CS/ENGR M148 L4:
Multiple Linear Regression and OLS

Sandra Batista

Administrative News

If you are still forming teams or modified your team, we will open a new form for you to state interests and reopen the team contract assignment**. <u>Announcement is on BruinLearn.</u>**

This week in discussion section:

Lab on simple regression

Project Data Check-in: Your team will need to demonstrate some data cleaning and EDA. How can you use EDA to help you plan for prediction and choose variables for simple linear regression?

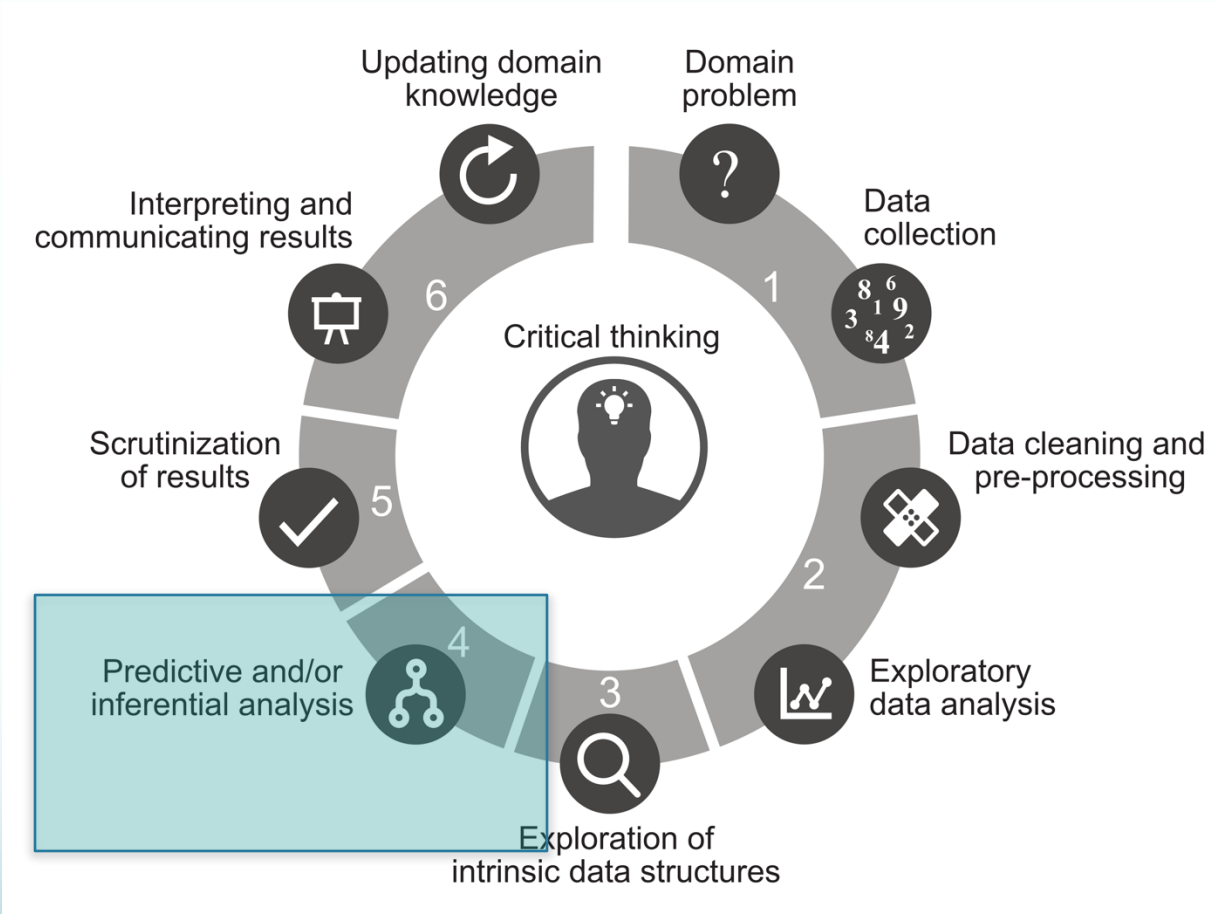
Projects

1. Projects will be graded on how well they demonstrate mastery of the methods taught in class and discussions.
2. You may choose your own data set or a data set supported by the course staff.
3. Team contract 5% - This week during discussion. A sample contract will be made available. Team contracts can be updated until 11:59 pm PT on 10/11/24
4. Project discussion check-ins: 30%, 6x5%
5. Final project code: 25%
6. Final project report: 40%

Join our slido for the week…

https://app.sli.do/event/kJ89kkneBvwrBoxTVCpmcK

Data Science Life Cycle (DSLC)

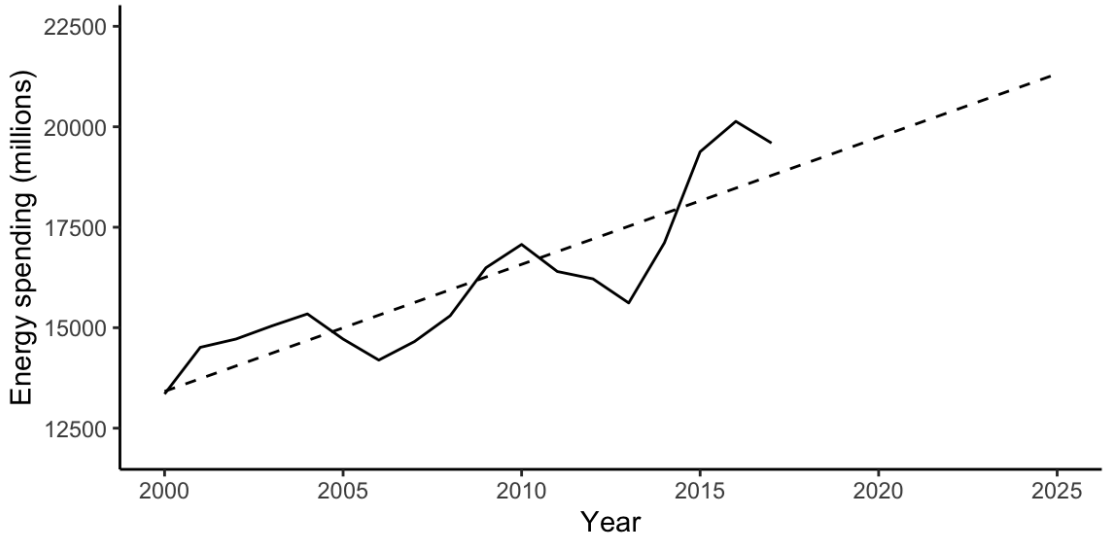


[Yu, Barter 2024]

DSLC Step 4: Predictive Analysis

In **prediction problems** our goal is to use past or current observable data to predict something about future unseen data.

Machine learning methods for prediction include **classification and regression.**



The techniques used are **supervised learning algorithms.**

[Yu, Barter 2024]

Today's Learning Objectives

Students will be able to:

- Review: Identify **predictive problems**
- Review: Plan for **prediction** using **sampling**
- Understand the relationship between **least squares and correlation**
- Evaluate the **fit** of a linear model using L1 and L2 loss functions
- Formulate **multiple linear regression problems** for **Ordinary Least Squares**

The Modeling Process

1. Choose a model	*How should we represent the world?*
2. Choose a loss function	How do we quantify prediction error?
3. Fit the model	How do we choose the best parameters of our model given our data?
4. Evaluate model performance	How do we evaluate whether this process gave rise to a good model?

- Regression: A process for modeling the relationship between variables of interest

Prediction problems

- The goal of a **prediction problem** is to predict the value of a response variable whose value is *unobserved* in future data.
- The variable being predicted is called the **response variable.** Last class our response variable was home sale price
- Variables used in the model to create the prediction are called **predictor variables (predictors, predictive features, covariates, or attributes).** Last class our predictor was living area.

[Yu, Barter 2024]

Prediction algorithms

- A **predictive algorithm** aims to predict the value of a response variable based on the values of predictor variables (also known as covariates or predictive features).
- Today we'll focus on Least Squares Algorithm in more general case of multiple predictor variables

[Yu, Barter 2024]

