Linear Regression

Jeremy Oldfather

November 2, 2016

```
# see: https://github.com/theoldfather/DC_Properties
sales<-read_csv("~/Projects/DC_Properties/data/sales_clean.csv")
features<-read_csv("~/Projects/DC_Properties/data/features_clean.csv")
details<-read_csv("~/Projects/DC_Properties/data/details_clean.csv")

d<-sales %>%
  filter(sale_price>0) %>%
  left_join(features,by="ssl") %>%
  filter(!is.na(living_sqft) & living_sqft>0) %>%
  select(sale_price,living_sqft)

fit_line<-function(y,x){
  m <- (y[2]-y[1])/(x[2]-x[1])
  b <- y[1] - m*x[1]
  return(c(b,m))
}

fit_simple_lm<-function(y,x){
  n<-length(y)
  syx<-sum(y*x)
  sy<-sum(y)
  sx<-sum(x)
  sxx<-sum(x^2)
  b1<- (syx - sy*sx/n)/(sxx - sx^2/n)
  b0<- (sy - b1*sx)/n
  return(c(b0,b1))
}

# subset some data for a regression
# and make sure both sale_price and living_sqft are log-scaled
data.model<-sales %>%
  filter(sale_price>0) %>%
  left_join(features,by="ssl") %>%
  left_join(select(details,-sale_price,-neighborhood),by="ssl") %>%
  filter(!is.na(living_sqft) & living_sqft>0) %>%
  mutate(log_sale_price=log(sale_price),
         log_living_sqft=log(living_sqft))
```

```
fit<-lm(log_sale_price ~ log_living_sqft, data=data.model)
```

Multiple Linear Regression

When we regress y on more than one independent variable, say $(x_1, x_2, ..., x_k)$ rather than simply (x_1), the regression framework is referred to as **multiple linear regression**.

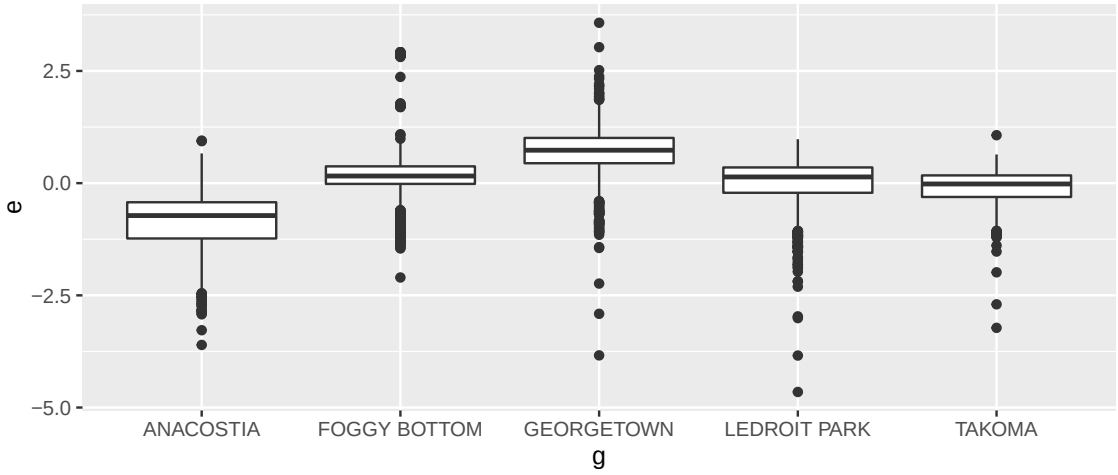
What might motivate us to introduce multiple regressors? One reason might be that we have reason to suspect **omitted variable bias**. In order for our model to be subject to omitted variable bias, we must have both of the following problems:

$E(y|x_1, x_2) \neq E(y|x_1)$: Meaning our expectation about the dependent variable y changes when we include the additional independent variable. If $E(y|x_1, x_2) = E(y|x_1)$, we would say that y is conditionally independent of x_2—meaning either x_2 is completely irrevelant to predicting y, or that any information contained in x_2 is already contained in x_1.

$Cor(x_1, x_2) \neq 0$: Meaning the additional variable x_2 must be correlated with a current dependent variable x_1.

