2. Ames Data Example

Hair Parra

2024-01-13

Libraries

```
library("glmnet")
```

Ames Data Example

- alternative to the well-known Boston Housing data set.
- It has 2,330 observations and 82 variables and contains information from the Ames Assessor's Office used in computing assessed values for individual residential properties sold in Ames, IA from 2006 to 2010.
- https://ww2.amstat.org/publications/jse/v19n3/decock/DataDocumentation.txt

Description

- The data has 82 columns which include 23 nominal, 23 ordinal, 14 discrete, and 20 continuous variables (and 2 additional observation identifiers).
- The usual goal is to predict the `SalePrice` of a house.
- The package `AmesHousing` contains the raw data and also pre-processed versions of them.

Pre-processing

First construct a function that will be used to combine together levels with few observations, for a factor variable.

```
# Function to combine levels, with less than a specified number of observations, of a factor variable
# Inputs:
#   x = variable (must be a factor; if not, the original variable is returned)
#   nmin = levels with fewer than nmin observations will be combined
# Output:
#   Returns the same variable with a new level called "othcomb" replacing the combined levels (NA's are not af
comblev <- function(x, nmin) {
  if (!is.factor(x)) {
    return(x)
  }

  # Load the 'rockchalk' library for the 'combineLevels' function
  library(rockchalk)

  # Create a frequency table of factor levels
  ta <- table(x)

  # Combine levels with fewer than 'nmin' observations into a new level "othcomb"
  combineLevels(x, levs = names(table(x))[table(x) < nmin], newLabel = c("othcomb"))
}
```

```
# to apply it to a data frame "mat" and get a data frame as the result
# data.frame(lapply(mat,comblev,nmin=2))
```

Prepare the ames housing data set.

- This version of the data uses the ordered factor as numeric variables and the non-ordered factor are left as factors, but the levels with less than 30 observations are combined.
- In the end, we have **22** factors and **57** numeric covariates, and **1 target** "Sale_Price".

```
# load the library
library(AmesHousing)

# load the data in ordinal form
ames=make_ordinal_ames()
ames
```

```
## # A tibble: 2,930 x 81
##    MS_SubClass            MS_Zoning Lot_Frontage Lot_Area Street Alley Lot_Shape
##    <fct>                  <fct>            <dbl>    <int> <fct>  <fct> <ord>
##  1 One_Story_1946_and_Ne~ Resident~           141    31770 Pave   No_A~ Slightly~
##  2 One_Story_1946_and_Ne~ Resident~            80    11622 Pave   No_A~ Regular
##  3 One_Story_1946_and_Ne~ Resident~            81    14267 Pave   No_A~ Slightly~
##  4 One_Story_1946_and_Ne~ Resident~            93    11160 Pave   No_A~ Regular
##  5 Two_Story_1946_and_Ne~ Resident~            74    13830 Pave   No_A~ Slightly~
##  6 Two_Story_1946_and_Ne~ Resident~            78     9978 Pave   No_A~ Slightly~
##  7 One_Story_PUD_1946_an~ Resident~            41     4920 Pave   No_A~ Regular
##  8 One_Story_PUD_1946_an~ Resident~            43     5005 Pave   No_A~ Slightly~
##  9 One_Story_PUD_1946_an~ Resident~            39     5389 Pave   No_A~ Slightly~
## 10 Two_Story_1946_and_Ne~ Resident~            60     7500 Pave   No_A~ Regular
## # i 2,920 more rows
## # i 74 more variables: Land_Contour <ord>, Utilities <ord>, Lot_Config <fct>,
## #   Land_Slope <ord>, Neighborhood <fct>, Condition_1 <fct>, Condition_2 <fct>,
## #   Bldg_Type <fct>, House_Style <fct>, Overall_Qual <ord>, Overall_Cond <ord>,
## #   Year_Built <int>, Year_Remod_Add <int>, Roof_Style <fct>, Roof_Matl <fct>,
## #   Exterior_1st <fct>, Exterior_2nd <fct>, Mas_Vnr_Type <fct>,
## #   Mas_Vnr_Area <dbl>, Exter_Qual <ord>, Exter_Cond <ord>, ...
```

```
colnames(ames)
```

```
##  [1] "MS_SubClass"        "MS_Zoning"          "Lot_Frontage"
##  [4] "Lot_Area"           "Street"             "Alley"
##  [7] "Lot_Shape"          "Land_Contour"       "Utilities"
## [10] "Lot_Config"         "Land_Slope"         "Neighborhood"
## [13] "Condition_1"        "Condition_2"        "Bldg_Type"
## [16] "House_Style"        "Overall_Qual"       "Overall_Cond"
## [19] "Year_Built"         "Year_Remod_Add"     "Roof_Style"
## [22] "Roof_Matl"          "Exterior_1st"       "Exterior_2nd"
## [25] "Mas_Vnr_Type"       "Mas_Vnr_Area"       "Exter_Qual"
## [28] "Exter_Cond"         "Foundation"         "Bsmt_Qual"
## [31] "Bsmt_Cond"          "Bsmt_Exposure"      "BsmtFin_Type_1"
## [34] "BsmtFin_SF_1"       "BsmtFin_Type_2"     "BsmtFin_SF_2"
## [37] "Bsmt_Unf_SF"        "Total_Bsmt_SF"      "Heating"
## [40] "Heating_QC"         "Central_Air"        "Electrical"
## [43] "First_Flr_SF"       "Second_Flr_SF"      "Low_Qual_Fin_SF"
## [46] "Gr_Liv_Area"        "Bsmt_Full_Bath"     "Bsmt_Half_Bath"
## [49] "Full_Bath"          "Half_Bath"          "Bedroom_AbvGr"
## [52] "Kitchen_AbvGr"      "Kitchen_Qual"       "TotRms_AbvGrd"
```

```
## [55] "Functional"         "Fireplaces"         "Fireplace_Qu"
## [58] "Garage_Type"        "Garage_Finish"      "Garage_Cars"
## [61] "Garage_Area"        "Garage_Qual"        "Garage_Cond"
## [64] "Paved_Drive"        "Wood_Deck_SF"       "Open_Porch_SF"
## [67] "Enclosed_Porch"     "Three_season_porch" "Screen_Porch"
## [70] "Pool_Area"          "Pool_QC"            "Fence"
## [73] "Misc_Feature"       "Misc_Val"           "Mo_Sold"
## [76] "Year_Sold"          "Sale_Type"          "Sale_Condition"
## [79] "Sale_Price"         "Longitude"          "Latitude"
```

```
# remove an observation with a missing
ames=ames[!is.na(ames$Electrical),]

# remove the variable "Utilities" because it is almost constant
# with frequencies (1,1,2909).
ames$Utilities=NULL

# converts the target variable (1 = 1000)
ames$Sale_Price=ames$Sale_Price/1000

# get the names of the ordinal variables
ord_vars=vapply(ames, is.ordered, logical(1))

# converts the ordered factors to numeric (this preserves the ordering of the factor)
namored=names(ord_vars)[ord_vars]
ames[,namored]=data.frame(lapply(ames[,namored],as.numeric))

# get the names of the factor variables
fac_vars=vapply(ames, is.factor, logical(1))
namfac=names(fac_vars)[fac_vars]

# display the factor variables
head(namfac)
```

```
## [1] "MS_SubClass"  "MS_Zoning"    "Street"       "Alley"        "Lot_Config"
## [6] "Neighborhood"
```

```
# group together levels with less than 30 observations
ames=data.frame(lapply(ames,comblev,nmin=30))
```

```
## The original levels One_Story_1946_and_Newer_All_Styles One_Story_1945_and_Older One_Story_with_Finished_Att
## have been replaced by One_Story_1946_and_Newer_All_Styles One_Story_1945_and_Older One_and_Half_Story_Finish
## The original levels Floating_Village_Residential Residential_High_Density Residential_Low_Density Residentia
## have been replaced by Floating_Village_Residential Residential_Low_Density Residential_Medium_Density othcom
## The original levels Grvl Pave
## have been replaced by Pave othcomb
## The original levels Gravel No_Alley_Access Paved
## have been replaced by Gravel No_Alley_Access Paved
## The original levels Corner CulDSac FR2 FR3 Inside
## have been replaced by Corner CulDSac FR2 Inside othcomb
## The original levels North_Ames College_Creek Old_Town Edwards Somerset Northridge_Heights Gilbert Sawyer Nor
## have been replaced by North_Ames College_Creek Old_Town Edwards Somerset Northridge_Heights Gilbert Sawyer N
## The original levels Artery Feedr Norm PosA PosN RRAe RRAn RRNe RRNn
## have been replaced by Artery Feedr Norm PosN RRAn othcomb
## The original levels Artery Feedr Norm PosA PosN RRAe RRAn RRNn
## have been replaced by Norm othcomb
## The original levels OneFam TwoFmCon Duplex Twnhs TwnhsE
## have been replaced by OneFam TwoFmCon Duplex Twnhs TwnhsE
## The original levels One_and_Half_Fin One_and_Half_Unf One_Story SFoyer SLvl Two_and_Half_Fin Two_and_Half_Un
```

```
## have been replaced by One_and_Half_Fin One_Story SFoyer SLvl Two_Story othcomb
## The original levels Flat Gable Gambrel Hip Mansard Shed
## have been replaced by Gable Hip othcomb
## The original levels ClyTile CompShg Membran Metal Roll Tar&Grv WdShake WdShngl
## have been replaced by CompShg othcomb
## The original levels AsbShng AsphShn BrkComm BrkFace CBlock CemntBd HdBoard ImStucc MetalSd Plywood PreCast S
## have been replaced by AsbShng BrkFace CemntBd HdBoard MetalSd Plywood Stucco VinylSd Wd Sdng WdShing othcomb
## The original levels AsbShng AsphShn Brk Cmn BrkFace CBlock CmentBd HdBoard ImStucc MetalSd Other Plywood Pre
## have been replaced by AsbShng BrkFace CmentBd HdBoard MetalSd Plywood Stucco VinylSd Wd Sdng Wd Shng othcomb
## The original levels BrkCmn BrkFace CBlock None Stone
## have been replaced by BrkFace None Stone othcomb
## The original levels BrkTil CBlock PConc Slab Stone Wood
## have been replaced by BrkTil CBlock PConc Slab othcomb
## The original levels Floor GasA GasW Grav OthW Wall
## have been replaced by GasA othcomb
## The original levels N Y
## have been replaced by N Y
## The original levels Attchd Basment BuiltIn CarPort Detchd More_Than_Two_Types No_Garage
## have been replaced by Attchd Basment BuiltIn Detchd No_Garage othcomb
## The original levels Elev Gar2 None Othr Shed TenC
## have been replaced by None Shed othcomb
## The original levels COD Con ConLD ConLI ConLw CWD New Oth VWD WD
## have been replaced by COD New WD  othcomb
## The original levels Abnorml AdjLand Alloca Family Normal Partial
## have been replaced by Abnorml Family Normal Partial othcomb
```

```
# remove the space in the values (string) of some variables to prevent problems later
ames[,"Exterior_1st"]=as.factor(gsub(" ","",ames[,"Exterior_1st"]))
ames[,"Exterior_2nd"]=as.factor(gsub(" ","",ames[,"Exterior_2nd"]))

# get the names of the factor variables
num_vars=vapply(ames, is.numeric, logical(1))

# names of the numeric variables
namnum=names(num_vars)[num_vars]
```

```
head(ames)
```

```
##                          MS_SubClass               MS_Zoning Lot_Frontage
## 1 One_Story_1946_and_Newer_All_Styles Residential_Low_Density          141
## 2 One_Story_1946_and_Newer_All_Styles                 othcomb           80
## 3 One_Story_1946_and_Newer_All_Styles Residential_Low_Density           81
## 4 One_Story_1946_and_Newer_All_Styles Residential_Low_Density           93
## 5            Two_Story_1946_and_Newer Residential_Low_Density           74
## 6            Two_Story_1946_and_Newer Residential_Low_Density           78
##   Lot_Area Street           Alley Lot_Shape Land_Contour Lot_Config Land_Slope
## 1    31770   Pave No_Alley_Access         3            4     Corner          3
## 2    11622   Pave No_Alley_Access         4            4     Inside          3
## 3    14267   Pave No_Alley_Access         3            4     Corner          3
## 4    11160   Pave No_Alley_Access         4            4     Corner          3
## 5    13830   Pave No_Alley_Access         3            4     Inside          3
## 6     9978   Pave No_Alley_Access         3            4     Inside          3
##   Neighborhood Condition_1 Condition_2 Bldg_Type House_Style Overall_Qual
## 1   North_Ames        Norm        Norm    OneFam   One_Story            6
## 2   North_Ames       Feedr        Norm    OneFam   One_Story            5
## 3   North_Ames        Norm        Norm    OneFam   One_Story            6
## 4   North_Ames        Norm        Norm    OneFam   One_Story            7
## 5      Gilbert        Norm        Norm    OneFam   Two_Story            5
## 6      Gilbert        Norm        Norm    OneFam   Two_Story            6
##   Overall_Cond Year_Built Year_Remod_Add Roof_Style Roof_Matl Exterior_1st
```

```
## 1            5       1960           1960        Hip   CompShg      BrkFace
## 2            6       1961           1961      Gable   CompShg      VinylSd
## 3            6       1958           1958        Hip   CompShg       WdSdng
## 4            5       1968           1968        Hip   CompShg      BrkFace
## 5            5       1997           1998      Gable   CompShg      VinylSd
## 6            6       1998           1998      Gable   CompShg      VinylSd
##   Exterior_2nd Mas_Vnr_Type Mas_Vnr_Area Exter_Qual Exter_Cond Foundation
## 1      Plywood        Stone          112          3          3     CBlock
## 2      VinylSd         None            0          3          3     CBlock
## 3       WdSdng      BrkFace          108          3          3     CBlock
## 4      BrkFace         None            0          4          3     CBlock
## 5      VinylSd         None            0          3          3      PConc
## 6      VinylSd      BrkFace           20          3          3      PConc
##   Bsmt_Qual Bsmt_Cond Bsmt_Exposure BsmtFin_Type_1 BsmtFin_SF_1 BsmtFin_Type_2
## 1         4         5             5              5           2              2
## 2         4         4             2              4           6              3
## 3         4         4             2              6           1              2
## 4         4         4             2              6           1              2
## 5         5         4             2              7           3              2
## 6         4         4             2              7           3              2
##   BsmtFin_SF_2 Bsmt_Unf_SF Total_Bsmt_SF Heating Heating_QC Central_Air
## 1            0         441          1080    GasA          2           Y
## 2          144         270           882    GasA          3           Y
## 3            0         406          1329    GasA          3           Y
## 4            0        1045          2110    GasA          5           Y
## 5            0         137           928    GasA          4           Y
## 6            0         324           926    GasA          5           Y
##   Electrical First_Flr_SF Second_Flr_SF Low_Qual_Fin_SF Gr_Liv_Area
## 1          5         1656             0               0        1656
## 2          5          896             0               0         896
## 3          5         1329             0               0        1329
## 4          5         2110             0               0        2110
## 5          5          928           701               0        1629
## 6          5          926           678               0        1604
##   Bsmt_Full_Bath Bsmt_Half_Bath Full_Bath Half_Bath Bedroom_AbvGr Kitchen_AbvGr
## 1              1              0         1         0             3             1
## 2              0              0         1         0             2             1
## 3              0              0         1         1             3             1
## 4              1              0         2         1             3             1
## 5              0              0         2         1             3             1
## 6              0              0         2         1             3             1
##   Kitchen_Qual TotRms_AbvGrd Functional Fireplaces Fireplace_Qu Garage_Type
## 1            3             7          8          2           5      Attchd
## 2            3             5          8          0           1      Attchd
## 3            4             6          8          0           1      Attchd
## 4            5             8          8          2           4      Attchd
## 5            3             6          8          1           4      Attchd
## 6            4             7          8          1           5      Attchd
##   Garage_Finish Garage_Cars Garage_Area Garage_Qual Garage_Cond Paved_Drive
## 1             4           2         528           4           4           2
## 2             2           1         730           4           4           3
## 3             2           1         312           4           4           3
## 4             4           2         522           4           4           3
## 5             4           2         482           4           4           3
## 6             4           2         470           4           4           3
##   Wood_Deck_SF Open_Porch_SF Enclosed_Porch Three_season_porch Screen_Porch
## 1          210            62              0                  0           0
## 2          140             0              0                  0         120
## 3          393            36              0                  0           0
## 4            0             0              0                  0           0
```

```
## 5          212            34              0                 0            0
## 6          360            36              0                 0            0
##   Pool_Area Pool_QC Fence Misc_Feature Misc_Val Mo_Sold Year_Sold Sale_Type
## 1         0       1     1         None        0       5      2010       WD
## 2         0       1     4         None        0       6      2010       WD
## 3         0       1     1      othcomb    12500       6      2010       WD
## 4         0       1     1         None        0       4      2010       WD
## 5         0       1     4         None        0       3      2010       WD
## 6         0       1     1         None        0       6      2010       WD
##   Sale_Condition Sale_Price Longitude Latitude
## 1         Normal      215.0 -93.61975 42.05403
## 2         Normal      105.0 -93.61976 42.05301
## 3         Normal      172.0 -93.61939 42.05266
## 4         Normal      244.0 -93.61732 42.05125
## 5         Normal      189.9 -93.63893 42.06090
## 6         Normal      195.5 -93.63893 42.06078
```

