Data Analysis Project 1

MA8701

Group 5 : Yellow Submarine

15 February, 2021

Note on Open Science

To pursue the idea of reproducible research, the chosen dataset as well as the code for our analysis are publicly accessible:

- dataset: https://data.ub.uni-muenchen.de/2/1/miete03.asc
- code: https://github.com/FlorianBeiser/MA8701

Regression

We start with a vanilla LM regression for reference. Only significant coeffcients are printed. Clearly, the area `wfl` is strongly related to the rent price. Surprisingly in the regression, the significance of different `bj`s and `bez`s varies a lot.

```
##             summary.lm_mod..coef.summary.lm_mod..coef...4.....0.05..4..1.4.
## (Intercept)                                                    6.944363e-09
## wfl                                                           1.183420e-130
## rooms                                                          4.474346e-02
## bj1924                                                         3.936400e-07
```

Ridge

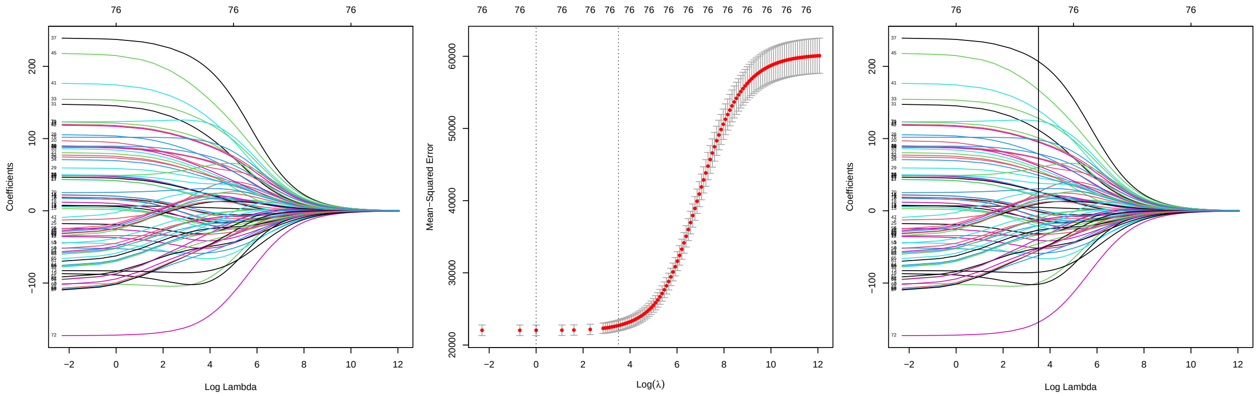
```
start <- glmnet(x = x_mod, y = y_mod, standardize = TRUE, alpha = 0)
autolambda <- start$lambda
newlambda <- c(autolambda, 10, 5, 3, 1, 0.5, 0.1)  # add more to approach zero lambda
ridge_fit <- glmnet(x_mod, y_mod, standardize = TRUE, alpha = 0, lambda = newlambda)
cv.ridge <- cv.glmnet(x_mod, y_mod, standardize = TRUE, alpha = 0, lambda = newlambda)
print(paste("The lamda giving the smallest CV error", cv.ridge$lambda.min))
```

```
## [1] "The lamda giving the smallest CV error 1"
```

```
print(paste("The 1sd err method lambda", cv.ridge$lambda.1se))
```

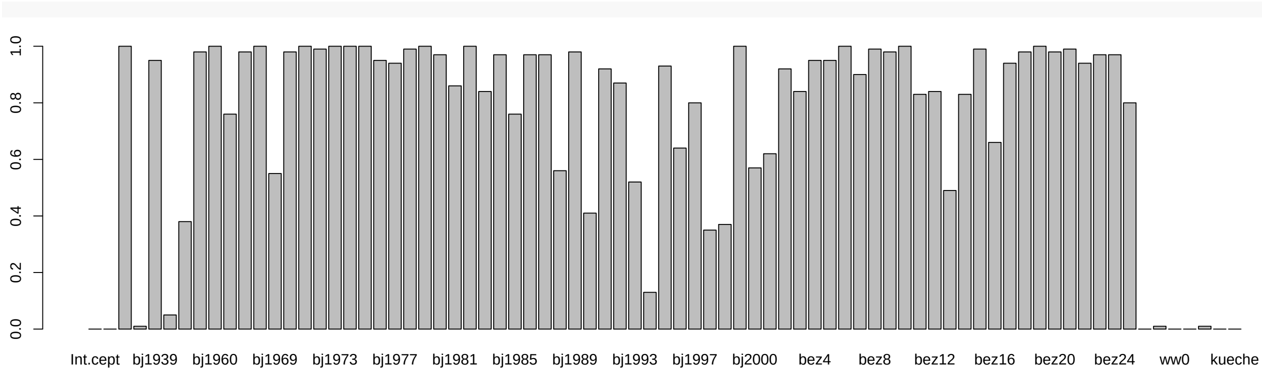
```
## [1] "The 1sd err method lambda 33.2936676208139"
```

```
par(mfrow = c(1, 3), mar = c(4, 4, 4, 1), oma = c(0.5, 0.5, 0.5, 0))
plot(ridge_fit, xvar = "lambda", label = T)
plot(cv.ridge)
plot(ridge_fit, xvar = "lambda", label = T)
abline(v = log(cv.ridge$lambda.1se))
```