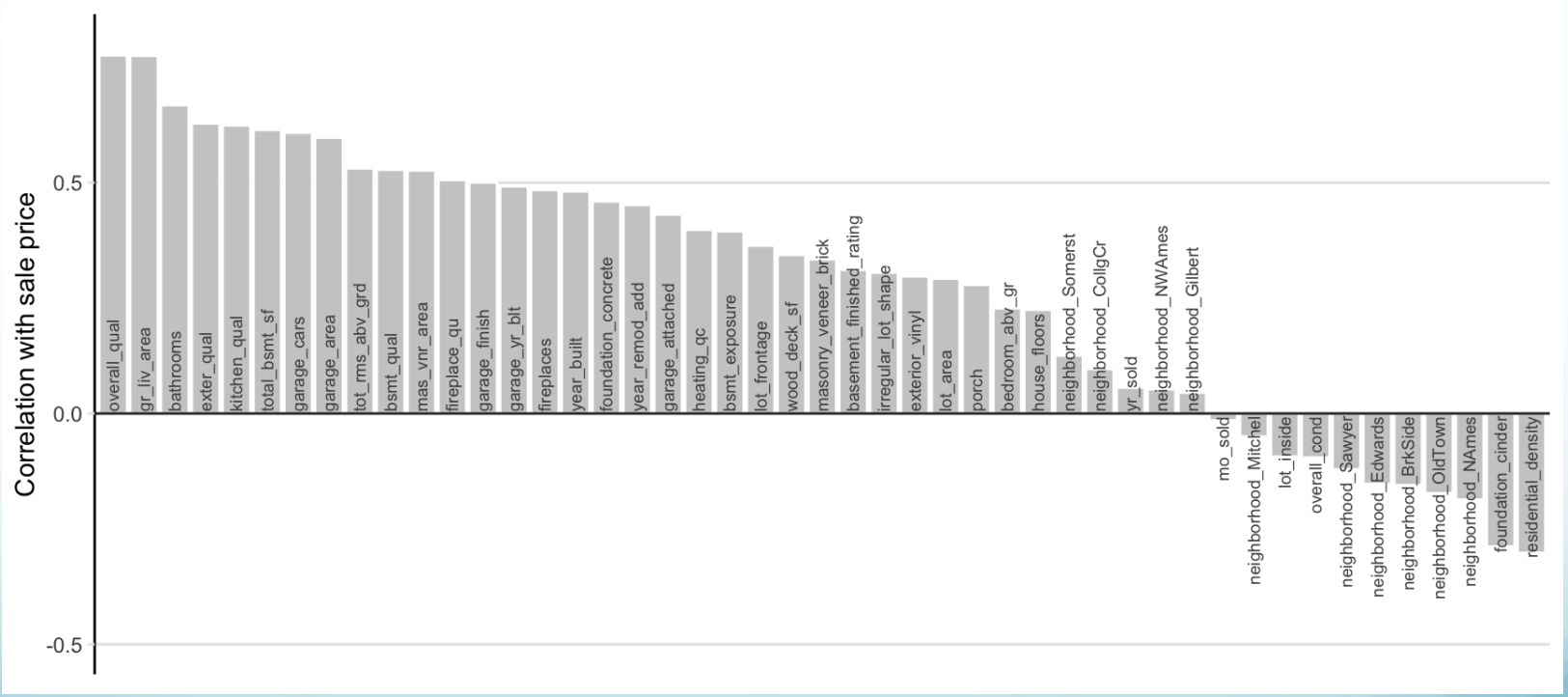
Define Predictors

- How do you know what variables to use as predictors?
- Example: predicting the sale price of a house (the response variable predictive features: house size, age, condition
- **Use domain knowledge**
- **Use EDA** to find a small set of high quality predictors

[Yu, Barter 2024]

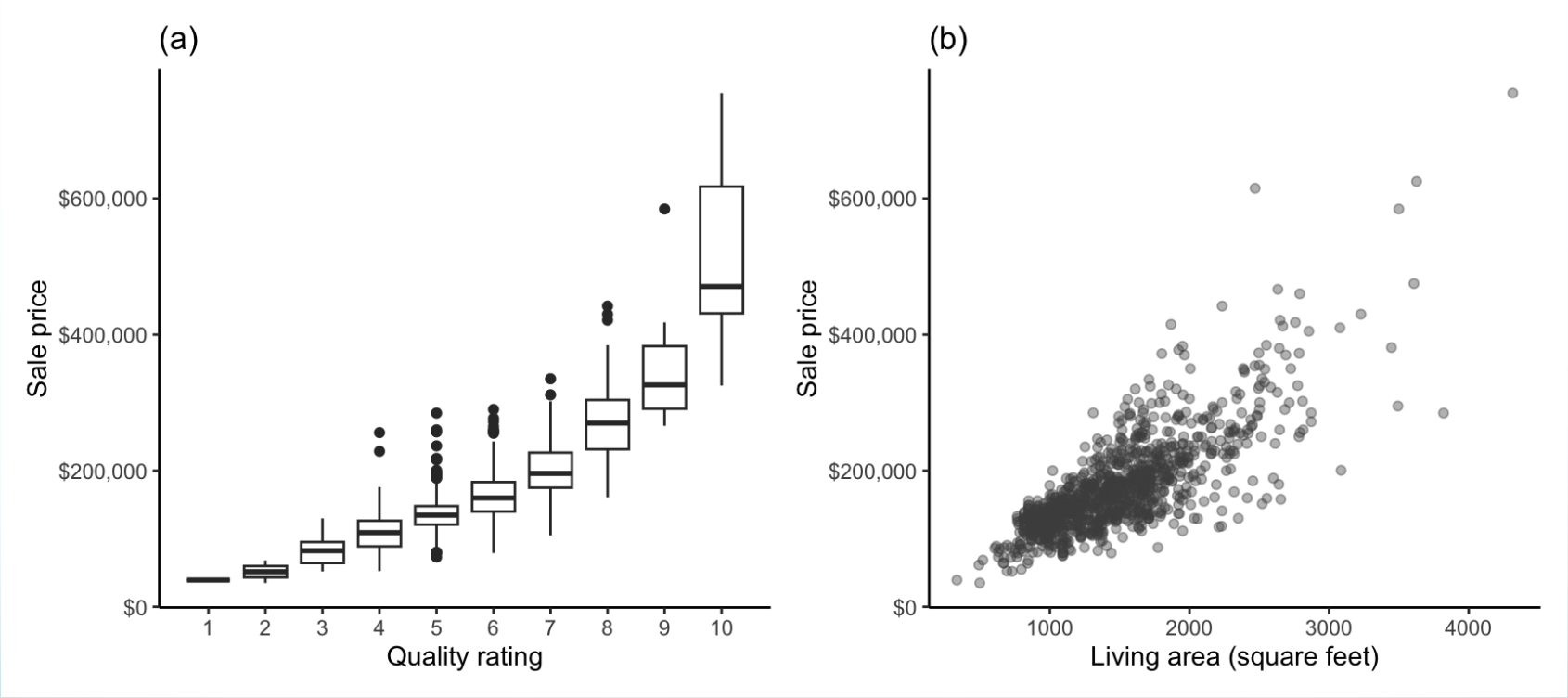
EDA for house prices

Correlation screening: keep only the predictive features that are most correlated with the response



[Yu, Barter 2024]

EDA for house prices



[Yu, Barter 2024]

Today's Learning Objectives

Students will be able to:

- ✓ Review: Identify **predictive problems**
- ✗ Review: Plan for **prediction** using **sampling**
- ✗ Understand the relationship between **least squares and correlation**
- ✗ Evaluate the **fit** of a linear model using L1 and L2 loss functions
- ✗ Formulate **multiple linear regression problems** for **Ordinary Least Squares**

Plan for Predictability

- We need to make sure that we have a data set for training our model and a data set for validating our model.
- If possible it is useful to have a testing data set to evaluate the final model.
- A single data set can be partitioned into a **training set (60%), validation set (20%), test set(20%)**

[Yu, Barter 2024]

Ways to Partition Data

- **Time-based split:** Partition data into subsets based on time collected. Example: home sales data from 2007-2010 for training and then use data from 2011-2014 for validation and 2015-2016 for testing.
- **Group-based split**: Partition the data based on logical subsets of its origin. Example: home sales from Des Moines, IA as training and home sales from Ames, IA for validation
- **Random sampling of data**
- **Stratified sampling:** Split data into groups such as classes and randomly sample within groups

[Yu, Barter 2024]

Sampling

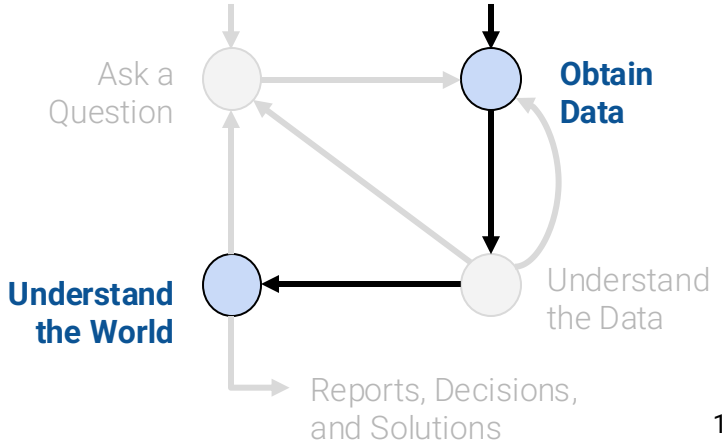
A **sample** is (usually) a subset of the population.

- Samples are often used to make **inferences about the population**.

- How you draw the sample will affect your accuracy.

Sources of error:

- **chance error**: random samples can vary from what is expected, in any direction.
- **bias**: a systematic error in one direction.
  - Could come from our sampling scheme and survey methods.



Probability Sample (aka Random Sample)

Why sample at random?

1. To get more representative samples → **reduce bias**
   - However, the **choice of randomization** can still introduce bias.
2. More importantly, with random samples we can **estimate** the **bias** and **chance error** → **quantify uncertainty**

For a **probability sample**,

- We have to be able to provide the **chance** that any specified **set** of individuals will be in the sample.
- All individuals in the population **need not** have the same chance of being selected.
- Because we know all the probabilities, we will be able to **estimate the errors**.

Common random sampling schemes

A **simple random sample (SRS)** is a sample drawn **uniformly** at random **without** replacement.

- **Every individual (and subset of individuals) has the same chance of being selected.**
- Every pair has the same chance as every other pair.
- Every triple has the same chance as every other triple.
- And so on.

A **uniform random sample with replacement** is a sample drawn **uniformly** at random **with** replacement.

- Similar to SRS but some individuals in the population might get picked more than once.
- **Easier to compute probabilities then SRS**
  - Approximation of large SRS

A raffle could use either sampling scheme, depending on if winners are eligible for multiple prizes.

Example Scheme 1: Probability Sample

Suppose I have 3 TA's (**A**lan, **B**ennett, **C**eline):
I decide to sample 2 of them as follows:

- I choose **A** with probability 1.0
- I choose either **B** or **C**, each with probability 0.5.

