

# sandbox.r

hoener

2022-11-07

```
# Setup -----
# load packages
if(!require(pacman)){install.packages("pacman")}
p_load(tidyverse, singlm, rlist, latex2exp, glmnet)

# set seed
set.seed(321)

# import custom functions
source("../dev/toolbox.r")

# load data
load("supermarket1996.RData")

# writing format objects
mytheme <- theme_bw() + theme(legend.position = "bottom")

# Synth. Data -----
sim_arguments <- list(
  formula = y ~ 1 + x1 + x2 + x3 + x4,
  fixed = list(
    x1 = list(var_type = 'continuous', mean = 180, sd = 50),
    x2 = list(var_type = 'continuous', mean = 75, sd = 20),
    x3 = list(var_type = 'continuous', mean = -23, sd = 4),
    x4 = list(var_type = 'continuous', mean = 1, sd = 20)),
  sample_size = 1000000,
  reg_weights = c(2, 5, -0.7, 100, -23)
)

# data set
dfData <- simulate_fixed(data = NULL, sim_arguments) %>%
  generate_response(sim_arguments) %>%
  select(-c(level1_id, random_effects, error, fixed_outcome, X.Intercept.))

# data objects
mX <- dfData %>% select(-y) %>% data.matrix() %>% scale()
vY <- dfData %>% select(y) %>% data.matrix()

# Real Data -----
vY <- supermarket1996 %>% select(GROCERY_sum) %>% as.matrix()
mX <- supermarket1996 %>% select(-c("STORE", "CITY", "GROCCOUP_sum", "SHPINDX", "GROCERY_sum")) %>% as.mat
```

```

# Model -----

# parameters
dLambda <- 10
dAlpha <- 0.5
dEps <- 10e-10

vBeta_MM <- ElasticNetMM(mX, vY, dEps, dAlpha, dLambda)

print("Beta Estimate")

## [1] "Beta Estimate"

print(vBeta_MM)

```

```

##          GROCERY_sum
## ZIP          -13308.8561
## AGE9          36491.7742
## AGE60         15899.2921
## ETHNIC        -8985.4173
## EDUC          56245.3754
## NOCAR         12396.9794
## INCOME        39518.8773
## INCSIGMA      58596.5671
## HSIZEAVG      1242.3858
## HSIZE1       -20253.9593
## HSIZE2        32106.8191
## HSIZE34       25068.0663
## HSIZE567     -26207.0852
## HH3PLUS       8256.1493
## HH4PLUS       2725.9835
## HHSINGLE      -20253.9587
## HHLARGE      -26207.0856
## WORKWOM      -39063.8507
## SINHOUSE      29664.8326
## DENSITY      -17254.9870
## HVAL150       70583.6651
## HVAL200       80424.5164
## HVALMEAN      64183.7327
## SINGLE       -19226.4996
## RETIRED       33708.1139
## UNEMP        46902.1949
## WRKCH5       -59062.7369
## WRKCH17      -10887.6918
## NWRKCH5       -640.4477
## NWRKCH17      55668.2785
## WRKCH        -34490.4860
## NWRKCH        26674.4721
## WRKWCH       -59237.8397
## WRKWCH        -20883.9918
## TELEPHN       3583.5593
## MORTGAGE     -39675.2231
## NWHITE       -24353.2361
## POVERTY       9710.2297

```

```

## SHPCONS      7602.5546
## SHPHURR      47890.0155
## SHPAVID     -41761.4891
## SHPKSTR     -47815.7897
## SHPUNFT      17799.3436
## SHPBIRD      27265.3385
## SHOPINDX     9993.1608

# Comparison with glmnet -----
# uses an intercept per default
model_glm <- glmnet(x = mX, y = vY, alpha = dAlpha, lambda = dLambda,
                    intercept = FALSE, standardize = FALSE)
vBeta_glm <- model_glm %>% coef() %>% as.matrix()
vBeta_glm <- vBeta_glm[-1, ]
vNameCoef <- names(vBeta_glm)

dfCompareBetaTable <- CompareEstimates(vBeta_glm, vBeta_MM)

plot_coef_rmse <- dfCompareBetaTable %>% mutate(MAPE = MAPE(GLMNET, MM)) %>%
  ggplot(aes(x = Predictor, y = MAPE)) +
  geom_bar(stat = "identity") +
  labs(x = "", y = "APE") +
  scale_x_discrete(breaks = vNameCoef, labels = abbreviate) +
  scale_y_continuous(labels = scales::percent) +
  mytheme +
  theme(axis.text.x = element_text(angle = 45, size = 3, vjust = 0.5))

# Development for Epsilon -----
iEpsStart <- 1
iEpsEnd <- -100
iEpsStep <- -1

# epsilon steps
vEps <- 10^seq(iEpsStart, iEpsEnd, iEpsStep)
lBeta_MM <- list()
lCompare <- list()

# estimate model for each epsilon and compare to glmnet
for (i in seq_along(vEps)) {
  vBeta_MM <- ElasticNetMM(mX, vY, vEps[i], dAlpha, dLambda)
  lCompare[[i]] <- CompareEstimates(vBeta_glm, vBeta_MM)
  lCompare[[i]]$Epsilon <- vEps[i]
}

model_glm <- glmnet(x = mX, y = vY, alpha = dAlpha, lambda = dLambda,
                    intercept = FALSE, standardize = FALSE)
vBeta_glm <- model_glm %>% coef() %>% as.matrix()
vBeta_glm <- vBeta_glm[-1, ]
vNameCoef <- names(vBeta_glm)

# transform to dataframe
dfBetaCompareEps <- list.stack(lCompare)
dfBetaCompareEps <- dfBetaCompareEps %>%

```

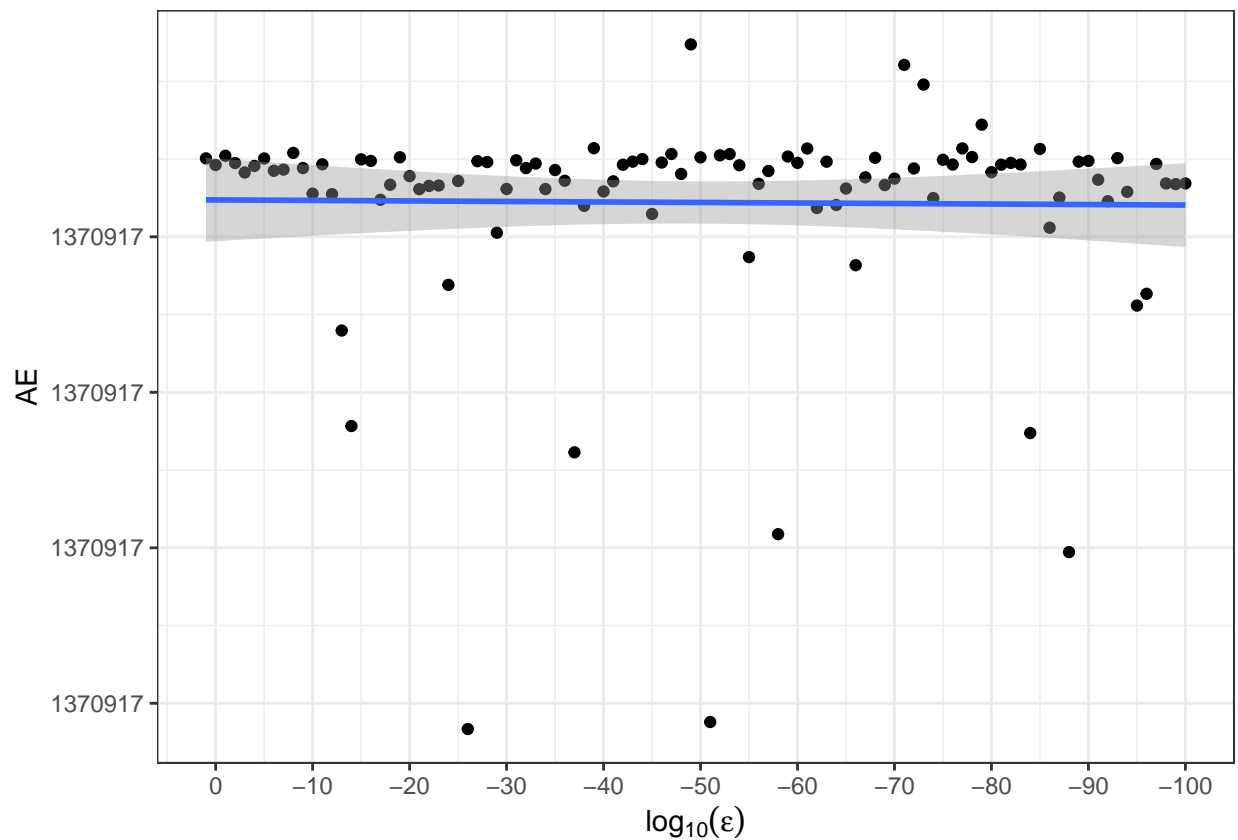
```

mutate(MAPE = MAPE(GLMNET, MM),
       MAE = MAE(GLMNET, MM)) %>%
as_tibble()

plot_MAPE_eps <- dfBetaCompareEps %>%
  group_by(Epsilon) %>%
  summarise(MAPE = mean(MAPE),
            MAE = median(MAE)) %>%
  ggplot(aes(x = log10(Epsilon), y = MAE)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_x_continuous(trans = "reverse", breaks = seq(iEpsStart - 1, iEpsEnd, -10)) +
  labs(x = TeX("$log_{10} (\\epsilon)$"), y = "AE") +
  mytheme
plot_MAPE_eps

```

```
## `geom_smooth()` using formula 'y ~ x'
```



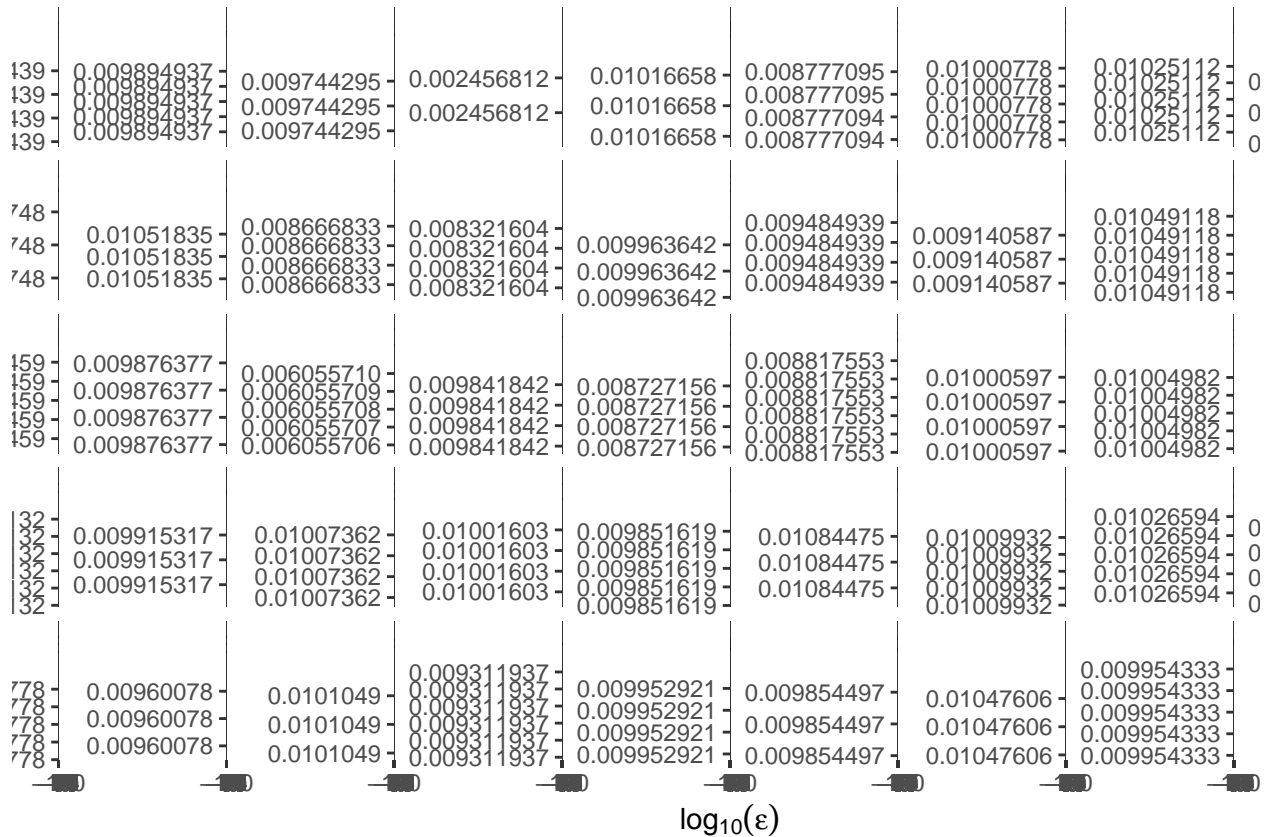
```

plot_MAPE_eps_pred <- dfBetaCompareEps %>%
  ggplot(aes(x = log10(Epsilon), y = MAPE, group = Predictor)) +
  facet_wrap(~Predictor, scales = "free_y", ncol = 9) +
  geom_smooth() +
  scale_x_continuous(trans = "reverse", breaks = seq(iEpsStart, iEpsEnd, iEpsStep)) +
  labs(x = TeX("$log_{10} (\\epsilon)$")) +
  theme(axis.text.y = element_blank()) +
  mytheme

```

```
plot_MAPE_eps_pred
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

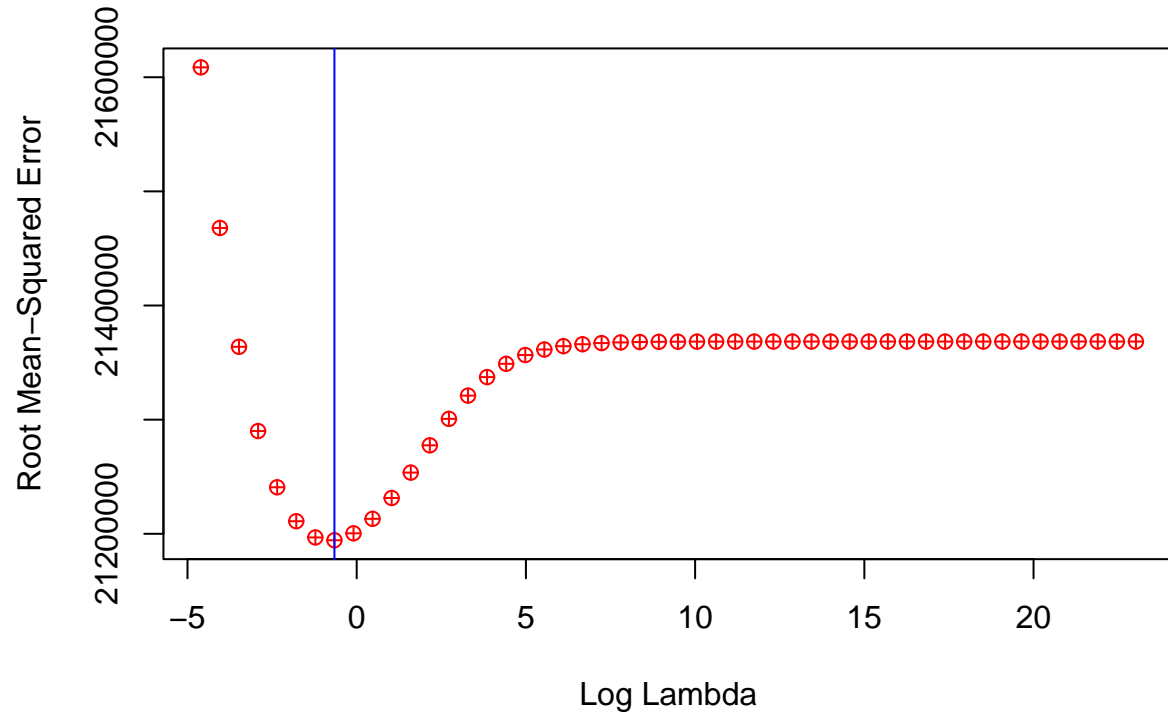


```
load("supermarket1996.Rdata")
df <- data.frame(supermarket1996)
sub_df <- subset(df, select=-c(STORE, CITY, ZIP, GROCCOUP_sum, SHPINDX))
vy <- as.vector(sub_df$GROCERY_sum)      # y variable
mX <- as.matrix(sub_df[, -1])
dEps = 10^(-10)
vBeta= rep(1, ncol(mX))
lAlpha = seq(0, 1, length.out = 50)
lLambda = 10^seq(-2, 10, length.out = 50)
nfolds = 10

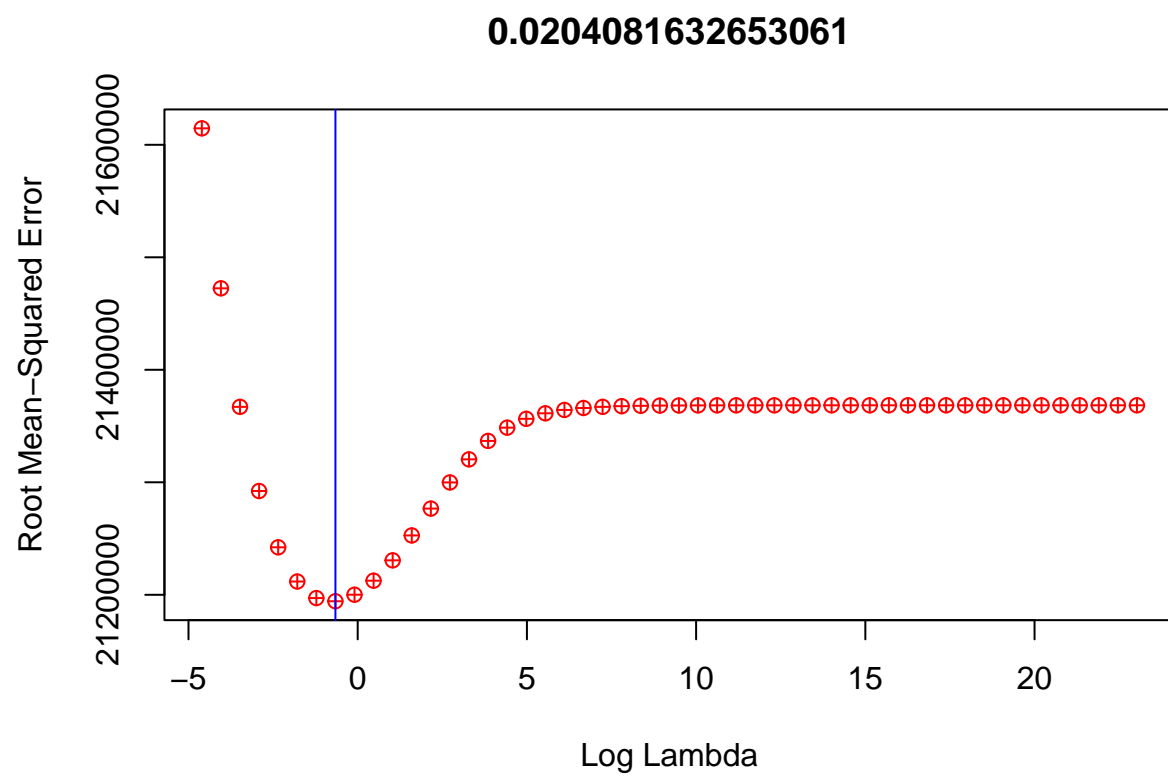
dAlpha = k_fold_plots(mX,vy,nfolds,vBeta,dEps, lAlpha,lLambda)

