STAT 135
13. Linear predictions with multiple features

Spring 2022

Lecturer: Dr Rebecca Barter (*she/her*)
Office hours: Tu 9:30-10:30 (in person), Th 4-5pm (virtual)
Office: Evans 339

Email: rebeccabarter@berkeley.edu
Twitter: @rlbarter
GitHub: rlbarter

Multiple predictive features

Multiple predictive features

So far we have only considered linear relationships that involve one covariate:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

What if we have many covariates? $x_1, x_2, \ldots x_p$ that we want to use to generate a prediction

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_p x_{i,p} + \epsilon_i$$

For example:

$$price_i = \beta_0 + \beta_1 area_i + \beta_2 quality_i + \beta_3 year_i + \beta_4 bedrooms_i + \epsilon_i$$

Terminology

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_p x_{i,p} + \epsilon_i$$

The **response** variable goes by many names:

- Dependent variable
- Outcome
- Output variable
- Label

Covariates also go by many names:

- Feature
- Predictive feature,
- Independent variable

Categorical covariates

What do we do when one of our features are **categorical**?

$$price_i = \beta_0 + \beta_1 area_i + \beta_2 quality_i + \beta_3 year_i + \beta_4 bedrooms_i + \boxed{\beta_5 neighborhood_i} + \epsilon_i$$

As an example, suppose you want to generate a price prediction for a house with:

- 3-bedrooms
- 1,013ft^2 living area
- built in 1926
- quality score of 6
- In the neighborhood of North Ames (”NAmes")

Setting β_0 to 0 and all other β coefficients to 1, the prediction is

$$\widehat{price} = 1013 + 6 + 1926 + 3 + \text{NAmes}$$

But how do you add “NAmes”???

Dummy variables

$$price_i = \beta_0 + \beta_1 area_i + \beta_2 quality_i + \beta_3 year_i + \beta_4 bedrooms_i + \boxed{\beta_5 neighborhood_i} + \epsilon_i$$

Solution: create dummy variables (binary variables) for each level of the categorical variable

$$\begin{aligned} price_i = \beta_0 &+ \beta_1 area_i + \beta_2 quality_i + \beta_3 year_i + \beta_4 bedrooms_i + \boxed{\beta_5 NAmes_i} \\ &+ \boxed{\beta_6 Gilbert_i} + \boxed{\beta_7 Somerset_i} + \boxed{\beta_8 NridgHt_i} + \boxed{\beta_9 Edwards_i} + \epsilon_i \end{aligned}$$

Where

$$NAmes_i = \begin{cases} 1 \text{ if house } i \text{ is in neighborhood NAmes} \\ 0 \text{ if house } i \text{ is not in neighborhood NAmes} \end{cases}$$

$$Gilbert_i = \begin{cases} 1 \text{ if house } i \text{ is in neighborhood Gilbert} \\ 0 \text{ if house } i \text{ is not in neighborhood Gilbert} \end{cases}$$

etc

Dummy variables reference level

$$price_i = \beta_0 + \beta_1 area_i + \beta_2 quality_i + \beta_3 year_i + \beta_4 bedrooms_i + \boxed{\beta_5 NAmes_i} + \boxed{\beta_6 Gilbert_i} + \boxed{\beta_7 Somerset_i} + \boxed{\beta_8 NridgHt_i} + \boxed{\beta_9 Edwards_i} + \epsilon_i$$

How would you interpret β_5?

The increase in predicted property price associated with the property moving to the North Ames neighborhood…. from where?

If there are 5 neighborhoods in total, knowing the value of 4 of the neighborhood dummy variables tells us the value of the final dummy variable $\Longrightarrow$ including all levels of a categorical variable is redundant.

(Noe: the data matrix is also not invertible when there exist perfectly collinear variables)

Dummy variables reference level

$$price_i = \beta_0 + \beta_1 area_i + \beta_2 quality_i + \beta_3 year_i + \beta_4 bedrooms_i + \boxed{\beta_5 NAmes_i}$$
$$+ \boxed{\beta_6 Gilbert_i} + \boxed{\beta_7 Somerset_i} + \boxed{\beta_8 NridgHt_i} + \epsilon_i$$

We typically take one level as the reference level, e.g. <u>Edwards</u>
(by removing it from the equation)

How would you interpret β_5 now?

The increase in predicted property price associated with the property moving to the North Ames neighborhood…. from Edwards

Matrix notation of multiple linear regression

Matrix notation

Observation 1 : $y_1 = \beta_0 \times 1 + \beta_1 x_{1,1} + \beta_2 x_{1,2} + ... + \beta_{p-1} x_{1,(p-1)} + \epsilon_1$

Observation 2 : $y_2 = \beta_0 \times 1 + \beta_1 x_{2,1} + \beta_2 x_{2,2} + ... + \beta_{p-1} x_{2,(p-1)} + \epsilon_2$

$$\vdots$$

Observation n : $y_n = \beta_0 \times 1 + \beta_1 x_{n,1} + \beta_2 x_{n,2} + ... + \beta_{p-1} x_{n,(p-1)} + \epsilon_n$

This can be written:

$$\underset{(n\times 1)}{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}} = \underset{(n\times p)}{\begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p-1} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p-1} \\ 1 & x_{3,1} & x_{3,2} & \cdots & x_{3,p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p-1} \end{bmatrix}} \underset{(p\times 1)}{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix}} + \underset{(n\times 1)}{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{bmatrix}}$$

