2014 年 臺灣資料分析競賽

預測房屋價格模型之分析

組別名稱:62531

摘要

近年來政府建立了實價登錄系統,而每個人皆可取得實價登錄系統之資料,而此份資料與實價登錄資料相類似,我們希冀藉由此份分析可應用於實價登錄系統之資料上,並提供對於欲購買住宅、商用住宅之消費者提供一個可預測欲購建物之每平方公尺單價。

對於樣本的選擇,我們選定只有建物單價之資料做為我們主要的分析目標,希望透過線性模型建立房價預測模型以供購屋者參考。我們使用了兩種估計方式 OLS 與 group lasso 進行估計,並以 AIC 與 ten-fold CV 進行模型評估,結果顯示在差不多好的預測能力下,group lasso 展現了較佳的稀疏性,且模型估計結果希望能用來幫助評估影響房價的因素。

目錄

資料整理及變數說明

原始資料共有 682724 筆,變數共有 28 個,我們挑選或經過轉換後,變數剩下 17 個,我們設定目標變數為單價(元/平方公尺)。此份報告目的為找出影響房價之因素,進而建立房價預測模型以供購屋者參考。因此,我們剔除下列變數,茲建說明如下:

1. 土地區段位置/建物區段門牌:此項變數過於詳細,以「縣市」此變數描述即可。(註:「縣市」為一新轉換之變數,見下面說明。)

2. 主要用途:與變數「使用分區或編定」類似，因此擇「使用分區或編定」作為解釋變數。
3. 移轉層次:變數目的不清且填寫規格未統一。
4. 非都市土地使用分區:所提供的資訊太少。
5. 非都市土地使用地:所提供的資訊太少。
6. 總價(元):由於總價與坪數大小有關，故考慮以「單價(元/平方公尺)」此項變數描述。
7. 車位類別:此處我們僅考慮有無車位。
8. 車位總價(元):與報告目的無關。
9. 因為「建築完成年月」遺失值的比例過高，因此，我們不考此變數。

其中「交易筆棟數」、「建築完成年月」及「鄉鎮市區」皆做下列之調整:

1. 「交易筆棟數」利用 R 的程式將此變數分割為 3 個新變數，分別為該筆交易的土地數目、建物數目及車位數，例如：若交易筆棟數為土地 9 建物 1 車位 0，則轉為土地數目 9、建物數目 1 以及車位數目 0 等三個變數。
2. 交易年月以民國 1 年 1 月做基準設定為 1，例如：若「交易年月」為 99 年 6 月則轉成 1194。
3. 將所有鄉鎮做合併，僅以「縣市」做為地區之劃分。

經由上面的變數調整，我們先透過主觀想法討論為何考慮其他變數，說明如下：

1. 土地移轉總面積(平方公尺)、建物移轉總面積(平方公尺)及總樓層數：我們認為土地面積或建物面積影響消費者購買意願以及購買者願付價格，因此納入模型之中。
2. 交易標的：我們只考慮有含蓋建物的標的，這個變數僅用來篩選資料，不納入模型之中。
3. 使用分區或編定：用途主要為三種，住、商以及其他，政府對於不同土地編定的稅額不同，因此此變數會影響目標變數單價。
4. 建物型態: 消費者可以根據自己所需要選擇不同建物類型，如公寓或住宅大樓等。
5. 主要建材:在此我們將建物型態分成鋼筋混凝土與其他兩類，因台灣易發生地震且颱風頻繁，故消費者在購屋時會特別考量建材，故對其成交價亦有影響。
6. 建物現況格局-房、建物現況格局-廳、建物現況格局-衛、建物現況格局-隔間等建物格局影響消費者需求。
7. 有無管理組織：有無管理組織亦是消費者考量的重點之一，此變數對於住宅安全、公共建物之維護有影響。

