Lista 2 - Mineração

Victor Alves Dogo Martins, RA: 744878 Ana Beatriz Alves Monteiro, RA: 727838
Larissa Torres, RA: 631914

17-07-2022

Item 1

Para a divisão dos dados entre treino e teste, utilizamos a função initial_split do pacote rsample, com porcentagem dada no enunciado e código dado abaixo:

```
# ITEM 1

set.seed(727838)

library(tidyverse)
library(rsample)
library(glmnet)
library(ggrepel)
library(forcats)
library(knitr)
library(kableExtra)

df <- ISLR::Carseats
df$US <-  as.factor(df$US)
df$Urban <- as.factor(df$Urban)
df$ShelveLoc <- as.factor(df$ShelveLoc)

# Divisao treino e teste

split <- initial_split(df, prop=0.6)

tre <- training(split)
tes <- testing(split)

x_tre <- model.matrix(Sales~., tre)
y_tre <- tre[,1]

x_tes <- model.matrix(Sales~., tes)
y_tes <- tes[,1]
```

Item 2

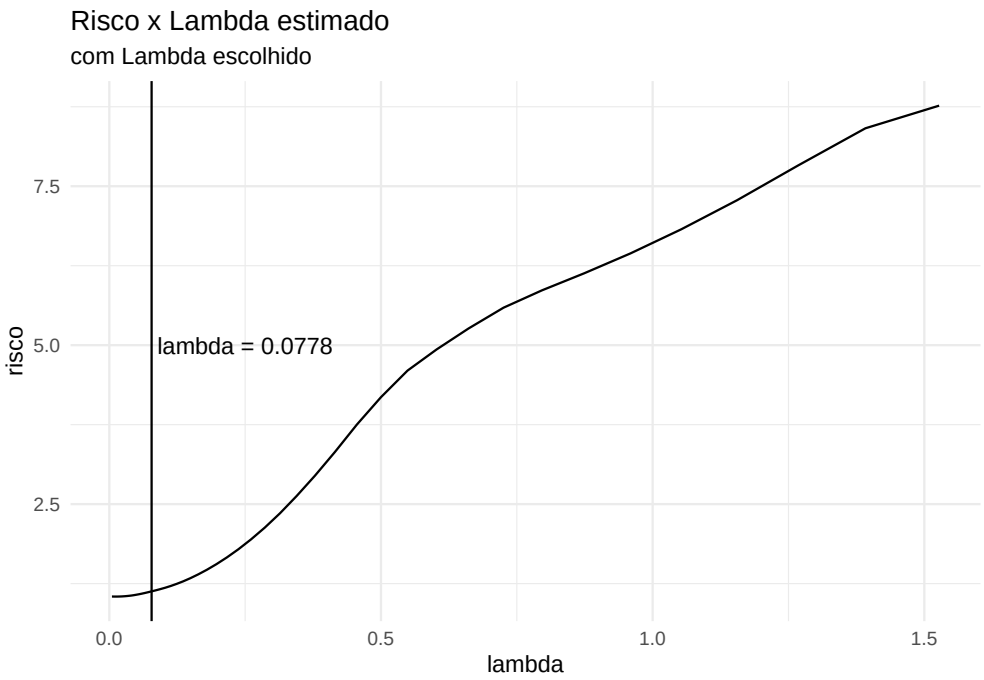
Sabendo que a equação do ajuste via lasso é dada por:

$$\arg\min_{\beta} EQM(g_\beta) + \lambda \sum_{j=1}^{d} \mid \beta_j \mid$$

Se o lambda é igual a zero temos exatamente o estimador de mínimos quadrados, isto é, a penalização de variável não ocorre (beta igual a zero), deixando assim todas as variáveis implementadas no modelo e consequentemente uma alta variância. Em contrapartida, um lambda de valor grande penaliza muito o modelo, e consequentemente aumenta o viés.

