =
exponential(100))
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.116 seconds (Warm-up)
## Chain 1:                0.106 seconds (Sampling)
## Chain 1:                0.222 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
```

```
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.107 seconds (Warm-up)
## Chain 2:                0.108 seconds (Sampling)
## Chain 2:                0.215 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.099 seconds (Warm-up)
## Chain 3:                0.106 seconds (Sampling)
## Chain 3:                0.205 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
```

```
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.077 seconds (Warm-up)
## Chain 4:                0.106 seconds (Sampling)
## Chain 4:                0.183 seconds (Total)
## Chain 4:
```

```
summary(bayesm3)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      df$Y.house.price.of.unit.area ~ .
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 414
##  predictors:   7
##
## Estimates:
##                                            mean     sd       10%
50%
## (Intercept)                              -14593.4   4374.2 -20304.7 -1
4576.4
## X1.transaction.date                           5.2      1.0      3.9
   5.2
## X2.house.age                                 -0.3      0.0     -0.3
  -0.3
## X3.distance.to.the.nearest.MRT.station      0.0      0.0      0.0
   0.0
## X4.number.of.convenience.stores              1.1      0.1      1.0
   1.1
## X5.latitude                                 225.4     28.8    188.3
 225.2
## X6.longitude                                -11.4     31.6    -51.0
 -12.3
## sigma                                         5.7      0.1      5.6
   5.7
##                                            90%
## (Intercept)                               -9008.1
## X1.transaction.date                          6.4
## X2.house.age                                -0.2
```

```
## X3.distance.to.the.nearest.MRT.station      0.0
## X4.number.of.convenience.stores             1.3
## X5.latitude                               262.5
## X6.longitude                               30.0
## sigma                                       5.9
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 38.0    0.4 37.5  38.0  38.5
##
## The mean_ppd is the sample average posterior predictive distribution
 of the outcome variable (for details see help('summary.stanreg')).
##
## MCMC diagnostics
##                                         mcse Rhat n_eff
## (Intercept)                             71.7  1.0 3718
## X1.transaction.date                      0.0  1.0 5307
## X2.house.age                             0.0  1.0 5596
## X3.distance.to.the.nearest.MRT.station  0.0  1.0 3286
## X4.number.of.convenience.stores          0.0  1.0 3709
## X5.latitude                              0.4  1.0 4599
## X6.longitude                             0.5  1.0 3575
## sigma                                    0.0  1.0 4994
## mean_PPD                                 0.0  1.0 4572
## log-posterior                            0.0  1.0 1887
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a c
rude measure of effective sample size, and Rhat is the potential scale
reduction factor on split chains (at convergence Rhat=1).
```

4

```
library(robustHD)

## Loading required package: ggplot2

## Loading required package: perry

## Loading required package: parallel

## Loading required package: robustbase

x2 <- data$`X2 house age`
x3 <- data$`X3 distance to the nearest MRT station`

s2<-standardize(x2)
s3<-standardize(x3)

bayesm4<- stan_glm(data$`Y house price of unit area`~s2+s3,data=data)

##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
```

```
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.057 seconds (Warm-up)
## Chain 1:                0.073 seconds (Sampling)
## Chain 1:                0.13 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.059 seconds (Warm-up)
## Chain 2:                0.078 seconds (Sampling)
## Chain 2:                0.137 seconds (Total)
## Chain 2:
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.044 seconds (Warm-up)
## Chain 3:                0.067 seconds (Sampling)
## Chain 3:                0.111 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.048 seconds (Warm-up)
## Chain 4:                0.077 seconds (Sampling)
```

```
## Chain 4:                0.125 seconds (Total)
## Chain 4:
```

```
summary(bayesm4)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      data$`Y house price of unit area` ~ s2 + s3
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 414
##  predictors:   3
##
## Estimates:
##                mean   sd   10%   50%   90%
## (Intercept) 38.0    0.5 37.4  38.0  38.6
## s2          -2.6    0.5 -3.3  -2.6  -2.0
## s3          -9.1    0.5 -9.7  -9.1  -8.5
## sigma        9.7    0.3  9.3   9.7  10.2
##
## Fit Diagnostics:
##             mean   sd   10%   50%   90%
## mean_PPD 38.0    0.7 37.1  38.0  38.8
##
## The mean_ppd is the sample average posterior predictive distribution
 of the outcome variable (for details see help('summary.stanreg')).
##
## MCMC diagnostics
##                mcse Rhat n_eff
## (Intercept)   0.0  1.0  5171
## s2            0.0  1.0  4699
## s3            0.0  1.0  4916
## sigma         0.0  1.0  5197
## mean_PPD      0.0  1.0  4663
## log-posterior 0.0  1.0  1580
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a c
rude measure of effective sample size, and Rhat is the potential scale
reduction factor on split chains (at convergence Rhat=1).
```

5

