p8106_mtp_rq2166

Ruoyuan Qian

3/3/2020

Data cleaning

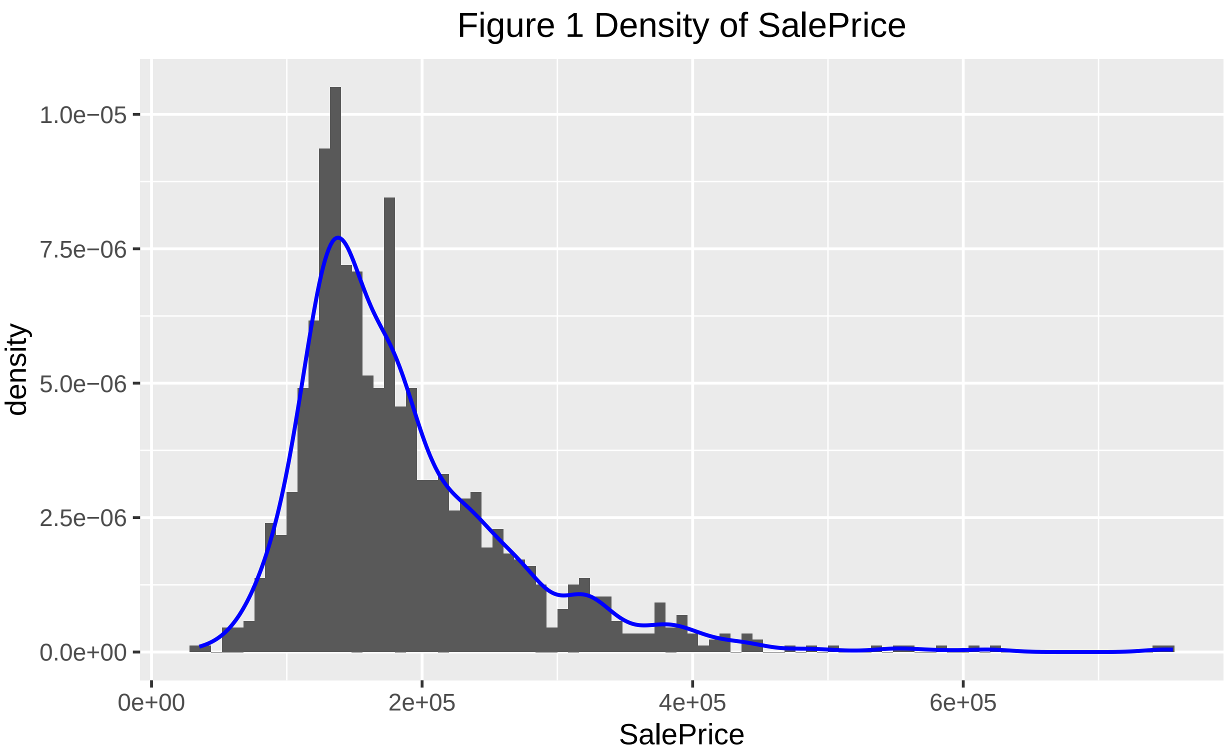
```r
house = read.csv(file = "train.csv")
#skimr::skim(house)

sum_na = function(x){
  sum = sum(is.na(x))
  sum}

# names of predictor when its missing value larger than 500
missing_var = map(house,sum_na) %>%
  as.data.frame() %>%
  pivot_longer(
    Id : SalePrice,
    names_to = "variable",
    values_to = "value"
  ) %>%
  filter(value > 500 ) %>%
  pull(variable)

#house %>%
#select(-Alley,-FireplaceQu,-PoolQC,-Fence,-MiscFeature) %>%
#map(.,sum_na)
# names of variables when its value nears zero
near_0_var =
  house %>%
  nearZeroVar( names = TRUE)

final_house =
  house %>%
  #nearZeroVar( names = TRUE)
  select(-near_0_var,-missing_var,-Id) %>%
  #select(-Alley,-FireplaceQu,-PoolQC,-Fence,-MiscFeature) %>%
  drop_na()
```

