SAS/R商業資料分析作業二

107508006 歐西四 陳葳芃

1. [20pts] a. 生成一筆資料：Xi=a+ε, i=1,…,20
>>a為0~10 任意數字ε~N(0,2) >>注意：X必須在0~11內。

Ans:

```
> #1a
> ##生成一筆資料
> data <- data.frame(
+   a = sample(1:10,20,replace=T),
+   ε = rnorm(20,mean=0,sd=2))
> ##觀察sum (用來檢查ifelse的結果)
> library(tidyverse)
> data<- data%>%
+   mutate(sum= a+ε)
> ##利用ifelse來確保X會落在0-11區間
> data$X <- ifelse( data$a + data$ε >= 11, 11,
+                       ifelse( 0 <= data$a + data$ε, data$a + data$ε ,0))
> ##所求
> X=data$X
> head(X)
[1] 7.4145659 3.9574619 2.0682391 0.8035686 9.7202959 0.0000000
>
```

1. [20pts] b. Cauchy(θ, 1) 的密度函數，取log後一次微分如下，請寫出此function

Ans:

```
> #1b
> ##Cauchy density function 取log及一階導數的function
> Lcauchy <- function(theta){
+   k = 0
+   for (i in 1:length(X)){
+     k <- k+(theta-X[i])/(1+(theta-X[i])^2)
+   }
+   return(-2*k)}
>
```

1. [20pts] c.代入a生成的資料至b的function，並令θ=0.3

Ans:

```
> #1c
> ##令theta=0.3，代入先前的X data
> Lcauchy(0.3)
[1] 8.76137
>
```

2. [10pts] a. 根據Build_year，建立一個新類別變數year_type，1899年以前的房子為”centennial”，1900~1959年為”old”，1960年以上為”new”。

Ans:

```
> setwd("~/Downloads/1102 R/HW/hw 2")
> #2
> ##載入資料
> houseprice.df <- read.csv("houseprice.csv")
> ##觀察資料並看看年份有沒有異常值
> str(houseprice.df)
'data.frame':   10659 obs. of  11 variables:
 $ Record     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Sale_amount: int  295000 240000 385000 268000 186000 302500 223000 225000 215000 285100 ...
 $ Sale_date  : chr  "2016/5/31" "2016/6/20" "2016/5/31" "2016/4/12" ...
 $ Beds       : int  5 4 5 3 3 4 3 3 5 3 ...
 $ Baths      : num  3 2 4 2.5 1.25 3 2 3 2 4 ...
 $ Sqft_home  : int  2020 1498 4000 2283 1527 3117 1218 3000 2963 1680 ...
 $ Sqft_lot   : num  38333 54014 85813 118919 15682 ...
 $ Type       : chr  "Single Family" "Single Family" "Single Family" "Single Family" ...
 $ Build_year : int  1976 2002 2001 1972 1975 1976 1975 1969 1965 1987 ...
 $ Town       : chr  "Ames, IA" "Ames, IA" "Ames, IA" "Ames, IA" ...
 $ University : chr  "Iowa State University" "Iowa State University" "Iowa State University" "Iowa State University"
 ...
```

```
> #2a
> ##方便操作
> library(tidyverse)
> houseprice.df<- mutate(houseprice.df, year_type=Build_year)
> range(houseprice.df$year_type)
[1] 1806 2016
> houseprice.df$year_type<- ifelse( year_type <= 1899 , "centennial" ,
+           ifelse( 1960 <= year_type , "new","old"))
> head(houseprice.df)
  Record Sale_amount Sale_date Beds Baths Sqft_home Sqft_lot          Type Build_year     Town
1      1      295000 2016/5/31    5  3.00      2020  38332.8 Single Family       1976 Ames, IA
2      2      240000 2016/6/20    4  2.00      1498  54014.4 Single Family       2002 Ames, IA
3      3      385000 2016/5/31    5  4.00      4000  85813.2 Single Family       2001 Ames, IA
4      4      268000 2016/4/12    3  2.50      2283 118918.8 Single Family       1972 Ames, IA
5      5      186000  2016/4/5    3  1.25      1527  15681.6 Single Family       1975 Ames, IA
6      6      302500  2016/3/2    4  3.00      3117  33105.6 Single Family       1976 Ames, IA
              University year_type
1 Iowa State University       new
2 Iowa State University       new
3 Iowa State University       new
4 Iowa State University       new
5 Iowa State University       new
6 Iowa State University       new
>
```

2. [40pts] b. 決定好你的最佳配適模型後，總結你的發現並根據解釋變數預測房屋價格。

Note:

