Lista 1 Mineração de dados

Luben Miguel, Luiz Piccin, Vinicius Hideki

15/09/2021

Exercício 1:

Utilizando a linguagem R, primeiramente, importemos os dados e realizemos um pequeno pré processamento:

```
setwd("~/estatistica_UFSCAR/Mineracao de dados/listas/lista_1")
houses_to_rent = read.csv("houses_to_rent_v2.csv")

# selecionando os imoveis apenas em sao paulo, rio de janeiro e belo horizonte
houses_to_rent %<>% filter(city %in% c("SÃ£o Paulo", "Rio de Janeiro", "Belo Horizonte"))
# verificando se há NA
sum(is.na(houses_to_rent))
```

```
## [1] 0
```

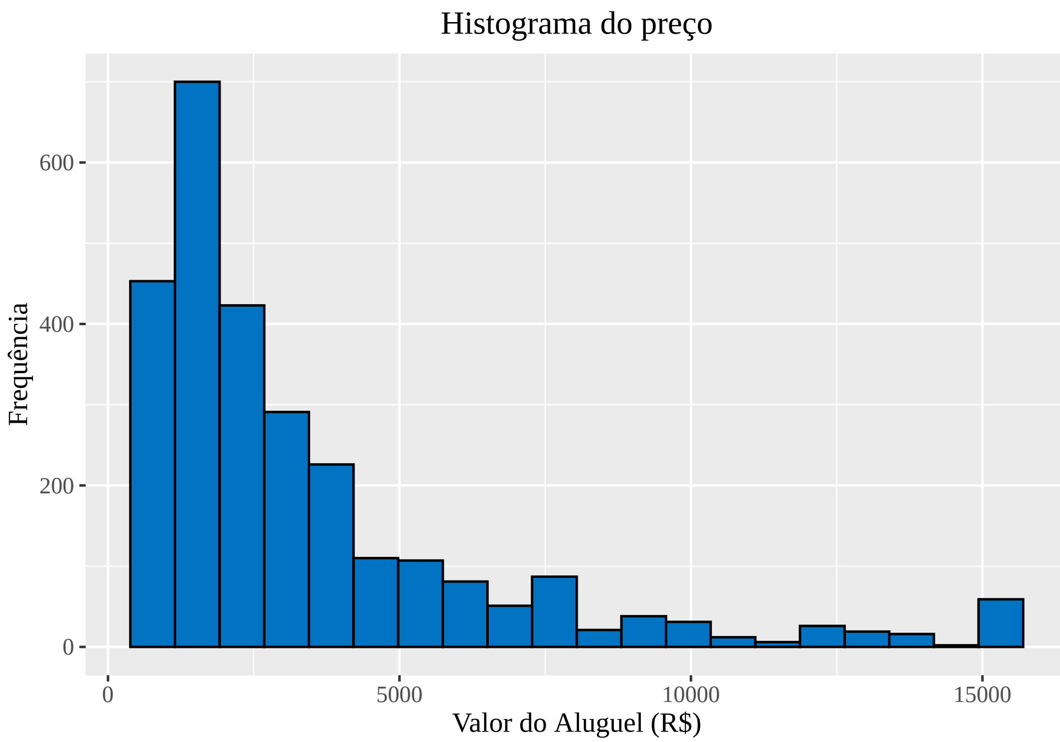
Notamos que não há nenhum valor faltante no banco de dados. Visualiza-se as 6 primeiras linhas do banco de dados como a seguir:

```
# nao ha NA
# mudando floor, animal, furniture e city para fator:
houses_to_rent %<>% mutate_if(is.character, as.factor) %>%
  mutate(city = recode_factor(city, "SÃ£o Paulo" = "São Paulo"))
colnames(houses_to_rent)[9] = "rent.amount"
head(houses_to_rent)
```

city	area	rooms	bathroom	parking.spaces	floor	animal	furniture	rent.amount
Rio de Janeiro	72	2	1	0	7	acept	not furnished	8000
Rio de Janeiro	35	1	1	0	2	acept	furnished	2300
Rio de Janeiro	88	2	3	1	9	not acept	furnished	3500
Rio de Janeiro	56	2	1	0	8	acept	not furnished	1220
Belo Horizonte	42	1	1	1	17	not acept	furnished	2690
Rio de Janeiro	90	3	2	1	7	acept	not furnished	1800

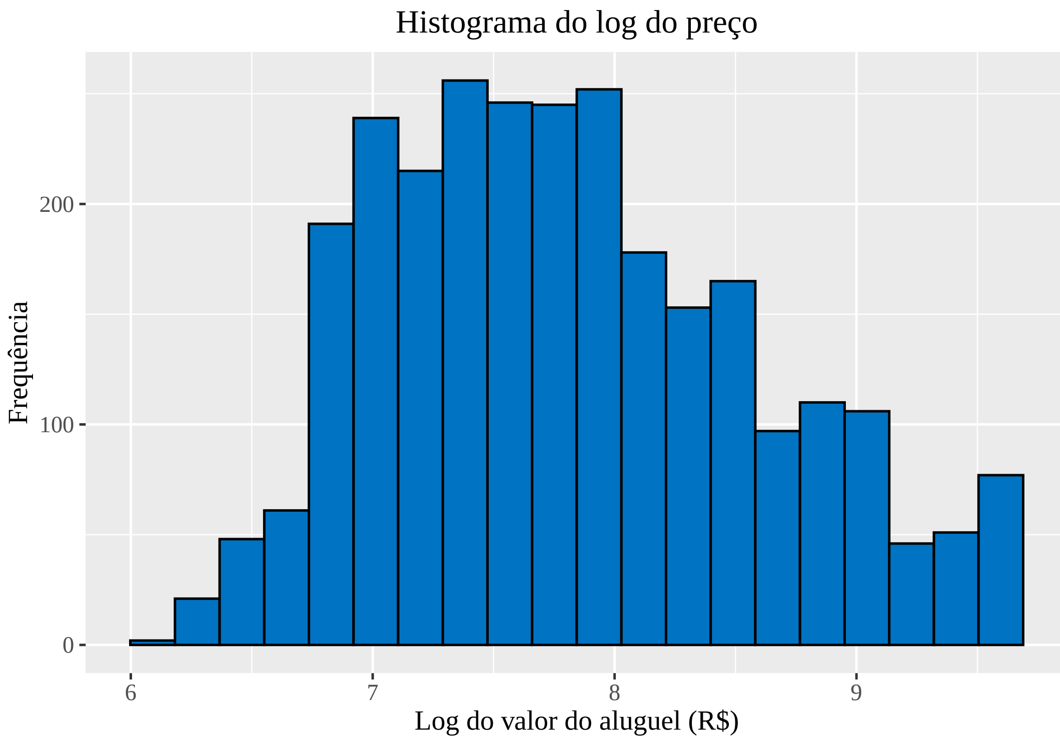
Podemos analisar como se distribui a variável resposta que nos dá o valor do aluguel através de um simples histograma do preço do apartamento ou casa:

```r
houses_to_rent %>%
  ggplot(aes(x = rent.amount)) +
  geom_histogram(fill = "#0073C2FF", color = "black", bins = 20)+
  labs(x = "Valor do Aluguel (R$)",
       y = "Frequência",
       title = "Histograma do preço")+
  theme(text = element_text(size = 12,
                            family ="serif"),
        plot.title = element_text(hjust = 0.5))
```



Nota-se a presença de certos valores extremos, tendo certa assimetria. Essa grande variação na variável de interesse pode fazer com que os hiperparâmetros (λ) do lasso e do ridge fiquem inflado. De modo a evitar isso faz-se a aplicação de uma transformação com log, na expectativa de suavizar a assimetria e diminuir a presença de valores extremos, obtendo:

```r
houses_to_rent %>%
  mutate(rent.amount = log(rent.amount)) %>%
  ggplot(aes(x = rent.amount)) +
  geom_histogram(fill = "#0073C2FF", color = "black", bins = 20)+
  labs(x = "Log do valor do aluguel (R$)",
       y = "Frequência",
       title = "Histograma do log do preço")+
  theme(text = element_text(size = 12,
                            family ="serif"),
        plot.title = element_text(hjust = 0.5))
```