## Alpha is: 0 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194346
```

0

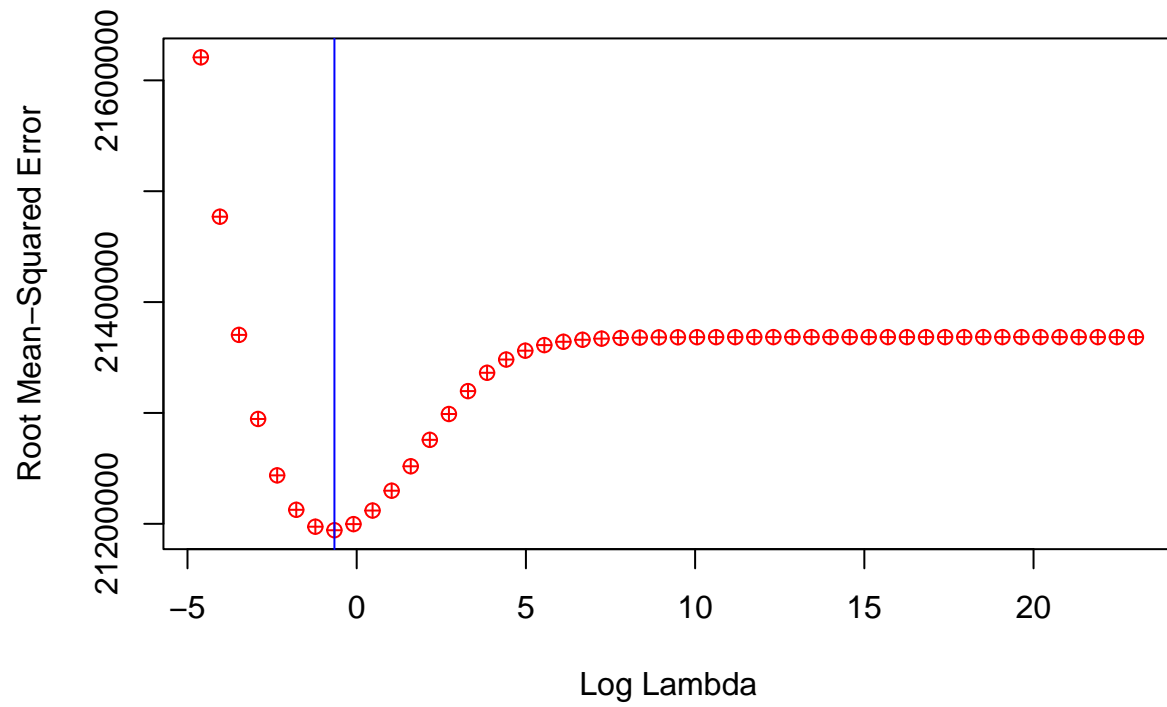


```
## Alpha is: 0.02040816 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194268
```



```
## Alpha is: 0.04081633 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194201
```

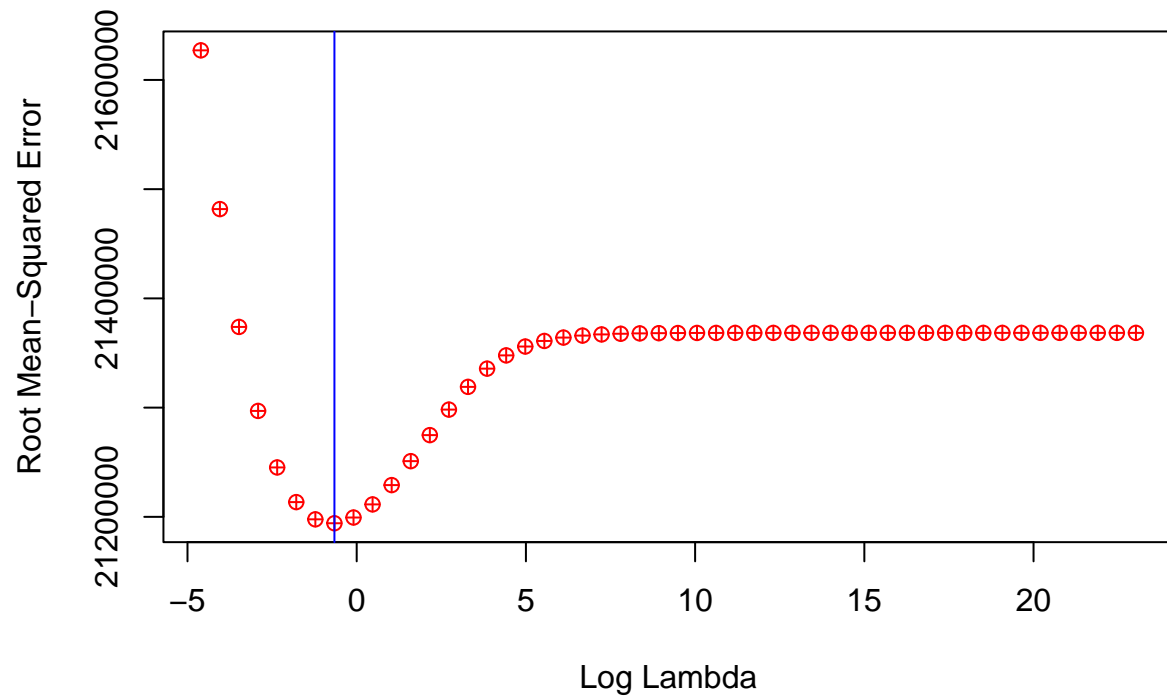
**0.0408163265306122**



```
## Alpha is: 0.06122449 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194145
```

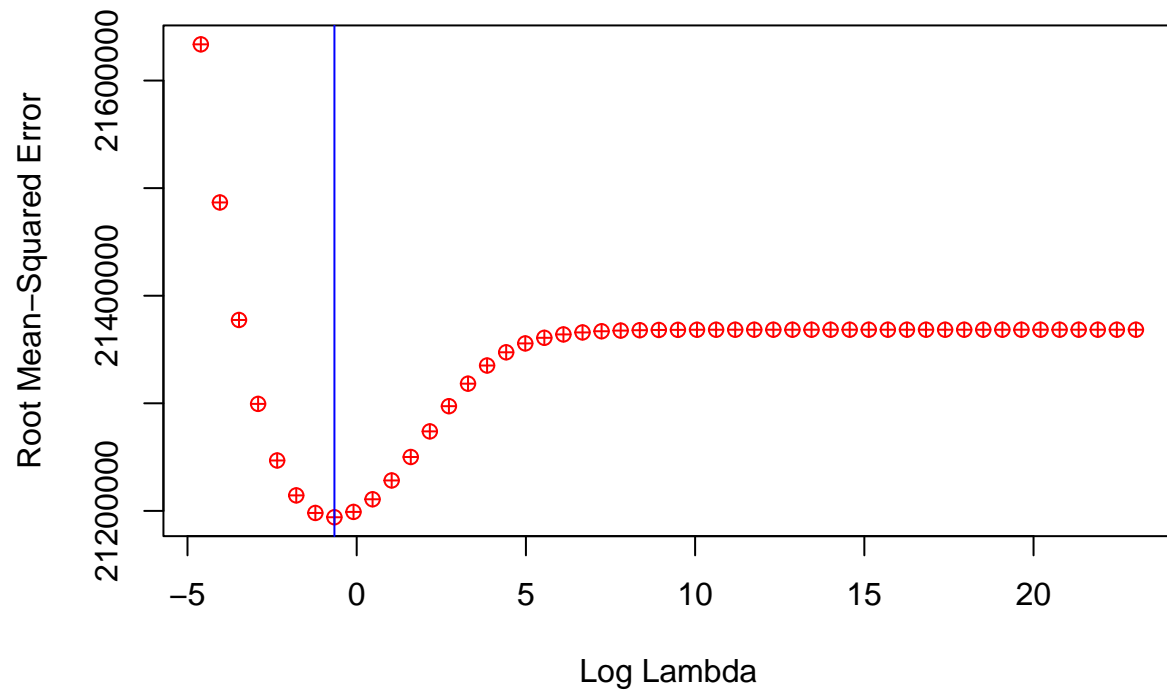


**0.0612244897959184**



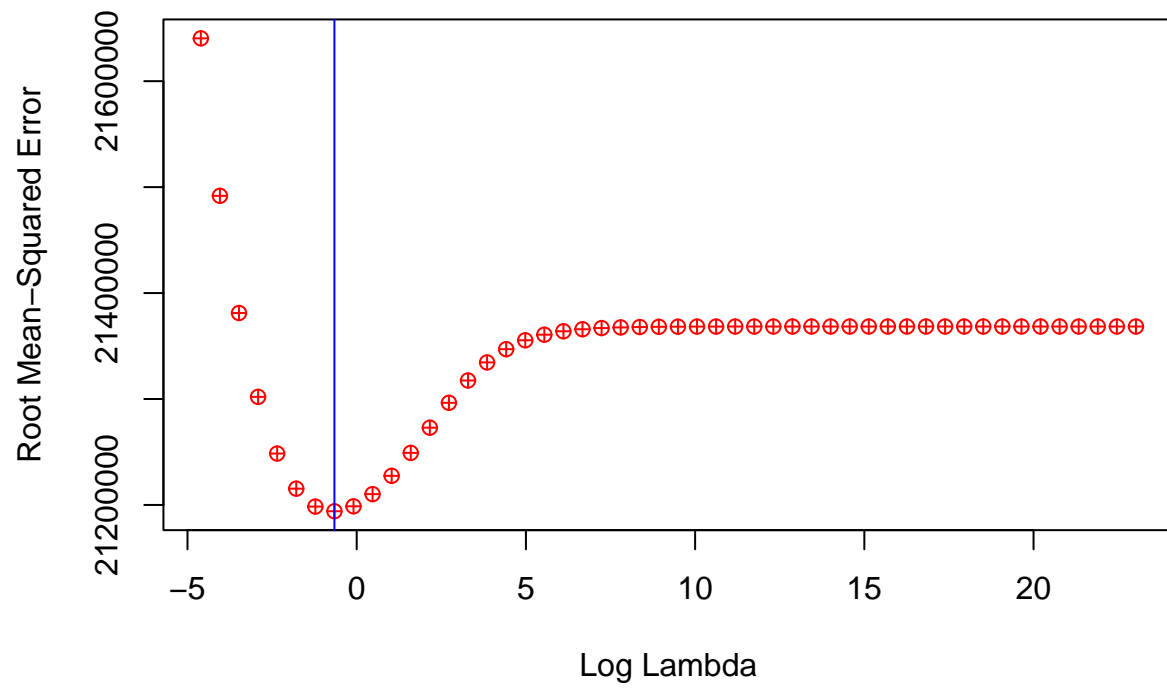
```
## Alpha is: 0.08163265 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194100
```

**0.0816326530612245**



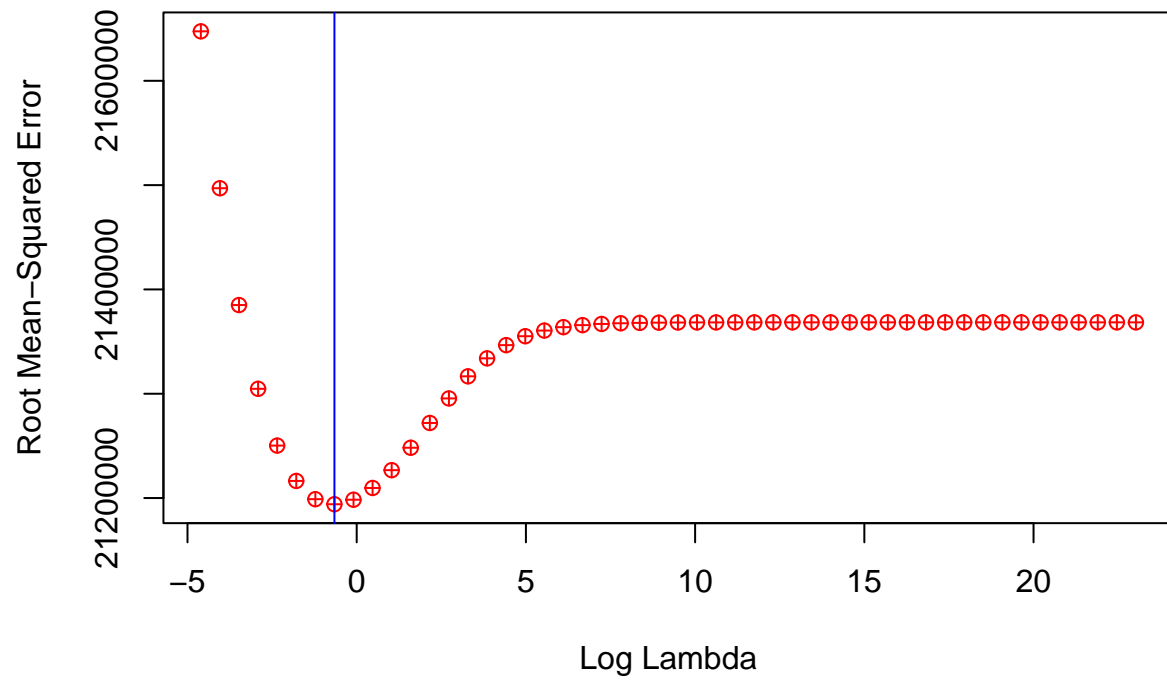
```
## Alpha is: 0.1020408 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194068
```

**0.102040816326531**



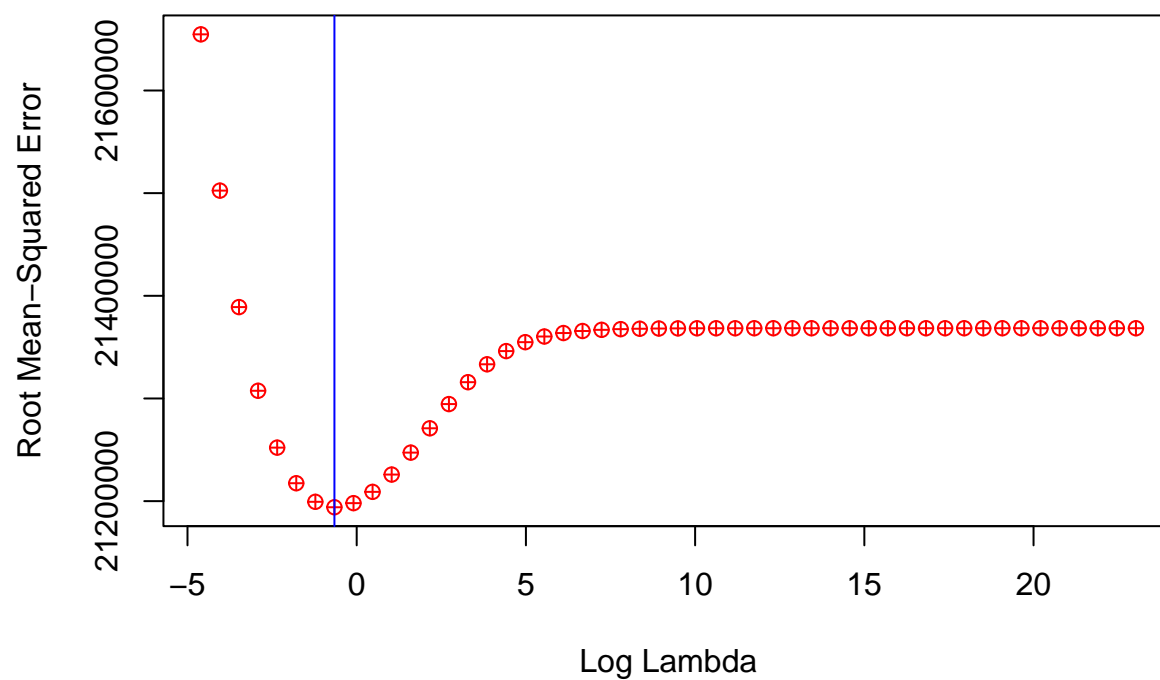
```
## Alpha is: 0.122449 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194050
```

**0.122448979591837**



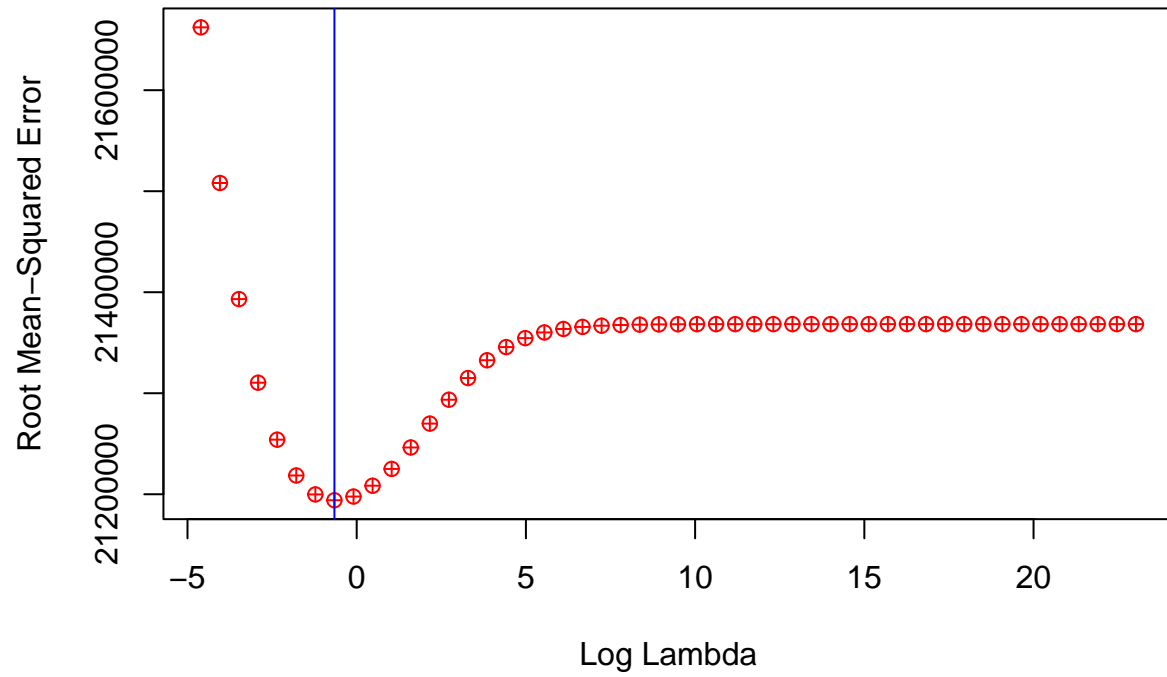
```
## Alpha is: 0.1428571 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194048
```

**0.142857142857143**



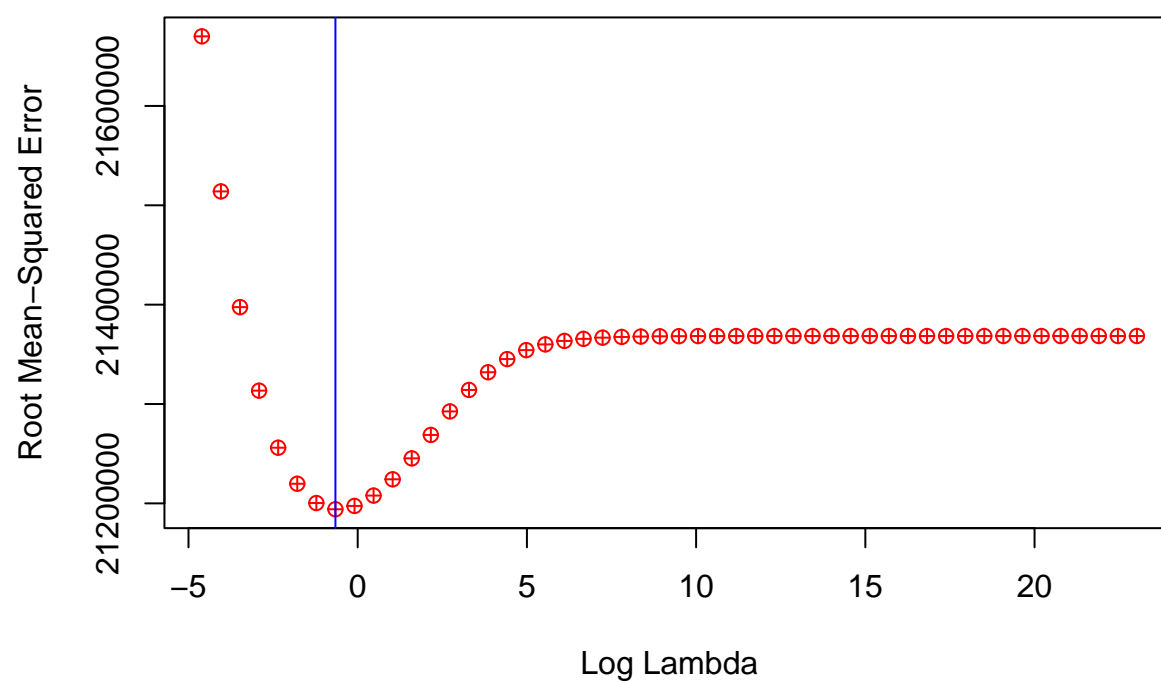
```
## Alpha is: 0.1632653 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194061
```

**0.163265306122449**



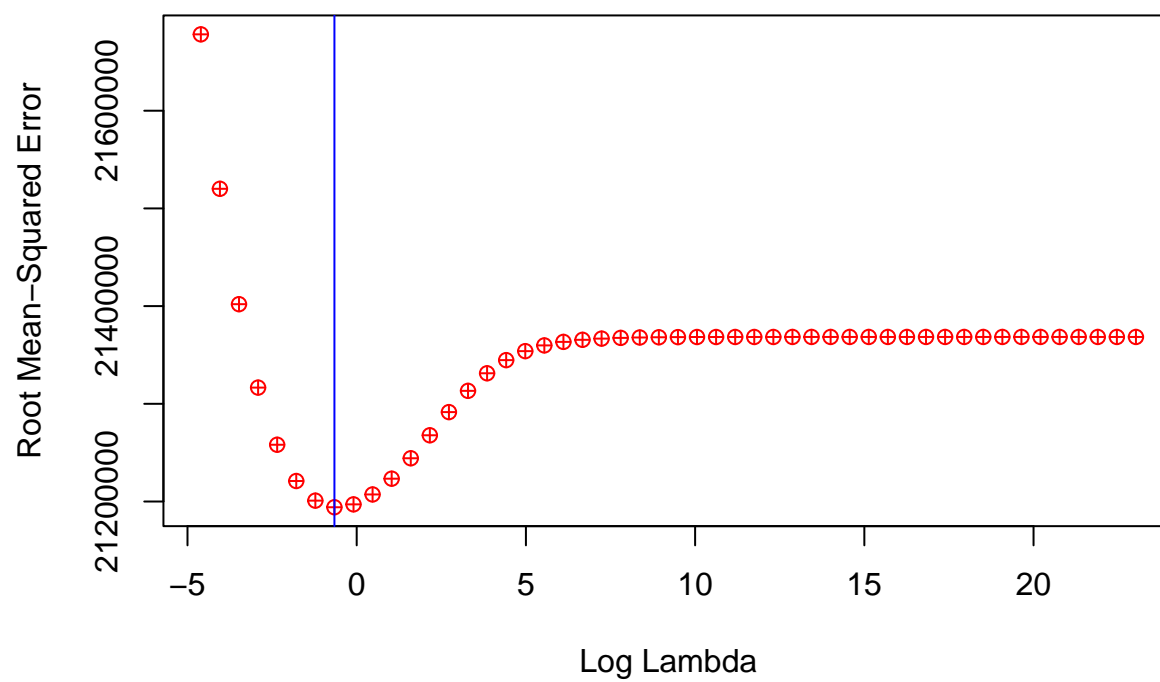
```
## Alpha is: 0.1836735 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194093
```

**0.183673469387755**