在評估各種你建立的模型之前，你必須篩選或過濾掉某些數據，並找出變數的子集以獲得適合的分析資料。

選擇你的最佳模型時，不用一定要符合殘差檢驗，只要能正確解釋你想預測的東西即可。

Ans:

說明：我先將各個類別資料轉成factor，並觀察個factor variable有什麼level。

```
> ## Preparation to change the variable type into factor variable
> houseprice.df$Town<-as.factor(houseprice.df$Town)
> houseprice.df$University<-as.factor(houseprice.df$University)
> houseprice.df$year_type<-as.factor(houseprice.df$year_type)
> houseprice.df$Type<-as.factor(houseprice.df$Type)
> ## Find the each factor name
> levels(houseprice.df$Type)
[1] "Multi Family"       "Multiple Occupancy" "Single Family"
> levels(houseprice.df$Town)
 [1] "Ames, IA"                 "Amherst, MA"              "Ann Arbor, MI"            "Athens, GA"
 [5] "Berkeley, CA"             "Binghamton, NY"           "Blacksburg, VA"           "Bloomington, IL"
 [9] "Bloomington, IN"          "Boulder, CO"              "Bozeman, MT"              "Burlington, VT"
[13] "Cambridge, MA"            "Champaign-Urbana, IL"     "Chapel Hill, NC"          "Charlottesville, VA"
[17] "Claremont, CA"            "College Station, TX"      "Columbia, MO"             "Corvallis, OR"
[21] "East Lansing, MI"         "Eugene, OR"               "Fargo, ND"                "Fayetteville, AR"
[25] "Flagstaff, AZ"            "Fort Collins, CO"         "Gainesville, FL"          "Grand Forks, ND"
[29] "Hartford, CT"             "Iowa City, IA"            "Ithaca, NY"               "Lafayette, IN"
[33] "Lawrence, KS"             "Lexington, KY"            "Lincoln, NE"              "Logan, UT"
[37] "Madison, WI"              "Manhattan, KS"            "Minneapolis, MN"          "Morgantown, WV"
[41] "Oxford, MS"               "Pittsburgh, PA"           "San Luis Obispo, CA"      "State College, PA"
[45] "Syracuse, NY"             "Tacoma, WA"               "Tallahassee, FL"          "Tempe, AZ"
[49] "Tuscaloosa, AL"           "Waterloo-Cedar Falls, IA"
```

```
> levels(houseprice.df$University)
 [1] "Arizona State university"
 [2] "Bringhamton University"
 [3] "California Polytechnic State University San Luis Obispo"
 [4] "Colorado State University"
 [5] "Cornell University"
 [6] "Florida State University"
 [7] "Harvard University"
 [8] "Illinois State university"
 [9] "Indiana University Bloomington"
[10] "Iowa State University"
[11] "Kansas State University"
[12] "Michigan State University"
[13] "Montana State university"
[14] "North Dakota State University"
[15] "Northern Arizona University"
[16] "Oregon State university"
[17] "Pennsylvania State University"
[18] "Pomona College"
[19] "Purdue University"
[20] "Syracuse University"
[21] "Texas A&M University"
[22] "University Kentucky"
[23] "University of Alabama"
[24] "University of Arkansas"
[25] "University of California Berkeley"
[26] "University of Colorado Boulder"
[27] "University of Florida"
[28] "University of Georgia"
[29] "University of Hartford"
[30] "University of Illinois at Urbana-Champaign"
[31] "University of Iowa"
[32] "University of Kansas"
[33] "University of Massachusetts Amherst"
[34] "University of Michigan"
[35] "University of Minnesota"
[36] "University of Mississippi"
```

```
[37] "University of Missouri"
[38] "University of Nebraska Lincoln"
[39] "University of North Carolina at Chapel Hill"
[40] "University of North Dakota"
[41] "University of Northern Iowa"
[42] "University of Oregon"
[43] "University of Pittsburgh"
[44] "University of Vermont"
[45] "University of Virginia"
[46] "University of Washington Tacoma"
[47] "University of Wisconsin Madison"
[48] "Utah State University"
[49] "Virginia Tech"
[50] "West Virginia University"
> levels(houseprice.df$year_type)
[1] "centennial" "new"        "old"
>
```

