

Regresión - Buscando un Hogar

Leonardo Santella

Friday, March 11, 2016

Introducción

Se nos ha otorgado un set de datos con una serie de características (features) acerca de viviendas estudiantiles en Roma. A través de la generación de uno o varios modelos de Regresión Lineal se busca poder predecir el precio de alquiler de una vivienda y además, utilizando el/los modelos, escoger inteligentemente un hogar.

Carga de librerías y funciones

```
#sources and libraris
```

```
library(xlsx)
```

```
## Loading required package: rJava
```

```
## Loading required package: xlsxjars
```

```
library(curl)
```

```
library(FactoMineR)
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
```

```
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##   cov, smooth, var
```

```
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
```

```
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
```

```
##
```

```
##   lowess
```

```
source("google_api.R")
```

```
## Loading required package: jsonlite
```

Carga de los datos

#Reading data

```
data <- read.xlsx("hogares.xlsx", sheetIndex = 1, header = T, startRow = 1, endRow = 104)
```

Preprocesamiento - Eliminación de características irrelevantes. Agregación de características derivadas

#Irrelevant Features

```
data$Foto <- NULL  
data$Piso <- NULL
```

#Adding Services

```
data$Condominio <- 0  
data$Servicio.Basura <- 0  
data$Agua <- 0  
data$Internet <- 0  
data$Calefaccion <- 0
```

#Miscs

```
data$Entrada <- 0  
data$Habitacion <- 0  
data$Cocina <- 0  
data$Bano <- 0  
data$Terraza <- 0  
data$Sala <- 0
```

#Para hombres, mujeres o unisex (1,2 o 3)

```
data$Disponibilidad <- 0
```

#Tipo de habitacion (Monolocale, Singola, Doppia, Intero appartamento, Mini-Apartamento, Posto Letto)

```
data$Tipo.Habitacion <- 0
```

#Duracion entre origen y destino, segun google maps, en minutos

```
data$Tiempo <- 0
```

#Fixing. The API Returned status!=OK for these values before

```
data$Dirección <- as.character(data$Dirección)  
data[11,]$Dirección = "Galliate"  
data[33,]$Dirección = "Via San Roberto Bellarmino"  
data[61,]$Dirección = "Via di Monte Verde"
```

Utilización del API de Google.

Uno de los objetivos propuestos era transformar la columna (feature) de Dirección por otra columna, con datos obtenidos a través del API de

google. En este caso utilizaremos el Tiempo. Dicho tiempo es el que se necesita para ir desde el origen (proporcionado por la columna Dirección) hasta el destino, que en este caso es 'Piazzale Aldo Moro', tomando una ruta y con una velocidad definida por el API.

```
#Using Google API for distance
APIKey <- "AlzaSyCCcqG81xzm1jWcgRs095rx7s913eOiQvc"
data$Dirección <- gsub("[\\n\\r]", " ", data$Dirección)
data$Dirección <- as.character(data$Dirección)
destino <- "Piazzale Aldo Moro"
for(i in seq(1:nrow(data)))
{
  origen <- as.character(data$Dirección[i])
  api_url <- get_url(origen, destino, APIKey)
  datos <- get_data(api_url)
  rAPI <- parse_data(datos)
  if(rAPI$status == "OK")
  {
    aux <- strsplit(as.character(rAPI$duration$text), " ")
    aux <- paste(aux[[1]], collapse = "")
    aux <- strsplit(as.character(aux), "min")
    aux <- strsplit(as.character(aux), "h")
    if (length(aux[[1]]) == 1)
    {
      data$Tiempo[i] = aux[[1]][1]
    }
    else
    {
      data$Tiempo[i] = (as.integer(aux[[1]][1])*60) + as.integer(aux[[1]][2])
    }
  }
}
data$Tiempo <- as.numeric(data$Tiempo)
```

Preprocesamiento - Llenar las variables antes agregadas

En este caso, se obtendrán datos de las columnas existentes en el set de datos y se derivarán otros.

En el caso de que el precio de alquiler contemplara todo los gastos incluidos ('Tutto incluso') asumimos que se refiere unicamente a los servicios (agua, calefacción, condominio, internet, etc)

Se agregó un atributo que indica cierto valor, derivado de las existencia de algunas propiedades escogidas bajo un criterio personal.