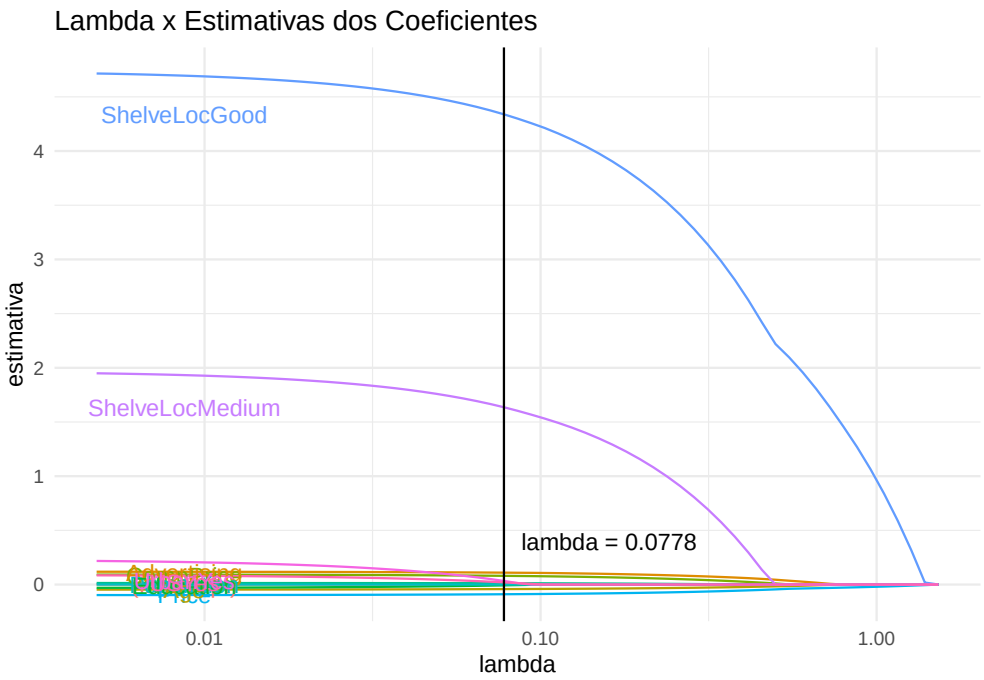
Abaixo, seguem os ajustes via mínimos quadrados e via lasso feitos com o auxílio da função glmnet:

```
# ITEM 2

## Ajuste Minimos Quadrados

ajuste_mq <- glmnet(x_tre, y_tre, alpha=0, lambda=0)

## Ajuste Lasso

cv_lasso <- cv.glmnet(x_tre, y_tre, alpha=1)
ajuste_lasso <- glmnet(x_tre, y_tre, alpha=1, lambda = cv_lasso$lambda.1se)

## Erro x Lambda Lasso

tibble(
  lambda=cv_lasso$lambda,
  risco=cv_lasso$cvm
) |>
  ggplot()+
  aes(x=lambda, y=risco)+
  geom_line()+
  geom_vline(xintercept = cv_lasso$lambda.1se)+
  annotate(geom = 'text', y=5, x=0.25,
           label=paste0('lambda = ', round(ajuste_lasso$lambda,5)))+
  theme_minimal()+
  labs(title='Risco x Lambda estimado',
       subtitle = 'com Lambda escolhido')
```




```
## Apresentando melhor lambda (cv_lasso$lambda.1se)

lambdas <- cv_lasso$glmnet.fit$lambda

lam <- lambdas |>
  as.data.frame() |>
  mutate(penalty=names(cv_lasso$glmnet.fit$a0)) %>%
  rename(lambda=1)

results <- cv_lasso$glmnet.fit$beta |>
  as.matrix() |>
  as.data.frame() |>
  rownames_to_column() |>
  gather(penalty,coefficients,-rowname) |>
  left_join(lam)

result_labels <- results |>
  group_by(rowname) |>
  filter(lambda==ajuste_lasso$lambda) |>
  ungroup()

ggplot()+
  geom_line(data=results, aes(lambda, coefficients,
                              group=rowname, color=rowname),
            show.legend = FALSE)+
  scale_x_log10()+
  geom_text(data=result_labels, aes(0.01, coefficients,
                                    label=rowname, color=rowname),
            nudge_x=-.06, show.legend = FALSE)+
  geom_vline(xintercept = ajuste_lasso$lambda)+
  annotate(geom = 'text', y=0.4, x=0.16,
```

```
          label=paste0('lambda = ', round(ajuste_lasso$lambda,5)))+
  theme_minimal()+
  labs(y='estimativa',
       title='Lambda x Estimativas dos Coeficientes')
```



Assim, com o auxílio da função "cv.glmnet", podemos escolher um lambda que penalize nossas covariáveis da melhor forma: o lambda escolhido é o "lambda.1se" igual a 0.0778, que mantém o valor absoluto do desvio padrão do risco estimado menor do que 1. Dessa forma, teremos um modelo parcimonioso que não será tão penalizado quanto aquele com o lambda com risco mínimo, mas que possuirá um poder preditivo tão bom quanto, equilibrando o viés e a variância do modelo. Com o valor de lambda escolhido, temos que as variáveis de localização boa e média do produto na prateleira demonstraram valores consideravelmente maiores do que as outras estimativas.

