Home Valuation[1]

August 2016

Home Values for Property Tax Assessment

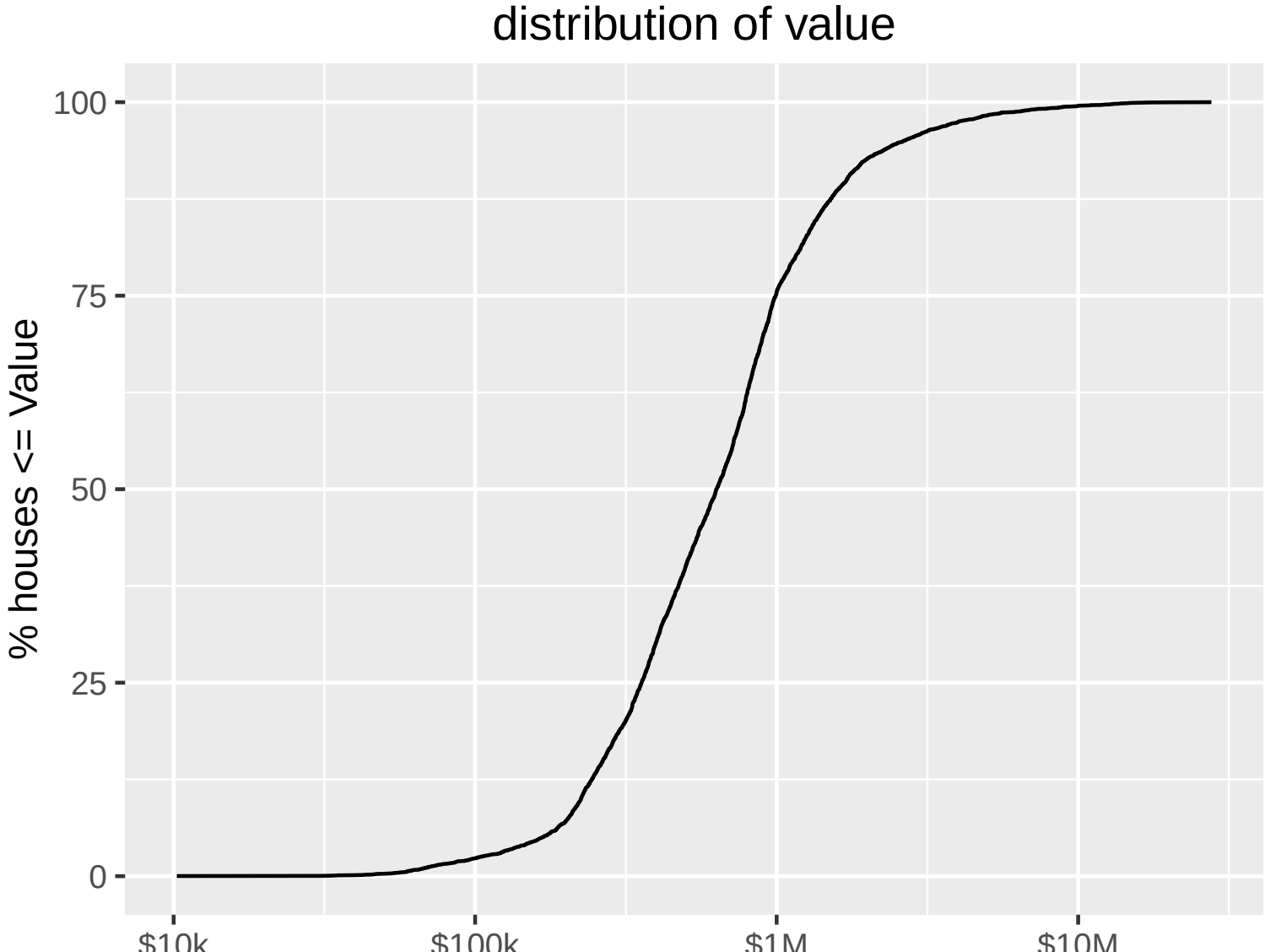
- Look at real data: property values in San Francisco circa 2014
  - Data courtesy city of San Francisco
  - Will use ggplot2 and car packages
- Have value, sqft, bedrooms, bathrooms, neighborhood, year_built...
- Note that California Prop13 means values out-of-wack for long-owned homes

	id	total_value	neighborhood	pclass	p13_date	sqft	bathrooms	bedrooms	all
1	1998283	936968	Noe Valley	Dwelling	2010-08-01	1017	1.00	2	
2	973406	289108	Parkside	Dwelling	1999-02-01	1614	2.00	3	
3	259382	6879410	Pacific Heights	Dwelling	2014-01-29	4641	5.00	4	

- QUESTION: Want to estimate property value (response) from other (predictor) columns

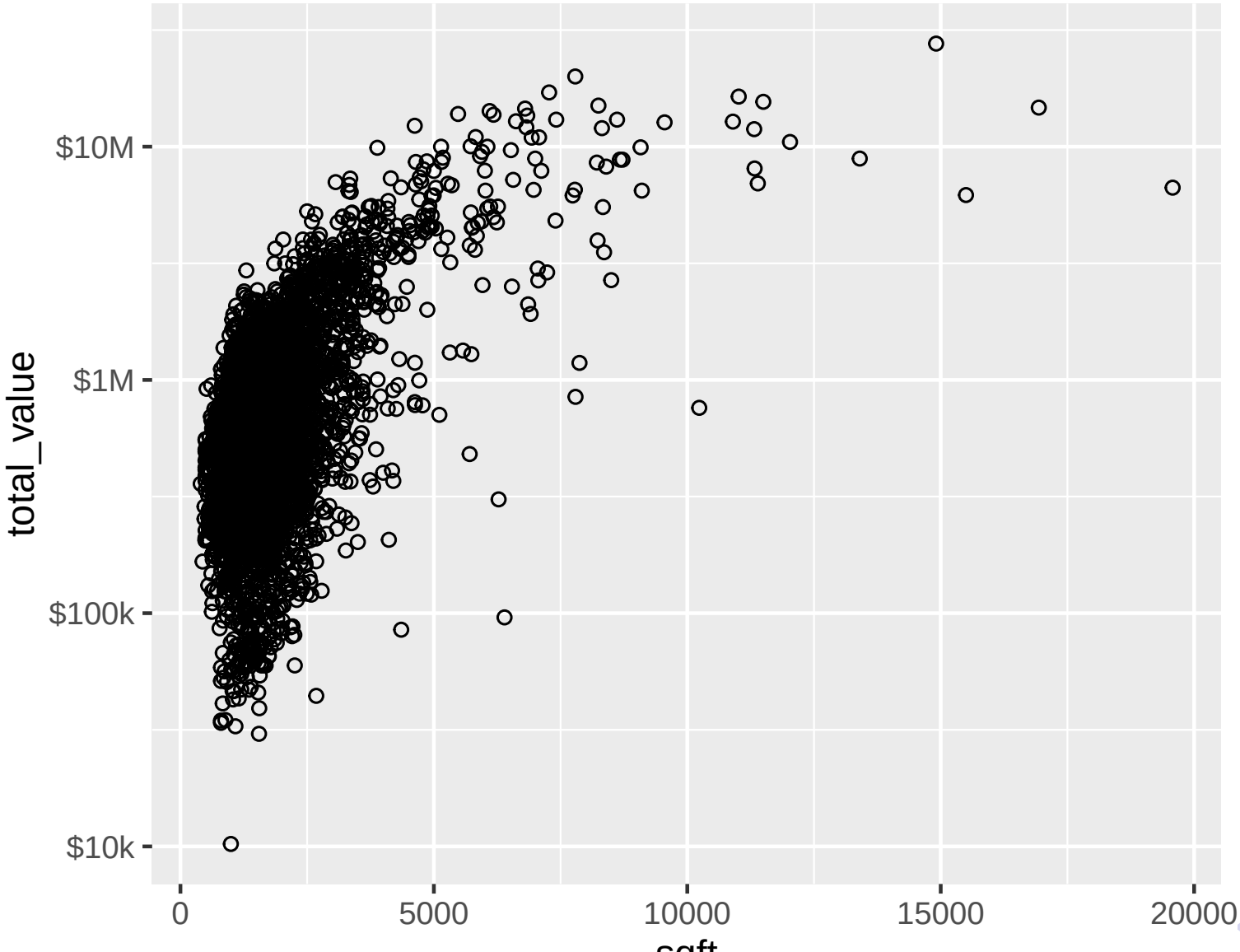
Start by Looking at Response

- Always start by looking at data
- Values run from $<$ \$10k to $>$ \$30M, and most are \$100k to \$1M.