#Disponibilidad Hombres: 1 Mujeres: 2 Ambos: 3

```
ambos <- grep("ragazzi/ragazze", as.character(data$Notas))
aux <- grep('ragazze/ragazzi', as.character(data$Notas))
ambos <- union(ambos, aux)
hombres <- grep('ragazzi', as.character(data$Notas))
hombres <- setdiff(hombres, ambos)
aux = union(ambos, hombres)
data[hombres,]$Disponibilidad <- 1
data[-aux,]$Disponibilidad <- 2
data[ambos,]$Disponibilidad <- 3
```

#Tipo de Cocina Cucina:1 Cucina/living:2 Angolo Cottura:3 Cucina abitabile:4

```
cAbitabile <- grep('cucina abitabile', data$Descripción, ignore.case = T)
aCottura <- grep('angolo cottura', data$Descripción, ignore.case = T)
cLiving <- grep('cucina/living', data$Descripción, ignore.case = T)
aux <- union(cAbitabile, aCottura)
aux <- union(aux, cLiving)
data[-aux,]$Cocina <- 1
data[cLiving,]$Cocina <- 2
data[aCottura,]$Cocina <- 3
data[cAbitabile,]$Cocina <- 4
```

#Baños

```
banos1 <- grep('bagno', data$Descripción, ignore.case = T)
banos2 <- grep('2 bagni', data$Descripción, ignore.case = T)
banos3 <- grep('3 bagni', data$Descripción, ignore.case = T)
banos4 <- grep('4 bagni', data$Descripción, ignore.case = T)
data[banos1,]$Bano <- 1
data[banos2,]$Bano <- 2
data[banos3,]$Bano <- 3
data[banos4,]$Bano <- 4
data[76,]$Bano <- 2
```

#Habitaciones

```
habitaciones5 <- grep('5 camere', data$Descripción, ignore.case = T)
habitaciones4 <- grep('4 camere', data$Descripción, ignore.case = T)
habitaciones3 <- grep('3 camere', data$Descripción, ignore.case = T)
habitaciones2 <- grep('2 camere', data$Descripción, ignore.case = T)
habitaciones1 <- grep('camera', data$Descripción, ignore.case = T)
habitaciones4 <- union(16, habitaciones4)
habitaciones2 <- union(92, habitaciones2)
habitaciones3 <- union(1, habitaciones3)
data[habitaciones1,]$Habitacion <- 1
data[habitaciones2,]$Habitacion <- 2
data[habitaciones3,]$Habitacion <- 3
data[habitaciones4,]$Habitacion <- 4
data[habitaciones5,]$Habitacion <- 5
rownames(data[, data$Habitacion == 0, ]) #For fixing rare values
```

```
## character(0)
```

#Terraza Terrazo: 1 Terrazzino: 2 Balcone: 3

```
terrazza1 <- grep('terrazzo', data$Descripción, ignore.case = T)
terrazza2 <- grep('balcone', data$Descripción, ignore.case = T)
data[terrazza1, ]$Terraza <- 1
data[terrazza2, ]$Terraza <- 2
```

#Sala de estar Soggiorno:1 salone/living:2 salotto:3 salottino:4

```
sala1 <- grep('soggiorno', data$Descripción, ignore.case = T)
sala2 <- grep('salone', data$Descripción, ignore.case = T)
sala3 <- grep('salotto', data$Descripción, ignore.case = T)
sala4 <- grep('salottino', data$Descripción, ignore.case = T)
aux <- grep('living', data$Descripción, ignore.case = T)
sala2 <- union(sala2, aux)
data[sala1, ]$Sala <- 1
data[sala2, ]$Sala <- 2
data[sala3, ]$Sala <- 3
data[sala4, ]$Sala <- 4
```

#Entrada

```
entrada <- grep('ingresso', data$Descripción, ignore.case = T)
data[entrada, ]$Entrada <- 1
```