Matrix notation

Multiple regression equations in matrix form:

$$Y = X\beta + \epsilon$$

$$\underset{(n\times 1)}{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}} = \underset{(n\times p)}{\begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,p-1} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,p-1} \\ 1 & x_{3,1} & x_{3,2} & \dots & x_{3,p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \dots & x_{n,p-1} \end{bmatrix}} \underset{(p\times 1)}{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix}} + \underset{(n\times 1)}{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{bmatrix}}$$

- Y is the $(n \times 1)$ (observed) **response vector**
- X is the $(n \times p)$ (observed) **design matrix**
- β is the $(p \times 1)$ (unobserved) **coefficient vector**
- ϵ is the $(n \times 1)$ (unobserved) **error vector**

Matrix notation

$$Y = X\beta + \epsilon$$

$$\underset{(n\times 1)}{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}} = \underset{(n\times p)}{\begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p-1} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p-1} \\ 1 & x_{3,1} & x_{3,2} & \cdots & x_{3,p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p-1} \end{bmatrix}} \underset{(p\times 1)}{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix}} + \underset{(n\times 1)}{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{bmatrix}}$$

The response for an individual observation can be written as:

$$y_i = \boldsymbol{x}_i\boldsymbol{\beta} + \epsilon_i \ = [1 \ x_{i,1} \ \ldots \ x_{i,p-1}] \begin{bmatrix} \beta_0 \\ \beta_1 \\ \cdots \\ \beta_{p-1} \end{bmatrix} + \epsilon_i \ = \beta_0 + \beta_1 x_{i,i} + \ldots + \beta_p x_{p-1,i} + \epsilon_i$$

Matrix notation

$$Y = X\beta + \epsilon$$

$$\underset{(n\times 1)}{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}} = \underset{(n\times p)}{\begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,p-1} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,p-1} \\ 1 & x_{3,1} & x_{3,2} & \dots & x_{3,p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \dots & x_{n,p-1} \end{bmatrix}} \underset{(p\times 1)}{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix}} + \underset{(n\times 1)}{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{bmatrix}}$$

And we have the following requirements on the error *vector*:

$$E[\epsilon] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad Cov(\epsilon) = \sigma^2 I_{n\times n} = \begin{bmatrix} \sigma^2 & 0 & \dots & 0 \\ 0 & \sigma^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \sigma^2 \end{bmatrix}$$

So $Cov(Y) = Cov(X\beta + \epsilon) = Cov(\epsilon) = \sigma^2 I_{n\times n}$

Matrix notation

$$Y = X\beta + \epsilon$$

$$\underset{(n\times 1)}{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}} = \underset{(n\times p)}{\begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p-1} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p-1} \\ 1 & x_{3,1} & x_{3,2} & \cdots & x_{3,p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p-1} \end{bmatrix}} \underset{(p\times 1)}{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix}} + \underset{(n\times 1)}{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{bmatrix}}$$

To generate a prediction, we need to estimate the vector, β, and then compute:

$$\hat{Y} = X\hat{\beta}$$

Matrix notation prediction example

Our fitted line for the four-feature fit is

$$\widehat{price} = -890{,}474 + 88\ area + 20{,}208\ quality + 435\ year - 14{,}245\ beds$$

To generate a prediction for the following two houses:

Area	Quality	Year	Beds
1200	6	1956	2
890	8	1970	1

$$(1 \times -890474) + (1200 \times 88) + \ldots + (2 \times -14245)$$

$$X = \begin{bmatrix} 1 & 1200 & 6 & 1956 & 2 \\ 1 & 890 & 8 & 1970 & 1 \end{bmatrix} \quad \hat{\beta} = \begin{bmatrix} -890474 \\ 88 \\ 20208 \\ 435 \\ -14245 \end{bmatrix} \quad \text{So } \hat{Y} = X\hat{\beta} = \begin{bmatrix} 158{,}744 \\ 192{,}215 \end{bmatrix}$$

LS estimates using matrix notation of multiple linear regression

LS estimates of β

Note that finding the values of $\beta_0, \beta_1, ...\beta_{p-1}$ that minimizes

$$S(\beta_0, \beta_1, ..., \beta_{p-1}) = \sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_{i,1} - \beta_2 x_{i,2} - ... - \beta_{p-1} x_{i,p-1})^2$$

is the same as finding the vector $\boldsymbol{\beta}$ that minimizes

$$\boxed{S(\boldsymbol{\beta}) = \|Y - X\boldsymbol{\beta}\|_2^2 = (Y - X\boldsymbol{\beta})^T(Y - X\boldsymbol{\beta})}$$

(this is just a number: $(Y - X\boldsymbol{\beta})^T(Y - X\boldsymbol{\beta})$ has dim (1×1))

where $\|\cdot\|_2$ is the L^2-norm which is defined for a vector $\mathbf{x} = (x_1, ..., x_n)$ by

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{\mathbf{x}^T\mathbf{x}}$$

LS estimates of β

We want to find the vector $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_{p-1})$ that minimizes

$$S(\boldsymbol{\beta}) = \|Y - X\boldsymbol{\beta}\|_2^2 = (Y - X\boldsymbol{\beta})^T(Y - X\boldsymbol{\beta})$$