All subsets of 2:	{**A**, **B**}	{**A**, **C**}	{**B**, **C**}
Probabilities:	0.5	0.5	0

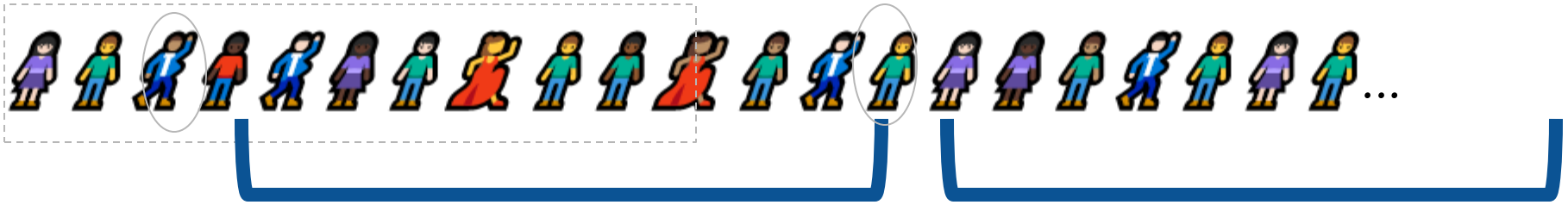
This is a **probability sample**

- Of the 3 people in the population, I know the chance of getting each subset.
  - This scheme does not see the entire population!
  - My estimate using the single sample I take has some **chance error** depending on if I see AB or AC.
  - This scheme **biases** towards A's response

Example Scheme 2: Simple Random Sample?

We have the following sampling scheme:

- A class roster has 1100 students listed alphabetically.
- Pick one of the first 10 students on the list at random (e.g. Student 3).
- To create your sample, take that student and every 10th student listed after that (e.g. Students 8, 18, 28, 38, etc).



1. Is this a probability sample?

2. Does each student have the same probability of being selected?

3. Is this a simple random sample?

Random Sampling Code

```
## this code would define the training, validation, and test set equivalent

 ames_train = ames.query("`Mo Sold` <= @split_date_month & `Yr Sold` <=
@split_date_year")

## filter to houses not in training set

 ames_val = ames.query("~PID.isin(@ames_train.PID)")

## randomly select half of the houses for the validation set

ames_val = ames_val.sample(round(len(ames_val.index)*0.5), random_state=3789)

## filter to houses not in training and validation sets for the test set

 ames_test = ames.query("~PID.isin(@ames_train.PID) & ~PID.isin(@ames_val.PID)")
```