Visualization

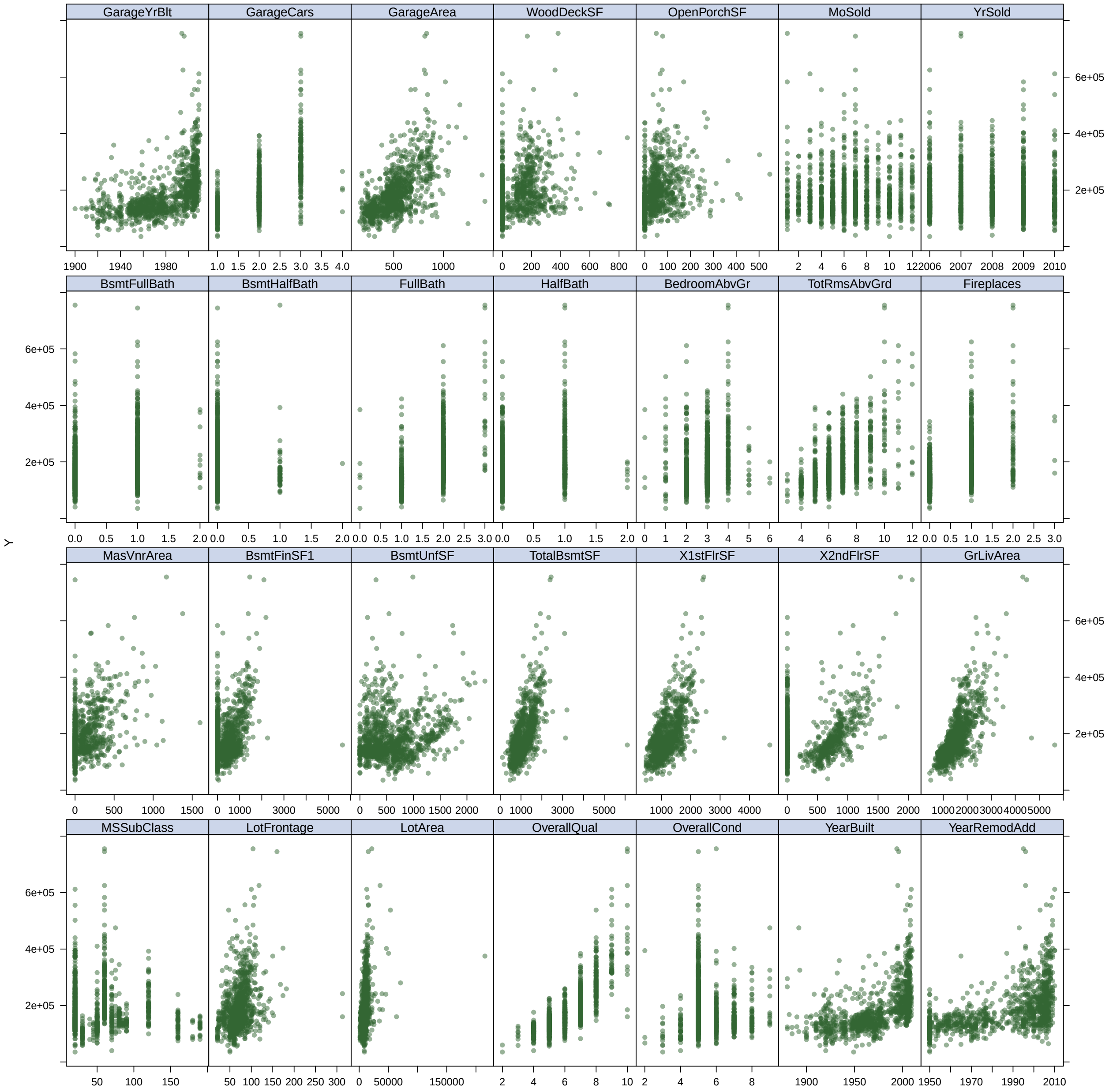
The response `SalePrice` is right skewed

```r
density_sale =
ggplot(final_house, aes(x = SalePrice, ..density..)) +
  geom_histogram(binwidth = 8000) +
  geom_line(stat = 'density',size = 0.7,color = "blue")+
  ggtitle("Figure 1 Density of SalePrice") +
  #ylab("Houses") +
  xlab("SalePrice") +
  theme(plot.title = element_text(hjust = 0.5))
```

```
density_sale
```

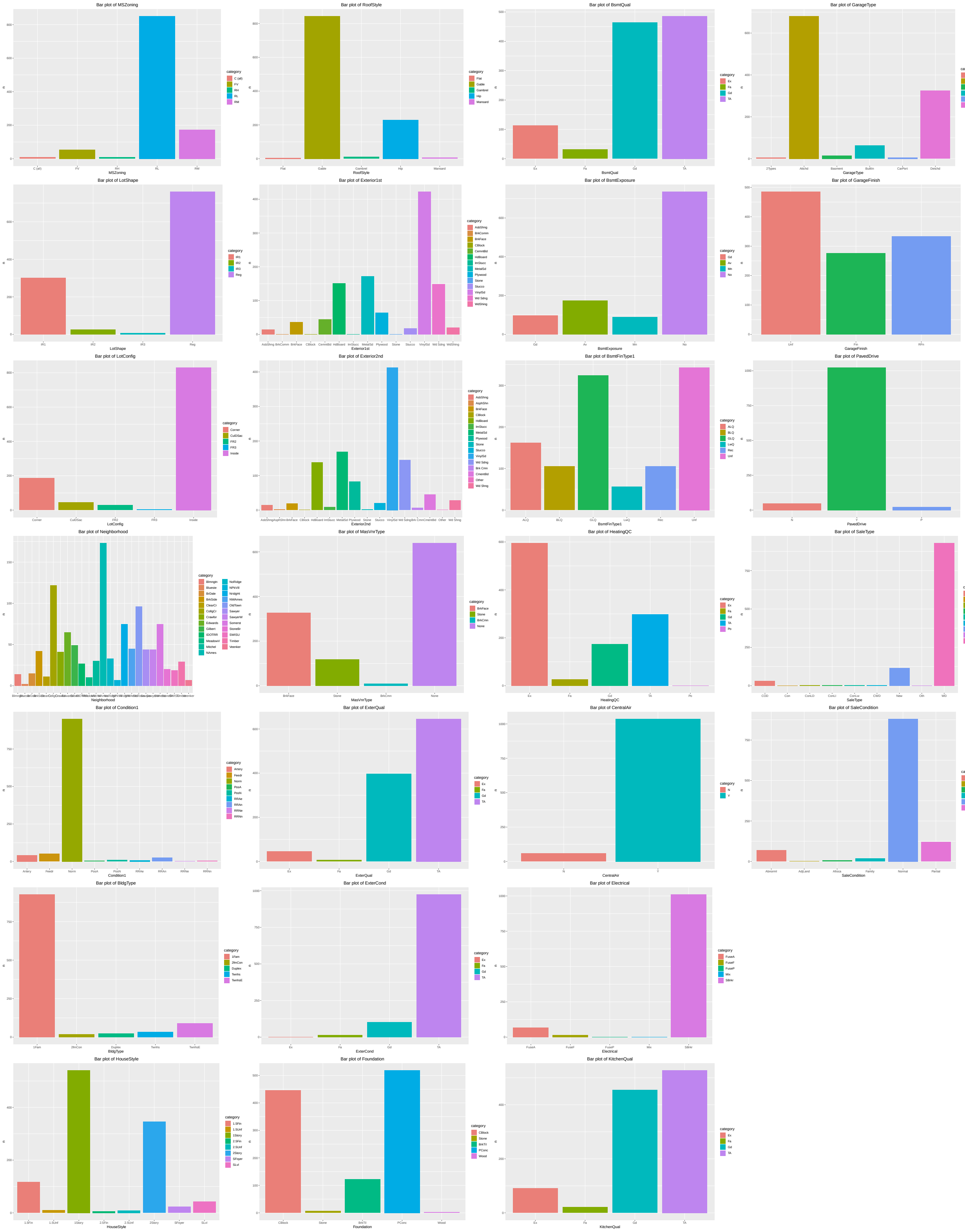


```
numeric_var_index =
 final_house%>%
  map(.,is.numeric) %>%
  unlist() %>%
  as.vector()

x <- model.matrix(SalePrice~.,
                  final_house[,which(numeric_var_index == TRUE)])[,-1]
y <- final_house$SalePrice


theme1 <- trellis.par.get()
theme1$plot.symbol$col <- rgb(.2, .4, .2, .5)
theme1$plot.symbol$pch <- 16
theme1$plot.line$col <- rgb(.8, .1, .1, 1)
theme1$plot.line$lwd <- 2
theme1$strip.background$col <- rgb(.0, .2, .6, .2)
trellis.par.set(theme1)
featurePlot(x, y, plot = "scatter", labels = c("","Y"),
            type = c("p"), layout = c(7, 4))
```



```
variable_name = names(final_house %>%
  select(which(numeric_var_index == FALSE)))

dataframe =
  final_house %>%
  select(which(numeric_var_index == FALSE))
plots = function(dataframe){
variable_name = names(dataframe)

summary =
final_house %>%
  select(which(numeric_var_index == FALSE)) %>%
  pivot_longer(
   everything(),
```