Percebe-se uma melhora na simetria dos dados, com uma diminuição do range alto averiguado anteriormente, o que pode melhorar nossos futuros ajustes. Podemos agora verificar e realizar uma pequena exploração das variáveis de nosso banco de dados. Primeiramente, a dimensão do banco de dados:

```
dim(houses_to_rent)
```

```
## [1] 2759    9
```

Verifica-se também a configuração dos níveis em cada fator:

```
dados_fac = houses_to_rent %>% dplyr::select(where(is.factor))
# niveis
dados_fac %>%
  purrr::map(levels)
```

```
## $city
## [1] "Belo Horizonte" "Rio de Janeiro"
##
## $floor
##  [1] "-"   "1"   "10"  "11"  "12"  "13"  "14"  "15"  "16"  "17"  "18"  "19"
## [13] "2"   "20"  "21"  "22"  "24"  "25"  "26"  "29"  "3"   "301" "4"   "5"
## [25] "6"   "7"   "8"   "9"
##
## $animal
## [1] "acept"     "not acept"
##
## $furniture
## [1] "furnished"     "not furnished"
```

```r
# numero de niveis
dados_fac %>%
  purrr::map(nlevels)
```

```
## $city
## [1] 2
##
## $floor
## [1] 28
##
## $animal
## [1] 2
##
## $furniture
## [1] 2
```

Percebe-se que a variável *floor* tem 34 níveis, enquanto o restante das covariáveis categóricas tem no máximo 2 níveis. A distribuição de cada nível é dada da seguinte forma:

```r
dados_fac %>%
  purrr::map(table)
```

```
## $city
##
## Belo Horizonte Rio de Janeiro
##           1258           1501
##
## $floor
##
##   -   1  10  11  12  13  14  15  16  17  18  19   2  20  21  22  24  25  26  29
## 449 293 111  83  59  30  25  26  11   9   8   4 363   8   6   2   1   1   1   1
##   3 301   4   5   6   7   8   9
## 322   1 270 186 135 136 126  92
##
## $animal
##
##     acept not acept
##      2136       623
##
## $furniture
##
##     furnished not furnished
##           583          2176
```

Verifica-se a presença de níveis com pequena frequência de observações na variável *floor*, tendo um dos níveis com valor 301, que é um valor absurdo (o Burj Khalifa tem 161 andares e é o maior arranha céu do mundo). Pode-se remover essa observação ou corrigi-la pelo possível andar, que no caso, deve ser 31. Optaremos pela segunda opção:

```r
houses_to_rent %<>%
  mutate(floor = recode_factor(floor, "301" = "31"))
```

Visto que o fator piso tem muitos níveis, uma alternativa de se evitar o aumento da dimensionalidade do problema é considerar tal variável como quantitativa. O nível "-" quer dizer que o imóvel seria possivelmente uma casa, algo que se pode fazer é considerar *floor* como uma variável quantitativa, adicionando uma nova variável que reporta se o imóvel é uma casa ou um apartamento. Além disso, já deixaremos a variável *rent.amount* transformada com log:

```
# criando um novo houses_to_rent
rent_data_reg = houses_to_rent %>%
  mutate(house = as.factor(ifelse(floor == "-", "casa", "ap")),
    floor = as.numeric(paste(recode_factor(floor, "-" = "0"))),
    rent.amount = log(rent.amount))
rent_data_reg %>% head()
```

city	area	rooms	bathroom	parking.spaces	floor	animal	furniture	rent.amount	house
Rio de Janeiro	72	2	1	0	7	acept	not furnished	8.987197	ap
Rio de Janeiro	35	1	1	0	2	acept	furnished	7.740664	ap
Rio de Janeiro	88	2	3	1	9	not acept	furnished	8.160518	ap
Rio de Janeiro	56	2	1	0	8	acept	not furnished	7.106606	ap
Belo Horizonte	42	1	1	1	17	not acept	furnished	7.897297	ap
Rio de Janeiro	90	3	2	1	7	acept	not furnished	7.495542	ap

Item (a)
Assim, tratado os fatores, inicialmente parece interessante dividir em 70% para 30%, visto que teremos cerca de 2000 observações nos dados de teste, o que parece mais que o suficiente para se ter uma boa precisão estimação do risco. Fixaremos como semente o valor de 975:

```
set.seed(975, sample.kind="Rounding")
```

```
## Warning in set.seed(975, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
# fazendo a particao pelo caret
split = initial_split(rent_data_reg, prop = 0.7)
train_data = training(split)
test_data = testing(split)
```

Tomando a matriz de covariáveis e y do conjunto de treinamento:

```
# matriz de preditores
x_train <- model.matrix(rent.amount ~ ., train_data)[,-1]

# vetor resposta
y_train <- train_data$rent.amount
```

Item (b):
(i) Treinando primeiramente um modelo de regressão linear e averiguando seus coeficientes:

```
mod_reg = glmnet(x_train, y_train, alpha = 0, lambda = 0)
coef(mod_reg)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                                   s0
## (Intercept)              6.694863e+00
## cityRio de Janeiro       3.441516e-01
## area                    -1.435135e-05
## rooms                    1.912253e-01
## bathroom                 2.483509e-01
## parking.spaces           8.307309e-02
## floor                    1.448997e-02
## animalnot acept          2.667474e-02
## furniturenot furnished -3.439137e-01
## housecasa                1.192580e-01
```

```
length(coef(mod_reg))
```

```
## [1] 10
```

(ii) Ajustando uma regressão com regularização lasso, tendo primeiramente o valor de λ obtido por validação cruzada:

```
# tomando os valores de coeficiente do lasso pela validacao cruzada:
cv.lasso = cv.glmnet(x_train, y_train, alpha = 1, stantardize = T)

# vendo o melhor valor de lambda
cv.lasso$lambda.min
```

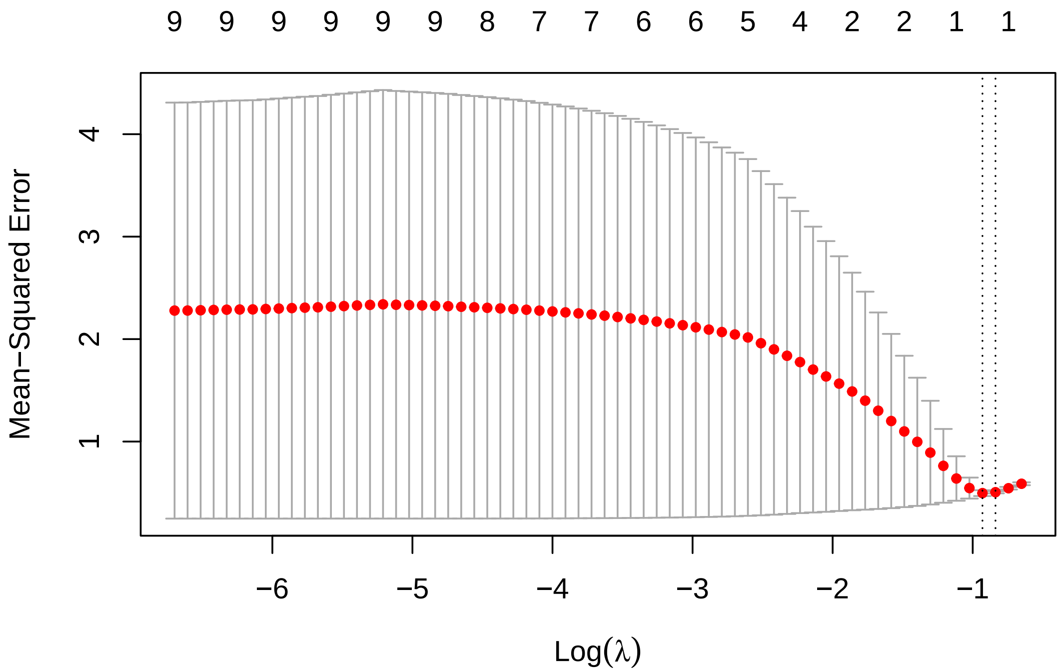
```
## [1] 0.3944433
```

```
cv.lasso$lambda.1se
```

```
## [1] 0.4329011
```