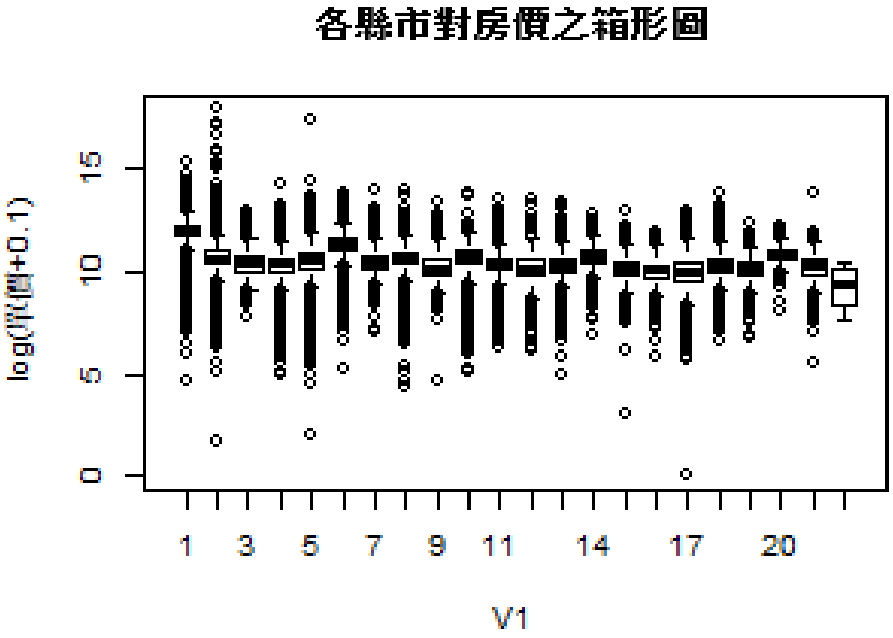
8. 車位移轉總面積(平方公尺)：車位大小影響有車之消費者的購屋意願。

最終的資料筆數為 463591 筆，因為資料遺失比例低，因此只要該筆資料有遺失值便予以刪除，並因為目標變數為 0 為不合理之情形，因此刪除出現該情形之資料。小結以上述標準篩選後，剩下變數之整理如下：

變數名稱	變數代號
單價(元/平方公尺)	Y
縣市	V_1
土地移轉總面積(平方公尺)	V_2
使用分區或編定	V_3
交易年月	V_4
總層數	V_5
建物型態	V_6
主要建材(是否為鋼筋混凝土造)	V_7
建物移轉總面積(平方公尺)	V_8
建物現況格局-房	V_9
建物現況格局-廳	V_{10}
建物現況格局-衛	V_{11}
建物現況格局-格局	V_{12}
有無管理組織	V_{13}
車位移轉總面積(平方公尺)	V_{14}
土地數目	V_{15}
建物數目	V_{16}
車位數目	V_{17}

Model-free graphical data analysis

根據下張圖，我們可以看到台北市(編號 1)的房價相對較高的，其他圖形由於資料複雜度較高，無法提供直接的訊息。



分析過程與模型建立

模型

我們對目標變數 Y 與解釋變數 V’s 建立以下 log-linear 模型:

$$\log(y_i + 0.1) = \beta_0 + \sum_{i=1}^{17} \beta_i V_i + \epsilon_i$$

其中反應變數多加 0.1 的原因為避免反應變數過小時造成的問題。

接下來我們針對該模型考慮兩種不同的估計方式，分別為常見的最小平方法(OLS)以及 group lasso。在最小平方法中，我們使用了 stepAIC 進行模型選取，並以 ten-fold cross-validation 驗證其預測能力。在 group lasso 中，我們同樣以 ten-fold cross-validation 進行 sparsity parameter 的選擇以及驗證其預測能力。

1. OLS

首先由假設ϵ_i為獨立同分配之常態分配，其變異數為σ^2。(註：此處為表示式，若V_i為類別型變數，則轉為 dummy variable 做配適)。在使用 OLS 建立模型後，

我們使用 R function：“stepAIC”進行下一步模型選擇。但 stepAIC 的結果(下表)顯示我們最初配適的模型就具有最低的 AIC，因此所有解釋變數皆被納入模型。

Start:AIC=-741134

log(Y+ 0.1)~ V1 +V2 + V3 +V4 + V5 +V6+V8+V7+V9+V10+ V11 + V12 + V13+V14+V15 + V16 +V17

	Df	Su m of Sq	RSS	AIC
<none>			93675	-741134
0	1	3	93678	-741120
0	1	39	93714	-740942
0	1	60	93735	-740839
0	1	123	93798	-740528
0	1	125	93800	-740518
0	1	149	93824	-740400
0	1	168	93842	-740308
0	1	184	93859	-740226
0	1	315	93990	-739581
0	1	423	94098	-739048
0	1	562	94237	-738363
0	1	905	94580	-736677
0	1	1080	94755	-735822
0	1	1586	95261	-733352
0	8	6004	99679	-712353
0	5	6872	100547	-78327
0	21	108258	201933	-385127

其相對應的參數估計結果如下: 有關於 V1，由於我們選擇 V1 中的 baseline 組別為台北市，因此可看到其他縣市的參數估計值都是負的。這些 V1 的參數估計值提供了跨縣市房價的比較。其餘類別變數的性質符號提供了它們與 baseline 的相對影響力大小關係，而估計值則進一步展現該關係。