Item 3

Após o ajuste do modelo via método de Mínimos Quadrados e via Lasso, temos as seguintes estimativas dos coeficientes:

```
# ITEM 3

coef_mq <- coefficients(ajuste_mq) |>
  round(3)

coef_lasso <- coefficients(ajuste_lasso) |>
  round(3)

## Apresentando coeficientes
```

```
tibble(
  var=names(coef_mq[,1]),
  mq=coef_mq[,1],
  lasso=coef_lasso[,1]
) |> slice(-2) |>
  kable('latex', align='ccc',
        caption = 'Estimativas de Coeficientes para Mínimos Quadrados e Lasso',
        col.names=c('Variável', 'Mínimos Quadrados', 'Lasso')) |>
  kable_styling(position="center",
                latex_options="HOLD_position")
```

Table 1: Estimativas de Coeficientes para Mínimos Quadrados e Lasso

Variável	Mínimos Quadrados	Lasso
(Intercept)	6.018	6.672
CompPrice	0.093	0.080
Income	0.014	0.012
Advertising	0.119	0.110
Population	0.000	0.000
Price	-0.098	-0.089
ShelveLocGood	4.741	4.339
ShelveLocMedium	1.970	1.636
Age	-0.047	-0.042
Education	-0.032	-0.008
UrbanYes	0.230	0.033
USYes	0.089	0.021

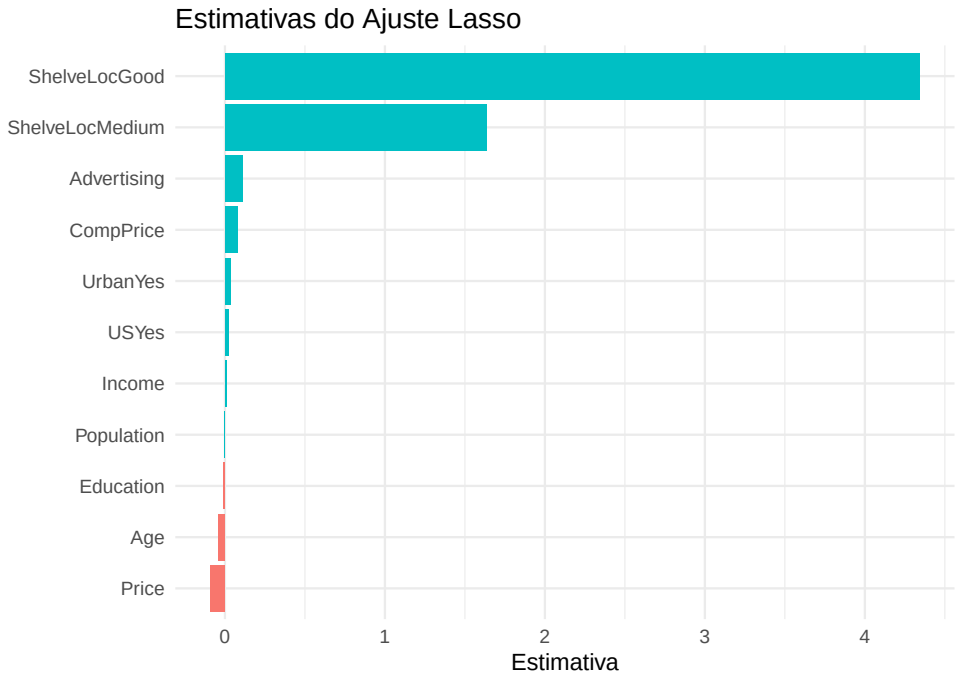
Ao comparar as estimativas entre cada método é notável que todos os coeficientes estimados para o Lasso possuem valores inferiores ao do Mínimos Quadrados. Os coeficientes via lasso tendem a se aproximar de zero, principalmente as variáveis Education, UrbanYes, USYes.

Para ilustrar melhor os resultados da tabela acima foi utilizado o auxílio dos gráficos de barras horizontais:

```
## Barras coeficientes minimos quadrados

tibble(
  var=names(coef_mq[,1]),
  val=coef_mq[,1]
) |>
  mutate(sinal=ifelse(val<0, 'negativo', 'positivo'),
         var=fct_reorder(var, val)) |>
  arrange(val) |>
  filter(var!='(Intercept)') |>
  ggplot()+
  aes(x=val, y=var, fill=sinal)+
  geom_bar(stat='identity', show.legend = FALSE)+
  theme_minimal()+
  labs(x='Estimativa', y='',
       title='Estimativas do Ajuste de Mínimos Quadrados')
```

