Craigslist apartments

Sue Chew

Nov 21, 2005

Overview

We wish to build an application that helps a user find out the price and size of the apartment that the user is likely to get, given a city and the desired number of bedrooms.

To do this we are going to obtain recent data from craigslist's "apt/housing" for a few United States cities and use it to train a model to predict price and size.

Data processing

It may not be exactly enough data, but lets try doing this with just the 100 most recent posts from each city. `get_craigslist_data.R` was used to download the data, and it stored into `data/citiesdf.rds`

Print summary.

```
citiesdf <- readRDS('data/citiesdf.rds')
str(citiesdf)
```

```
## 'data.frame':    800 obs. of  7 variables:
##  $ city    : Factor w/ 8 levels "chicago","honolulu",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ date    : Factor w/ 1 level "Nov 21": 1 1 1 1 1 1 1 1 1 1 ...
##  $ title   : Factor w/ 740 levels "**SHORT TERM!** 1 bedroom Located in Lincoln Park",..: 26 62 1 49
##  $ price   : num  2848 985 1295 1225 1300 ...
##  $ bedrooms: num  2 NA 1 3 2 NA 1 4 1 2 ...
##  $ sqft    : num  1123 NA NA NA NA ...
##  $ href    : Factor w/ 800 levels "http://chicago.craigslist.org/chc/apa/5283655742.html",..: 76 62 (
```

```
summary(citiesdf)
```

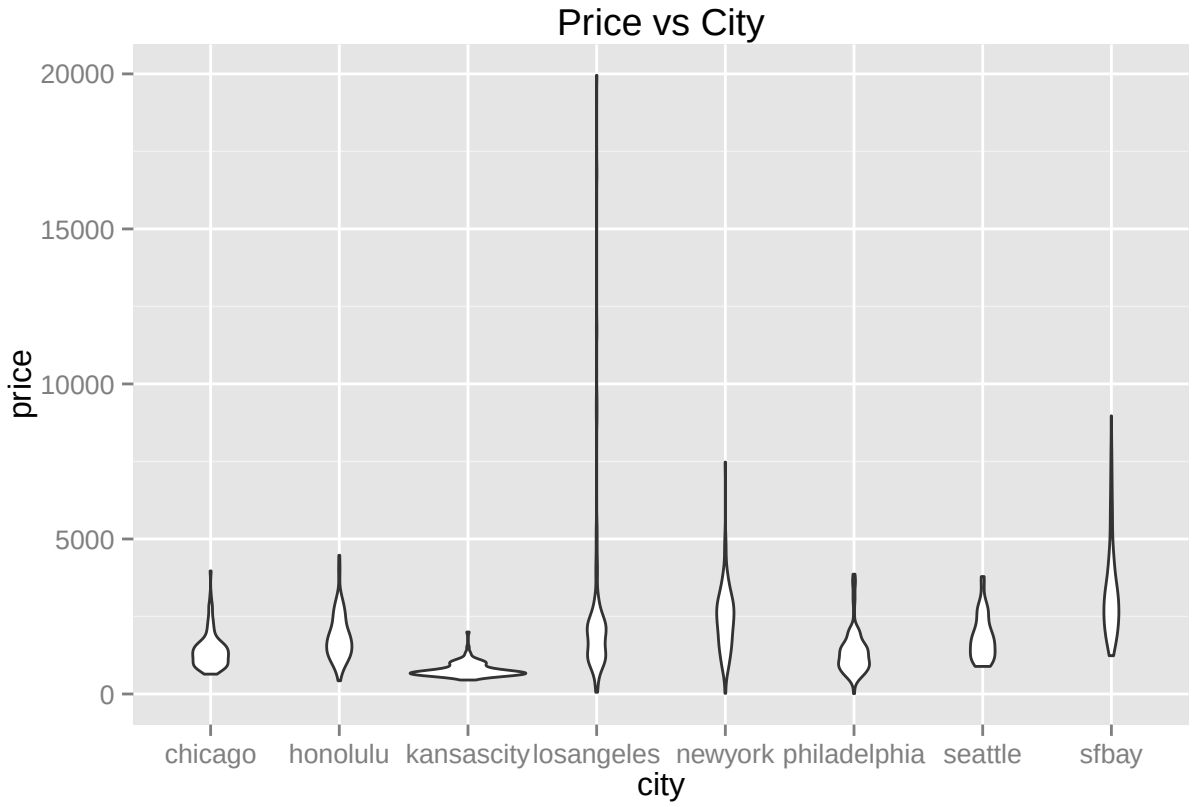
```
##           city         date
##  chicago     :100   Nov 21:800
##  honolulu    :100
##  kansascity  :100
##  losangeles  :100
##  newyork     :100
##  philadelphia:100
##  (Other)     :200
##                                                                 title
##  The Park Apartments                                              : 17
##  Your New Home Is Waiting in Olathe, KS                           :  8
##  Live in the Heart of It All, Community Clubhouse, Quartz Countertops:  7
##  Beverly Plaza Apartments                                         :  6
##  1BR LEFFERT GARDEN NEAR EVERYTHING                               :  5
##  Centrally Located Studio, 1 Bath in Lakeview, Available: Now     :  3
```

```
##  (Other)                                                                :754
##      price          bedrooms          sqft
##  Min.   :    1   Min.   :1.000   Min.   : 175
##  1st Qu.: 1032   1st Qu.:1.000   1st Qu.: 700
##  Median : 1595   Median :2.000   Median : 904
##  Mean   : 1928   Mean   :1.954   Mean   :1038
##  3rd Qu.: 2406   3rd Qu.:2.000   3rd Qu.:1200
##  Max.   :20010   Max.   :8.000   Max.   :4490
##                  NA's   :101     NA's   :300
##                                                          href
##  http://chicago.craigslist.org/chc/apa/5283655742.html:  1
##  http://chicago.craigslist.org/chc/apa/5285745015.html:  1
##  http://chicago.craigslist.org/chc/apa/5286189411.html:  1
##  http://chicago.craigslist.org/chc/apa/5286848681.html:  1
##  http://chicago.craigslist.org/chc/apa/5286911383.html:  1
##  http://chicago.craigslist.org/chc/apa/5286912144.html:  1
##  (Other)                                                :794
```