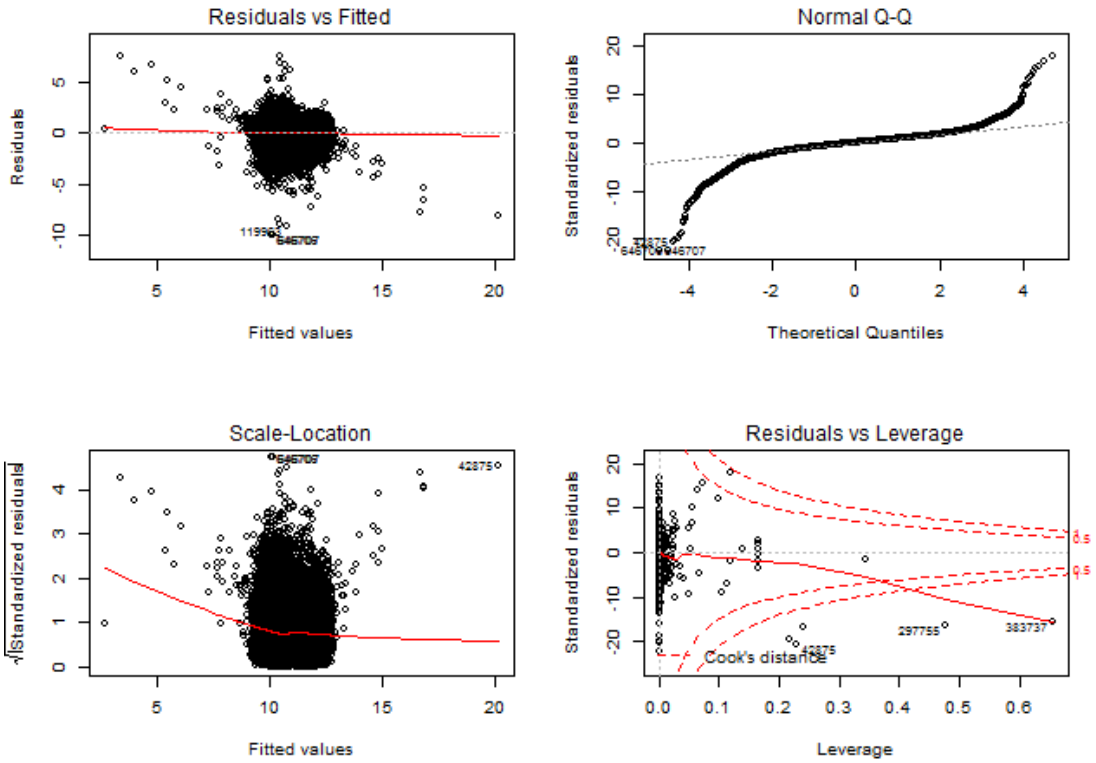
Call:
lm(formula = as.formula(paste("log(Y+0.1)~", paste0("V", c(1:7, 9:18), collapse = "+"))), data = dat4.sub)

Coefficients	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	3.28E+00	1.22E-01	26.993	< 2e-16	***