Podemos averiguar qual o comportamento do λ com relação ao erro quadratico médio:

```
plot(cv.lasso)
```



Antes de estudar a figura anterior, salienta-se que *lambda.min* é o valor de λ que dá o erro quadrático médio mínimo da validação cruzada. Enquanto *lambda.1se* é o valor de λ que fornece o modelo mais regularizado de forma que o erro da validação cruzada esteja dentro de um erro padrão mínimo.Sendo assim, ao observar a Figura, percebe-se que no lado esquerdo tem-se o erro quadrático médio encontrado na validação cruzada. Na parte inferior, tem-se o log (na base e) dos possíveis λ. E na parte superior, há o número de coeficientes estimados que foram diferentes de 0. Ou seja, o número de variáveis selecionadas pelo lasso. Para o *lambda.min*, que fora $\lambda = 0.00602$ e portanto $\log(0.00602) = -5.111$, notou-se que 9 covariáveis foram selecionadas. Já para o *lambda.1se*, fora obtido $\lambda = 0.038$ com $\log(0.038) = -3.27$.Desse modo, o *lambda.1se* foi mais rigoroso que o mínimo, selecionando menos covariáveis. Utilizando aquilo que foi aprendido em aula, isto é, o λ que minimiza o erro quadrático estimado, escolhe-se o λ de 0.006029787. Salienta-se também que o λ mínimo obtido tem apenas a covariável "área" como não significativa:

```r
# ajustando o modelo
mod_lasso <- glmnet(x_train, y_train, alpha = 1, lambda = cv.lasso$lambda.min)
# coeficientes
coef(mod_lasso)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                                s0
## (Intercept)             7.6059044
## cityRio de Janeiro      .
## area                    .
## rooms                   .
## bathroom                0.1026446
## parking.spaces          .
## floor                   .
## animalnot acept         .
## furniturenot furnished .
## housecasa               .
```

(iii) Ajustando uma regressão com regularização ridge, tendo primeiramente o valor de λ obtido por validação cruzada:

```r
# tomando os valores de coeficiente do ridge pela validacao cruzada:
cv.ridge = cv.glmnet(x_train, y_train, alpha = 0)

# vendo o melhor valor de lambda
cv.ridge$lambda.min
```

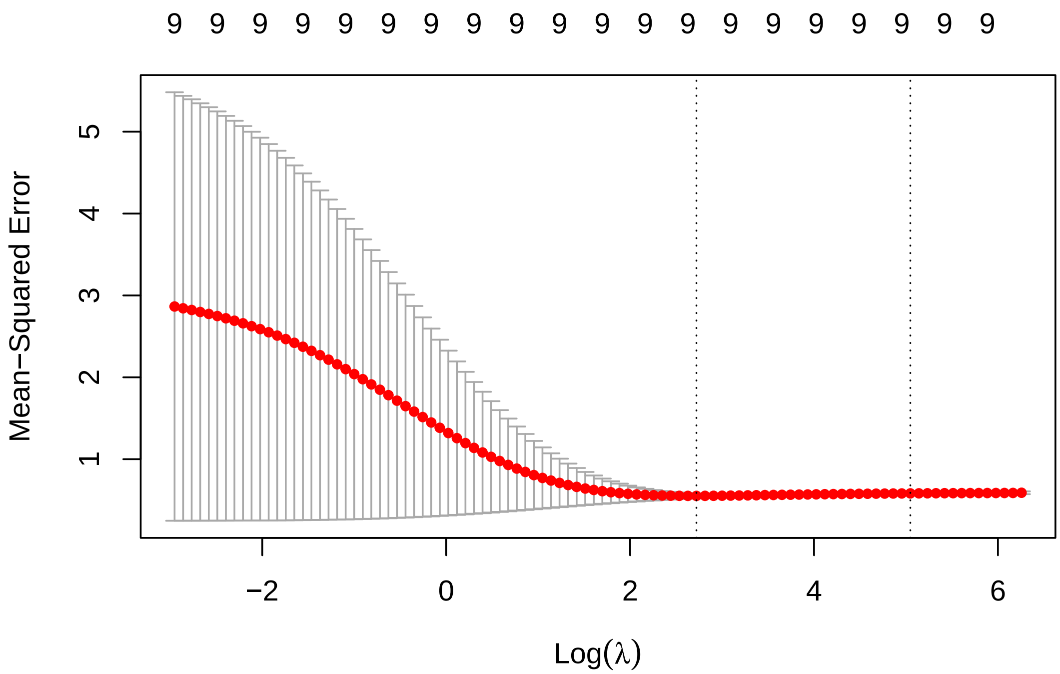
```
## [1] 15.19999
```

```r
cv.ridge$lambda.1se
```

```
## [1] 155.5766
```

Podemos averiguar qual o comportamento do λ com relação ao erro quadratico médio:

```r
plot(cv.ridge)
```



Assim como a figura anterior, esta figura possui a mesma interpretação daquela analisada anteriormente. Com o *lambda.min* $= 0.052444$ (ou seja, $\log(0.0524) = -2.948$). Para este λ, nota-se que o número de covariáveis selecionadas é 10. E o mesmo ocorre para o *lambda.1se*, em que seu valor é 0.1757 (ou seja, $\log(0.1757) = -1.7389$). Mas para este λ, o erro quadrático médio aumentou um pouco. O número de covariáveis selecionadas continua sendo 10, o que é eseperado, visto que a regularização Ridge não resulta em esparcidade. Assim como visto em sala, optando por minimizar o erro quadrático médio, escolhe-se o *lambda.min* $= 0.0524$. Aplicando o λ mínimo obtemos os coeficientes:

```r
# ajustando o modelo
mod_ridge <- glmnet(x_train, y_train, alpha = 0, lambda = cv.ridge$lambda.min)
# coeficientes
coef(mod_ridge)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                                   s0
## (Intercept)             7.725417e+00
## cityRio de Janeiro      8.762751e-04
## area                    3.614394e-06
## rooms                   1.699441e-02
## bathroom                1.909987e-02
## parking.spaces          1.221088e-02
## floor                   7.047400e-04
## animalnot acept        -6.918238e-03
## furniturenot furnished -1.792450e-02
## housecasa               1.962095e-02
```

Item (c):
Para a validação dos modelos, escolheremos o erro médio absoluto para analise para evitar erros muito inflados, ou seja, tomaremos a função de perda $L(g(\boldsymbol{X}); Y) = |Y - g(\boldsymbol{X})|$ e teremos interesse em estimar a função de risco $\mathbb{E}\left[|Y - g(\boldsymbol{X})|\right]$. Assim, predizendo y para cada observação do conjunto de teste, (voltando a escala original utilizando exp()) analisa-se a seguir a função de risco estimada para todos os modelos, com seus respectivos IC de 95% de confiança:

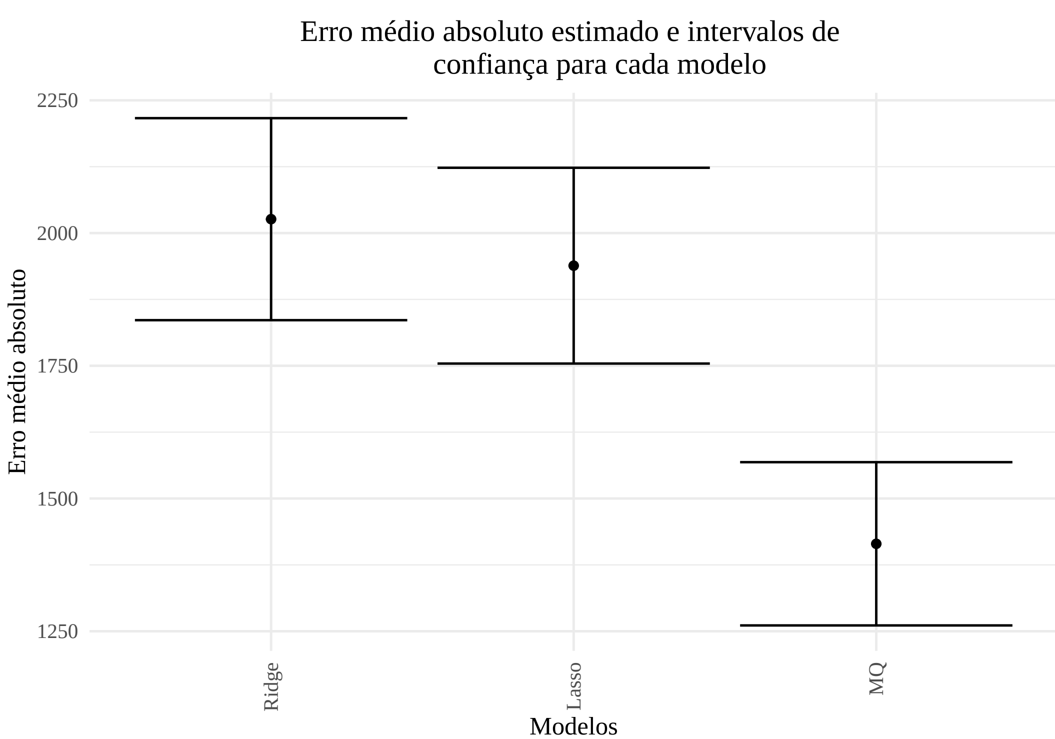
```r
# matriz de preditores
x_test <- model.matrix(rent.amount ~ ., test_data)[,-1]
y_test = test_data$rent.amount

pred.lasso = predict(mod_lasso,
s = cv.lasso$lambda.min,
newx = x_test)

pred.ridge = predict(mod_ridge,
s = cv.ridge$lambda.min,
newx = x_test)

pred.reg = predict(mod_reg,
s = 0,
newx = x_test)

# calculando as medidas de validacao
medidas = data.frame(
  Modelos = c("MQ", "Lasso", "Ridge"),
  MAE = c(MAE(exp(pred.reg), exp(y_test)), MAE(exp(pred.lasso), exp(y_test)),
          MAE(exp(pred.ridge), exp(y_test)))
)

std_error = function(loss_func, preds, y){
  SD = sqrt((1/length(y))*mean((abs(preds - y) - (loss_func(preds, y)))^2))
  return(2*SD)
}

errors = c(std_error(MAE, exp(pred.reg), exp(y_test)),
           std_error(MAE, exp(pred.lasso), exp(y_test)),
           std_error(MAE, exp(pred.ridge), exp(y_test)))
# calculando erro padrao para cada metodo
medidas$IC_lower = medidas$MAE - errors
medidas$IC_upper = medidas$MAE + errors
```

```
medidas
```

Modelos	MAE	IC_lower	IC_upper
MQ	1414.670	1260.916	1568.424
Lasso	1938.586	1754.204	2122.968
Ridge	2026.176	1835.921	2216.432

Ou seja, a regularização ridge parece nos dar um ajuste melhor, visto que sua estimativa pontual nos dá um bom erro em comparação aos outros dois modelos e o intervalo de confiança também parece melhor que os demais por possuir menor amplitude. Graficamente, pode-se observar melhor essas diferenças:

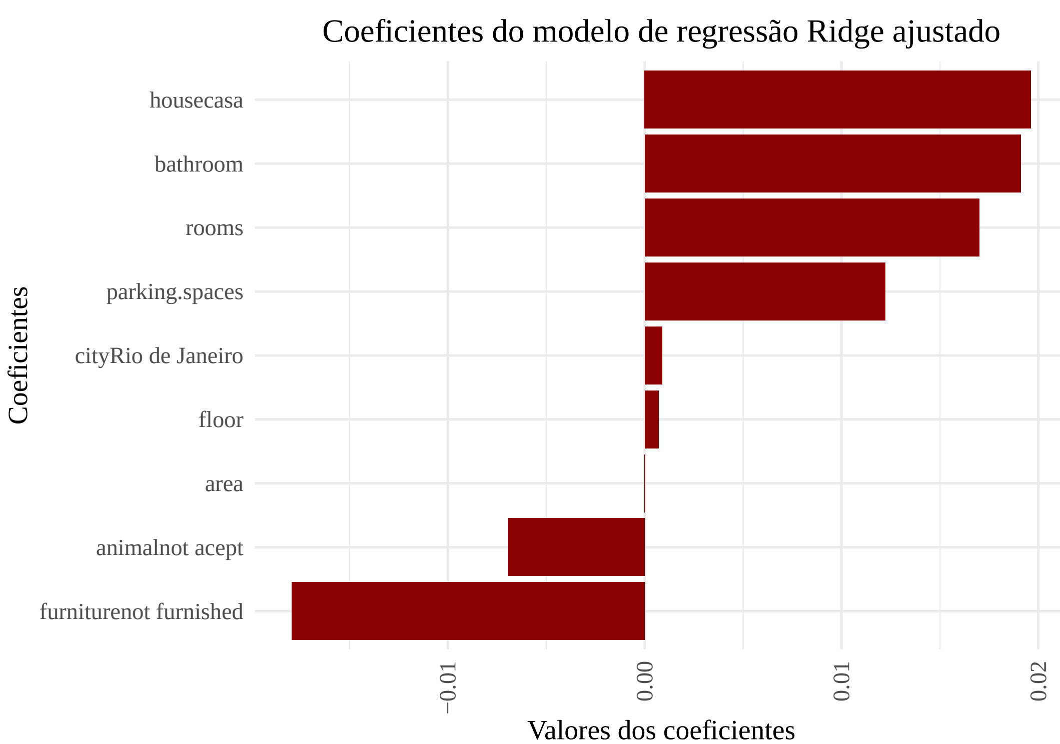
```
medidas %>%
  mutate(Modelos = fct_reorder(Modelos, MAE, .desc = T)) %>%
  ggplot(aes(x = Modelos, y = MAE)) +
  geom_point() +
  geom_errorbar(aes(ymin = IC_lower, ymax = IC_upper))+
  labs(title = "Erro médio absoluto estimado e intervalos de
       confiança para cada modelo",
       x = "Modelos",
       y = "Erro médio absoluto")+
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        text = element_text(size = 11,
                            family ="serif"),
        plot.title = element_text(hjust = 0.5))
```



Percebemos ampla semelhança entre os desempenhos dos modelos, tendo porém o intervalo de confiança do modelo Ridge mais deslocado "para baixo" em comparação aos outros dois modelos, com a estimativa pontual também mais abaixo que os demais. Assim a regressão Ridge de fato parece ser um melhor modelo que os demais, nesse cenário, para esses dados nesse estudo. Podemos interpretar seus coeficientes com auxílio do gráfico abaixo:

Item (d):

```
coefs_data = data.frame(coefs = coef(mod_ridge)[-1],
                        names = as.factor(row.names(coef(mod_ridge))[-1]))
coefs_data %>%
  mutate(names = fct_reorder(names, coefs, .desc = F)) %>%
  ggplot(aes(x = names, y = coefs))+
  geom_bar(stat = "identity", fill = "darkred")+
  labs(title = "Coeficientes do modelo de regressão Ridge ajustado",
       x = "Coeficientes",
       y = "Valores dos coeficientes")+
  coord_flip()+
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        text = element_text(size = 12,
                            family ="serif"),
        plot.title = element_text(hjust = 0.5))
```