```
## Alpha is: 0.2040816 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194145
```

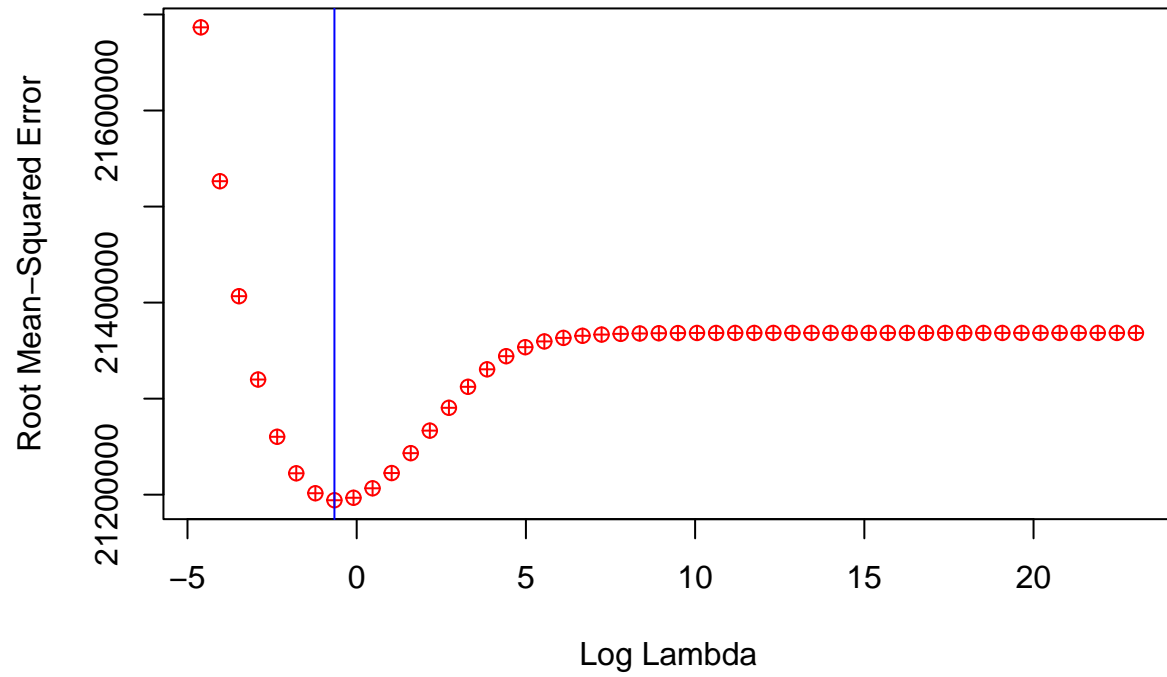
**0.204081632653061**



```
## Alpha is: 0.2244898 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194218
```

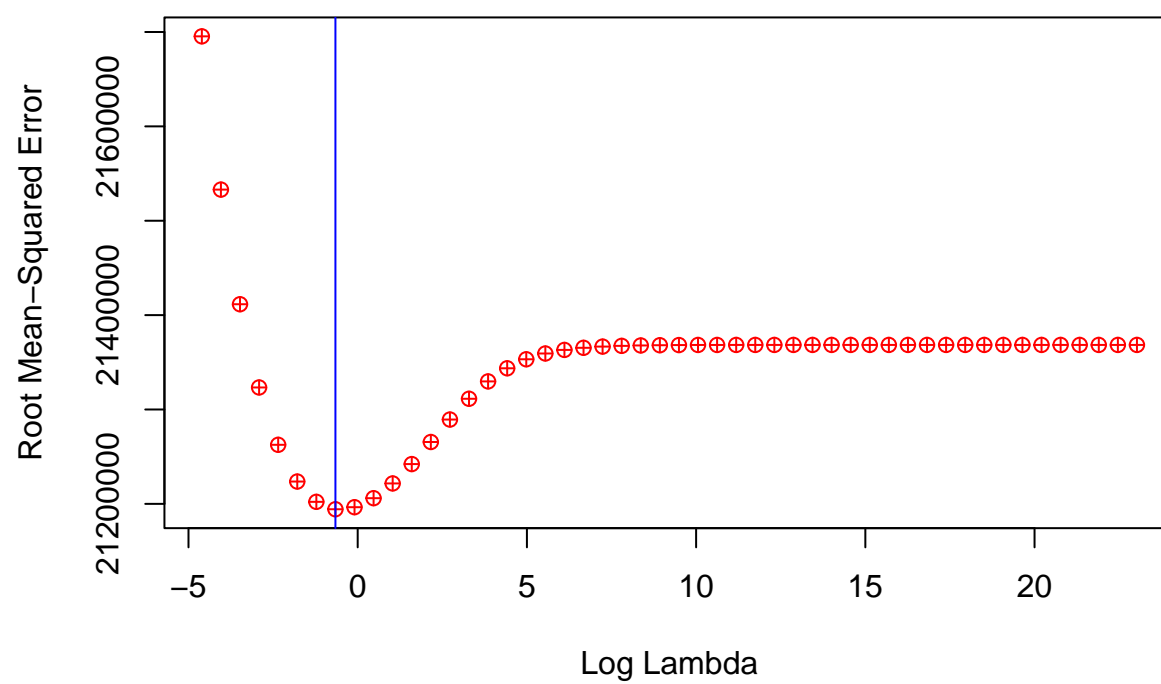


**0.224489795918367**



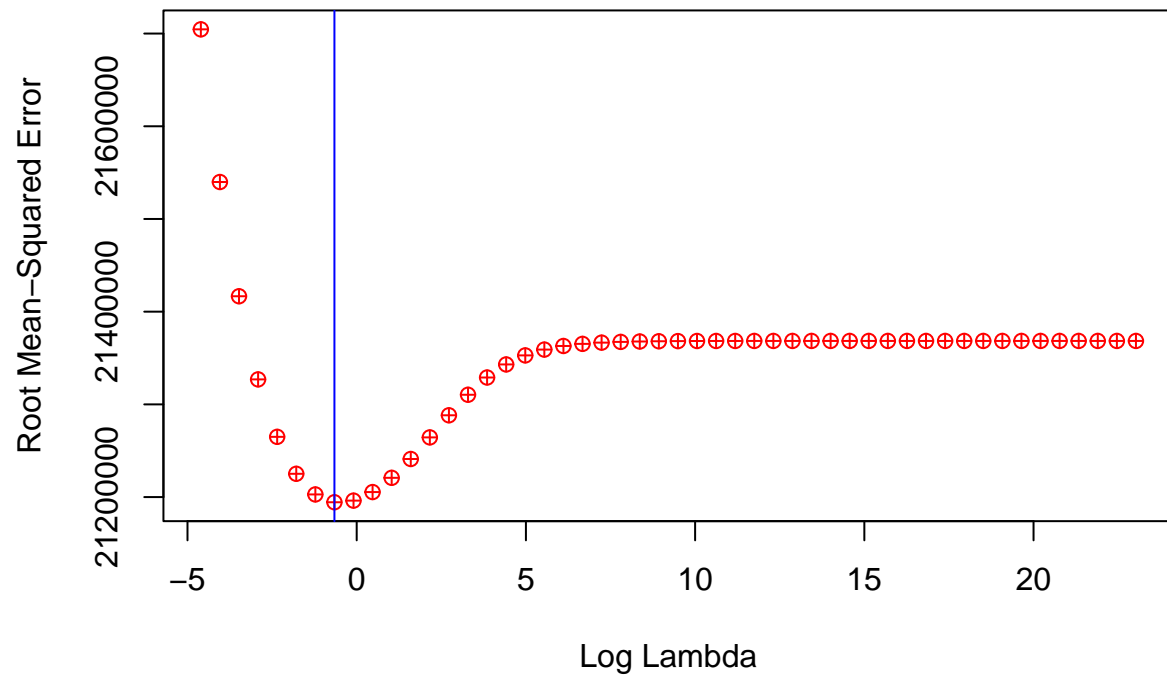
```
## Alpha is: 0.244898 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194314
```

**0.244897959183673**



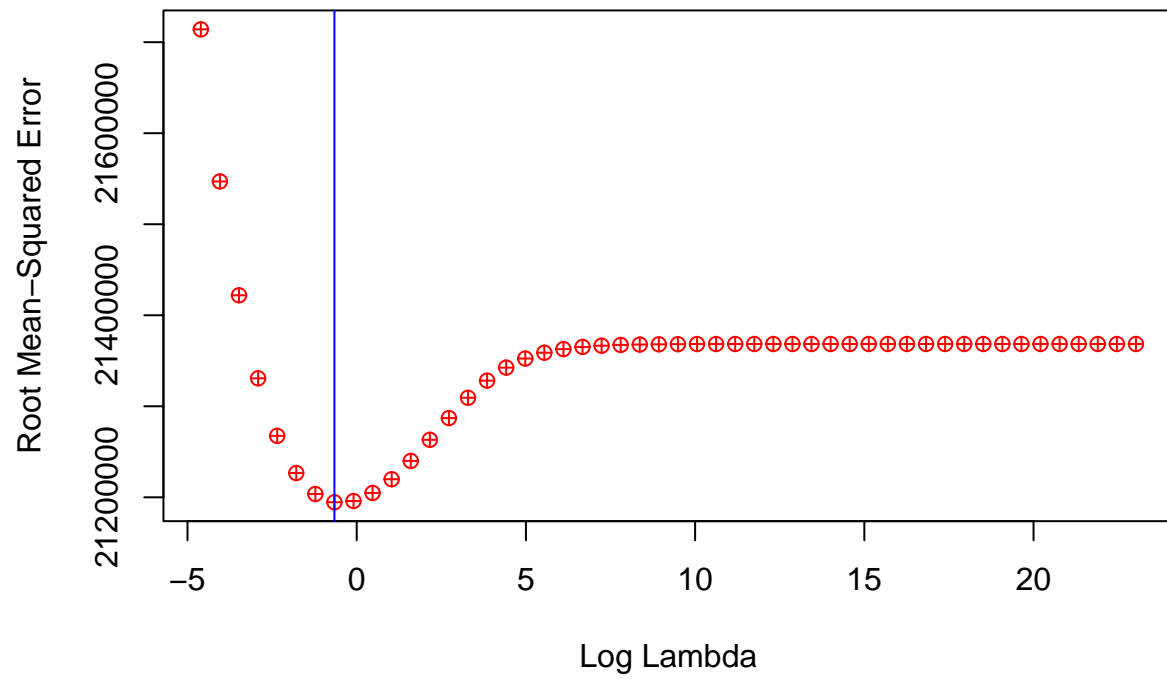
```
## Alpha is: 0.2653061 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194436
```

**0.26530612244898**



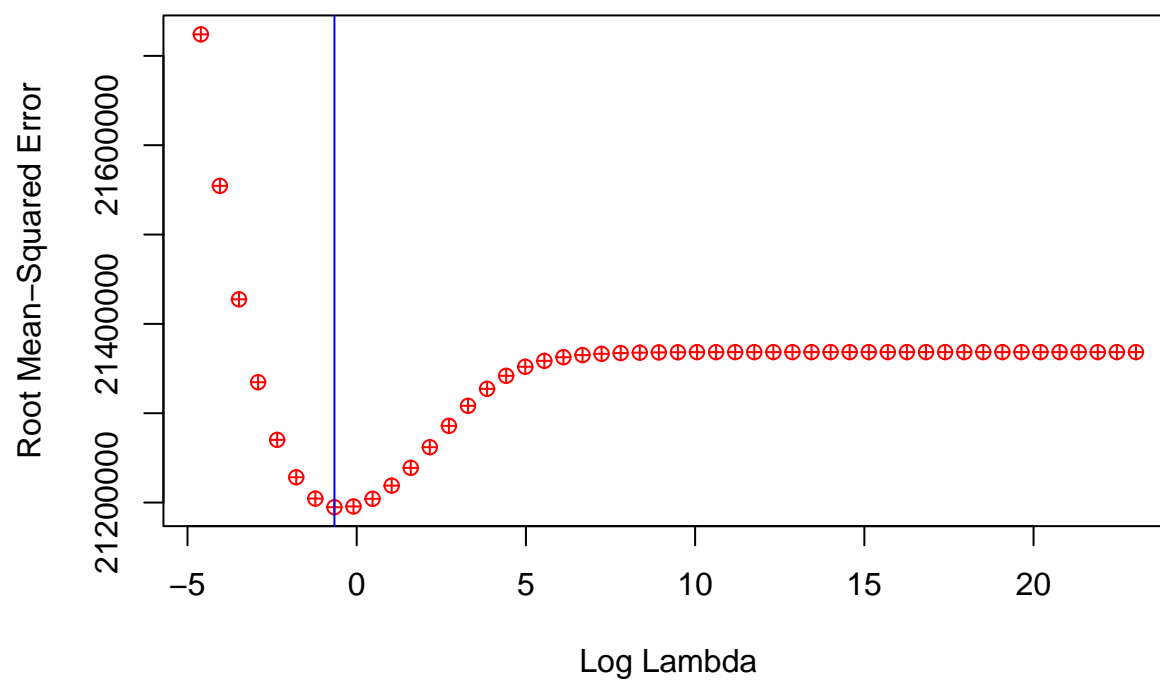
```
## Alpha is: 0.2857143 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194587
```

**0.285714285714286**



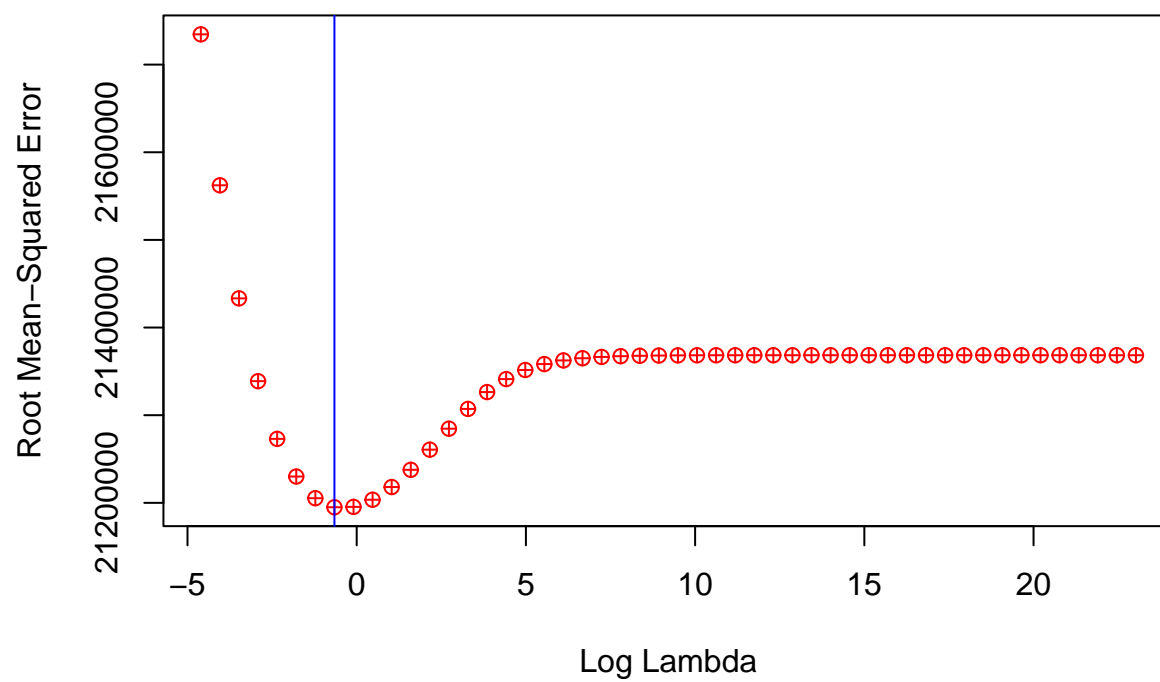
```
## Alpha is: 0.3061224 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194768
```

**0.306122448979592**



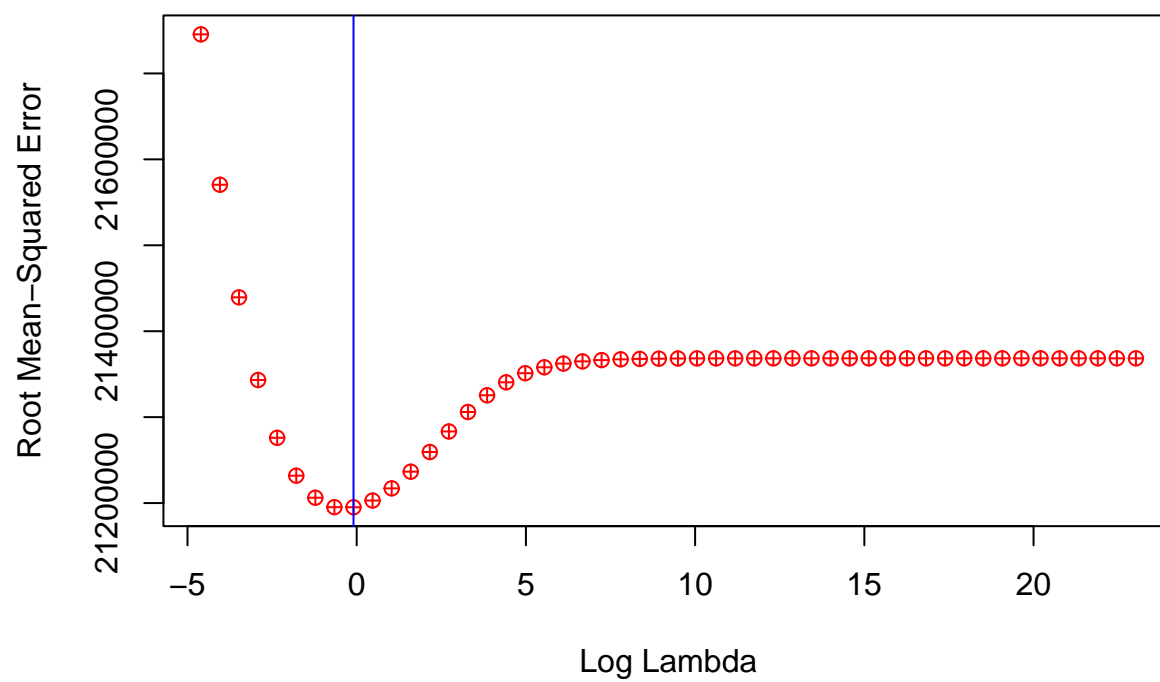
```
## Alpha is: 0.3265306 . The minimum lambda is: 0.5179475
## The minimum RMSE is: 21194984
```

**0.326530612244898**



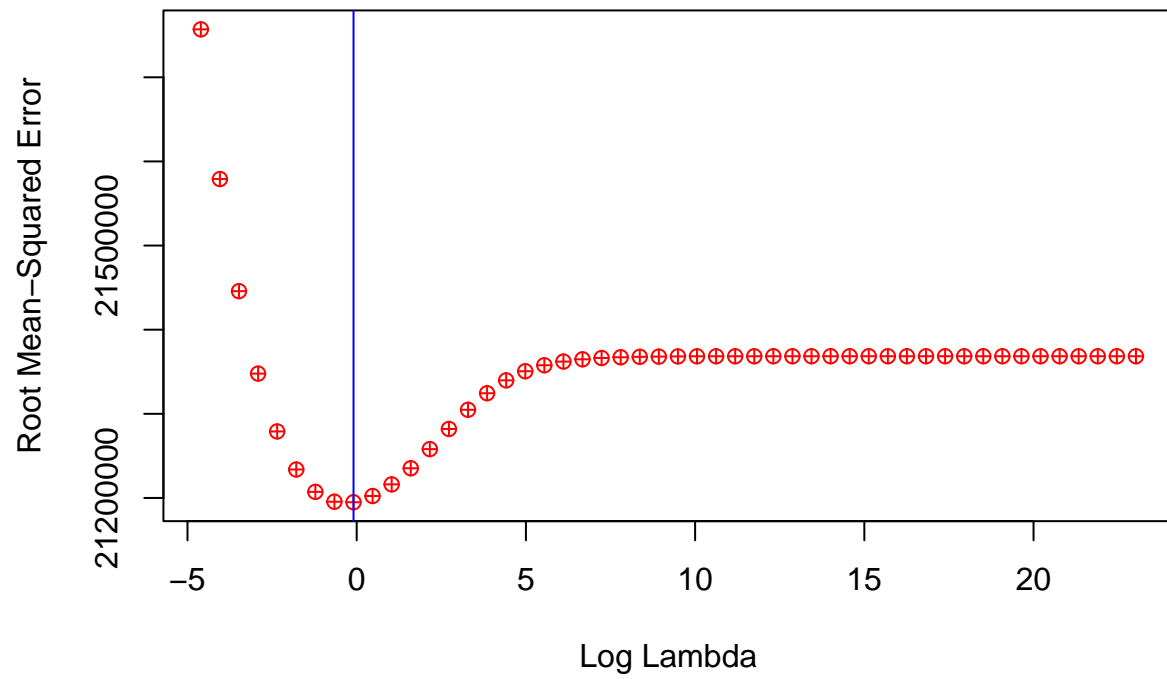
```
## Alpha is: 0.3469388 . The minimum lambda is: 0.9102982
## The minimum RMSE is: 21195144
```

**0.346938775510204**



