Ridge

Sherry Cai, Nancy Guan, Maggie Sellers, Cas Sweeney, Xiruo Zheng

12/11/2018

```
#load libraries that we will use
library(ISLR)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-16
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:Matrix':
##
##     expand
```

```
library(readr)
library(mosaic)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggformula
```

```
## Loading required package: ggplot2
```

```
## Loading required package: ggstance
```

```
##
## Attaching package: 'ggstance'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh
```

```
##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")
```

```
## Loading required package: mosaicData
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following object is masked from 'package:ggplot2':
##
##     stat
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median,
##     prop.test, quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
#load dataset as appliances
appliances <- read_csv("energydata_complete.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   date = col_datetime(format = ""),
##   Appliances = col_integer(),
##   lights = col_integer()
## )
```

```
## See spec(...) for full column specifications.
```

Transform date variable

```
appliances <- appliances %>%
  mutate(hours=hour(date)) %>%
  mutate(months=month(date))
```

Fit data to Anova model for choosing parameters

```
model_aov <- aov(Appliances ~. , appliances)
anova(model_aov)
```

```
## Analysis of Variance Table
##
## Response: Appliances
##                Df    Sum Sq Mean Sq  F value    Pr(>F)
## date            1     19237   19237   2.1990 0.1381130
## lights          1   8220200 8220200 939.6934 < 2.2e-16 ***
## T1              1    664613  664613  75.9753 < 2.2e-16 ***
## RH_1            1    572276  572276  65.4198 6.400e-16 ***
## T2              1   2974379 2974379 340.0166 < 2.2e-16 ***
## RH_2            1   8220575 8220575 939.7362 < 2.2e-16 ***
## T3              1   5030789 5030789 575.0953 < 2.2e-16 ***
## RH_3            1    674988  674988  77.1613 < 2.2e-16 ***
## T4              1    672282  672282  76.8520 < 2.2e-16 ***
## RH_4            1    923041  923041 105.5176 < 2.2e-16 ***
## T5              1     79597   79597   9.0991 0.0025606 **
## RH_5            1        35      35   0.0040 0.9496154
## T6              1   1296066 1296066 148.1600 < 2.2e-16 ***
## RH_6            1    265651  265651  30.3679 3.619e-08 ***
## T7              1       993     993   0.1135 0.7362324
## RH_7            1   1581791 1581791 180.8226 < 2.2e-16 ***
## T8              1    233291  233291  26.6687 2.438e-07 ***
## RH_8            1   1835157 1835157 209.7862 < 2.2e-16 ***
## T9              1    323010  323010  36.9249 1.250e-09 ***
## RH_9            1    112831  112831  12.8983 0.0003297 ***
## T_out           1    462175  462175  52.8336 3.766e-13 ***
## Press_mm_hg     1     11515   11515   1.3163 0.2512623
## RH_out          1     21890   21890   2.5023 0.1136928
## Windspeed       1    208368  208368  23.8196 1.066e-06 ***
## Visibility      1     50235   50235   5.7426 0.0165675 *
## Tdewpoint       1     77687   77687   8.8808 0.0028853 **
## rv1             1      6721    6721   0.7683 0.3807613
## hours           1    447234  447234  51.1256 8.967e-13 ***
## months          1     70052   70052   8.0080 0.0046619 **
## Residuals   19705 172374367    8748
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Variables that have an impact: lights, T1, RH_1, T2, RH_2, T3, T4, RH_4, T5, T6, RH_7, T8, RH_8, T9, T_out, Windspeed, Tdewpoint, hours, months

Full model with all variables except date

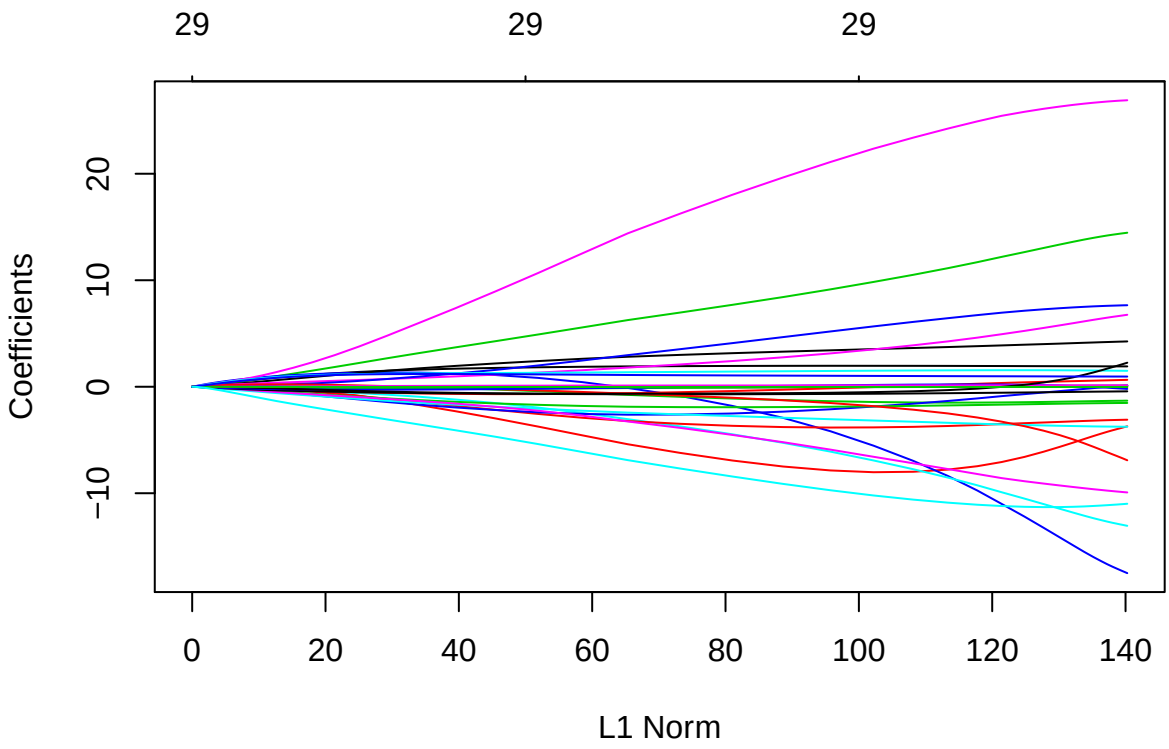
```
#trim off date because it is not strictly numerical
Energy3 <- appliances %>%
  dplyr::select(-date)
appliances_data <- Energy3
# trim off the first column
# leaving only the predictors
x = model.matrix(Appliances~., appliances_data)[,-1]
y = appliances_data %>%
dplyr::select(Appliances) %>%
unlist() %>%
as.numeric()
```

Ridge Regression on the full model