Através das estimativas do Ajuste de Mínimos Quadrados, temos que a boa localização do produto é a variável que apresentou a maior estimativa do coeficiente para o modelo e, contribui fortemente para a variável resposta, ou seja, quanto mais bem alocado estiver o produto nas prateleiras das lojas, mais poder de venda ele tem, em seguida a média localização do produto na prateleira também contribui positivamente para a venda. Em contrapartida, a variável que apresentou o menor coeficiente foi a de preço, concluindo que o preço é o fator que menos contribui para a venda das cadeirinhas infantis. População é uma variável neutra (beta estimado igual a zero), o tamanho populacional não aumenta nem diminui a venda do produto.

```
## Barras coeficientes lasso

tibble(
  var=names(coef_lasso[,1]),
  val=coef_lasso[,1]
) |>
  mutate(sinal=ifelse(val<0, 'negativo', 'positivo'),
         var=fct_reorder(var, val)) |>
  arrange(val) |>
  filter(var!='(Intercept)') |>
  ggplot()+
  aes(x=val, y=var, fill=sinal)+
  geom_bar(stat='identity', show.legend = FALSE)+
  theme_minimal()+
  labs(x='Estimativa', y='',
       title='Estimativas do Ajuste Lasso')
```



Para o método Lasso os betas trazem informações equivalentes ao Mínimos Quadrados, a boa localização do produto é fundamental para a venda, e em seguida a média localização também contribui positivamente para a venda de cadeirinhas. A população novamente é indiferente para a venda do produto. E por fim, a medida que o preço aumenta, menos temos a venda de cadeirinhas.

Item 4

```
# ITEM 4

funcao_risco <- function(y_pred, y_obs){
  w <- (y_pred-y_obs)^2
  sigma <- var(w)
  risco <- mean(w)
  liminf <- risco - (2*sqrt((1/length(w))*sigma))
  limsup <- risco + (2*sqrt((1/length(w))*sigma))

  return(tibble(risco, liminf, limsup))
}

y_pred_mq <- predict(ajuste_mq, x_tes)
y_pred_lasso <- predict(ajuste_lasso, x_tes)

risco_mq <- funcao_risco(y_pred_mq, y_tes)
risco_lasso <- funcao_risco(y_pred_lasso, y_tes)

tibble(
  Estimativa=c('Risco', 'Limite Inferior', 'Limite Superior'),
  `Mínimos Quadrados`=unlist(c(risco_mq)),
```

```
  `Lasso`=unlist(c(risco_lasso))
) |>
  kable('latex', align='ccc',
        caption = 'Risco e Intervalos de Confiança para Lasso e Mínimos Quadrados',
        col.names=c('Variável', 'Mínimos Quadrados', 'Lasso')) |>
  kable_styling(position="center",
                latex_options="HOLD_position")
```

Table 2: Risco e Intervalos de Confiança para Lasso e Mínimos Quadrados

Variável	Mínimos Quadrados	Lasso
Risco	1.1815323	1.1869124
Limite Inferior	0.9231082	0.9441733
Limite Superior	1.4399564	1.4296515

Realizando a verificação utilizando o conjunto de teste percebemos que o modelo de mínimos quadrados apresenta risco inferior ao do lasso. Apresentando portanto, melhores resultados, já que os valores previstos pelo modelo se aproximam mais dos reais.

Por outro lado é importante ressaltarmos que o intervalo de confiança do lasso possui menor amplitude e limite superior menor do que o de mínimos quadrados.