```
## Alpha is: 0.3673469 . The minimum lambda is: 0.9102982
## The minimum RMSE is: 21194918
```

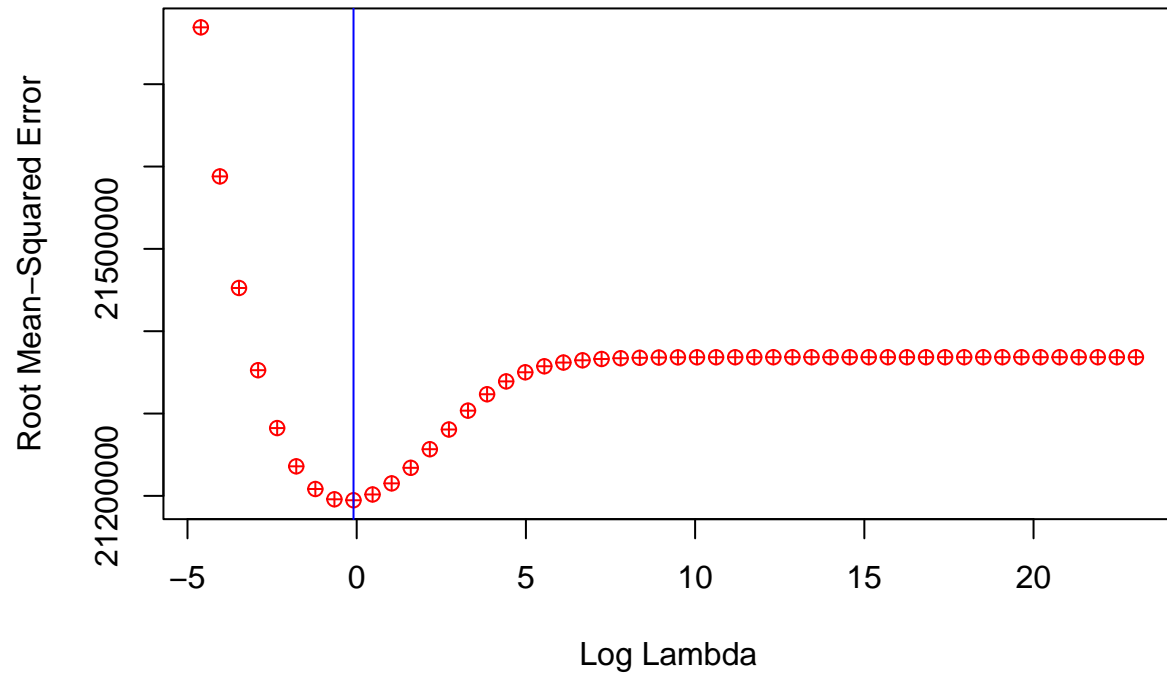
**0.36734693877551**



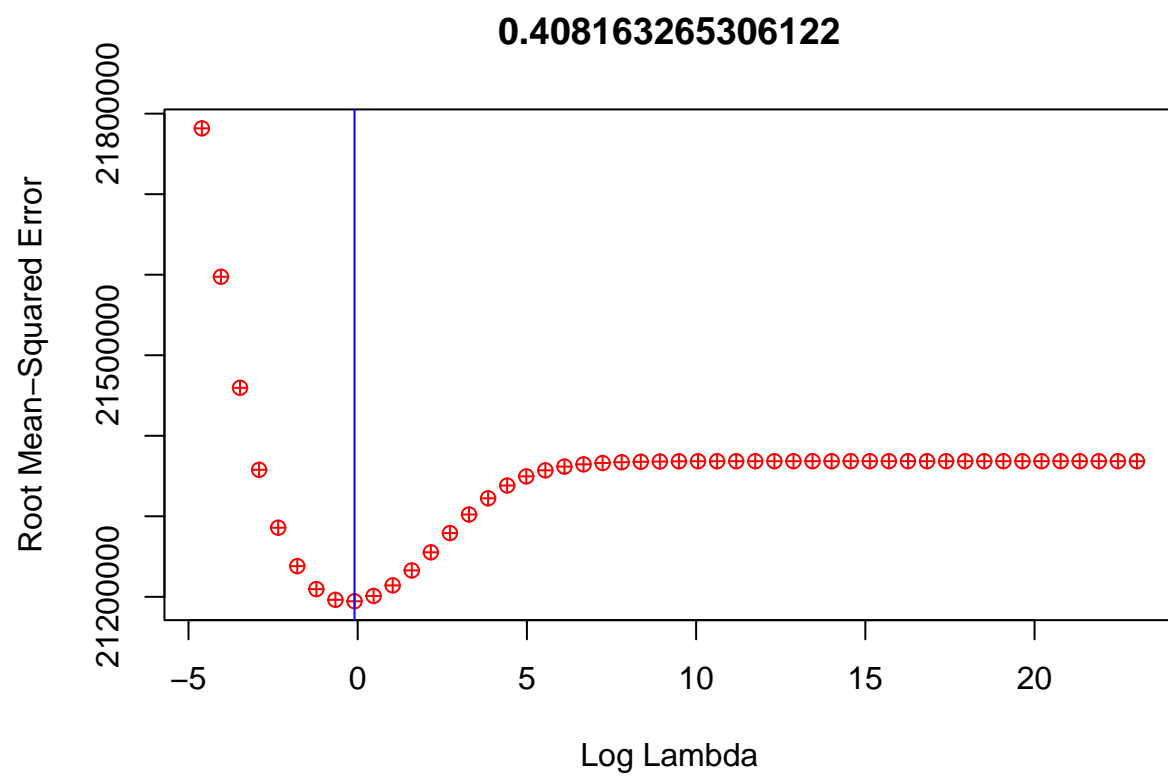
```
## Alpha is: 0.3877551 . The minimum lambda is: 0.9102982
## The minimum RMSE is: 21194711
```



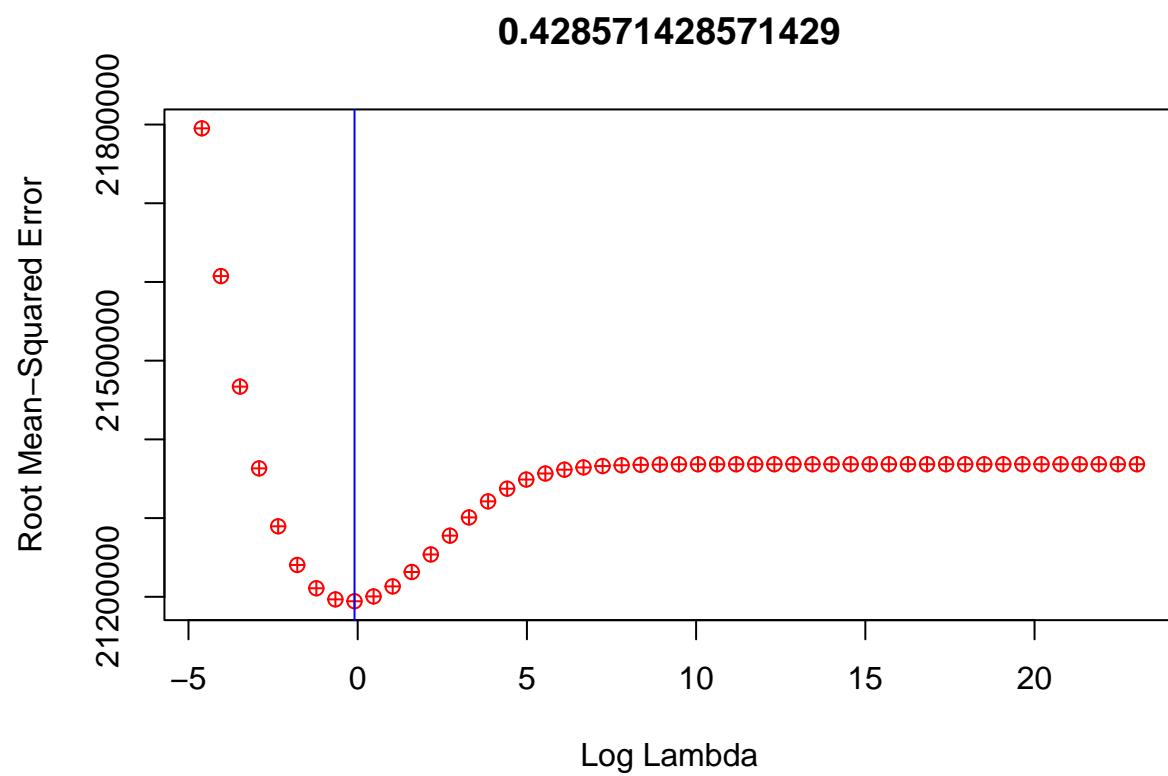
**0.387755102040816**



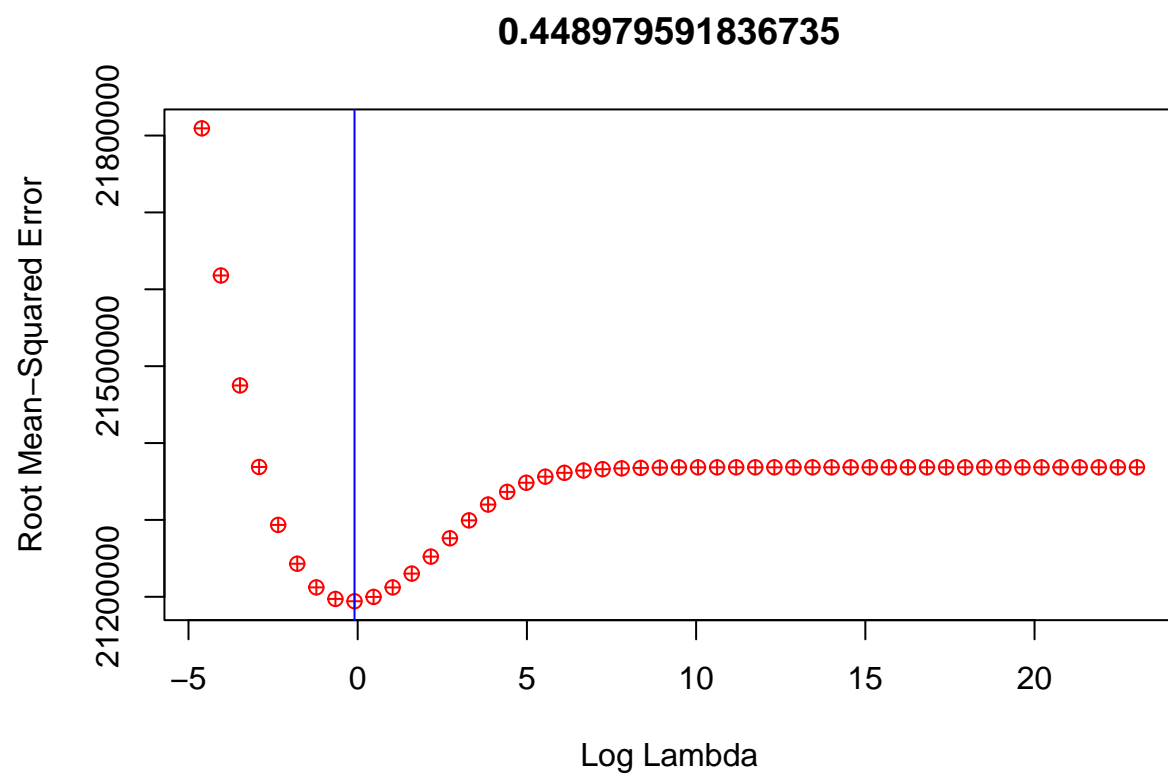
```
## Alpha is: 0.4081633 . The minimum lambda is: 0.9102982
## The minimum RMSE is: 21194525
```