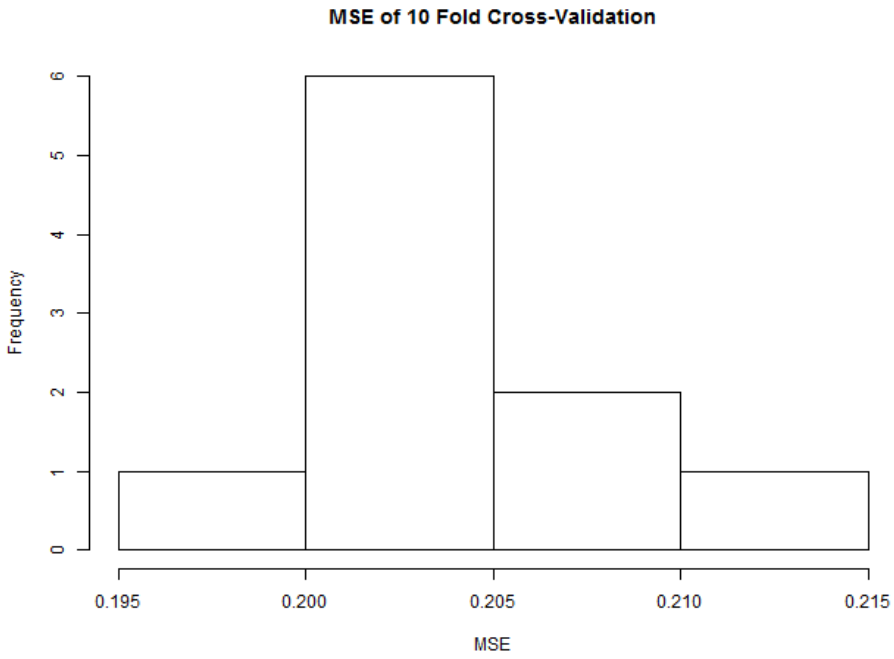
V1 台中市	-1.39E+00	2.84E-03	-490.288	< 2e-16	***
V1 基隆市	-1.59E+00	5.05E-03	-315.216	< 2e-16	***
V1 台南市	-1.77E+00	3.44E-03	-513.815	< 2e-16	***
V1 高雄市	-1.57E+00	2.94E-03	-534.691	< 2e-16	***
V1 新北市	-7.06E-01	2.65E-03	-266.201	< 2e-16	***
V1 宜蘭縣	-1.46E+00	5.22E-03	-278.966	< 2e-16	***
V1 桃園縣	-1.34E+00	2.87E-03	-467.142	< 2e-16	***
V1 嘉義市	-1.87E+00	6.70E-03	-278.581	< 2e-16	***
V1 新竹縣	-1.30E+00	4.34E-03	-300.806	< 2e-16	***
V1 苗栗縣	-1.66E+00	5.85E-03	-284.117	< 2e-16	***
V1 南投縣	-1.83E+00	7.31E-03	-250.436	< 2e-16	***
V1 彰化縣	-1.69E+00	5.22E-03	-323.252	< 2e-16	***
V1 新竹市	-1.25E+00	4.52E-03	-277.354	< 2e-16	***
V1 雲林縣	-1.89E+00	6.55E-03	-289.14	< 2e-16	***
V1 嘉義縣	-1.99E+00	7.79E-03	-254.897	< 2e-16	***
V1 屏東縣	-2.04E+00	5.60E-03	-364.044	< 2e-16	***
V1 花蓮縣	-1.71E+00	6.26E-03	-273.333	< 2e-16	***
V1 台東縣	-2.07E+00	9.85E-03	-209.812	< 2e-16	***
V1 金門縣	-1.14E+00	2.04E-02	-55.89	< 2e-16	***
V1 澎湖縣	-1.72E+00	2.01E-02	-85.732	< 2e-16	***
V1 台中市	-2.54E+00	1.84E-01	-13.811	< 2e-16	***
V2 土地移轉總面積(平方公尺)	4.74E-05	1.65E-06	28.791	< 2e-16	***
V3 工	4.00E-01	5.99E-03	66.725	< 2e-16	***
V3 住	4.74E-01	2.69E-03	176.497	< 2e-16	***
V3 其他	3.96E-01	4.09E-03	96.848	< 2e-16	***
V3 商	5.59E-01	3.33E-03	167.747	< 2e-16	***
V3 農	3.27E-01	9.39E-03	34.796	< 2e-16	***
V4 交易年月	6.60E-03	9.87E-05	66.933	< 2e-16	***
V5 總樓層數	1.63E-02	1.84E-04	88.596	< 2e-16	***
V6 公寓(5 樓含以下無電梯)	9.01E-02	3.33E-03	27.087	< 2e-16	***
V6 店面(店鋪)	1.27E-01	8.77E-03	14.452	< 2e-16	***
V6 透天厝	6.44E-01	5.26E-03	122.444	< 2e-16	***
V6 華廈(10 層含以下有電梯)	5.01E-02	4.19E-03	11.964	< 2e-16	***
V7 主要建材為 CRT	3.35E-01	2.68E-03	124.943	< 2e-16	***

V8 建物移轉面積	2.80E-02	3.01E-03	9.313	< 2e-16	***
V4 交易年月	6.52E-02	1.16E-02	5.624	1.87E-08	***
V5 總樓層數	1.06E-02	7.46E-03	1.42	0.156	
V6 公寓(5 樓含以下無電梯)	-1.65E-01	2.25E-03	-73.1	< 2e-16	***
V6 店面(店鋪)	-9.18E-05	2.01E-06	-45.749	< 2e-16	***
V9 建物現況格局-房	-1.75E-02	7.08E-04	-24.658	< 2e-16	***
V10 建物現況格局廳數	2.47E-02	9.93E-04	24.869	< 2e-16	***
V11 建物現況格局衛浴數	3.86E-02	7.31E-04	52.739	< 2e-16	***
V12 建物現況格局-格局	9.78E-02	3.61E-03	27.132	< 2e-16	***
V13 有無管理組織	-8.53E-03	2.14E-03	-3.988	6.65E-05	***
V14 車位面積	2.97E-04	1.72E-05	17.243	< 2e-16	***
V15 交易土地數	8.72E-03	6.25E-04	13.943	< 2e-16	***
V16 交易建物數	-2.71E-02	8.97E-04	-30.185	< 2e-16	***
V17 交易車位數	2.78E-02	7.04E-04	39.47	< 2e-16	***