We will show that the **Least Squares estimate** for $\boldsymbol{\beta}$ is:

$$\boxed{\widehat{\boldsymbol{\beta}} = \left(X^T X\right)^{-1} X^T Y}$$

Proof of LS estimates of β

To show that $\widehat{\boldsymbol{\beta}} = \left(X^T X\right)^{-1} X^T Y$, we want to find the vector $\boldsymbol{\beta}$ that satisfies

$$\frac{\partial S(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 0$$

where

$$\begin{aligned}
\frac{\partial S(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} &= \frac{\partial}{\partial \boldsymbol{\beta}} (Y - X\boldsymbol{\beta})^T (Y - X\boldsymbol{\beta}) \\
&= \frac{\partial}{\partial \boldsymbol{\beta}} \left[Y^T Y - \boldsymbol{\beta}^T X^T Y - Y^T X \boldsymbol{\beta} + \boldsymbol{\beta}^T X^T X \boldsymbol{\beta}\right] \\
&= -X^T Y - X^T Y + 2X^T X \boldsymbol{\beta}
\end{aligned}$$

where we have used the following differentiation identities ($\mathbf{x}$ and $\mathbf{a}$ are both $(p \times 1)$ vectors and A is a symmetric $(p \times p)$ matrix):

$$\tfrac{\partial}{\partial \mathbf{x}} \mathbf{a}^T \mathbf{x} = \tfrac{\partial}{\partial \mathbf{x}} \mathbf{x}^T \mathbf{a} = \mathbf{a} \qquad \tfrac{\partial}{\partial \mathbf{x}} \mathbf{x}^{\mathbf{T}} A \mathbf{x} = 2A\mathbf{x} = 2\mathbf{x}^T A$$

Proof of LS estimates of β

Thus we need to solve for $\widehat{\boldsymbol{\beta}}$ in:

$$-X^T Y - X^T Y + 2X^T X\widehat{\boldsymbol{\beta}} = 0$$

rearranging gives the **normal equations**

$$X^T X\widehat{\boldsymbol{\beta}} = X^T Y$$

so that if $(X^T X)$ is invertible gives

$$\boxed{\widehat{\boldsymbol{\beta}} = \left(X^T X\right)^{-1} X^T Y}$$

and thus our **fitted model** is given by

$$\boxed{\widehat{Y} = X\widehat{\boldsymbol{\beta}} = X\left(X^T X\right)^{-1} X^T Y}$$

Exercise:

For the simple linear regression case ($y_i = \beta_0 + \beta_1 x_i + \epsilon_i$), show that

$$Y = X\beta + \epsilon: \qquad \underset{(n\times 1)}{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}} = \underset{(n\times 2)}{\begin{bmatrix} 1 & x_{1,1} \\ 1 & x_{2,1} \\ 1 & x_{3,1} \\ \vdots & \vdots \\ 1 & x_{n,1} \end{bmatrix}} \underset{(2\times 1)}{\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}} + \underset{(n\times 1)}{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{bmatrix}}$$

Derive the following three expressions:

1. $X^T X = \begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix}$

2. $X^T Y = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{bmatrix}$

3. $(X^T X)^{-1} = \dfrac{1}{n\sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i\right)^2} \begin{bmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{bmatrix}$

When Matrix A = $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$

$$A^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

The “Hat” matrix

$$\boxed{\widehat{\boldsymbol{\beta}} = \left(X^T X\right)^{-1} X^T Y}$$

$$\boxed{\widehat{Y} = X\widehat{\boldsymbol{\beta}} = X\left(X^T X\right)^{-1} X^T Y}$$

If we define the **“hat” matrix** to be: $H = X(X^T X)^{-1} X^T$
Then $\hat{Y} = HY$

i.e., the hat matrix, H, is the matrix transformation of Y that generates our predictions!

- H is a symmetric $n \times n$ matrix

- H is idempotent, i.e, $H^2 = H$

- $HX = X$

- The **residuals** are $e = Y - \hat{Y} = (I - H)Y$

Exercise:

For the simple linear regression case ($y_i = \beta_0 + \beta_1 x_i + \epsilon_i$), show that

$$Y = X\beta + \epsilon: \quad \underset{(n\times 1)}{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}} = \underset{(n\times 2)}{\begin{bmatrix} 1 & x_{1,1} \\ 1 & x_{2,1} \\ 1 & x_{3,1} \\ \vdots & \vdots \\ 1 & x_{n,1} \end{bmatrix}} \underset{(2\times 1)}{\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}} + \underset{(n\times 1)}{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{bmatrix}}$$

Show that

$$\widehat{\boldsymbol{\beta}} = \left(X^T X\right)^{-1} X^T Y$$

yields the same estimates as when we solved for β_0 and β_1 individually.

Example

lm_multi.R

Interpreting the coefficients of a linear fit with multiple predictive features

Multiple linear fit

For the following fit, can you use the coefficients to determine which feature is the most “important” for predicting sale price?

$$\widehat{price} = -890{,}474 + 88\ area + 20{,}208\ quality + 435\ year - 14{,}245\ beds$$

The scale of the coefficient is the same as the scale of the variable!