#Tipo de Habitación singola:1 doppia:2 posto letto:3 intero appartamento:4 monolocale:5

```
tHab1 <- grep('singola|singole', data$Habitaciones.Disponibles, ignore.case = T)
tHab2 <- grep('doppia|doppie', data$Habitaciones.Disponibles, ignore.case = T)
tHab2 <- setdiff(tHab2, tHab1)
tHab3 <- grep('posto letto', data$Habitaciones.Disponibles, ignore.case = T)
tHab3 <- setdiff(tHab3, tHab1)
tHab3 <- setdiff(tHab3, tHab2)
tHab4 <- grep('intero appartamento', data$Habitaciones.Disponibles, ignore.case = T)
tHab5 <- grep('monolocale', data$Habitaciones.Disponibles, ignore.case = T)
tHab6 <- grep('mini appartamento', data$Habitaciones.Disponibles, ignore.case = T)
data[tHab1, ]$Tipo.Habitacion <- 1
data[tHab2, ]$Tipo.Habitacion <- 2
data[tHab3, ]$Tipo.Habitacion <- 3
data[tHab4, ]$Tipo.Habitacion <- 4
data[tHab5, ]$Tipo.Habitacion <- 5
data[tHab6, ]$Tipo.Habitacion <- 6
```

#Servicios

```
tlIncluido <- grep('tutto incluso', data$Precio.Mensual, ignore.case = T)
tlIncluido <- union(tlIncluido, 78)
noIncluido <- grep('spese escluse', data$Precio.Mensual, ignore.case = T)
complemento <- setdiff(1:103, union(tlIncluido, noIncluido))
```

```

calefaccion <- grep('riscaldamento', data$Precio.Mensual, ignore.case = T)
agua <- grep('acqua', data$Precio.Mensual, ignore.case = T)
internet <- grep('internet', data$Precio.Mensual, ignore.case = T)
condominio <- grep('condominio', data$Precio.Mensual, ignore.case = T)
sBasura <- grep('rifiuti', data$Precio.Mensual, ignore.case = T)
data[calefaccion, ]$Calefaccion <- 1
data[agua, ]$Agua <- 1
data[internet, ]$Internet <- 1
data[condominio, ]$Condominio <- 1
data[sBasura, ]$Servicio.Basura <- 1
#Todo incluido (el precio incluye todos los servicios)
data[tIncluido, ]$Agua <- 1
data[tIncluido, ]$Internet <- 1
data[tIncluido, ]$Servicio.Basura <- 1
data[tIncluido, ]$Condominio <- 1
data[tIncluido, ]$Calefaccion <- 1
data[48,]$Internet <- 0

#Asignar valor a partir de características
vTiempo <- (round(1/data$Tiempo * 500))
valor <- 1:nrow(data)
for( i in 1:nrow(data) ){
  vCondominio <- (data[2,]$Condominio) * 50
  vBasura <- data[i,]$Servicio.Basura * 50
  vAgua <- data[i,]$Agua * 50
  vInternet <- data[i,]$Internet * 75
  vCalefaccion <- data[i,]$Calefaccion * 75
  valor[i] <- vCondominio + vBasura + vAgua + vInternet + vCalefaccion +
vTiempo[i]
}
data$Valor <- valor

```

Preprocesamiento - Agregando Filas

En el set de datos existe una columna en el que se refleja si en una vivienda se encuentra mas de una habitación disponible para alquilar, como también el precio de ésta, en otra columna. A partir de esta información se tomó la decisión de agregar nuevas filas, de manera que cada fila sea una habitación disponible.

```

#Adding Rows. One for each available room (only if there are more than 1)
aux <- data$Precio.Mensual
data$Num.Habitaciones <- gsub(pattern = '([0-9]+).*$', replacement = '\\1',
x = data$Habitaciones.Disponibles)
data$Num.Habitaciones <- gsub(pattern = '(intero appartamento)|(mini
appartamento)|monolocale',
replacement = '1', x = data$Num.Habitaciones,
ignore.case = T)
data$Num.Habitaciones <- as.numeric(data$Num.Habitaciones)