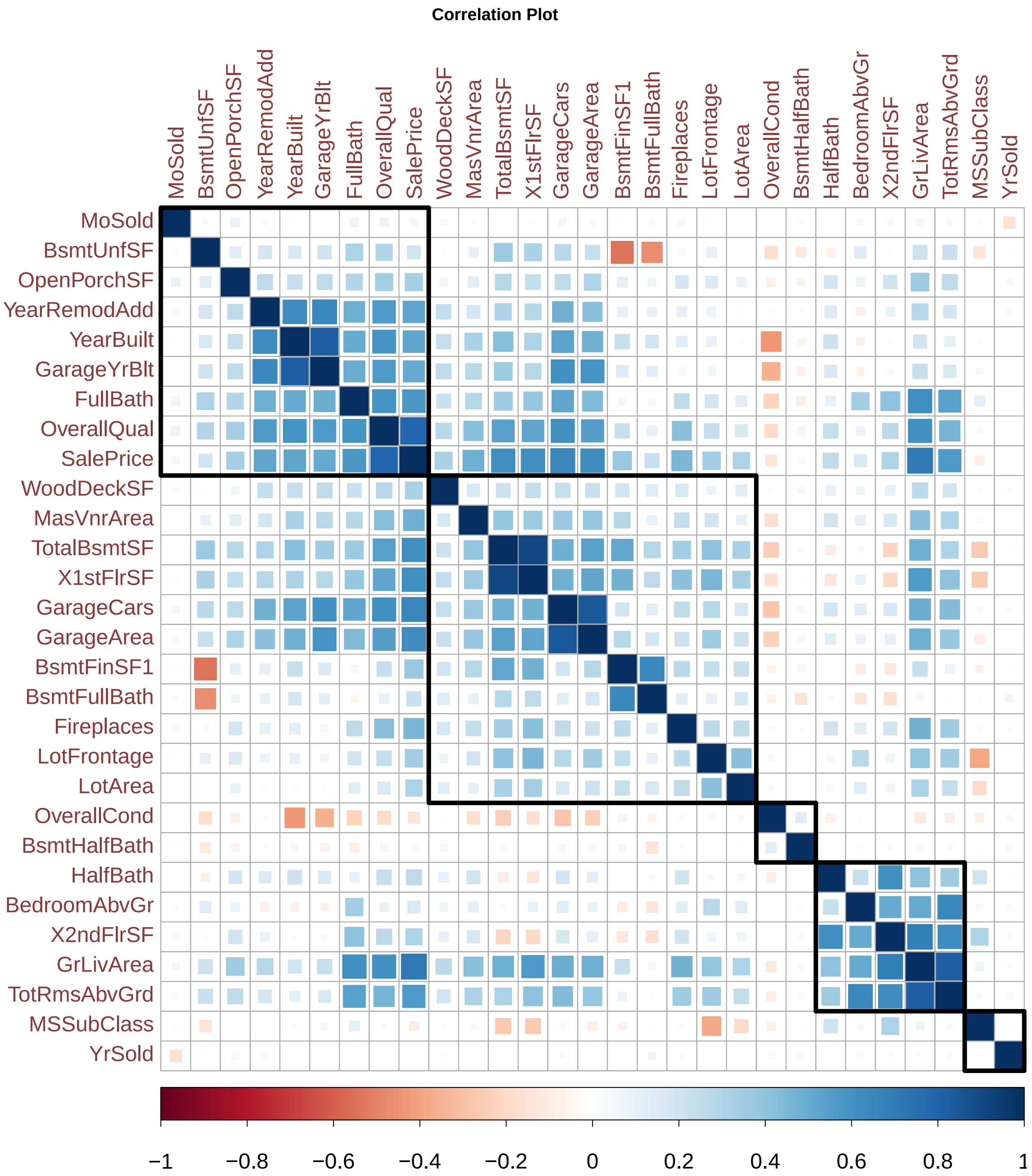
```r
   names_to = "variable",
   values_to = "category"
  ) %>%
  group_by(variable,category)%>%
  count() %>%
  mutate(n = freq) %>%
  select(-freq)

plot_tem =
  summary %>%
  filter(variable == variable_name) %>%
  ggplot(mapping = aes(x = category,
                       y = n,fill = category)) +
   geom_bar(stat = 'identity',position = 'dodge') +
 scale_fill_hue(c = 80)+
 ggtitle(paste("Bar plot of",variable_name))+
  labs(x = variable_name) +
 theme(plot.title = element_text(hjust = 0.5),
       legend.position="right")
    #plots = paste(plots,plot_tem,"+")
plot_tem
  }


plot_name = NULL
for(i in 1: length(dataframe)){
  plot_name_tem = paste("plots(dataframe %>% select(",i,"))",",")
  plot_name = c(plot_name,plot_name_tem)
}

#as.factor(plot_name)

multiplot(
 plots(dataframe %>% select( 1 )) ,
plots(dataframe %>% select( 2 )) ,
plots(dataframe %>% select( 3 )) ,
plots(dataframe %>% select( 4 )) ,
plots(dataframe %>% select( 5 )) ,
plots(dataframe %>% select( 6 )) ,
plots(dataframe %>% select( 7 )) ,
plots(dataframe %>% select( 8 )) ,
plots(dataframe %>% select( 9 )) ,
plots(dataframe %>% select( 10 )) ,
plots(dataframe %>% select( 11 )) ,
plots(dataframe %>% select( 12 )) ,
plots(dataframe %>% select( 13 )) ,
plots(dataframe %>% select( 14 )) ,
plots(dataframe %>% select( 15 )) ,
plots(dataframe %>% select( 16 )) ,
plots(dataframe %>% select( 17 )) ,
plots(dataframe %>% select( 18 )) ,
plots(dataframe %>% select( 19 )) ,
plots(dataframe %>% select( 20 )) ,
plots(dataframe %>% select( 21 )) ,
plots(dataframe %>% select( 22 )) ,
plots(dataframe %>% select( 23 )) ,
plots(dataframe %>% select( 24 )) ,
plots(dataframe %>% select( 25 )) ,
plots(dataframe %>% select( 26 )) ,
          cols=4)
```



