Linear Regression in R

Connor Brubaker

This notebook is to accompany the lectures on linear regression and will demonstrate how to work with simple linear regression models in R.

Installing and Loading Packages

In order to access the data sets used in this package, **you need to first install the package that contains the data**. **You only ever need to run this command once even if you quit R and open it again.** Run the following:

```
install.packages("modeldata") # contains the housing data
```

Once this package has finished installing, you must load it into the current R session in order to use the data it contains. Run the following:

```
library(modeldata)
```

Loading Data

With the required package installed and loaded, we can now load the data we will use in these examples:

```
data("Sacramento")
```

You can refer to the data by the name above. For example, to view the first few rows, you can run

```
head(Sacramento)
```

```
# A tibble: 6 x 9
  city       zip     beds baths  sqft type        price latitude longitude
  <fct>      <fct>  <int> <dbl> <int> <fct>       <int>    <dbl>     <dbl>
1 SACRAMENTO z95838     2     1   836 Residential 59222     38.6     -121.
2 SACRAMENTO z95823     3     1  1167 Residential 68212     38.5     -121.
3 SACRAMENTO z95815     2     1   796 Residential 68880     38.6     -121.
4 SACRAMENTO z95815     2     1   852 Residential 69307     38.6     -121.
5 SACRAMENTO z95824     2     1   797 Residential 81900     38.5     -121.
6 SACRAMENTO z95841     3     1  1122 Condo       89921     38.7     -121.
```

About the Data

The `Sacramento` data set contains information on 932 houses for sale in the Sacramento, CA area obtained from the SpatialKey website. The important variables to know are

- `sqft` - the square footage of the house
- `price` - the price of the house

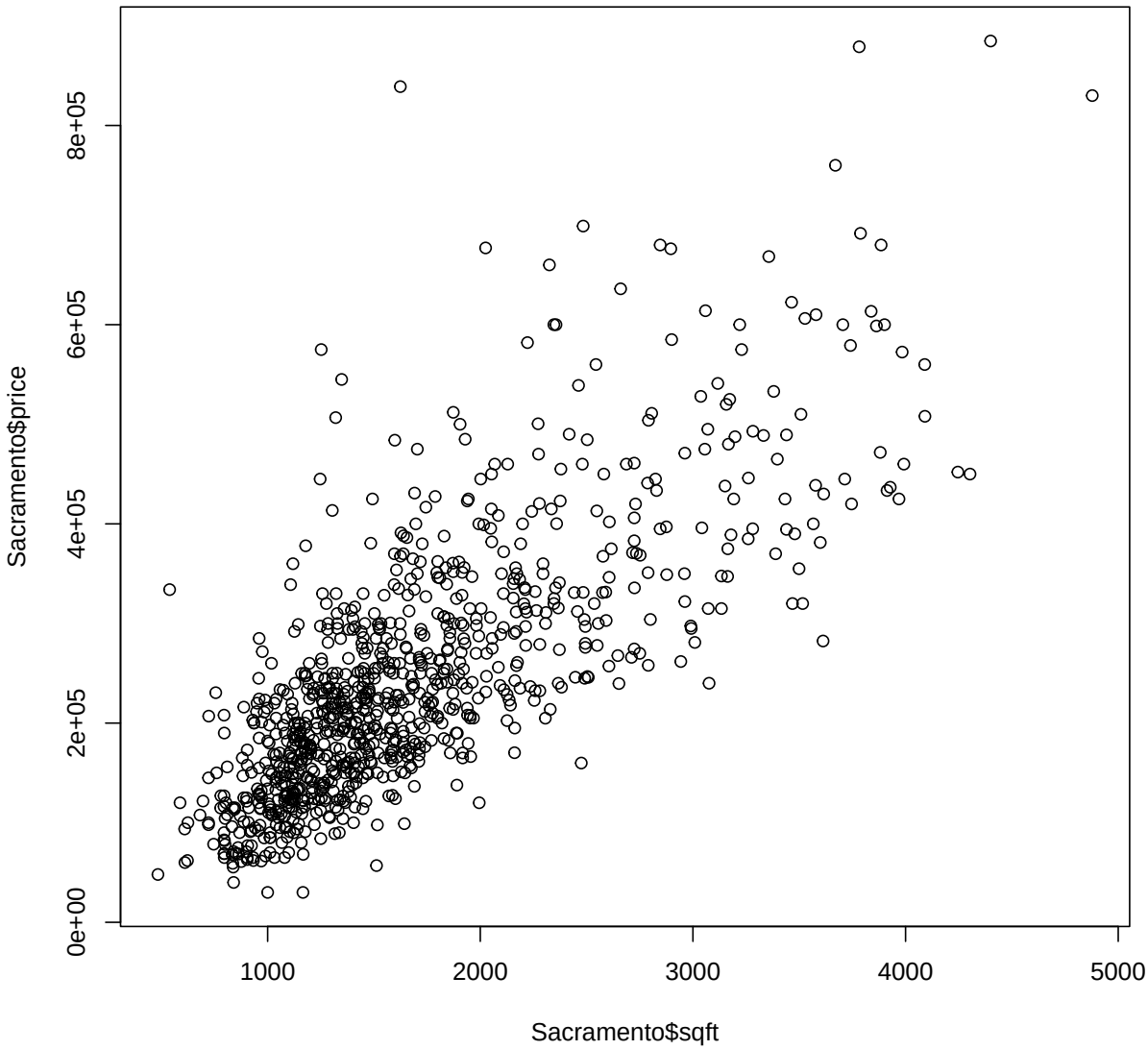
We will build a linear regression model to understand the relationship between price and square footage (which you can imagine is a positive relationship - the bigger the house, the more expensive).