Item 5

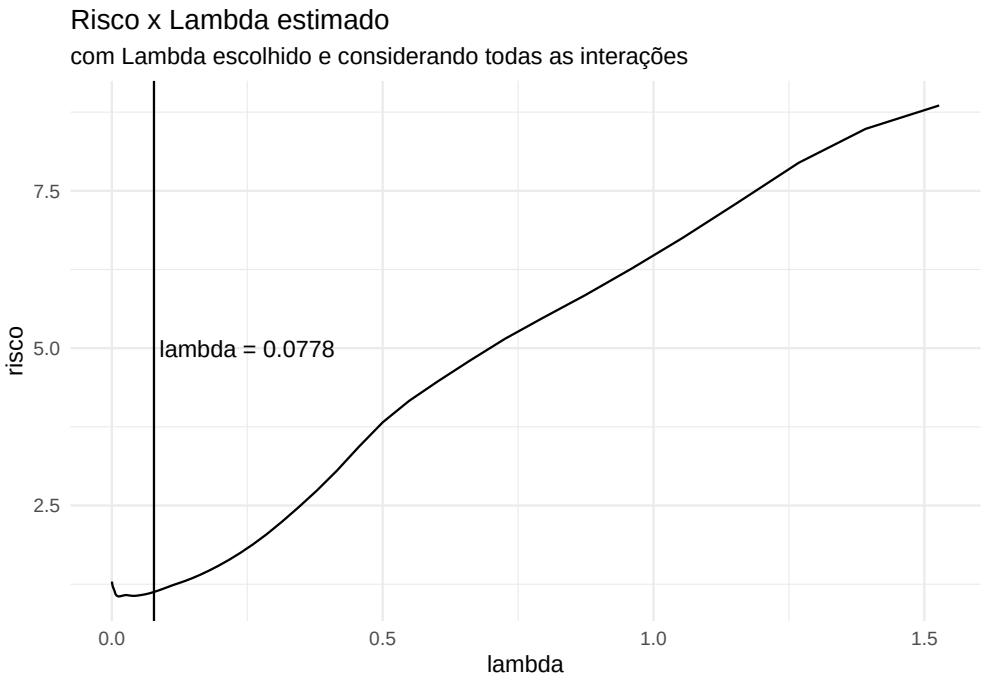
Para o último item desta lista, iremos realizar todos os mesmos procedimentos anteriores, mas agora pensando num contexto de modelos que levam em conta todas as interações. Isso é feito através da fórmula "y~.^2", como demonstrado abaixo no comando model.matrix

```
# ITEM 5

x_tre2 <- model.matrix(Sales~.^2, tre)

x_tes2 <- model.matrix(Sales~.^2, tes)

## Realizando ajuste para todas as variáveis

## Ajuste Minimos Quadrados

ajuste_mq2 <- glmnet(x_tre2, y_tre, alpha=0, lambda=0)

## Ajuste Lasso

cv_lasso2 <- cv.glmnet(x_tre2, y_tre, alpha=1)
ajuste_lasso2 <- glmnet(x_tre2, y_tre, alpha=1, lambda = cv_lasso2$lambda.1se)
```

Abaixo, visualizaremos de que forma os lambdas no ajuste com penalização via lasso se comportaram para esta situação:

```
## Erro x Lambda Lasso
```

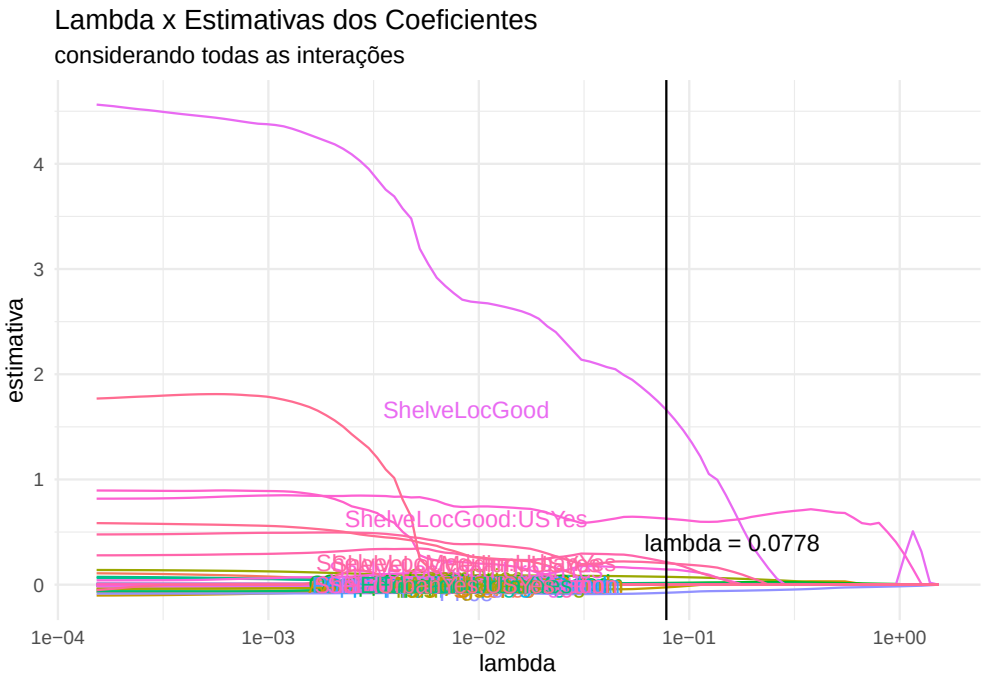
```
tibble(
  lambda=cv_lasso2$lambda,
  risco=cv_lasso2$cvm
) |>
  ggplot()+
  aes(x=lambda, y=risco)+
  geom_line()+
  geom_vline(xintercept = cv_lasso2$lambda.1se)+
  annotate(geom = 'text', y=5, x=0.25,
           label=paste0('lambda = ', round(ajuste_lasso2$lambda,5)))+
  theme_minimal()+
  labs(title='Risco x Lambda estimado',
       subtitle = 'com Lambda escolhido e considerando todas as interações')
```