heat map

```
M<-cor(final_house[,which(numeric_var_index == TRUE)])
```

```
#corrplot(M,title = "Correlation Plot", method = "square", addgrid.col = "darkgray", order="hclust", ma

corrplot(M,title = "Correlation Plot", method = "square", addgrid.col = "darkgray", order="hclust", mar
```



```
reg_data = as.data.frame(map(final_house,as.numeric))

x <- model.matrix(SalePrice~.,reg_data)[,-1]
y <- reg_data$SalePrice
```

```
ctrl1 <- trainControl(method = "repeatedcv", number = 10, repeats = 5)
```

Multiple linear regression

```
set.seed(1)
lm.fit <- train(x, y,
                method = "lm",
                trControl = ctrl1,
                preProcess = c("center", "scale"))

tibble("MSE" = lm.fit $ results $ RMSE)%>%
knitr::kable()
```

MSE
37505.04

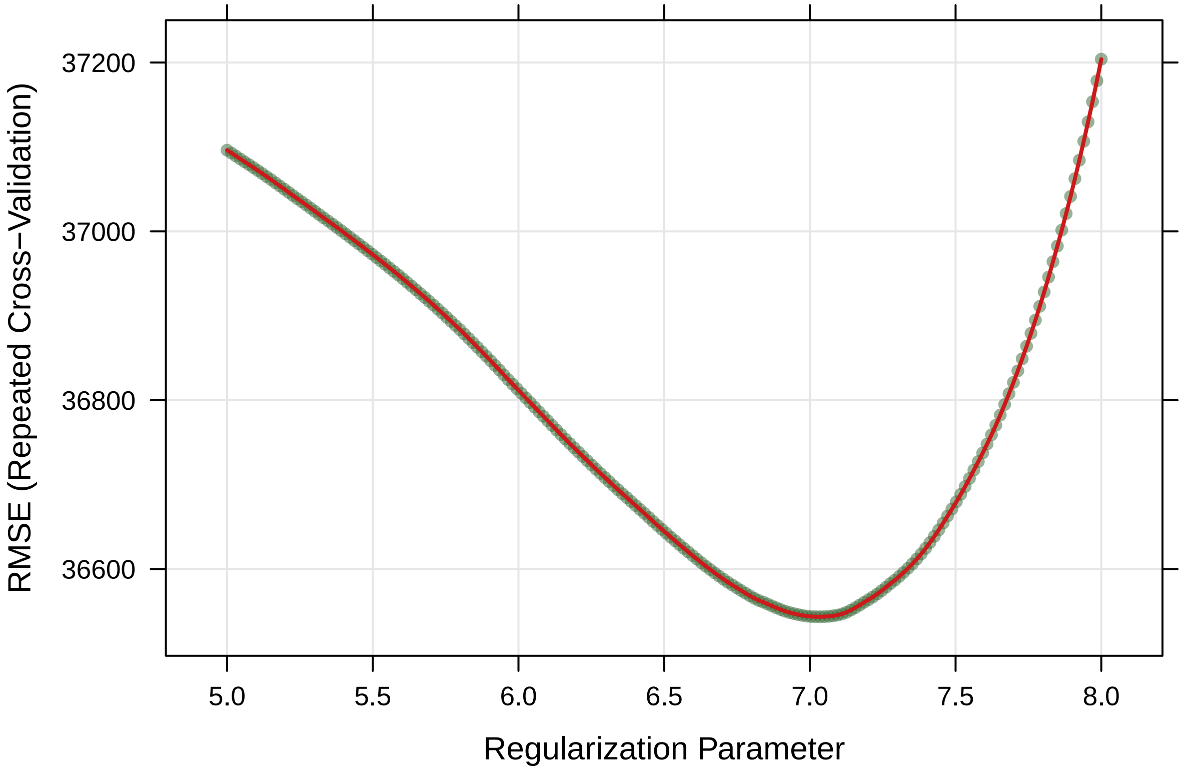
K-nn

```
set.seed(1)
knnFit <- train(x, y,
                method = "knn",
                trControl = ctrl1,
                preProcess = c("center", "scale"),
                tuneLength = 20)
               #  tuneGrid = tibble(n = 7:10))


as.data.frame(knnFit$ results ) %>%
  select(k,RMSE) %>%
  filter(RMSE == min(RMSE)) %>%
  knitr::kable()
```