Percebemos que fatores como o imóvel não ter decoração/móveis e não aceitar animais parecem influenciar negativamente no preço, tendo as cidades de Belo Horizonte e Rio de Janeiro menores preços de aluguel que São Paulo. Além disso, o imóvel ser uma casa e as variáveis relacionadas ao número de banheiros, quartos e vagas de estacionamento estão positivamente ligadas ao preço do aluguel. Tais constatações fazem sentido no contexto analisado, visto que de fato se espera que quanto mais completa a residência (mais banheiros, mais quartos e mais vaga de estacionamento) mais caro seja o aluguel e quanto menos completa ou mais

restritiva (não aceitar animal e não ter decoração) seja as condições de moradia, menos se espera de cobrança de aluguel. Ademais, casas naturalmente são mais caras que apartamentos em cidades grandes. Ressalta-se também que o preço ser mais baixo se o imóvel for de Belo Horizonte ou Rio de Janeiro em comparação a São Paulo é algo esperado, visto que São Paulo é uma cidade maior (territorialmente e populacionalmente) e mais caótica que essas outras duas. Fazendo agora a interação dupla entre as diferentes variáveis:
Item (e):

```
x_all_int = model.matrix(rent.amount ~ (.)^2 - floor:house, train_data)[,-1]

# treinando uma regressao linear pelo glmnet:
mod_reg_int = glmnet(x_all_int, y_train, alpha = 0, lambda = 0)
length(coef(mod_reg_int))
```

```
## [1] 45
```

Com as interações duplas teremos um total de 55 coeficientes. Repetindo o ajuste do lasso, agora para o caso com interações duplas envolvidas:

```
# tomando os valores de coeficiente do lasso pela validacao cruzada:
cv.lasso_int = cv.glmnet(x_all_int, y_train, alpha = 1)

# vendo o melhor valor de lambda
cv.lasso_int$lambda.min
```

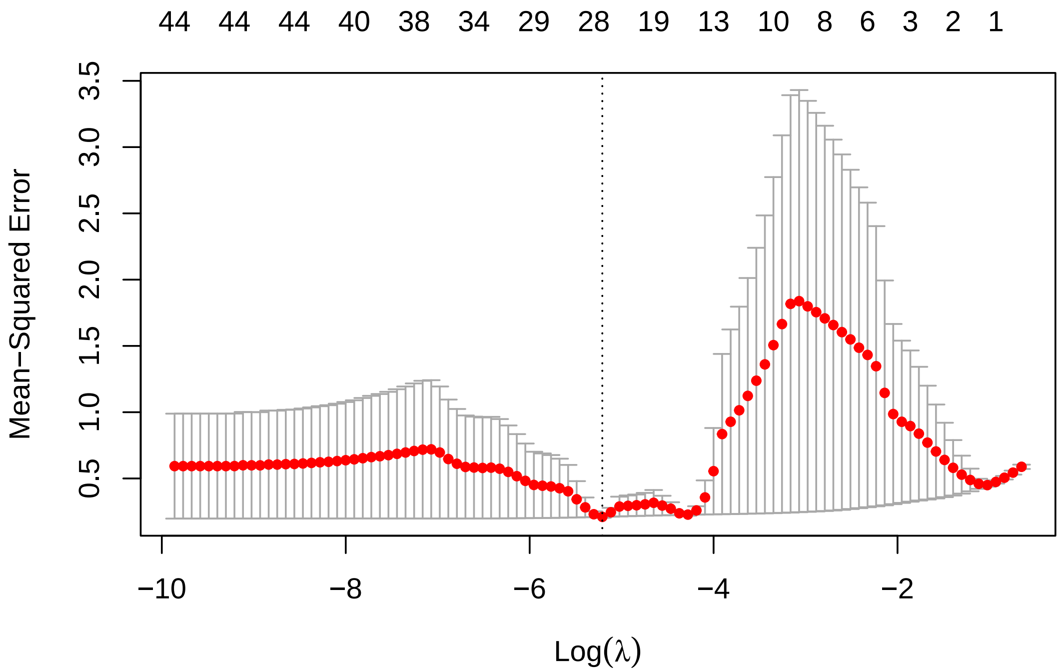
```
## [1] 0.005462592
```

```
cv.lasso_int$lambda.1se
```

```
## [1] 0.005462592
```

Obtemos um λ menor em comparação ao obtido anteriormente. Caso analisemos o gráfico de λ versus o erro quadrático médio:

```
plot(cv.lasso_int)
```



Observando o gráfico anterior, percebe-se que uma grande redução no número de coeficientes. Percebe-se a redução de 55 coeficientes para apenas 5 ou 6.Todavia, não há diferença significativa entre o $lambda.1se = 0.098$ e o $lambda.min = 0.0815$. Dado que ambos irão selecionar quase que o menos número de covariáveis. Ainda assim, para se utilizar o que foi visto em aula, escolhe-se o $lambda.min = 0.08158$, isto é, aquele que minimizar o erro. Ajustando o modelo com o λ mínimo:

```
# ajustando o modelo
mod_lasso_int <- glmnet(x_all_int, y_train, alpha = 1, lambda = cv.lasso_int$lambda.min)
```

Comparando desempenhos, utilizando a mesma medida do **item (c)** (transformando a predição para a escala original com exp():

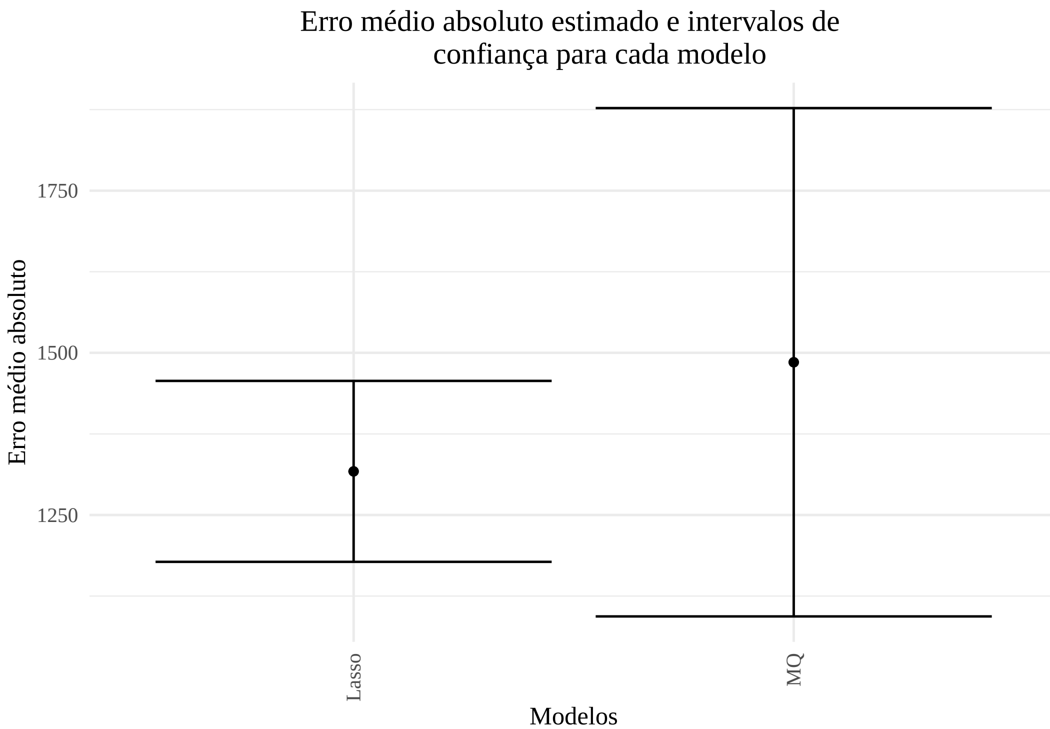
```
# matriz de preditores
x_test <- model.matrix(rent.amount ~ (.)^2 - floor:house, test_data)[,-1]

pred.lasso_int = predict(mod_lasso_int,
s = cv.lasso_int$lambda.min,
newx = x_test)

pred.reg_int = predict(mod_reg_int,
s = 0,
newx = x_test)

# calculando as medidas de validacao
medidas = data.frame(
  Modelos = c("MQ", "Lasso"),
  MAE = c(MAE(exp(pred.reg_int), exp(y_test)), MAE(exp(pred.lasso_int), exp(y_test)))
)

std_error = function(loss_func, preds, y){
  SD = sqrt((1/length(y))*mean((abs(exp(preds) - exp(y)) - (loss_func(exp(preds), exp(y))))^2))
```

```r
  return(2*SD)
}

errors = c(std_error(MAE, pred.reg_int, y_test), std_error(MAE, pred.lasso_int, y_test))
# calculando erro padrao para cada metodo
medidas$IC_lower = medidas$MAE - errors
medidas$IC_upper = medidas$MAE + errors
medidas
```