>>說明：我將town這個類別變數取縮寫方便資料分析，並篩選出我需要的變數（過濾出：date, build year, TownF and NA column四個variables），再將我心縮寫TownS(城市縮寫)設立為factor variable。

```
> ## Classify the area into short name
> library(dplyr)
> houseprice.df<-houseprice.df %>% separate(Town, c("TownF", "TownS",sep=","))
警告訊息：
1: Expected 3 pieces. Additional pieces discarded in 372 rows [8712, 8713, 8714, 8715, 8716, 8717, 8718, 8719, 8720, 87
21, 8722, 8723, 8724, 8725, 8726, 8727, 8728, 8729, 8730, 8731, ...].
2: Expected 3 pieces. Missing pieces filled with `NA` in 8613 rows [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
 16, 17, 18, 19, 20, ...].
> ## check the data
> str(houseprice.df)
'data.frame':   10659 obs. of  14 variables:
 $ Record     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Sale_amount: int  295000 240000 385000 268000 186000 302500 223000 225000 215000 285100 ...
 $ Sale_date  : chr  "2016/5/31" "2016/6/20" "2016/5/31" "2016/4/12" ...
 $ Beds       : int  5 4 5 3 3 4 3 3 5 3 ...
 $ Baths      : num  3 2 4 2.5 1.25 3 2 3 2 4 ...
 $ Sqft_home  : int  2020 1498 4000 2283 1527 3117 1218 3000 2963 1680 ...
 $ Sqft_lot   : num  38333 54014 85813 118919 15682 ...
 $ Type       : Factor w/ 3 levels "Multi Family",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ Build_year : int  1976 2002 2001 1972 1975 1976 1975 1969 1965 1987 ...
 $ TownF      : chr  "Ames" "Ames" "Ames" "Ames" ...
 $ TownS      : chr  "IA" "IA" "IA" "IA" ...
 $ ,          : chr  NA NA NA NA ...
 $ University : Factor w/ 50 levels "Arizona State university",..: 10 10 10 10 10 10 10 10 10 10 ...
 $ year_type  : Factor w/ 3 levels "centennial","new",..: 2 2 2 2 2 2 2 2 2 2 ...
```

```
> ## Delete date, build year, TownF and NA column
> df<-houseprice.df[,c(-3,-9,-10,-12)]
> str(df)
'data.frame':   10659 obs. of  10 variables:
 $ Record     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Sale_amount: int  295000 240000 385000 268000 186000 302500 223000 225000 215000 285100 ...
 $ Beds       : int  5 4 5 3 3 4 3 3 5 3 ...
 $ Baths      : num  3 2 4 2.5 1.25 3 2 3 2 4 ...
 $ Sqft_home  : int  2020 1498 4000 2283 1527 3117 1218 3000 2963 1680 ...
 $ Sqft_lot   : num  38333 54014 85813 118919 15682 ...
 $ Type       : Factor w/ 3 levels "Multi Family",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ TownS      : chr  "IA" "IA" "IA" "IA" ...
 $ University : Factor w/ 50 levels "Arizona State university",..: 10 10 10 10 10 10 10 10 10 10 ...
 $ year_type  : Factor w/ 3 levels "centennial","new",..: 2 2 2 2 2 2 2 2 2 2 ...
> df$TownS<-as.factor(df$TownS)
> levels(df$TownS)
 [1] "AL"      "AR"      "Arbor"   "AZ"      "CA"      "Cedar"   "City"    "CO"      "College" "Collins"
[11] "CT"      "FL"      "Forks"   "GA"      "Hill"    "IA"      "IL"      "IN"      "KS"      "KY"
[21] "Lansing" "Luis"    "MA"      "MN"      "MO"      "MS"      "MT"      "ND"      "NE"      "NY"
[31] "OR"      "PA"      "Station" "Urbana"  "UT"      "VA"      "VT"      "WA"      "WI"      "WV"
```

>>說明：接下來將類別型變數轉換為dummy variable，並過濾出各dummy variables的其中一項避免共線性（#SingleFamily #AL #University ofWest Virginia #yeartype-old）。

```
> ## Use the dummy variable to predict factor variable
> library(dummies)
> dummies=dummy.data.frame(df)
警告訊息：
1: 於 model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE)：
  non-list contrasts argument ignored
2: 於 model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE)：
  non-list contrasts argument ignored
3: 於 model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE)：
  non-list contrasts argument ignored
4: 於 model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE)：
  non-list contrasts argument ignored
> str(dummies)
'data.frame':   10659 obs. of  102 variables:
 $ Record                                                : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Sale_amount                                           : int  295000 240000 385000 268000 186000 302500 22
3000 225000 215000 285100 ...
 $ Beds                                                  : int  5 4 5 3 3 4 3 3 5 3 ...
 $ Baths                                                 : num  3 2 4 2.5 1.25 3 2 3 2 4 ...
 $ Sqft_home                                             : int  2020 1498 4000 2283 1527 3117 1218 3000 2963
 1680 ...
 $ Sqft_lot                                              : num  38333 54014 85813 118919 15682 ...
 $ TypeMulti Family                                      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ TypeMultiple Occupancy                                : int  0 0 0 0 0 0 0 0 0 0 ...
 $ TypeSingle Family                                     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ TownSAL                                               : int  0 0 0 0 0 0 0 0 0 0 ...
 $ TownSAR                                               : int  0 0 0 0 0 0 0 0 0 0 ...
 $ TownSArbor                                            : int  0 0 0 0 0 0 0 0 0 0 ...
 $ TownSAZ                                               : int  0 0 0 0 0 0 0 0 0 0 ...
 $ TownSCA                                               : int  0 0 0 0 0 0 0 0 0 0 ...
 $ TownSCedar                                            : int  0 0 0 0 0 0 0 0 0 0 ...
 $ TownSCity                                             : int  0 0 0 0 0 0 0 0 0 0 ...
 $ TownSCO                                               : int  0 0 0 0 0 0 0 0 0 0 ...
 $ TownSCollege                                          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ TownSCollins                                          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ TownSCT                                               : int  0 0 0 0 0 0 0 0 0 0 ...
 $ TownSFL                                               : int  0 0 0 0 0 0 0 0 0 0 ...
```