```
library(bayestestR)
map_estimate(bayesm)
```

```
## MAP Estimate
##
## Parameter                              | MAP_Estimate
```

```
## -----------------------------------------------------------
## (Intercept)                                |     -13565.89
## X1.transaction.date                        |          5.16
## X2.house.age                               |         -0.28
## X3.distance.to.the.nearest.MRT.station |     -4.41e-03
## X4.number.of.convenience.stores            |          1.15
## X5.latitude                                |        223.37
## X6.longitude                               |        -18.29

map_estimate(bayesm4)

## MAP Estimate
##
## Parameter    | MAP_Estimate
## --------------------------
## (Intercept) |        38.03
## s2          |        -2.60
## s3          |        -9.16

M = 7
N = 414

lnp1 <- 1727- 1/2*7*log(414)
lnp2 <- 1747 -1/2*7*log(414)
lnp1

## [1] 1705.909

lnp2

## [1] 1725.909
```

7 Gaussian basis model have a high evidence

6

```
#kfold1<-kfold(bayesm,K = 10)
#kfold2 <- kfold(bayesm4,K=10)
#loo_compare(kfold1,kfold2)
```

bayes model too large to run k fold cv.

Classification Problem

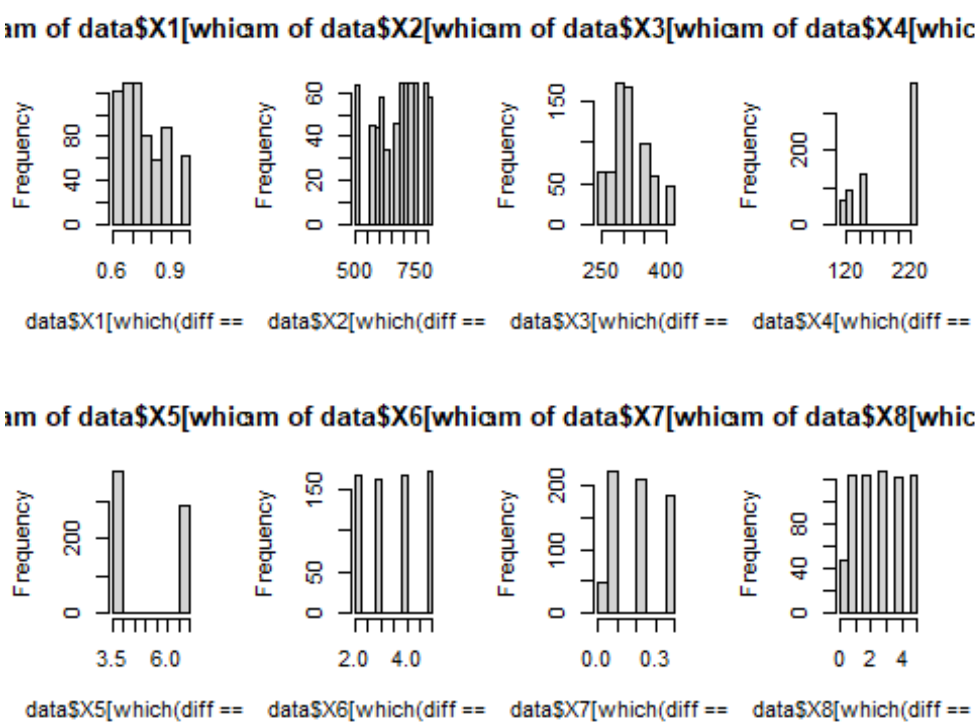
1