Exploratory plots

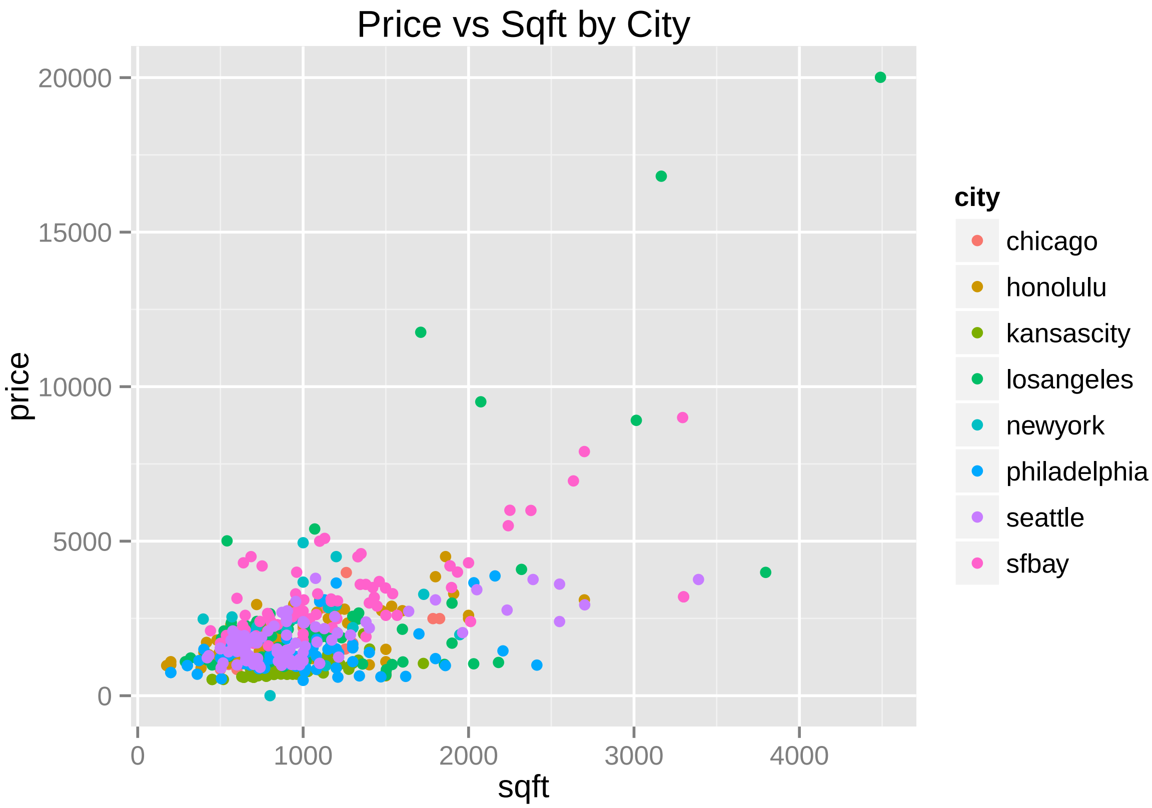
```
library(Hmisc)
library(dplyr)
library(ggplot2)
```

```
ggplot(citiesdf, aes(y=price, x=city)) + geom_violin() +
    labs(title="Price vs City")
```