Sacramento Data

I will first demonstrate how to do everything I showed in class with the Sacramento housing data. You won't need to know how to do everything here for the exam, but you will need to know how to read output from the model fit.

First, let's create a scatter plot of price against square footage to understand the functional relationship between these two variables.

```
plot(x = Sacramento$sqft, y = Sacramento$price)
```



Recall that the first assumption of the simple linear model is that the relationship between X and Y is linear. You can see from the plot that the relationship appears to be approximately linear making this data a good candidate for using a linear regression model.

To obtain least squares estimates, use the `lm` function as done below.

```
model <- lm(price ~ sqft, data = Sacramento)
```

The `y ~ x` is called a *formula* in R and tells R to regress `y` onto `x`, in this case, to regress `price` onto `sqft`. To get the output you need to understand how to read, use the `summary` function.

```
summary(model)
```

```
Call:
lm(formula = price ~ sqft, data = Sacramento)

Residuals:
    Min      1Q  Median      3Q     Max
-231889  -54717  -11822   38993  600141

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 13859.393   6948.714   1.995   0.0464 *
sqft          138.546      3.796  36.495   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 84130 on 930 degrees of freedom
Multiple R-squared:  0.5888,    Adjusted R-squared:  0.5884
F-statistic:  1332 on 1 and 930 DF,  p-value: < 2.2e-16
```

Refer to the notes from class to see the different components of this output.

To get confidence and prediction intervals, use the `predict` function. Remember that confidence intervals are for the unknown parameter (the value of the regression line at some given value of $X = X^*$ which is the expected value of the response when the predictor equals X^*. As an example, let's get a confidence interval for the average price of a home with 2,000 square feet.

```
conf <- predict(model, newdata = data.frame(sqft = 2000),
                interval = "confidence", level = 0.95)
print(conf)
```

```
       fit      lwr      upr
1 290952.3 285043.1 296861.5
```

The `interval = "confidence"` argument tells R to get a confidence interval (as opposed to a prediction interval) and the `level = 0.95` argument says to compute a 95% confidence interval. The output above says that the model predicts the **average** price of a home with 2000 square feet to be $290,952.30 (this might be a little different from what I put on the slides because of rounding). The lower and upper bounds of the confidence interval for the average

price of a 2000 square foot home in Sacramento, CA are $285,043.10 and $296,861.50. That is, we are 95% confident the average price of all homes for sale in Sacramento, CA with 2000 square feet is between $285,043.10 and $296,861.50.

To get a prediction interval for the *actual* value of a randomly selected house for sale in Sacramento with 2000 square feet, just change to 'interval = "prediction" from above.

