GLM Model

Group 6

```
devtools::install_github(c('rstudio/rmarkdown', 'yihui/tinytex'))
tinytex::install_tinytex()
```

#Model used: GLM Generalized linear model (GLM) is a flexible generalization of ordinary linear regression. The GLM generalizes linear regression by allowing the linear model to be related to the response variable via a link function and by allowing the magnitude of the variance of each measurement to be a function of its predicted value.

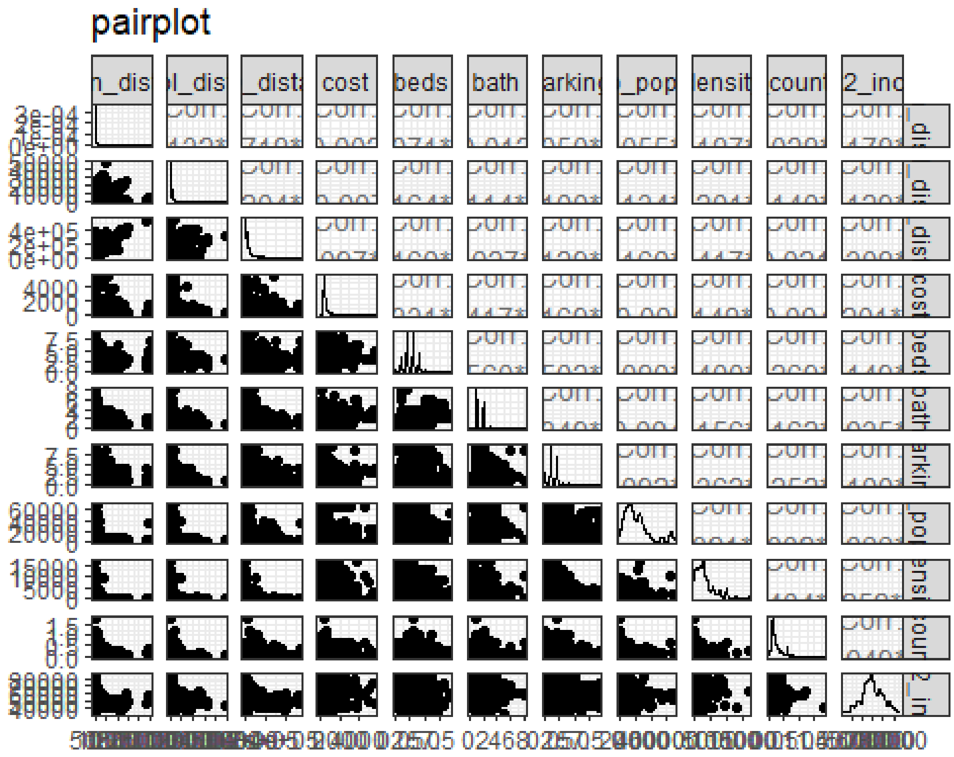
```
data <- read.csv(file = "property_final.csv",header = TRUE)
data$cloest.school<-factor(data$cloest.school)
data$type <- factor(data$type)
data$cloest.station <- factor(data$cloest.station)
str(data)
```

```
## 'data.frame':    14725 obs. of  19 variables:
##  $ X                   : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ type                : Factor w/ 14 levels "Acreage / Semi-Rural",..: 2
2 2 2 2 2 2 2 2 2 ...
##  $ cloest.station      : Factor w/ 306 levels "0","1","2","3",..: 29 171
27 136 27 29 171 92 171 171 ...
##  $ station_distance    : num  1804 372 248 404 525 ...
##  $ cloest.school       : Factor w/ 1707 levels "0","1","4","5",..: 981
1686 1625 1629 1686 1686 1625 1619 1686 1625 ...
##  $ school_distance     : num  667 352 402 587 780 ...
##  $ CBD_distance        : num  1702 2267 1937 790 1658 ...
##  $ postcode            : int  3000 3000 3000 3000 3000 3000 3000 3000 3000
3000 ...
##  $ address             : chr  "1901/368 St Kilda Road Melbourne" "1211/200
Spencer Street Melbourne" "1008/380 Little Lonsdale Street Melbourne" "3/27
Flinders Lane Melbourne" ...
##  $ cost                : num  1800 480 400 420 350 600 600 550 440 650 ...
##  $ beds                : int  3 1 1 1 1 2 2 2 1 2 ...
##  $ bath                : int  2 1 1 1 1 1 2 1 1 2 ...
##  $ parking             : int  2 1 0 0 0 1 1 0 1 1 ...
##  $ LOC_PID             : chr  "loc9901d119afda" "loc9901d119afda"
"loc9901d119afda" "loc9901d119afda" ...
##  $ LOC_NAME            : chr  "Melbourne" "Melbourne" "Melbourne" "Mel-
bourne" ...
##  $ suburb_population   : num  64538 64538 64538 64538 64538 ...
##  $ density             : num  9903 9903 9903 9903 9903 ...
##  $ offence_count_scaled: num  0.615 0.615 0.615 0.615 0.615 ...
##  $ X2022_income        : num  59708 59708 59708 59708 59708 ...
```

```
# pair plot for numerical features
library(GGally)

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

ggpairs(data, columns=c(4,6,7,10,11,12,13,16,17,18,19))+ggtitle("pairplot")+theme_bw()
```