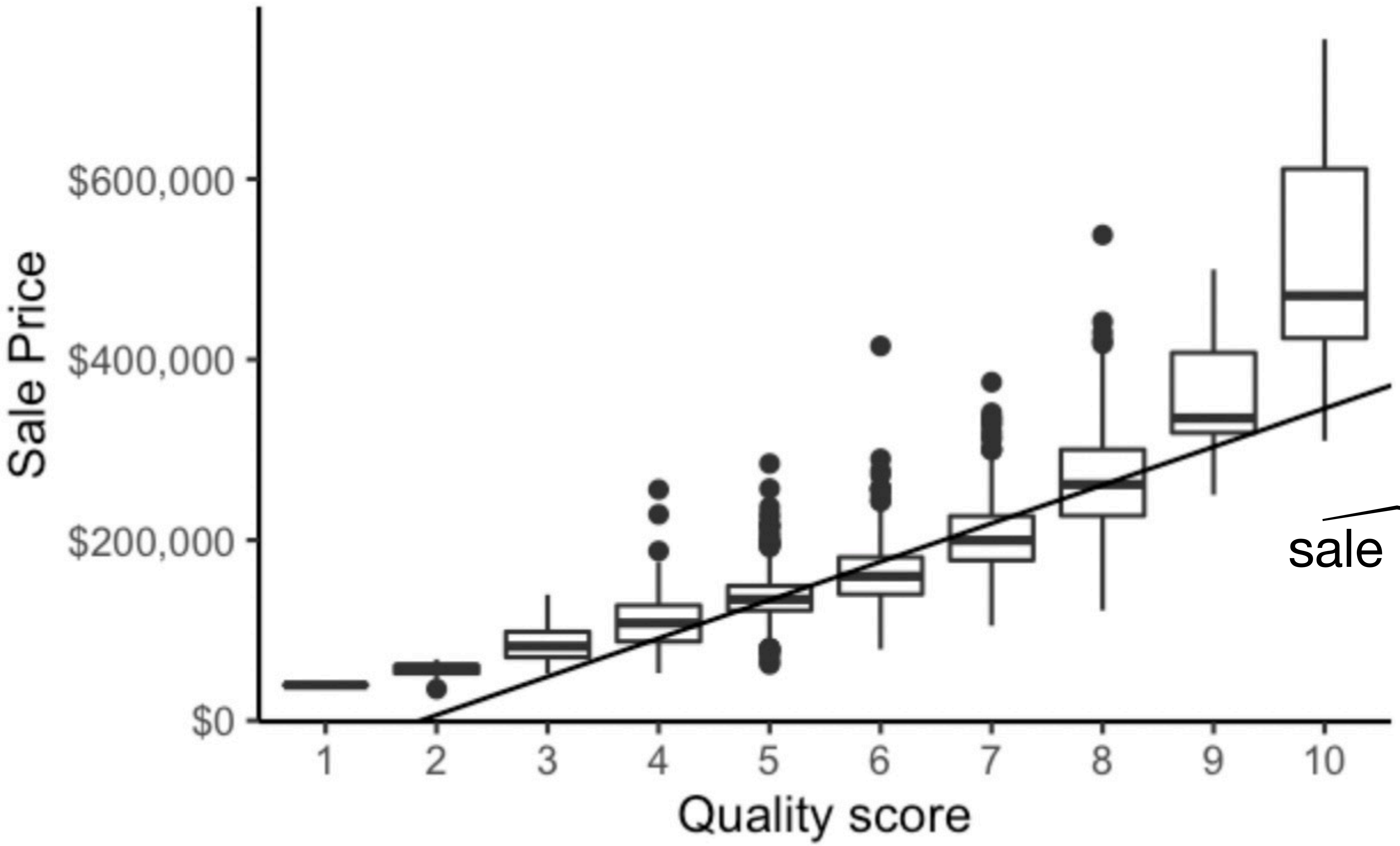
Increasing 1 unit (square ft) for *area* is nowhere near as dramatic as increasing 1 unit (point) for *quality*

The coefficients are not directly comparable unless we scale them (which we will discuss in the context of inference)

Feature transformations

Linearizing non-linear relationships

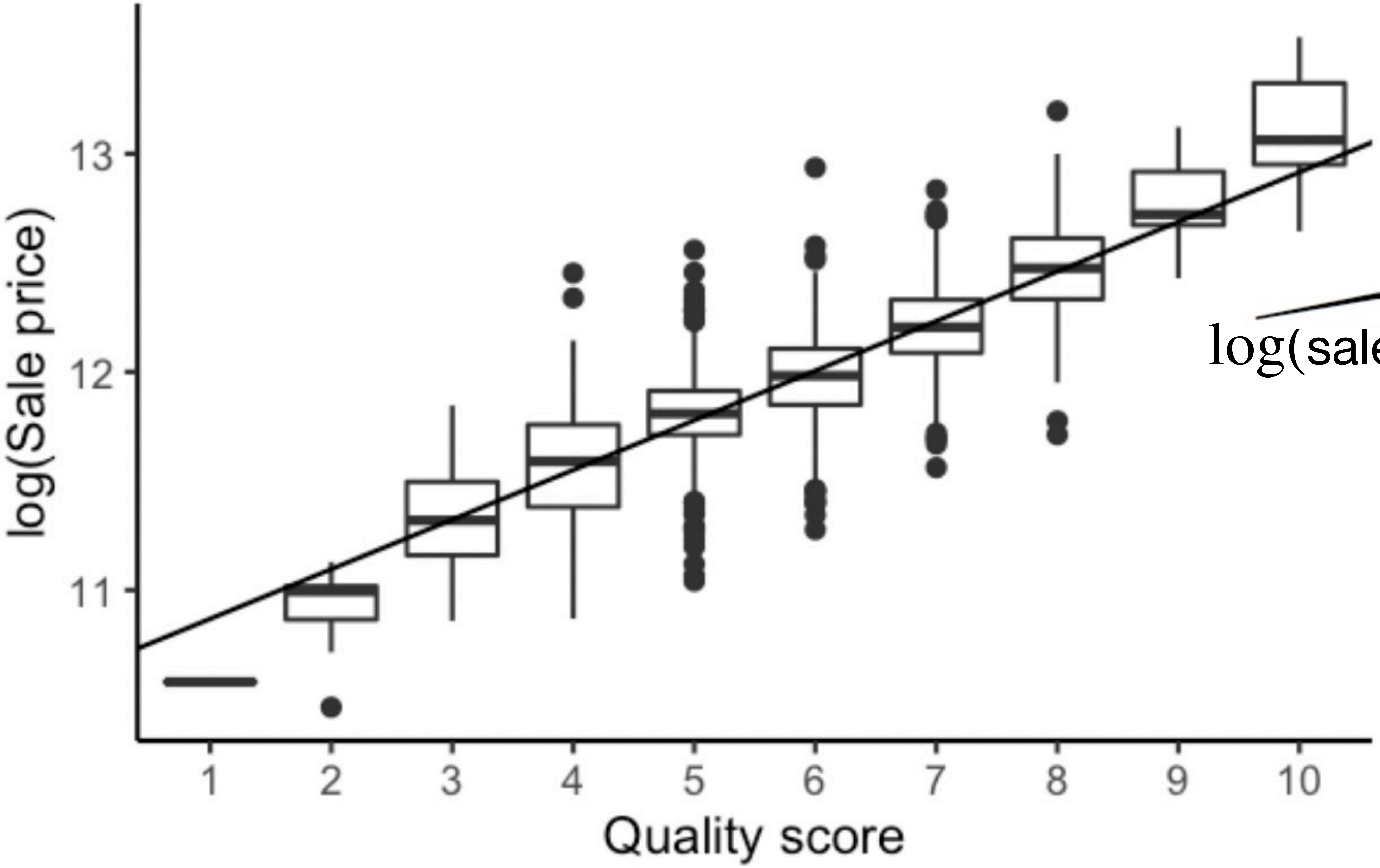
Do you think that sale price and quality score have a linear relationship?