```
#a grid of values ranging from lambda = 10^10 to lambda = 10^-2 for cross-val
grid = 10^seq(10, -2, length = 100)
ridge_mod = glmnet(x, y, alpha = 0, lambda = grid)
dim(coef(ridge_mod))
```

```
## [1]  30 100
```

```
#Draw plot for coefficients
plot(ridge_mod)
```




```
#Display 50th lambda value
ridge_mod$lambda[50]
```

```
## [1] 11497.57
```

```
#Display coefficients associated with 50th lambda value
coef(ridge_mod)[,50]
```

```
##   (Intercept)        lights            T1          RH_1            T2
##  9.814437e+01  2.250640e-02  2.861348e-02  1.961470e-02  4.763767e-02
##          RH_2            T3          RH_3            T4          RH_4
## -1.314793e-02  3.666957e-02  1.045203e-02  1.580101e-02  3.593912e-03
##            T5          RH_5            T6          RH_6            T7
##  7.566819e-03  7.367249e-04  1.680593e-02 -2.285429e-03  9.174754e-03
##          RH_7            T8          RH_8            T9          RH_9
## -9.828067e-03  1.650469e-02 -1.609451e-02  2.542674e-03 -1.105492e-02
##         T_out   Press_mm_hg        RH_out     Windspeed    Visibility
##  1.606826e-02 -4.198142e-03 -9.015392e-03  3.200817e-02  6.770481e-05
##     Tdewpoint           rv1           rv2         hours        months
##  2.682480e-03 -6.827832e-04 -6.827832e-04  2.805384e-02 -1.061803e-02
```