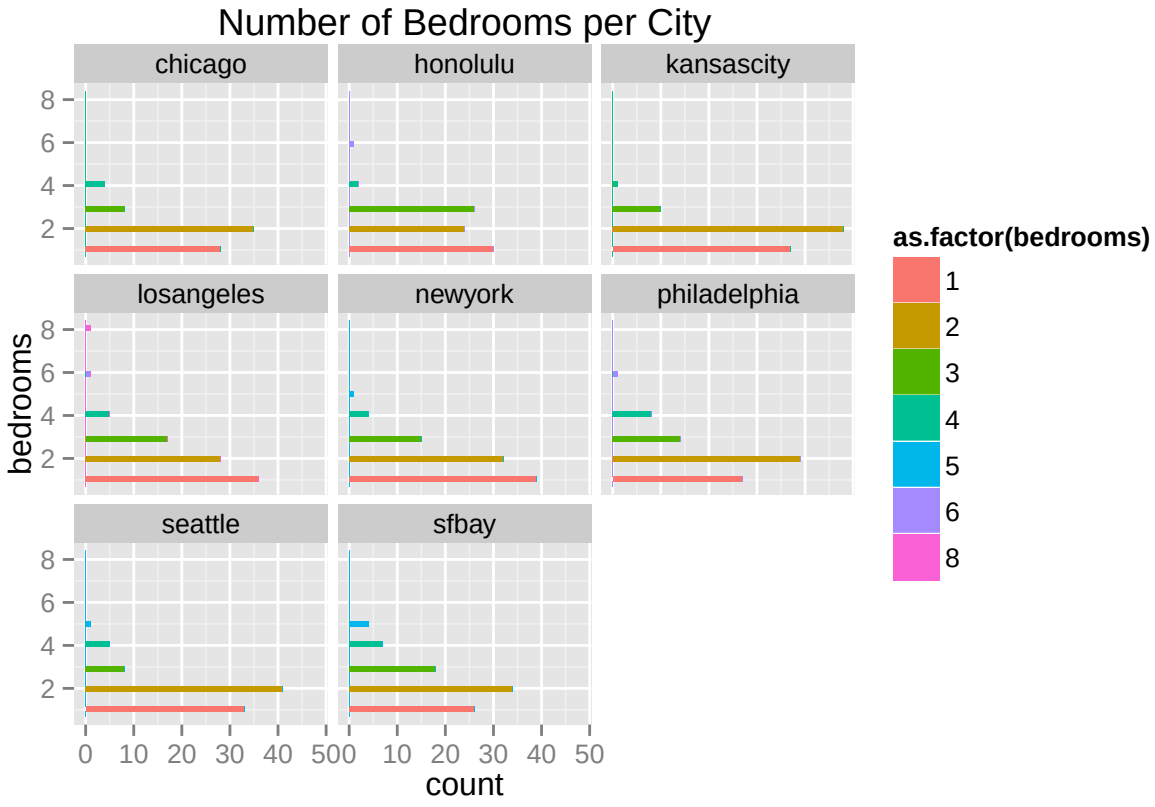

```
ggplot(citiesdf, aes(x=sqft, y=price, color=city)) + geom_point() +
    labs(title="Price vs Sqft by City")
```

```
## Warning: Removed 300 rows containing missing values (geom_point).
```




```r
ggplot(citiesdf, aes(x=bedrooms, fill=as.factor(bedrooms))) + geom_histogram() +
    coord_flip() + facet_wrap(~ city) +
    labs(title="Number of Bedrooms per City")
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

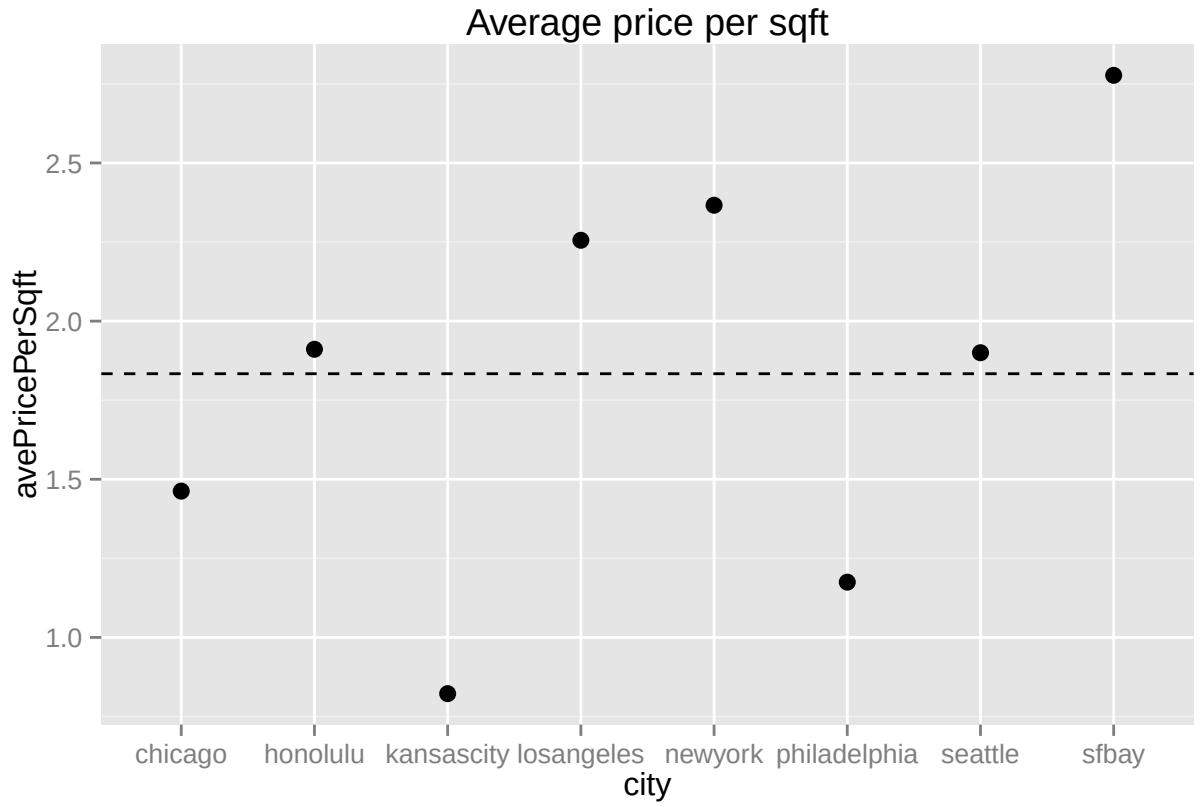


Since sqft is not always available, imputing missing values so that we can calculate price per square foot.