>>說明：接著將data(df1)切成測試集及訓練集。

```
> ##Delete the one dummy of different factor
> #SingleFamily
> #AL
> #University ofWest Virginia
> #yeartype-old
> dummies1.2<-dummies[,c(-9,-10,-99,-102)]
> ## SPLIT THE DATA INTO TRAINING AND TESTING
> train_df1 <- dummies1.2 %>% sample_frac(0.7)
> test_df1  <- anti_join(dummies1.2, train_df1, by = 'Record')
> which(is.na(train_df1))
integer(0)
> train_df1 <-train_df1[,-1]
> test_df1  <-test_df1[,-1]
```

>>說明：配飾線性回歸模型。（共用以下這些 97 個 predictor:
[1] "Baths" [2] "Beds" [3] "Sale_amount"
[4] "Sqft_home" [5] "Sqft_lot" [6] "TownSAR" ….[44]"TownSWV"
[45] "TypeMulti Family" [46] "TypeMultiple Occupancy"
[47] "UniversityArizona State university" …. [95] "UniversityVirginia Tech"
[96] "year_typecentennial" [97] "year_typenew"）

並由model1模型結果R-square我們可以觀測到模型變異佔了約72%的總變異。可能訓練地不是到非常精確，因此我決定試試不用2.a剛開始的年分3分類預測，而是直接用原始的年份來進行配適model2，看會不會比較佳。

```
> #Regression model
> m1 <- lm(Sale_amount~ ., data=train_df1)
> summary(m1)

Call:
lm(formula = Sale_amount ~ ., data = train_df1)

Residuals:
     Min       1Q   Median       3Q      Max
-1946426   -60892    -8720    40092  3407269

Coefficients: (39 not defined because of singularities)
                                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                                       -5.820e+04  1.801e+04  -3.231 0.001238 **
Beds                                               3.906e+03  2.819e+03   1.386 0.165894
Baths                                              8.253e+04  3.422e+03  24.117  < 2e-16 ***
Sqft_home                                          5.554e+01  1.742e+00  31.878  < 2e-16 ***
Sqft_lot                                           3.067e-01  2.996e-02  10.237  < 2e-16 ***
`TypeMulti Family`                                -1.355e+05  1.616e+04  -8.388  < 2e-16 ***
`TypeMultiple Occupancy`                          -1.176e+05  2.606e+04  -4.513 6.50e-06 ***
TownSAR                                           -1.490e+04  1.992e+04  -0.748 0.454606
TownSArbor                                         1.381e+05  2.049e+04   6.738 1.73e-11 ***
TownSAZ                                            1.578e+05  2.014e+04   7.832 5.46e-15 ***
TownSCA                                            9.051e+05  2.097e+04  43.156  < 2e-16 ***
TownSCedar                                        -1.862e+04  2.306e+04  -0.807 0.419477
TownSCity                                          1.789e+04  2.012e+04   0.889 0.373951
TownSCO                                            6.636e+05  2.026e+04  32.746  < 2e-16 ***
TownSCollege                                       9.096e+04  3.334e+04   2.728 0.006385 **
TownSCollins                                       1.150e+05  1.949e+04   5.899 3.82e-09 ***
TownSCT                                           -5.375e+04  2.528e+04  -2.126 0.033561 *
TownSFL                                            3.882e+04  2.101e+04   1.848 0.064681 .
TownSForks                                         2.864e+03  3.233e+04   0.089 0.929429
TownSGA                                           -1.747e+04  2.016e+04  -0.867 0.386140
TownSHill                                          1.233e+05  1.960e+04   6.290 3.36e-10 ***
TownSIA                                            1.475e+04  2.205e+04   0.669 0.503656
TownSIL                                           -8.090e+04  1.953e+04  -4.142 3.48e-05 ***
```

```
`UniversityUniversity of Alabama`                          NA          NA       NA       NA
`UniversityUniversity of Arkansas`                         NA          NA       NA       NA
`UniversityUniversity of California Berkeley`              NA          NA       NA       NA
`UniversityUniversity of Colorado Boulder`                 NA          NA       NA       NA
`UniversityUniversity of Florida`                          NA          NA       NA       NA
`UniversityUniversity of Georgia`                          NA          NA       NA       NA
`UniversityUniversity of Hartford`                         NA          NA       NA       NA
`UniversityUniversity of Illinois at Urbana-Champaign`     NA          NA       NA       NA
`UniversityUniversity of Iowa`                             NA          NA       NA       NA
`UniversityUniversity of Kansas`                           NA          NA       NA       NA
`UniversityUniversity of Massachusetts Amherst`            NA          NA       NA       NA
`UniversityUniversity of Michigan`                         NA          NA       NA       NA
`UniversityUniversity of Minnesota`                        NA          NA       NA       NA
`UniversityUniversity of Mississippi`                      NA          NA       NA       NA
`UniversityUniversity of Missouri`                         NA          NA       NA       NA
`UniversityUniversity of Nebraska Lincoln`                 NA          NA       NA       NA
`UniversityUniversity of North Carolina at Chapel Hill`    NA          NA       NA       NA
`UniversityUniversity of North Dakota`                     NA          NA       NA       NA
`UniversityUniversity of Northern Iowa`                    NA          NA       NA       NA
`UniversityUniversity of Oregon`                           NA          NA       NA       NA
`UniversityUniversity of Pittsburgh`                       NA          NA       NA       NA
`UniversityUniversity of Vermont`                          NA          NA       NA       NA
`UniversityUniversity of Virginia`                  1.163e+05   2.970e+04    3.917 9.06e-05 ***
`UniversityUniversity of Washington Tacoma`                NA          NA       NA       NA
`UniversityUniversity of Wisconsin Madison`                NA          NA       NA       NA
`UniversityUtah State University`                          NA          NA       NA       NA
`UniversityVirginia Tech`                                  NA          NA       NA       NA
year_typecentennial                                 5.663e+04   1.674e+04    3.383 0.000721 ***
year_typenew                                       -4.928e+04   5.785e+03   -8.518  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 184700 on 7403 degrees of freedom
Multiple R-squared:  0.7229,    Adjusted R-squared:  0.7207
F-statistic: 338.8 on 57 and 7403 DF,  p-value: < 2.2e-16
```