下圖為殘差圖。可見到有些違反假設，但此模型的預測能力極佳因此我們仍可接受此模型。



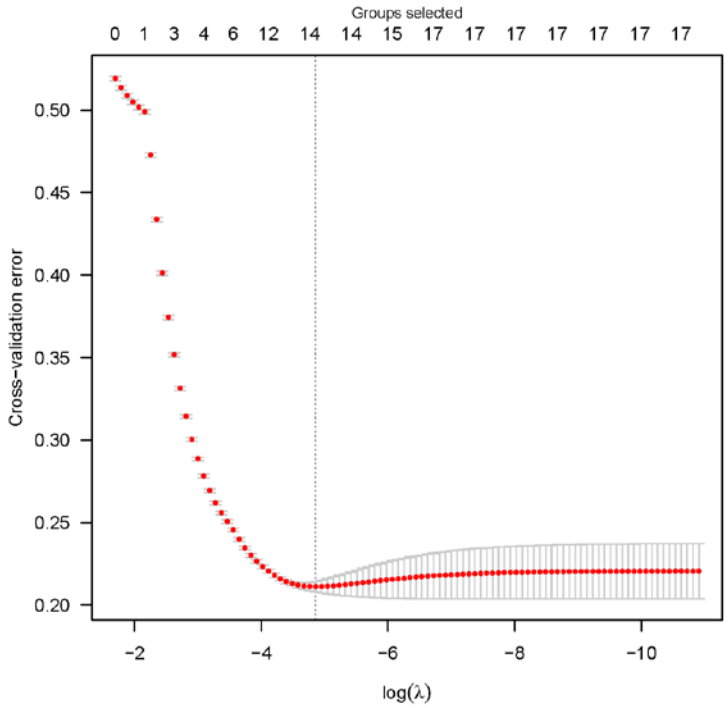
下圖為進行多次 ten-fold CV 的 MSE 直方圖。可看到 MSE 大部分介於 0.195~0.215 間，相較於原始反應變數的變動範圍可算是相當小，因此也說明了此模型的預測能力佳。



2. group lasso:

lasso 的一項重要性質為，L1-penalty 會導致部分參數結果估計為 0，因此同時達到了變數選取與參數估計的目的。但 lasso 的一項明顯缺點為當解釋變數間有高度共線性時，參數估計結果會傾向於只選擇其中一個解釋變數。這種結構的解釋變數常見於 categorical variable。在我們所考慮的模型中，categorical variable 占了大部分。因此我們應用了 group lasso 於此模型中。

下表為隨著 sparsity parameter 而變動的 CV error，ten-fold CV 建議 log(sparsity parameter)= - 4.75 達到了最低的 CV error=0.212，故我們採用此值。



其相對應的配適結果如下。由下表可知: V2, V9 and V13 被屏除於模型外，為較不重要的變數。此估計結果相較於 OLS 達到了較低稀疏性，且個估計值的結果與 OLS 相去不遠。

Glasso	
(Intercept)	6.35
V1 台中市	-1.031
V1 基隆市	-1.227
V1 台南市	-1.377
V1 高雄市	-1.207
V1 新北市	-0.373
V1 宜蘭縣	-1.068
V1 桃園縣	-0.996
V1 嘉義市	-1.427

V1 新竹縣	-0.935
V1 苗栗縣	-1.249
V1 南投縣	-1.394
V1 彰化縣	-1.272
V1 新竹市	-0.877
V1 雲林縣	-1.459
V1 嘉義縣	-1.535
V1 屏東縣	-1.624
V1 花蓮縣	-1.294
V1 台東縣	-1.572
V1 金門縣	-0.561
V1 澎湖縣	-1.137
V3 工	0.227
V3 住	0.34
V3 其他	0.256
V3 商	0.45
V3 農	0.108
V4 交易年月	0.004
V5 總樓層數	0.012
V6 公寓(5 樓含以下無電梯)	-0.097
V6 店面(店鋪)	0.452
V6 透天厝	0.103
V6 華廈(10 層含以下有電梯)	-0.069
V7 主要建材為 CRT	-0.146
V8 建物移轉面積	0
V10 建物現況格局廳數	0.006
V11 建物現況格局衛浴數	0.015
V12 建物現況有隔間	0.064
V14 車位面積	0
V15 交易土地數	0.006
V16 交易建物數	-0.013
V17 交易車位數	0.011