k	RMSE
11	38064.71

LASSO

```
set.seed(1)
lasso.fit <- train(x, y,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 1,
                                          lambda = exp(seq(5,8, length=200))),
                   preProc = c("center", "scale"),
                   trControl = ctrl1)

plot(lasso.fit, xTrans = function(x) log(x))
```

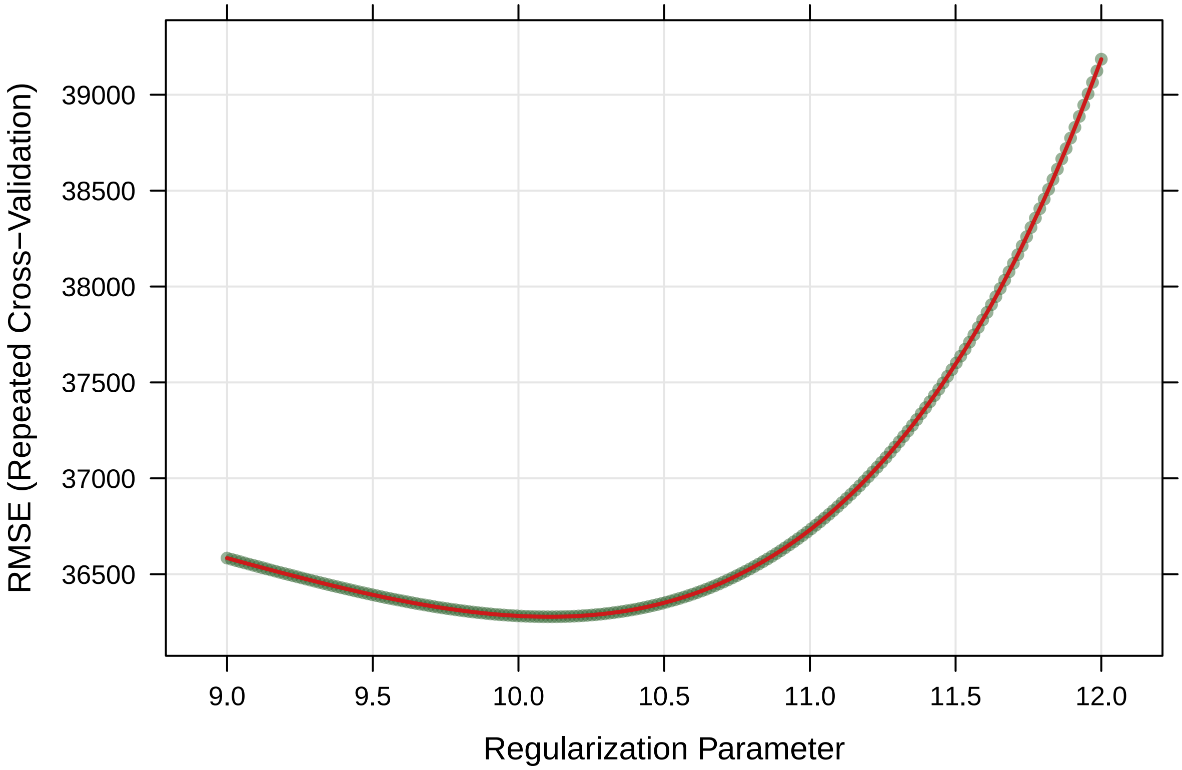


```
coe = coef(lasso.fit$finalModel,lasso.fit$bestTune$lambda)

as.data.frame(lasso.fit$ results ) %>%
  select(lambda,RMSE) %>%
  filter(RMSE == min(RMSE)) %>%
  mutate("Number of non-zero coefficient" = length(which(coe[-1] != 0))) %>%
  knitr::kable()
```

lambda	RMSE	Number of non-zero coefficient
1135.895	36543.47	32

Ridge

```
set.seed(1)
ridge.fit <- train(x, y,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 0,
                                          lambda = exp(seq(9, 12, length=200))),
                   preProc = c("center", "scale"),
                   trControl = ctrl1)


plot(ridge.fit, xTrans = function(x) log(x))
```



```
as.data.frame(ridge.fit$ results ) %>%
  select(lambda,RMSE) %>%
  filter(RMSE == min(RMSE)) %>%
  knitr::kable()
```