For example, when we plot the distribution of residuals from our previous model by neighborhood, we can see that the variable **neighborhood** meets the first critera for omitted variable bias since for Anacostia we would consistently overstate the impact of a change in living area on the change in sales price.

```
selected.nhoods<-c("FOGGY BOTTOM","ANACOSTIA","LEDROIT PARK","TAKOMA","GEORGETOWN")
data.frame(e=resid(fit),g=data.model$neighborhood) %>%
  filter(g %in% selected.nhoods) %>%
  ggplot(aes(y=e,x=g)) + geom_boxplot()
```



To check the second criteria, that $Cor(living\_area, neighborhood) \neq 0$, we can plug a dummy variable for each neighborhood into R's **cor()** function.

```
cor(data.model$log_living_sqft,data.model$neighborhood=="ANACOSTIA")

## [1] 0.02040842

cor(data.model$log_living_sqft,data.model$neighborhood=="FOGGY BOTTOM")

## [1] -0.1058928

cor(data.model$log_living_sqft,data.model$neighborhood=="GEORGETOWN")

## [1] 0.06697943

cor(data.model$log_living_sqft,data.model$neighborhood=="LEDROIT PARK")

## [1] 0.03272786

cor(data.model$log_living_sqft,data.model$neighborhood=="TAKOMA")

## [1] 0.0146363
```

Anacostia, Ledroit Park, and Takoma are either weakly or not all correlated with living area, but Foggy Bottom is negatively correlated and Georgetown is weakly positively correlated.

To see the connection between including these neighborhoods in the regression, I will create a new variable that sets all other neighborhoods to "other". I put "aa" at the beginning because R treats the first element of a factor variable as the base value for the regression. So by doing this, we can see how the neighborhoods we analyzed above affect our regression in terms of all other DC neighborhoods.

```
fit2<-data.model %>%
  mutate(nhoods5=ifelse(neighborhood %in% selected.nhoods,
                        neighborhood,
                        "aa_other")
         ) %>%
  lm(log_sale_price ~ log_living_sqft + factor(nhoods5),data=.)
summary(fit2)

##
## Call:
## lm(formula = log_sale_price ~ log_living_sqft + factor(nhoods5),
##     data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.6024  -0.3463   0.1282   0.4260   6.5927
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  7.913192   0.033502 236.204  < 2e-16 ***
## log_living_sqft              0.699649   0.004723 148.133  < 2e-16 ***
## factor(nhoods5)ANACOSTIA    -0.832172   0.023094 -36.034  < 2e-16 ***
```

```
## factor(nhoods5)FOGGY BOTTOM  0.192594   0.024741   7.784 7.09e-15 ***
## factor(nhoods5)GEORGETOWN    0.715779   0.015717  45.543  < 2e-16 ***
## factor(nhoods5)LEDROIT PARK -0.007375   0.022117  -0.333 0.738790
## factor(nhoods5)TAKOMA       -0.129247   0.036654  -3.526 0.000422 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7052 on 85290 degrees of freedom
## Multiple R-squared:  0.2355,Adjusted R-squared:  0.2354
## F-statistic:  4378 on 6 and 85290 DF,  p-value: < 2.2e-16
```

Comparing the coefficients for each neighborhood with the distribution of their residuals in the boxplot above, we can see the pattern. The coefficient for a dummy shifts our fitted line up or down by the amount of the coefficient. For example, when all the dummies are turned off (equal zero),

$$\hat{y}_i = 7.913192 + 0.699649 x_i$$

But when Anacostia equals 1,

$$\begin{aligned} \hat{y}_i &= 7.913192 + 0.699649 x_i - 0.832172(1) \\ &= 7.08102 + 0.699649 x_i \end{aligned}$$

Now I will add dummies for the rest of the DC neighborhoods and run the regression again.

```
fit3<-lm(log_sale_price ~ log_living_sqft + factor(neighborhood),data=data.model)
coef(fit3)[1:2]

##     (Intercept) log_living_sqft
##       7.634167        0.716211

summary(resid(fit3))

##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -13.33000  -0.22140   0.07746   0.00000   0.29570   6.22800
```