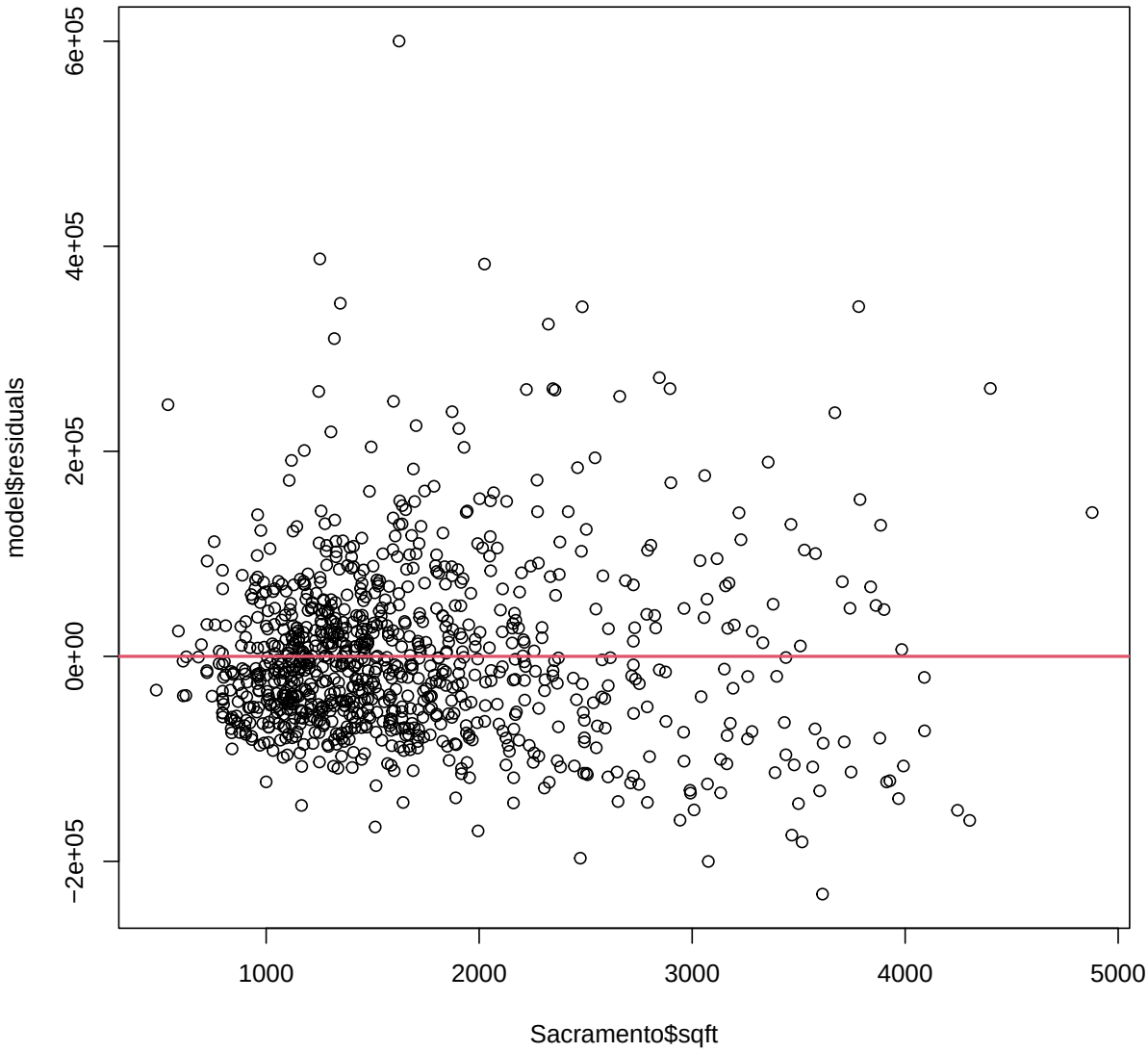
```
pred <- predict(model, newdata = data.frame(sqft = 2000),
                interval = "prediction", level = 0.95)
print(pred)
```

```
       fit      lwr      upr
1 290952.3 125748.3 456156.3
```

Notice the predicted value of $290,952.30 doesn't change, but the bounds do and the bounds give a wider interval than before (which is expected since a prediction interval accounts for the extra fluctuation of realized values around the regression line). This says that we are 95% confident a randomly selected home for sale in Sacramento with 2000 square feet will have a price between $125,748.30 and $456,156.3.

For model validation, one of the first things to look at is a plot of the residuals against the x-values. This allows us to visually check for violations of the linearity and homoscedasticity assumptions.

```
plot(x = Sacramento$sqft, y = model$residuals)
abline(h = 0, lwd = 2, col = 2) # add horizontal line at 0
```



The residual plot for the most part shows an even scatter around zero and no discernible pattern so I would argue that these assumptions are likely satisfied.