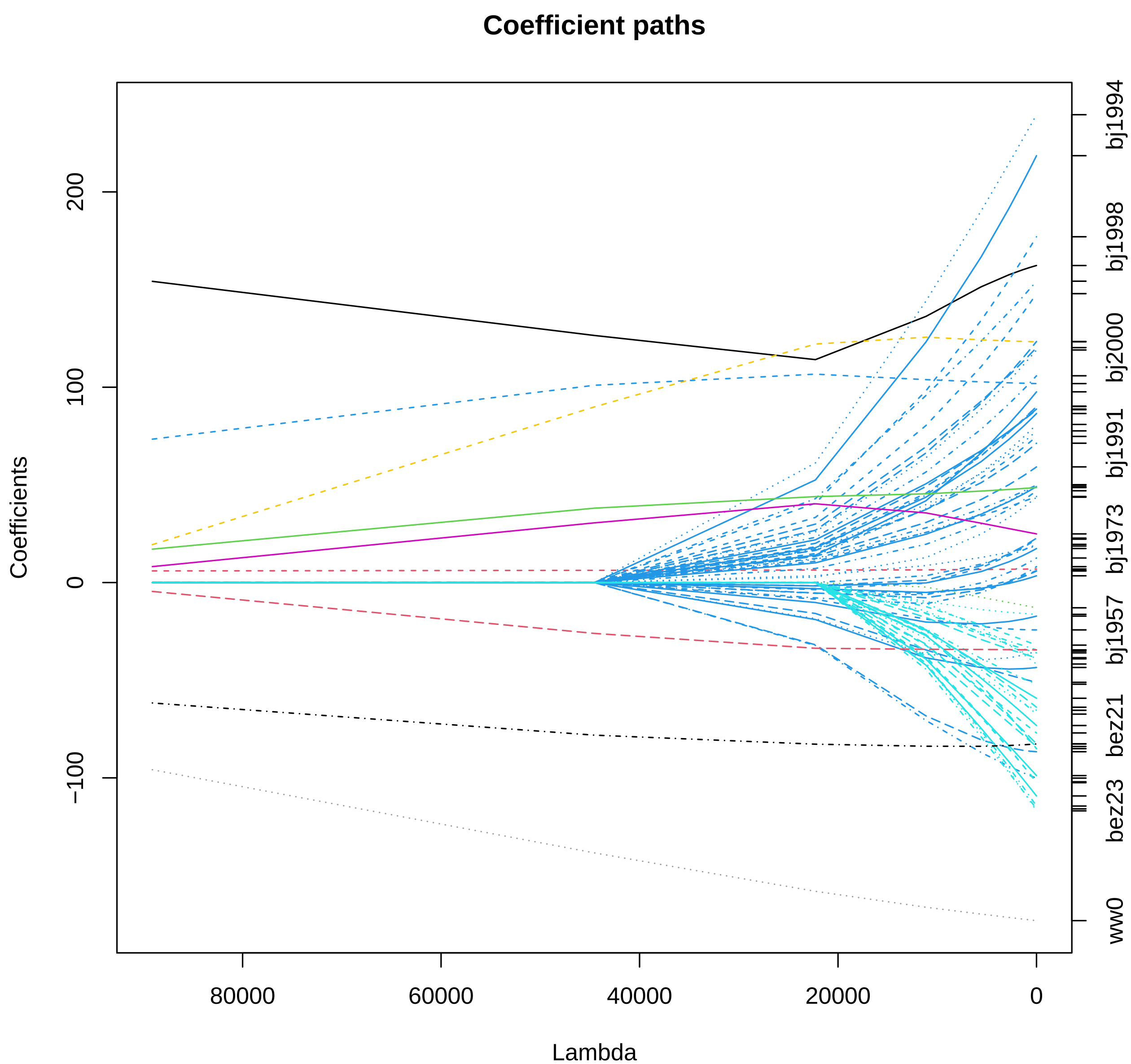


#Bootstrap validation Bootstrap can be applied here to find the proportion of times each element in the coefficients vector of being zero. So it is a way of validation.

```
# boostrap loop
set.seed(2021)
B=100
n=nrow(x_mod)
p=ncol(x_mod)
lassomat=matrix(ncol=p+1,nrow=B)
ridgemat=matrix(ncol=p+1,nrow=B)

# no need or separate function for steps 1-6 since can use cv.glmnet
# and weight argument for giving the new bootstrapped data
for (b in 1:B)
{
  ids=sort(sample(1:n,replace=TRUE))
  wids=rep(0,n)
  for (i in 1:n)
    wids[i]=sum(ids==i)
  resl=cv.glmnet(x_mod,y_mod,weights=wids)
  resr=cv.glmnet(x_mod,y_mod,weights=wids,alpha=0)
  lassomat[b,]=as.vector(coef(resl)) #automatic lambda 1sd
  ridgemat[b,]=as.vector(coef(resr)) #automatic lambda 1sd
}
colnames(lassomat)=colnames(ridgemat)=c("Int.cept",colnames(x_mod))
# plotting boxplots
lassomatUI=lassomat[,-1]