- This version of the data set, `ames`, has **2929 observations** and contains **22 factors** and **58 numeric covariates**, including **1 target Sale Price**.
- As explained in the `make_ames` function documentation, some observations and variables were removed, and 2 new variables were added.
- The factors have been consolidated. All levels **with less than 30 observations** are grouped together.

ANother version of the data: `amesdum`

We also prepare another version of the data, `amesdum` where the **factor variables are replaced by dummy variables**.

```
# Load the 'fastDummies' library for creating dummy variables
library(fastDummies)
```

```
## Thank you for using fastDummies!

## To acknowledge our work, please cite the package:

## Kaplan, J. & Schlegel, B. (2023). fastDummies: Fast Creation of Dummy (Binary) Columns and Rows from Categor
```

```
# Create dummy variables for the factors in the 'ames' dataset
amesdum = dummy_cols(ames, remove_first_dummy = TRUE, remove_selected_columns = TRUE)

# Check the dimensions of the 'amesdum' dataset
# It has 2929 rows and 161 columns (160 covariates and 1 target "Sale_Price")
dim(amesdum)
```

```
## [1] 2929  161
```

```
# Display a summary of the 'amesdum' dataset
summary(amesdum)
```

```
##   Lot_Frontage       Lot_Area        Lot_Shape      Land_Contour
##  Min.   :  0.00   Min.   :  1300   Min.   :1.000   Min.   :1.000
##  1st Qu.: 43.00   1st Qu.:  7440   1st Qu.:3.000   1st Qu.:4.000
##  Median : 63.00   Median :  9434   Median :4.000   Median :4.000
##  Mean   : 57.64   Mean   : 10148   Mean   :3.597   Mean   :3.817
##  3rd Qu.: 78.00   3rd Qu.: 11556   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :313.00   Max.   :215245   Max.   :4.000   Max.   :4.000
##    Land_Slope     Overall_Qual     Overall_Cond     Year_Built   Year_Remod_Add
##  Min.   :1.000   Min.   : 1.000   Min.   :1.000   Min.   :1872   Min.   :1950
```

```
##  1st Qu.:3.000   1st Qu.: 5.000   1st Qu.:5.000   1st Qu.:1954   1st Qu.:1965
##  Median :3.000   Median : 6.000   Median :5.000   Median :1973   Median :1993
##  Mean   :2.946   Mean   : 6.095   Mean   :5.563   Mean   :1971   Mean   :1984
##  3rd Qu.:3.000   3rd Qu.: 7.000   3rd Qu.:6.000   3rd Qu.:2001   3rd Qu.:2004
##  Max.   :3.000   Max.   :10.000   Max.   :9.000   Max.   :2010   Max.   :2010
##   Mas_Vnr_Area      Exter_Qual      Exter_Cond      Bsmt_Qual
##  Min.   :   0.0   Min.   :2.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:   0.0   1st Qu.:3.000   1st Qu.:3.000   1st Qu.:4.000
##  Median :   0.0   Median :3.000   Median :3.000   Median :5.000
##  Mean   : 101.1   Mean   :3.399   Mean   :3.085   Mean   :4.479
##  3rd Qu.: 163.0   3rd Qu.:4.000   3rd Qu.:3.000   3rd Qu.:5.000
##  Max.   :1600.0   Max.   :5.000   Max.   :5.000   Max.   :6.000
##    Bsmt_Cond     Bsmt_Exposure  BsmtFin_Type_1  BsmtFin_SF_1   BsmtFin_Type_2
##  Min.   :1.000   Min.   :1.00   Min.   :1.00   Min.   :0.000   Min.   :1.000
##  1st Qu.:4.000   1st Qu.:2.00   1st Qu.:2.00   1st Qu.:3.000   1st Qu.:2.000
##  Median :4.000   Median :2.00   Median :5.00   Median :3.000   Median :2.000
##  Mean   :3.923   Mean   :2.63   Mean   :4.55   Mean   :4.177   Mean   :2.275
##  3rd Qu.:4.000   3rd Qu.:3.00   3rd Qu.:7.00   3rd Qu.:7.000   3rd Qu.:2.000
##  Max.   :6.000   Max.   :5.00   Max.   :7.00   Max.   :7.000   Max.   :7.000
##   BsmtFin_SF_2      Bsmt_Unf_SF     Total_Bsmt_SF    Heating_QC
##  Min.   :   0.00   Min.   :   0.0   Min.   :   0   Min.   :1.00
##  1st Qu.:   0.00   1st Qu.: 219.0   1st Qu.: 793   1st Qu.:3.00
##  Median :   0.00   Median : 466.0   Median : 990   Median :5.00
##  Mean   :  49.72   Mean   : 559.1   Mean   :1051   Mean   :4.15
##  3rd Qu.:   0.00   3rd Qu.: 802.0   3rd Qu.:1302   3rd Qu.:5.00
##  Max.   :1526.00   Max.   :2336.0   Max.   :6110   Max.   :5.00
##    Electrical     First_Flr_SF  Second_Flr_SF    Low_Qual_Fin_SF
##  Min.   :1.000   Min.   : 334   Min.   :   0.0   Min.   :   0.000
##  1st Qu.:5.000   1st Qu.: 877   1st Qu.:   0.0   1st Qu.:   0.000
##  Median :5.000   Median :1084   Median :   0.0   Median :   0.000
##  Mean   :4.892   Mean   :1160   Mean   : 335.4   Mean   :   4.678
##  3rd Qu.:5.000   3rd Qu.:1384   3rd Qu.: 704.0   3rd Qu.:   0.000
##  Max.   :5.000   Max.   :5095   Max.   :2065.0   Max.   :1064.000
##   Gr_Liv_Area   Bsmt_Full_Bath   Bsmt_Half_Bath      Full_Bath
##  Min.   : 334   Min.   :0.0000   Min.   :0.00000   Min.   :0.000
##  1st Qu.:1126   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:1.000
##  Median :1442   Median :0.0000   Median :0.00000   Median :2.000
##  Mean   :1500   Mean   :0.4312   Mean   :0.06111   Mean   :1.566
##  3rd Qu.:1743   3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:2.000
##  Max.   :5642   Max.   :3.0000   Max.   :2.00000   Max.   :4.000
##    Half_Bath      Bedroom_AbvGr   Kitchen_AbvGr    Kitchen_Qual
##  Min.   :0.0000   Min.   :0.000   Min.   :0.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:3.000
##  Median :0.0000   Median :3.000   Median :1.000   Median :3.000
##  Mean   :0.3793   Mean   :2.854   Mean   :1.044   Mean   :3.511
##  3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:1.000   3rd Qu.:4.000
##  Max.   :2.0000   Max.   :8.000   Max.   :3.000   Max.   :5.000
##  TotRms_AbvGrd      Functional      Fireplaces      Fireplace_Qu
##  Min.   : 2.000   Min.   :1.000   Min.   :0.0000   Min.   :1.000
##  1st Qu.: 5.000   1st Qu.:8.000   1st Qu.:0.0000   1st Qu.:1.000
##  Median : 6.000   Median :8.000   Median :1.0000   Median :2.000
##  Mean   : 6.443   Mean   :7.844   Mean   :0.5995   Mean   :2.771
##  3rd Qu.: 7.000   3rd Qu.:8.000   3rd Qu.:1.0000   3rd Qu.:5.000
##  Max.   :15.000   Max.   :8.000   Max.   :4.0000   Max.   :6.000
##  Garage_Finish    Garage_Cars     Garage_Area      Garage_Qual
##  Min.   :1.000   Min.   :0.000   Min.   :   0.0   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:1.000   1st Qu.: 320.0   1st Qu.:4.000
##  Median :3.000   Median :2.000   Median : 480.0   Median :4.000
##  Mean   :2.719   Mean   :1.766   Mean   : 472.7   Mean   :3.802
##  3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.: 576.0   3rd Qu.:4.000
```

```
##  Max.   :4.000   Max.   :5.000   Max.   :1488.0   Max.   :6.000
##   Garage_Cond     Paved_Drive     Wood_Deck_SF     Open_Porch_SF
##  Min.   :1.000   Min.   :1.000   Min.   :   0.00   Min.   :  0.00
##  1st Qu.:4.000   1st Qu.:3.000   1st Qu.:   0.00   1st Qu.:  0.00
##  Median :4.000   Median :3.000   Median :   0.00   Median : 27.00
##  Mean   :3.809   Mean   :2.831   Mean   :  93.75   Mean   : 47.55
##  3rd Qu.:4.000   3rd Qu.:3.000   3rd Qu.: 168.00   3rd Qu.: 70.00
##  Max.   :6.000   Max.   :3.000   Max.   :1424.00   Max.   :742.00
##  Enclosed_Porch    Three_season_porch  Screen_Porch      Pool_Area
##  Min.   :   0.00   Min.   :  0.000    Min.   :  0.00   Min.   :  0.000
##  1st Qu.:   0.00   1st Qu.:  0.000    1st Qu.:  0.00   1st Qu.:  0.000
##  Median :   0.00   Median :  0.000    Median :  0.00   Median :  0.000
##  Mean   :  23.02   Mean   :  2.593    Mean   : 16.01   Mean   :  2.244
##  3rd Qu.:   0.00   3rd Qu.:  0.000    3rd Qu.:  0.00   3rd Qu.:  0.000
##  Max.   :1012.00   Max.   :508.000    Max.   :576.00   Max.   :800.000
##     Pool_QC          Fence         Misc_Val           Mo_Sold
##  Min.   :1.000   Min.   :1.00   Min.   :    0.00   Min.   : 1.000
##  1st Qu.:1.000   1st Qu.:1.00   1st Qu.:    0.00   1st Qu.: 4.000
##  Median :1.000   Median :1.00   Median :    0.00   Median : 6.000
##  Mean   :1.017   Mean   :1.58   Mean   :   50.65   Mean   : 6.216
##  3rd Qu.:1.000   3rd Qu.:1.00   3rd Qu.:    0.00   3rd Qu.: 8.000
##  Max.   :6.000   Max.   :5.00   Max.   :17000.00   Max.   :12.000
##    Year_Sold      Sale_Price       Longitude         Latitude
##  Min.   :2006   Min.   : 12.79   Min.   :-93.69   Min.   :41.99
##  1st Qu.:2007   1st Qu.:129.50   1st Qu.:-93.66   1st Qu.:42.02
##  Median :2008   Median :160.00   Median :-93.64   Median :42.03
##  Mean   :2008   Mean   :180.80   Mean   :-93.64   Mean   :42.03
##  3rd Qu.:2009   3rd Qu.:213.50   3rd Qu.:-93.62   3rd Qu.:42.05
##  Max.   :2010   Max.   :755.00   Max.   :-93.58   Max.   :42.06
##  MS_SubClass_One_Story_1945_and_Older
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.04746