```
rm(list=ls())
data<-read_excel("ENB2012_data.xlsx")

diff <- rep(0,768)

for( i in 1: 768){
```

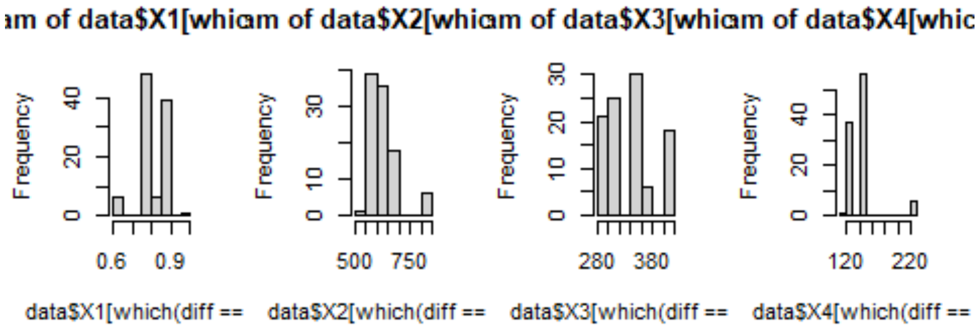
```
  if(data$Y1[i]-data$Y2[i]>0){
     diff[i] <- 1
  }
}
```

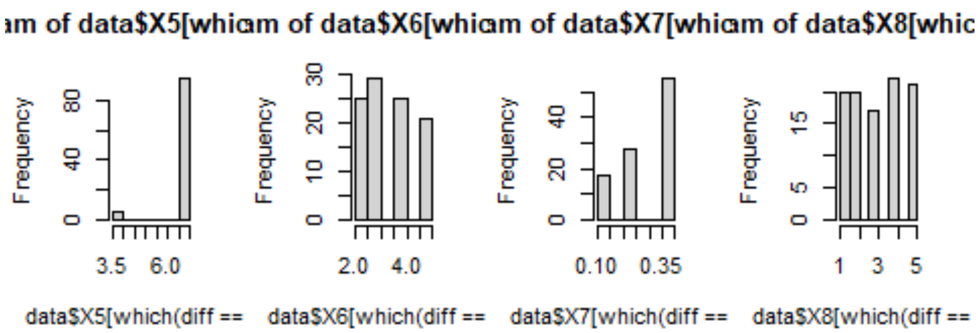
1

```
# label 0
par(mfrow=c(2,4))
hist(data$X1[which(diff==0)])
hist(data$X2[which(diff==0)])
hist(data$X3[which(diff==0)])
hist(data$X4[which(diff==0)])
hist(data$X5[which(diff==0)])
hist(data$X6[which(diff==0)])
hist(data$X7[which(diff==0)])
hist(data$X8[which(diff==0)])
```




```
# label 1
hist(data$X1[which(diff==1)])
hist(data$X2[which(diff==1)])
hist(data$X3[which(diff==1)])
hist(data$X4[which(diff==1)])
hist(data$X5[which(diff==1)])
hist(data$X6[which(diff==1)])
hist(data$X7[which(diff==1)])
hist(data$X8[which(diff==1)])
```






```
df<-data.frame(data[,1:8],diff)
cor(df)

##                 X1            X2         X3            X4         X5
## X1    1.000000e+00 -9.919015e-01 -0.2037817 -8.688234e-01  0.8277473
## X2   -9.919015e-01  1.000000e+00  0.1955016  8.807195e-01 -0.8581477
## X3   -2.037817e-01  1.955016e-01  1.0000000 -2.923165e-01  0.2809757
## X4   -8.688234e-01  8.807195e-01 -0.2923165  1.000000e+00 -0.9725122
## X5    8.277473e-01 -8.581477e-01  0.2809757 -9.725122e-01  1.0000000
## X6    0.000000e+00  0.000000e+00  0.0000000  0.000000e+00  0.0000000
## X7    7.617400e-20  4.664140e-20  0.0000000 -1.197187e-19  0.0000000
## X8    0.000000e+00  0.000000e+00  0.0000000  0.000000e+00  0.0000000
## diff  1.787384e-01 -2.042417e-01  0.2022058 -2.968207e-01  0.3404822
##               X6            X7         X8        diff
## X1    0.00000000  7.617400e-20 0.00000000  0.17873840
## X2    0.00000000  4.664140e-20 0.00000000 -0.20424166
## X3    0.00000000  0.000000e+00 0.00000000  0.20220577
## X4    0.00000000 -1.197187e-19 0.00000000 -0.29682073
## X5    0.00000000  0.000000e+00 0.00000000  0.34048222
## X6    1.00000000  0.000000e+00 0.00000000 -0.02768514
## X7    0.00000000  1.000000e+00 0.21296422  0.21106177
## X8    0.00000000  2.129642e-01 1.00000000  0.05679049
## diff -0.02768514  2.110618e-01 0.05679049  1.00000000
```

x5 have the highest predictive power

2