Modelos	MAE	IC_lower	IC_upper
MQ	1485.454	1093.634	1877.274
Lasso	1317.245	1177.772	1456.717

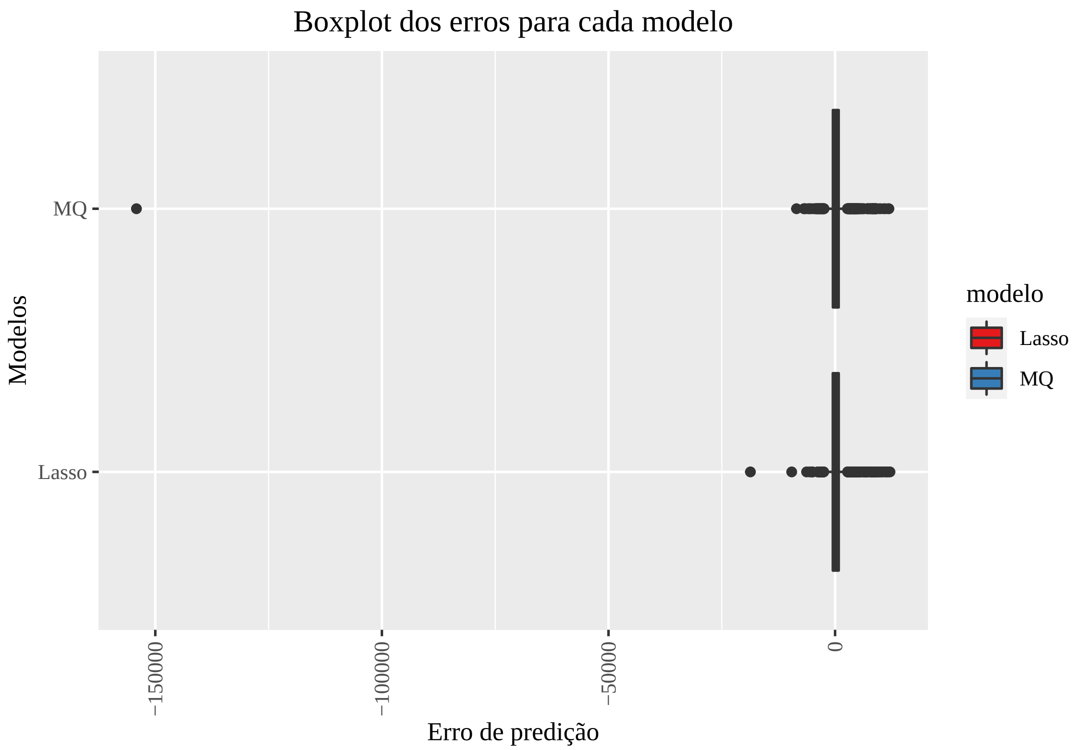
Percebemos uma diferença brusca entre o MQ e Lasso do item **(c)** para o item **(e)** e isso se pode explicar pelo fato de ao adicionarmos muitas interações, estamos também aumentando a variância associada modelo de mínimos quadrados, e dessa forma, aumentando o Risco estimado associado a esse modelo. Como a penalização lasso tem uma boa filtragem das covariáveis, o modelo obtido não só evita o aumento de variância como também a diminui através da penalização, tendo ao final um risco estimado pontual melhor que aquele obtido no MQ e também que aquele obtido no item **(c)**. Outrossim, percebe-se uma grande amplitude associada ao modelo MQ, com seu IC englobando também o intervalo de confiança do Lasso. Isso pode-se explicar pelo fato de possivelmente o MQ acertar muito em certos casos e errar muito em outros. A estimativa pontual e intervalar do Risco para cada modelo é dada graficamente abaixo:

```r
medidas %>%
  ggplot(aes(x = Modelos, y = MAE)) +
  geom_point() +
  geom_errorbar(aes(ymin = IC_lower, ymax = IC_upper))+
  labs(title = "Erro médio absoluto estimado e intervalos de
       confiança para cada modelo",
       x = "Modelos",
       y = "Erro médio absoluto")+
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        text = element_text(size = 11,
                            family ="serif"),
        plot.title = element_text(hjust = 0.5))
```



Vemos claramente que o gráfico acima tem uma alta variância do modelo MQ, com o intervalo do MQ englobando todo o intervalo do Lasso. Caso analisemos o boxplot dos erros ($\hat{y} - y$, sem usar nenhuma perda específica) associados a cada modelo, visualizaremos:

```
erros = data.frame("MQ" = as.numeric(exp(y_test) - exp(pred.reg_int)),
                   "Lasso" = as.numeric(exp(y_test) - exp(pred.lasso_int)))


erros %>%
  pivot_longer(cols = MQ:Lasso, names_to = 'modelo', values_to = 'erro') %>%
  ggplot(aes(x = modelo, y = erro, fill = modelo))+
  geom_boxplot()+
  labs(title = "Boxplot dos erros para cada modelo",
       x = "Modelos",
       y = "Erro de predição")+
  coord_flip()+
  theme_grey()+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        text = element_text(size = 11,
                            family ="serif"),
        plot.title = element_text(hjust = 0.5))+
  scale_fill_brewer(palette = "Set1")
```



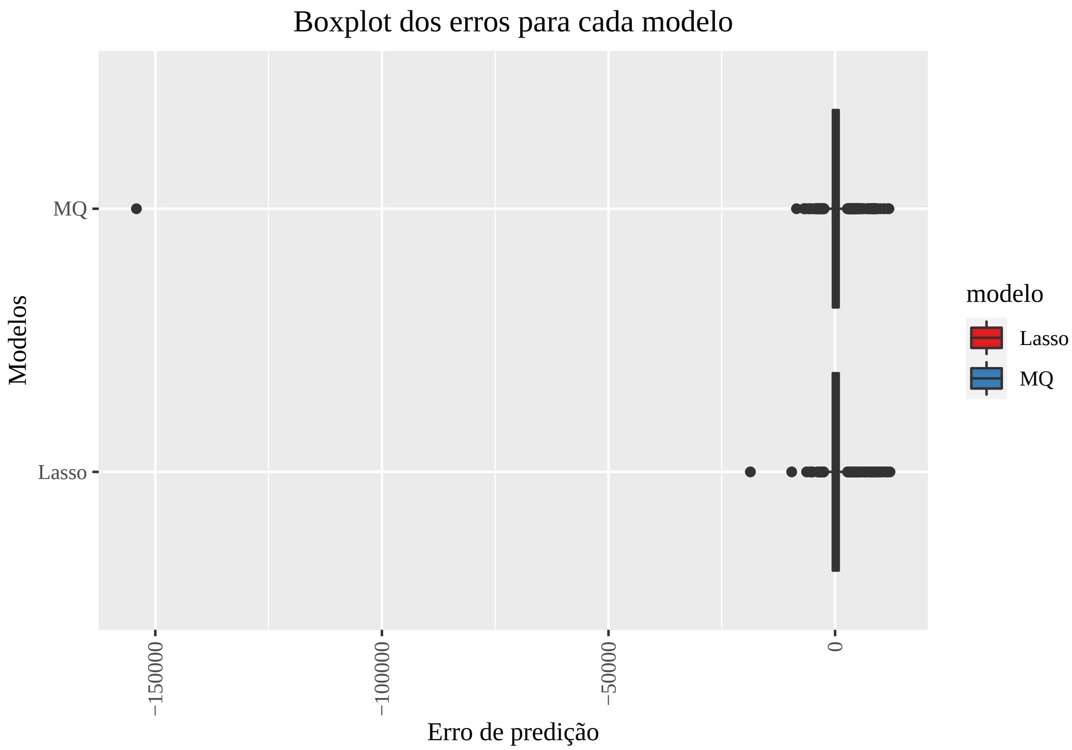
Evidencia-se pela figura anterior que um ponto em específico, no método MQ, possui um erro extremamente mais acentuado que os demais. Tal erro chega na casa dos milhões. Dessa forma, é interessante que tal observação 2151 (numeração de acordo com a amostra teste) seja explorada com mais atenção.

```
which(erros$MQ < -1000000)
```

```
## integer(0)
```