##  3rd Qu.:0.00000
##  Max.   :1.00000
##  MS_SubClass_One_and_Half_Story_Finished_All_Ages
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.09799
##  3rd Qu.:0.00000
##  Max.   :1.00000
##  MS_SubClass_Two_Story_1946_and_Newer MS_SubClass_Two_Story_1945_and_Older
##  Min.   :0.0000                       Min.   :0.0000
##  1st Qu.:0.0000                       1st Qu.:0.0000
##  Median :0.0000                       Median :0.0000
##  Mean   :0.1963                       Mean   :0.0437
##  3rd Qu.:0.0000                       3rd Qu.:0.0000
##  Max.   :1.0000                       Max.   :1.0000
##  MS_SubClass_Split_or_Multilevel MS_SubClass_Split_Foyer
##  Min.   :0.00000                 Min.   :0.00000
##  1st Qu.:0.00000                 1st Qu.:0.00000
##  Median :0.00000                 Median :0.00000
##  Mean   :0.03995                 Mean   :0.01639
##  3rd Qu.:0.00000                 3rd Qu.:0.00000
##  Max.   :1.00000                 Max.   :1.00000
##  MS_SubClass_Duplex_All_Styles_and_Ages
##  Min.   :0.00000
##  1st Qu.:0.00000
```

```
##  Median :0.00000
##  Mean   :0.03721
##  3rd Qu.:0.00000
##  Max.   :1.00000
##  MS_SubClass_One_Story_PUD_1946_and_Newer
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.06555
##  3rd Qu.:0.00000
##  Max.   :1.00000
##  MS_SubClass_Two_Story_PUD_1946_and_Newer
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.04404
##  3rd Qu.:0.00000
##  Max.   :1.00000
##  MS_SubClass_Two_Family_conversion_All_Styles_and_Ages MS_SubClass_othcomb
##  Min.   :0.00000                                       Min.   :0.00000
##  1st Qu.:0.00000                                       1st Qu.:0.00000
##  Median :0.00000                                       Median :0.00000
##  Mean   :0.02083                                       Mean   :0.02219
##  3rd Qu.:0.00000                                       3rd Qu.:0.00000
##  Max.   :1.00000                                       Max.   :1.00000
##  MS_Zoning_Residential_Low_Density MS_Zoning_Residential_Medium_Density
##  Min.   :0.0000                    Min.   :0.0000
##  1st Qu.:1.0000                    1st Qu.:0.0000
##  Median :1.0000                    Median :0.0000
##  Mean   :0.7757                    Mean   :0.1577
##  3rd Qu.:1.0000                    3rd Qu.:0.0000
##  Max.   :1.0000                    Max.   :1.0000
##  MS_Zoning_othcomb Street_othcomb     Alley_No_Alley_Access  Alley_Paved
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.0000        Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:1.0000        1st Qu.:0.00000
##  Median :0.00000   Median :0.000000   Median :1.0000        Median :0.00000
##  Mean   :0.01912   Mean   :0.004097   Mean   :0.9324        Mean   :0.02663
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:1.0000        3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.000000   Max.   :1.0000        Max.   :1.00000
##  Lot_Config_CulDSac Lot_Config_FR2    Lot_Config_Inside Lot_Config_othcomb
##  Min.   :0.00000    Min.   :0.00000   Min.   :0.0000    Min.   :0.00000
##  1st Qu.:0.00000    1st Qu.:0.00000   1st Qu.:0.0000    1st Qu.:0.00000
##  Median :0.00000    Median :0.00000   Median :1.0000    Median :0.00000
##  Mean   :0.06145    Mean   :0.02902   Mean   :0.7303    Mean   :0.00478
##  3rd Qu.:0.00000    3rd Qu.:0.00000   3rd Qu.:1.0000    3rd Qu.:0.00000
##  Max.   :1.00000    Max.   :1.00000   Max.   :1.0000    Max.   :1.00000
##  Neighborhood_College_Creek Neighborhood_Old_Town Neighborhood_Edwards
##  Min.   :0.00000            Min.   :0.0000        Min.   :0.00000
##  1st Qu.:0.00000            1st Qu.:0.0000        1st Qu.:0.00000
##  Median :0.00000            Median :0.0000        Median :0.00000
##  Mean   :0.09116            Mean   :0.0816        Mean   :0.06623
##  3rd Qu.:0.00000            3rd Qu.:0.0000        3rd Qu.:0.00000
##  Max.   :1.00000            Max.   :1.0000        Max.   :1.00000
##  Neighborhood_Somerset Neighborhood_Northridge_Heights Neighborhood_Gilbert
##  Min.   :0.00000       Min.   :0.00000                 Min.   :0.00000
##  1st Qu.:0.00000       1st Qu.:0.00000                 1st Qu.:0.00000
##  Median :0.00000       Median :0.00000                 Median :0.00000
##  Mean   :0.06214       Mean   :0.05667                 Mean   :0.05633
##  3rd Qu.:0.00000       3rd Qu.:0.00000                 3rd Qu.:0.00000
##  Max.   :1.00000       Max.   :1.00000                 Max.   :1.00000
```

```
##  Neighborhood_Sawyer Neighborhood_Northwest_Ames Neighborhood_Sawyer_West
##  Min.   :0.00000     Min.   :0.00000             Min.   :0.00000
##  1st Qu.:0.00000     1st Qu.:0.00000             1st Qu.:0.00000
##  Median :0.00000     Median :0.00000             Median :0.00000
##  Mean   :0.05155     Mean   :0.04473             Mean   :0.04268
##  3rd Qu.:0.00000     3rd Qu.:0.00000             3rd Qu.:0.00000
##  Max.   :1.00000     Max.   :1.00000             Max.   :1.00000
##  Neighborhood_Mitchell Neighborhood_Brookside Neighborhood_Crawford
##  Min.   :0.00000       Min.   :0.00000        Min.   :0.00000
##  1st Qu.:0.00000       1st Qu.:0.00000        1st Qu.:0.00000
##  Median :0.00000       Median :0.00000        Median :0.00000
##  Mean   :0.03892       Mean   :0.03687        Mean   :0.03517
##  3rd Qu.:0.00000       3rd Qu.:0.00000        3rd Qu.:0.00000
##  Max.   :1.00000       Max.   :1.00000        Max.   :1.00000
##  Neighborhood_Iowa_DOT_and_Rail_Road Neighborhood_Timberland
##  Min.   :0.00000                     Min.   :0.00000
##  1st Qu.:0.00000                     1st Qu.:0.00000
##  Median :0.00000                     Median :0.00000
##  Mean   :0.03175                     Mean   :0.02424
##  3rd Qu.:0.00000                     3rd Qu.:0.00000
##  Max.   :1.00000                     Max.   :1.00000
##  Neighborhood_Northridge Neighborhood_Stone_Brook
##  Min.   :0.00000         Min.   :0.00000
##  1st Qu.:0.00000         1st Qu.:0.00000
##  Median :0.00000         Median :0.00000
##  Mean   :0.02424         Mean   :0.01741
##  3rd Qu.:0.00000         3rd Qu.:0.00000
##  Max.   :1.00000         Max.   :1.00000
##  Neighborhood_South_and_West_of_Iowa_State_University Neighborhood_Clear_Creek
##  Min.   :0.00000                                      Min.   :0.00000
##  1st Qu.:0.00000                                      1st Qu.:0.00000
##  Median :0.00000                                      Median :0.00000
##  Mean   :0.01639                                      Mean   :0.01502
##  3rd Qu.:0.00000                                      3rd Qu.:0.00000
##  Max.   :1.00000                                      Max.   :1.00000
##  Neighborhood_Meadow_Village Neighborhood_Briardale Neighborhood_othcomb
##  Min.   :0.00000             Min.   :0.00000        Min.   :0.00000
##  1st Qu.:0.00000             1st Qu.:0.00000        1st Qu.:0.00000
##  Median :0.00000             Median :0.00000        Median :0.00000
##  Mean   :0.01263             Mean   :0.01024        Mean   :0.03278
##  3rd Qu.:0.00000             3rd Qu.:0.00000        3rd Qu.:0.00000
##  Max.   :1.00000             Max.   :1.00000        Max.   :1.00000
##  Condition_1_Feedr Condition_1_Norm Condition_1_PosN  Condition_1_RRAn
##  Min.   :0.00000   Min.   :0.0000  Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:1.0000  1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :1.0000  Median :0.00000   Median :0.00000
##  Mean   :0.05599   Mean   :0.8607  Mean   :0.01332   Mean   :0.01707
##  3rd Qu.:0.00000   3rd Qu.:1.0000  3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.0000  Max.   :1.00000   Max.   :1.00000
##  Condition_1_othcomb Condition_2_othcomb Bldg_Type_TwoFmCon Bldg_Type_Duplex
##  Min.   :0.00000     Min.   :0.00000     Min.   :0.00000    Min.   :0.00000
##  1st Qu.:0.00000     1st Qu.:0.00000     1st Qu.:0.00000    1st Qu.:0.00000
##  Median :0.00000     Median :0.00000     Median :0.00000    Median :0.00000
##  Mean   :0.02151     Mean   :0.01024     Mean   :0.02117    Mean   :0.03721
##  3rd Qu.:0.00000     3rd Qu.:0.00000     3rd Qu.:0.00000    3rd Qu.:0.00000
##  Max.   :1.00000     Max.   :1.00000     Max.   :1.00000    Max.   :1.00000
##  Bldg_Type_Twnhs   Bldg_Type_TwnhsE  House_Style_One_Story House_Style_SFoyer
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.0000        Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000        1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :1.0000        Median :0.00000