Using sklearn train_test_split

```
from sklearn.model_selection import train_test_split

df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data', header=None)

df = df.iloc[:,:2]

df.columns = ['Class', 'Alcohol']

X, y = df.iloc[:, 1:].values, df.iloc[:, 0].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Today's Learning Objectives

Students will be able to:

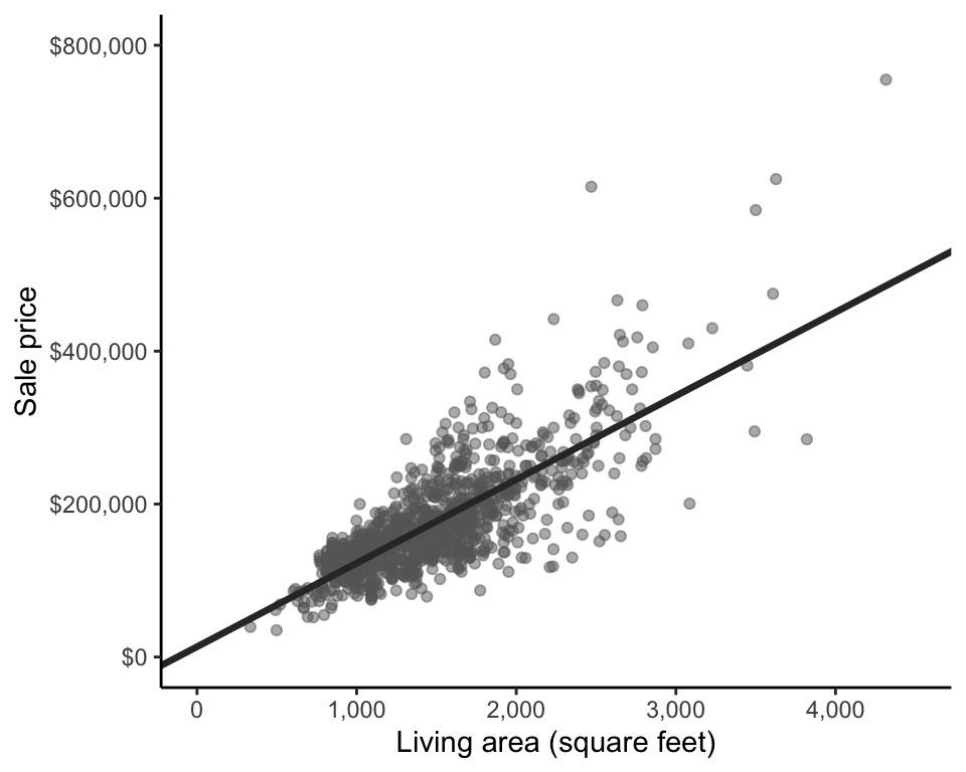
- ✓ Review: Identify **predictive problems**
- ✓ Review: Plan for **prediction** using **sampling**
- Understand the relationship between **least squares and correlation**
- Evaluate the **fit** of a linear model using L1 and L2 loss functions
- Formulate **multiple linear regression problems** for **Ordinary Least Squares**

Visualize Predictive Relationship

Fitted line for linear relationship but is it "best"?

Caution: Are there any **confounders?**

A **confounder** is a common cause of affecting both response and predictor variables.



[Yu, Barter 2024]

The Regression Line

From Data 8 (textbook):

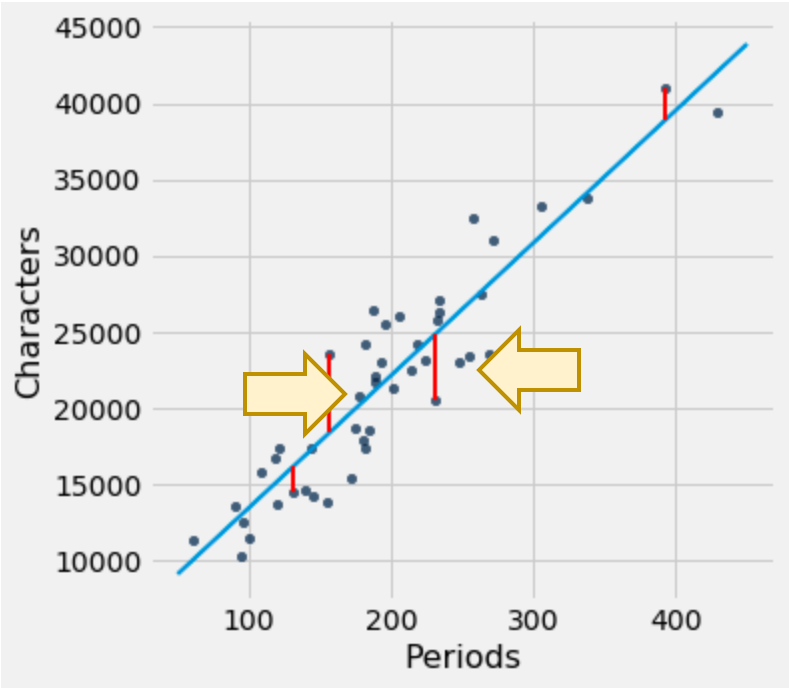
> The **regression line** is the unique straight line that minimizes the **mean squared error** of estimation among all straight lines.

$$\text{slope} = r \times \frac{\text{SD of } y}{\text{SD of } x}$$

$$\text{intercept} = \text{average of } y - \text{slope} \times \text{average of } x$$

$$\text{regression estimate} = \text{intercept} + \text{slope} \times x$$

$$\text{residual} = \text{observed } y - \text{regression estimate}$$



For every chapter of the novel *Little Women*, Estimate the **# of characters** $\hat{y}$ based on the **number of periods** x in that chapter.

[Data 8 Review] The Regression Line

From Data 8 (textbook):

> The **regression line** is the unique straight line that minimizes the **mean squared error** of estimation among all straight lines.

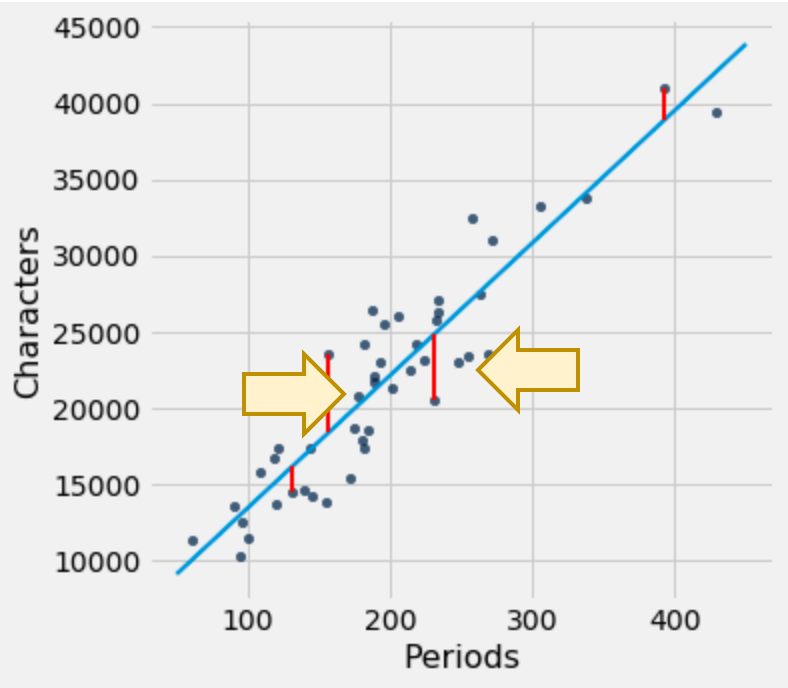
correlation

$$\text{slope} = r \times \frac{\text{SD of } y}{\text{SD of } x}$$

$$\text{intercept} = \text{average of } y - \text{slope} \times \text{average of } x$$

$$\text{regression estimate} = \text{intercept} + \text{slope} \times x$$

$$\text{residual} = \text{observed } y - \text{regression estimate}$$



For every chapter of the novel *Little Women*, Estimate the **# of characters** $\hat{y}$ based on the **number of periods** x in that chapter.

Covariance of random variables

- The covariance of two random variables X and Y is
  $cov[X,Y] = E[\,(X-E[X])(Y-E[Y])\,]$
  $= E[XY]-E[X]E[Y]$.
- $cov[X,X] = var[X]$.
- If X and Y are independent random variables, then the covariance is zero:
  Proof: Independence means that $E[XY]=E[X]E[Y]$.
  The converse is not true.

Sample covariance

- The sample covariance formula follows from *cov[X,Y] = E[XY]-E[X]E[Y]* :

$$\frac{1}{n}\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}$$

- For unbiased estimator 1/n is replaced by 1/(n-1)

Correlation

- The covariance of two random variables *X* and *Y* is in units that are a product of those of *X* and *Y*. To obtain a dimensionless number, the covariance can be divided by the product of the standard deviation of *X* and the standard deviation of *Y*. This is called the ***correlation coefficient***:

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}.$$

Other names include Pearson's product-moment correlation coefficient, Pearson's coefficient, or Pearson's correlation.

Sample correlation coefficient

- The sample correlation coefficient, denoted by *r*, is an estimate of the (population) Pearson correlation. There are several equivalent expressions; the analogue of the sample covariance formula we used is:

$$r = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}.$$

- Note that while Pearson's correlation coefficient lies between -1 and 1 (inclusive), sampling error will reduce the range of *r*.

[Data 8 Review] Correlation

From Data 8 (textbook):

> The **correlation** r is the **average** of the **product** of x and y, both measured in standard units.

$$r = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{\sigma_x}\right)\left(\frac{y_i - \bar{y}}{\sigma_y}\right)$$

Define the following:

data $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$

means $\bar{x}, \bar{y}$ standard deviations σ_x, σ_y

- x_i in standard units: $\frac{x_i - \bar{x}}{\sigma_x}$
- r is also known as Pearson's correlation coefficient

.

- Note: for categorical variables, you may need to use Spearman's correlation. This will be covered in discussion sections.

[Data 8 Review] Correlation

From Data 8 (textbook):

> The **correlation** r is the average of the product of x and y, both measured in standard units.

Define the following:

data $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$

means $\bar{x}, \bar{y}$ standard deviations σ_x, σ_y