```
# Calculate l_2 norm
sqrt(sum(coef(ridge_mod)[-1,50]^2))
```

```
## [1] 0.09666727
```

```
#Display 60th lambda value
ridge_mod$lambda[60]
```

```
## [1] 705.4802
```

```
#Display coefficients associated with 60th lambda value
coef(ridge_mod)[,60]
```

```
##   (Intercept)        lights            T1          RH_1            T2
## 122.164261527   0.314158642   0.163542178   0.294308541   0.486940865
##          RH_2            T3          RH_3            T4          RH_4
##  -0.147986589   0.406332055   0.183681219   0.041469998   0.053001411
##            T5          RH_5            T6          RH_6            T7
##  -0.075147996   0.010343968   0.173010298  -0.020340497  -0.027228306
##          RH_7            T8          RH_8            T9          RH_9
##  -0.139458359   0.091800779  -0.209670329  -0.121508092  -0.136731175
##         T_out   Press_mm_hg        RH_out     Windspeed    Visibility
##   0.146088312  -0.053334919  -0.101481598   0.409097107   0.005856810
##     Tdewpoint           rv1           rv2         hours        months
##  -0.018175968  -0.007821871  -0.007821891   0.355580647  -0.377150046
```

```
#Calculate l2 norm
sqrt(sum(coef(ridge_mod)[-1,60]^2))
```

```
## [1] 1.134723
```

```
set.seed(1)
#Specifying training and test set
#Half of the data set is used as training set, the other half as test set
train = appliances_data %>%
sample_frac(0.5)
test = appliances_data %>%
setdiff(train)
x_train = model.matrix(Appliances~., train)[,-1]
x_test = model.matrix(Appliances~., test)[,-1]
y_train = train %>%
```

```
dplyr::select(Appliances) %>%
unlist() %>%
as.numeric()
y_test = test %>%
dplyr::select(Appliances) %>%
unlist() %>%
as.numeric()
#Fit the model to training set
ridge_mod = glmnet(x_train, y_train, alpha=0, lambda = grid, thresh = 1e-12)
#predict with lambda=4
ridge_pred = predict(ridge_mod, s = 4, newx = x_test)
#testMSE for lambda=4
mean((ridge_pred - y_test)^2)
```

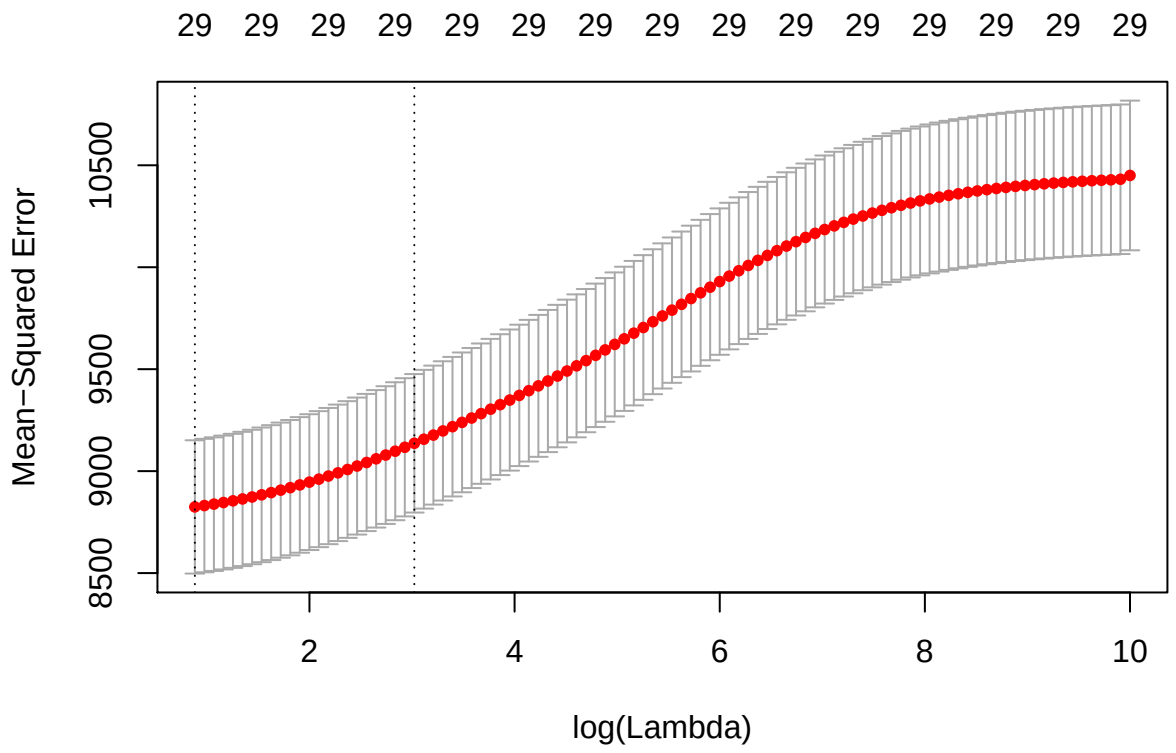
```
## [1] 8864.027
```

Predict with cross validation