```

```
##  Mean   :0.03448   Mean   :0.07955   Mean   :0.5056        Mean   :0.02834
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.0000        3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.0000        Max.   :1.00000
##  House_Style_SLvl  House_Style_Two_Story House_Style_othcomb Roof_Style_Hip
##  Min.   :0.00000   Min.   :0.0000        Min.   :0.00000     Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000        1st Qu.:0.00000     1st Qu.:0.0000
##  Median :0.00000   Median :0.0000        Median :0.00000     Median :0.0000
##  Mean   :0.04336   Mean   :0.2981        Mean   :0.01741     Mean   :0.1881
##  3rd Qu.:0.00000   3rd Qu.:1.0000        3rd Qu.:0.00000     3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :1.0000        Max.   :1.00000     Max.   :1.0000
##  Roof_Style_othcomb Roof_Matl_othcomb Exterior_1st_BrkFace Exterior_1st_CemntBd
##  Min.   :0.0000     Min.   :0.00000   Min.   :0.00000      Min.   :0.00000
##  1st Qu.:0.0000     1st Qu.:0.00000   1st Qu.:0.00000      1st Qu.:0.00000
##  Median :0.0000     Median :0.00000   Median :0.00000      Median :0.00000
##  Mean   :0.0198     Mean   :0.01468   Mean   :0.03004      Mean   :0.04302
##  3rd Qu.:0.0000     3rd Qu.:0.00000   3rd Qu.:0.00000      3rd Qu.:0.00000
##  Max.   :1.0000     Max.   :1.00000   Max.   :1.00000      Max.   :1.00000
##  Exterior_1st_HdBoard Exterior_1st_MetalSd Exterior_1st_othcomb
##  Min.   :0.0000       Min.   :0.0000       Min.   :0.00000
##  1st Qu.:0.0000       1st Qu.:0.0000       1st Qu.:0.00000
##  Median :0.0000       Median :0.0000       Median :0.00000
##  Mean   :0.1509       Mean   :0.1536       Mean   :0.00478
##  3rd Qu.:0.0000       3rd Qu.:0.0000       3rd Qu.:0.00000
##  Max.   :1.0000       Max.   :1.0000       Max.   :1.00000
##  Exterior_1st_Plywood Exterior_1st_Stucco Exterior_1st_VinylSd
##  Min.   :0.00000      Min.   :0.00000     Min.   :0.0000
##  1st Qu.:0.00000      1st Qu.:0.00000     1st Qu.:0.0000
##  Median :0.00000      Median :0.00000     Median :0.0000
##  Mean   :0.07545      Mean   :0.01468     Mean   :0.3499
##  3rd Qu.:0.00000      3rd Qu.:0.00000     3rd Qu.:1.0000
##  Max.   :1.00000      Max.   :1.00000     Max.   :1.0000
##  Exterior_1st_WdSdng Exterior_1st_WdShing Exterior_2nd_BrkFace
##  Min.   :0.0000      Min.   :0.00000      Min.   :0.00000
##  1st Qu.:0.0000      1st Qu.:0.00000      1st Qu.:0.00000
##  Median :0.0000      Median :0.00000      Median :0.00000
##  Mean   :0.1434      Mean   :0.01912      Mean   :0.01605
##  3rd Qu.:0.0000      3rd Qu.:0.00000      3rd Qu.:0.00000
##  Max.   :1.0000      Max.   :1.00000      Max.   :1.00000
##  Exterior_2nd_CmentBd Exterior_2nd_HdBoard Exterior_2nd_MetalSd
##  Min.   :0.00000      Min.   :0.0000       Min.   :0.0000
##  1st Qu.:0.00000      1st Qu.:0.0000       1st Qu.:0.0000
##  Median :0.00000      Median :0.0000       Median :0.0000
##  Mean   :0.04302      Mean   :0.1386       Mean   :0.1526
##  3rd Qu.:0.00000      3rd Qu.:0.0000       3rd Qu.:0.0000
##  Max.   :1.00000      Max.   :1.0000       Max.   :1.0000
##  Exterior_2nd_othcomb Exterior_2nd_Plywood Exterior_2nd_Stucco
##  Min.   :0.00000      Min.   :0.00000      Min.   :0.00000
##  1st Qu.:0.00000      1st Qu.:0.00000      1st Qu.:0.00000
##  Median :0.00000      Median :0.00000      Median :0.00000
##  Mean   :0.01775      Mean   :0.09355      Mean   :0.01605
##  3rd Qu.:0.00000      3rd Qu.:0.00000      3rd Qu.:0.00000
##  Max.   :1.00000      Max.   :1.00000      Max.   :1.00000
##  Exterior_2nd_VinylSd Exterior_2nd_WdSdng Exterior_2nd_WdShng Mas_Vnr_Type_None
##  Min.   :0.0000       Min.   :0.0000      Min.   :0.00000     Min.   :0.0000
##  1st Qu.:0.0000       1st Qu.:0.0000      1st Qu.:0.00000     1st Qu.:0.0000
##  Median :0.0000       Median :0.0000      Median :0.00000     Median :1.0000
##  Mean   :0.3462       Mean   :0.1355      Mean   :0.02765     Mean   :0.6057
##  3rd Qu.:1.0000       3rd Qu.:0.0000      3rd Qu.:0.00000     3rd Qu.:1.0000
##  Max.   :1.0000       Max.   :1.0000      Max.   :1.00000     Max.   :1.0000
##  Mas_Vnr_Type_Stone Mas_Vnr_Type_othcomb Foundation_CBlock Foundation_PConc
```

```
##  Min.   :0.00000    Min.   :0.000000     Min.   :0.0000    Min.   :0.0000
##  1st Qu.:0.00000    1st Qu.:0.000000     1st Qu.:0.0000    1st Qu.:0.0000
##  Median :0.00000    Median :0.000000     Median :0.0000    Median :0.0000
##  Mean   :0.08501    Mean   :0.008877     Mean   :0.4247    Mean   :0.4469
##  3rd Qu.:0.00000    3rd Qu.:0.000000     3rd Qu.:1.0000    3rd Qu.:1.0000
##  Max.   :1.00000    Max.   :1.000000     Max.   :1.0000    Max.   :1.0000
##  Foundation_Slab   Foundation_othcomb Heating_othcomb   Central_Air_Y
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:1.0000
##  Median :0.00000   Median :0.000000   Median :0.00000   Median :1.0000
##  Mean   :0.01673   Mean   :0.005463   Mean   :0.01536   Mean   :0.9331
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:1.0000
##  Max.   :1.00000   Max.   :1.000000   Max.   :1.00000   Max.   :1.0000
##  Garage_Type_Basment Garage_Type_BuiltIn Garage_Type_Detchd
##  Min.   :0.00000     Min.   :0.00000     Min.   :0.000
##  1st Qu.:0.00000     1st Qu.:0.00000     1st Qu.:0.000
##  Median :0.00000     Median :0.00000     Median :0.000
##  Mean   :0.01229     Mean   :0.06316     Mean   :0.267
##  3rd Qu.:0.00000     3rd Qu.:0.00000     3rd Qu.:1.000
##  Max.   :1.00000     Max.   :1.00000     Max.   :1.000
##  Garage_Type_No_Garage Garage_Type_othcomb Misc_Feature_Shed
##  Min.   :0.0000        Min.   :0.00000     Min.   :0.00000
##  1st Qu.:0.0000        1st Qu.:0.00000     1st Qu.:0.00000
##  Median :0.0000        Median :0.00000     Median :0.00000
##  Mean   :0.0536        Mean   :0.01297     Mean   :0.03243
##  3rd Qu.:0.0000        3rd Qu.:0.00000     3rd Qu.:0.00000
##  Max.   :1.0000        Max.   :1.00000     Max.   :1.00000
##  Misc_Feature_othcomb Sale_Type_New    Sale_Type_WD     Sale_Type_othcomb
##  Min.   :0.000000     Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.000000     1st Qu.:0.0000   1st Qu.:1.0000   1st Qu.:0.00000
##  Median :0.000000     Median :0.0000   Median :1.0000   Median :0.00000
##  Mean   :0.003756     Mean   :0.0816   Mean   :0.8655   Mean   :0.02322
##  3rd Qu.:0.000000     3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.00000
##  Max.   :1.000000     Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##  Sale_Condition_Family Sale_Condition_Normal Sale_Condition_Partial
##  Min.   :0.00000       Min.   :0.0000        Min.   :0.00000
##  1st Qu.:0.00000       1st Qu.:1.0000        1st Qu.:0.00000
##  Median :0.00000       Median :1.0000        Median :0.00000
##  Mean   :0.01571       Mean   :0.8235        Mean   :0.08365
##  3rd Qu.:0.00000       3rd Qu.:1.0000        3rd Qu.:0.00000
##  Max.   :1.00000       Max.   :1.0000        Max.   :1.00000
##  Sale_Condition_othcomb
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.01229
##  3rd Qu.:0.00000
##  Max.   :1.00000
```