$$\widehat{\text{sale price}} = \hat{\beta}_0 + \hat{\beta}_1 \text{quality}$$

Transforming the response

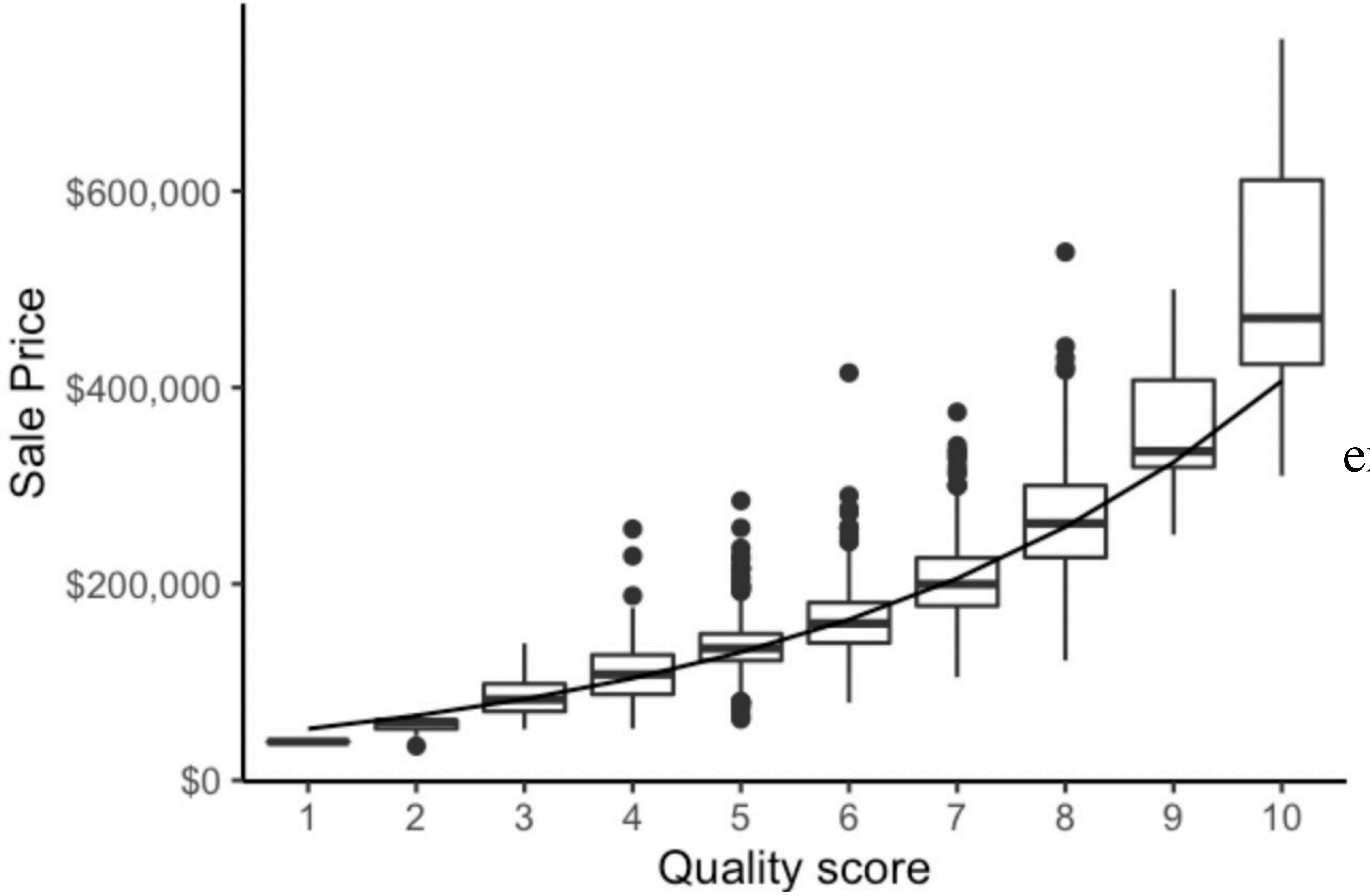
If we instead fit a line to the log-transformed response it fits a lot better



$$\widehat{\log(\text{sale price})} = \hat{\beta}_0 + \hat{\beta}_1 \text{quality}$$