```
citiesdf.ar <- aregImpute(~ bedrooms + sqft, data=citiesdf, n.impute=1)
```

```
## Iteration 1
Iteration 2
Iteration 3
Iteration 4
```

```
# Create a new data frame with the imputed sqft.
citiesdf.imp <- citiesdf
citiesdf.imp$sqft[as.numeric(row.names(citiesdf.ar$imputed$sqft))] <-
    citiesdf.ar$imputed$sqft
summcity <- summarise(group_by(citiesdf.imp, city),
                      avePricePerSqft = sum(price) / sum(sqft))
ggplot(summcity, aes(x=city, y=avePricePerSqft)) + geom_point(size=3) +
    ggtitle('Average price per sqft') +
    geom_hline(aes(yintercept=mean(summcity$avePricePerSqft)), linetype="dashed")
```



Model training

`train_models.R` was used to train two models, which are as follows.

model formula	saved to file
price ~ city + bedrooms	data/pricemod.rds
sqft ~ city + bedrooms	data/sqftmod.rds

```
pricemod <- readRDS('data/pricemod.rds')
summary(pricemod)
```

```
##
## Call:
## lm(formula = price ~ city + bedrooms, data = citiesdf)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2575.5  -523.2  -117.4   274.1 16421.5
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       639.51     174.18   3.672 0.000260 ***
```

```
## cityhonolulu       297.52     203.51   1.462 0.144208
## citykansascity    -730.79     196.66  -3.716 0.000219 ***
## citylosangeles     955.74     200.67   4.763 2.33e-06 ***
## citynewyork        893.34     198.96   4.490 8.34e-06 ***
## cityphiladelphia  -272.08     200.32  -1.358 0.174840
## cityseattle        321.24     200.48   1.602 0.109543
## citysfbay         1691.29     200.80   8.423  < 2e-16 ***
## bedrooms           498.32      50.52   9.864  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1276 on 690 degrees of freedom
##   (101 observations deleted due to missingness)
## Multiple R-squared:  0.3414, Adjusted R-squared:  0.3337
## F-statistic:  44.7 on 8 and 690 DF,  p-value: < 2.2e-16
```