lassods=reshape2::melt(lassomatUI,
         variable.name ="variable",value.name="value")
lassopp=ggplot(lassods,aes(x=Var2,y=value))+
  geom_boxplot()+ggtitle("Boxplots for boostrapped lasso for diabetes data")
# lassopp

ridgematUI=ridgemat[,-1]
ridgeds=reshape2::melt(ridgematUI,variable.name="variable",value.name="value")
ridgepp=ggplot(ridgeds,aes(x=Var2,y=value))+
  geom_boxplot()+ggtitle("Boxplots for boostrapped ridge for diabetes data")
# ridgepp

lasso0perc=apply(abs(lassomat)<.Machine$double.eps,2,mean)
par(mfrow = c(1, 1))
barplot(lasso0perc)
```

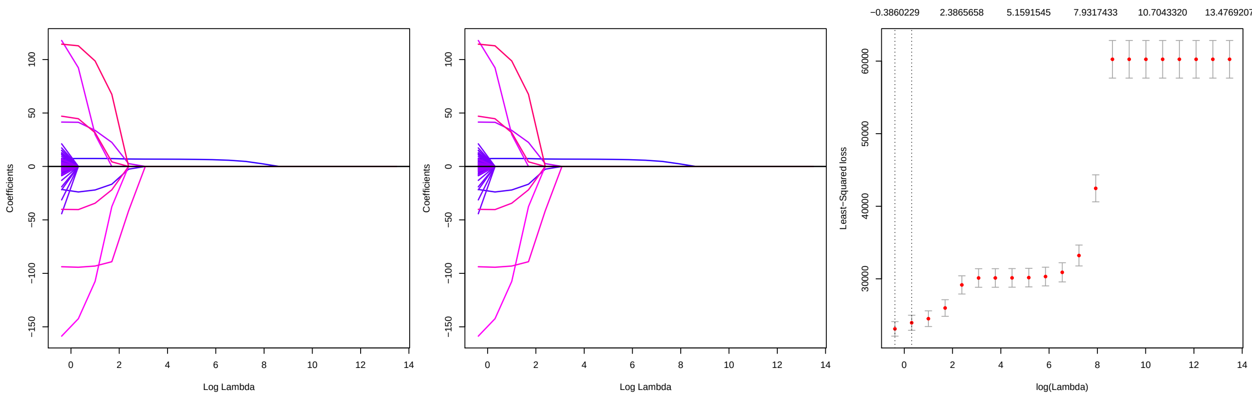

0.0
0.2
0.4
0.6
0.8
1.0
Int.cept
bj1939
bj1960
bj1969
bj1973
bj1977
bj1981
bj1985
bj1989
bj1993
bj1997
bj2000
bez4
bez8
bez12
bez16
bez20
bez24
ww0
kueche

Group lasso


Coefficient paths
Coefficients
Lambda
200
100
0
−100
80000
60000
40000
20000
0
bj1994
bj1998
bj2000
bj1991
bj1973
bj1957
bez21
bez23
ww0