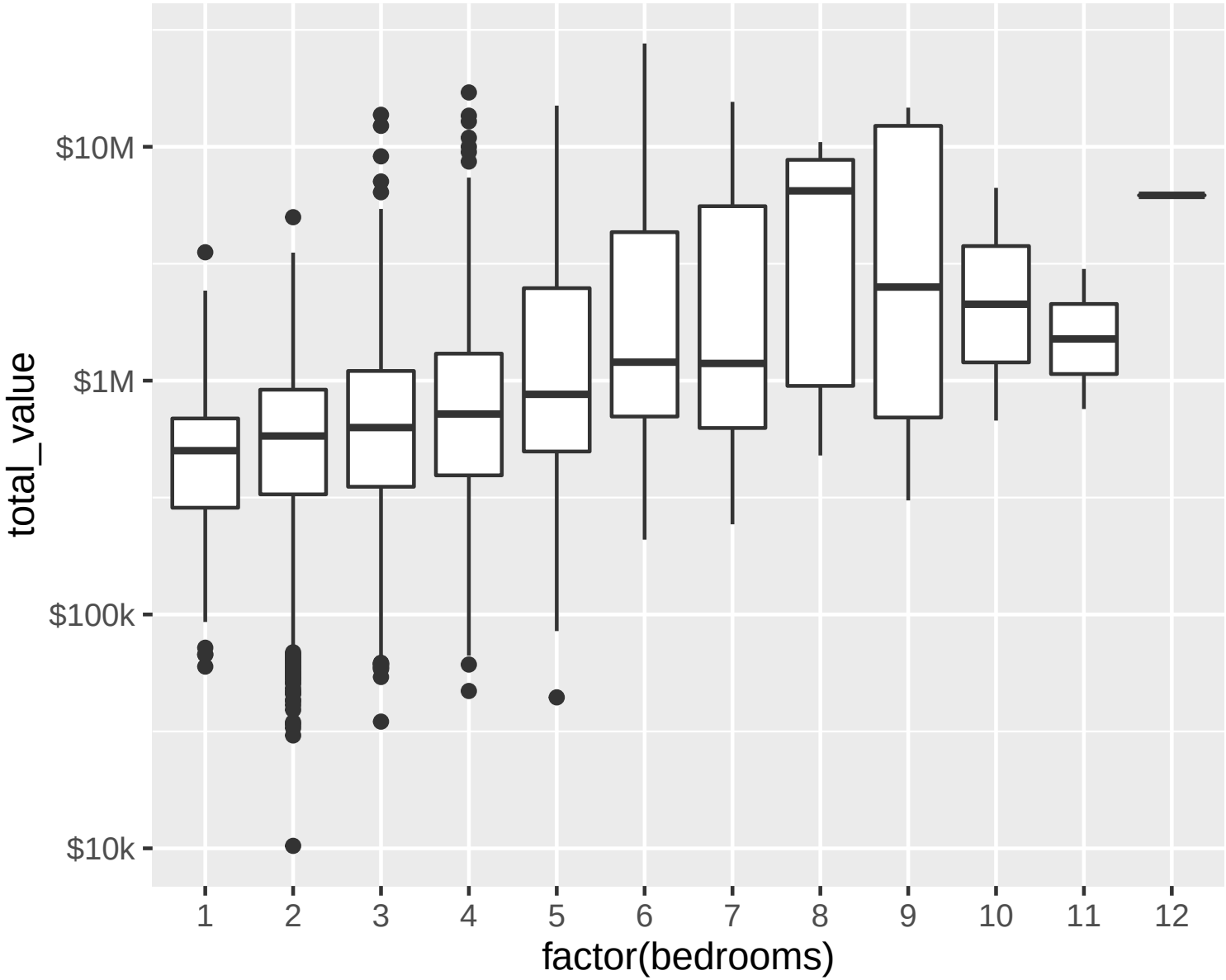


Look at Response vs Predictors

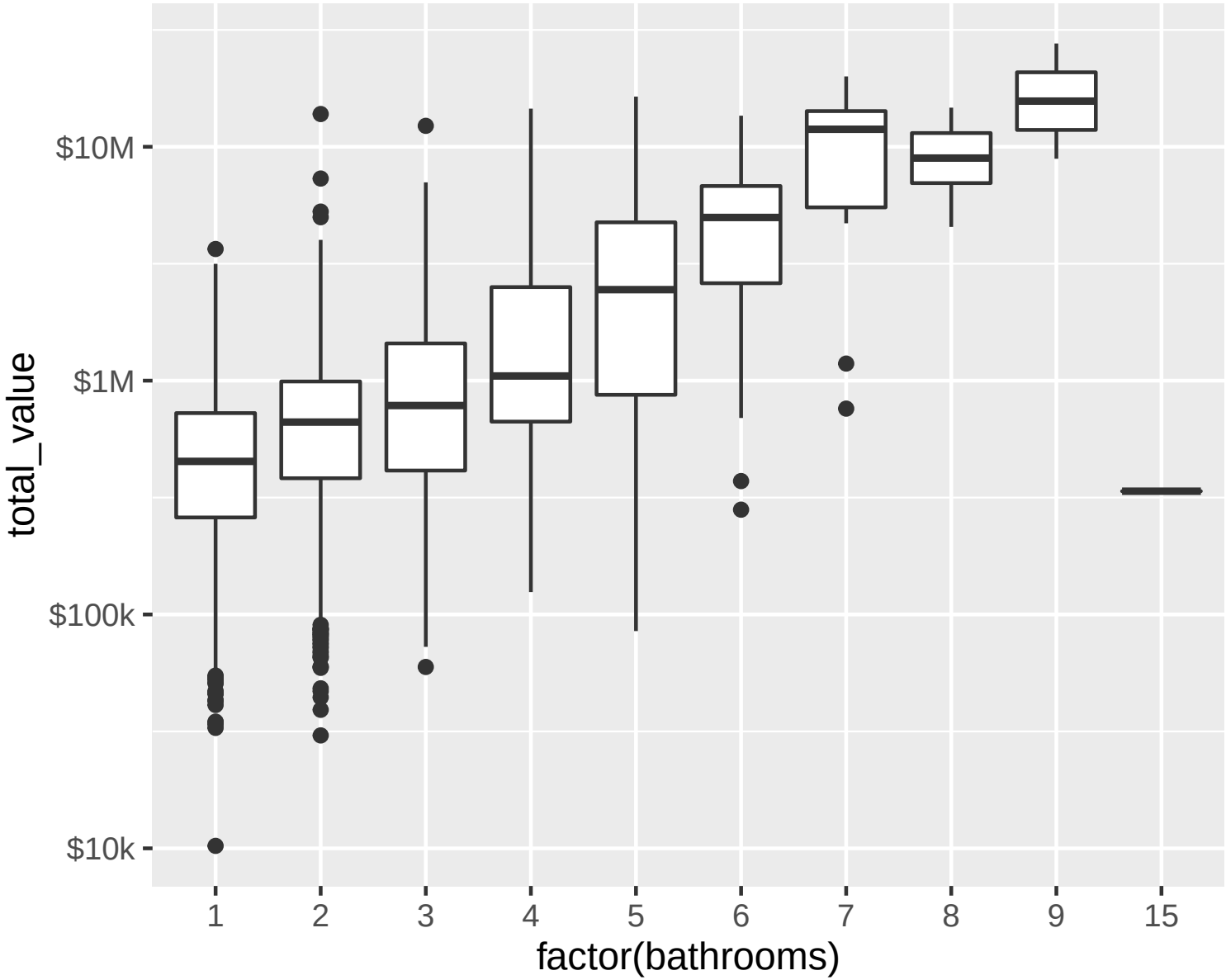
- Theory: key predictors are sqft, bedrooms, bathrooms, neighborhood