Transforming the response

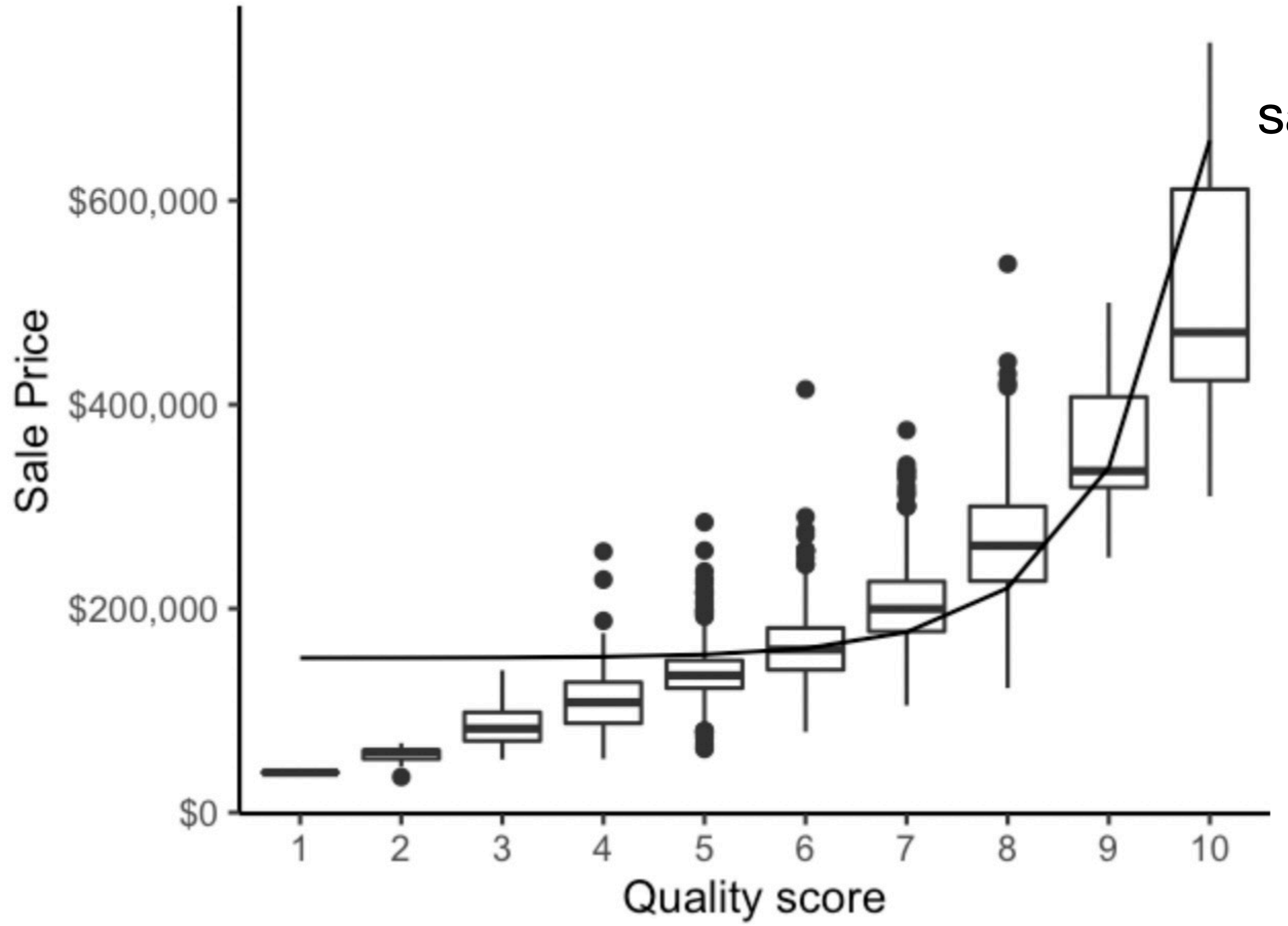
We can view this log-transformed prediction in the original data space by exponentiating the sale price predictions:



$$\exp(\widehat{\log(\text{sale price})}) = \hat{\beta}_0 + \hat{\beta}_1 \text{quality}$$

Transforming the predictor variable

What if, instead of log-transforming the sale price response, we exponentiate the quality score predictor variable?



$$\widehat{\text{sale price}} = \hat{\beta}_0 + \hat{\beta}_1 \exp(\text{quality})$$

This clearly does **not** look the same as the log-transformed response version.

The LS algorithm is not symmetric in x and y.

Transformations in multiple predictor settings

Transforming the predictor/covariate versus transforming the response:

Transforming the response affects the feature-response relationship for all predictive features

Transforming individual predictive features does not affect the format of the relationships between the other features and the response, e.g.:

$$\widehat{\text{sale price}} = \hat{\beta}_0 + \hat{\beta}_1 \text{quality}^2 + \hat{\beta}_2 \exp(\text{bedrooms}) + \hat{\beta}_3 \text{area}$$

(Area is still linearly related to sale price)

Transformations in multiple predictor settings

Tips for deciding whether or not to transform your data:

Visualize the relationship between your features and response variable

LS will often perform better on **symmetric features**, so visualize the distribution of your features individually

Try a few reasonable transformations and see if the prediction performance **on the test set** improves.