Train-test split

```r
# Splitting the data into a training (ntrain=1000) and a test (ntest=1929) set

# Set a random seed for reproducibility
set.seed(489565)

# Define the number of training samples (ntrain) and test samples (ntest)
ntrain = 1000
ntest = nrow(ames) - ntrain
```

```r
# Randomly select 'ntrain' indices for the training set without replacement
indtrain = sample(1:nrow(ames), ntrain, replace = FALSE)

# Create a copy of 'amesdum' and remove the 'Sale_Price' column
xdum = amesdum
xdum$Sale_Price = NULL
xdum = as.matrix(xdum)

# Create separate datasets for training and testing
amestrain = ames[indtrain,]
amestest = ames[-indtrain,]
amesdumtrain = amesdum[indtrain,]
amesdumtest = amesdum[-indtrain,]
xdumtrain = xdum[indtrain,]
xdumtest = xdum[-indtrain,]
```

Data Analysis

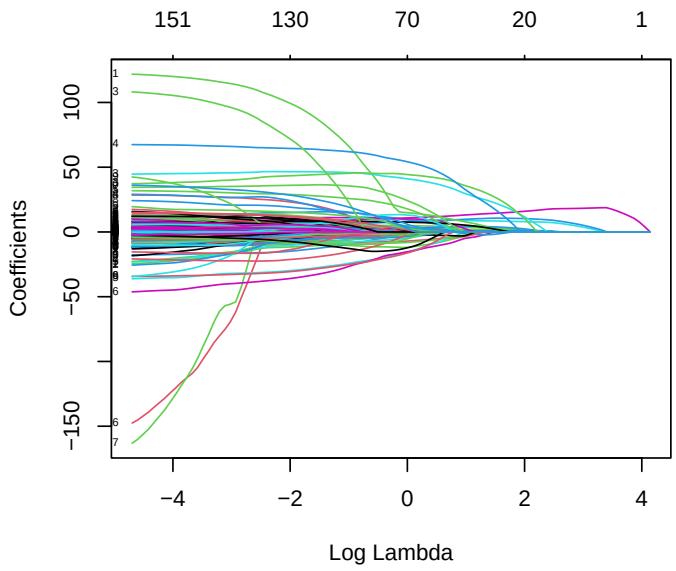
1. Next is first a simple wrapper function to apply `glmnet` and get the predictions and `coeficients` for the tuning parameter with minimum CV error, and with the 1 SE rule.
2. It also computes the MAE and MSE
3. It also produces some plots

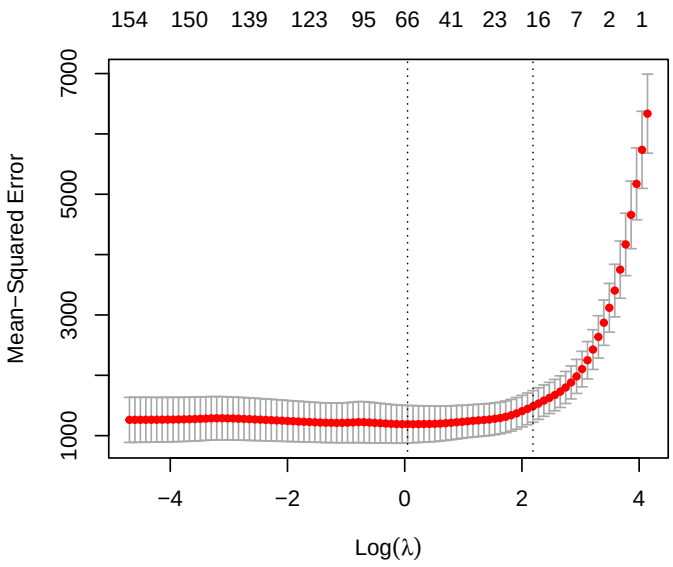
```r
# Variable selection examples, with all 161 covariates
# (including the dummies for the categorical covariates)

# Function to apply glmnet and get predictions, coefficients, and errors
# - xtrain: training data predictors
# - ytrain: training data target
# - xtest: test data predictors
# - ytest: test data target (optional)
# - alpha: alpha parameter for glmnet

wrapglmnet = function(xtrain, ytrain, xtest, ytest = NULL, alpha) {
  require(glmnet)

  # Set the layout for multiple plots
  par(mfrow = c(2, 2))

  # Plot the glmnet results
  plot(glmnet(x = xtrain, y = ytrain, alpha = alpha), xvar = "lambda", label = TRUE)

  # Cross-validate and plot the results
  cv = cv.glmnet(x = xtrain, y = ytrain, alpha = alpha)
  plot(cv)

  # Predict using lambda.min and lambda.1se
  pred = predict(cv, new = xtest, s = "lambda.min")
  pred1se = predict(cv, new = xtest, s = "lambda.1se") # this is calculate automatically by cv.glmnet

  # Initialize error metrics
  err = NA

  # If ytest is available, compute MAE and MSE
  if (!is.null(ytest)) {
    # plot predictions
    plot(ytest, pred)
    plot(ytest, pred1se)
```

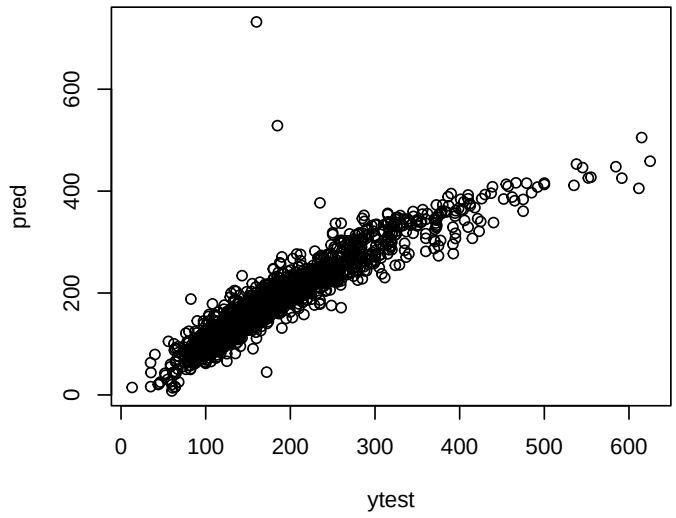
```
    # calculate errors for each framework
    err = data.frame(
      mean(abs(pred - ytest)), mean((pred - ytest)^2),
      mean(abs(pred1se - ytest)), mean((pred1se - ytest)^2)
    )
    names(err) = c("MAE", "MSE", "MAE_1SE", "MSE_1SE")
  }

  # Get coefficients for lambda.min and lambda.1se
  co = predict(cv, s = "lambda.min", type = "coefficients")
  co = as.matrix(co)
  co = co[co[, 1] != 0,, drop = FALSE]

  co1se = predict(cv, s = "lambda.1se", type = "coefficients")
  co1se = as.matrix(co1se)
  co1se = co1se[co1se[, 1] != 0,, drop = FALSE]

  # Create a list of results
  out = list(err, co, co1se, pred, pred1se)
  names(out) = c("error", "coef", "coef1se", "pred", "pred1se")

  # Return the results
  out
}
```