Assumption:

1.Linear relationship:According to the pair plot above, it can be seen that there is a potential linear relationship between cost and other features. 2.Independence:No clear pattern can be seen in Scatter plot between different features, and all these data tend to be approximately independent. 3. Normality: Cost itself is in accordance with normal distribution, and ‘cost vs cost’ can be seen in the figure above

```
m1 <- glm(
cost ~
type+station_distance+CBD_distance+beds+bath+parking+suburb_population+densit
y+offence_count_scaled+X2022_income,
data=data,
family = gaussian(link = "identity")
)
summary(m1)
```

```
##
## Call:
## glm(formula = cost ~ type + station_distance + CBD_distance + 
##     beds + bath + parking + suburb_population + density + of-
fence_count_scaled + 
##     X2022_income, family = gaussian(link = "identity"), data = data)
##
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1311.5   -101.0    -24.5     54.3   4818.5  
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       -9.351e+01  6.503e+01  -1.438  0.15050
## typeApartment / Unit / Flat       -1.878e+02  6.310e+01  -2.977  0.00291
**
## typeCarspace                      -4.676e+02  1.122e+02  -4.167 3.11e-05
***
## typeDuplex                        -2.532e+02  1.118e+02  -2.265  0.02353 *
## typeHouse                         -1.976e+02  6.290e+01  -3.141  0.00169
**
## typeNew Apartments / Off the Plan -8.883e+01  1.193e+02  -0.744  0.45661
## typeNew House & Land              -3.355e+02  2.350e+02  -1.428  0.15340
## typePenthouse                     -4.418e+01  2.350e+02  -0.188  0.85088
## typeRural                          1.443e+02  2.350e+02   0.614  0.53934
## typeSemi-Detached                 -1.118e+02  9.290e+01  -1.203  0.22888
## typeStudio                        -1.986e+02  6.559e+01  -3.028  0.00247
**
## typeTerrace                       -4.101e+01  1.019e+02  -0.402  0.68745
## typeTownhouse                     -1.719e+02  6.313e+01  -2.724  0.00646
**
## typeVilla                         -8.236e+01  7.286e+01  -1.130  0.25835
## station_distance                   1.503e-03  1.299e-04  11.569  < 2e-16
***
## CBD_distance                      -3.398e-04  4.636e-05  -7.329 2.44e-13
***
## beds                               7.925e+01  2.892e+00  27.398  < 2e-16
***
## bath                               1.307e+02  3.863e+00  33.847  < 2e-16
***
## parking                            1.468e+01  2.355e+00   6.234 4.67e-10
***
## suburb_population                 -1.391e-03  1.304e-04 -10.666  < 2e-16
***
## density                            2.297e-02  8.211e-04  27.977  < 2e-16
***
## offence_count_scaled               1.408e+02  1.510e+01   9.323  < 2e-16
***
## X2022_income                       5.153e-03  2.073e-04  24.853  < 2e-16
***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 51264.67)
##
##     Null deviance: 1103958170  on 14724  degrees of freedom
## Residual deviance:  753693211  on 14702  degrees of freedom
## AIC: 201502
##
## Number of Fisher Scoring iterations: 2
```

```
library("DALEX")
```

```
## Warning: 程辑包'DALEX'是用R版本4.1.3 来建造的
```

```
## Welcome to DALEX (version: 2.4.2).
## Find examples and detailed introduction at: http://ema.drwhy.ai/
## Additional features will be available after installation of: ggpubr.
## Use 'install_dependencies()' to get all suggested dependencies
```

```
library("ingredients")
```