$$r = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{\sigma_x} \right) \left(\frac{y_i - \bar{y}}{\sigma_y} \right)$$

- Correlation measures the strength of a **linear association** between two variables.
- It ranges **between -1 and 1.**
  - r = 1 indicates perfect linear association; r = -1 perfect negative association.
  - The closer r is to 0, the weaker the linear association is.
- It says nothing about **causation** or **non-linear association.**
  - Correlation does not imply causation.
  - When r = 0, the two variables are **uncorrelated**. However, they could still be related through some non-linear relationship.

[Data 8 Review] Correlation

From Data 8 (textbook):

> The **correlation** r is the average of the product of x and y, both measured in standard units.

Define the following:

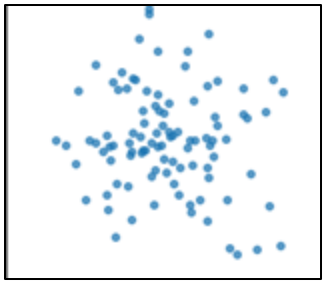
data $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$

means $\bar{x}, \bar{y}$ standard deviations σ_x, σ_y

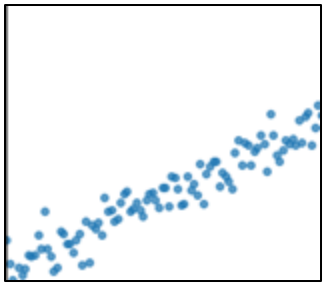
$$r = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{\sigma_x} \right) \left(\frac{y_i - \bar{y}}{\sigma_y} \right)$$

Correlation measures the strength of a **linear association** between two variables.

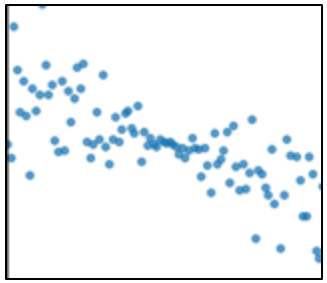
$$|r| \leq 1$$



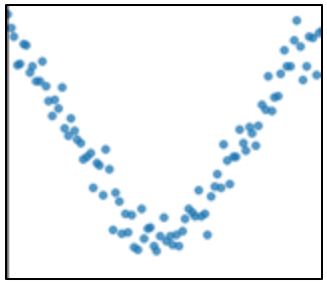
r = −0.121



r = 0.951



r = −0.723



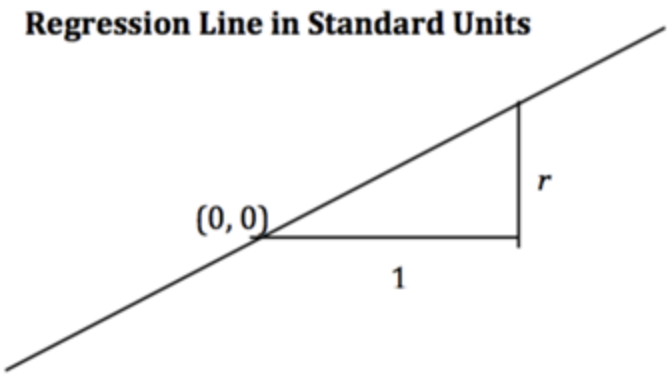
⚠ **r = 0.056**

[Data 8 Review] The Regression Line

- When the variables x and y are measured in **standard units**, the regression line for predicting y based on x has slope r passes through the origin and the equation will be:

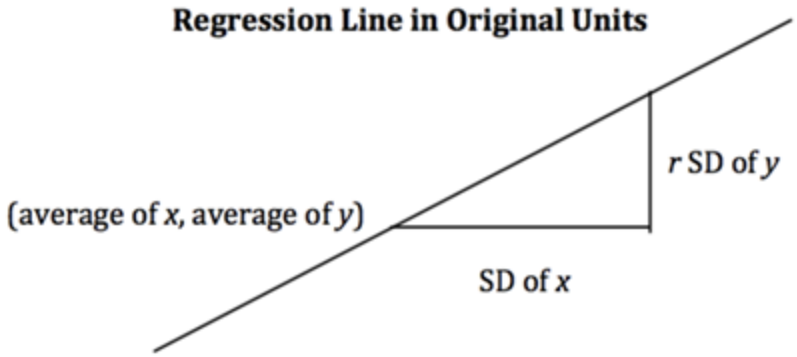
$$\hat{y} = r \times x$$

(both measured in standard units)



- In the original units of the data, this becomes:

$$\frac{\hat{y} - \bar{y}}{\sigma_y} = r \times \frac{x - \bar{x}}{\sigma_x}$$



[Data 8 Review] The Regression Line

$$\frac{\hat{y} - \bar{y}}{\sigma_y} = r \times \frac{x - \bar{x}}{\sigma_x}$$

$$\hat{y} = \sigma_y \times r \times \frac{x - \bar{x}}{\sigma_x} + \bar{y}$$

$$\hat{y} = \left(\frac{r\sigma_y}{\sigma_x}\right) \times x + \left(\bar{y} - \frac{r\sigma_y}{\sigma_x}\bar{x}\right)$$

Recall regression line equation is defined as:

$$\hat{y} = \hat{a} + \hat{b}x$$

slope: $r\frac{SD\ of\ y}{SD\ of\ x} = r\frac{\sigma_y}{\sigma_x}$

intercept: $\bar{y} - slope \times \bar{x}$

Error for the i-th data point: $e_i = y_i - \hat{y}_i$

Today's Learning Objectives

Students will be able to:

- ✓ Review: Identify **predictive problems**
- ✓ Review: Plan for **prediction** using **sampling**
- ✓ Understand the relationship between **least squares and correlation**
- Evaluate the **fit** of a linear model using L1 and L2 loss functions
- Formulate **multiple linear regression problems** for **Ordinary Least Squares**

Least Absolute Deviation

L1 loss function or absolute value loss function:

$$\frac{1}{n}\sum_{i=1}^{n}\left|\text{observed response}_i - \text{predicted response}_i\right|,$$

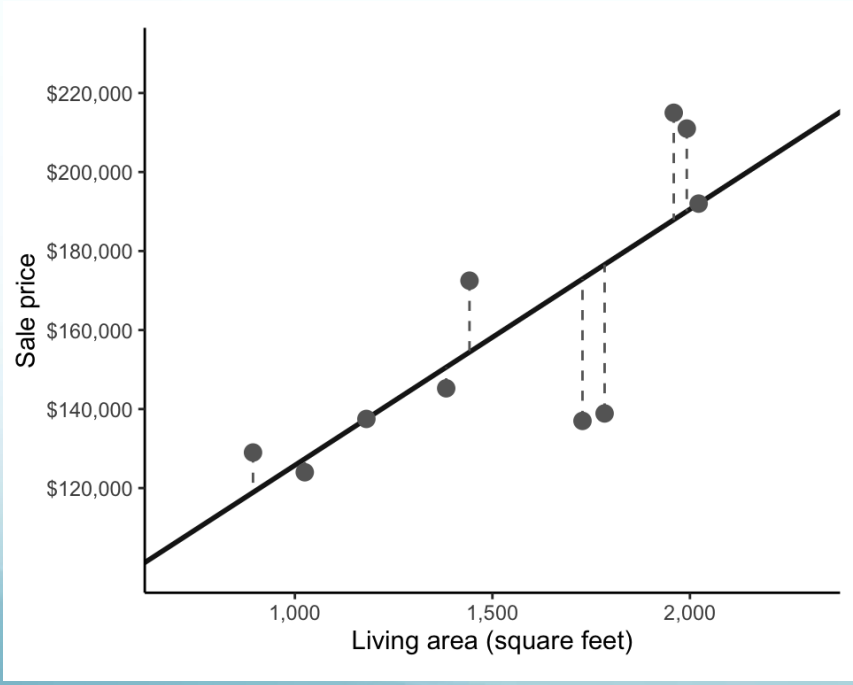
$$\frac{1}{n}\sum_{i=1}^{n}|\text{observed price}_i - (b_0 + b_1 \times \text{area}_i)|.$$

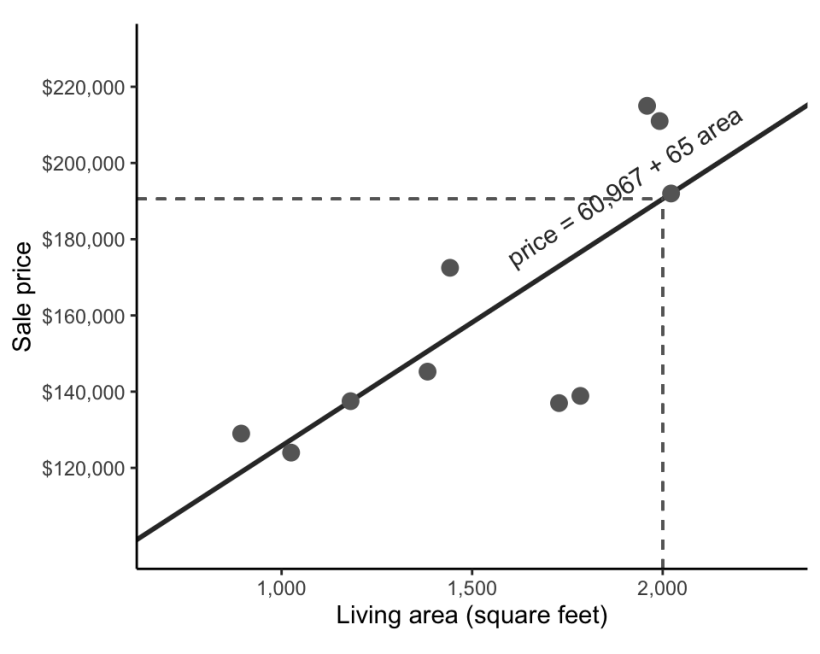
$$\frac{1}{n}\sum_{i=1}^{n}|y_i - (b_0 + b_1 x_i)|.$$

[Yu, Barter 2024]

Least Absolute Deviation

The Least Absolute Deviation (LAD) algorithm generates a fitted line to minimize the absolute value (or L1) loss function





[Yu, Barter 2024]

Least Squares Loss Function

L2 loss function or squared loss:

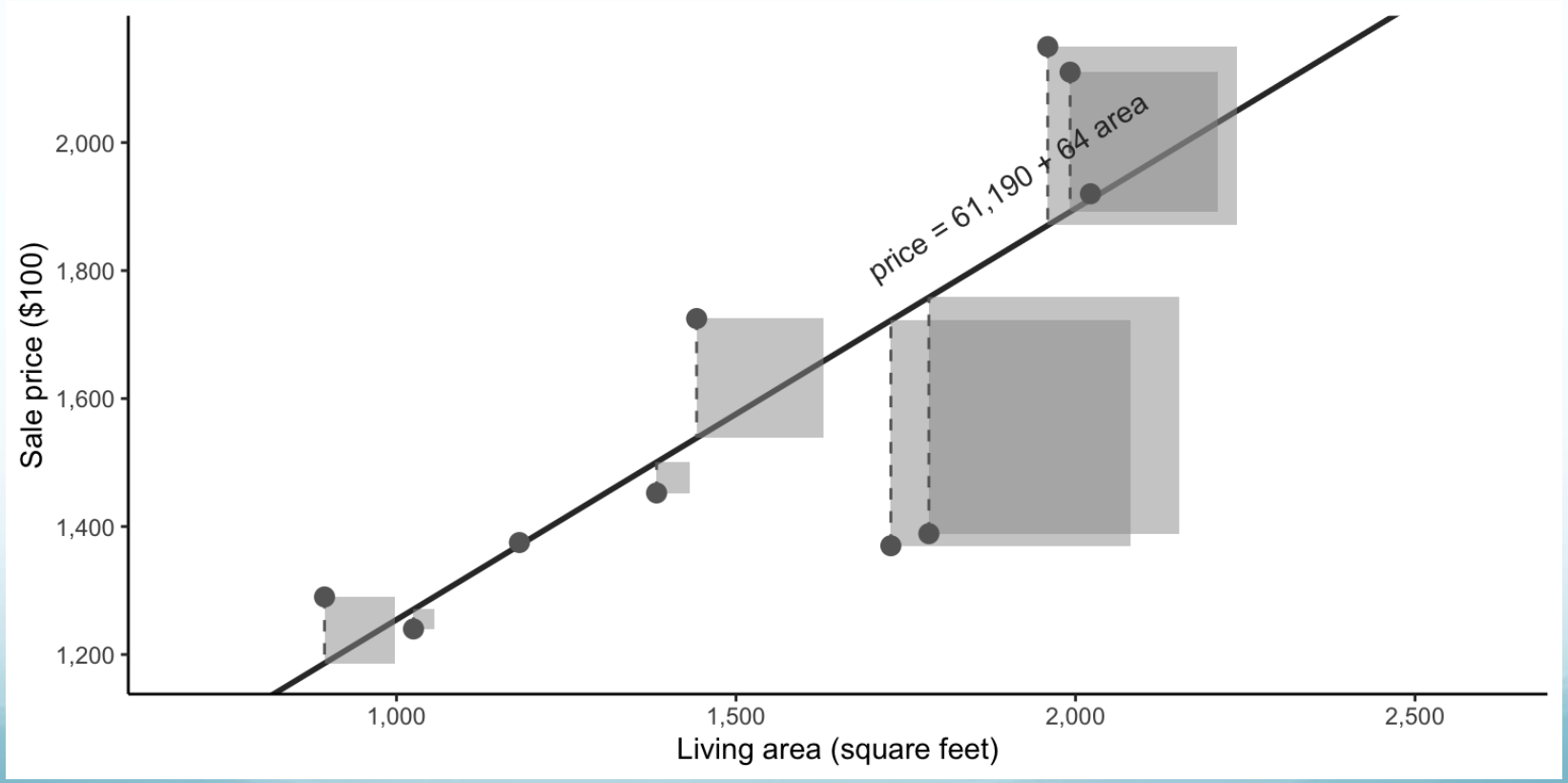
$$\frac{1}{n}\sum_{i=1}^{n}(\text{observed response}_i - \text{predicted response}_i)^2.$$

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - (b_0 + b_1 x_i))^2.$$

[Yu, Barter 2024]

Least Squares

The **Least Squares (LS) algorithm** generates a fitted line by minimizing the squared (or L2) loss function



[Yu, Barter 2024]

Empirical Risk is Average Loss over Data

We care about how bad our model's predictions are for our entire data set, not just for one point.

A natural measure, then, is of the **average loss** (aka **empirical risk**) across all points.

Given data $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$:

$$\hat{R}(\theta) = \frac{1}{n} \sum_i L(y_i, \hat{y}_i)$$

The average loss on the sample tells us how well the model fits the data (not the population).

Empirical Risk is Average Loss over Data

We care about how bad our model's predictions are for our entire data set, not just for one point.

A natural measure, then, is of the **average loss** across all points.

Given data $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$:

$$\hat{R}(\theta) = \frac{1}{n} \sum_i L(y_i, \hat{y}_i)$$

L2 loss → **Mean Squared Error (MSE)** $\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

L1 loss → **Mean Absolute Error (MAE)** $\frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$

From the L04 code