In the grouped lasso, the `bj` and `bez` are all shrinked or are all inclueded, respectively. This coincides better with our intuition, that this criterion is considered or not considered. Whereas in the regression and lasso before, just some years of construction and some areas where significant.

```
## [1] 1.35951
```




```
## 77 x 4 sparse Matrix of class "dgCMatrix"
##               vanilla LS        ridge general lasso group lasso
## (Intercept)  162.310441  168.4715817    122.673487  106.107086
## wfl            6.921638    4.8629070      6.342412    7.413067
## rooms        -12.919931   25.7264339      .         -23.871982
## bj1924      -100.109344  -99.4046212    -73.446432    .
## bj1939       -51.082040  -58.9465412      .           .
## bj1948       -43.469920  -65.6687241    -37.034493    .
## bj1957       -24.238117  -40.8564769    -10.676323    .
## bj1957.5      18.713838   -0.9125516      .           .
## bj1960        19.561674   -9.5305143      .           .
## bj1966         5.920349  -17.5316242      .           .
## bj1967        17.432638   -7.9666613      .           .
## bj1968         6.161898  -21.3940887      .           .
## bj1969       -35.123926  -51.6304994    -14.711354    .
## bj1970         8.146714  -12.6099436      .           .
## bj1971        22.738843   -1.7844297      .           .
## bj1972         3.464200  -15.6345329      .           .
## bj1973        22.219275    1.6222145      .           .
## bj1974        43.700203   21.5328485      .           .
## bj1975        12.564953  -15.3913215      .           .
## bj1976       -86.605034 -101.6647975      .           .
## bj1977        97.644285   64.2430125      .           .
## bj1978        44.068520   19.8998308      .           .
## bj1979        50.112745   26.2832574      .           .
## bj1980        49.937326   38.2516186      .           .
## bj1981        88.509713   72.1350495      .           .
## bj1982       -17.165153  -31.7554851      .           .
## bj1983        74.815843   52.9317513      .           .
## bj1984        80.953167   55.7123009      .           .
## bj1985       105.867818   78.7009273      .           .
## bj1986        59.225499   45.5847650      .           .
## bj1987        49.115827   25.9989203      .           .
## bj1988       147.915666  111.9539126     16.588404    .
## bj1989        77.648956   53.1657915      .           .
## bj1990       154.290945  127.9670624     34.664189    .
```

```
## bj1991        71.347309   49.1283501       .           .
## bj1992        86.541067   58.8138218       .           .
## bj1993        90.312924   62.1667748     8.340729     .
## bj1994       239.532748  206.6996228   117.656488     .
## bj1995        90.135389   72.9381986       .           .
## bj1996       123.421116  100.8788309    10.486620     .
## bj1997        88.819228   78.8415667       .           .
## bj1998       177.049378  138.9134856    43.126355     .
## bj1998.5     119.079298   94.9563515    29.919294     .
## bj1999        47.001514   26.6754469       .           .
## bj2000       120.284699   96.0137413    20.408401     .
## bj2001       218.551590  167.2567758    19.861128     .
## bez2         -35.985131   14.1742602       .           .
## bez3         -16.274425   25.3125976       .           .
## bez4         -34.474015   18.7801598       .           .
## bez5         -38.466358    9.5381755       .           .
## bez6         -59.243092  -13.2742914       .           .
## bez7        -101.994969  -45.4102319       .           .
## bez8         -65.397522  -20.4966813       .           .
## bez9         -52.053469   -5.9794115       .           .
## bez10        -63.833161  -12.9174696       .           .
## bez11        -98.831306  -51.7946291       .           .
## bez12        -32.035394   20.6130943       .           .
## bez13        -41.710326   17.1253215     6.729042     .
## bez14       -115.863027  -61.9261117       .           .
## bez15        -85.041679  -25.4459111       .           .
## bez16       -109.255107  -52.4098604    -2.965043     .
## bez17        -76.998642  -26.9632238       .           .
## bez18        -39.053201   11.1194434       .           .
## bez19        -67.355571  -10.6046787       .           .
## bez20        -82.574987  -29.6472488       .           .
## bez21        -73.198994  -20.4163988       .           .
## bez22       -102.468535  -38.3224415       .           .
## bez23       -116.883323  -54.4625628       .           .
## bez24       -114.417039  -55.0778235       .           .
## bez25        -83.937882  -33.2424724       .           .
## wohngut       24.911148   30.3952336    34.414495   41.318858
## wohnbest     123.264686  124.1723666   101.928210   92.318048
## ww0         -173.087458 -154.3111090  -146.292424 -142.415965
## zh0          -82.624164  -84.9891552   -78.010891  -94.286712
## badkach0     -34.489575  -33.1824107   -29.551194  -40.350166
## badextra      48.627634   59.9369537    37.441777   44.631976
## kueche       101.861941   98.1190664   102.748488  112.865650
```