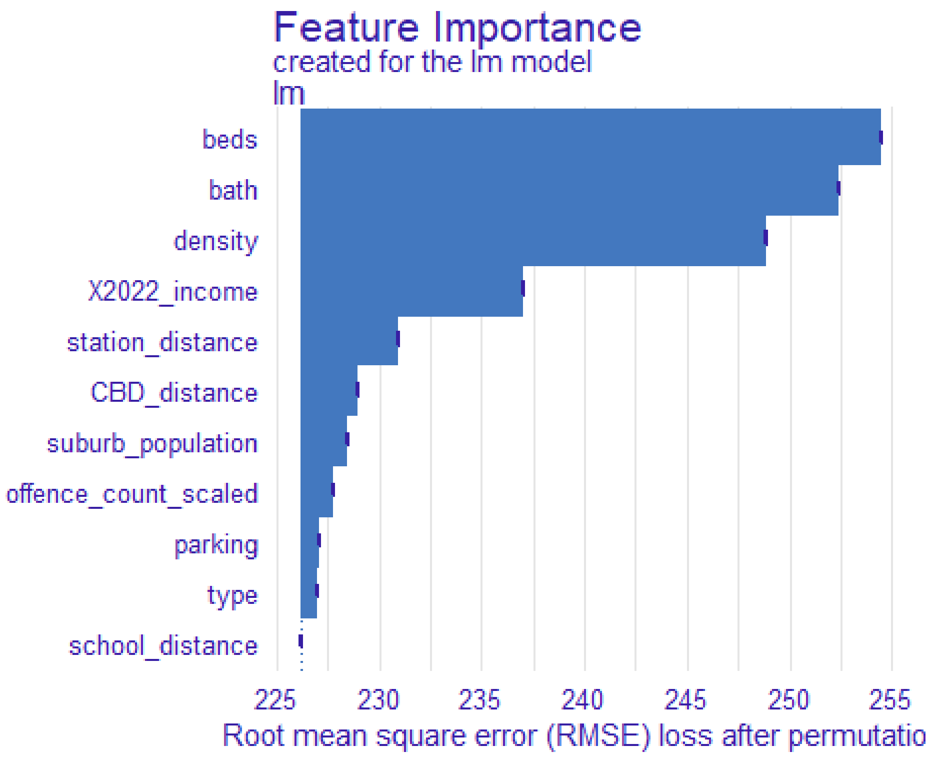
```
## The following object is masked from 'package:DALEX':
##
##     feature_importance
```

```
explain_titanic_glm <- explain(m1,data=data[,-c(1,3,5,8,9,10,14,15)],y = data[,10])
```

```
## Preparation of a new explainer is initiated
##   -> model label       :  lm  (  default  )
##   -> data              :  14725  rows  11  cols
##   -> target variable   :  14725  values
##   -> predict function  :  yhat.glm  will be used (  default  )
##   -> predicted values  :  No value for predict function target column.
(  default  )
##   -> model_info        :  package stats , ver. 4.1.2 , task regression
(  default  )
##   -> predicted values  :  numerical, min =  -30.166 , mean =  514.5317 ,
max =  1491.142
##   -> residual function :  difference between y and yhat (  default  )
##   -> residuals         :  numerical, min =  -1311.507 , mean =  -
5.511811e-11 , max =  4818.523
##   A new explainer has been created!
```

```
fig<- feature_importance(explain_titanic_glm, B = 1)

plot(fig)
```



```
library(glmnet)

## Loaded glmnet 4.1-4

m2 <- glmnet(data[,-c(1,2,3,5,8,9,10,22,12,13,14,15)], y = data[,10],alpha =
1, family = 'gaussian')
summary(m2)

##           Length Class     Mode
## a0         66    -none-    numeric
## beta      528    dgCMatrix S4
## df         66    -none-    numeric
## dim         2    -none-    numeric
## lambda     66    -none-    numeric
## dev.ratio  66    -none-    numeric
## nulldev     1    -none-    numeric
## npasses     1    -none-    numeric
## jerr        1    -none-    numeric
## offset      1    -none-    logical
## call        5    -none-    call
## nobs        1    -none-    numeric
```