```
# calculate the rMSE, MAE, MAD, correlation, and R2 of the true price with the LS and
LAD predictions

print('LS rMSE:', np.sqrt(mean_squared_error(pred_train_df['true'],
pred_train_df['ls_pred'])))

print('LS MAE:', mean_absolute_error(pred_train_df['true'], pred_train_df['ls_pred']))

print('LS MAD:', np.median(np.abs(pred_train_df['true'] - pred_train_df['ls_pred'])))

print('LS correlation:', np.corrcoef(pred_train_df['true'], pred_train_df['ls_pred'])[0, 1])

print('LS R2:', r2_score(pred_train_df['true'], pred_train_df['ls_pred']))
```

But what does it all mean??

mean_squared_error

```
print('LS rMSE:', np.sqrt(mean_squared_error(pred_train_df['true'],
pred_train_df['ls_pred'])))
```

Note better to use **root_mean_squared_error**

What is **root mean squared error, rMSE**?

mean_absolute error

```
print('LS MAE:', mean_absolute_error(pred_train_df['true'],
pred_train_df['ls_pred']))
```

What is **mean absolute error, MAE**?

Median Absolute Deviation

```
print('LS MAD:', np.median(np.abs(pred_train_df['true'] -
pred_train_df['ls_pred'])))
```

What is the **median absolute deviation, MAD?**

Correlation

```
print('LS correlation:', np.corrcoef(pred_train_df['true'],
pred_train_df['ls_pred'])[0, 1])
```

What is the **correlation**?

R2 score