```
## Alpha is: 0.4285714 . The minimum lambda is: 0.9102982
## The minimum RMSE is: 21194363
```

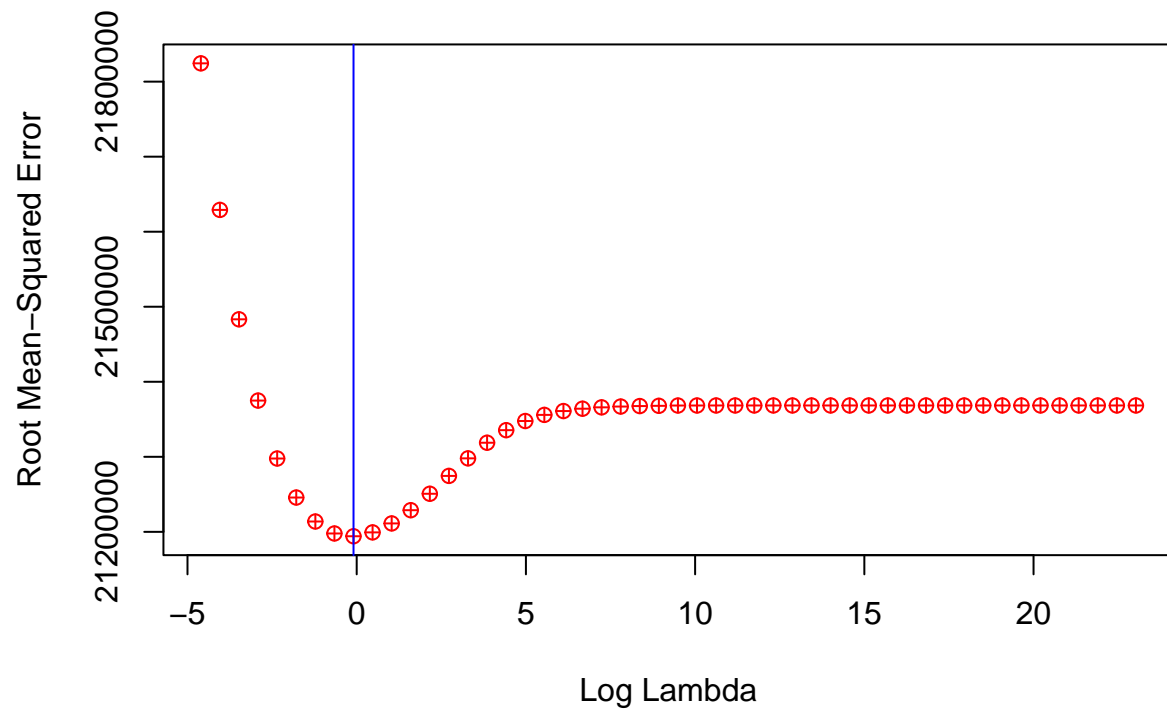


```
## Alpha is: 0.4489796 . The minimum lambda is: 0.9102982
## The minimum RMSE is: 21194230
```



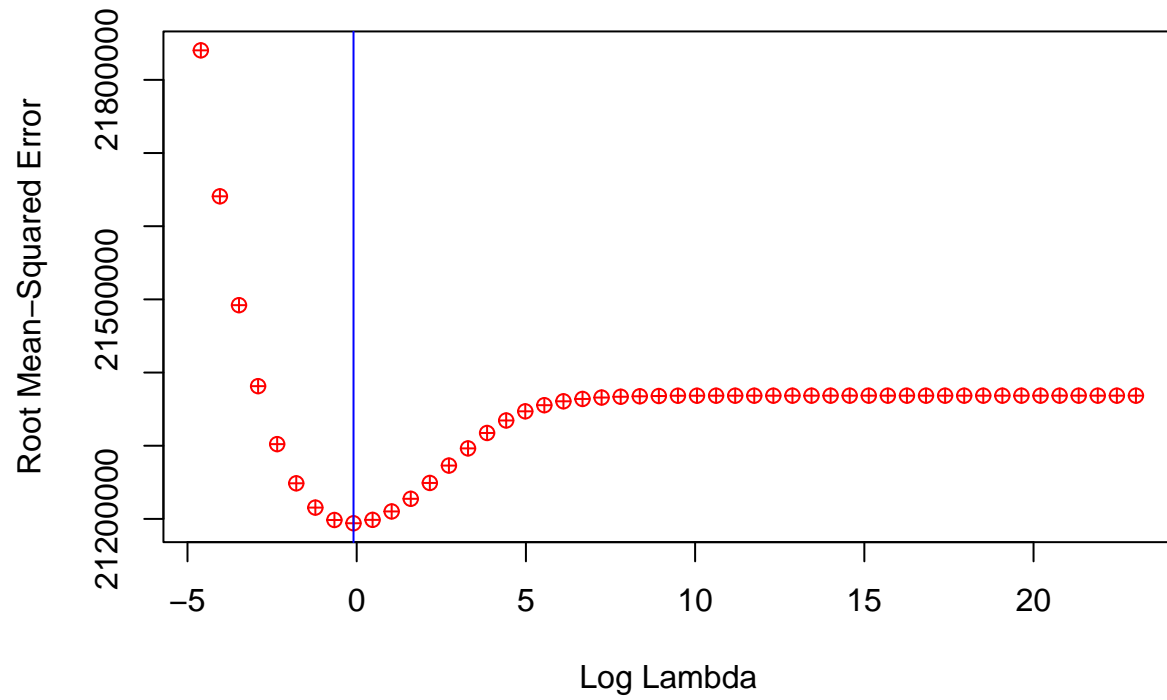
```
## Alpha is: 0.4693878 . The minimum lambda is: 0.9102982
## The minimum RMSE is: 21194130
```

**0.469387755102041**



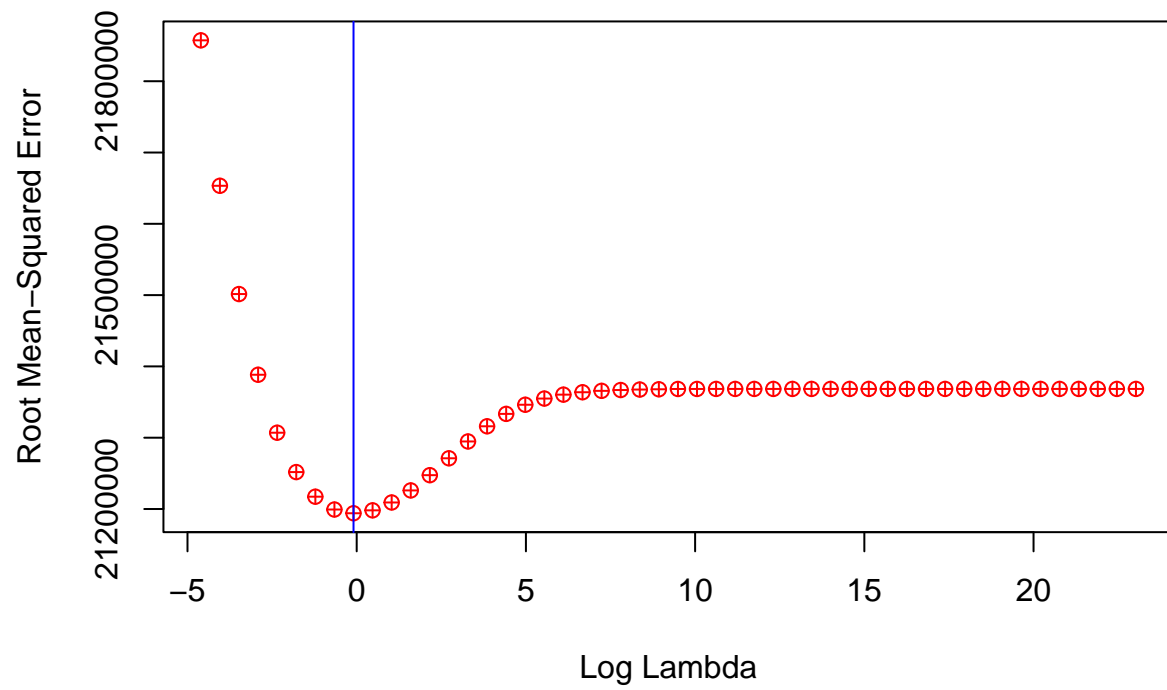
```
## Alpha is: 0.4897959 . The minimum lambda is: 0.9102982
## The minimum RMSE is: 21194067
```

**0.489795918367347**



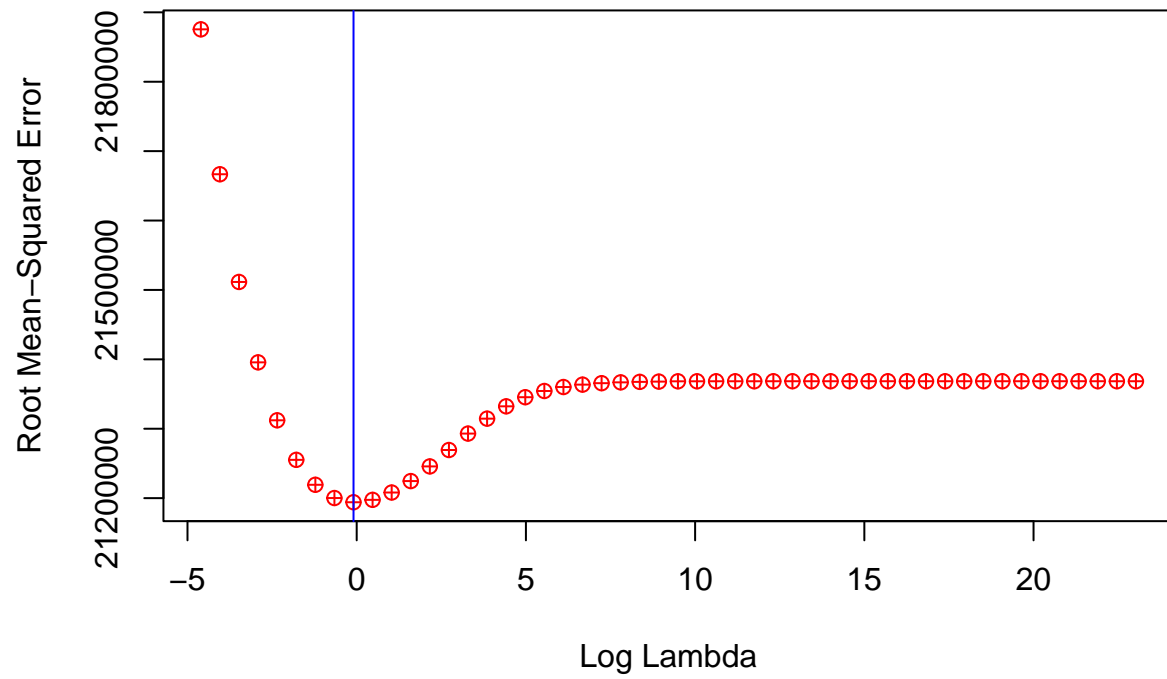
```
## Alpha is: 0.5102041 . The minimum lambda is: 0.9102982
## The minimum RMSE is: 21194047
```

**0.510204081632653**



```
## Alpha is: 0.5306122 . The minimum lambda is: 0.9102982
## The minimum RMSE is: 21194078
```

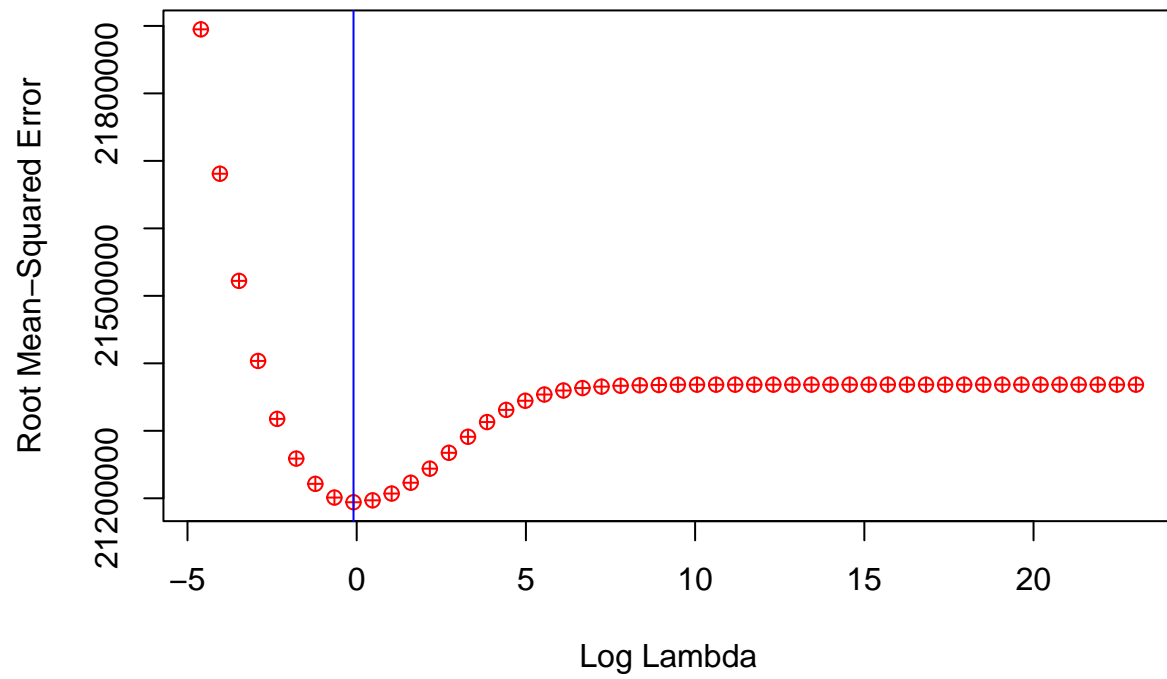
**0.530612244897959**



```
## Alpha is: 0.5510204 . The minimum lambda is: 0.9102982
## The minimum RMSE is: 21194167
```

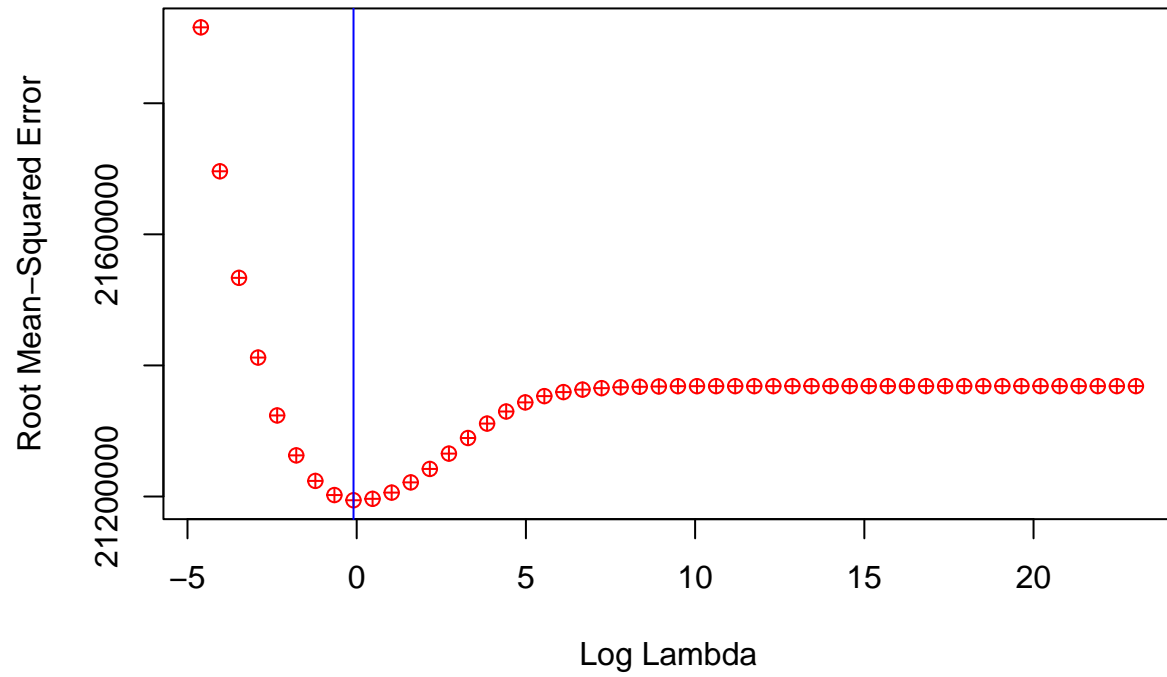


**0.551020408163265**



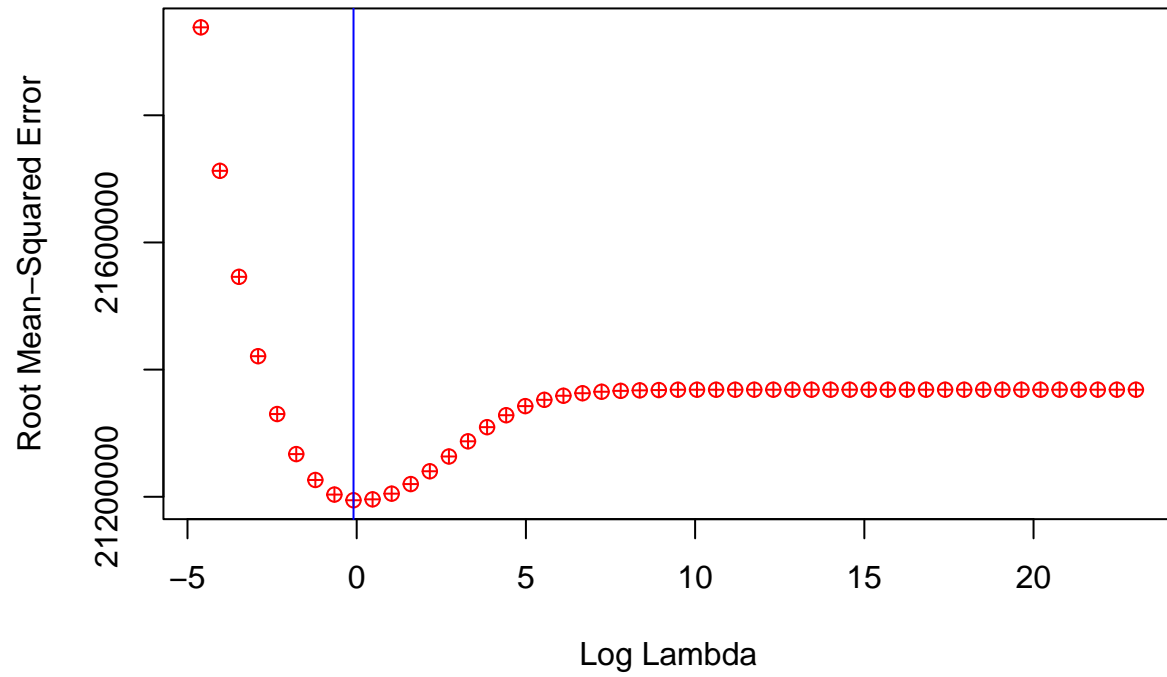
```
## Alpha is: 0.5714286 . The minimum lambda is: 0.9102982
## The minimum RMSE is: 21194324
```

**0.571428571428571**



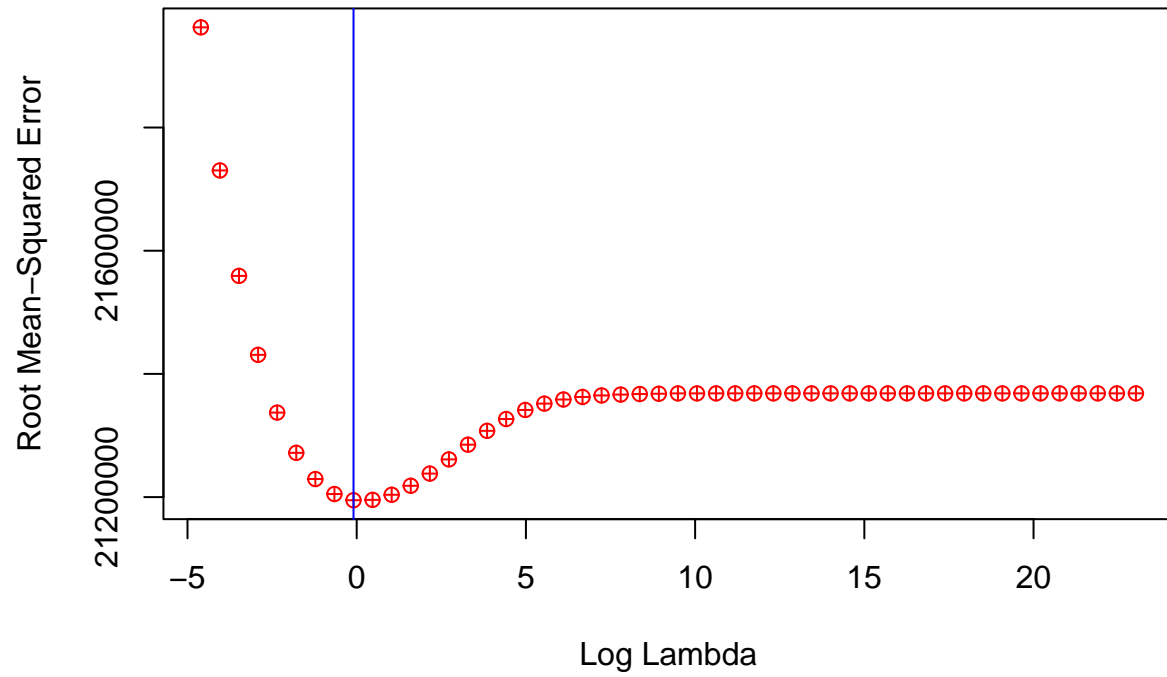
```
## Alpha is: 0.5918367 . The minimum lambda is: 0.9102982
## The minimum RMSE is: 21194562
```

**0.591836734693878**

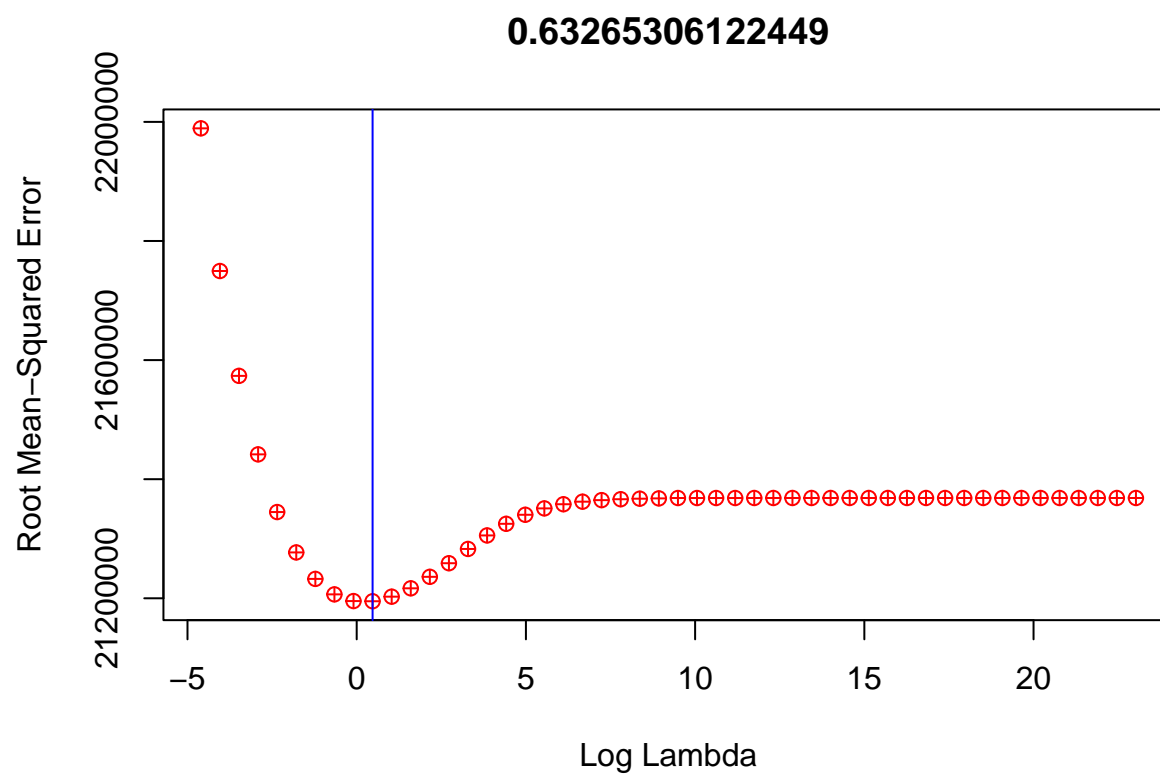