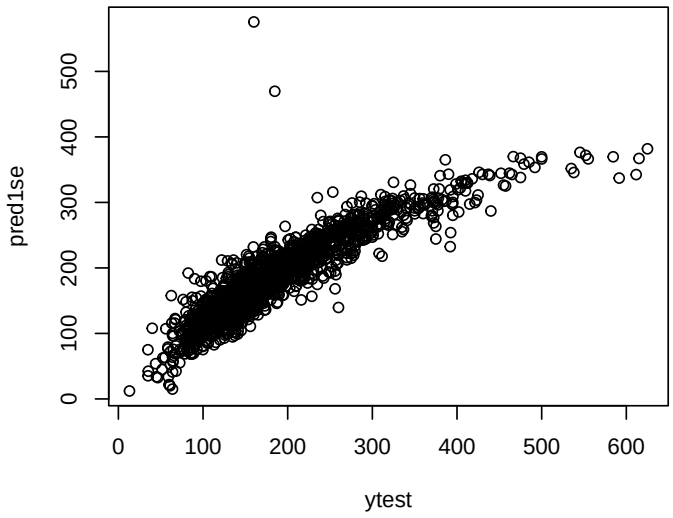
Lasso

```
# basic lasso
las=wrapglmnet(xdumtrain, # xtrain
               amesdumtrain$Sale_Price, # ytrain
               xdumtest, # xtest
               amesdumtest$Sale_Price, # ytest
               1) # alpha glmnet
```










```
# number of coefficients with 1SE vs min lambda
dim(las$coef1se)
```

```
## [1] 18  1
```

```
dim(las$coef)
```

```
## [1] 67  1
```

```
# COefficients for 1SE rule
las$coef1se
```

```
##                                lambda.1se
## (Intercept)                -1.240019e+02
## Overall_Qual                1.665411e+01
## Year_Built                  1.381639e-02
## Exter_Qual                  7.721331e+00
```

```
## Bsmt_Qual                          6.864769e-01
## Bsmt_Exposure                      9.889567e-01
## BsmtFin_Type_1                     5.971854e-01
## Total_Bsmt_SF                      1.308751e-02
## First_Flr_SF                       5.651597e-03
## Gr_Liv_Area                        4.322929e-02
## Bsmt_Full_Bath                     1.156929e+00
## Kitchen_Qual                       9.917927e+00
## Fireplace_Qu                       4.793228e-01
## Garage_Finish                      8.154756e-01
## Garage_Cars                        1.068530e+00
## Garage_Area                        3.054890e-02
## Neighborhood_Northridge_Heights    6.274777e+00
## Neighborhood_Northridge            1.359554e+00
```

```
# coefficients without the rule
las$coef
```

```
##                                                  lambda.min
## (Intercept)                                  -6.205519e+02
## Lot_Frontage                                  8.554987e-03
## Lot_Area                                      5.081698e-04
## Land_Slope                                   -1.326315e+00
## Overall_Qual                                  1.093504e+01
## Overall_Cond                                  3.309886e+00
## Year_Built                                    1.741835e-01
## Year_Remod_Add                                7.311357e-02
## Exter_Qual                                    8.395572e+00
## Bsmt_Qual                                     7.895797e-01
## Bsmt_Cond                                    -3.040714e-01
## Bsmt_Exposure                                 4.851189e+00
## BsmtFin_Type_1                                6.056344e-01
## Bsmt_Unf_SF                                  -1.084490e-02
## Total_Bsmt_SF                                 1.901530e-02
## Heating_QC                                    3.068524e-01
## Low_Qual_Fin_SF                              -8.442164e-03
## Gr_Liv_Area                                   5.415002e-02
## Bsmt_Full_Bath                                2.927332e+00
## Full_Bath                                     1.059828e+00
## Bedroom_AbvGr                                -1.206071e+00
## Kitchen_AbvGr                                -1.170498e+01
## Kitchen_Qual                                  7.642529e+00
## Functional                                    3.060611e+00
## Fireplace_Qu                                  1.336560e+00
## Garage_Finish                                 1.764630e+00
## Garage_Cars                                   2.371741e+00
## Garage_Area                                   1.904856e-02
## Screen_Porch                                  4.002882e-02
## Pool_QC                                       1.468140e+01
## Misc_Val                                     -9.874349e-03
## MS_SubClass_One_Story_PUD_1946_and_Newer     -1.489545e+01
## MS_SubClass_Two_Story_PUD_1946_and_Newer     -1.495072e+01
## Lot_Config_FR2                               -1.273378e+00
## Neighborhood_Old_Town                        -6.881164e+00
## Neighborhood_Edwards                         -7.945886e+00
## Neighborhood_Somerset                         9.175624e+00
## Neighborhood_Northridge_Heights               4.072750e+01
## Neighborhood_Northwest_Ames                  -1.557089e+00
## Neighborhood_Sawyer_West                     -2.426533e+00
## Neighborhood_Crawford                         1.006086e+01
```

```
## Neighborhood_Northridge                                    4.410654e+01
## Neighborhood_Stone_Brook                                   5.367552e+01
## Neighborhood_South_and_West_of_Iowa_State_University -4.335001e+00
## Neighborhood_Clear_Creek                                   7.320059e-01
## Condition_1_Norm                                           3.270997e+00
## Condition_2_othcomb                                        1.707547e+01
## Bldg_Type_TwnhsE                                          -1.577680e+00
## House_Style_One_Story                                      1.802063e+00
## House_Style_SLvl                                          -3.313434e+00
## Roof_Style_Hip                                             7.533545e+00
## Roof_Style_othcomb                                        -1.559362e+01
## Roof_Matl_othcomb                                          2.327010e+01
## Exterior_1st_BrkFace                                       5.080142e+00
## Exterior_1st_Stucco                                        3.416659e+00
## Exterior_2nd_BrkFace                                      -9.382322e-01
## Exterior_2nd_CmentBd                                       1.616271e+00
## Exterior_2nd_othcomb                                       8.597813e-01
## Mas_Vnr_Type_Stone                                         1.353848e+00
## Foundation_CBlock                                         -1.613384e+00
## Foundation_Slab                                            1.288378e+01
## Garage_Type_Basment                                       -4.670047e+00
## Garage_Type_othcomb                                       -1.170163e+01
## Sale_Type_New                                              2.530036e+00
## Sale_Condition_Normal                                      1.219544e+00
## Sale_Condition_Partial                                     2.357320e+00
## Sale_Condition_othcomb                                     1.780051e+00
```

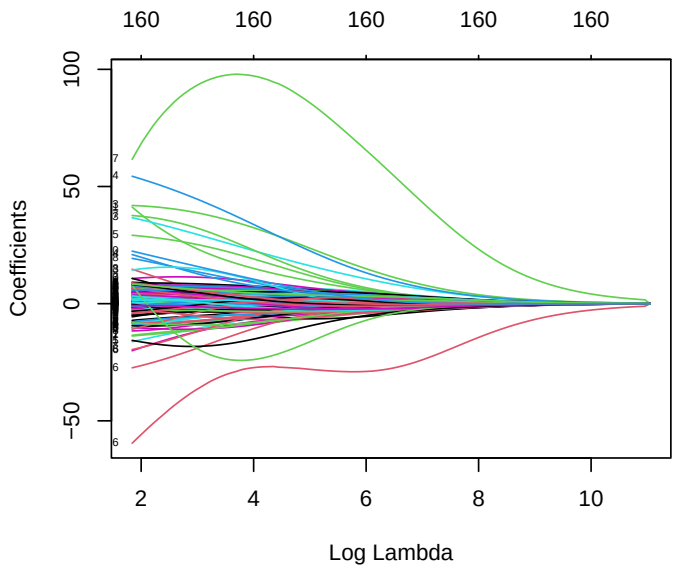
```
# Error metrics for each method
las$err
```

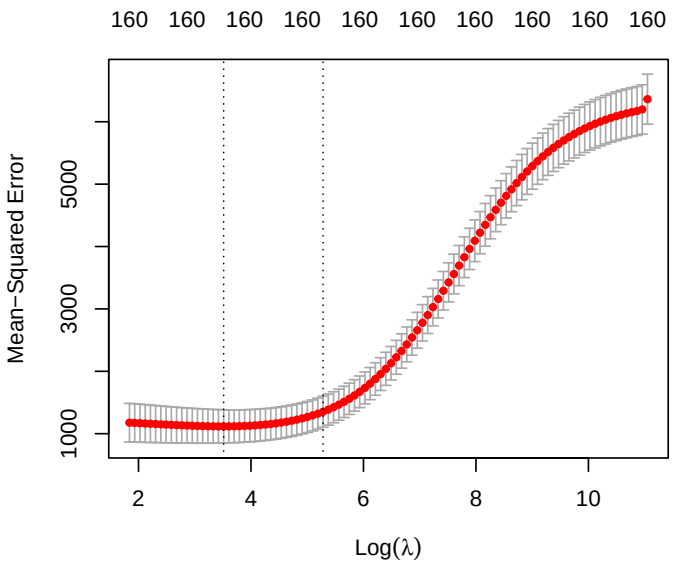
```
##        MAE      MSE  MAE_1SE  MSE_1SE
## 1 17.97478 900.6332 22.71371 1311.192
```

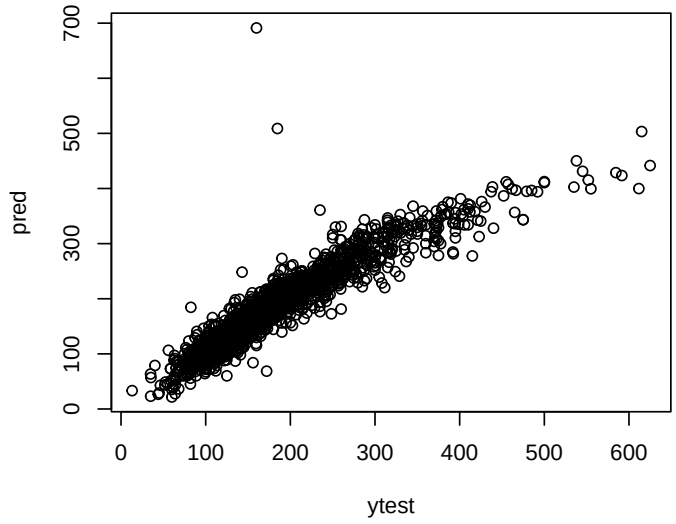
- We see the 1 SE rule selects 19 variables, while the lasso retains 65 variables.
- However, the MAE and the MSE are bigger (more explainable model but less predictive power).

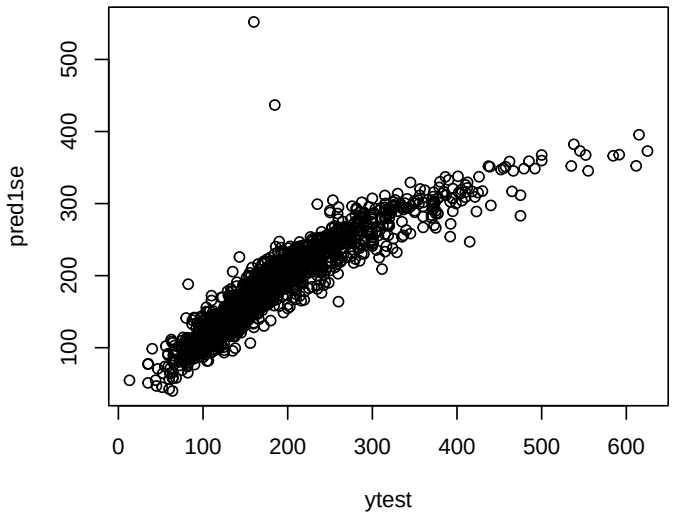
Ridge

```
# basic ridge
rid=wrapglmnet(xdumtrain,
               amesdumtrain$Sale_Price,
               xdumtest,
               amesdumtest$Sale_Price,
               0) # alpha glmnet: ridge
```










```
dim(rid$coef1se)
```

```
## [1] 161   1
```

```
dim(rid$coef)
```

```
## [1] 161   1
```

```
rid$err
```

```
##        MAE      MSE  MAE_1SE  MSE_1SE
## 1 17.83458 889.6937 20.49653 1165.339
```

```
las$err
```

```
##        MAE      MSE  MAE_1SE  MSE_1SE
## 1 17.97478 900.6332 22.71371 1311.192
```

- RIdge does not perform variable selection.
- The MSE of Ridge is very close to the one of lasso

OLS Regression

```
# Ordinary OLS regression

# Fit an Ordinary Least Squares (OLS) regression model using all covariates
lmfit = lm(Sale_Price ~ ., data = amesdumtrain)

# Make predictions on the test data using the OLS model
predlmfit = predict(lmfit, newdata = amesdumtest)
```

```
## Warning in predict.lm(lmfit, newdata = amesdumtest): prediction from
## rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```
# Compute Mean Absolute Error (MAE) and Mean Squared Error (MSE) for the OLS model
errlmfit = data.frame(
  mean(abs(predlmfit - amesdumtest$Sale_Price)),
  mean((predlmfit - amesdumtest$Sale_Price)^2)
)

# Rename the columns in the error data frame
names(errlmfit) = c("MAE", "MSE")

# Display the MAE and MSE for the OLS model
errlmfit
```

```
##        MAE      MSE
## 1 19.64333 1278.402
```

We see that the OLS does not do as well as lasso and ridge.