```

```

data <- data[rep(seq_len(nrow(data)), data$Num.Habitaciones),]
aux <-
na.omit(as.numeric(unlist(strsplit(unlist(as.character(data$Precio.Mensual)), "[
^0-9]+"))))
aux[147]

## [1] 370

length(aux)

## [1] 255

data$Precio.Habitacion <- 0
i <- 1
while (i != (nrow(data)+1)) {
  array <-
na.omit(as.numeric(unlist(strsplit(unlist(as.character(data$Precio.Mensual[i]))
,
                                "[^0-9]+")))))
  if (data$Num.Habitaciones[i] == 1){
    data$Precio.Habitacion[i] <- array[1]
    i <- i + 1
  }else{
    for (j in 1:length(array)) {
      data$Precio.Habitacion[i + (j-1)] <- array[j]
    }
    i <- i + as.numeric(data$Num.Habitaciones[i])
  }
}
rownames(data) <- 1:nrow(data)
aux <- rownames(data[ data$Precio.Habitacion == 0, ])
aux <- as.numeric(aux)
for( i in aux ){
  data[i,]$Precio.Habitacion <- data[i-1, ]$Precio.Habitacion
}

```

Muestreo

Para la generación de los sets de datos de entrenamiento y prueba para los modelos de regresión lineal de hombres y mujeres, se realizó un muestreo aleatorio. El 80% de las habitaciones disponibles para hombres/mujeres corresponderá al set de datos de entrenamiento y su complemento corresponderá al set de datos de prueba.

```

####Sampling####
set.seed(666)
hombres <- data$Disponibilidad == 1
aux <- data$Disponibilidad == 3
aux <- rownames(data[aux,])
hombres <- rownames(data[hombres,])

```

```

hombres <- union(hombres, aux)
mujeres <- data$Disponibilidad == 2
aux <- data$Disponibilidad == 3
aux <- rownames(data[aux,])
mujeres <- union(mujeres, aux)
data_h <- data[hombres,]
data_m <- data[mujeres,]
sample_h <- sample(nrow(data_h), nrow(data_h)*0.8, replace = F)
sample_m <- sample(nrow(data_m), nrow(data_m)*0.8, replace = F)
train_h <- data_h[sample_h,]
test_h <- data_h[-sample_h,]
train_m <- data_m[sample_m,]
test_m <- data_m[-sample_m,]

```

Modelos de Regresión

Se realizaron modelos de regresión lineal, tomando en cuenta algunas de las variables derivadas del preprocesamiento, luego, son observados los resultados.

```

### Modelo para hombres ####
linearModel_h <- lm(Precio.Habitacion ~ Bano +
  Cocina + Terraza + Sala +
  Valor, data = train_h)
regresion_h1 <- predict(linearModel_h, newdata = test_h, type='response')
table(regresion_h1, test_h$Precio.Habitacion)

```

```

##
## regresion_h1      350 380 400 450 475 480 500 525 550 650 750 800
850
## 348.700603208912  0  0  1  0  0  0  0  0  0  0  0  0
## 384.369065726503  0  0  0  0  0  0  0  0  1  0  0  0
## 419.564008904418  0  0  0  1  0  0  0  0  0  0  0  0
## 446.603824783213  0  0  1  0  0  0  0  0  0  0  0  0
## 461.797746038485  0  1  0  0  0  0  0  0  0  0  0  0
## 462.079114950619  0  0  0  0  0  0  0  0  0  0  0  1
## 464.048697335562  0  0  0  0  0  1  0  0  0  0  0  0
## 470.827941384751  0  0  0  0  0  1  0  0  0  0  0  0
## 473.451490234466  0  0  0  1  0  0  0  0  0  0  0  0
## 478.140972103377  0  0  0  0  0  0  0  0  0  0  1  0
## 478.234761740755  0  0  0  0  0  0  0  0  0  1  0  0
## 487.332356566442  1  0  0  0  0  0  0  0  0  0  0  0
## 537.095654848724  0  0  0  0  1  0  0  1  0  0  0  0
## 555.668080897524  0  0  0  1  0  0  0  0  0  0  0  0
## 597.454621157869  0  0  0  0  1  0  0  0  0  0  0  0
## 604.582633598614  0  0  0  0  0  0  1  0  0  0  0  0
## 616.869076095161  0  0  0  0  0  0  0  0  1  0  0  0
## 681.684296832209  0  0  0  0  0  1  0  0  0  0  0  0
## 698.619451011558  0  0  0  0  0  0  0  0  0  0  0  0
## 763.742669954054  1  0  0  0  0  0  0  0  0  0  0  0