```r
model <- glm(diff~.,data = df, family=binomial)
summary(model)
```

```
##
## Call:
## glm(formula = diff ~ ., family = binomial, data = df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.54464  -0.49154  -0.09946  -0.00436   2.53343
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.212e+02  9.823e+01  -3.270 0.001074 **
## X1           1.520e+02  4.982e+01   3.052 0.002273 **
## X2           2.651e-01  8.074e-02   3.283 0.001027 **
## X3          -3.370e-02  9.573e-03  -3.520 0.000431 ***
## X4                  NA         NA      NA       NA
## X5           6.019e+00  1.508e+00   3.992 6.54e-05 ***
## X6          -1.001e-01  1.121e-01  -0.893 0.371862
## X7           6.340e+00  1.058e+00   5.995 2.04e-09 ***
## X8           6.424e-02  8.584e-02   0.748 0.454258
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 594.10  on 767  degrees of freedom
## Residual deviance: 400.32  on 760  degrees of freedom
## AIC: 416.32
##
## Number of Fisher Scoring iterations: 9
```

```r
## predicted weights: estimate
## classifier accuracy: AIC
```

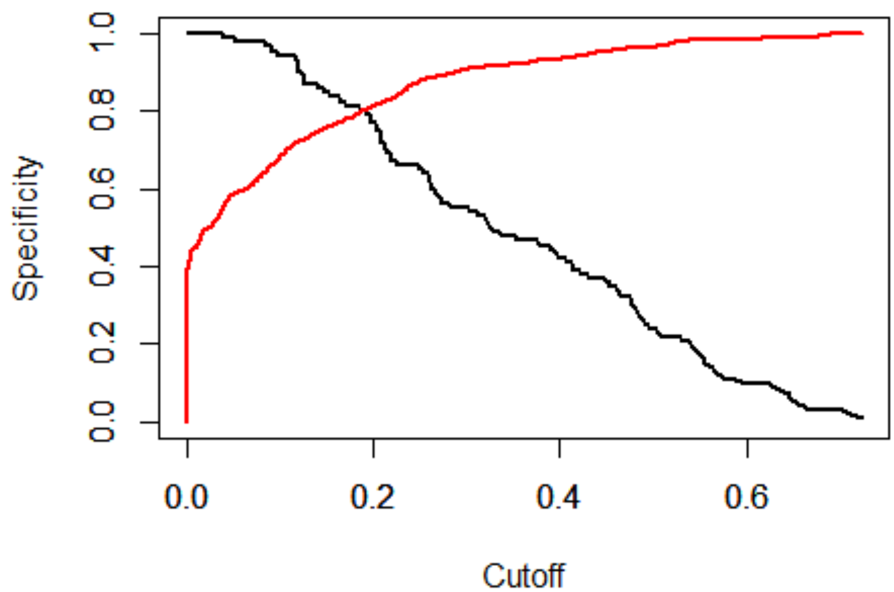
3

```r
library(ROCR)
pred = predict(model, newdata=df, type="response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if
(type == :
## prediction from a rank-deficient fit may be misleading
```

```r
predictions = prediction(pred, diff, label.ordering = NULL)
plot(unlist(performance(predictions, "sens")@x.values), unlist(performa
nce(predictions, "sens")@y.values),
     type="l", lwd=2, ylab="Specificity", xlab="Cutoff")
par(new=TRUE)
```

```r
plot(unlist(performance(predictions, "spec")@x.values), unlist(performa
nce(predictions, "spec")@y.values),
     type="l", lwd=2, col='red', ylab="", xlab="")
```



4

```r
library(rstanarm)
bayes<- stan_glm(diff~.,data=df,prior = normal(location = 0, scale = 0.
1, autoscale = FALSE) )

##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
```

```
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.052 seconds (Warm-up)
## Chain 1:                0.133 seconds (Sampling)
## Chain 1:                0.185 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.072 seconds (Warm-up)
## Chain 2:                0.14 seconds (Sampling)
## Chain 2:                0.212 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
```

```
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.059 seconds (Warm-up)
## Chain 3:                0.104 seconds (Sampling)
## Chain 3:                0.163 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.057 seconds (Warm-up)
## Chain 4:                0.1 seconds (Sampling)
## Chain 4:                0.157 seconds (Total)
## Chain 4:

## Warning: There were 57 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-w
armup
## to find out why this is a problem and how to eliminate them.