>>說明：與model1流程大致相同，取而代之的是我不打算利用剛開始的年分3分類預測（因此過濾出它另存成df2），而是直接用原始的年份（保留下來）來進行配適model2。

```
> ## Adjusted model
> ## Delete date, TownF, NA column and year type
> df2<-houseprice.df[,c(-3,-10,-12,-14)]
> str(df2)
'data.frame':   10659 obs. of  10 variables:
 $ Record     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Sale_amount: int  295000 240000 385000 268000 186000 302500 223000 225000 215000 285100 ...
 $ Beds       : int  5 4 5 3 3 4 3 3 5 3 ...
 $ Baths      : num  3 2 4 2.5 1.25 3 2 3 2 4 ...
 $ Sqft_home  : int  2020 1498 4000 2283 1527 3117 1218 3000 2963 1680 ...
 $ Sqft_lot   : num  38333 54014 85813 118919 15682 ...
 $ Type       : Factor w/ 3 levels "Multi Family",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ Build_year : int  1976 2002 2001 1972 1975 1976 1975 1969 1965 1987 ...
 $ TownS      : chr  "IA" "IA" "IA" "IA" ...
 $ University : Factor w/ 50 levels "Arizona State university",..: 10 10 10 10 10 10 10 10 10 10 ...
> df2$TownS<-as.factor(df2$TownS)
> levels(df2$TownS)
 [1] "AL"       "AR"       "Arbor"    "AZ"       "CA"       "Cedar"    "City"     "CO"       "College"  "Collins"
[11] "CT"       "FL"       "Forks"    "GA"       "Hill"     "IA"       "IL"       "IN"       "KS"       "KY"
[21] "Lansing"  "Luis"     "MA"       "MN"       "MO"       "MS"       "MT"       "ND"       "NE"       "NY"
[31] "OR"       "PA"       "Station"  "Urbana"   "UT"       "VA"       "VT"       "WA"       "WI"       "WV"
> dummies2.1=dummy.data.frame(df2)
警告訊息：
1: 於 model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE)：
  non-list contrasts argument ignored
2: 於 model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE)：
  non-list contrasts argument ignored
3: 於 model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE)：
  non-list contrasts argument ignored
```