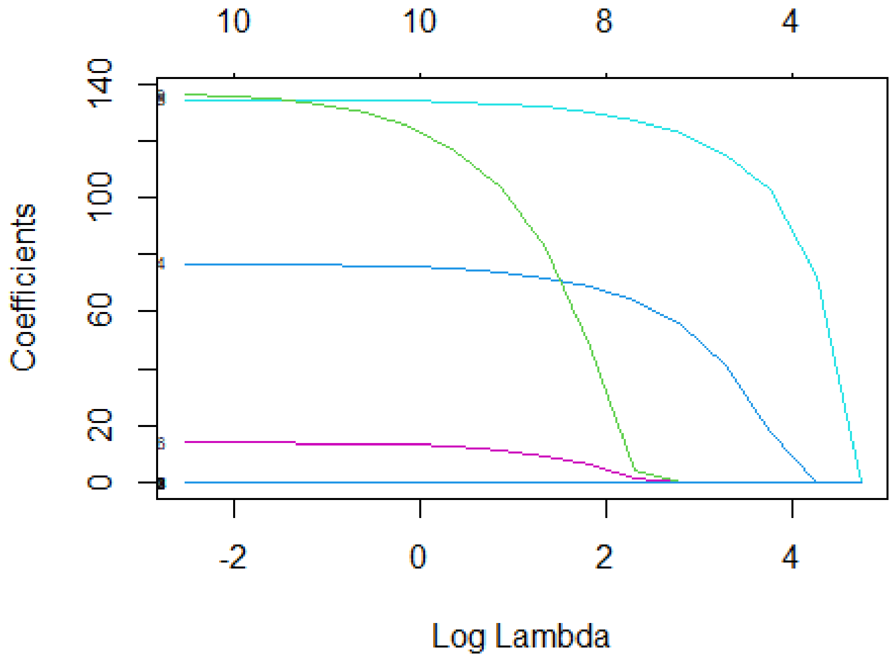
regularization

In the glmnet package, Lasso or Ridge is considered for the regression model of multi-dimensional features. However, the parameter β of ridge will not be equal to 0, but the parameter β of Lasso can be equal to 0, so Lasso is chosen.

```
x <- as.matrix(data[,-c(1,2,3,5,8,9,10,14,15)])
y <- as.matrix(data[,10])
f1 = glmnet(x, y, family="gaussian", nlambda=20, alpha=1)
print(f1)
```