## Warning: There were 4 chains where the estimated Bayesian Fraction o
f Missing Information was low. See
## https://mc-stan.org/misc/warnings.html#bfmi-low

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

## Warning: The largest R-hat is 4.55, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#r-hat
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating po
sterior means and medians may be unreliable.
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating po
sterior variances and tail quantiles may be unreliable.
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#tail-ess

## Warning: Markov chains did not converge! Do not analyze results!
```

```
summary(bayes)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      diff ~ .
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 768
##  predictors:   9
##
## Estimates:
##               mean   sd    10%   50%   90%
## (Intercept) 115.6   19.0  93.2 113.7 141.8
## X1            0.1    0.1  -0.1   0.2   0.2
## X2           -0.1    0.0  -0.2  -0.1  -0.1
## X3           -0.1    0.1  -0.2  -0.1   0.0
## X4           -0.1    0.1  -0.2  -0.1   0.1
## X5            0.0    0.1  -0.1  -0.1   0.2
## X6            0.0    0.1  -0.1   0.0   0.1
## X7            0.1    0.1  -0.1   0.1   0.2
## X8            0.0    0.1  -0.2   0.0   0.1
## sigma         0.3    0.3   0.1   0.1   0.9
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD  0.2    0.9 -1.3   0.7   0.9
##
## The mean_ppd is the sample average posterior predictive distribution
 of the outcome variable (for details see help('summary.stanreg')).
##
## MCMC diagnostics
##               mcse         Rhat         n_eff
## (Intercept)   1.340000e+01 9.655939e+07 2
## X1            1.000000e-01 6.003400e+08 2
## X2            0.000000e+00 1.147783e+08 2
## X3            0.000000e+00 2.500334e+08 2
```

```
## X4            1.000000e-01 4.303424e+08 2
## X5            1.000000e-01 5.977296e+08 2
## X6            1.000000e-01 5.491989e+08 2
## X7            1.000000e-01 4.648210e+08 2
## X8            1.000000e-01 4.433299e+08 2
## sigma         2.000000e-01 4.957460e+04 2
## mean_PPD      6.000000e-01 5.830000e+01 2
## log-posterior 1.202593e+12 9.599700e+03 2
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a c
rude measure of effective sample size, and Rhat is the potential scale
reduction factor on split chains (at convergence Rhat=1).
```

```
bayes2<- stan_glm(diff~.,data=df,prior = normal(location = 0, scale = 1,
 autoscale = FALSE) )
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.054 seconds (Warm-up)
## Chain 1:                0.124 seconds (Sampling)
## Chain 1:                0.178 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
```

```
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.104 seconds (Warm-up)
## Chain 2:                0.087 seconds (Sampling)
## Chain 2:                0.191 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.042 seconds (Warm-up)
## Chain 3:                0.097 seconds (Sampling)
## Chain 3:                0.139 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
```

```
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.064 seconds (Warm-up)
## Chain 4:                0.11 seconds (Sampling)
## Chain 4:                0.174 seconds (Total)
## Chain 4:

## Warning: There were 195 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-w
armup
## to find out why this is a problem and how to eliminate them.

## Warning: There were 4 chains where the estimated Bayesian Fraction o
f Missing Information was low. See
## https://mc-stan.org/misc/warnings.html#bfmi-low

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

## Warning: The largest R-hat is 4.55, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#r-hat

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating po
sterior means and medians may be unreliable.
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating po
sterior variances and tail quantiles may be unreliable.
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#tail-ess

## Warning: Markov chains did not converge! Do not analyze results!

summary(bayes2)