>>說明：一樣過濾出各dummy variables的其中一項避免共線性，並切割資料成訓練集與測試集。

```
> ##Delete the one dummy of different factor
> #SingleFamily
> #AL
> #University ofWest Virginia
> dummies2.2<-dummies2.1[,c(-9,-11,-100)]
> ## SPLIT THE DATA INTO TRAINING AND TESTING
> train_df2 <- dummies2.2 %>% sample_frac(0.7)
> test_df2  <- anti_join(dummies2.2, train_df2, by = 'Record')
> train_df2 <-train_df2[,-1]
> test_df2  <-test_df2[,-1]
>
```

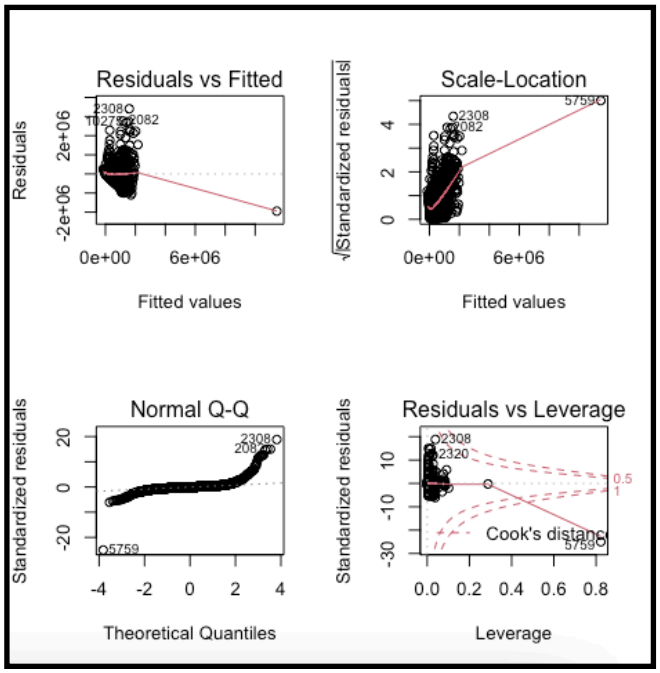
>>說明：配飾模型2（共用以下這些 97 個 predictor:
[1] "Baths" [2] "Beds" [3]"Build_year"
[4] "Sale_amount" [5] "Sqft_home" [6] "Sqft_lot"
[7] "TownSAR" …. [45]"TownSWV"
[46] "TypeMulti Family" [47] "TypeMultiple Occupancy"
[48] "UniversityArizona State university" …. [97] "UniversityVirginia Tech"
）並由model2模型結果R-square我們可以觀測到模型變異佔了約70%的總變異。

```
> #Regression model 2
> m2 <- lm(Sale_amount~ ., data=train_df2)
> summary(m2)

Call:
lm(formula = Sale_amount ~ ., data = train_df2)

Residuals:
     Min       1Q   Median       3Q      Max
-1346299   -57777    -2540    45396  3507810

Coefficients: (39 not defined because of singularities)
                                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                                        8.892e+05  1.641e+05   5.419 6.17e-08 ***
Beds                                              -2.184e+03  2.697e+03  -0.810 0.418104
Baths                                              4.441e+04  3.607e+03  12.315  < 2e-16 ***
Sqft_home                                          1.210e+02  3.489e+00  34.675  < 2e-16 ***
Sqft_lot                                           2.721e-01  4.036e-02   6.741 1.70e-11 ***
`TypeMulti Family`                                -1.277e+05  1.520e+04  -8.403  < 2e-16 ***
`TypeMultiple Occupancy`                          -9.479e+04  2.530e+04  -3.747 0.000181 ***
Build_year                                        -5.215e+02  8.324e+01  -6.265 3.93e-10 ***
TownSAR                                            7.004e+03  1.843e+04   0.380 0.703943
TownSArbor                                         1.751e+05  1.903e+04   9.197  < 2e-16 ***
TownSAZ                                            1.606e+05  1.891e+04   8.492  < 2e-16 ***
TownSCA                                            9.594e+05  1.994e+04  48.115  < 2e-16 ***
TownSCedar                                         4.110e+03  2.109e+04   0.195 0.845496
TownSCity                                          2.908e+04  1.916e+04   1.518 0.129151
TownSCO                                            6.601e+05  1.940e+04  34.031  < 2e-16 ***
TownSCollege                                       1.163e+05  3.219e+04   3.613 0.000305 ***
TownSCollins                                       1.147e+05  1.828e+04   6.275 3.69e-10 ***
TownSCT                                           -4.261e+04  2.278e+04  -1.870 0.061488 .
TownSFL                                            7.710e+04  1.933e+04   3.989 6.71e-05 ***
TownSForks                                         1.663e+04  3.184e+04   0.522 0.601513
TownSGA                                            1.843e+04  1.929e+04   0.956 0.339333
TownSHill                                          1.285e+05  1.823e+04   7.045 2.03e-12 ***
TownSIA                                            7.722e+04  2.082e+04   3.710 0.000209 ***
```