Refitting the Lasso

We could fit an ordinary OLS to the variables selected by the lasso, or run the lasso again on them.

Lasso + OLS

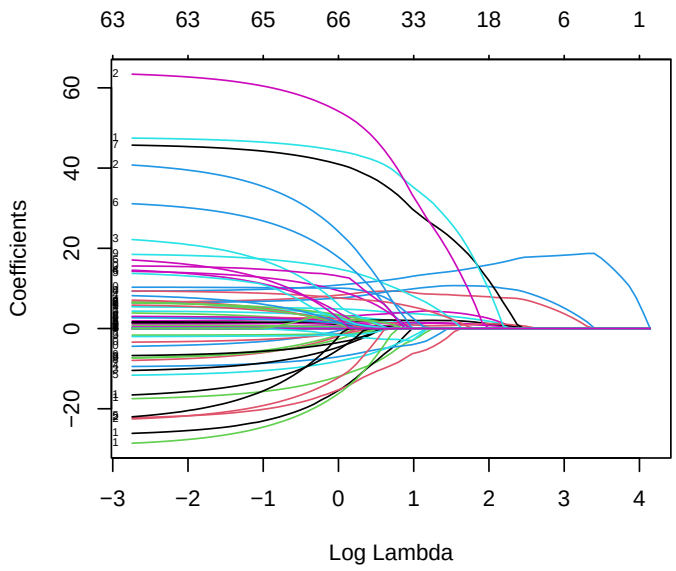
```
# Fit an OLS model using only the lasso-selected variables

# Extract the names of lasso-selected variables (excluding intercept)
namlas = rownames(las$coef)[-1]

# Fit an Ordinary Least Squares (OLS) model using the lasso-selected variables and Sale_Price
laslm = lm(Sale_Price ~ ., data = amesdumtrain[, c(namlas, "Sale_Price")])

# Make predictions on the test data using the OLS model with lasso-selected variables
predlaslm = predict(laslm, newdata = amesdumtest)

# Compute Mean Absolute Error (MAE) and Mean Squared Error (MSE) for the OLS model with lasso-selected variable
errlaslm = data.frame(
  mean(abs(predlaslm - amesdumtest$Sale_Price)),
  mean((predlaslm - amesdumtest$Sale_Price)^2)
```

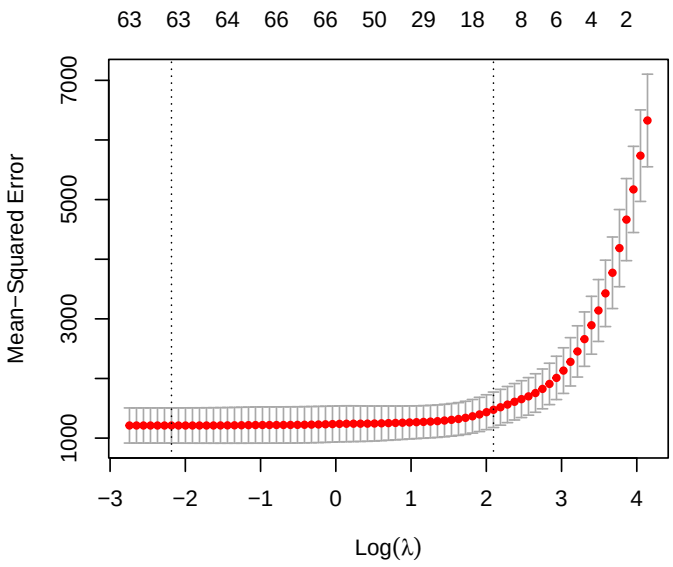
```
)

# Rename the columns in the error data frame
names(errlaslm) = c("MAE", "MSE")

# Display the MAE and MSE for the OLS model with lasso-selected variables
errlaslm
```

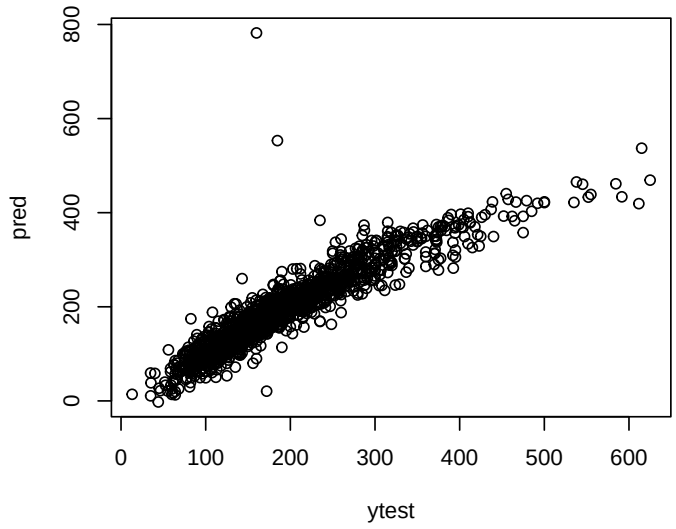
```
##       MAE      MSE
## 1 18.4807 960.3534
```

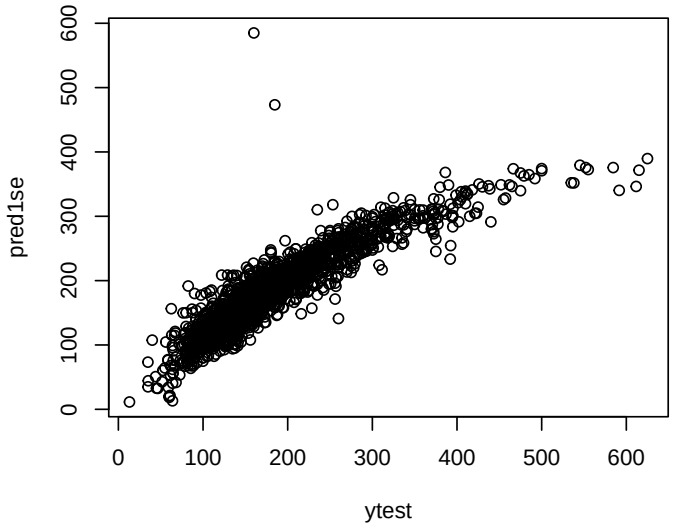
Lasso + Lasso

```
# Apply the lasso regression again to the lasso-selected variables only

# Use the 'wrapglmnet' function to apply lasso regression
laslas = wrapglmnet(
  xdumtrain[, namlas],
  amesdumtrain$Sale_Price,
  xdumtest[, namlas],
  amesdumtest$Sale_Price,
  1 # alpha parameter
)
```










```
# Check if all variables are kept in the lasso-lasso solution
rownames(laslas$coef) == rownames(las$coef)
```

```
## Warning in rownames(laslas$coef) == rownames(las$coef): longer object length is
## not a multiple of shorter object length
```

```
##  [1]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
# Calculate the proportion of parameters greater in magnitude in the lasso-lasso solution compared to the lasso
# mean(abs(laslas$coef) > abs(las$coef))

# Get the variable names from both 'las' and 'laslas' models
common_vars <- intersect(rownames(las$coef), rownames(laslas$coef))
```

```
# Calculate the proportion of parameters with greater magnitude in 'laslas' compared to 'las'
prop_greater_magnitude <- mean(abs(laslas$coef[common_vars, ]) > abs(las$coef[common_vars, ]))

# Display the proportion
prop_greater_magnitude
```

```
## [1] 0.859375
```

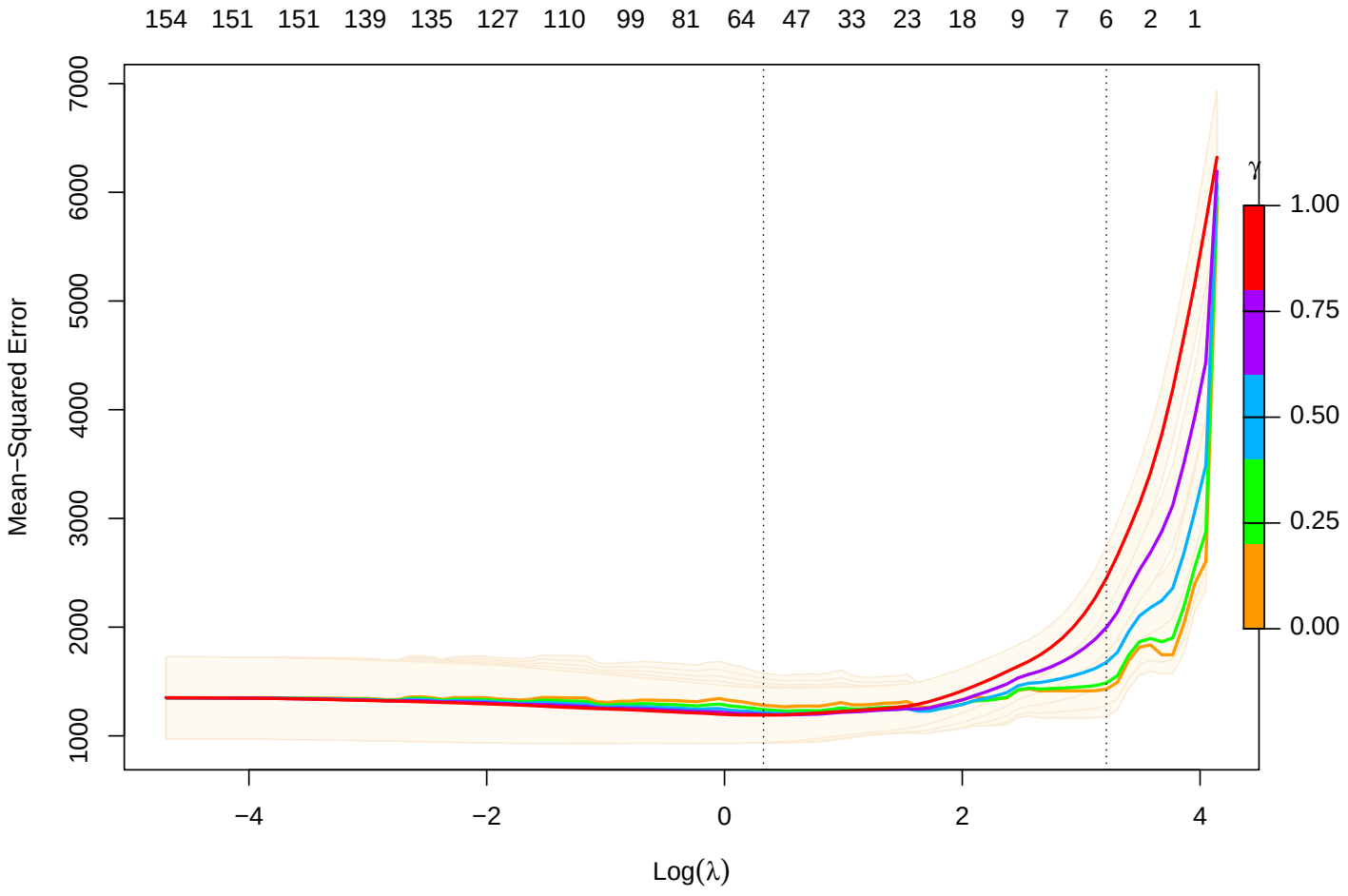
```
# Display the error metrics from the lasso-lasso solution
laslas$er
```

```
##        MAE      MSE  MAE_1SE  MSE_1SE
## 1 18.33727 946.9394 22.23764 1263.399
```