```
sqftmod <- readRDS('data/sqftmod.rds')
summary(sqftmod)
```

```
##
## Call:
## lm(formula = sqft ~ city + bedrooms, data = citiesdf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1585.32  -168.28   -17.71    98.83  2402.07
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        207.29      77.66   2.669  0.00788 **
## cityhonolulu      -157.92      81.81  -1.930  0.05422 .
## citykansascity     -31.11      81.90  -0.380  0.70420
## citylosangeles      63.25      81.79   0.773  0.43975
## citynewyork        -14.82     102.97  -0.144  0.88565
## cityphiladelphia    20.93      82.41   0.254  0.79964
## cityseattle        -24.93      79.35  -0.314  0.75355
## citysfbay           31.60      80.85   0.391  0.69609
## bedrooms           454.35      15.60  29.117  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 313 on 440 degrees of freedom
##   (351 observations deleted due to missingness)
## Multiple R-squared:  0.6791, Adjusted R-squared:  0.6732
## F-statistic: 116.4 on 8 and 440 DF,  p-value: < 2.2e-16
```

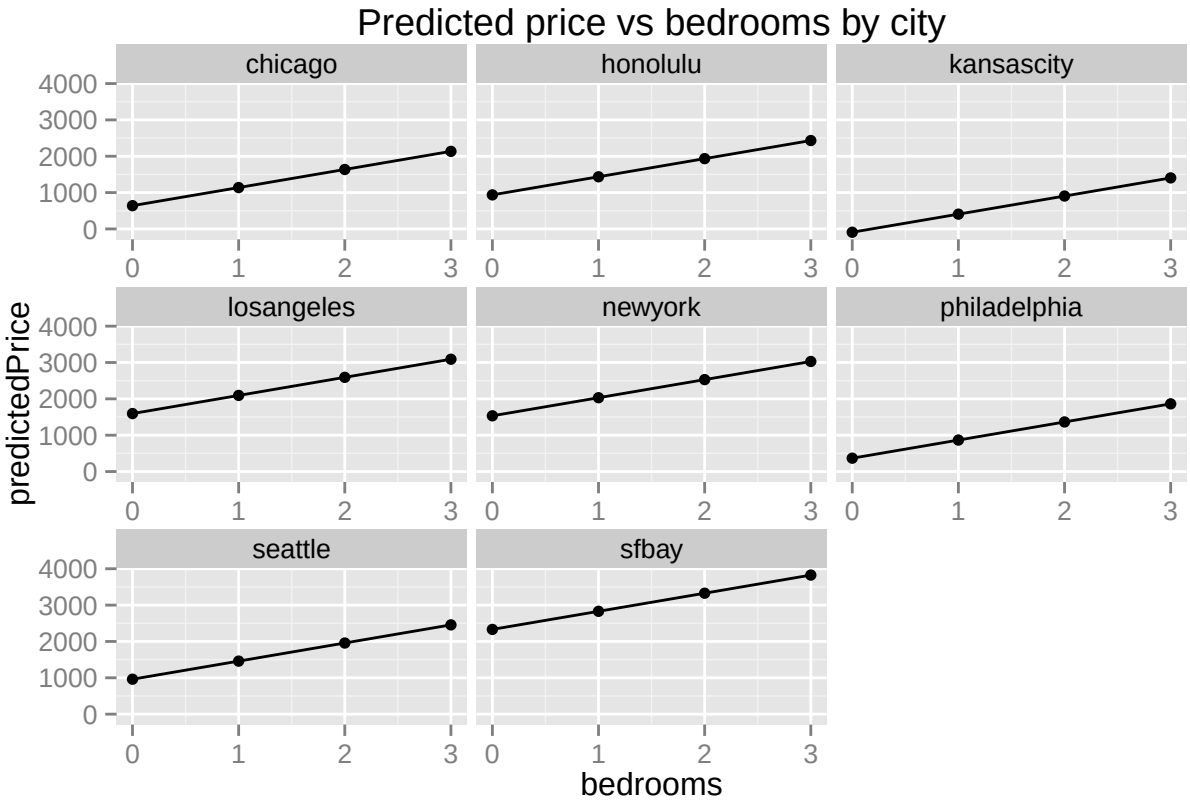
Check out some predictions

```
test <- data.frame(
    city = rep(c('chicago', 'honolulu', 'kansascity', 'losangeles', 'newyork',
                 'philadelphia', 'sfbay', 'seattle'), 4),
```

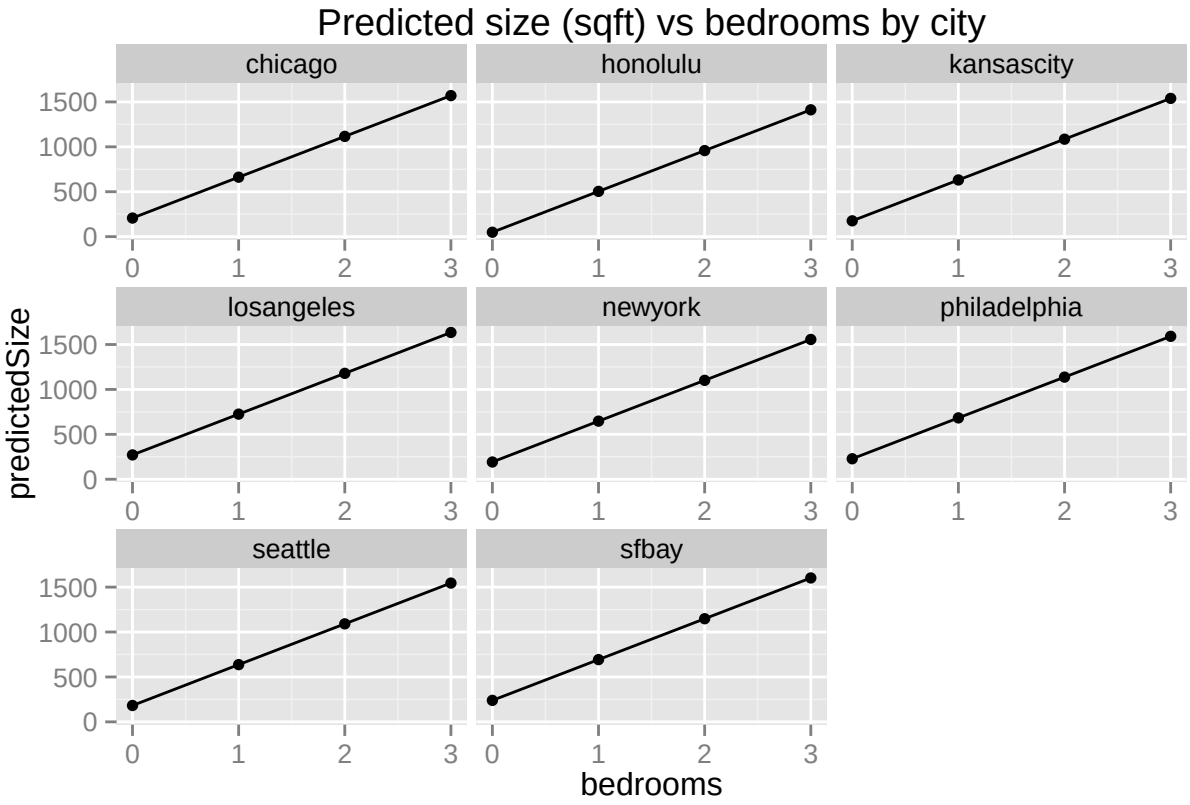
```
    bedrooms = rep(c(0,1,2,3), each=8))
test <- test[order(test$city, test$bedrooms),]