Agora, se faz o mesmo gráfico, retirando a observação 2151:

```
erros %>% filter(MQ > -1000000) %>%
pivot_longer(cols = MQ:Lasso, names_to = 'modelo', values_to = 'erro') %>%
  ggplot(aes(x = modelo, y = erro, fill = modelo))+
  geom_boxplot()+
  labs(title = "Boxplot dos erros para cada modelo",
       x = "Modelos",
       y = "Erro de predição")+
  coord_flip()+
  theme_grey()+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        text = element_text(size = 11,
                            family ="serif"),
        plot.title = element_text(hjust = 0.5))+
  scale_fill_brewer(palette = "Set1")
```



Estudando a figura anterior, percebe-se que com a retirada da 2151, o boxplot relativo ao MQ fica mais comportado. Ainda assim, nota-se que alguns pontos em específico têm erros grandes no caso do MQ. Entretanto, tais observações não são tão gritantes quanto aquela 2151 anterior, que tinha erro na casa dos milhões.

```r
erros %>% filter(MQ > -1000000) %>%
  abs() %>%
  colMeans()
```

```
##       MQ    Lasso
## 1485.454 1317.245
```

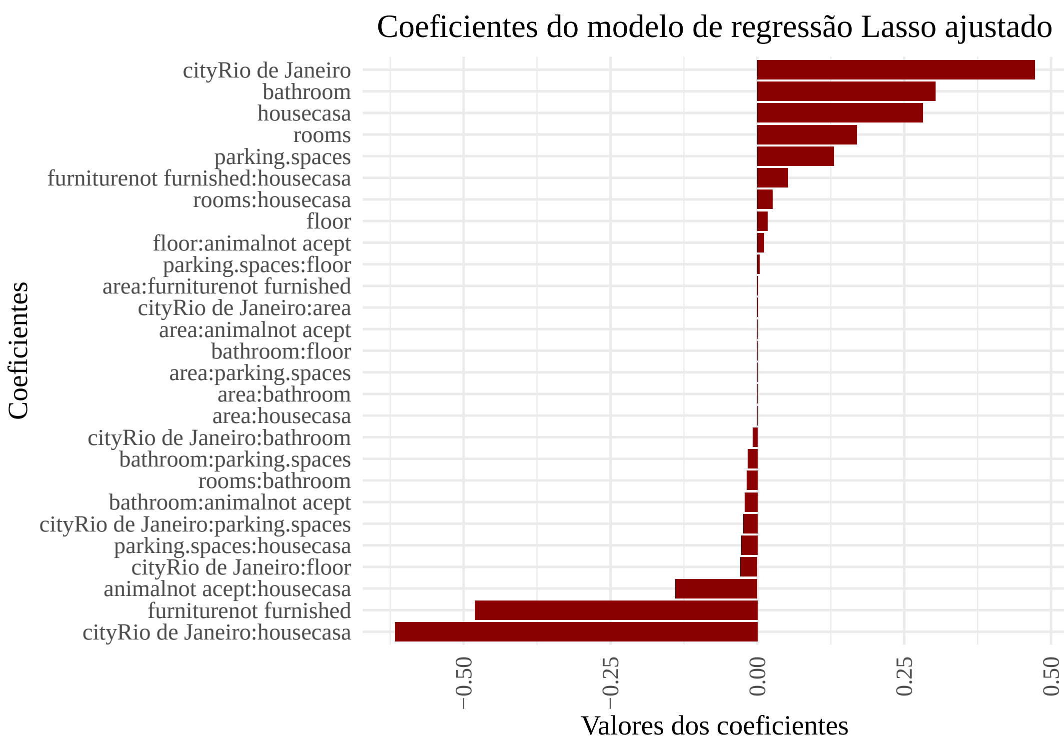
Continuando com a obserevação 2151 retirada da amostra teste, é notável que ao calcular a estimativa pontual do erro absoluto médio, o MQ e Lasso ficam mais próximos.Não somente isso, mas o MQ tem agora o risco estimado menor.

```r
test_data[2151, ]
```

	city	area	rooms	bathroom	parking.spaces	floor	animal	furniture	rent.amount	house
NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Analisando a observação 2151 em específico, percebe-se que sua área é de 2000 metros^2. Tal área de fato é muito grande, contudo a anotação está correta. Visto que em Belo Horizonte há casas com mais de 3000 metros^2. Tomando novamente o conjuntod e teste inteiro, podemos analisar novamente os coeficientes da regressão Lasso diferentes de zero:

```
coefs_data = data.frame(coefs = coef(mod_lasso_int)[-1],
                        names = as.factor(row.names(coef(mod_lasso_int))[-1]))
coefs_data %>%
  filter(coefs != 0) %>%
  mutate(names = fct_reorder(names, coefs, .desc = F))%>%
  ggplot(aes(x = names, y = coefs))+
  geom_bar(stat = "identity", fill = "darkred")+
  labs(title = "Coeficientes do modelo de regressão Lasso ajustado",
       x = "Coeficientes",
       y = "Valores dos coeficientes")+
  coord_flip()+
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
        text = element_text(size = 12,
                            family ="serif"),
        plot.title = element_text(hjust = 0.5))
```



Retiram-se conclusões semelhantes as anteriories, com a diferença de que nesse caso temos menos coeficientes com as variáveis número de banheiros, salas e vagas de estacionamento como positivamente relacionadas ao valor do aluguel, enquanto que o imóvel ser de Belo Horizonte e não decorado está negativamente relacionado ao preço do aluguel.

Exercício 2:

Demonstrar que

$$E[(Y - g(X))^2|X = x] = Var[Y|X = x] + (r(x) - E[g(x)])^2 + Var[g(x)]$$

Para iniciar, somando e subtraindo a função de regressão ($r(x) = E[Y|X = x]$), temos:

$$\begin{aligned}E[(Y - r(X) + r(X) - g(X))^2|X = x] = E[(Y - r(X))^2|X = x] + E[(r(X) - g(X))^2|X = x] \\ -2\,E[(Y - r(X)) \cdot (r(X) - g(X))|X = x]\end{aligned}$$

Para facilitar, desenvolveremos os termos separadamente. Primeiramente, $E[(Y - r(X))^2|X = x]$:

$$\begin{aligned}E[(Y - r(X))^2|X = x] &= E[Y^2|X = x] - 2\,E[Yr(X)|X = x] + E[r(X)^2|X = x] \\ &= E[Y^2|X = x] - 2\,r(x)\,E[Y|X = x] + r(x)^2 \\ &= E[Y^2|X = x] - E[Y|X = x]^2 + (r(x) - E[Y|X = x])^2 \\ &= Var[Y|X = x] + (r(x) - E[Y|X = x])^2 \\ &= Var[Y|X = x]\end{aligned}$$

Agora iremos desenvolver, $E[(r(X) - g(X))^2|X = x]$:

$$\begin{aligned}E[(r(X) - g(X))^2|X = x] &= E[(r(X) - E[g(X)] + E[g(X)] - g(X))^2|X = x] \\ &= E[((r(X) - E[g(X)]) - (g(X) - E[g(X)]))^2|X = x] \\ &= E[(r(X) - E[g(X)])^2|X = x] + E[g(X) - E[g(X)])^2|X = x] \\ -2\,E[(r(X) - E[g(X)]) \cdot (g(X) - E[g(X)])|X = x] \\ &= E[r(X)^2 - 2r(X)E[g(X)] + E[g(X)]^2|X = x] + Var[g(x)] \\ &= r(x)^2 - 2r(x)E[g(x)] + E[g(x)]^2 + Var[g(x)] \\ &= (r(x) - E[g(x)])^2 + Var[g(x)]\end{aligned}$$

E por fim $2\,E[(Y - r(X)) \cdot (r(X) - g(X))|X = x]$:

$$\begin{aligned}2\,E[(Y - r(X)) \cdot (r(X) - g(X))|X = x] &= 2\,E[Y - r(X)|X = x] \cdot E[r(X) - g(X)|X = x] \\ &= 2 \cdot 0 \cdot E[r(X) - g(X)|X = x] \\ &= 0\end{aligned}$$

Agrupando os resultados obtidos:

$$E[(Y - g(X))^2|X = x] = Var[Y|X = x] + (r(x) - E[g(x)])^2 + Var[g(x)]$$

sendo que os dois últimos termos da expressão estão relacionados ao estimador escolhido, podendo ser minimizado quando a escolha é adequada.