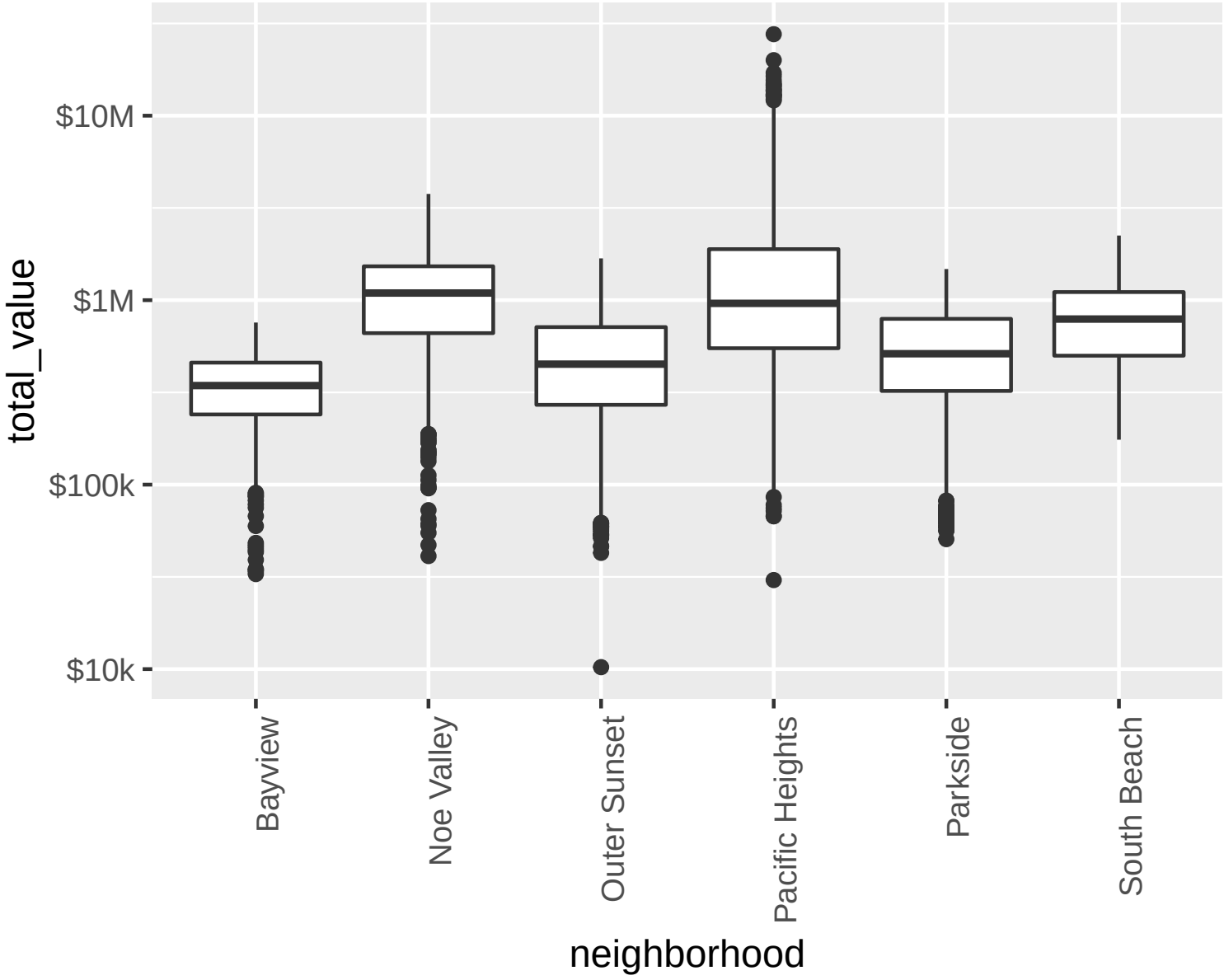
bedrooms


$10M
$1M
$100k
$10k
total_value
1
2
3
4
5
6
7
8
9
10
11
12
factor(bedrooms)

bathrooms

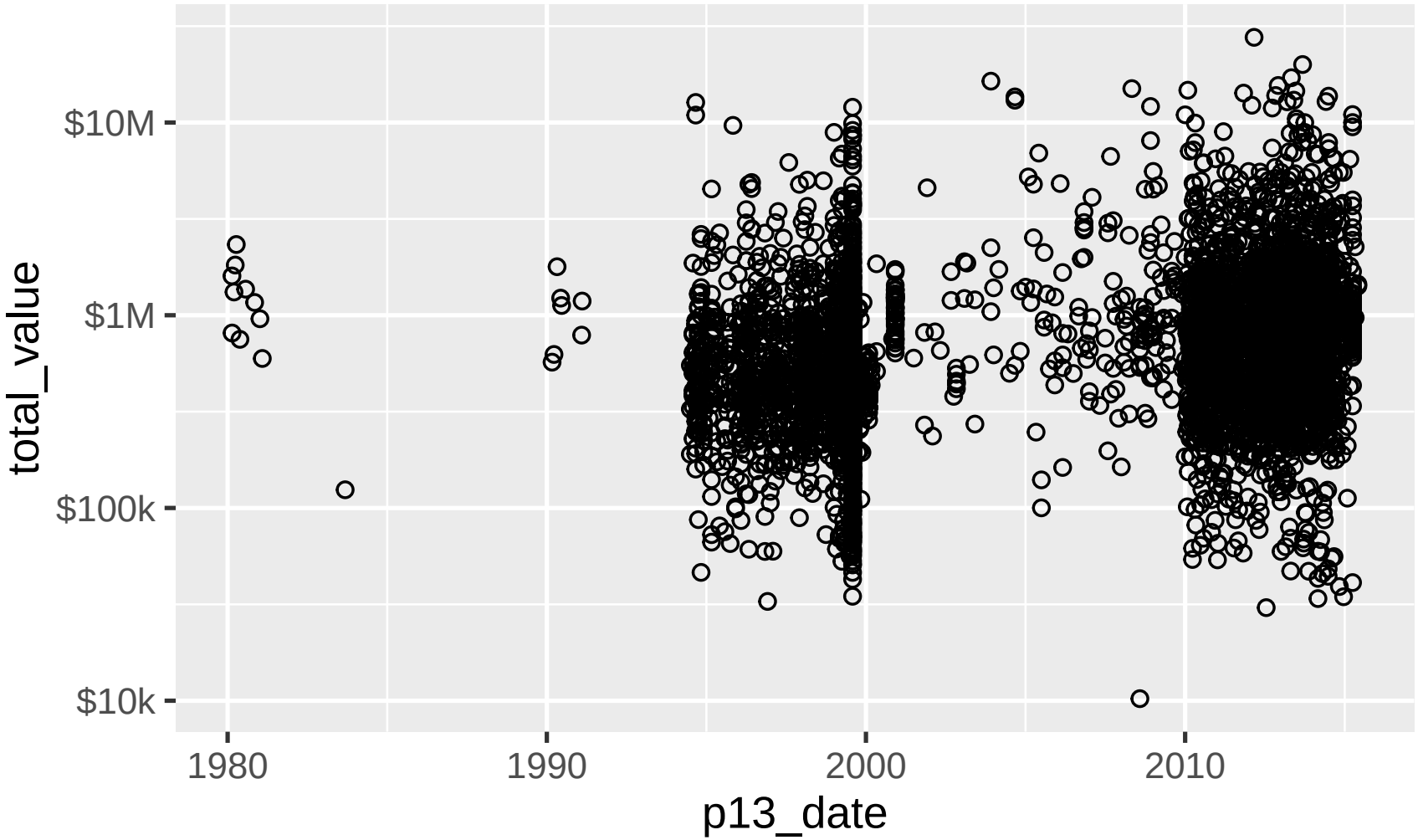


Neighborhoods



Prop 13

- Raw data has ‘recordation date’, ‘sales date’ and ‘change date’.
- Using latest of those to make a ‘p13_date’