結論

我們考慮了地區、交易地區等變數，對於每平方公尺單價建立 log-linear 模型，我們嘗試用了兩種估計方式，第一種是 Ordinary Least Square，另一種為 Grouped Lasso。最終我們選擇 Grouped Lasso 做為最終模型，此處因為我們參數的稀疏性，這裡使用 Grouped Lasso 相比於 Ordinary Least Square 減少解釋變數的數量，同時還保留模型較好的預測能力，這裡我們使用 10- fold cross validation 得到近似的 MSE。在 Grouped lasso 模型下，其中以下變數 V2(土地移轉總面積)、 V9(建物現況格局-房數)和 V13(有無管理組織)皆不顯著，而 V2 不顯著的原因為與 V8(建物移轉總面積)為高度相關，因為 grouped lasso 的特性，可將高度相關的變數予以刪除，而 V9 不顯著的原因與 V10(建物現況格局-廳)以及 V11(建物現況格局-衛)為高度相關，因此，V9 不顯著。我們選擇的最終模型

此處我們沒應用到交易標的縱坐標和橫坐標，可能以縣市做切割太過廣泛，可以使用 distance based linear model 去做處理，但是經過畫圖之後，發現資料點太過於稀疏，因此，我們並沒有納入考量，未來資料增加之後，可以將之納入考量之中。