```
print('LS R2:', r2_score(pred_train_df['true'], pred_train_df['ls_pred’]))
```

What is **R2 score**?

Today's Learning Objectives

Students will be able to:

- ✓ Review: Identify **predictive problems**
- ✓ Review: Plan for **prediction** using **sampling**
- ✓ Understand the relationship between **least squares and correlation**
- ✓ Evaluate the **fit** of a linear model using L1 and L2 loss functions
- Formulate **multiple linear regression problems** for **Ordinary Least Squares**

Statistical representation of data

- *rows* for observations, and the *columns* for features.
- *n* is used to represent the number of observations and *p* the number of features, so that a data table has size *n x p*.
  - One reason for this convention is the form of regression models, which describe observations as linear combinations of explanatory variables with some added noise using the form:
  - With this matrix notation, *X*, which is also known as the design matrix, has dimensions *n x p*.

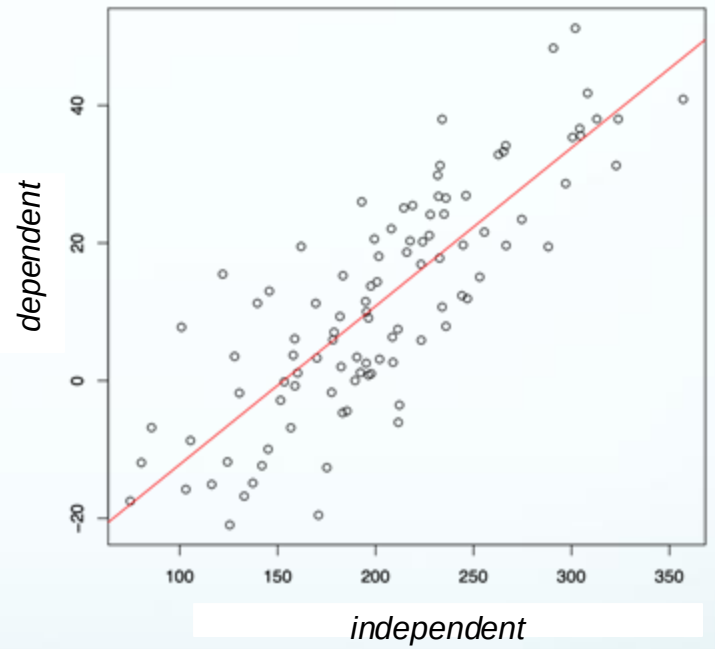
$$y = X\beta + \epsilon.$$

Least squares

- Fit a line of the form *y=mx+b*.
- **Goal:** find a line with the property that the average (vertical) loss between the points and the line is minimized.
- use squared distance for the loss function because its optimization is easier than the alternatives.

$$r_i = y_i - f(x_i, \boldsymbol{\beta}).$$

$$S = \sum_{i=1}^{n} r_i^2.$$



Solving the least squares problem

- Method 1: (Multivariable) calculus.

$$f(m,b) = \sum_{i=1}^{n} (y_i - (mx_i + b))^2.$$

- The task is to minimize *f(m,b)* over the parameters *m* and *b*. The square is helpful because the derivatives of *f* with respect to *m* and *b* are linear.

- Compute the two partial derivatives (with respect to *m* and *b*), set them equal to 0, …

Solving the least squares problem

- Method 2: Linear algebra.
- Consider a column vector formed from the dependent variables, i.e. $Y = (y_1,\ldots,y_n)^T$, as a point in an n-dimensional vector space.
- Least squares optimization problem is equivalent to finding the nearest point on a subspace spanned by a column matrix X defined from the dependent variables

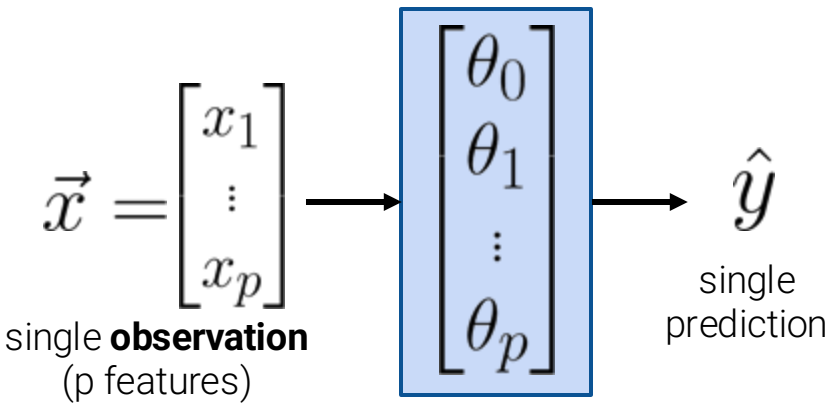
- Findthe value β that minimizes $(\|X\beta - Y\|_2)^2$; the minimal β is denoted $\hat{\beta}$.
- Using orthogonality, the solution emerges naturally as $(X^TX)^{-1}X^TY$.

Multiple Linear Regression

Define the **multiple linear regression** model:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$

Predicted value of y



single **observation** (p features)

single prediction

Multiple Linear Regression

Define the **multiple linear regression** model:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$

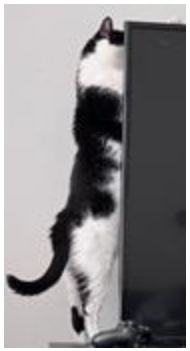
Predicted value of y

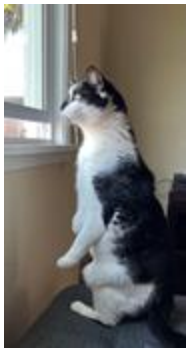
$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix} \rightarrow \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_p \end{bmatrix} \rightarrow \hat{y}$$

single **observation** (p features)

single prediction

Example: Predict cat's ages $\hat{y}$ as a linear model of 2 features: length x_1 **and** weight x_2 .





$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

intercept | parameter for length | parameter for weight

NBA 2018-2019 Dataset

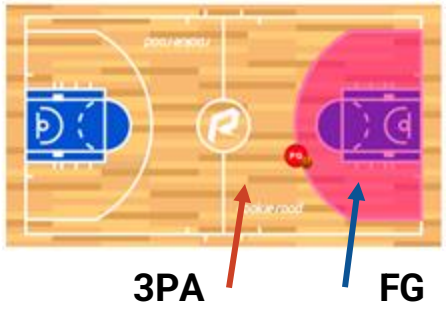
How many points does an athlete score per game?
PTS (average points/game)

To name a few factors:

- **FG**: average # 2 point field goals
- **AST**: average # of assists
- **3PA**: average # 3 point field goals attempted

	FG	AST	3PA	PTS
1	1.8	0.6	4.1	5.3
2	0.4	0.8	1.5	1.7
3	1.1	1.9	2.2	3.2
4	6.0	1.6	0.0	13.9
5	3.4	2.2	0.2	8.9
6	0.6	0.3	1.2	1.7

Rows correspond to individual players.



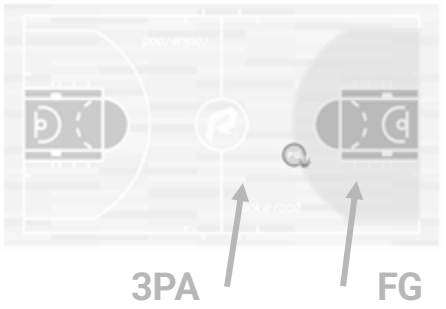
assist: a pass to a teammate that directly leads to a goal

Multiple Linear Regression Model

How many points does an athlete score per game?
PTS (average points/game)

To name a few factors:

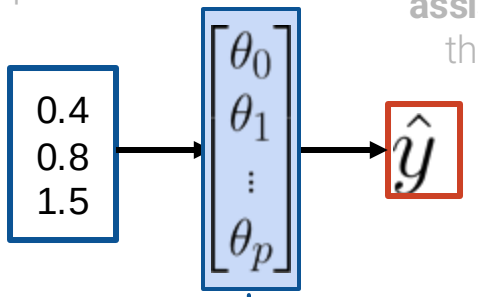
- **FG**: average # 2 point field goals
- **AST**: average # of assists
- **3PA**: average # 3 point field goals attempted



assist: a pass to a teammate that directly leads to a goal

	FG	AST	3PA	PTS
1	1.8	0.6	4.1	5.3
2	0.4	0.8	1.5	1.7
3	1.1	1.9	2.2	3.2
4	6.0	1.6	0.0	13.9
5	3.4	2.2	0.2	8.9
6	0.6	0.3	1.2	1.7

Rows correspond to individual players.



$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$
$$= \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

FG AST 3PA

From One Feature to Many Features

Dataset for SLR

x	y
x_1	y_1
x_2	y_2
$\vdots$	$\vdots$
x_n	y_n

	FG	PTS
1	1.8	5.3
2	0.4	1.7
3	1.1	3.2
4	6.0	13.9
5	3.4	8.9
...	...	...

Dataset for Multiple Linear Regression

$x_{:,1}$	$x_{:,2}$		$x_{:,p}$	y
x_{11}	x_{12}	$\cdots$	x_{1p}	y_1
x_{21}	x_{22}	$\cdots$	x_{2p}	y_2
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x_{n1}	x_{n2}	$\cdots$	x_{np}	y_n

	FG	AST	3PA	PTS
1	1.8	0.6	4.1	5.3
2	0.4	0.8	1.5	1.7
3	1.1	1.9	2.2	3.2
4	6.0	1.6	0.0	13.9
5	3.4	2.2	0.2	8.9
...	...	...	...	...

From One Feature to Many Features

Dataset for Multiple Linear Regression

$x_{:1}$	$x_{:2}$		$x_{:p}$	y
x_{11}	x_{12}	$\cdots$	x_{1p}	y_1
x_{21}	x_{22}	$\cdots$	x_{2p}	y_2
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
x_{n1}	x_{n2}	$\cdots$	x_{np}	y_n

Feature 2

$$\{x_{12}, x_{22}, \ldots x_{n2}\}$$

Observation i

$$\{x_{i1}, x_{i2}, \ldots x_{ip}, y_i\}$$

	FG	AST	3PA	PTS
1	1.8	0.6	4.1	5.3
2	0.4	0.8	1.5	1.7
3	1.1	1.9	2.2	3.2
4	6.0	1.6	0.0	13.9
5	3.4	2.2	0.2	8.9
...	...	...	...	...

Model

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$

$$\begin{cases} \hat{y}_1 = \theta_0 + \theta_1 x_{11} + \theta_2 x_{12} + \cdots + \theta_p x_{1p} \\ \hat{y}_2 = \theta_0 + \theta_1 x_{21} + \theta_2 x_{22} + \cdots + \theta_p x_{2p} \\ \vdots \\ \hat{y}_n = \theta_0 + \theta_1 x_{n1} + \theta_2 x_{n2} + \cdots + \theta_p x_{np} \end{cases}$$

[Linear Algebra] Vector Dot Product

The **dot product (or inner product)** is a vector operation that

- Can only be carried out on two vectors of the **same length,**
- Sums up the products of the corresponding entries of the two vectors, and
- Returns a single number.

$$\vec{u} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \qquad \vec{v} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\vec{u} \cdot \vec{v} = \vec{u}^\top \vec{v} = \vec{v}^\top \vec{u}$$
$$= 1 \cdot 1 + 2 \cdot 1 + 3 \cdot 1$$
$$= 6$$

"u dot v"

Sidenote (not in scope): We can interpret dot product geometrically:

- It is the product of three things: the **magnitude** of both vectors, and the **cosine** of the angles between them. $\vec{u} \cdot \vec{v} = ||\vec{u}|| \cdot ||\vec{v}|| \cdot \cos\theta$
- Another interpretation: 3Blue1Brown

Vector Notation

$$\hat{y} = \theta_0 + \underbrace{\theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p}$$

This part looks a little like a dot product…

We want to collect all the θ_i's into a single vector.

$$= \boxed{\theta_0} + \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} = \theta_0 \cdot 1 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$

😖 What about this one???

Vector Notation

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$

$$= \theta_0 \cdot 1 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_p x_p$$

$$= \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} = x^\top \theta$$

bias term, intercept term

We want to collect all the θ_i's into a single vector.

Matrix Notation

$$\begin{cases} \hat{y}_1 = \theta_0 + \theta_1 x_{11} + \theta_2 x_{12} + \cdots + \theta_p x_{1p} \\ \hat{y}_2 = \theta_0 + \theta_1 x_{21} + \theta_2 x_{22} + \cdots + \theta_p x_{2p} \\ \vdots \\ \hat{y}_n = \theta_0 + \theta_1 x_{n1} + \theta_2 x_{n2} + \cdots + \theta_p x_{np} \end{cases}$$

$$\begin{cases} \hat{y}_1 = x_1^\top \theta & \text{where } x_1^\top = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \end{bmatrix} \text{is datapoint/observation 1} \\ \hat{y}_2 = x_2^\top \theta & \text{where } x_2^\top = \begin{bmatrix} 1 & x_{21} & x_{22} & \dots & x_{2p} \end{bmatrix} \text{is datapoint/observation 2} \\ \vdots \\ \hat{y}_n = x_n^\top \theta & \text{where } x_n^\top = \begin{bmatrix} 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \text{is datapoint/observation n} \end{cases}$$

Matrix Notation

$$\begin{cases} \hat{y}_1 = x_1^\top \theta & \text{where } x_1^\top = [1 \quad x_{11} \quad x_{12} \quad \dots \quad x_{1p}] \text{ is datapoint/observation 1} \\ \hat{y}_2 = x_2^\top \theta & \text{where } x_2^\top = [1 \quad x_{21} \quad x_{22} \quad \dots \quad x_{2p}] \text{ is datapoint/observation 2} \\ \vdots & \\ \hat{y}_n = x_n^\top \theta & \text{where } x_n^\top = [1 \quad x_{n1} \quad x_{n2} \quad \dots \quad x_{np}] \text{ is datapoint/observation n} \end{cases}$$

	FG	AST	3PA	PTS
1	1.8	0.6	4.1	5.3
2	0.4	0.8	1.5	1.7
3	1.1	1.9	2.2	3.2
4	6.0	1.6	0.0	13.9
5	3.4	2.2	0.2	8.9
...	...	...	...	...

For data point/observation 2, we have

$$x_2 = \begin{bmatrix} 1 \\ 0.4 \\ 0.8 \\ 1.5 \end{bmatrix} \quad y_2 = 1.7 \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

$$\begin{aligned} \hat{y}_2 &= x_2^\top \theta \\ &= \theta_0 + \theta_1 \cdot 0.4 + \theta_2 \cdot 0.8 + \theta_3 \cdot 1.5 \end{aligned}$$

Dimension check

$x_2 \in \mathbb{R}^4$ or $\mathbb{R}^{(p+1)}$

$\theta \in \mathbb{R}^4$ or $\mathbb{R}^{(p+1)}$

$y_2 \in \mathbb{R}$ $\hat{y}_2 \in \mathbb{R}$

also called scalars

Matrix Notation