Modeling in R

- Linear model in R use lm() with a "formula"
  - Y ~ x1 : basic prediction of Y using x1 with (implied) intercept
  - Y ~ x1 - 1 : same, but (implied) intercept removed
  - Y ~ x1 + x2 : predict on x1 and x2
  - Y ~ x1 + I(x1^2) : Polynomial term. Use I to protect math.
  - Y ~ poly(x1, 2) : More numerically stable form, harder to analyize
  - Y ~ x1 + x2 + x1:x2 : An interaction term
  - Y ~ x1*x1 : shorthand for above
- See ?formula

Simple model

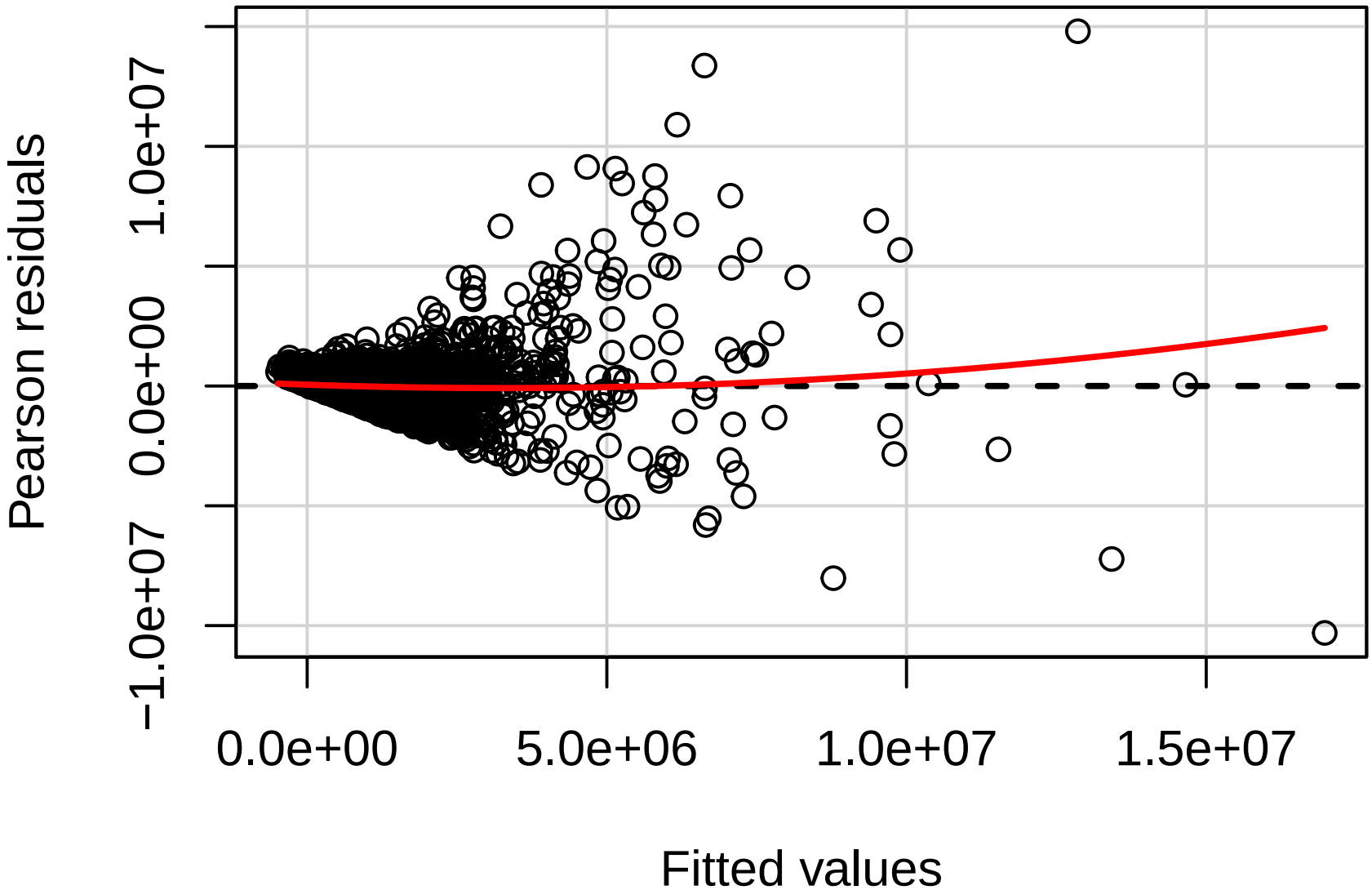
- From prior graphs, try a model of total_value ~ (baseline) + sqft + bathrooms + neighborhood
- You might want to include p13_date, or other columns

```
m1 <- lm(total_value ~ sqft + bathrooms + neighborhood,
         data=train)
```

- Use summary(m1) to look at the model.
  - Too big to show in slide, unfortunately
- Use BIC(m1) to get BIC.

Looking at a Model

- Quality of a model is all about errors / residuals. Look at them!