我們認為這個方法不只可以提供購屋者一個參考，亦可提供房地產業者對於不同地區、不同格局等有一個簡單方便的估價機制。

附錄:R 程式碼

```
##Read data
setwd("E:\\new\\main_code")
dat = lapply(paste0("data\\List_", LETTERS[LETTERS!="L" & LETTERS!="R" & LETTERS!="S"
&
LETTERS!="Y"], ".csv"), read.csv) # read CSV file
dat2 = do.call("rbind", dat) # combind CSV file
region = unlist(lapply(1:length(dat), function(i) rep(i, nrow(dat[[i]]))))
dat2 = data.frame(dat2, region = region) #combined into CSV file
# Data Preprocessing
# 將不合理的交易年月份刪去
```

```
dat2[dat2[,8] < 5510, ] = NA # 把年份太小的都去掉
dat2$year.trading = sapply(dat2[,8], function(v) substr(as.character(v), 0, nchar(
as.character(v))-2))
dat2$month.trading = sapply(dat2[,8], function(v) substr(as.character(v),
nchar(as.character(
v))-1, nchar(as.character(v))))
#將不合理的建築年份刪去
dat2$year.construction = sapply(dat2[,15], function(v){
if(nchar(v) == 2)
as.integer(v)
else if(nchar(v) == 4 & substr(as.character(v), 1,1) == 0)
as.integer(substr(as.character(v), 1, 2))
else if(nchar(v) == 4 & substr(as.character(v), 1,1) == 1)
as.integer(substr(as.character(v), 1, 3))
else if(nchar(v) == 5 & substr(as.character(v), 1,1) != 0 & substr(as.character(v), 1,1)
!= 1)
as.integer(substr(as.character(v), 1, 2))
else if(nchar(v) == 5 & (substr(as.character(v), 1,1) == 0 | substr(as.character(v),
1,1)
== 1))
as.integer(substr(as.character(v), 1, 3))
else if(nchar(v) == 6 & substr(as.character(v), 1,1) == 0)
as.integer(substr(as.character(v), 1, 2))
else if(nchar(v) == 6 & substr(as.character(v), 1,1) == 1)
as.integer(substr(as.character(v), 1, 3))
else if(nchar(v) == 7)
as.integer(substr(as.character(v), 1, 3))
else
NA
})
dat2$year.construction[dat2$year.construction > 103] = NA
# dat2[dat2$year.construction==10 & !is.na(dat2$year.construction), 15]
#
dat2$month.construction = sapply(dat2[,15], function(v){
if(nchar(v) == 4)
as.integer(substr(as.character(v), nchar(as.character(v))-1,
nchar(as.character(v))))
else if(nchar(v) == 5 & substr(as.character(v), 1,1) != 0 & substr(as.character(v), 1,1)
```

```
!= 1)
as.integer(substr(as.character(v), nchar(as.character(v))-2,
nchar(as.character(v))-2
))
else if(nchar(v) == 5 & (substr(as.character(v), 1,1) == 0 | substr(as.character(v),
1,1)
== 1))
as.integer(substr(as.character(v), nchar(as.character(v))-1,
nchar(as.character(v))))
else if(nchar(v) == 6)
as.integer(substr(as.character(v), nchar(as.character(v))-3,
nchar(as.character(v))-2
))
else if(nchar(v) == 7)
as.integer(substr(as.character(v), nchar(as.character(v))-3,
nchar(as.character(v))-2
))
else
NA
})
dat2$month.construction[dat2$month.construction == 0] = NA
dat2$month.construction[dat2$month.construction == 20] = NA
#建物個數
dat2$building = sapply(dat2[, 2], function(v) grepl("建物", v))
#車位數量
dat2$parking_lot = sapply(dat2[, 2], function(v) grepl("車位", v) & grepl("建物", v))
ttt = do.call("rbind", lapply(dat2[,9], function(v){
if(!is.na(v)){
temp = strsplit(as.character(v), "土地")[[1]][2]
temp2 = strsplit(as.character(temp), "建物")[[1]]
temp3 = strsplit(as.character(temp2[2]), "車位")[[1]]
as.integer(c(temp2[1], temp3))
}
else{
rep(NA, 3)
}
}))
dat2$n.land = ttt[,1]
```

```
dat2$n.building = ttt[,2]
dat2$n.parking_lot = ttt[,3]
dat2[which(dat2[,14]==""),14] = NA
#建築材料
dat2$CRC = sapply(dat2[,14], function(v){
if(grepl("混凝土", v))
TRUE
else if(grepl("鋼骨構造", v))
TRUE
else if(is.na(v))
NA
else
FALSE
})
#交易年份
dat2$trading = sapply(1:nrow(dat2), function(v){
if( !is.na(dat2[v,30]) & !is.na(dat2[v,31]))
as.integer(dat2[v,30]) * 12 + as.integer(dat2[v,31])
else
NA
})
#屋齡
dat2$age_building = sapply(1:nrow(dat2), function(v){
if( !is.na(dat2[v,32]) & !is.na(dat2[v,33]))
dat2[v,40] - dat2[v,32] * 12 + dat2[v,33]
else if(!is.na(dat2[v,32]) & is.na(dat2[v,33]))
dat2[v,40] - dat2[v,32] * 12 + 6
else
NA
})
save(dat2, file = "dat2.RData")
dat3 = dat2[dat2$building,]
dat3 = dat3[,c(23,29,4,5,40,11,12,39,41,16:21,25,36:38)]
names(dat3) = c("Y", paste0("V", 1:(ncol(dat3)-1)))
dat3$V5 = as.numeric(dat3$V5)
dat3$V1 = as.factor(dat3$V1)
dat3$V7 = as.factor(dat3$V7)
dat4 = dat3[-9]
```

```
dat4 = dat4[dat4$Y!=0,]
dat4 = dat4[!is.na(dat4$Y),]
dat4 = dat4[which(apply(dat4,1,function(vec)(!any(is.na(vec))))),]
save(dat4, file = "dat4.RData")