```



```
##
## regresion_h1      1000
## 348.700603208912  0
## 384.369065726503  0
## 419.564008904418  0
## 446.603824783213  0
## 461.797746038485  0
## 462.079114950619  0
## 464.048697335562  0
## 470.827941384751  0
## 473.451490234466  0
## 478.140972103377  0
## 478.234761740755  0
## 487.332356566442  0
## 537.095654848724  0
## 555.668080897524  0
## 597.454621157869  0
## 604.582633598614  0
## 616.869076095161  0
## 681.684296832209  0
## 698.619451011558  1
## 763.742669954054  0

### Modelo para mujeres ####
linearModel_m <- lm(Precio.Habitacion ~ Bano +
                    Cocina + Terraza + Sala +
                    Valor + Distrito, data = data_m)
regresion_m1 <- predict(linearModel_m, newdata=test_m,type='response')
table(regresion_m1, test_m$Precio.Habitacion)

##
## regresion_m1      300 350 380 420 450 460 475 480 500 525 550 700
750
## 300          1  0  0  0  0  0  0  0  0  0  0  0
## 379.047030986463  0  0  1  0  0  0  0  0  0  0  0  0
## 400.0000000000001  0  1  0  0  0  0  0  0  0  0  0  0
## 401.415448975886  0  0  0  1  0  0  0  0  0  0  0  0
## 445            0  0  0  0  0  1  0  0  0  0  0  0
## 446.080652345787  0  0  0  0  0  0  0  2  0  0  0  0
## 471.34472156376  0  0  0  0  0  0  0  0  0  0  0  1
## 474.854426430506  0  0  0  0  0  0  0  0  1  0  0  0
## 500            0  0  0  0  0  0  0  0  0  1  0  0
## 504.294064614495  0  0  0  0  0  0  0  0  0  1  0  0
## 516.094045820691  0  0  0  0  0  0  0  0  0  0  1  0
## 530.886346239647  0  0  0  0  0  0  1  0  0  0  0  0
## 531.468192256213  0  0  0  0  0  0  1  0  0  0  0  0
## 532.172678207538  0  0  0  0  1  0  0  0  0  0  0  0
## 719.939825895395  0  0  0  0  0  0  0  0  0  0  0  0
## 936.927843339604  0  0  0  0  0  0  0  0  0  0  0  0
## 978.505531575809  0  0  0  0  0  0  0  0  0  0  1  0
##
```

```
## regresion_m1      1000 1300
## 300              0  0
## 379.047030986463  0  0
## 400.0000000000001  0  0
## 401.415448975886  0  0
## 445              0  0
## 446.080652345787  0  0
## 471.34472156376   0  0
## 474.854426430506  0  0
## 500              0  0
## 504.294064614495  0  0
## 516.094045820691  0  0
## 530.886346239647  0  0
## 531.468192256213  0  0
## 532.172678207538  0  0
## 719.939825895395  1  0
## 936.927843339604  0  1
## 978.505531575809  0  0
```

Escogiendo un hogar

Para la selección de un hogar, se utilizaron los modelos de regresión para estimar el precio según las características que posee el inmueble. Las predicciones que cercanas indican un valor proporcional al precio, las que sean menores indican una sobrevaloración y las que sean mayores, se cosiderarán candidatos para la elección de un hogar.

```
## Escogiendo un Hogar
## Escogiendo un Hogar
regresion_h2 <- predict(linearModel_h, newdata = data_h)
regresion_m2 <- predict(linearModel_m, newdata = data_m)
data_h$Regresion <- regresion_h2
data_m$Regresion <- regresion_m2
comp <- cbind.data.frame(regresion = (regresion_h2), precio
=(data_h$Precio.Habitacion))
comp$diff <- comp$regresion - comp$precio
comp[order(-comp$diff),]
```

```
##   regresion precio      diff
## 77  763.7427   350 413.74266995
## 104 703.7561   300 403.75612975
## 27  505.6260   260 245.62597470
## 137 719.7241   500 219.72412958
## 138 719.7241   500 219.72412958
## 150 681.6843   480 201.68429683
## 46  487.7075   300 187.70751512
## 47  487.7075   300 187.70751512
## 48  487.7075   300 187.70751512
## 49  487.7075   300 187.70751512
```