```
set.seed(1)
# Fit ridge regression model on training data
cv.out = cv.glmnet(x_train, y_train, alpha = 0)
# Select lamda that minimizes training MSE
bestlam = cv.out$lambda.min
bestlam
```

```
## [1] 2.417414
```

```
# Draw plot of training MSE as a function of lambda
plot(cv.out)
```



```
# Use best lambda to predict test data
ridge_pred = predict(ridge_mod, s = bestlam, newx = x_test)
# Calculate test MSE
mean((ridge_pred - y_test)^2)
```

```
## [1] 8818.822
```

```
# Fit ridge regression model on full dataset
out = glmnet(x, y, alpha = 0)
# Display coefficients using lambda chosen by CV
predict(out, type = "coefficients", s = bestlam)[1:30,]
```

```
##  (Intercept)       lights           T1         RH_1           T2
## 14.619151719  1.968380921 -7.861282633  9.383705189 -4.532743962
##         RH_2           T3         RH_3           T4         RH_4
## -6.342951672 21.547492307  3.452044671 -3.891017133 -1.411720646
##           T5         RH_5           T6         RH_6           T7
## -2.135222756  0.016267843  3.196666817 -0.039475437 -0.149999292
##         RH_7           T8         RH_8           T9         RH_9
## -1.876174774  5.300466222 -3.090007331 -6.032630405 -0.670794628
##        T_out  Press_mm_hg       RH_out    Windspeed   Visibility
## -1.568955393 -0.002926499  0.144319854  1.486100717  0.124279935
##    Tdewpoint          rv1          rv2        hours       months
## -0.541064304 -0.021529592 -0.021559262  1.022031043 -9.865584099
```

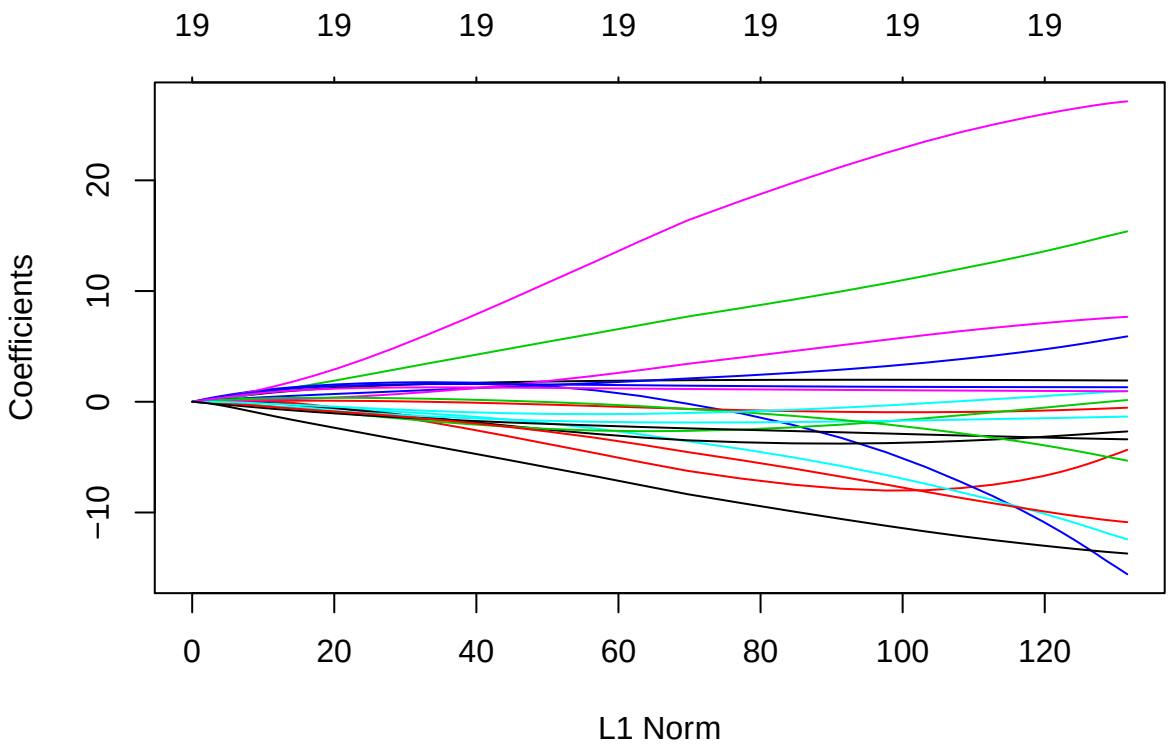
Limited Model with Select Variables

