Introduction to R, Part III

Rebecca C. Steorts, STA 360

Agenda

- Example of housing in PA
- Review of linear models
- Using variables and names versus hard coding

Example: Price of houses in PA

Census data for California and Pennsylvania on housing prices, by Census "tract"

```
# read in data
calif_penn <-
  read.csv("http://www2.stat.duke.edu/~rcs46/modern_bayes17/data/calif_penn_2011.csv")
# inspect the variables associated with this dataset
names(calif_penn)
```

```
##  [1] "X"                           "GEO.id2"
##  [3] "STATEFP"                     "COUNTYFP"
##  [5] "TRACTCE"                     "POPULATION"
##  [7] "LATITUDE"                    "LONGITUDE"
##  [9] "GEO.display.label"           "Median_house_value"
## [11] "Total_units"                 "Vacant_units"
## [13] "Median_rooms"                "Mean_household_size_owners"
## [15] "Mean_household_size_renters" "Built_2005_or_later"
## [17] "Built_2000_to_2004"          "Built_1990s"
## [19] "Built_1980s"                 "Built_1970s"
## [21] "Built_1960s"                 "Built_1950s"
## [23] "Built_1940s"                 "Built_1939_or_earlier"
## [25] "Bedrooms_0"                  "Bedrooms_1"
## [27] "Bedrooms_2"                  "Bedrooms_3"
## [29] "Bedrooms_4"                  "Bedrooms_5_or_more"
## [31] "Owners"                      "Renters"
## [33] "Median_household_income"     "Mean_household_income"
```

```
# STATEFP is the FIPS code, where there is one for each state. 42 belongs to PA.
# 6 corresponds to CA.
# https://en.wikipedia.org/wiki/Federal_Information_Processing_Standard_state_code#FIPS_state_codes
penn <- calif_penn[calif_penn[,"STATEFP"]==42,]
# fitting a simple linear model
coefficients(lm(Median_house_value ~ Median_household_income, data=penn))
```

```
##             (Intercept) Median_household_income
##           -26206.564325                3.651256
```

```
summary(lm(Median_house_value ~ Median_household_income, data=penn))
```

```
##
## Call:
## lm(formula = Median_house_value ~ Median_household_income, data = penn)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -207567  -36051  -11257   21146  560715
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -2.621e+04  2.696e+03  -9.721   <2e-16 ***
## Median_household_income   3.651e+00  4.516e-02  80.851   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63000 on 3166 degrees of freedom
##   (50 observations deleted due to missingness)
## Multiple R-squared:  0.6737, Adjusted R-squared:  0.6736
## F-statistic:  6537 on 1 and 3166 DF,  p-value: < 2.2e-16
```

Goal: fit a simple linear model, and predict the median house price (y) from median household income (x). Before doing this, let's investigate the census tracts that correspond to Allegheny county (24—425).

Tract 24 has a median income of $14,719; actual median house value is $34,100 — is that above or below the observed median?

```
34100 < -26206.564 + 3.651*14719
```

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

Tract 25 has income $48,102 and house price $155,900

```
155900 < -26206.564 + 3.651*48102
```

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

What about tract 26?

We *could* just keep plugging in numbers like this, but that's

- boring and repetitive
- error-prone
- confusing (what *are* these numbers?)

Using variables and names

```
penn.coefs <- coefficients(lm(Median_house_value ~ Median_household_income, data=penn))
penn.coefs
```

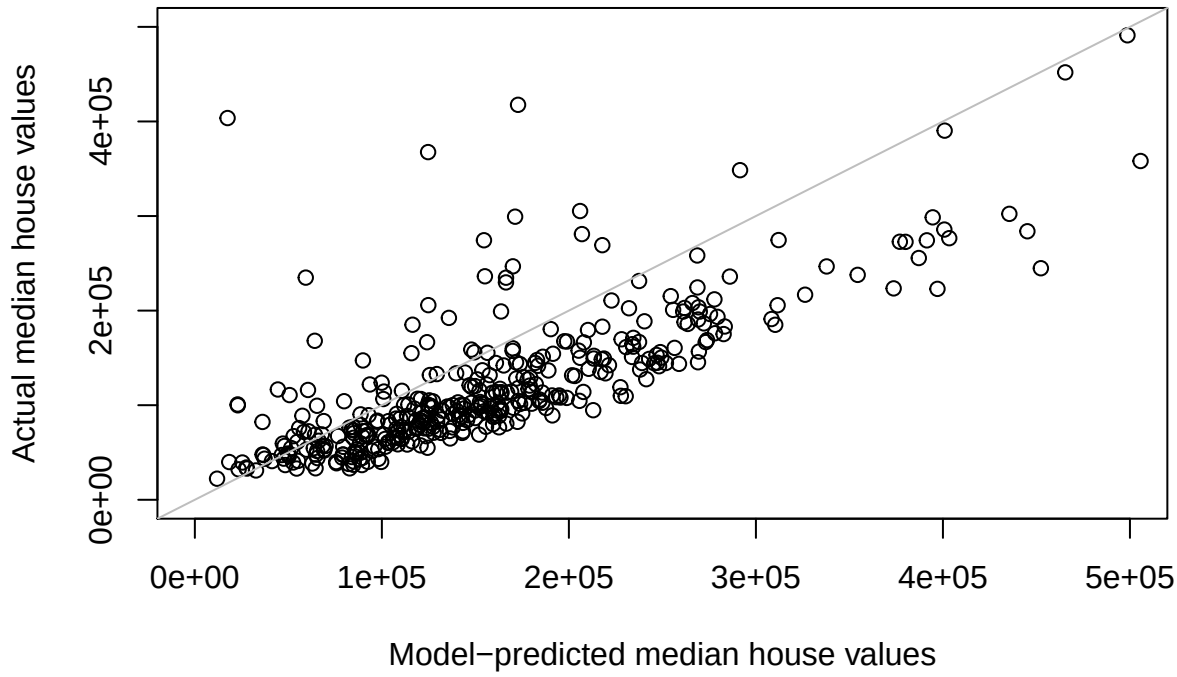
```
##             (Intercept) Median_household_income
##           -26206.564325                3.651256
```

```
allegheny.rows <- 24:425
allegheny.medinc <- penn[allegheny.rows,"Median_household_income"]
allegheny.values <- penn[allegheny.rows,"Median_house_value"]
```

```
allegheny.fitted <- penn.coefs["(Intercept)"]+
  penn.coefs["Median_household_income"]*allegheny.medinc
```

Actual median house values versus Predicted Median House Values

```
plot(x=allegheny.fitted, y=allegheny.values,
     xlab="Model-predicted median house values",
     ylab="Actual median house values",
     xlim=c(0,5e5),ylim=c(0,5e5))
abline(a=0,b=1,col="grey")
```



Summary

- We have reviewed simple linear models.
- We used variable and naming schemes.
- We reviewed how to plot.
- We have looked at a real application from the Census in the state of PA, where we avoided hard coding for easy automation.