Relaxed Lasso

We could use the relaxed lasso using the option `relax=TRUE`. Here, an extra tuning parameter (noted γ) must be estimated.

```
# The right code for the relax lasso
library("glmnet")

# Perform cross-validation with relax lasso using cv.glmnet
# cv.relax = cv.glmnet(x = xtrain, y = ytrain, alpha = 1, relax = TRUE)
cv.relax = cv.glmnet(x = xdumtrain, y = amesdumtrain$Sale_Price, alpha = 1, relax = TRUE)

# Plot the cross-validation results for relax lasso
plot(cv.relax)
```



```r
# Predictions with optimal lambda and gamma for the relax lasso

# Make predictions using lambda.min and gamma.min for relax lasso
pred = predict(cv.relax, new = xdumtest, s = "lambda.min", gamma = "gamma.min")
pred1se = predict(cv.relax, new = xdumtest, s = "lambda.1se", gamma = "gamma.1se")

# Compute error metrics for the relax lasso predictions
errrla = data.frame(
  MAE = mean(abs(pred - amesdumtest$Sale_Price)),
  MSE = mean((pred - amesdumtest$Sale_Price)^2),
  MAE_1SE = mean(abs(pred1se - amesdumtest$Sale_Price)),
  MSE_1SE = mean((pred1se - amesdumtest$Sale_Price)^2)
)

# Display the error metrics for the relax lasso predictions
errrla
```

```
##        MAE      MSE  MAE_1SE MSE_1SE
## 1 18.22149 907.1532 23.57203 1261.42
```

Finally, we could perform relaxed elastic net, but we need to fix the value of α, which is not optimized in glmnet using CV.

```r
# Putting all the results together and sorting them according to the MAE and MSE

# Combine error metrics from different models
allres = rbind(
  las$err[, 1:2],          # Lasso regression
  rid$err[, 1:2],          # Ridge regression
  errlmfit,                # Ordinary Least Squares (OLS)
  errlaslm,                # Lasso regression with selected variables
  laslas$err[, 1:2],       # Lasso-lasso regression
  errrla[, 1:2]            # Relaxed Lasso regression
)

# Assign row names to the combined results
row.names(allres) = c("lasso", "ridge", "OLS", "lasso-OLS", "lasso-lasso", "relaxed lasso")

# Sort the results by MAE
sorted_by_MAE = allres[order(allres[, 1]), ]

# Sort the results by MSE
sorted_by_MSE = allres[order(allres[, 2]), ]

# Display the sorted results by MAE and MSE
sorted_by_MAE
```

```
##                    MAE       MSE
## ridge          17.83458  889.6937
## lasso          17.97478  900.6332
## relaxed lasso  18.22149  907.1532
## lasso-lasso    18.33727  946.9394
## lasso-OLS      18.48070  960.3534
## OLS            19.64333 1278.4017
```

```r
sorted_by_MSE
```

```
##                    MAE       MSE
## ridge          17.83458  889.6937
```

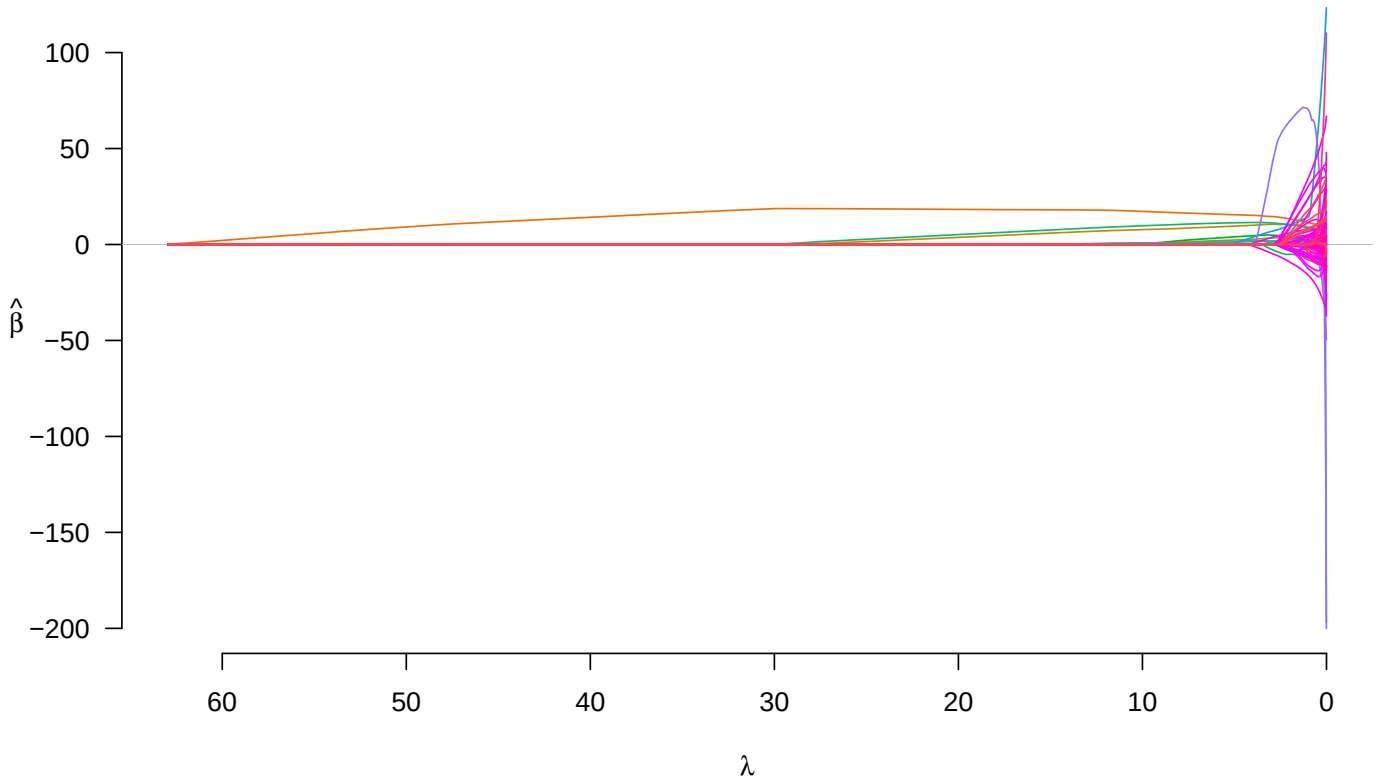
```
## lasso         17.97478  900.6332
## relaxed lasso 18.22149  907.1532
## lasso-lasso   18.33727  946.9394
## lasso-OLS     18.48070  960.3534
## OLS           19.64333 1278.4017
```

We see that the ridge did the best according to MAE and MSE.

Group Lasso and Exponential Lasso

Group Lasso

```
# Load the 'grpreg' library
library(grpreg)

# Treating the dummies of a categorical variable as a group

# Define the 'group' vector for grpreg
group <- c(1:57, rep(58, 11), rep(59, 3), 60, rep(61, 2), rep(62, 4), rep(63, 21), rep(64, 5),
           65, rep(66, 4), rep(67, 5), rep(68, 2), 69, rep(70, 10), rep(71, 10),
           rep(72, 3), rep(73, 4), 74, 75, rep(76, 5), rep(77, 2), rep(78, 3), rep(79, 4))

# Set a random seed for reproducibility
set.seed(14273)

# Fit a group lasso regression model
grlassofit <- grpreg(xdumtrain, amesdumtrain$Sale_Price, group, penalty = "grLasso")

# Plot the results of the group lasso regression
plot(grlassofit)
```




```
# Group lasso with cross-validation
grlassofitcv <- cv.grpreg(xdumtrain, amesdumtrain$Sale_Price, group, seed = 474659, penalty = "grLasso")

# Predict coefficients and test data
coefgrlasso <- predict(grlassofitcv, type = "coefficients")
predgrlasso <- predict(grlassofitcv, X = xdumtest)

# Calculate MAE and MSE
errgrlasso <- data.frame(
  MAE = mean(abs(predgrlasso - amesdumtest$Sale_Price)),
  MSE = mean((predgrlasso - amesdumtest$Sale_Price)^2)
)
names(errgrlasso) <- c("MAE", "MSE")
row.names(errgrlasso) <- c("group lasso")

# Display the error metrics
errgrlasso
```

```
##                  MAE      MSE
## group lasso 17.90152 1007.482
```

```
# Append the results to the 'allres' dataframe
allres <- rbind(allres, errgrlasso)
```

Exponential Lasso

```r
# random seed for replication
set.seed(73888)

# Exponential lasso with cross-validation
gelcv <- cv.grpreg(xdumtrain, amesdumtrain$Sale_Price, group, seed = 474659, penalty = "gel")

# Predict coefficients and test data
coefgel <- predict(gelcv, type = "coefficients")
predgel <- predict(gelcv, X = xdumtest)

# Calculate MAE and MSE
errgel <- data.frame(
  MAE = mean(abs(predgel - amesdumtest$Sale_Price)),
  MSE = mean((predgel - amesdumtest$Sale_Price)^2)
)
names(errgel) <- c("MAE", "MSE")
row.names(errgel) <- c("exponential lasso")

# Display the error metrics
errgel
```

```
##                       MAE      MSE
## exponential lasso 18.9298 1246.616
```

```r
# Append the results to the 'allres' dataframe
allres <- rbind(allres, errgel)
```

```r
# Best methods based on MAE and MSE
allres[order(allres[, 1]), ]
```

```
##                       MAE       MSE
## ridge             17.83458  889.6937
## group lasso       17.90152 1007.4815
## lasso             17.97478  900.6332
## relaxed lasso     18.22149  907.1532
## lasso-lasso       18.33727  946.9394
## lasso-OLS         18.48070  960.3534
## exponential lasso 18.92980 1246.6156
## OLS               19.64333 1278.4017
```

```r
allres[order(allres[, 2]), ]
```

```
##                       MAE       MSE
## ridge             17.83458  889.6937
## lasso             17.97478  900.6332
## relaxed lasso     18.22149  907.1532
## lasso-lasso       18.33727  946.9394
## lasso-OLS         18.48070  960.3534
## group lasso       17.90152 1007.4815
## exponential lasso 18.92980 1246.6156
## OLS               19.64333 1278.4017
```

Note: For predictive purposes, these methods do not have generally disadvantage over the ones that treat the dummies as individual variables.