```
appliances_data_limited <- Energy3 %>%
  dplyr::select(Appliances,lights, T1, RH_1, T2, RH_2, T3, T4, RH_4, T5, T6, RH_7, T8, RH_8, T9, T_out,
# trim off the first column
x = model.matrix(Appliances~., appliances_data_limited)[,-1]
# leaving only the predictors
y = appliances_data_limited %>%
dplyr::select(Appliances) %>%
unlist() %>%
as.numeric()
```

```
grid = 10^seq(10, -2, length = 100)
ridge_mod = glmnet(x, y, alpha = 0, lambda = grid)
dim(coef(ridge_mod))
```

```
## [1]  20 100
```

```
plot(ridge_mod)
```




```
# Display 50th lambda value
ridge_mod$lambda[50]
```

```
## [1] 11497.57
```

```
# Display coefficients associated with 50th lambda value
coef(ridge_mod)[,50]
```

```
##  (Intercept)       lights           T1         RH_1           T2
## 94.228062144  0.022488737  0.029163239  0.019507457  0.048110630
##         RH_2           T3           T4         RH_4           T5
## -0.013373369  0.037077630  0.016278913  0.003446106  0.008010346
##           T6         RH_7           T8         RH_8           T9
##  0.016993050 -0.009959999  0.016957637 -0.016269856  0.003000415
##        T_out    Windspeed    Tdewpoint        hours       months
##  0.016280323  0.032081648  0.002716952  0.028144936 -0.009893799
```

```
# Calculate l_2 norm
sqrt(sum(coef(ridge_mod)[-1,50]^2))
```

```
## [1] 0.09532564
```

```
#a test with lambda = 50
predict(ridge_mod, s = 50, type = "coefficients")[1:20,]
```

```
## (Intercept)      lights          T1        RH_1          T2        RH_2
## 40.41233628  1.44991019 -1.08485402  2.64340119  1.70450751 -0.81293626
##          T3          T4        RH_4          T5          T6        RH_7
##  4.34278958 -0.96083290  0.06081841 -1.34648910  0.87220316 -1.08026812
##          T8        RH_8          T9       T_out   Windspeed   Tdewpoint
##  0.59183023 -1.29923750 -1.17285490  0.33793872  1.53334360 -0.62999622
##       hours      months
##  1.26928743 -3.09375063
```

```r
set.seed(1)
#Specifying training and test set
#Half of the data set is used as training set, the other half as test set
train = appliances_data_limited %>%
  sample_frac(0.5)
test = appliances_data_limited %>%
  setdiff(train)
x_train = model.matrix(Appliances~., train)[,-1]
x_test = model.matrix(Appliances~., test)[,-1]
y_train = train %>%
  dplyr::select(Appliances) %>%
  unlist() %>%
  as.numeric()
y_test = test %>%
  dplyr::select(Appliances) %>%
  unlist() %>%
  as.numeric()
#Fit the model to training set
ridge_mod = glmnet(x_train, y_train, alpha=0, lambda = grid, thresh = 1e-12)
#predict with lambda = 4
ridge_pred = predict(ridge_mod, s = 4, newx = x_test)
#testMSE for lambda=4
mean((ridge_pred - y_test)^2)
```