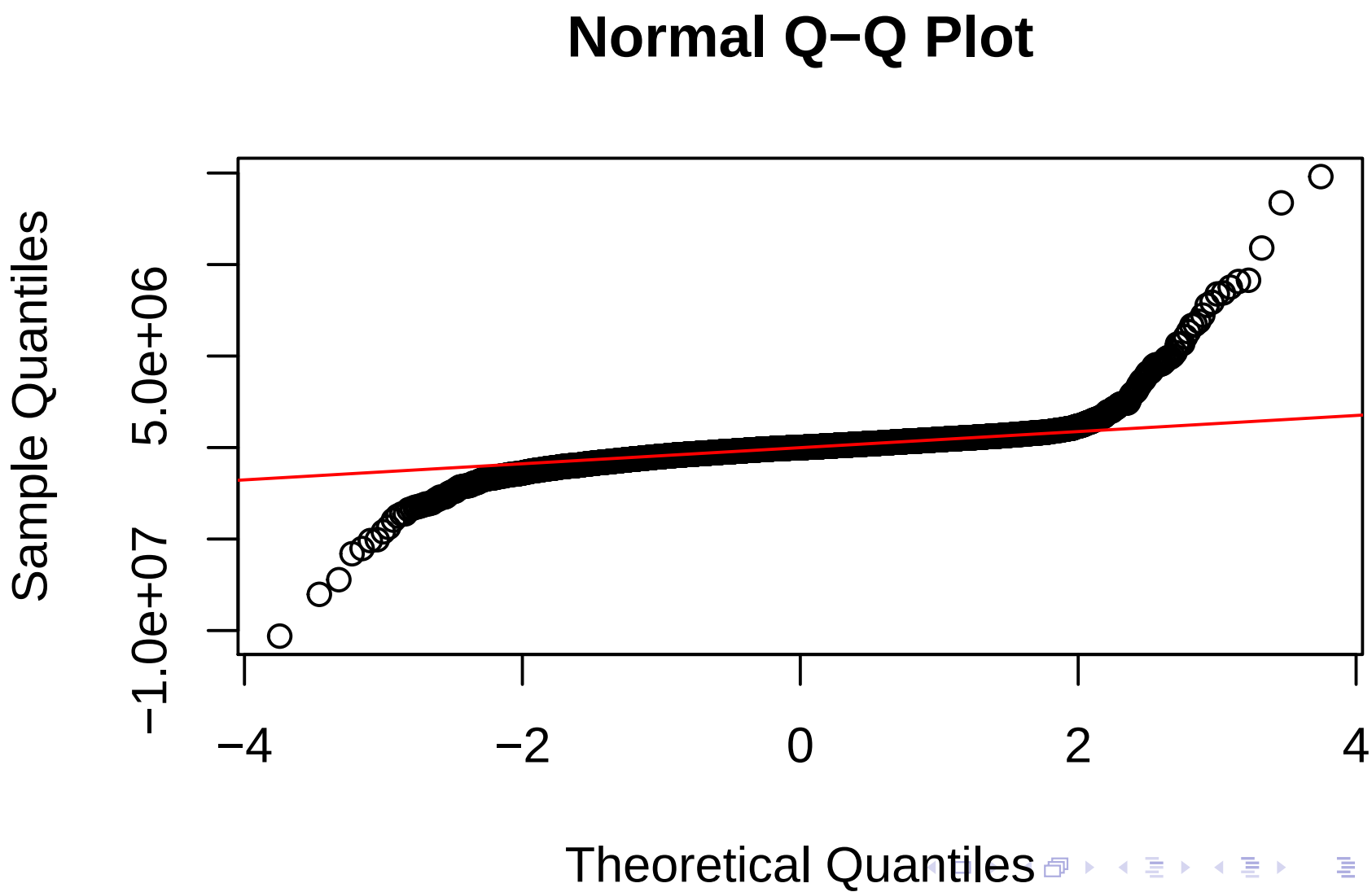
Reminder of Assumptions

- Errors independent and identically distributed
- Errors normally distributed

Are Errors Normally Distributed?

- Not even close!

```
qqnorm(m1$residuals); qqline(m1$residuals, col='red')
```



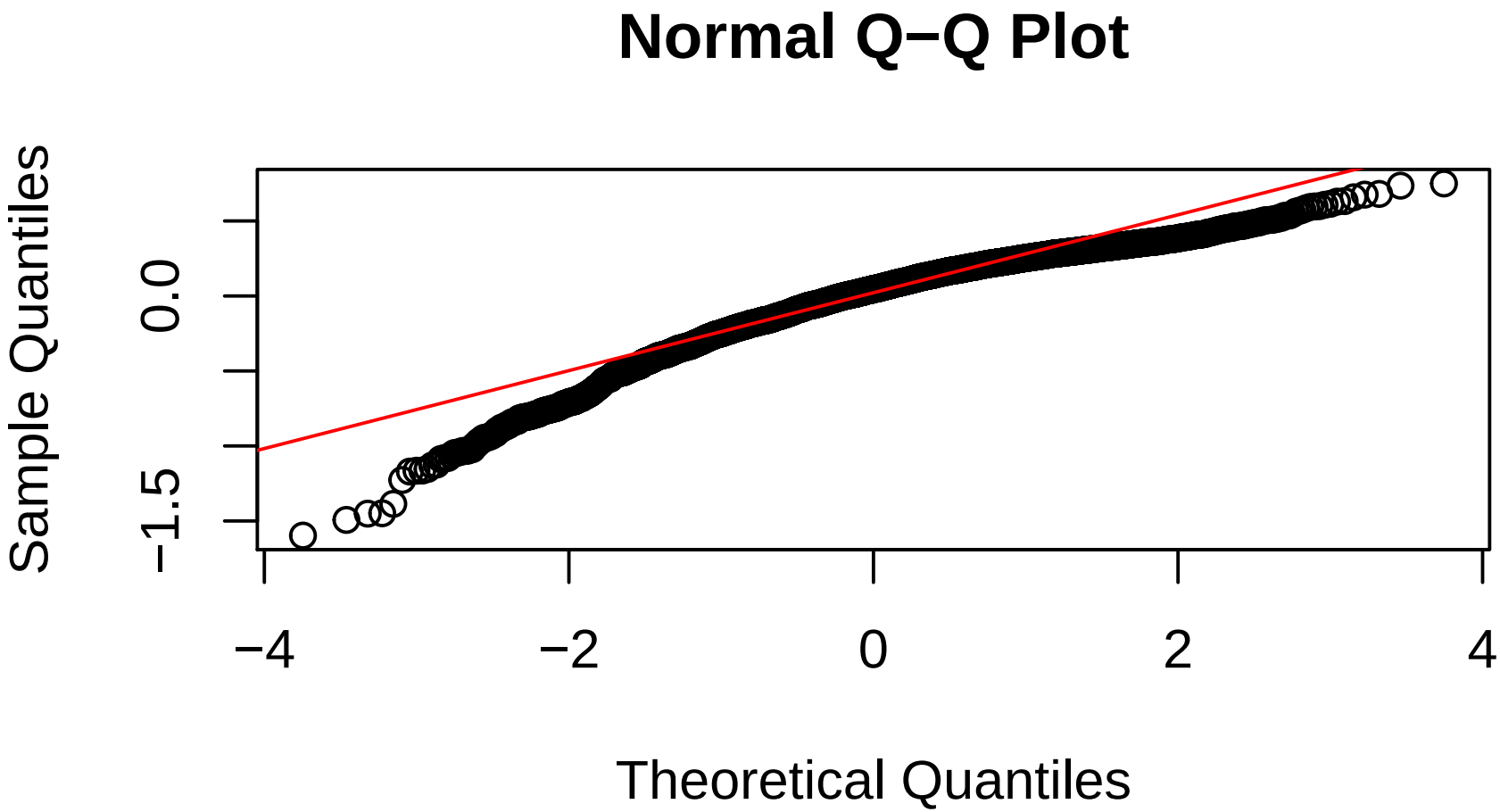
So, Not Good. Now What?

- Value spans several orders of magnitude, exp. distributed
- So log transform it!
  - General rule: log transform, or sqrt transform, to make gaussian
  - Viz "Box-Cox" transformation
- NB: after transforming response, cannot compare models!

```
train$lvalue <- log10(train$total_value)
m2 <- lm(lvalue ~ sqft + bathrooms + neighborhood,
         data=train)
c(summary(m2)$r.squared, BIC(m2))
```

```
## [1]    0.4758074 1376.0569430
```

Q-Q Plot After Transforming

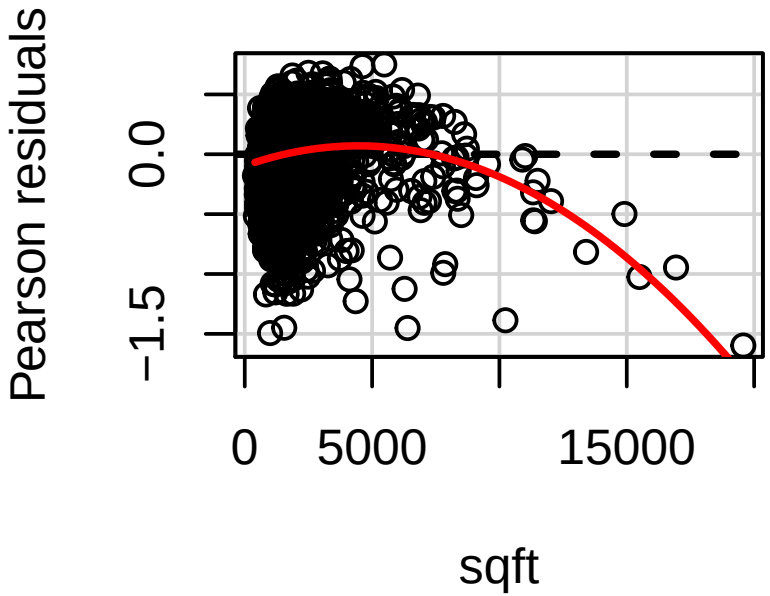


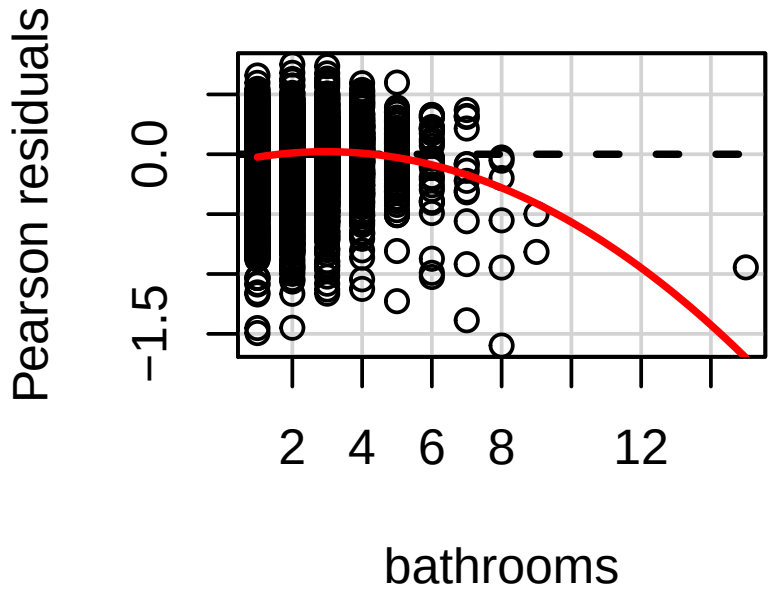
Errors vs Model Terms: IID?