lambda	RMSE
24355.25	36278.35

Elastic

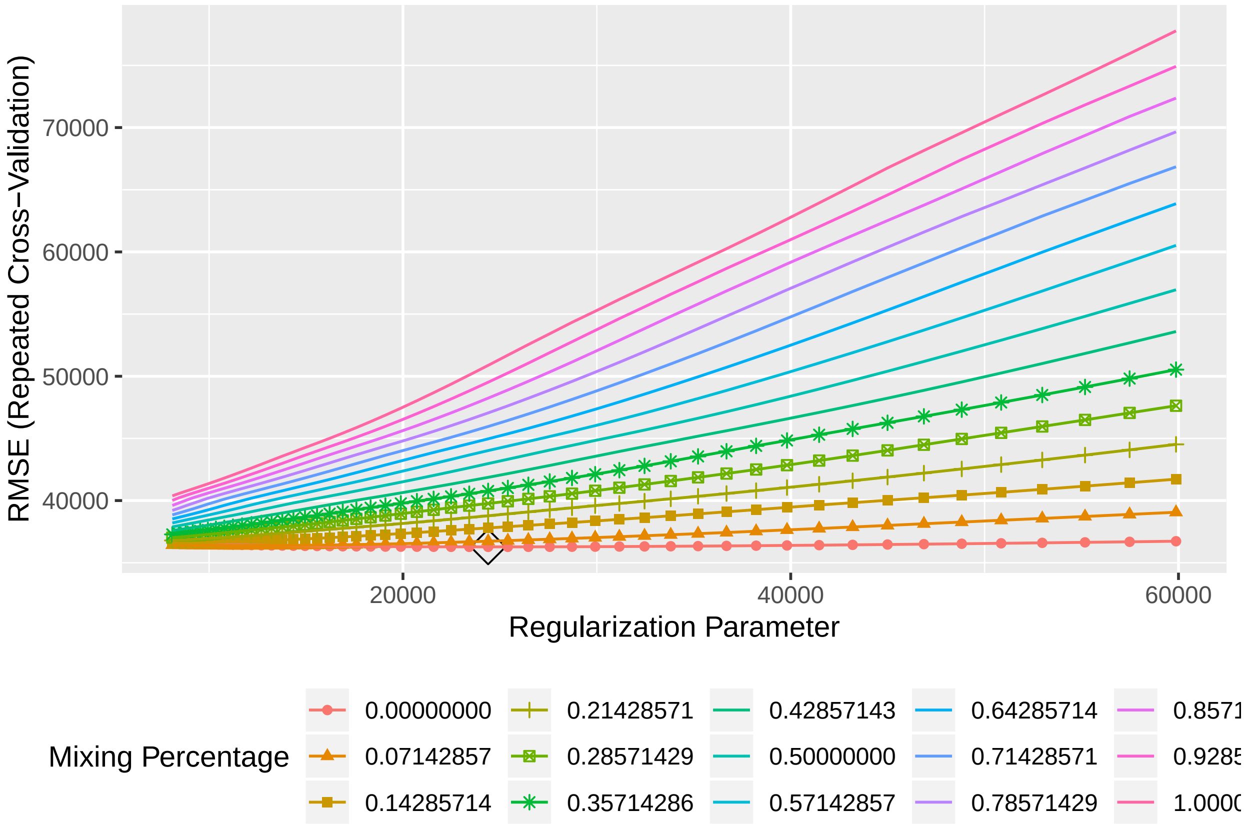
```
set.seed(1)
enet.fit <- train(x, y,
                  method = "glmnet",
                  tuneGrid =
                    expand.grid(alpha = seq(0, 1, length = 15),
                                lambda = exp(seq(9,11, length = 50))),
                  preProc = c("center", "scale"),
                  trControl = ctrl1)
enet.fit$bestTune
```

```
##    alpha   lambda
## 28     0 24392.74
```

```
ggplot(enet.fit, highlight = TRUE) +
  theme(legend.position="bottom")
```

```
## Warning: The shape palette can deal with a maximum of 6 discrete values
## because more than 6 becomes difficult to discriminate; you have
## 15. Consider specifying shapes manually if you must have them.
```

```
## Warning: Removed 450 rows containing missing values (geom_point).
```



```
as.data.frame(enet.fit$ results ) %>%
  select(lambda,RMSE) %>%
  filter(RMSE == min(RMSE)) %>%
  knitr::kable()
```