Podemos ver que o lambda escolhido é o mesmo que anteriormente: já que escolhemos o lambda cujos riscos estimados possuem desvio padrão menor do que 1 (ao invés do lambda com risco mínimo), como feito anteriormente nesta lista, não é um comportamento exatamente inesperado de nossos dados. Abaixo, vejamos como as estimativas dos coeficientes se comportaram de acordo com os valores de lambda:

```
## Apresentando melhor lambda (cv_lasso$lambda.1se)

lambdas <- cv_lasso2$glmnet.fit$lambda

lam <- lambdas |>
  as.data.frame() |>
  mutate(penalty=names(cv_lasso2$glmnet.fit$a0)) %>%
  rename(lambda=1)

results <- cv_lasso2$glmnet.fit$beta |>
  as.matrix() |>
```

```
  as.data.frame() |>
  rownames_to_column() |>
  gather(penalty,coefficients,-rowname) |>
  left_join(lam)

result_labels <- results |>
  group_by(rowname) |>
  filter(lambda==ajuste_lasso2$lambda) |>
  ungroup()

ggplot()+
  geom_line(data=results, aes(lambda, coefficients,
                              group=rowname, color=rowname), show.legend = FALSE)+
  scale_x_log10()+
  geom_text(data=result_labels, aes(0.01, coefficients,
                                    label=rowname, color=rowname),
            nudge_x=-.06, show.legend = FALSE)+
  geom_vline(xintercept = ajuste_lasso2$lambda)+
  annotate(geom = 'text', y=0.4, x=0.16,
           label=paste0('lambda = ', round(ajuste_lasso2$lambda,5)))+
  theme_minimal()+
  labs(y='estimativa',
       title='Lambda x Estimativas dos Coeficientes',
       subtitle = 'considerando todas as interações')
```



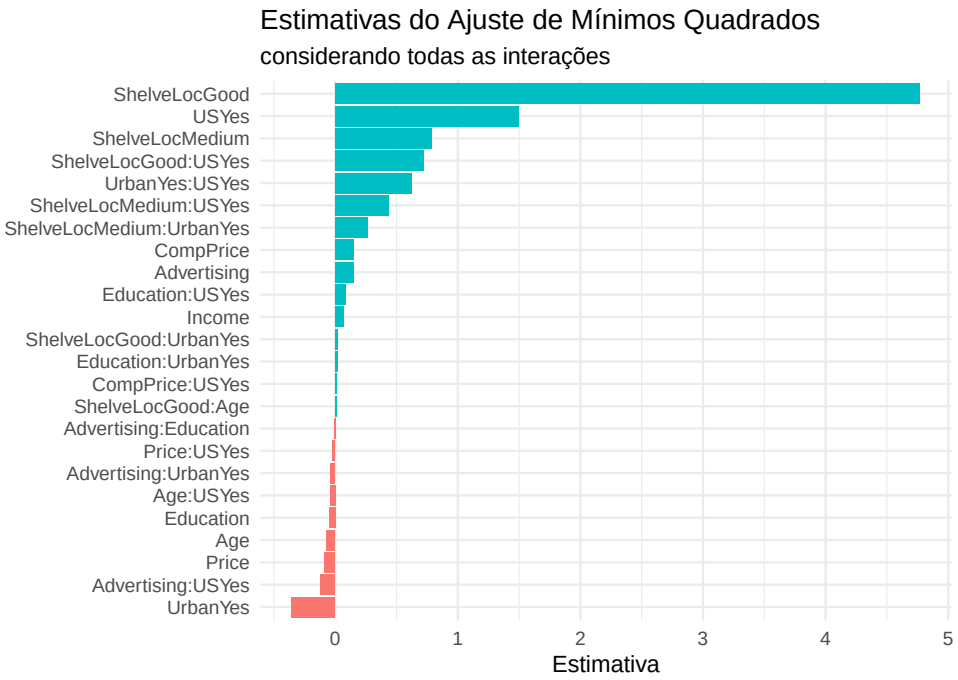
Houveram mais variáveis penalizadas desta vez, além do maior destaque para a localização boa do produto na prateleira. Para compreendermos melhor quais foram as variáveis mais importantes, iremos visualizar todas as estimativas numa tabela e visualmente para os dois ajustes realizados. Para uma melhor visualização, foram filtrados apenas coeficientes com valor absoluto maior do que 0.01.

```r
## Apresentando coeficientes

coef_mq2 <- coefficients(ajuste_mq2) |>
  round(3)
coef_mq2 <- coef_mq2[,1]
coef_mq3 <- coef_mq2[abs(coef_mq2)>=0.01]

coef_lasso2 <- coefficients(ajuste_lasso2) |>
  round(3)
coef_lasso2 <- coef_lasso2[,1]
coef_lasso3 <- coef_lasso2[abs(coef_lasso2)>=0.01]

tibble(
  var=names(coef_mq2),
  mq=coef_mq2,
  lasso=coef_lasso2
) |> slice(-2) |>
  kable('latex', align='ccc',
        caption = 'Coeficientes para Mínimos Quadrados e Lasso com todas as interações',
        col.names=c('Variável', 'Mínimos Quadrados', 'Lasso')) |>
  kable_styling(position="center",
                latex_options="HOLD_position")
```