##
## Model Info:
```

```
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      diff ~ .
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 768
##  predictors:   9
##
## Estimates:
##                mean   sd     10%    50%    90%
## (Intercept) 1014.7  729.0 -219.6 1301.8 1674.7
## X1            -0.8    1.3   -1.9   -1.4    1.4
## X2            -0.7    1.4   -1.5   -1.4    1.7
## X3            -1.2    0.8   -1.9   -1.3    0.0
## X4            -1.1    0.7   -1.6   -1.5    0.2
## X5             1.3    0.8    0.0    1.6    1.9
## X6             0.7    0.6   -0.2    0.8    1.4
## X7             0.2    1.7   -1.7    0.2    2.0
## X8            -0.1    1.3   -1.5   -0.2    1.5
## sigma          0.6    0.3    0.0    0.6    1.0
##
## Fit Diagnostics:
##             mean   sd   10%   50%   90%
## mean_PPD -0.3    1.2 -1.5  -0.7   1.7
##
## The mean_ppd is the sample average posterior predictive distribution
 of the outcome variable (for details see help('summary.stanreg')).
##
## MCMC diagnostics
##                mcse         Rhat         n_eff
## (Intercept)    5.153000e+02 4.483715e+07 2
## X1             9.000000e-01 4.209010e+07 2
## X2             1.000000e+00 8.960885e+07 2
## X3             5.000000e-01 2.450304e+07 2
## X4             5.000000e-01 3.358717e+07 2
## X5             5.000000e-01 2.795428e+07 2
## X6             4.000000e-01 1.984754e+07 2
## X7             1.200000e+00 6.895481e+07 2
## X8             1.000000e+00 5.535733e+07 2
## sigma          2.000000e-01 3.310300e+03 2
## mean_PPD       8.000000e-01 5.250000e+01 2
## log-posterior 7.388089e+12 4.910890e+04 2
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a c
rude measure of effective sample size, and Rhat is the potential scale
reduction factor on split chains (at convergence Rhat=1).
```

```
bayes3<- stan_glm(diff~.,data=df,prior = normal(location = 0, scale = 1
0, autoscale = FALSE) )
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.064 seconds (Warm-up)
## Chain 1:                0.101 seconds (Sampling)
## Chain 1:                0.165 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.072 seconds (Warm-up)
## Chain 2:                0.097 seconds (Sampling)
```

```
## Chain 2:                0.169 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 67.563 seconds (Warm-up)
## Chain 3:                0.316 seconds (Sampling)
## Chain 3:                67.879 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
```

```
## Chain 4:  Elapsed Time: 0.063 seconds (Warm-up)
## Chain 4:                0.117 seconds (Sampling)
## Chain 4:                0.18 seconds (Total)
## Chain 4:

## Warning: There were 139 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-w
armup
## to find out why this is a problem and how to eliminate them.

## Warning: There were 4 chains where the estimated Bayesian Fraction o
f Missing Information was low. See
## https://mc-stan.org/misc/warnings.html#bfmi-low

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

## Warning: The largest R-hat is 4.55, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#r-hat

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating po
sterior means and medians may be unreliable.
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating po
sterior variances and tail quantiles may be unreliable.
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#tail-ess

## Warning: Markov chains did not converge! Do not analyze results!
```

```
summary(bayes3)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      diff ~ .
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 768
##  predictors:   9
##
## Estimates:
##               mean    sd      10%     50%     90%
## (Intercept)  5607.5  7702.2 -3809.9  4305.7 17628.6
## X1              3.4    15.6   -17.1     5.9    18.9
## X2             -2.5     6.6   -13.7     0.4     3.0
## X3             -9.8     8.0   -18.3   -12.2     3.2
## X4             -4.4    12.2   -18.5    -4.9    10.8
```

```
## X5              -2.3     5.3    -7.9    -3.4     5.4
## X6               0.5    14.3   -19.2     1.3    18.5
## X7               2.5    16.1   -19.2     5.0    19.1
## X8             -10.1     5.7   -19.9    -7.1    -6.4
## sigma            1.1     0.7     0.1     1.0     2.1
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD  0.0    1.2 -1.2   0.0   1.4
##
## The mean_ppd is the sample average posterior predictive distribution
 of the outcome variable (for details see help('summary.stanreg')).
##
## MCMC diagnostics
##               mcse         Rhat         n_eff
## (Intercept)   5.444900e+03 2.513220e+08 2
## X1            1.100000e+01 3.055833e+08 2
## X2            4.600000e+00 1.939928e+08 2
## X3            5.600000e+00 2.206221e+08 2
## X4            8.700000e+00 1.782411e+08 2
## X5            3.700000e+00 1.062473e+08 2
## X6            1.010000e+01 2.989405e+08 2
## X7            1.140000e+01 4.211788e+08 2
## X8            4.000000e+00 7.816572e+07 2
## sigma         5.000000e-01 1.790955e+05 2
## mean_PPD      8.000000e-01 2.660000e+01 2
## log-posterior 1.112754e+12 3.129959e+06 2
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a c
rude measure of effective sample size, and Rhat is the potential scale
reduction factor on split chains (at convergence Rhat=1).
```