# Predict price (USD) and size (sqft).
predictedPrice <- predict(pricemod, newdata = test)
predictedSize <- predict(sqftmod, newdata = test)

pred <- cbind(test,
      predictedPrice = round(predictedPrice),
      predictedSize = round(predictedSize),
      predictedPricePerSqft = round(predictedPrice/predictedSize, 1))

print(pred, row.names=FALSE)
```

```
##          city bedrooms predictedPrice predictedSize predictedPricePerSqft
##       chicago        0            640           207                   3.1
##       chicago        1           1138           662                   1.7
##       chicago        2           1636          1116                   1.5
##       chicago        3           2134          1570                   1.4
##      honolulu        0            937            49                  19.0
##      honolulu        1           1435           504                   2.8
##      honolulu        2           1934           958                   2.0
##      honolulu        3           2432          1412                   1.7
##    kansascity        0            -91           176                  -0.5
##    kansascity        1            407           631                   0.6
##    kansascity        2            905          1085                   0.8
##    kansascity        3           1404          1539                   0.9
##    losangeles        0           1595           271                   5.9
##    losangeles        1           2094           725                   2.9
##    losangeles        2           2592          1179                   2.2
##    losangeles        3           3090          1634                   1.9
##       newyork        0           1533           192                   8.0
##       newyork        1           2031           647                   3.1
##       newyork        2           2529          1101                   2.3
##       newyork        3           3028          1556                   1.9
##  philadelphia        0            367           228                   1.6
##  philadelphia        1            866           683                   1.3
##  philadelphia        2           1364          1137                   1.2
##  philadelphia        3           1862          1591                   1.2
##       seattle        0            961           182                   5.3
##       seattle        1           1459           637                   2.3
##       seattle        2           1957          1091                   1.8
##       seattle        3           2456          1545                   1.6
##         sfbay        0           2331           239                   9.8
##         sfbay        1           2829           693                   4.1
##         sfbay        2           3327          1148                   2.9
##         sfbay        3           3826          1602                   2.4
```

```
ggplot(pred, aes(x=bedrooms, y=predictedPrice)) + geom_point() +
    geom_line() + facet_wrap( ~ city, scales="free_x") +
    ggtitle("Predicted price vs bedrooms by city")
```




```
ggplot(pred, aes(x=bedrooms, y=predictedSize)) + geom_point() +
    geom_line() + facet_wrap( ~ city, scales="free_x") +
    ggtitle("Predicted size (sqft) vs bedrooms by city")
```


Predicted size (sqft) vs bedrooms by city
chicago
honolulu
kansascity
losangeles
newyork
philadelphia
seattle
sfbay
predictedSize
bedrooms
0
500
1000
1500
0
1
2
3