## 64	521.5701	350	171.57014536
## 85	541.3162	380	161.31618853
## 86	541.3162	380	161.31618853
## 43	533.3177	380	153.31767920
## 124	473.2639	325	148.26391096
## 44	487.3324	350	137.33235657
## 45	487.3324	350	137.33235657
## 101	543.3796	410	133.37956055
## 41	533.3177	400	133.31767920
## 39	479.4540	350	129.45402703
## 54	475.7724	350	125.77240201
## 120	464.0487	340	124.04869734
## 121	464.0487	340	124.04869734
## 34	597.5484	475	122.54841080
## 29	597.4546	475	122.45462116
## 42	533.3177	420	113.31767920
## 134	555.6681	450	105.66808090
## 135	555.6681	450	105.66808090
## 136	555.6681	450	105.66808090
## 14	604.5826	500	104.58263360
## 15	604.5826	500	104.58263360
## 110	552.6647	450	102.66473465
## 37	479.4540	380	99.45402703
## 38	479.4540	380	99.45402703
## 123	458.7027	370	88.70268801
## 58	461.7977	380	81.79774604
## 59	461.7977	380	81.79774604
## 99	488.1765	410	78.17646330
## 100	488.1765	410	78.17646330
## 32	597.5484	525	72.54841080
## 35	597.5484	525	72.54841080
## 30	597.4546	525	72.45462116
## 65	521.5701	450	71.57014536
## 81	515.8490	450	65.84897748
## 93	764.3244	700	64.32442665
## 125	537.0957	475	62.09565485
## 95	476.5465	425	51.54654827
## 84	473.5215	425	48.52145071
## 55	478.0472	430	48.04718247
## 2	446.6038	400	46.60382478
## 127	489.9585	450	39.95846641
## 116	462.0791	425	37.07911495
## 40	485.8317	450	35.83172237
## 90	478.1410	450	28.14097210
## 96	476.5465	450	26.54654827
## 131	473.4515	450	23.45149023
## 33	597.5484	575	22.54841080
## 36	597.5484	575	22.54841080
## 31	597.4546	575	22.45462116
## 78	471.6695	450	21.66948712

```

## 126 537.0957 525 12.09565485
## 97 476.5465 475 1.54654827
## 87 450.0740 450 0.07404137
## 106 470.8279 480 -9.17205862
## 51 589.9950 600 -10.00497951
## 22 417.0317 430 -12.96831130
## 23 417.0317 430 -12.96831130
## 107 464.0487 480 -15.95130266
## 108 464.0487 480 -15.95130266
## 109 464.0487 480 -15.95130266
## 28 681.5016 700 -18.49840446
## 89 478.1410 500 -21.85902790
## 10 419.5640 450 -30.43599110
## 11 419.5640 450 -30.43599110
## 7 616.8691 650 -33.13092390
## 139 466.5111 500 -33.48894293
## 20 417.0317 460 -42.96831130
## 21 417.0317 460 -42.96831130
## 26 348.7006 400 -51.29939679
## 60 689.1923 750 -60.80765420
## 88 478.1410 550 -71.85902790
## 122 464.5415 550 -85.45852532
## 12 462.6657 550 -87.33431806
## 25 348.7006 450 -101.29939679
## 140 466.5111 600 -133.48894293
## 56 384.3691 550 -165.63093427
## 57 384.3691 550 -165.63093427
## 145 466.1385 650 -183.86154048
## 91 623.1578 850 -226.84220791
## 98 478.2348 750 -271.76523826
## 92 698.6195 1000 -301.38054899
## 102 478.1410 800 -321.85902790
## 79 647.2133 1000 -352.78665171
## 151 573.9571 950 -376.04293981
## 115 462.0791 850 -387.92088505
## 141 478.2348 900 -421.76523826
## 61 530.0833 1000 -469.91674895
## 80 590.8391 1100 -509.16087278
## 94 698.6195 1300 -601.38054899
## 75 590.3702 1200 -609.62982096
## 13 466.5111 1300 -833.48894293

```

```

comp <- cbind.data.frame(regresion = (regresion_m2), precio
=(data_m$Precio.Habitacion))
comp$diff <- comp$regresion - comp$precio
comp[order(-comp$diff),]