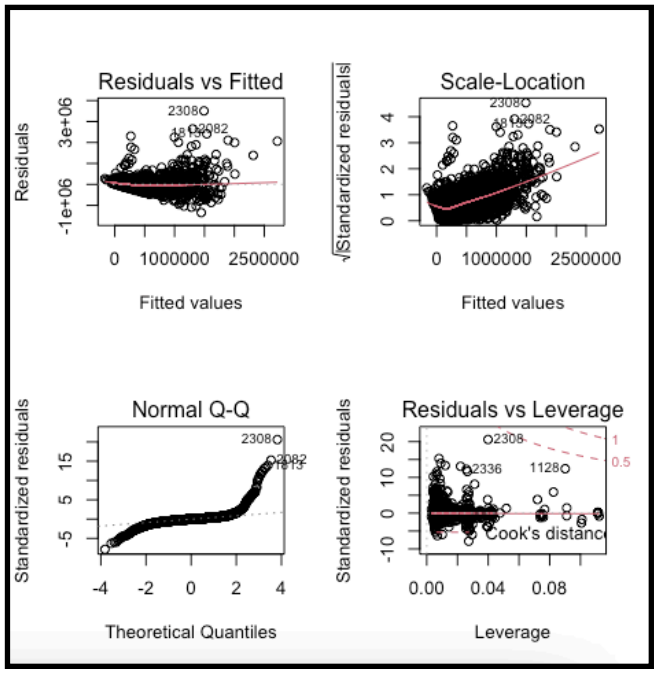
```
`UniversityUniversity of Michigan`                                NA          NA      NA       NA
`UniversityUniversity of Minnesota`                               NA          NA      NA       NA
`UniversityUniversity of Mississippi`                             NA          NA      NA       NA
`UniversityUniversity of Missouri`                                NA          NA      NA       NA
`UniversityUniversity of Nebraska Lincoln`                        NA          NA      NA       NA
`UniversityUniversity of North Carolina at Chapel Hill`           NA          NA      NA       NA
`UniversityUniversity of North Dakota`                            NA          NA      NA       NA
`UniversityUniversity of Northern Iowa`                           NA          NA      NA       NA
`UniversityUniversity of Oregon`                                  NA          NA      NA       NA
`UniversityUniversity of Pittsburgh`                              NA          NA      NA       NA
`UniversityUniversity of Vermont`                                 NA          NA      NA       NA
`UniversityUniversity of Virginia`                          9.941e+04   3.051e+04   3.258 0.001126 **
`UniversityUniversity of Washington Tacoma`                       NA          NA      NA       NA
`UniversityUniversity of Wisconsin Madison`                       NA          NA      NA       NA
`UniversityUtah State University`                                 NA          NA      NA       NA
`UniversityVirginia Tech`                                         NA          NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 173900 on 7404 degrees of freedom
Multiple R-squared:  0.7118,    Adjusted R-squared:  0.7096
F-statistic: 326.6 on 56 and 7404 DF,  p-value: < 2.2e-16

>
```

>>說明：接著透過畫圖來檢視2個model的配適情況。

```
> ##plot
> layout(matrix(c(1,2,3,4),2,2))
> plot(m1)
> plot(m2)
>
```



>>plot(m1)



>>plot(m2)

>>說明：最後來透過RMSE這個指標比較兩個model的配適良好情況（愈小表示配適情況越好），雖然2個model R square都有達到7成，但還是可以發現model1的配適似乎是比model2來的好一點。（與後來預期所做的假設有落差）

```
> ## test
> predict=predict(m1,test_df1[,-1])
警告訊息：
於 predict.lm(m1, test_df1[, -1])：
  prediction from a rank-deficient fit may be misleading
> RMSE=sqrt(mean(sum((test_df1$Sale_amount-predict)^2)))
> RMSE
[1] 9731461
> predict2=predict(m2,test_df2[,-1])
警告訊息：
於 predict.lm(m2, test_df2[, -1])：
  prediction from a rank-deficient fit may be misleading
> RMSE=sqrt(mean(sum((test_df2$Sale_amount-predict2)^2)))
> RMSE
[1] 13706549
>
```

