Housing Data

Amelia Farrell

October 18th 2021

Background: Real estate transactions recorded from 1964 to 2016

a.i. Transformations

If you recall, we already took a closer look at this data back on week 4 and 4 (Exercise 4.2 and 5.2). We identified some data that would impact any predictions we hope to make. These include the "homes" with 0 bathrooms and 0 bedrooms. We would like to exclude these from this excerise as well. We would only like to look at homes that are move in ready and not cabins, plots of land, or simply have missing data. In order to drop these from our data set we can use the subset function to remove any line items with more than 0 bathrooms and bedrooms. Lets first re-check that this data exists in our data set (after transforming it from a list to a dataframe) (note that we are using the has_element function within the purrr package to check this data)

```
housing <- dfhousing <- data.frame(housing)
dfhousing$bedrooms %>% has_element(0)
```

```
## [1] FALSE
```

```
dfhousing$bath_full_count %>% has_element(0)
```

```
## [1] FALSE
```

As we can see above, the data does include line items with 0 bathrooms and 0 bedrooms.

We can exude these using the subset function and check that they have been removed.

```
dfhousing2 <- subset(dfhousing, bedrooms!= 0 & bath_full_count!= 0)
dfhousing2$bedrooms %>% has_element(0)
```

```
## [1] FALSE
```

```
dfhousing2$bath_full_count %>% has_element(0)
```

```
## [1] FALSE
```

Next we like to add price per square foot. Note that this will not include the square footage of the lot, but is still an important peice of information when considering a home. We will create a price per square foot variable and add it to the housing data frame below.

```
piceperfoot <- (dfhousing2$Sale.Price/dfhousing2$square_feet_total_living)
cbind(dfhousing2, piceperfoot)
```

b.i. Transformations explained

Lets summarize what we did above; - Create a dataframe to hold our housing data set. This is will allow us to set restrictions such as, not using the same name for two variable, keeping all elements as vectors, and ensuring at all columns are named. - Checking for line items with 0 bathrooms and 0 bedrooms. - Removing line items with 0 bathrooms and 0 bedrooms (reason for doing so explained above). - Creating price per square foot variable and adding it to the data frame.

b.ii. Create two variables (Linear Regression)

We will first fit a linear model using the `Square Foot of Lot` variable as the predictor and `Sale Price` as the outcome.

```
lotSF_lm <- lm(Sale.Price ~ sq_ft_lot, data = dfhousing2)
```

Then fit a linear model with a couple more predictors. Adding `year renovated` (this may impact the price more than the year it was built since renovations/remolding can greatly impact home value), `Square Feet Living` (the total square footage of the home is correlated to the sale price), `Full Bath Count` (the number of full bathrooms is also correlated to home price but not necessarily correlated to total square feet, making it a great additional predictor) as additional the predictors to `Sale Price`.

```
lotSF_lm2 <- lm(Sale.Price ~ sq_ft_lot + year_renovated + square_feet_total_living + bath_full_count, da
```

b.iii. Execute a summary() function

Lets now compare our two models for predicting home sale price with the summary function.

```
##
## Call:
## lm(formula = Sale.Price ~ sq_ft_lot, data = dfhousing2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2046056  -194710   -63503    91200  3735135
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.417e+05  3.807e+03  168.58   <2e-16 ***
## sq_ft_lot   8.694e-01  6.277e-02   13.85   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 401600 on 12809 degrees of freedom
## Multiple R-squared:  0.01476,    Adjusted R-squared:  0.01468
## F-statistic: 191.9 on 1 and 12809 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = Sale.Price ~ sq_ft_lot + year_renovated + square_feet_total_living +
##     bath_full_count, data = dfhousing2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1925674  -119387   -39623    44816  3780658
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.458e+05  1.032e+04  14.122  < 2e-16 ***
## sq_ft_lot                1.328e-01  5.803e-02   2.289   0.0221 *
## year_renovated           1.717e+01  1.404e+01   1.223   0.2212
## square_feet_total_living 1.702e+02  3.890e+00  43.768  < 2e-16 ***
## bath_full_count          4.382e+04  5.820e+03   7.529 5.44e-14 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 359100 on 12806 degrees of freedom
## Multiple R-squared:  0.2124, Adjusted R-squared:  0.2122
## F-statistic: 863.5 on 4 and 12806 DF,  p-value: < 2.2e-16
```

b.iv.

b.v.

b.vi.

b.vii.

b.viii.

b.ix.

b.xii.

b.xiii.

b.xiv.

b.xv.

References

Field, A., J. Miles, and Z. Field. 2012. Discovering Statistics Using R. SAGE Publications. https://books.google.com/books?id=wd2K2zC3swIC.

Lander, J. P. 2014. R for Everyone: Advanced Analytics and Graphics. Addison-Wesley Data and Analytics Series. Addison-Wesley. https://books.google.com/books?id=3eBVAgAAQBAJ.