Transformations in multiple predictor settings

Common transformations

Logarithmic

Inversion (1/x)

Square-root

Example

ls_non_linear.R

Feature engineering and feature selection

Data cleaning and feature engineering

The full Ames data has over 80 predictive features.

Not all of the features are immediately well-formatted as predictive variables

- **Feature engineering:** is the process of using domain knowledge to extract and modify features so that they are better suited for the predictive problem

Tips:

- Remove ID variables
- Think about how a computer will interpret each feature (e.g. categorical variable with levels “high”, “med”, “low”)
- Look at each variable manually and assess its usefulness
- Is there any useful information in the data that isn’t explicitly coded as a variable?

Data cleaning and feature engineering

Let’s take a closer look at the raw Ames data for predictive modeling

clean_ames_explore.R

Two alternative ways of cleaning the data:

clean_ames_long.R (manually making a lot of cleaning choices)
clean_ames_recipe.R (automating the cleaning choices using “recipes”)

Collinearity

If there are features that are highly correlated with one another (highly collinear), then your algorithmic fit might be unstable.

Example from the Ames data:

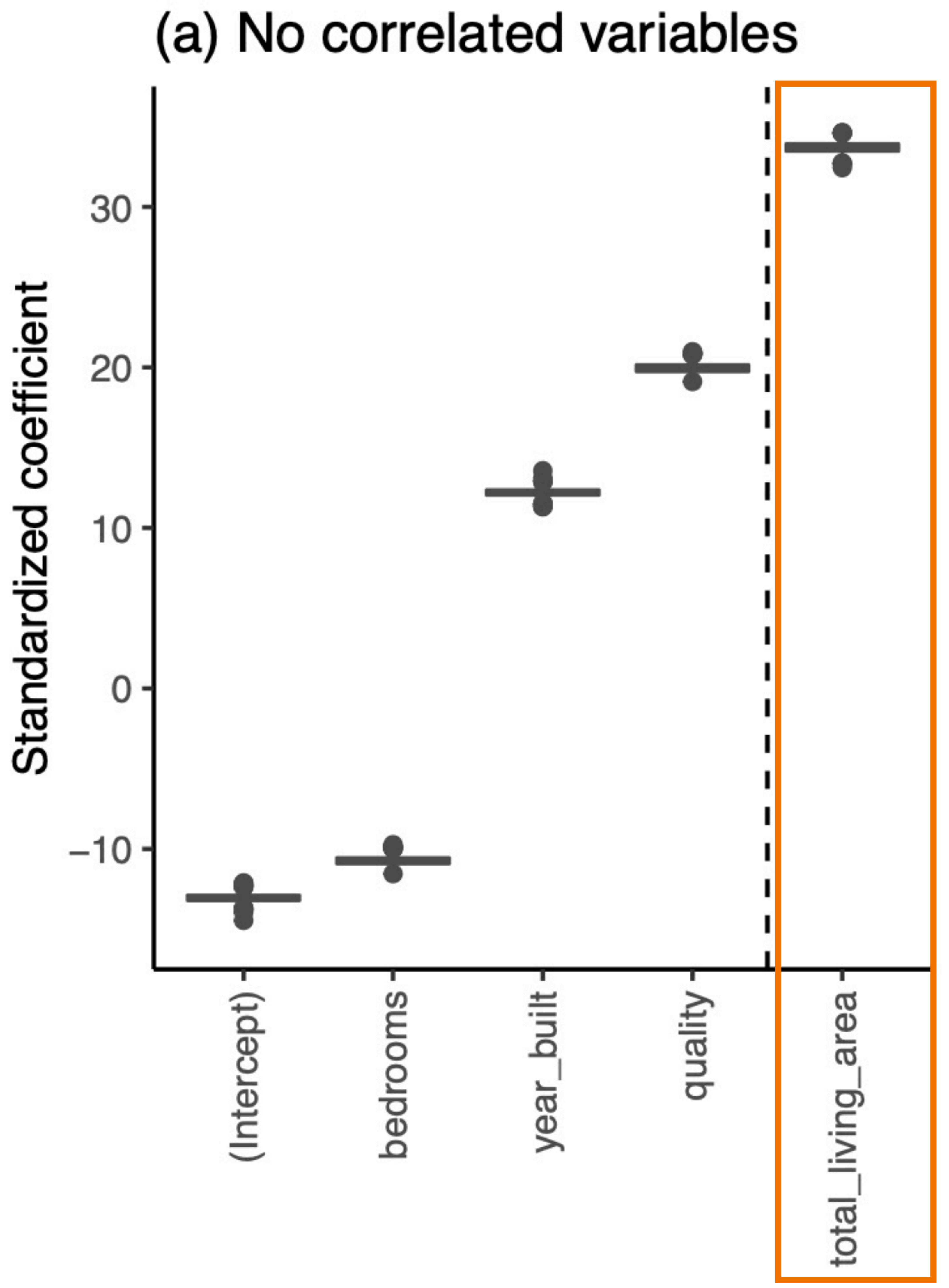
$$\text{predicted price} = b_0 + b_1 \mathbf{area} + b_2 \mathbf{area2} + b_3 \text{quality} + b_4 \text{year} + b_5 \text{bedrooms}$$