附錄: R 程式碼

```
#HW2

setwd("~/Downloads/1102 R/HW/hw 2")
#1a
##生成一筆資料
data <- data.frame(
  a = sample(1:10,20,replace=T),
  ε = rnorm(20,mean=0,sd=2))
##觀察sum（用來檢查ifelse的結果）
library(tidyverse)
data<- data%>%
  mutate(sum= a+ε)
##利用ifelse來確保X會落在0-11區間
data$X <- ifelse( data$a + data$ε >= 11, 11,
                          ifelse( 0 <= data$a + data$ε, data$a +
data$ε ,0))
##所求
X=data$X
head(X)

#1b
##Cauchy density function 取log及一階導數的function
Lcauchy <- function(theta){
  k = 0
  for (i in 1:length(X)){
    k <- k+(theta-X[i])/(1+(theta-X[i])^2)
  }
  return(-2*k)}

Lcauchy(theta)

#1c
##令theta=0.3，代入先前的X data
Lcauchy(0.3)
```

```
#2
##載入資料
houseprice.df <- read.csv("houseprice.csv")
##觀察資料並看看年份有沒有異常值
str(houseprice.df)
#2a
##方便操作
library(tidyverse)
houseprice.df<- mutate(houseprice.df, year_type=Build_year)
range(houseprice.df$year_type)
attach(houseprice.df)

##根據題目更改的條件
houseprice.df$year_type<- ifelse( year_type <= 1899 , "centennial"
,
          ifelse( 1960 <= year_type , "new","old"))
head(houseprice.df)

#################################
## Double check有沒有轉錯，不要跑

#which(year_type <= 1899)
#which(1960 <= year_type)

attach(houseprice.df)
which(year_type=="centennial")
which(year_type=="new")
which(year_type=="old")
#################################

## Preparation to change the variable type into factor variable
houseprice.df$Town<-as.factor(houseprice.df$Town)
houseprice.df$University<-as.factor(houseprice.df$University)
houseprice.df$year_type<-as.factor(houseprice.df$year_type)
houseprice.df$Type<-as.factor(houseprice.df$Type)

summary(houseprice.df)

## Find the each factor name
levels(houseprice.df$Type)
levels(houseprice.df$Town)
levels(houseprice.df$University)
levels(houseprice.df$year_type)


## Classify the area into short name
library(dplyr)
houseprice.df<-houseprice.df %>% separate(Town, c("TownF",
"TownS",sep=","))
```

```
## check the data
str(houseprice.df)
## Delete date, build year, TownF and NA column
df<-houseprice.df[,c(-3,-9,-10,-12)]
str(df)
df$TownS<-as.factor(df$TownS)
levels(df$TownS)

## Use the dummy variable to predict factor variable
library(dummies)
dummies=dummy.data.frame(df)
str(dummies)

##Delete the one dummy of different factor
#SingleFamily
#AL
#University ofWest Virginia
#yeartype-old
dummies1.2<-dummies[,c(-9,-10,-99,-102)]


## SPLIT THE DATA INTO TRAINING AND TESTING
train_df1 <- dummies1.2 %>% sample_frac(0.7)
test_df1  <- anti_join(dummies1.2, train_df1, by = 'Record')

which(is.na(train_df1))

train_df1 <-train_df1[,-1]
test_df1  <-test_df1[,-1]
#Regression model
m1 <- lm(Sale_amount~ ., data=train_df1)
summary(m1)

ls(train_df1)

## Adjusted model
## Delete date, TownF, NA column and year type
df2<-houseprice.df[,c(-3,-10,-12,-14)]
str(df2)
df2$TownS<-as.factor(df2$TownS)
levels(df2$TownS)

dummies2.1=dummy.data.frame(df2)
str(dummies)

##Delete the one dummy of different factor
#SingleFamily
#AL
#University ofWest Virginia
dummies2.2<-dummies2.1[,c(-9,-11,-100)]
```

```
## SPLIT THE DATA INTO TRAINING AND TESTING
train_df2 <- dummies2.2 %>% sample_frac(0.7)
test_df2  <- anti_join(dummies2.2, train_df2, by = 'Record')

which(is.na(train_df2))

train_df2 <-train_df2[,-1]
test_df2  <-test_df2[,-1]

#Regression model 2
m2 <- lm(Sale_amount~ ., data=train_df2)
summary(m2)


##plot
layout(matrix(c(1,2,3,4),2,2))
plot(m1)
plot(m2)


## test
predict=predict(m1,test_df1[,-1])
predict=predict(m1,test_df1)
RMSE=sqrt(mean(sum((test_df1$Sale_amount-predict)^2)))
RMSE

predict2=predict(m2,test_df2[,-1])
RMSE=sqrt(mean(sum((test_df2$Sale_amount-predict2)^2)))
RMSE
```