Table 3: Coeficientes para Mínimos Quadrados e Lasso com todas as interações

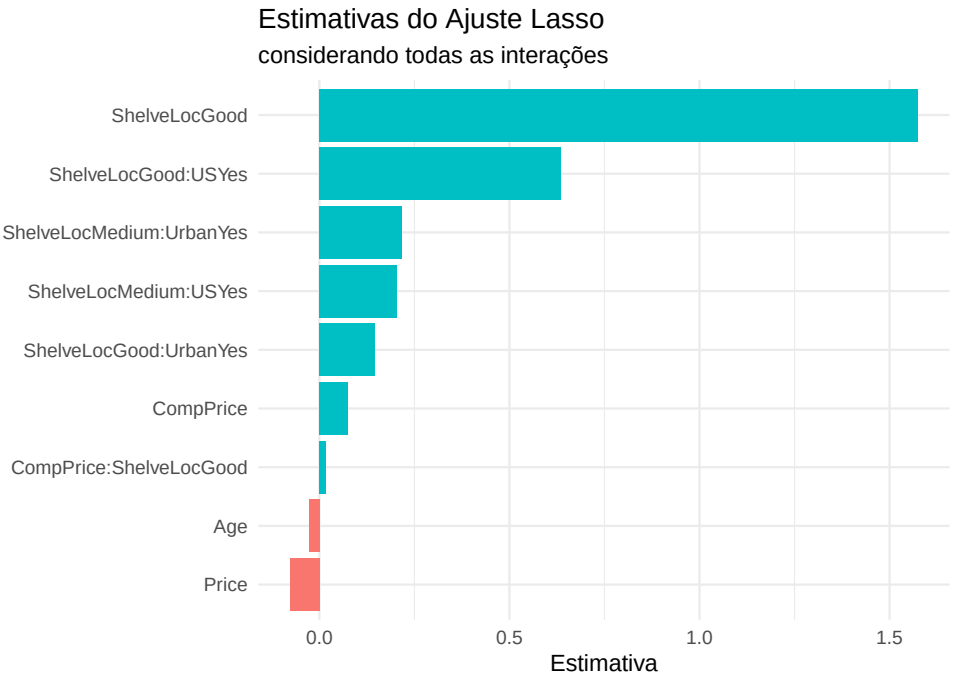
Variável	Mínimos Quadrados	Lasso
(Intercept)	0.407	7.316
CompPrice	0.149	0.074
Income	0.071	0.000
Advertising	0.148	0.000
Population	0.003	0.000
Price	-0.086	-0.078
ShelveLocGood	4.771	1.575
ShelveLocMedium	0.789	0.000
Age	-0.071	-0.027
Education	-0.052	0.000
UrbanYes	-0.355	0.000
USYes	1.494	0.000
CompPrice:Income	0.000	0.000
CompPrice:Advertising	-0.002	0.000
CompPrice:Population	0.000	0.000
CompPrice:Price	0.000	0.000
CompPrice:ShelveLocGood	-0.004	0.017
CompPrice:ShelveLocMedium	0.004	0.007
CompPrice:Age	0.000	0.000
CompPrice:Education	-0.001	0.000
CompPrice:UrbanYes	0.006	0.000
CompPrice:USYes	0.012	0.000
Income:Advertising	0.001	0.001
Income:Population	0.000	0.000
Income:Price	0.000	0.000
Income:ShelveLocGood	-0.008	0.000
Income:ShelveLocMedium	0.009	0.009
Income:Age	0.000	0.000
Income:Education	0.000	0.000
Income:UrbanYes	0.002	0.000
Income:USYes	-0.002	0.000
Advertising:Population	0.000	0.000
Advertising:Price	0.002	0.000
Advertising:ShelveLocGood	-0.006	0.000
Advertising:ShelveLocMedium	-0.006	0.000
Advertising:Age	0.003	0.000
Advertising:Education	-0.011	0.000
Advertising:UrbanYes	-0.039	0.001
Advertising:USYes	-0.120	0.000
Population:Price	0.000	0.000
Population:ShelveLocGood	-0.001	0.000
Population:ShelveLocMedium	-0.002	0.000
Population:Age	0.000	0.000
Population:Education	0.000	0.000
Population:UrbanYes	0.000	0.000
Population:USYes	0.001	0.000
Price:ShelveLocGood	0.003	0.000
Price:ShelveLocMedium	-0.001	0.000
Price:Age	0.000	0.000
Price:Education	0.000	0.000
Price:UrbanYes	-0.009	0.000
Price:USYes	-0.023	0.000
ShelveLocGood:Age	0.011	0.000
ShelveLocMedium:Age	0.004	0.000
ShelveLocGood:Education	-0.005	0.000

```r
## Graficos dos coeficientes de minimos quadrados

tibble(
  var=names(coef_mq3),
  val=coef_mq3
) |>
  mutate(sinal=ifelse(val<0, 'negativo', 'positivo'),
         var=fct_reorder(var, val)) |>
  arrange(val) |>
  filter(var!='(Intercept)') |>
  ggplot()+
  aes(x=val, y=var, fill=sinal)+
  geom_bar(stat='identity', show.legend = FALSE)+
  theme_minimal()+
  labs(x='Estimativa', y='',
       title='Estimativas do Ajuste de Mínimos Quadrados',
       subtitle='considerando todas as interações')
```




```r
## Graficos dos coeficientes de lasso

tibble(
  var=names(coef_lasso3),
  val=coef_lasso3
) |>
  mutate(sinal=ifelse(val<0, 'negativo', 'positivo'),
         var=fct_reorder(var, val)) |>
  arrange(val) |>
  filter(var!='(Intercept)') |>
  ggplot()+
  aes(x=val, y=var, fill=sinal)+
```

```
  geom_bar(stat='identity', show.legend = FALSE)+
  theme_minimal()+
  labs(x='Estimativa', y='',
       title='Estimativas do Ajuste Lasso',
       subtitle='considerando todas as interações')
```