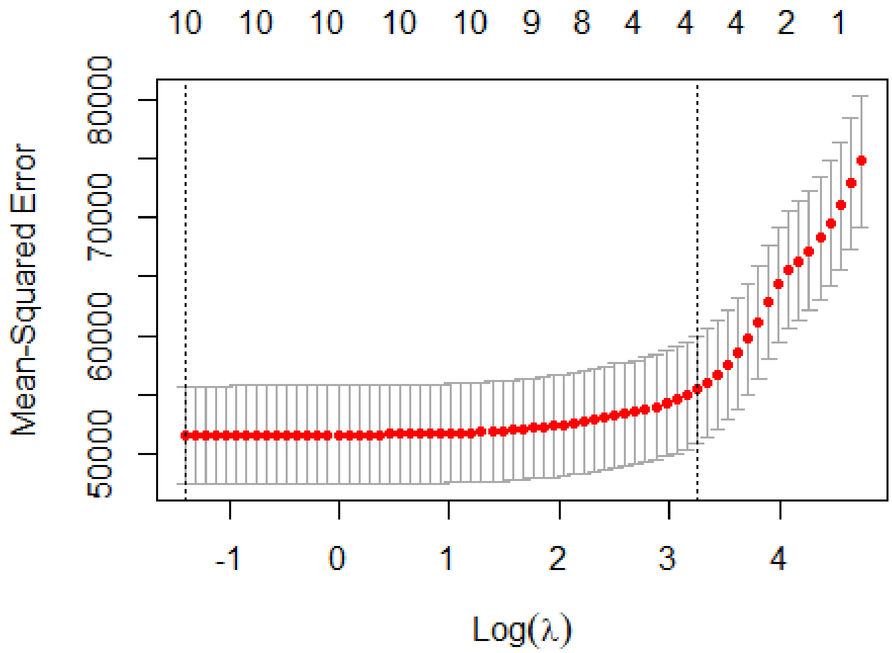
```
##
## Call:  glmnet(x = x, y = y, family = "gaussian", alpha = 1, nlambda = 20)
##
##    Df  %Dev  Lambda
## 1   0  0.00 114.100
## 2   1 10.78  70.280
## 3   4 19.55  43.280
## 4   4 26.07  26.660
## 5   4 28.54  16.420
## 6   8 29.75  10.110
## 7   8 30.65   6.226
## 8  10 31.17   3.834
## 9  10 31.38   2.361
## 10 10 31.46   1.454
## 11 10 31.49   0.896
## 12 10 31.50   0.551
## 13 10 31.51   0.340
## 14 10 31.51   0.209
## 15 10 31.51   0.129
## 16 10 31.51   0.079
```

```
plot(f1, xvar="lambda", label=TRUE)
```



# N-fold Cross Validation is a built-in function of GlMNet The Y-axis is the MSE (minimum error squared), and the corresponding X-axis superscript of the MSE minimum is the number of eigenvalues selected by Lasso.

```
cvfit=cv.glmnet(x,y,family = 'gaussian')
plot(cvfit)
```



```
cvfit$lambda.min

## [1] 0.245865

cvfit$lambda.1se

## [1] 25.75721

l.coef2<-coef(cvfit$glmnet.fit,s=0.245865,exact = F)
l.coef1<-coef(cvfit$glmnet.fit,s=28.2685,exact = F)
l.coef1

## 11 x 1 sparse Matrix of class "dgCMatrix"
##                                  s1
## (Intercept)           4.497697e+01
## station_distance      .
## school_distance       .
## CBD_distance          .
## beds                  3.889305e+01
## bath                  1.138809e+02
## parking               .
## suburb_population     .
## density               1.033878e-02
## offence_count_scaled .
## X2022_income          2.751898e-03

l.coef2
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                                 s1
## (Intercept)          -2.741735e+02
## station_distance      1.484284e-03
## school_distance      -2.837428e-03
## CBD_distance         -3.349338e-04
## beds                  7.657599e+01
## bath                  1.342418e+02
## parking               1.388497e+01
## suburb_population    -1.411258e-03
## density               2.290754e-02
## offence_count_scaled  1.339111e+02
## X2022_income          5.118330e-03

m1 <- glm(
cost ~ beds+bath+density+X2022_income,
data=data,
family = gaussian(link = "identity")
)
summary(m1)

##
## Call:
## glm(formula = cost ~ beds + bath + density + X2022_income, family = gauss-
ian(link = "identity"),
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1343.2   -104.6    -24.9     52.3   4972.8
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.824e+02  1.352e+01  -20.88   <2e-16 ***
## beds          7.840e+01  2.368e+00   33.10   <2e-16 ***
## bath          1.342e+02  3.773e+00   35.57   <2e-16 ***
## density       2.318e-02  6.708e-04   34.55   <2e-16 ***
## X2022_income  5.259e-03  1.989e-04   26.44   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 52464.93)
##
##     Null deviance: 1103958170  on 14724  degrees of freedom
## Residual deviance:  772283832  on 14720  degrees of freedom
## AIC: 201825
##
## Number of Fisher Scoring iterations: 2
```