```
bayes4<- stan_glm(diff~.,data=df,prior = normal(location = 0, scale = 1
0, autoscale = FALSE) )
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
```

```
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.216 seconds (Warm-up)
## Chain 1:                0.145 seconds (Sampling)
## Chain 1:                0.361 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.655 seconds (Warm-up)
## Chain 2:                0.137 seconds (Sampling)
## Chain 2:                0.792 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
```

```
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.064 seconds (Warm-up)
## Chain 3:                0.117 seconds (Sampling)
## Chain 3:                0.181 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition wou
ld take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.106 seconds (Warm-up)
## Chain 4:                0.102 seconds (Sampling)
## Chain 4:                0.208 seconds (Total)
## Chain 4:

## Warning: There were 137 divergent transitions after warmup. See
## https://mc-stan.org/misc/warnings.html#divergent-transitions-after-w
armup
## to find out why this is a problem and how to eliminate them.

## Warning: There were 4 chains where the estimated Bayesian Fraction o
f Missing Information was low. See
## https://mc-stan.org/misc/warnings.html#bfmi-low

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

```
## Warning: The largest R-hat is 4.55, indicating chains have not mixed.
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#r-hat

## Warning: Bulk Effective Samples Size (ESS) is too low, indicating po
sterior means and medians may be unreliable.
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#bulk-ess

## Warning: Tail Effective Samples Size (ESS) is too low, indicating po
sterior variances and tail quantiles may be unreliable.
## Running the chains for more iterations may help. See
## https://mc-stan.org/misc/warnings.html#tail-ess

## Warning: Markov chains did not converge! Do not analyze results!

summary(bayes4)

##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      diff ~ .
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 768
##  predictors:   9
##
## Estimates:
##                mean    sd      10%     50%     90%
## (Intercept)  5159.1  4495.1  -834.9  5079.5 11312.2
## X1             -0.7    11.5   -17.8     1.1    12.7
## X2             -6.4     6.1   -16.7    -4.3    -0.4
## X3             -1.8    12.2   -19.6    -1.1    14.7
## X4             -1.6     2.2    -4.9    -1.4     1.1
## X5             -0.9    10.7   -16.0    -0.4    13.2
## X6             -5.1    10.5   -19.8    -4.5     8.3
## X7             -0.6     8.3   -13.4     1.1     8.5
## X8              9.2     4.9     0.8    11.9    12.2
## sigma           0.5     0.3     0.1     0.4     0.9
##
## Fit Diagnostics:
##           mean   sd   10%   50%   90%
## mean_PPD -1.6    0.3 -1.9  -1.8  -1.2
##
## The mean_ppd is the sample average posterior predictive distribution
 of the outcome variable (for details see help('summary.stanreg')).
##
## MCMC diagnostics
##                mcse         Rhat         n_eff
```

```
## (Intercept)   3.177700e+03 4.838164e+08 2
## X1            8.100000e+00 1.067856e+09 2
## X2            4.300000e+00 5.144333e+08 2
## X3            8.600000e+00 8.271686e+08 2
## X4            1.500000e+00 1.615384e+08 2
## X5            7.600000e+00 9.842748e+08 2
## X6            7.500000e+00 7.890844e+08 2
## X7            5.900000e+00 7.882981e+08 2
## X8            3.400000e+00 4.095371e+08 2
## sigma         2.000000e-01 1.133206e+06 2
## mean_PPD      2.000000e-01 1.420000e+01 2
## log-posterior 1.210984e+12 1.549200e+04 2
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a c
rude measure of effective sample size, and Rhat is the potential scale
reduction factor on split chains (at convergence Rhat=1).
```

5

```
#kfold1<-kfold(bayes,K = 10)
#kfold2<-kfold(bayes2,K=10)
#kfold3<-kfold(bayes3,K = 10)
#kfold4<-kfold(bayes4,K = 10)
#loo_compare(kfold1, kfold2, kfold3,kfold4)
```

The code above should be correct, but since I have 1.5MB for each bayes model, It is not able to run the code.

6

```
X1 = c(0.8,600.0,286.0,138.1,5,4,0.25)
X2 = c(0.67,630.0,296.0,238.1,2,6,0.5)
```