######################
#### Model-free plots
dat4.sub<-dat4[-which(rownames(dat4)%in%c(425940,470677,613622)),]
lm.fit<-lm(log(Y+0.1)~.,data=dat4.sub)
png("E:\\plot1.png",640,480)
par(mfrow=c(2,2))
boxplot(log(Y+0.1)~V1,data=dat4.sub,xlab="V1",ylab="log(單價+0.1)",main=
"各縣市對房價之箱形圖")
plot(dat4.sub$V2,log(dat4.sub$Y+0.1),xlab="V2",ylab="log(單價+0.1)",main=
"土地轉移面積對房價之散布圖")
boxplot(log(Y+0.1)~V3,data=dat4.sub,xlab="V3",ylab="log(單價+0.1)",main=
"使用分區對房價之箱形圖")
plot(dat4.sub$V4,log(dat4.sub$Y+0.1),xlab="V4",ylab="log(單價+0.1)",main=
"交易年月對房價之散布圖")
dev.off()
png("E:\\plot2.png",640,480)
par(mfrow=c(2,2))
plot(dat4.sub$V5,log(dat4.sub$Y+0.1),xlab="V5",ylab="log(單價+0.1)",main=
"總樓層數對房價之散布圖")
boxplot(log(Y+0.1)~V6,data=dat4.sub,xlab="V6",ylab="log(單價+0.1)",main=
"建物型態對房價之箱形圖")
boxplot(log(Y+0.1)~V7,data=dat4.sub,xlab="V7",ylab="log(單價+0.1)",main=
"主要建材對房價之箱形圖")
plot(dat4.sub$V9,log(dat4.sub$Y+0.1),xlab="V9",ylab="log(單價+0.1)",main=
"建物移轉面積對房價之散布圖")
dev.off()
png("E:\\plot3.png",640,480)
par(mfrow=c(2,2))
plot(dat4.sub$V10,log(dat4.sub$Y+0.1),xlab="V10",ylab="log(單價+0.1)",main=
"建物現況格局-房數對房價之散布圖")
plot(dat4.sub$V11,log(dat4.sub$Y+0.1),xlab="V11",ylab="log(單價+0.1)",main=
"建物現況格局-廳數對房價之散布圖")
plot(dat4.sub$V12,log(dat4.sub$Y+0.1),xlab="V12",ylab="log(單價+0.1)",main=
"建物現況格局-衛浴數對房價之散布圖")
```

```
boxplot(log(Y+0.1)~V13,data=dat4.sub,xlab="V13",ylab="log(單價+0.1)",main=
"建物現況是否有隔間對房價之箱形圖")
E:\\plot4.png",640,480)
par(mfrow=c(2,2))
boxplot(log(Y+0.1)~V14,data=dat4.sub,xlab="V14",ylab="log(單價+0.1)",main=
"建物現況是否有管理組織對房價之箱形圖")
plot(dat4.sub$V15,log(dat4.sub$Y+0.1),xlab="V15",ylab="log(單價+0.1)",main=
"車位面積對房價之散布圖")
plot(dat4.sub$V16,log(dat4.sub$Y+0.1),xlab="V16",ylab="log(單價+0.1)",main=
"交易土地數對房價之散布圖")
plot(dat4.sub$V17,log(dat4.sub$Y+0.1),xlab="V17",ylab="log(單價+0.1)",main=
"交易建物數對房價之散布圖")
dev.off()
png("E:\\plot5.png",640,480)
plot(dat4.sub$V18,log(dat4.sub$Y+0.1),xlab="V18",ylab="log(單價+0.1)",main=
"交易車位數對房價之散布圖")
dev.off()
######################
##### Modelling
#### OLS
lm.fit <- lm(as.formula(paste("log(Y+0.1)~", paste0("V",c(1:7,9:18), collapse =
"+"))),data =
dat4)
summary(lm.fit)
plot(lm.fit)
### delete outlier
dat4.sub <- dat4[-which(rownames(dat4) %in% c(425940,470677,613622)),]
lm.fit.revised <- lm(as.formula(paste("log(Y+0.1)~", paste0("V",c(1:7,9:18), collapse
= "+"
))),data = dat4.sub)
### residual plot
png("lm_residual.png",width =640,height =480)
par(mfrow =c (2,2))
plot(lm.fit.revised)
dev.off()
### StepAIC
library(MASS)
lm.AIC <- stepAIC(lm.fit)
```

```
# PRESS
fold = 10
# divide complete data into 10 fold cross validation set.
cv_index_f = function(n, fold = 10){
fold_n = floor(n / fold)
rem = n - fold_n * fold
size = rep(fold_n, fold)
if(rem > 0)
size[1:rem] = fold_n + 1
cv_index = unlist(sapply(1:fold, function(i) rep(i, size[i])))
cv_index = sample(cv_index, length(cv_index))
return(cv_index)
}
index = cv_index_f(nrow(dat4.sub), fold)
lm.CV <- sapply(1:fold, function(v){
dat4.train = dat4.sub[index != v,]
dat4.test = dat4.sub[index == v,]
```



```
lm.fit.train = lm(as.formula(paste("log(Y+0.1)~", paste0("V",c(1:7,9:18), collapse =
"+"
))),data = dat4.train)
sum((log(dat4.test$Y+0.1) - predict(lm.fit.train, dat4.test))^2)/nrow(dat4.test)
})
mean(lm.CV)
## histogram of 10 fold cross -validation
png("lm_CV.png",width =640,height=480)
hist(lm.CV,main = "MSE of 10 Fold Cross-Validation",xlab = "MSE")
dev.off()
#### Lasso and Grouped Lasso
library(glmnet)
library(grpreg)
#删去有缺失值的樣本
index.v <- which(apply(dat4,1,function(vec)(!any(is.na(vec)))))
dat4.sub <- dat4[index.v,]
#產生放入"cv.glmnet"的公式
lasso.formula <- formula(paste0("log(Y+0.1)~",paste0("V",c(1:7,9:18),collapse="+")))
X.m <- model.matrix(lasso.formula , data=dat4.sub)[,-1]
```

```
#Set groups for "grpreg".
group.v <- substring(colnames(X.m),1,2)
group.v[43:51] <- substring(colnames(X.m),1,3)[43:51]
group.v <- as.numeric(factor(group.v, levels=paste("V",c(1:7,9:18),sep="")))
#Conduct Group Lasso
out.glasso <- cv.grpreg(X.m, log(dat4.sub$Y+0.1), group=group.v)
#Estimation Result
round(coef(out.glasso)[coef(out.glasso)!=0],3)
round(coef(out.lasso)[,1][coef(out.lasso)[,1]!=0],3)
#About C.V.
pdf("C:/GLasso_cv.pdf")
plot(out.lasso)
dev.off()
```