Exercício 3:

Fixando $0 < \alpha < 1$ e considerando a função de perda:

$$L(g; (\boldsymbol{X}, Y)) = (g(\boldsymbol{X}) - Y) \cdot (\mathbb{I}(Y \leq g(\boldsymbol{X})) - \alpha)$$

Para obter a função g que minimiza a função de risco $R(L(g; (\boldsymbol{X}, Y)))$ com g fixo, inicialmente podemos fazer:

$$\begin{aligned}\mathbb{E}\left[L(g; (\boldsymbol{X}, Y))\right] &= \mathbb{E}\left[(g(\boldsymbol{X}) - Y) \cdot (\mathbb{I}(Y \leq g(\boldsymbol{X})) - \alpha)\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[(g(\boldsymbol{X}) - Y) \cdot (\mathbb{I}(Y \leq g(\boldsymbol{X})) - \alpha)\,|\boldsymbol{X}\right]\right]\end{aligned}$$

Assim, podemos minimizar $\mathbb{E}\left[(g(\boldsymbol{X})-Y)\cdot(\mathbb{I}(Y\leq g(\boldsymbol{X}))-\alpha)\right]$, minimizando $\mathbb{E}\left[(g(\boldsymbol{X})-Y)\cdot(\mathbb{I}(Y\leq g(\boldsymbol{X}))-\alpha)\,|\boldsymbol{X}\right]$ em função de g. Dessa maneira, dado que g é fixo, queremos obter $arg\,min_{g\in\mathcal{A}}\mathbb{E}\left[(g(\boldsymbol{X})-Y)\cdot(\mathbb{I}(Y\leq g(\boldsymbol{X}))-\alpha)\,|\boldsymbol{X}\right]$ tal que $\mathcal{A}=\{f:\mathbb{R}^d\to\mathbb{R}\}$, ou seja, $\mathcal{A}$ é o conjunto de todas as possíveis funções de predição. Assim, podemos desenvolver essa esperança condicional da seguinte maneira:

$$\begin{aligned}
\mathbb{E}\left[(g(\boldsymbol{X})-Y)\cdot(\mathbb{I}(Y\leq g(\boldsymbol{X}))-\alpha)\,|\boldsymbol{X}\right] &= \mathbb{E}\left[\mathbb{I}(Y\leq g(\boldsymbol{X}))(g(\boldsymbol{X})-Y)|\boldsymbol{X}\right]-\mathbb{E}[\alpha(g(\boldsymbol{X})-Y)|\boldsymbol{X}]\\
&= g(\boldsymbol{X})\int_{-\infty}^{g(\boldsymbol{X})} f_{Y|\boldsymbol{X}}(y|x)dy-\int_{-\infty}^{g(\boldsymbol{X})} y\cdot f_{Y|\boldsymbol{X}}(y|x)dy-\alpha g(\boldsymbol{X})+\alpha\mathbb{E}[Y|\boldsymbol{X}]
\end{aligned}$$

Derivando tal expressão em função de g, utilizando o teorema fundamental do cálculo teremos:

$$\begin{aligned}
\frac{\partial\mathbb{E}\left[(g(\boldsymbol{X})-Y)\cdot(\mathbb{I}(Y\leq g(\boldsymbol{X}))-\alpha)\,|\boldsymbol{X}\right]}{\partial g(\boldsymbol{X})} &= \frac{\partial g(\boldsymbol{X})\int_{-\infty}^{g(\boldsymbol{X})} f_{Y|\boldsymbol{X}}(y|x)dy}{\partial g(\boldsymbol{X})}-\frac{\partial\int_{-\infty}^{g(\boldsymbol{X})} y\cdot f_{Y|\boldsymbol{X}}(y|x)dy}{\partial g(\boldsymbol{X})}-\frac{\partial\alpha g(\boldsymbol{X})+\alpha\mathbb{E}[Y|\boldsymbol{X}]}{\partial g(\boldsymbol{X})}\\
&= F_{Y|\boldsymbol{X}}(g(\boldsymbol{X}))+g(\boldsymbol{X})f_{Y|\boldsymbol{X}}(\boldsymbol{X})-g(\boldsymbol{X})f_{Y|\boldsymbol{X}}(\boldsymbol{X})-\alpha\\
&= F_{Y|\boldsymbol{X}}(g(\boldsymbol{X}))-\alpha
\end{aligned}$$

Igualando a zero, teremos:

$$g(\boldsymbol{X})=F^{-1}_{Y|\boldsymbol{X}}(\alpha)$$

E como a segunda derivada em $g(x)$ de $\mathbb{E}\left[(g(\boldsymbol{X})-Y)\cdot(\mathbb{I}(Y\leq g(\boldsymbol{X}))-\alpha)\,|\boldsymbol{X}\right]$ é dado por $f_{Y|\boldsymbol{X}}(g(\boldsymbol{X}))\geq 0$, teremos que $g(\boldsymbol{X})=F^{-1}_{Y|\boldsymbol{X}}(\alpha)$ é ponto de mínimo para tal esperança, e assim, obtemos que:

$$arg\,min_{g\in\mathcal{A}}\mathbb{E}\left[(g(\boldsymbol{X})-Y)\cdot(\mathbb{I}(Y\leq g(\boldsymbol{X}))-\alpha)\,|\boldsymbol{X}\right]=F^{-1}_{Y|\boldsymbol{X}}(\alpha)$$

Ou seja, temos que o quantil em função de α da função densidade de probabilidade de Y dado uma nova observação $\boldsymbol{X}$ é o preditor que minimiza o risco condicional observado, e como g é fixo, conclui-se que tal preditor minimiza o risco $\mathbb{E}\left[(g(\boldsymbol{X})-Y)\cdot(\mathbb{I}(Y\leq g(\boldsymbol{X}))-\alpha)\right]$. Em particular, se $\alpha=0.5$, o modelo é dado pela mediana de Y dado $\boldsymbol{X}$. Esse resultado pode ser interpretado pensando que a presente função de risco realiza uma ponderação e penalização sobre a subestimação ou sobrestimação de Y por $g(\boldsymbol{X})$ de acordo com uma constante fixada α. Se $\alpha<0.5$, penalizaremos mais a subestimação de Y visto que para $\boldsymbol{X_1},\boldsymbol{X_2}$ fixados tal que $|g(\boldsymbol{X_1})-Y|=|g(\boldsymbol{X_2})-Y|$ de forma que $g(\boldsymbol{X_2})>Y$ e $g(\boldsymbol{X_1})<Y$, teremos uma maior perda para $g(\boldsymbol{X_1})$ que $g(\boldsymbol{X_2})$. Caso $\alpha>0.5$, penalizaremos mais a sobrestimação. E para $\alpha=0.5$ não haverá uma penalização quanto a sobrestimação e subestimação. Assim, o estimador obtido leva em conta tais penalizações ao estimar Y pelos quantis dado a observação $\boldsymbol{X}$ fixados de acordo com α, corrigindo uma possível subestimação (se $\alpha<0.5$) ou sobrestimação (se $\alpha>0.5$) ou um balanço das duas ($\alpha=0.5$) a partir do quantil dado o quantil observado.