```
## Alpha is: 0.6122449 . The minimum lambda is: 0.9102982
## The minimum RMSE is: 21194895
```

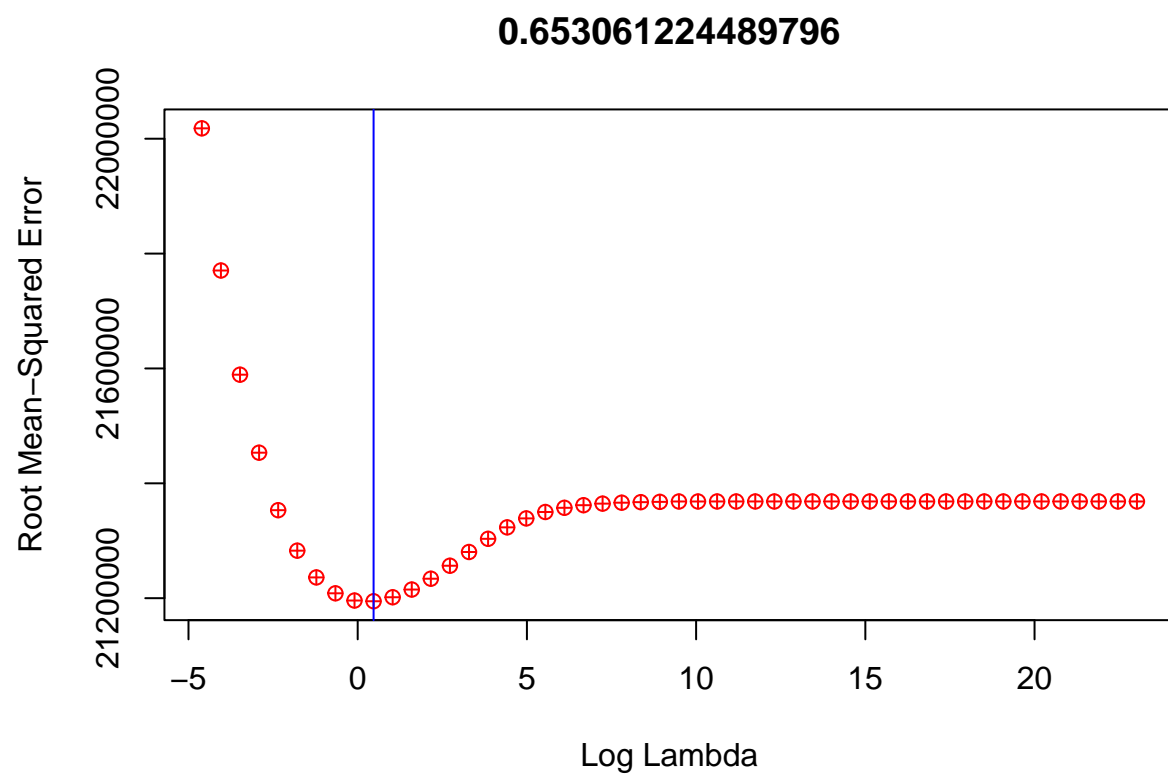
**0.612244897959184**



```
## Alpha is: 0.6326531 . The minimum lambda is: 1.599859
## The minimum RMSE is: 21195060
```

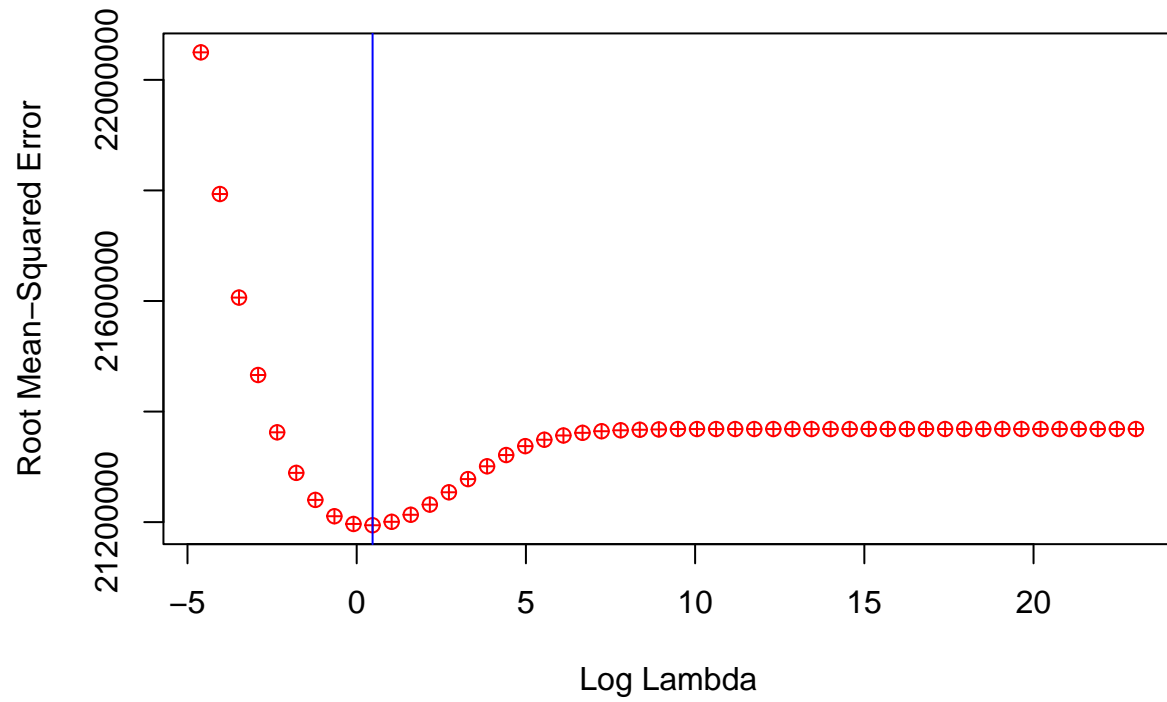


```
## Alpha is: 0.6530612 . The minimum lambda is: 1.599859
## The minimum RMSE is: 21194687
```



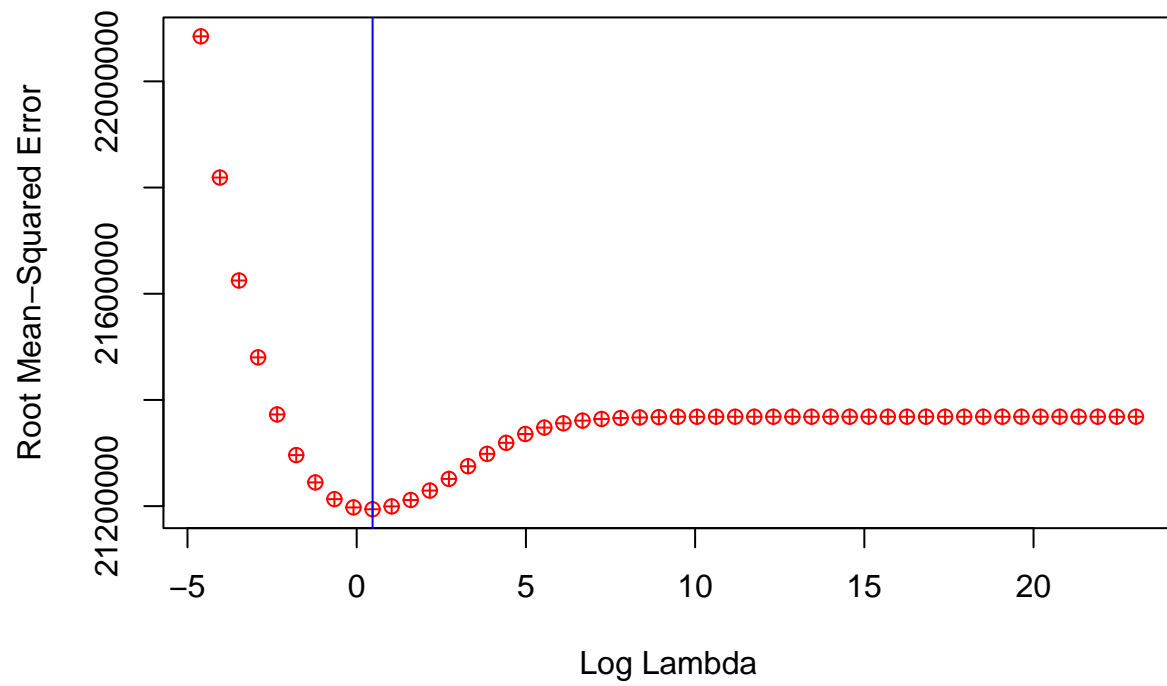
```
## Alpha is: 0.6734694 . The minimum lambda is: 1.599859
## The minimum RMSE is: 21194382
```

**0.673469387755102**



```
## Alpha is: 0.6938776 . The minimum lambda is: 1.599859
## The minimum RMSE is: 21194163
```

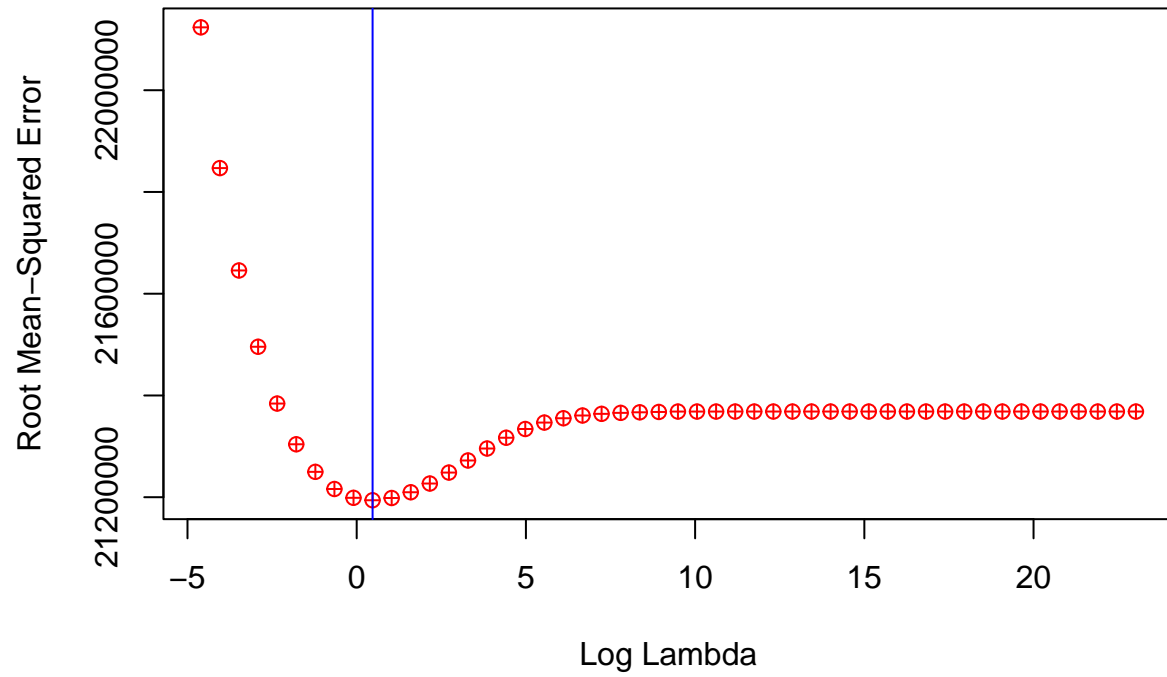
**0.693877551020408**