```

```

##      regresion precio      diff
## 12    913.0722    550 3.630722e+02
## 54    647.5809    350 2.975809e+02
## 78    744.4655    450 2.944655e+02

```

```
## 93 978.5055 700 2.785055e+02
## 101 647.2088 410 2.372088e+02
## 120 495.2031 340 1.552031e+02
## 121 495.2031 340 1.552031e+02
## 99 533.0204 410 1.230204e+02
## 100 533.0204 410 1.230204e+02
## 139 616.0827 500 1.160827e+02
## 124 428.6041 325 1.036041e+02
## 110 551.7580 450 1.017580e+02
## 127 532.1727 450 8.217268e+01
## 145 729.0211 650 7.902114e+01
## 116 482.9843 425 5.798435e+01
## 34 531.4682 475 5.646819e+01
## 29 530.8863 475 5.588635e+01
## 60 800.2911 750 5.029115e+01
## 64 400.0000 350 5.000000e+01
## 90 500.0000 450 5.000000e+01
## 95 460.8715 425 3.587149e+01
## 125 504.2941 475 2.929406e+01
## 39 379.0470 350 2.904703e+01
## 43 401.4154 380 2.141545e+01
## 134 466.7637 450 1.676372e+01
## 135 466.7637 450 1.676372e+01
## 136 466.7637 450 1.676372e+01
## 140 616.0827 600 1.608270e+01
## 22 445.0000 430 1.500000e+01
## 23 445.0000 430 1.500000e+01
## 96 460.8715 450 1.087149e+01
## 91 858.6056 850 8.605567e+00
## 32 531.4682 525 6.468192e+00
## 35 531.4682 525 6.468192e+00
## 92 1006.4445 1000 6.444451e+00
## 30 530.8863 525 5.886346e+00
## 41 401.4154 400 1.415449e+00
## 7 650.0000 650 2.160050e-12
## 75 1200.0000 1200 6.821210e-13
## 28 700.0000 700 5.684342e-13
## 51 600.0000 600 5.684342e-13
## 14 500.0000 500 3.979039e-13
## 15 500.0000 500 3.979039e-13
## 44 350.0000 350 2.842171e-13
## 45 350.0000 350 2.842171e-13
## 87 450.0000 450 2.842171e-13
## 46 300.0000 300 2.273737e-13
## 47 300.0000 300 2.273737e-13
## 48 300.0000 300 2.273737e-13
## 49 300.0000 300 2.273737e-13
## 89 500.0000 500 2.273737e-13
## 151 950.0000 950 1.136868e-13
## 10 450.0000 450 -5.684342e-14
```

## 11	450.0000	450	-5.684342e-14
## 37	379.0470	380	-9.529690e-01
## 38	379.0470	380	-9.529690e-01
## 97	460.8715	475	-1.412851e+01
## 20	445.0000	460	-1.500000e+01
## 21	445.0000	460	-1.500000e+01
## 42	401.4154	420	-1.858455e+01
## 131	429.7678	450	-2.023222e+01
## 126	504.2941	525	-2.070594e+01
## 137	474.8544	500	-2.514557e+01
## 138	474.8544	500	-2.514557e+01
## 40	418.6126	450	-3.138744e+01
## 56	516.0940	550	-3.390595e+01
## 57	516.0940	550	-3.390595e+01
## 107	446.0807	480	-3.391935e+01
## 108	446.0807	480	-3.391935e+01
## 109	446.0807	480	-3.391935e+01
## 33	531.4682	575	-4.353181e+01
## 36	531.4682	575	-4.353181e+01
## 31	530.8863	575	-4.411365e+01
## 65	400.0000	450	-5.000000e+01
## 88	500.0000	550	-5.000000e+01
## 79	945.3470	1000	-5.465298e+01
## 122	374.4924	550	-1.755076e+02
## 141	688.8135	900	-2.111865e+02
## 102	562.7912	800	-2.372088e+02
## 80	860.1875	1100	-2.398125e+02
## 98	471.3447	750	-2.786553e+02
## 61	719.9398	1000	-2.800602e+02
## 94	1006.4445	1300	-2.935555e+02
## 13	936.9278	1300	-3.630722e+02
## 115	482.9843	850	-3.670157e+02
## NA	NA	NA	NA
## NA.1	NA	NA	NA