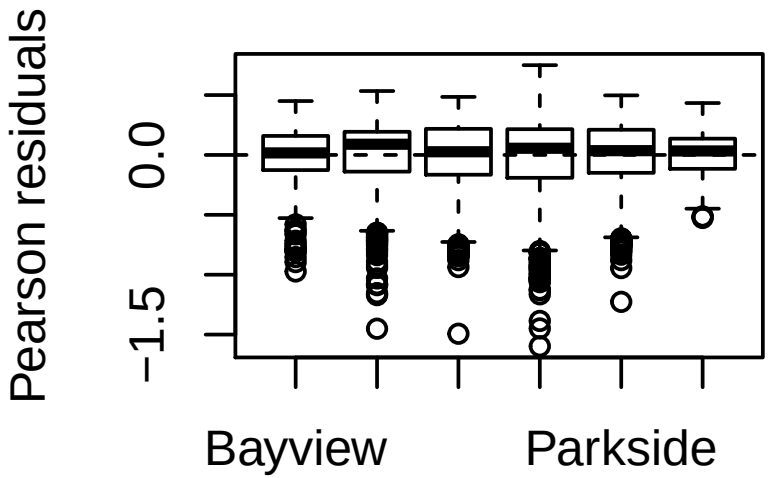
- Assumption is that terms have IID errors
- Regressing on a term, like sqft, means net sum of squares on sqft is zero.
- But remember graphs of linear models for different-shaped data, and look at the residuals vs each predictor
- If not IID, we should fix!
- The car package has a residualPlots function that graphs residuals
  - Even better, fits a quadratic curve to the residuals!

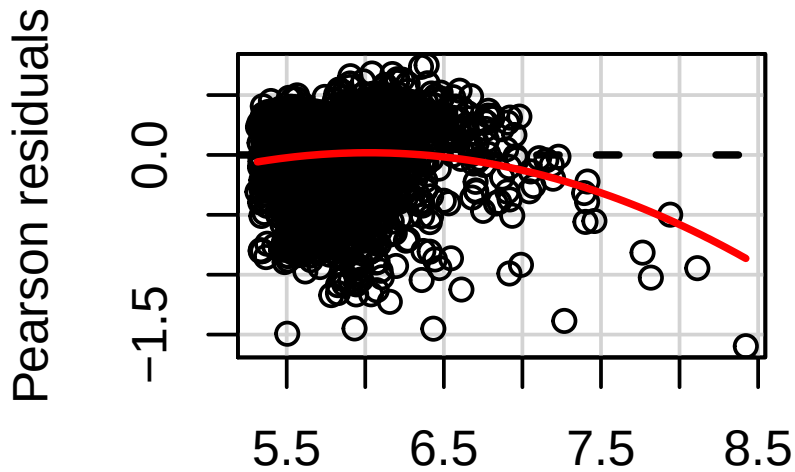
Residuals vs Predictors

```
junk <- residualPlots(m2)
```







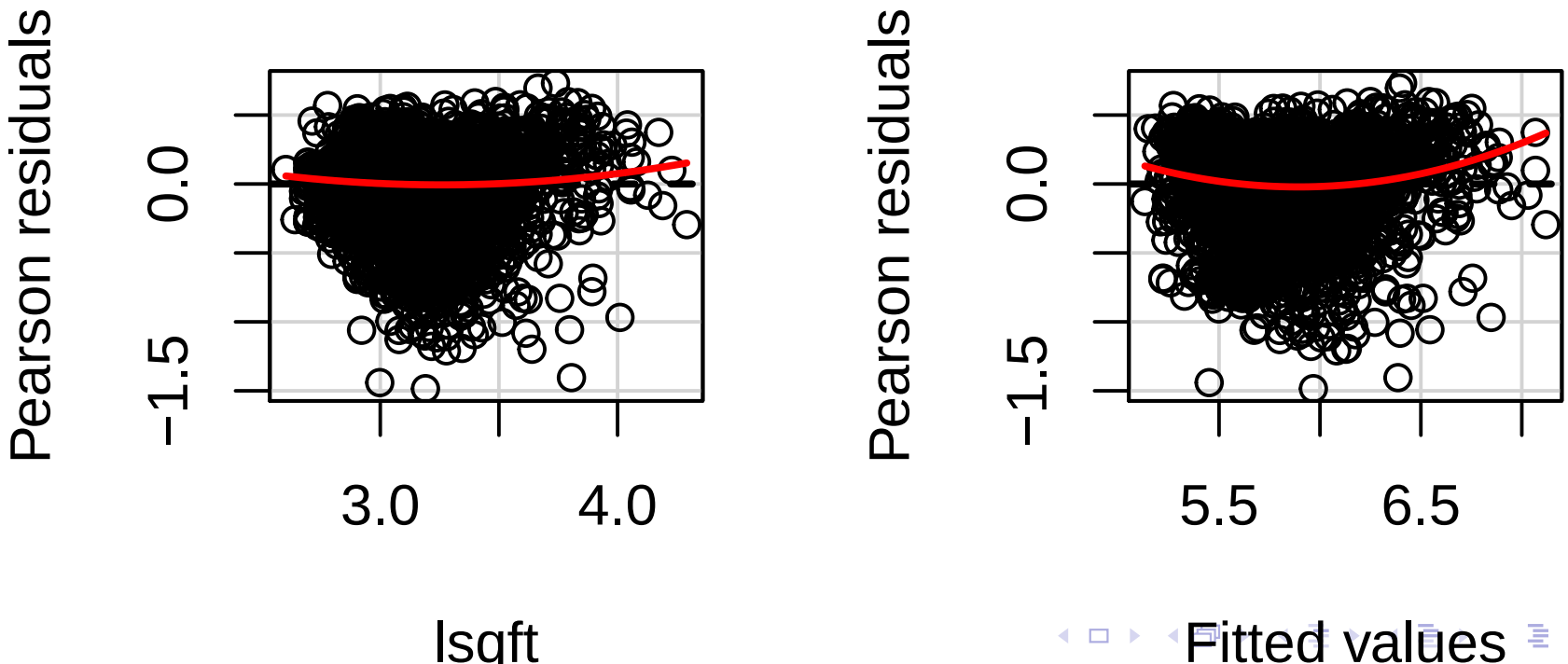


Sqft

- sqft also has several orders of magnitude, so log transform it

```
train$lsqft <- log10(train$sqft)
m3 <- lm(lvalue ~ lsqft + bathrooms + neighborhood,
         data=train)
c(summary(m3)$r.squared, BIC(m3))
```