A princípio, chama a atenção a quantidade de variáveis que foram penalizadas e resultaram em estimativa igual à 0 no caso do ajuste via lasso. Para o ajuste via mínimos quadrados, temos valores muito menores do que os obtidos para quando não consideramos todas as interações duas a duas. Isso indica um possível poder preditivo menor do que anteriormente, já que damos peso igual para todas as covariaveis, sem priorizar as mais importantes para o contexto.

Fora isso, temos que a localização boa na estante e lojas localizadas nos EUA foram as variáveis com maiores estimativas, demonstrando serem importantes neste contexto novamente. Além disso, a interação destas duas variáveis também recebem destaque, bem como lojas localizadas em regiões urbanas.

Por fim, vejamos as estimativas dos riscos para os modelos com todas as interações duas a duas:

```
## Calculando risco estimado

y_pred_mq2 <- predict(ajuste_mq2, x_tes2)
y_pred_lasso2 <- predict(ajuste_lasso2, x_tes2)

risco_mq2 <- funcao_risco(y_pred_mq2, y_tes)
risco_lasso2 <- funcao_risco(y_pred_lasso2, y_tes)

tibble(
  Estimativa=c('Risco', 'Limite Inferior', 'Limite Superior'),
  `Mínimos Quadrados`=unlist(c(risco_mq2)),
  `Lasso`=unlist(c(risco_lasso2))
) |>
```

```
kable('latex', align='ccc',
      caption = 'Risco e Intervalos de Confiança considerando todas as interações',
      col.names=c('Variável', 'Mínimos Quadrados', 'Lasso')) |>
kable_styling(position="center",
              latex_options="HOLD_position")
```

Table 4: Risco e Intervalos de Confiança considerando todas as interações

Variável	Mínimos Quadrados	Lasso
Risco	1.575369	1.2080729
Limite Inferior	1.257929	0.9333602
Limite Superior	1.892809	1.4827856

Temos que tanto os intervalos de 95% de confiança como os riscos pontuais estimados encontram-se numa faixa de valores maior. Com isso, a inclusão das interações duas a duas não foi vantajosa nem para o ajuste via lasso, nem para o ajuste via mínimos quadrados.