lambda	RMSE
24392.74	36278.34

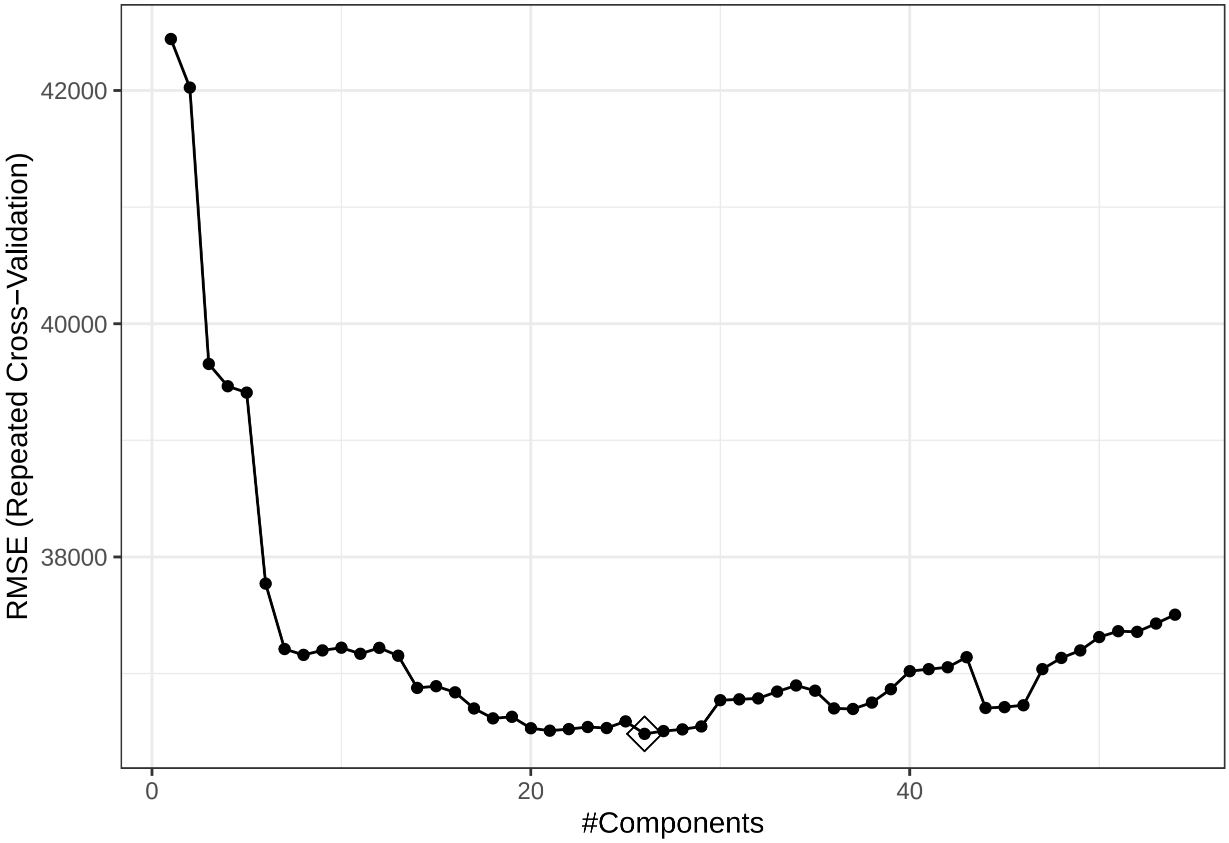
PCA

```
set.seed(1)

pcr.fit <- train(x, y,
                 method = "pcr",
                 tuneGrid = data.frame(ncomp = 1:54),
                 trControl = ctrl1,
                 scale = TRUE)
```

```
as.data.frame(pcr.fit$ results ) %>%
  select(ncomp,RMSE) %>%
  filter(RMSE == min(RMSE)) %>%
  knitr::kable()
```

ncomp	RMSE
26	36483.11

```
ggplot(pcr.fit, highlight = TRUE) + theme_bw()
```

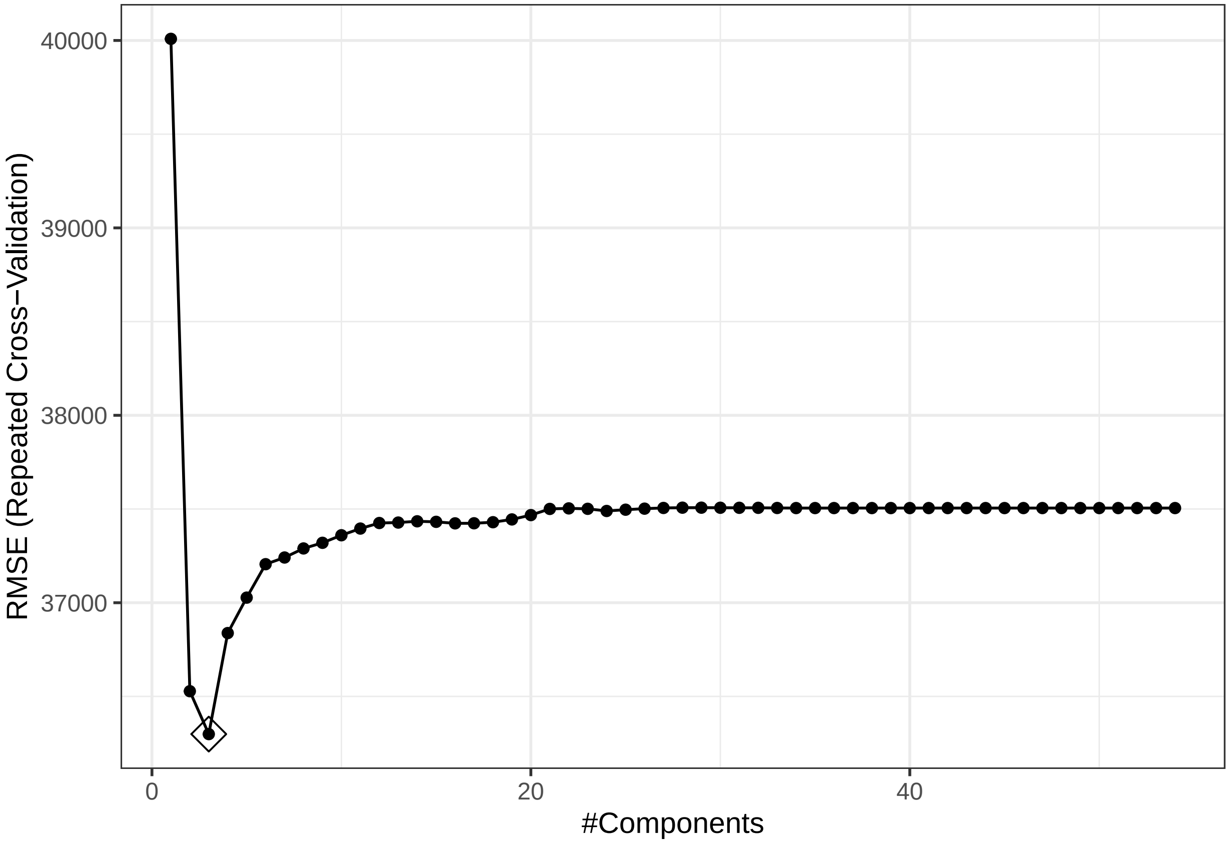


PLS

```
set.seed(1)

pls.fit <- train(x, y,
                 method = "pls",
                 tuneGrid = data.frame(ncomp = 1:54),
                 trControl = ctrl1,
                 scale = TRUE)

as.data.frame(pls.fit$ results ) %>%
  select(ncomp,RMSE) %>%
  filter(RMSE == min(RMSE)) %>%
  knitr::kable()
```