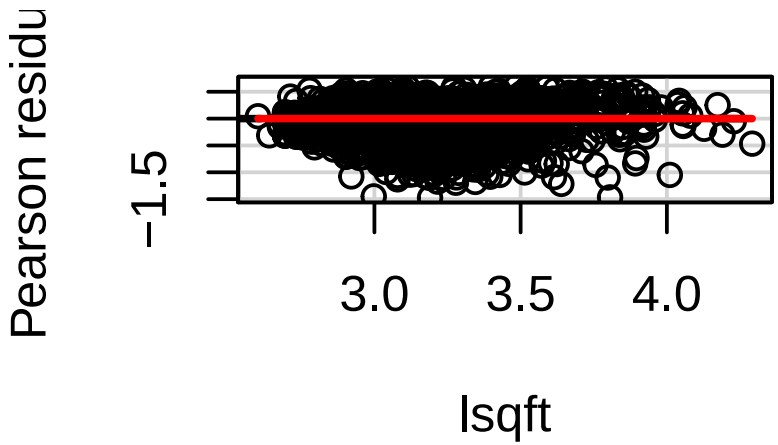
```
## [1]    0.4962334 1154.4726035
```

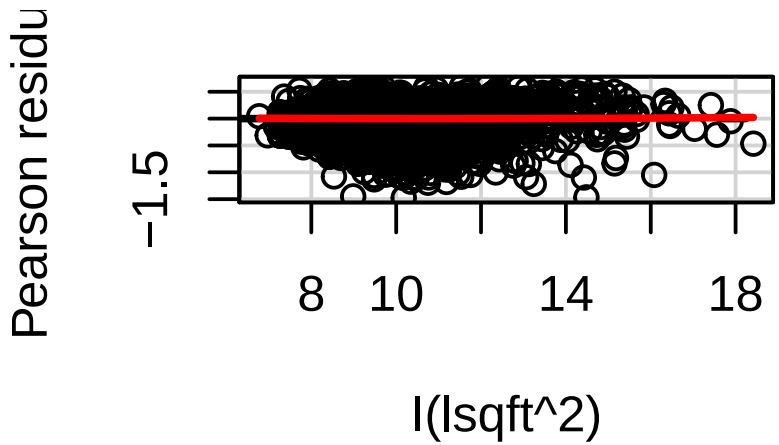


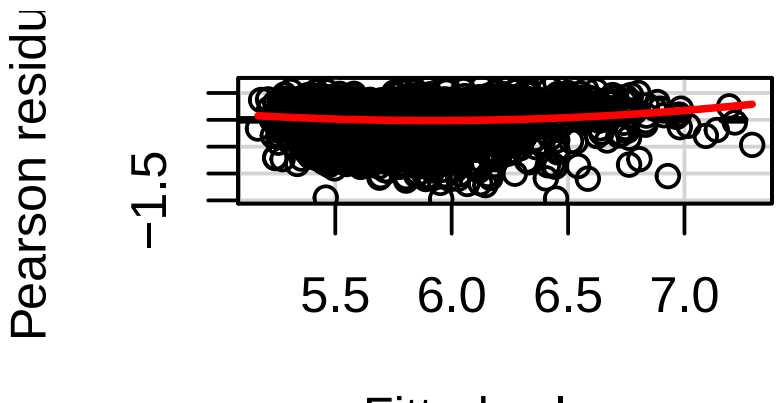
Try to Flatten sqft Errors with Poly Term

```
m4 <- lm(lvalue ~ lsqft + I(lsqft^2) + bathrooms +
         neighborhood, data=train)
c(summary(m4)$r.squared, BIC(m4))
```

```
## [1]    0.4971706 1152.7175298
```



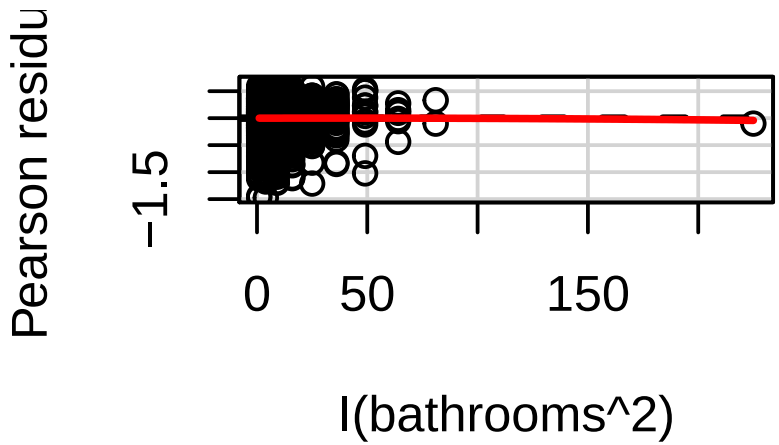


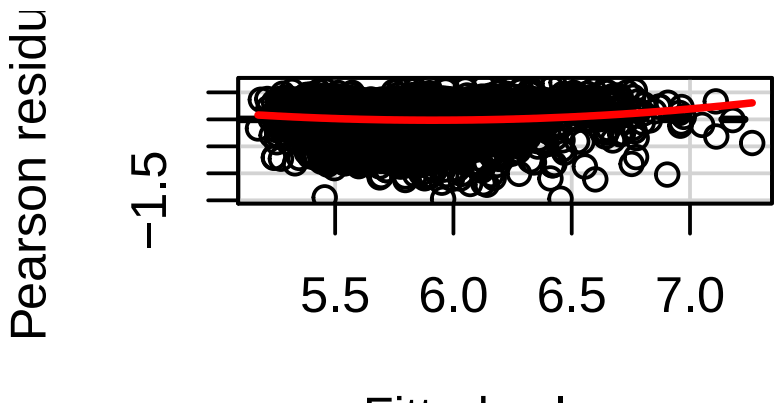


Bathroom Poly Term?

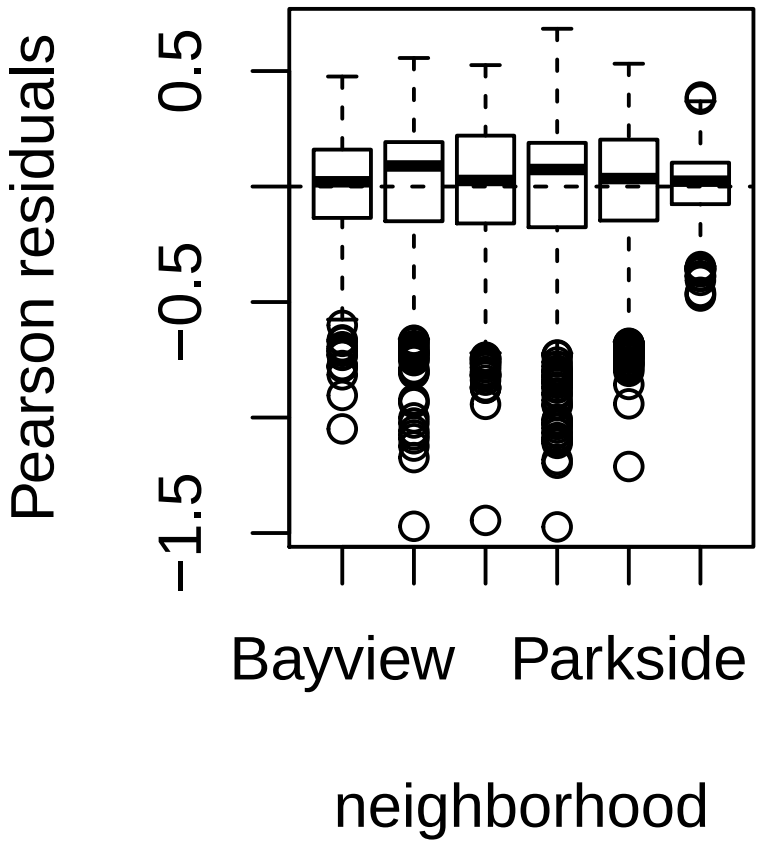
```
m5 <- lm(lvalue ~ lsqft + I(lsqft^2) + bathrooms
         + I(bathrooms^2) + neighborhood, data=train)
c(summary(m5)$r.squared, BIC(m5))
```