```
## Alpha is: 0.7142857 . The minimum lambda is: 1.599859
## The minimum RMSE is: 21194054
```

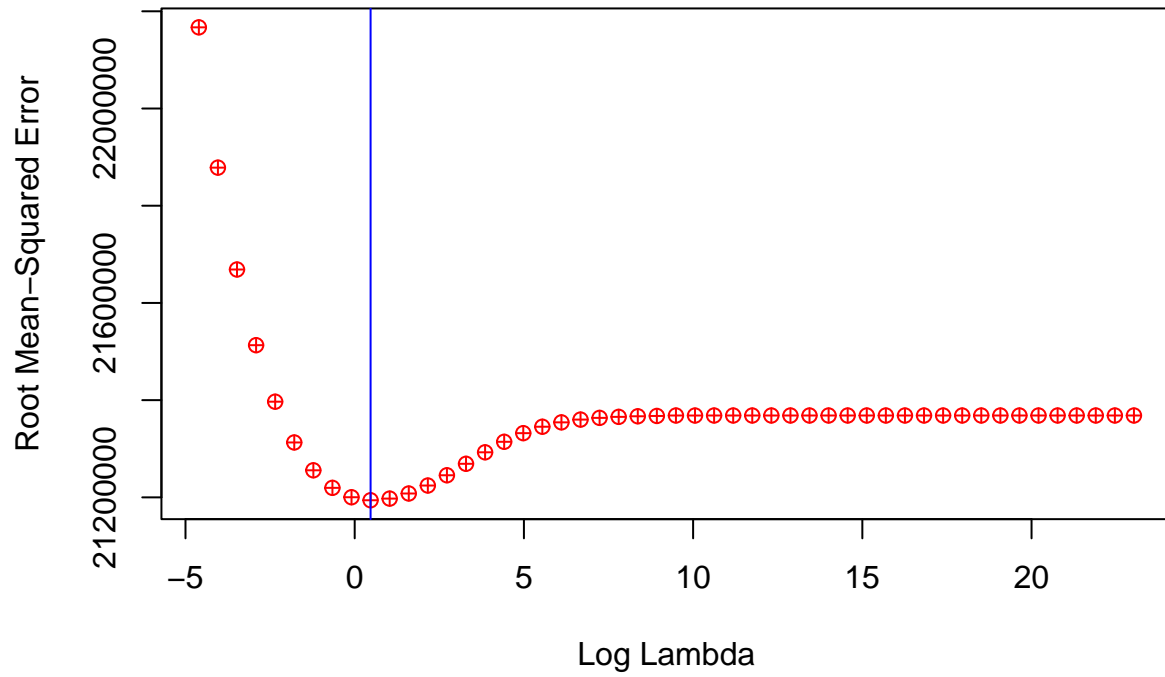


**0.714285714285714**



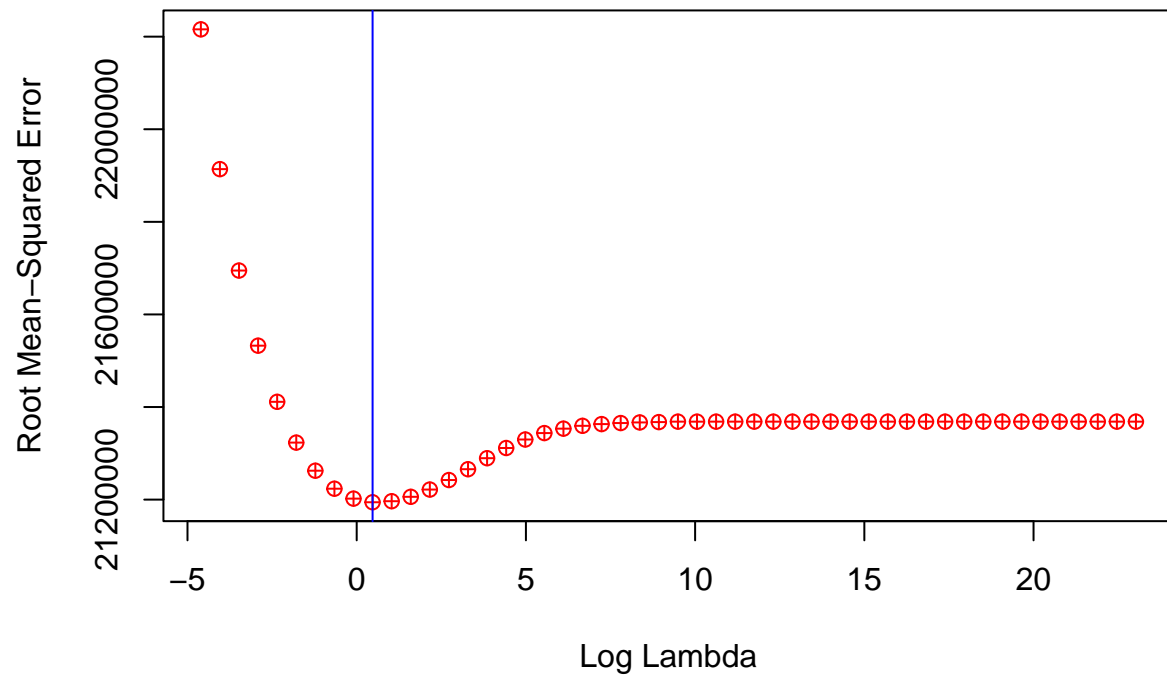
```
## Alpha is: 0.7346939 . The minimum lambda is: 1.599859
## The minimum RMSE is: 21194087
```

**0.73469387755102**



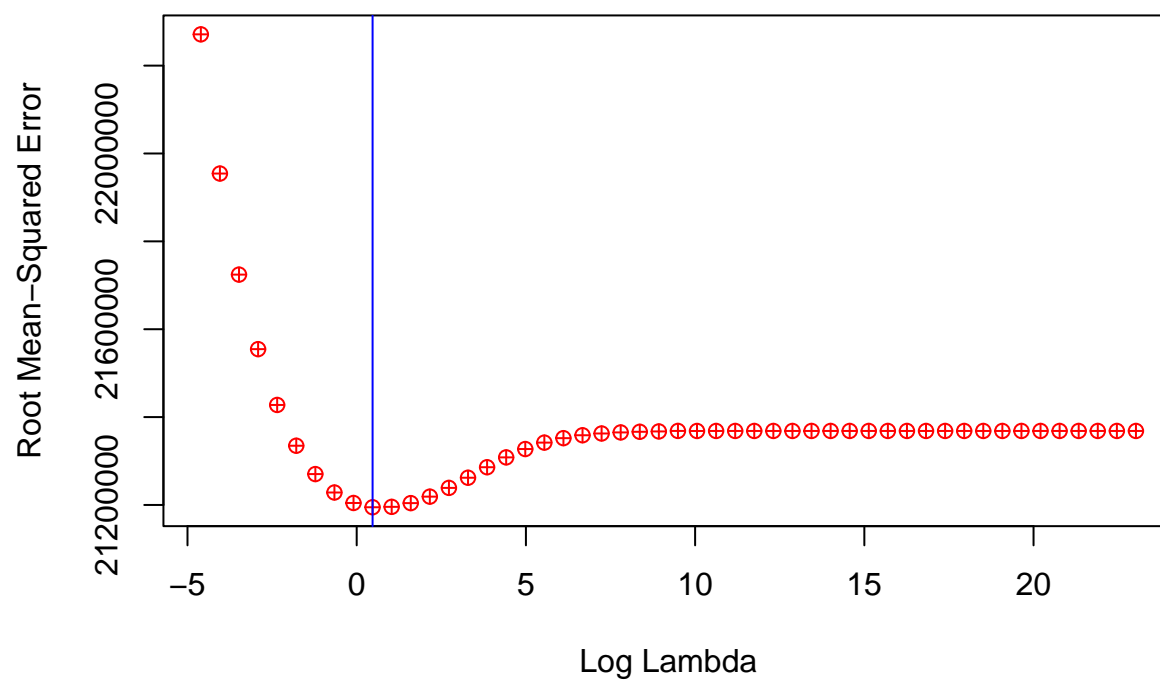
```
## Alpha is: 0.755102 . The minimum lambda is: 1.599859
## The minimum RMSE is: 21194307
```

**0.755102040816326**



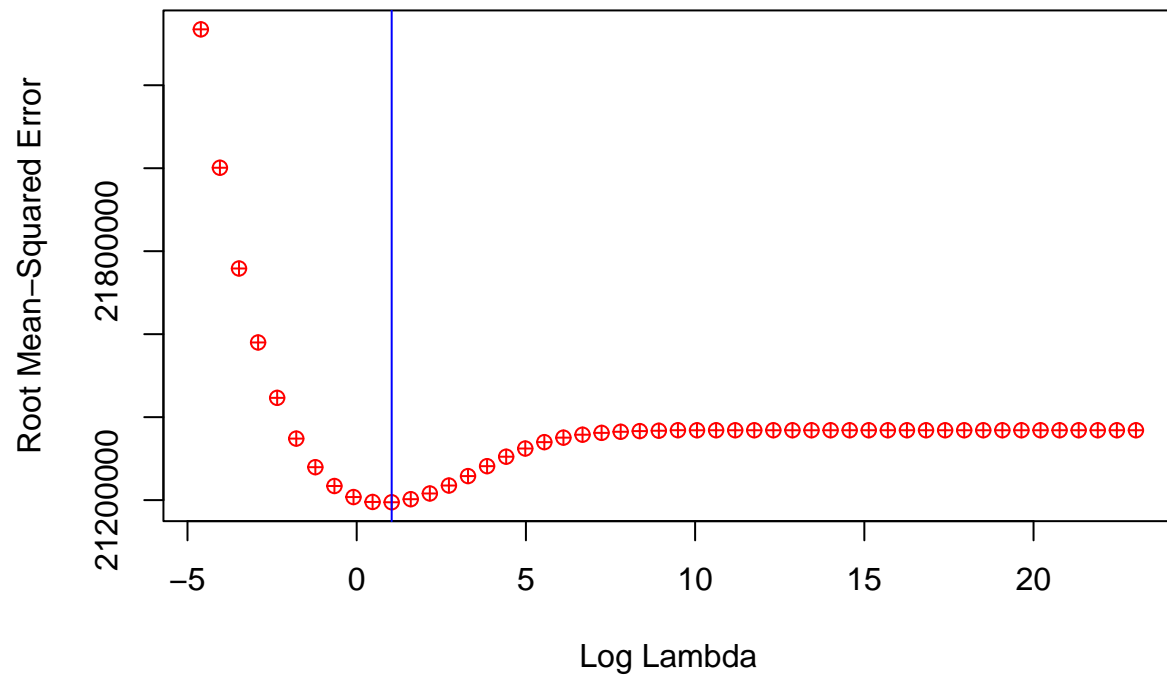
```
## Alpha is: 0.7755102 . The minimum lambda is: 1.599859
## The minimum RMSE is: 21194773
```

**0.775510204081633**

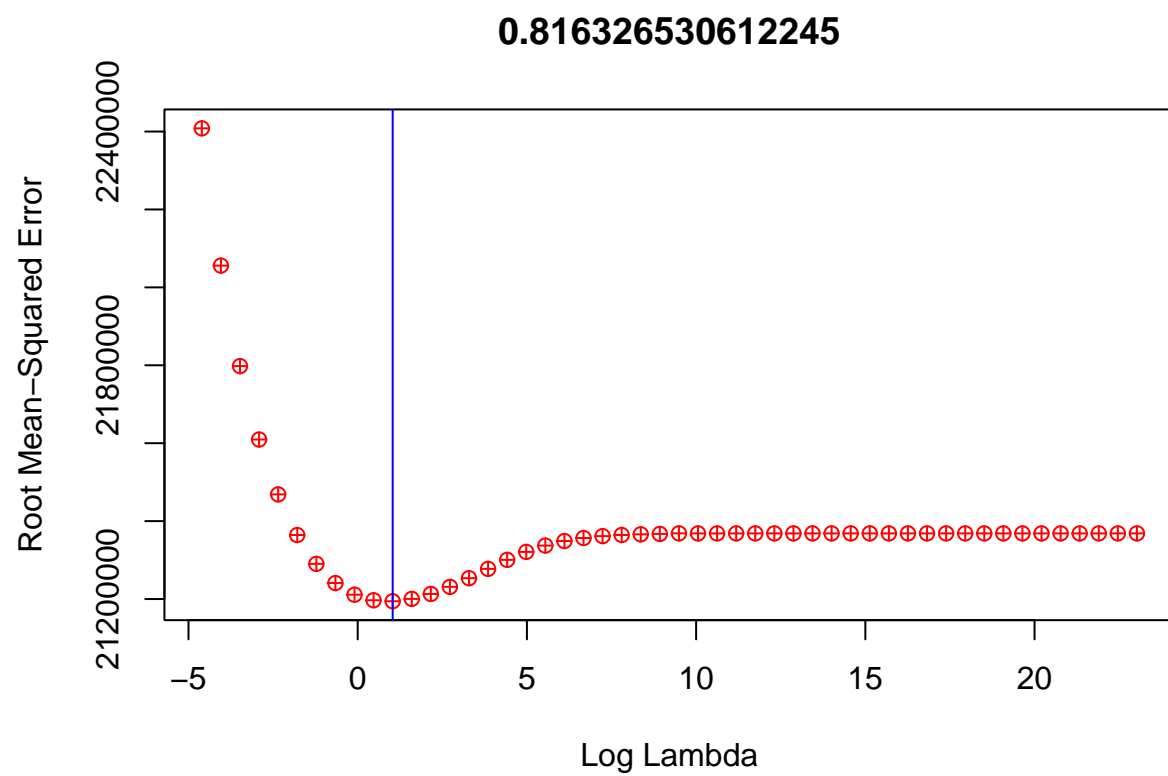


```
## Alpha is: 0.7959184 . The minimum lambda is: 2.811769
## The minimum RMSE is: 21194895
```

**0.795918367346939**

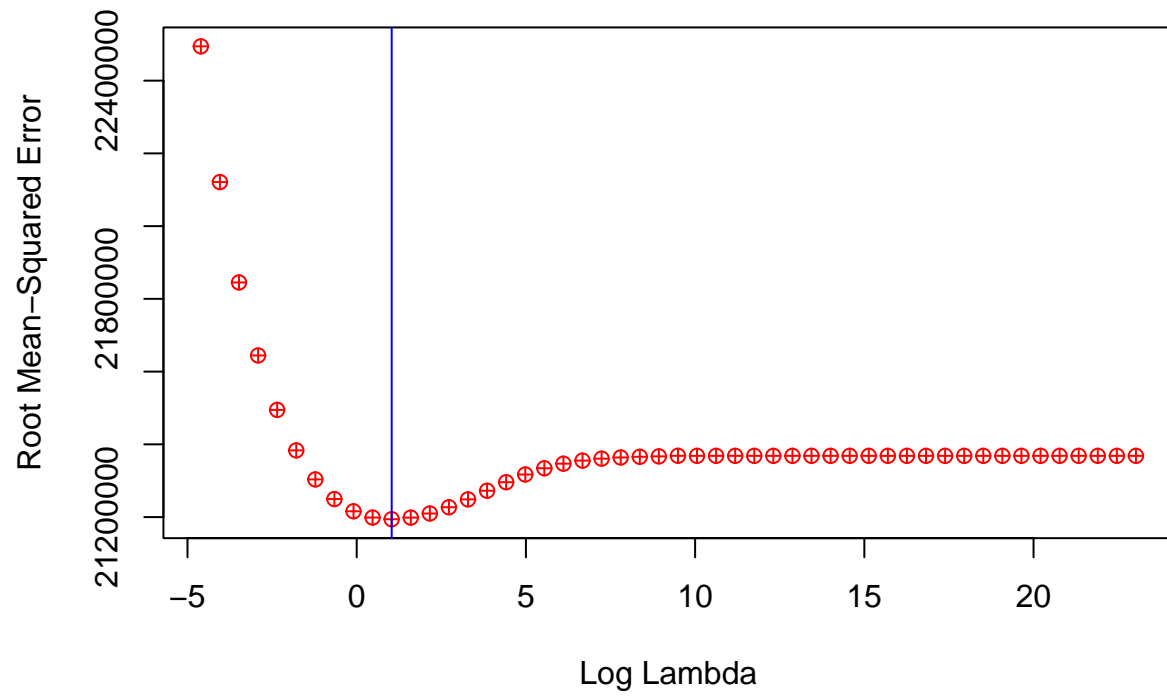


```
## Alpha is: 0.8163265 . The minimum lambda is: 2.811769
## The minimum RMSE is: 21194335
```



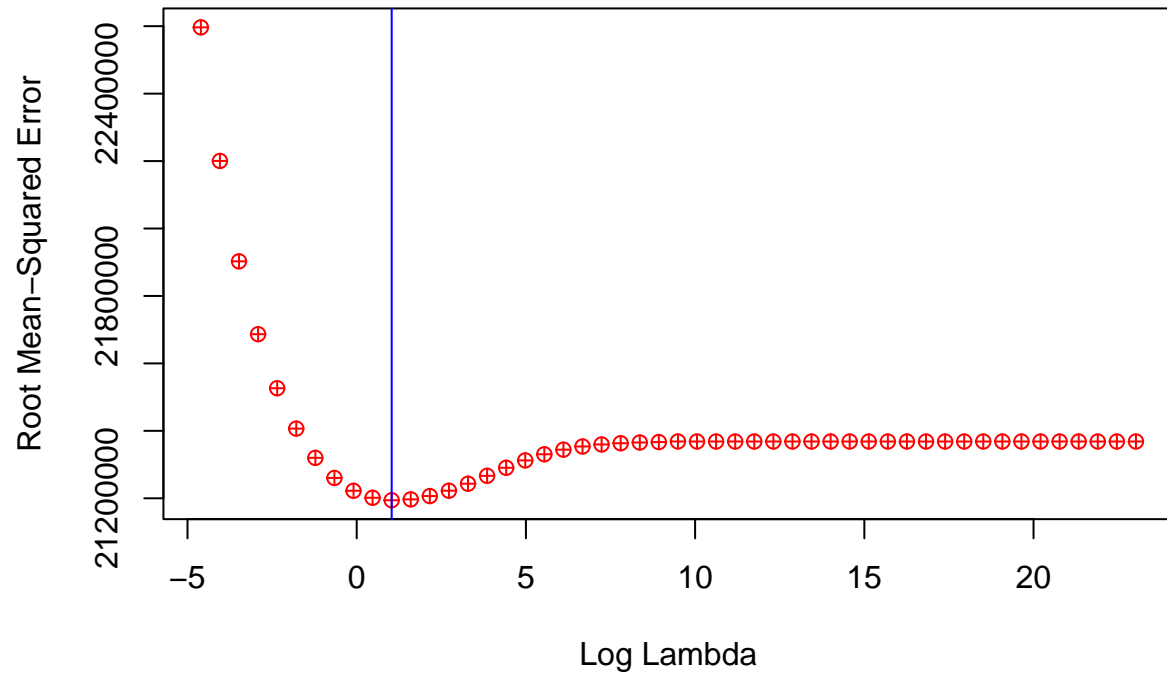
```
## Alpha is: 0.8367347 . The minimum lambda is: 2.811769
## The minimum RMSE is: 21194057
```

**0.836734693877551**



```
## Alpha is: 0.8571429 . The minimum lambda is: 2.811769
## The minimum RMSE is: 21194218
```

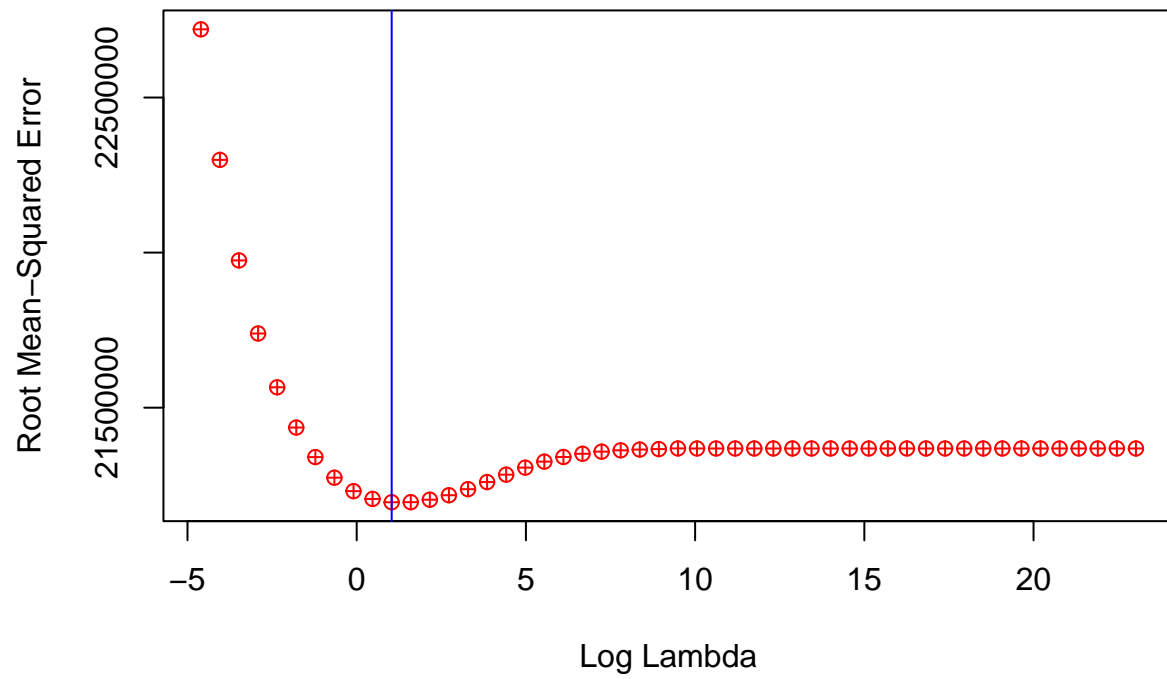
**0.857142857142857**