ncomp	RMSE
3	36299.01

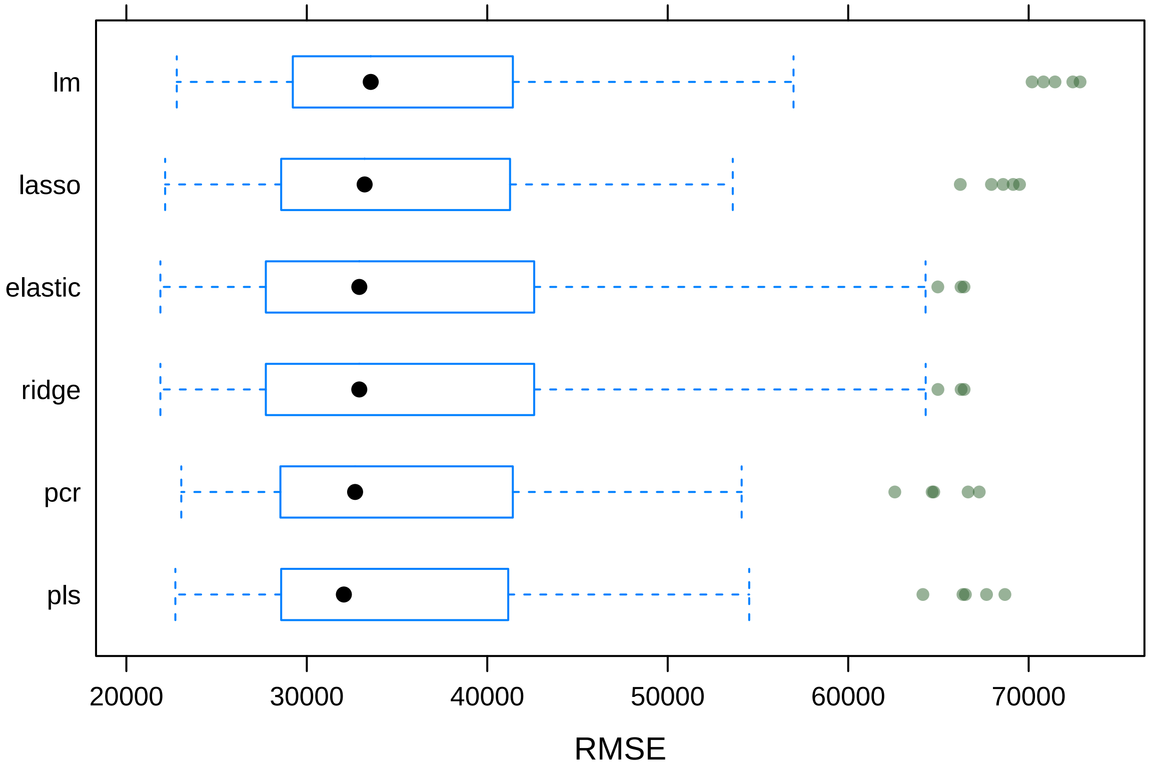
```
ggplot(pls.fit, highlight = TRUE) + theme_bw()
```



```
resamp <- resamples(list(lm = lm.fit,
                         lasso = lasso.fit,
                         ridge = ridge.fit,
                         elastic = enet.fit,
                         pcr = pcr.fit,
                         pls = pls.fit))
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: lm, lasso, ridge, elastic, pcr, pls
## Number of resamples: 50
##
## MAE
##              Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## lm       15923.09 19726.31 21750.77 21721.42 23511.57 27012.23    0
## lasso    15900.83 19058.02 20867.06 20921.76 22641.16 26051.75    0
## ridge   15575.82 19158.45 20901.52 20806.68 22652.30 25275.60    0
## elastic 15575.97 19157.92 20901.23 20806.40 22652.44 25274.02    0
## pcr     16858.57 19896.79 21085.68 21488.85 23098.37 26077.90    0
## pls     16492.29 19393.73 21098.78 21266.92 22956.01 26361.09    0
##
## RMSE
##             Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## lm      22794.36 29286.18 33542.84 37505.04 40606.39 72849.29    0
## lasso   22150.00 28606.22 33204.86 36543.47 40481.70 69494.01    0
## ridge   21886.61 27987.36 32908.06 36278.35 41709.31 66422.74    0
## elastic 21886.67 27987.44 32908.86 36278.34 41711.52 66418.03    0
## pcr     23044.04 28651.95 32675.95 36483.11 40740.88 67262.97    0
## pls     22715.90 28619.89 32053.85 36299.01 40513.81 68684.02    0
##
## Rsquared
##              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lm      0.4274838 0.8087112 0.8407350 0.8046284 0.8795539 0.9174944    0
## lasso   0.4529597 0.8247522 0.8496396 0.8139839 0.8843982 0.9123200    0
## ridge   0.4691210 0.8204653 0.8582029 0.8168154 0.8817302 0.9187655    0
## elastic 0.4691509 0.8204544 0.8582076 0.8168183 0.8817231 0.9187594    0
## pcr     0.4835260 0.8095016 0.8579884 0.8140533 0.8745354 0.9124738    0
## pls     0.4750938 0.8246057 0.8584529 0.8156554 0.8761639 0.9188337    0
```

```
bwplot(resamp, metric = "RMSE")
```



So elastic is the best with smallest MSE.