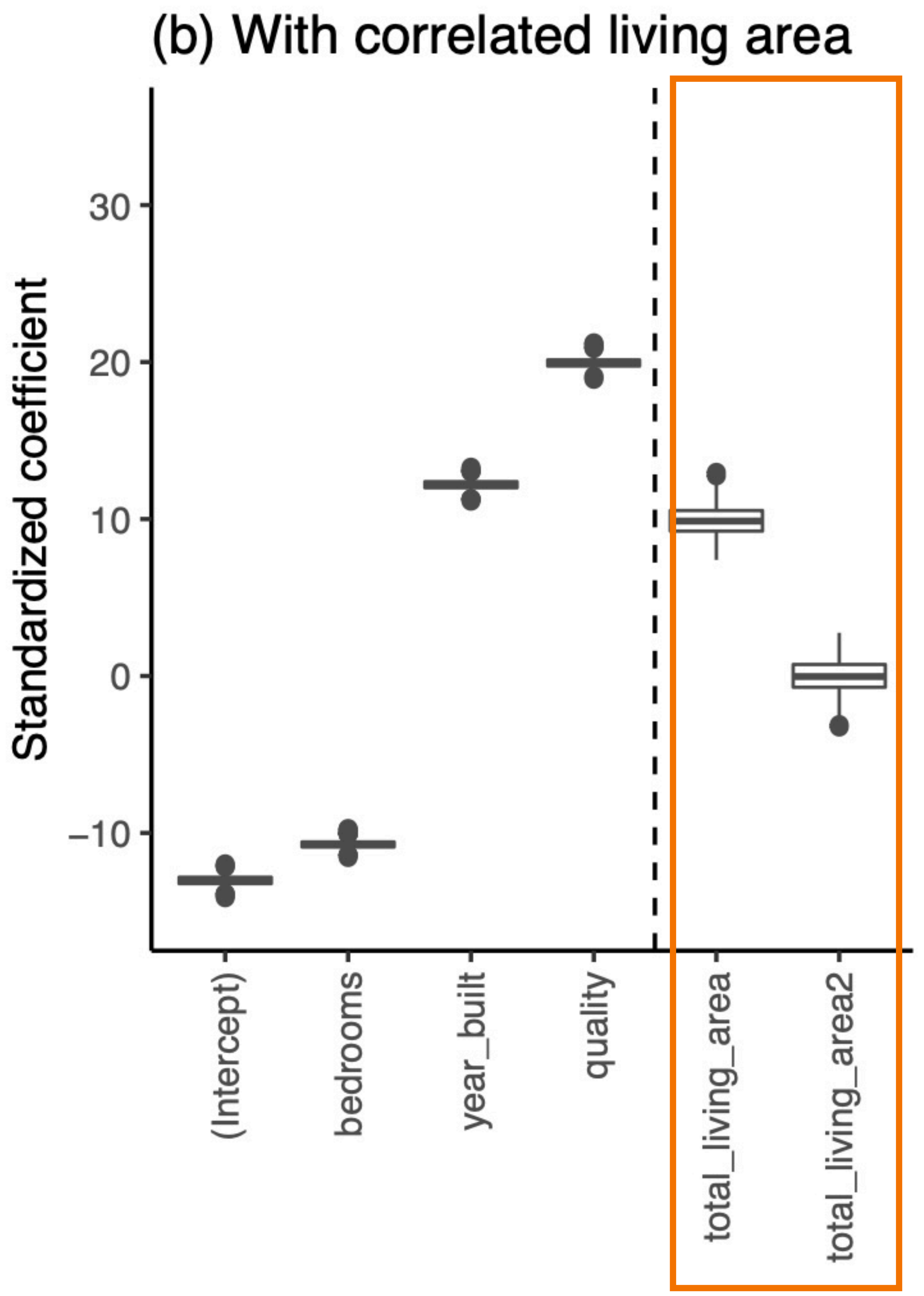
I created a slightly modified (highly correlated) version of the living area variable

I created a bunch of bootstrapped training datasets and recomputed the linear fit…

Collinearity

$$\text{predicted price} = b_0 + b_1 \textbf{area} + \qquad b_3 \text{quality} + b_4 \text{year} + b_5 \text{bedrooms}$$





The coefficient of the area variables is "shared" across the two variables, and they are also less stable (more variable)

Collinearity

If there are features that are highly correlated with one another (highly collinear), then your algorithmic fit might be unstable.

- In the case of perfect collinearity, the design matrix (X) will not be invertible.

- If you have highly correlated features, you should combine them into a single variable or remove the additional variables (keep only one).

Overfitting

Creating a fit that is too complex (has too many features) can lead to **overfitting**

- The algorithm will learn too precisely the specific nuances of the training data and will no longer be able to generalize well to new data

- While using all of the features you have available to you will often lead to the most accurate results, sometimes choosing a sensible subset will lead to more accurate and generalizable results.

- If you have two predictive fits each with similar prediction accuracy (e.g. similar MSE or R-squared values), choose the one that is simpler (i.e., has fewer terms in it)

Feature selection: correlation screening

The full Ames data has over 80 predictive features. How could we go about choosing which ones to be in our linear fit?

One feature selection option for numeric features: **correlation screening**

1. Compute the correlation of each numeric feature with the <u>response</u>
2. Keep only the features whose correlation is above some threshold

Feature selection and feature engineering

Let’s implement the correlation-based feature selection technique

ls_feature_select.R