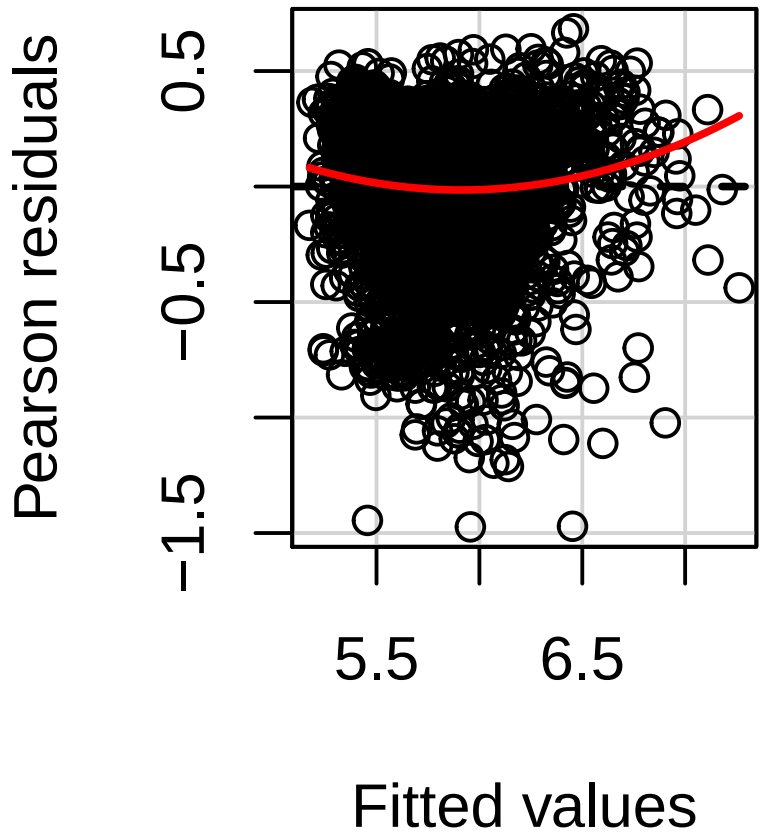
```
## [1]    0.4977772 1154.6136363
```





Neighborhood





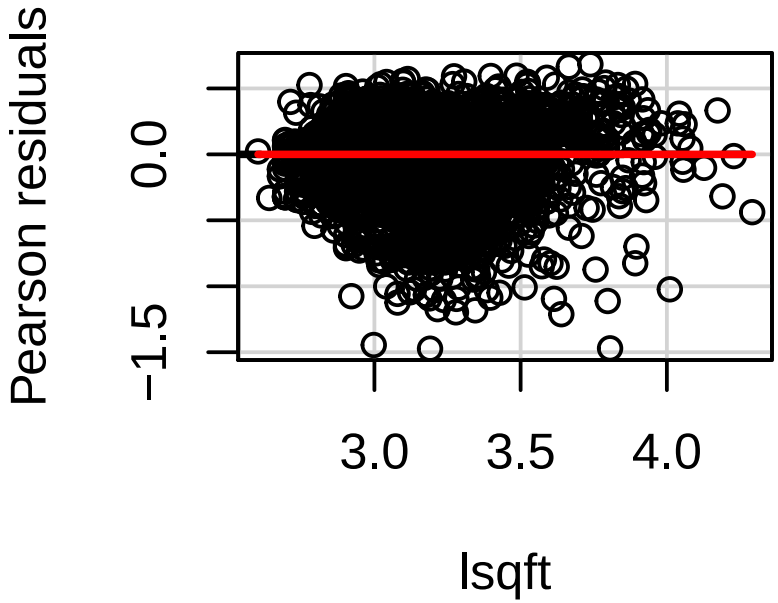
Neighborhood

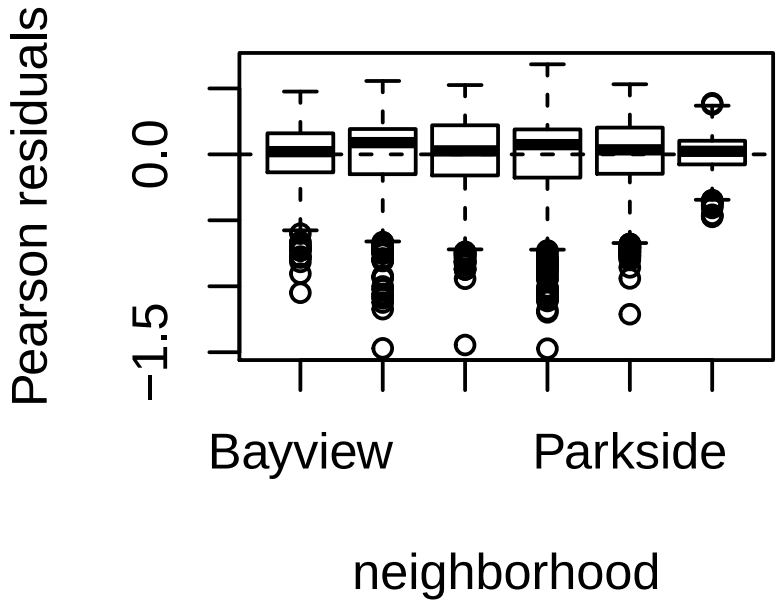
- Cannot add a poly term for a neighborhood.
- But what about a model where the coef. for sqft depends on neighborhood?

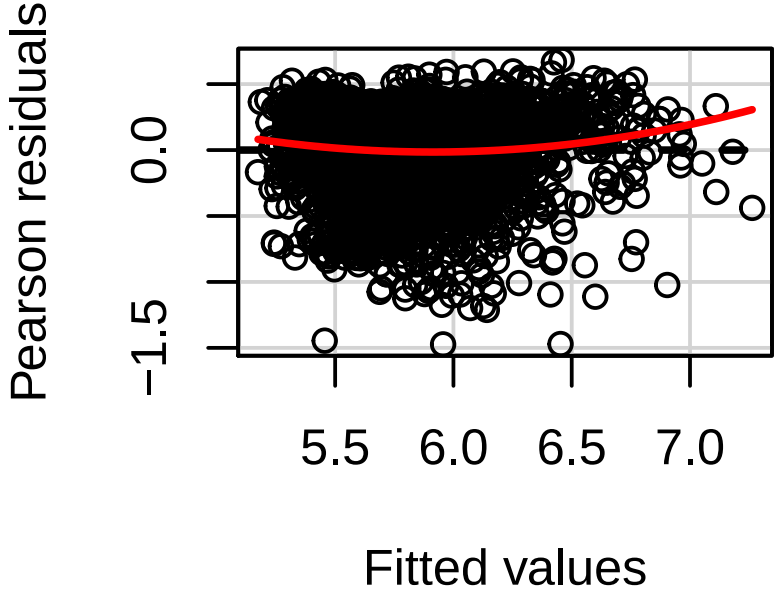
```
m6 <- lm(lvalue ~ lsqft + I(lsqft^2) + bathrooms
         + neighborhood + lsqft:neighborhood,
         data=train)
c(summary(m6)$r.squared, BIC(m6))
```

```
## [1]   0.5147269 997.7174907
```

Neighborhood







Etc.

- Continue making models, looking at residual graphs, and trying out terms
- Strive for a high R^2, and a low BIC.

Caution: Co-linearity

- Had said "most" matricies have inverses. But if terms are co-linear, then there is no inverse.
  - Analogy: algbraic equations where one row is multiple of another
- Even if not exact, close alignment between terms is bad: inverse is ill-defined.
- Data has sqft + bedrooms + bathrooms + all_rooms (not used), which all suggest a notion of 'size'
  - Could end up with big positive coefficients for bedrooms + bathrooms, big negative coefficient for all_rooms.
- Techniques for dealing with this work to separate the ideas
  - Simplest apprach: think of replacing 'size' by 'avg room size'
  - More robust approaches (PCA) won't fit in this lecture
- For today, will leave at "be cautious."

Recap: "Doing" Data Science

- Have a question (property values $\sim$?)
- Look at data
- Build simple model (m1)
- Extend the model (m2...)