```
## Alpha is: 0.877551 . The minimum lambda is: 2.811769
## The minimum RMSE is: 21195088
```

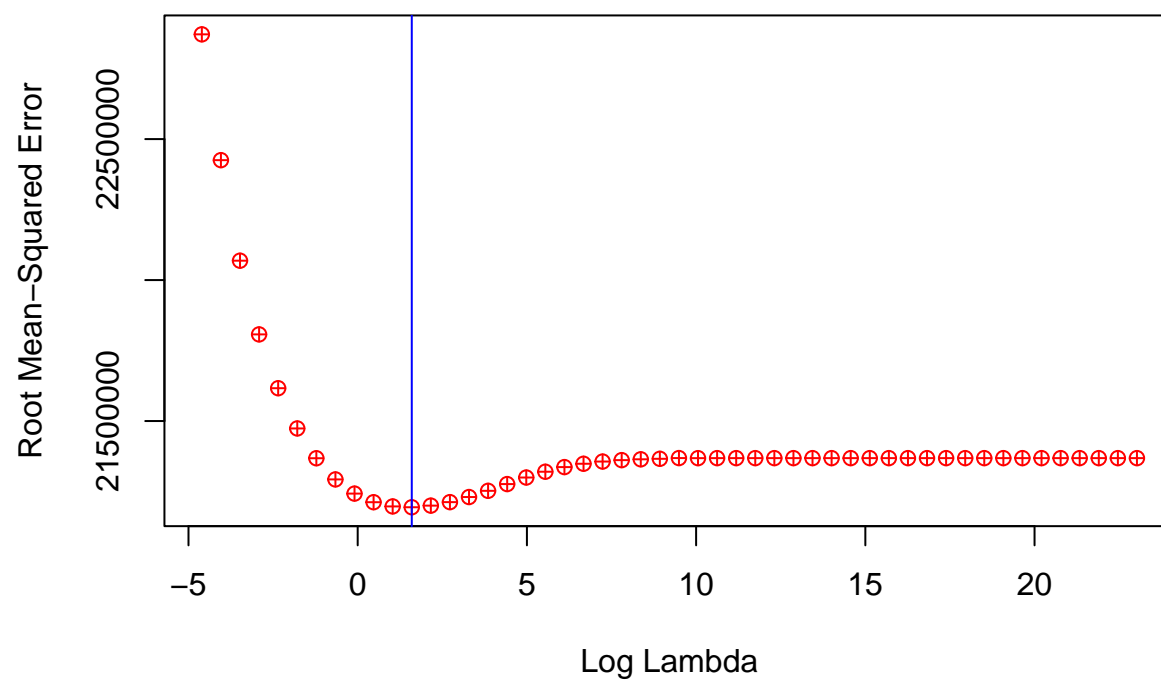


**0.877551020408163**



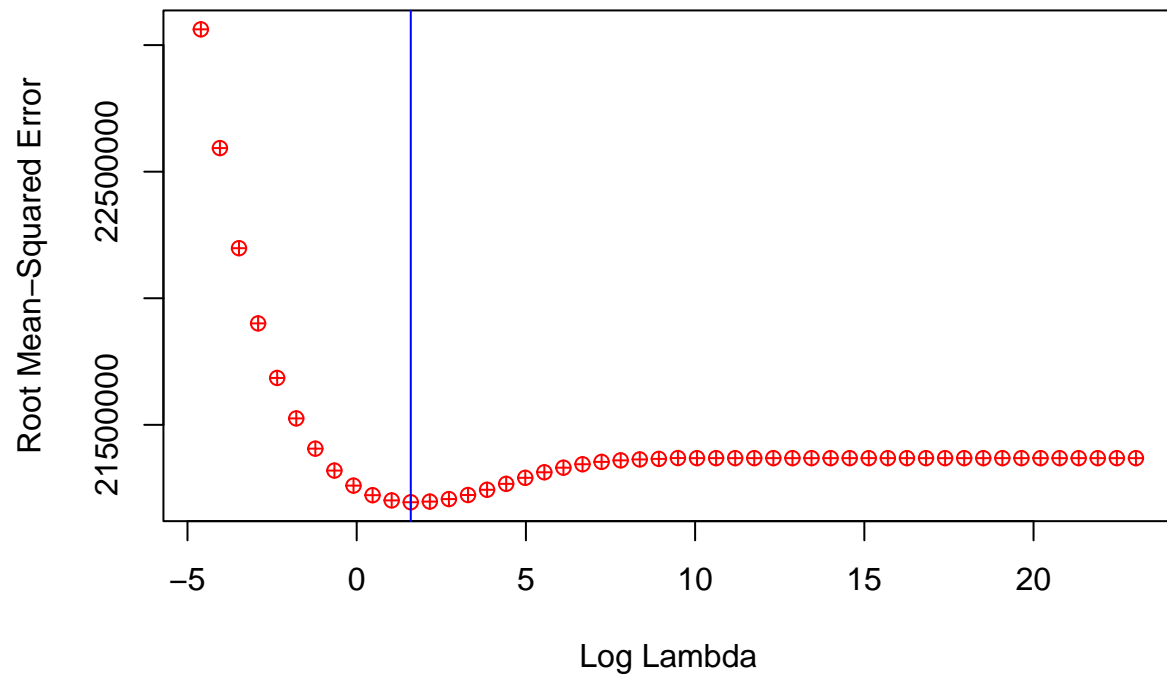
```
## Alpha is: 0.8979592 . The minimum lambda is: 4.941713
## The minimum RMSE is: 21194249
```

**0.897959183673469**



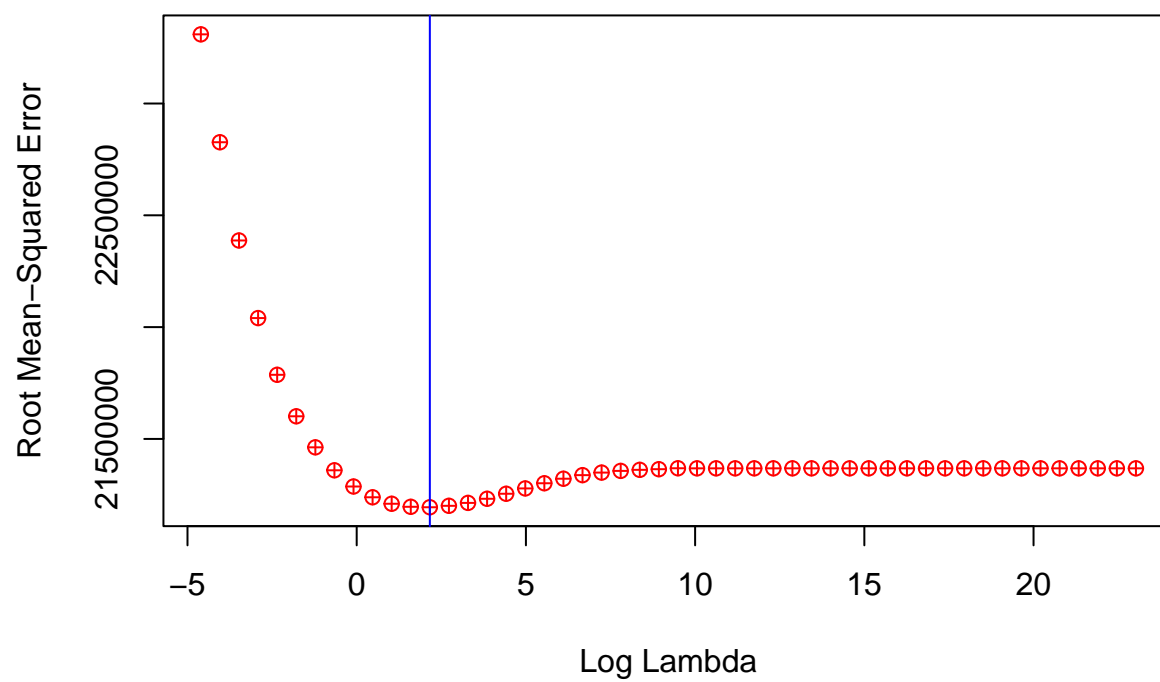
```
## Alpha is: 0.9183673 . The minimum lambda is: 4.941713
## The minimum RMSE is: 21194205
```

**0.918367346938775**



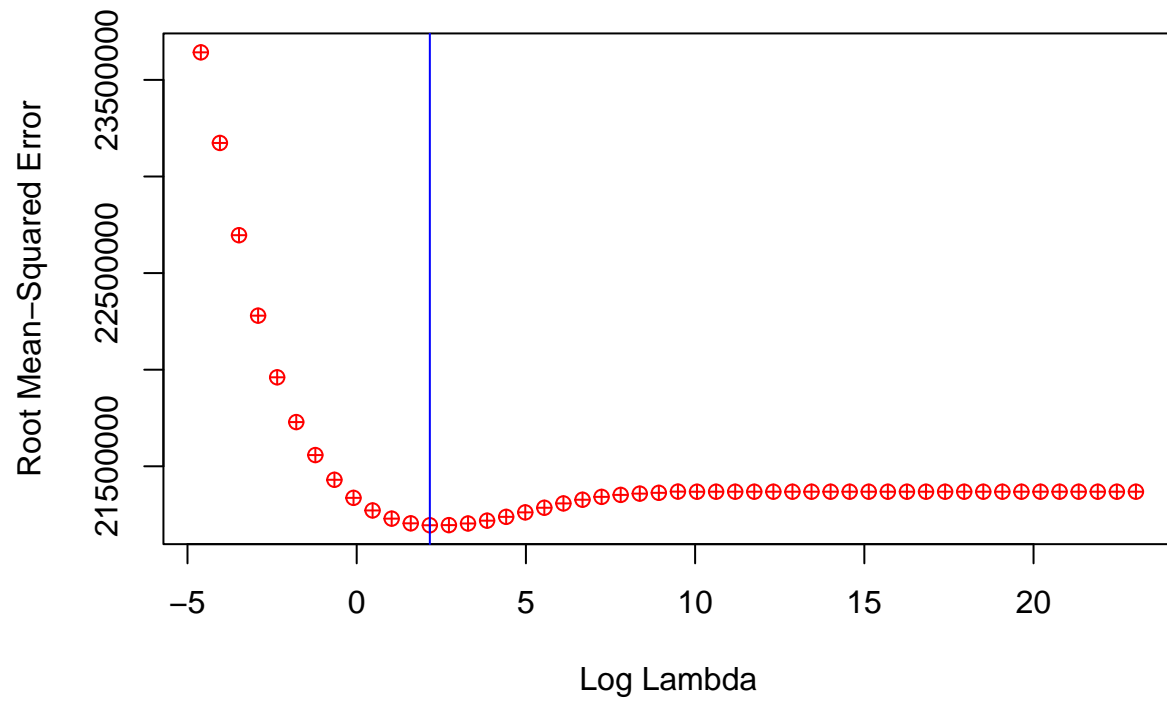
```
## Alpha is: 0.9387755 . The minimum lambda is: 8.685114
## The minimum RMSE is: 21194463
```

**0.938775510204082**



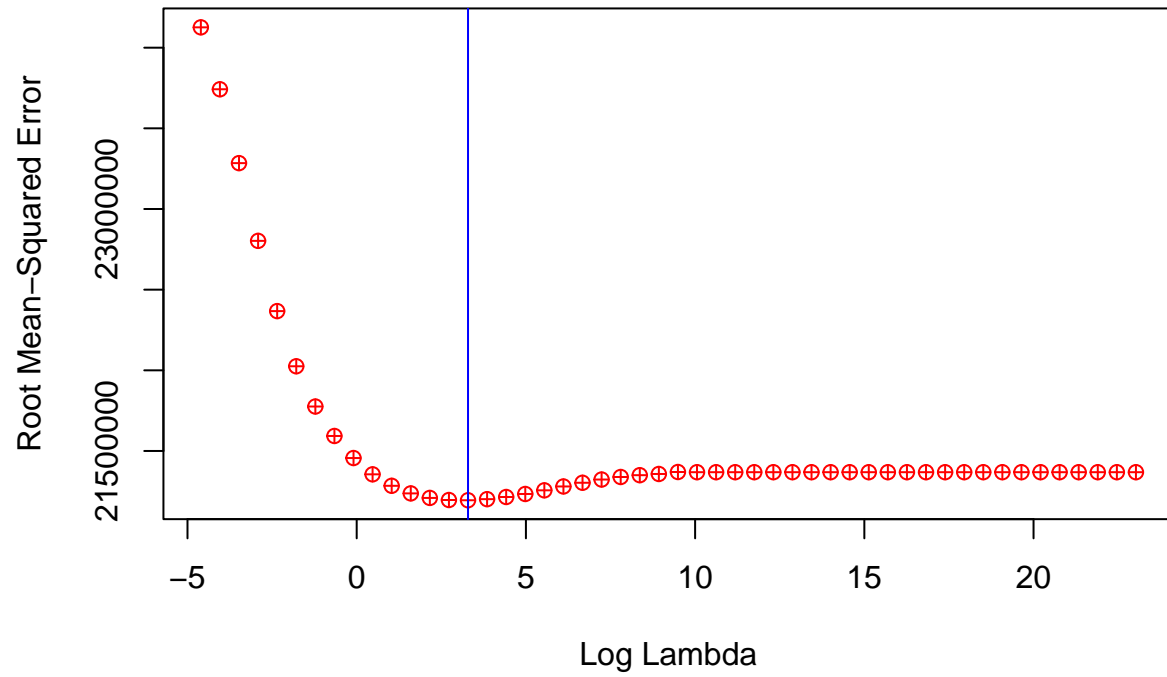
```
## Alpha is: 0.9591837 . The minimum lambda is: 8.685114
## The minimum RMSE is: 21194863
```

**0.959183673469388**



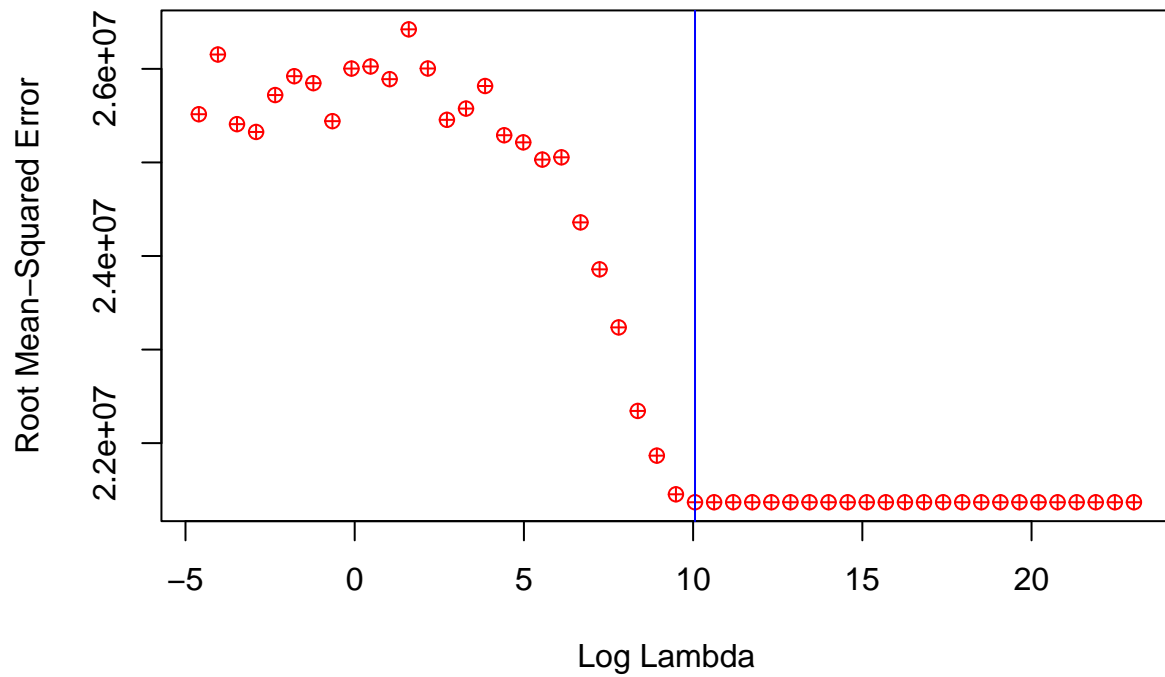
```
## Alpha is: 0.9795918 . The minimum lambda is: 26.82696
## The minimum RMSE is: 21194619
```

**0.979591836734694**



```
## Alpha is: 1 . The minimum lambda is: 23299.52
## The minimum RMSE is: 21368384
```

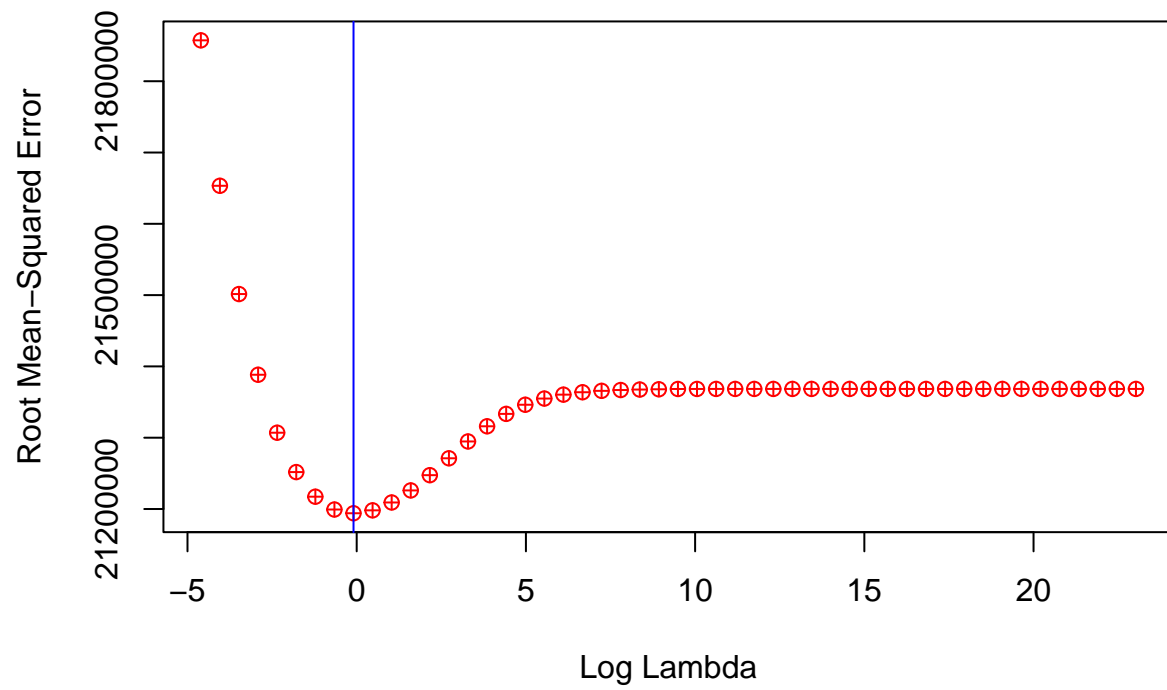
1



```
# Plot lambda behaviour with teh optimal alpha value
plot_optimal_alpha = k_fold_alpha_plots(mX,vy,nfolds,vBeta,dEps,dAlpha,lLambda)
```

```
## Alpha is: 0.5102041 . The minimum lambda is: 0.9102982
## The minimum RMSE is: 21194047
```

**0.510204081632653**



```
#plot optimal lambda, alpha using the package  
plot_cv_package = plot_cv_GLMET(mX,vy,alpha=dAlpha)
```