```
## [1] 8880.76
```

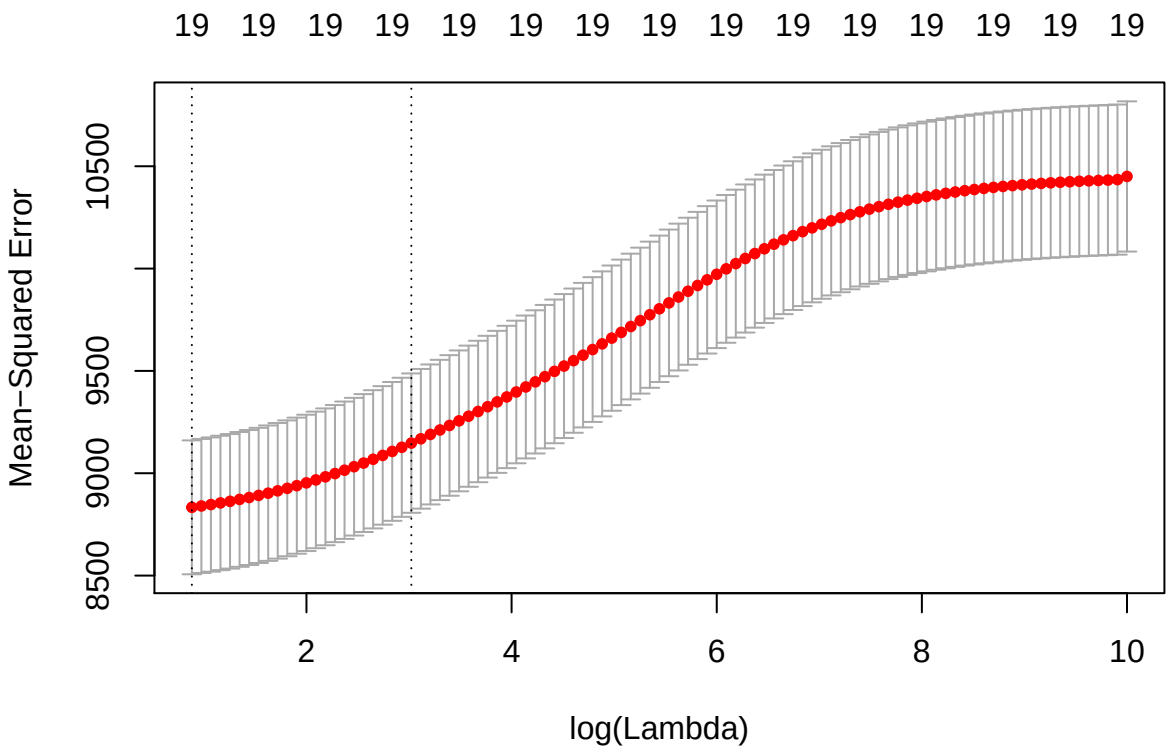
Select best lambda with cross validation

Predict test data and generate MSE

```r
set.seed(1)
# Fit ridge regression model on training data
cv.out = cv.glmnet(x_train, y_train, alpha = 0)
# Select lamda that minimizes training MSE
bestlam = cv.out$lambda.min
bestlam
```

```
## [1] 2.417414
```

```r
# Draw plot of training MSE as a function of lambda
plot(cv.out)
```




```
# Use best lambda to predict test data
ridge_pred = predict(ridge_mod, s = bestlam, newx = x_test)
# Calculate test MSE
mean((ridge_pred - y_test)^2)
```

```
## [1] 8836.032
```

Obtain coefficients with best lambda seleted through cross-val

```
# Fit ridge regression model on full dataset
out = glmnet(x, y, alpha = 0)
# Display coefficients using lambda chosen by CV
predict(out, type = "coefficients", s = bestlam)[1:20,]
```

```
## (Intercept)      lights          T1        RH_1          T2        RH_2
##  51.4539973   1.9898974  -7.8516443  10.2093191  -3.5901596  -6.0424822
##          T3          T4        RH_4          T5          T6        RH_7
##  21.6671624  -3.8361442  -0.9516740  -2.0693476   2.9491173  -1.7731762
##          T8        RH_8          T9       T_out   Windspeed   Tdewpoint
##   5.1999973  -2.7829241  -6.9323428  -1.7226453   1.3617670  -0.4702258
##       hours      months
##   1.0638973 -10.7547741
```