$$\begin{aligned}
\hat{y}_1 &= \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \end{bmatrix} \theta &= x_1^T \theta \\
\hat{y}_2 &= \begin{bmatrix} 1 & x_{21} & x_{22} & \dots & x_{2p} \end{bmatrix} \theta &= x_2^T \theta \\
&\vdots & & \vdots & \vdots \\
\hat{y}_n &= \begin{bmatrix} 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \theta &= x_n^T \theta
\end{aligned}$$

n row vectors, each with dimension **(p+1)**

Expand out each datapoint's (transposed) input

Matrix Notation

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} [1 & x_{11} & x_{12} & \dots & x_{1p}] \\ [1 & x_{21} & x_{22} & \dots & x_{2p}] \\ & & \vdots & & \\ [1 & x_{n1} & x_{n2} & \dots & x_{np}] \end{bmatrix} \theta$$

n row vectors, each with dimension **(p+1)**

Vectorize predictions and parameters to encapsulate all n equations into a single matrix equation.

Matrix Notation

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \mathbb{X} \theta$$

Design matrix with dimensions n x (p + 1)

The Design Matrix $\mathbb{X}$

We can use linear algebra to represent our predictions of all n data points at once.

One step in this process is to stack all of our input features together into a **design matrix**:

$$\mathbb{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ 1 & x_{31} & x_{32} & \dots & x_{3p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}$$

What do the **rows** and **columns** of the design matrix represent in terms of the observed data?

Field Goals, Assists, 3-Point Attempts

Bias	FG	AST	3PA	PTS
1	1.8	0.6	4.1	5.3
1	0.4	0.8	1.5	1.7
1	1.1	1.9	2.2	3.2
1	6.0	1.6	0.0	13.9
1	3.4	2.2	0.2	8.9
...	...	...	...	...
1	4.0	0.8	0.0	11.5
1	3.1	0.9	0.0	7.8
1	3.6	1.1	0.0	8.9
1	3.4	0.8	0.0	8.5
1	3.8	1.5	0.0	9.4

Example design matrix
708 rows x (3+1) cols

The Design Matrix $\mathbb{X}$

We can use linear algebra to represent our predictions of all n data points at once.

One step in this process is to stack all of our input features together into a **design matrix**:

$$\mathbb{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ 1 & x_{31} & x_{32} & \dots & x_{3p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix}$$

A **row** corresponds to one **observation**, e.g., all (p+1) features for datapoint 3

A **column** corresponds to a **feature**, e.g. feature 1 for all n data points

Special all-ones feature often called the **bias/intercept**

Bias	FG (Field Goals)	AST (Assists)	3PA (3-Point Attempts)	PTS
1	1.8	0.6	4.1	5.3
1	0.4	0.8	1.5	1.7
1	1.1	1.9	2.2	3.2
1	6.0	1.6	0.0	13.9
1	3.4	2.2	0.2	8.9
...	...	...	...	...
1	4.0	0.8	0.0	11.5
1	3.1	0.9	0.0	7.8
1	3.6	1.1	0.0	8.9
1	3.4	0.8	0.0	8.5
1	3.8	1.5	0.0	9.4

Example design matrix
708 rows x (3+1) cols

The Multiple Linear Regression Model Using Matrix Notation

We can express our linear model on our entire dataset as follows:

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ 1 & x_{31} & x_{32} & \dots & x_{3p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_p \end{bmatrix}$$

$$\hat{\mathbb{Y}} = \mathbb{X}\theta$$

Prediction vector $\mathbb{R}^n$

Design matrix $\mathbb{R}^{n \times (p+1)}$

Parameter vector $\mathbb{R}^{(p+1)}$

Note that our **true output** is also a vector:

$$\mathbb{Y} \in \mathbb{R}^n$$

Linear in Theta

An expression is "**linear in theta**" if it is a **linear combination** of parameters $\theta = [\theta_0, \theta_1, \ldots, \theta_p]$

.

1. $\hat{y} = \theta_0 + \theta_1(2) + \theta_2(4 \cdot 8) + \theta_3(\log 42)$

2. $\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 x_3 + \theta_3 . \log(x_4)$

3. $\hat{y} = \theta_0 + \theta_1 x_1 + \log(\theta_2) x_2 + \theta_3 \theta_4$

4. $$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 5 & 6 & 7 \\ 1 & 8 & 9 & 0 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

5. $$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} \\ 1 & x_{21} & x_{22} & x_{23} \\ 1 & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

Which of these expressions are linear in theta?

[Linear Algebra] Vector Norms and the L2 Vector Norm

The **norm** of a vector is some measure of that vector's **size/length**.

- The two norms we need to know for Data 100 are the L_1 and L_2 norms (sound familiar?).
- Today, we focus on L_2 norm. We'll define the L_1 norm another day.

For the n-dimensional vector $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$, the **L2 vector norm** is

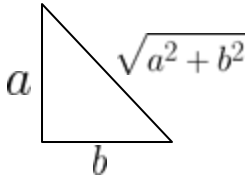
$$||\vec{x}||_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = \sqrt{\sum_{i=1}^{n} (x_i^2)}$$

[Linear Algebra] The L2 Norm as a Measure of Length

The L2 vector norm is a generalization of the Pythagorean theorem into n dimensions.

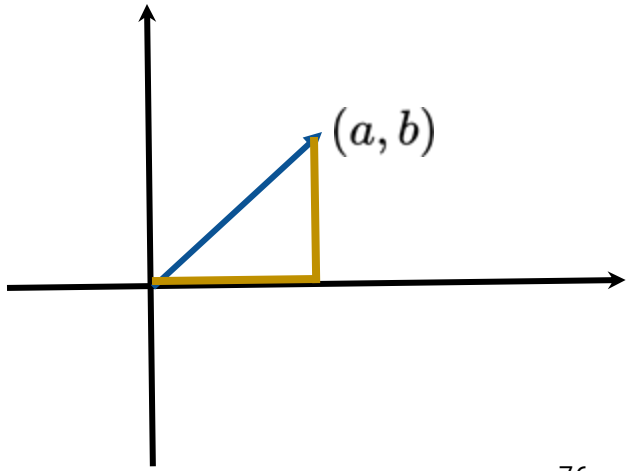
$$||\vec{x}||_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = \sqrt{\sum_{i=1}^{n}(x_i^2)}$$

In $\mathbb{R}^2$



It can therefore be used as a measure of **length** of a vector

- The vector on the right has length $||\vec{v}||_2 = \sqrt{a^2 + b^2}$

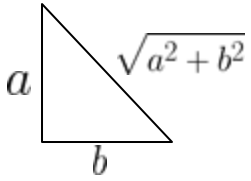


[Linear Algebra] The L2 Norm as a Measure of Distance

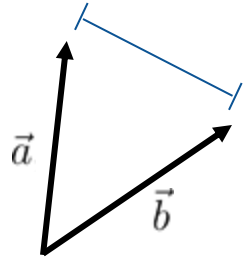
The L2 vector norm is a generalization of the Pythagorean theorem into n dimensions.

$$||\vec{x}||_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} = \sqrt{\sum_{i=1}^{n}(x_i^2)}$$

In $\mathbb{R}^2$



In $\mathbb{R}^n$



It can also be used as a measure of **distance** between two vectors.

- For n-dimensional vectors $\vec{a}, \vec{b}$ $\quad ||\vec{a} - \vec{b}||_2$

Note: The square of the L2 norm of a vector is the sum of the squares of the vector's elements:

$$(||\vec{x}||_2)^2 = \sum_{i=1}^{n} x_i^2$$

Looks like Mean Squared Error!!

Mean Squared Error with L2 Norms

We can rewrite mean squared error as a squared L2 norm:

$$R(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$= \frac{1}{n}(||\mathbb{Y} - \hat{\mathbb{Y}}||_2)^2$$

$$||\vec{x}||_2 = \sqrt{\sum_{i=1}^{n}(x_i^2)}$$

With our linear model $\hat{\mathbb{Y}} = \mathbb{X}\theta$:

$$R(\theta) = \frac{1}{n}(||\mathbb{Y} - \hat{\mathbb{Y}}||_2)^2$$

Ordinary Least Squares

The **least squares estimate** $\hat{\theta}$ is the parameter that **minimizes** the objective function $R(\theta)$:

$$R(\theta) = \frac{1}{n}(||\mathbb{Y} - \hat{\mathbb{Y}}||_2)^2$$

How should we interpret the OLS problem?

A. Minimize the mean squared error for the linear model $\hat{\mathbb{Y}} = \mathbb{X}\theta$

B. Minimize the **distance** between true and predicted values $\mathbb{Y}$ and $\hat{\mathbb{Y}}$

C. Minimize the **length** of the residual vector, $e = \mathbb{Y} - \hat{\mathbb{Y}} = \begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \vdots \\ y_n - \hat{y}_n \end{bmatrix}$

D. All of the above

E. Something else

Ordinary Least Squares

The **least squares estimate** $\hat{\theta}$ is the parameter that **minimizes** the objective function $R(\theta)$:

$$R(\theta) = \frac{1}{n}(||\mathbb{Y} - \hat{\mathbb{Y}}||_2)^2$$

How should we interpret the OLS problem?

A. Minimize the mean squared error for the linear model $\hat{\mathbb{Y}} = \mathbb{X}\theta$

B. Minimize the **distance** between true and predicted values $\mathbb{Y}$ and $\hat{\mathbb{Y}}$

C. Minimize the **length** of the residual vector, $e = \mathbb{Y} - \hat{\mathbb{Y}} = \begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \vdots \\ y_n - \hat{y}_n \end{bmatrix}$ Important for today

D. All of the above

E. Something else

Home Sales Data

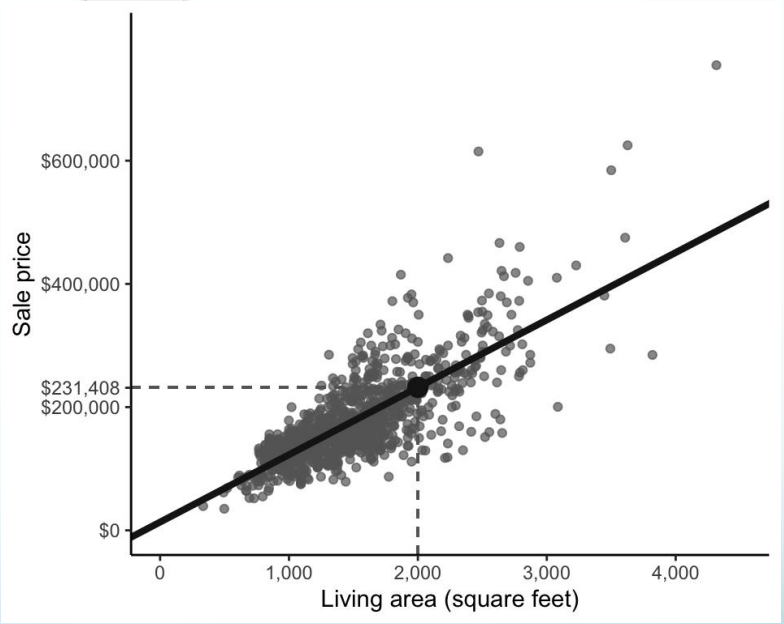
$$\boldsymbol{predicted\ price} = \boldsymbol{b\_0} + \boldsymbol{b\_1} \times \boldsymbol{area}.$$

b0 = 13,408 is **the intercept** of the line

b1 = 109 is the **coefficient** of

the predictor variable

How do we interpret b1?



[Yu, Barter 2024]

Home Sales Data

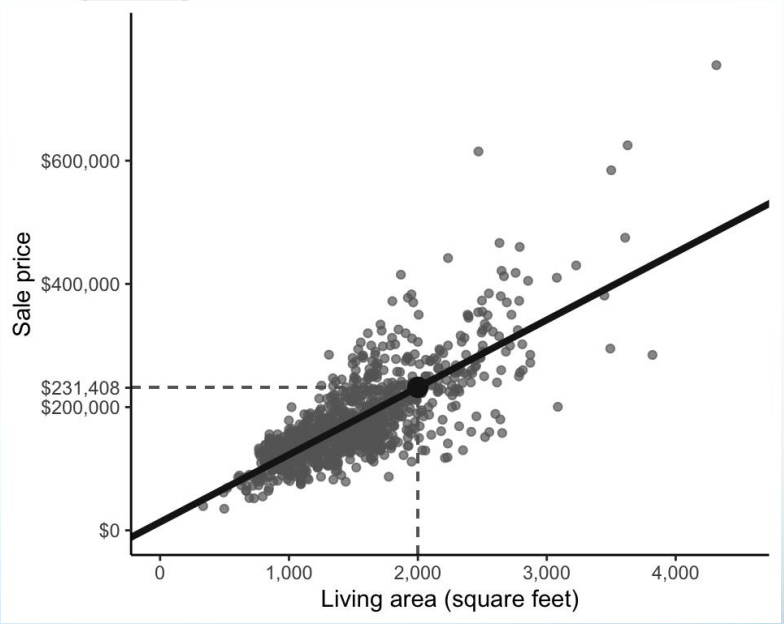
$$\boldsymbol{predicted\ price} = \boldsymbol{b\_0} + \boldsymbol{b\_1} \times \boldsymbol{area}.$$

b0 = 13,408 is **the intercept** of the line

b1 = 109 is the **coefficient** of

the predictor variable

How do we interpret b1?



[Yu, Barter 2024]

Using multiple predictors

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

$$\frac{1}{n}\sum_{i=1}^{n}(\text{observed price}_i - \text{predicted price}_i)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}(\text{observed price}_i - (b_0 + b_1\text{area}_i + b_2\text{quality}_i + b_3\text{year}_i + b_4\text{bedrooms}_i))^2.$$

[Yu, Barter 2024]

Using multiple predictors

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

How do we interpret the coefficients?

[Yu, Barter 2024]

Using the model in matrix form

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

Can you extract this data from the house data?
To fit the model can you call sklearn linear_model.LinearRegression()?

Today's Learning Objectives

Students will be able to:

- ✓ Review: Identify **predictive problems**
- ✓ Review: Plan for **prediction** using **sampling**
- ✓ Understand the relationship between **least squares and correlation**
- ✓ Evaluate the **fit** of a linear model using L1 and L2 loss functions
- ✓ Formulate **multiple linear regression problems** for **Ordinary Least Squares**

Citations:

Yu, B., & Barter, R. L. (2024). Veridical data science: The practice of responsible data analysis and decision making. The MIT Press.

Shah. C. (2020) A hands-on introduction to data science. Cambridge University Press.

Data 100, Fall 2024, UC Berkeley.