Introduction to Regression

k-NN and Linear Regression

Pavlos Protopapas

Lecture Outline

Part A: Statistical Modeling

- k-Nearest Neighbors (kNN)

Part B: Error Evaluation and Model Comparison

- How do we evaluate our model?
- How do we choose from two different models?

Part C: Linear Models

- Linear Regression
- Multi-linear regression

Response and Predictor Variables

Predicting a Variable

Let's consider a scenario in which we aim to predict the value of one variable based on another variable or a set of other variables.

Examples:

Predicting the number of views that a **TikTok** video will receive next week, based on factors such as video length, posting date, and previous view count.

Predicting a Variable

Let's consider a scenario in which we aim to predict the value of one variable based on another variable or a set of other variables.

Examples:

Forecasting which movies, a Netflix user is likely to rate highly, considering their previous movie ratings and demographic data.

Working example

The Advertising dataset contains sales (in 1000 units) data for a specific product across 200 different markets. It also includes advertising budgets in $1000 allocated to three different media channels: *TV, radio, and newspaper*, for each of those markets.

TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9

Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani "

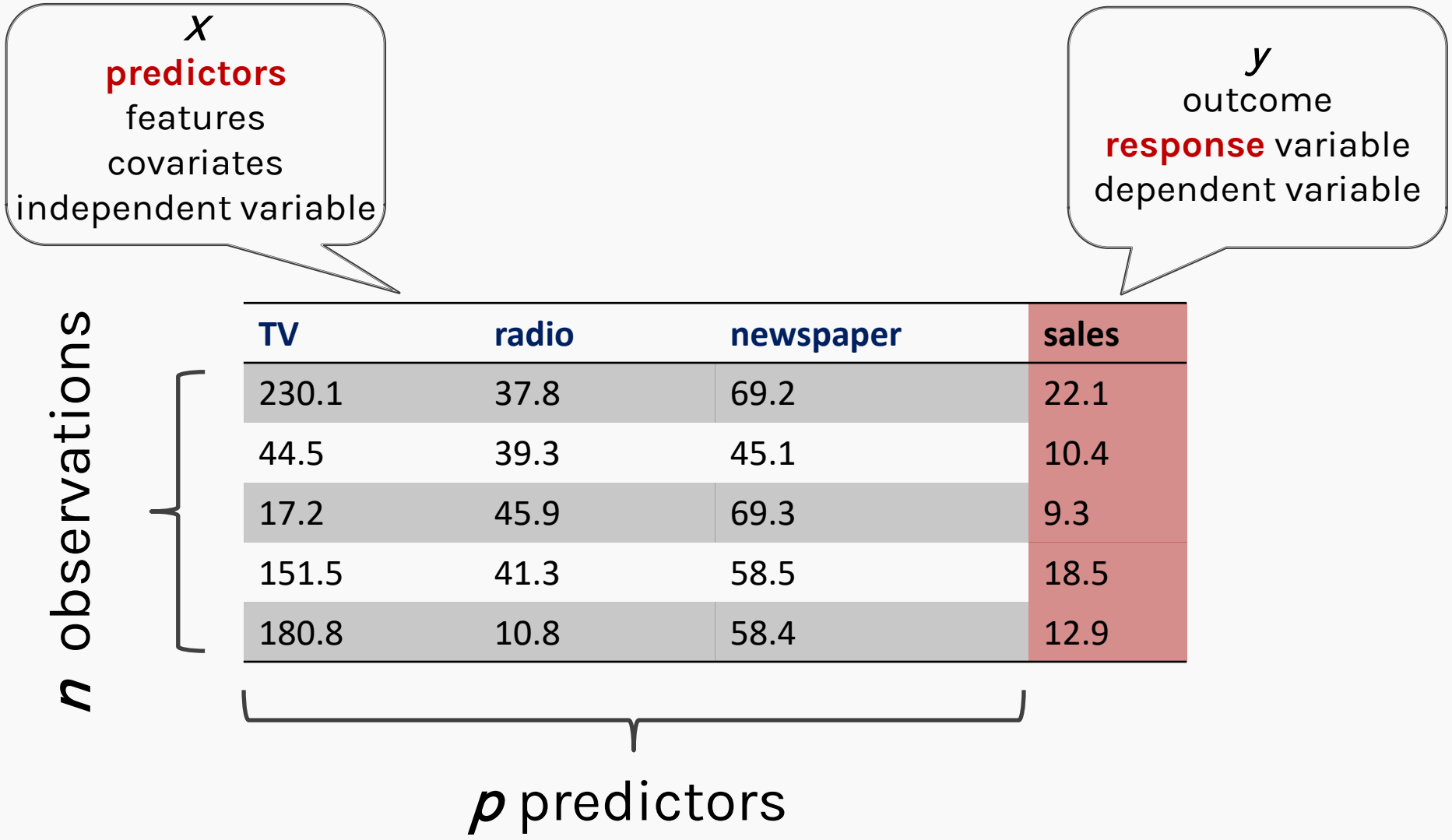
Response vs. Predictor Variables

Many of these problems exhibit an asymmetry: the variable we aim to predict may be harder to measure, more significant, or directly or indirectly influenced by other variables.

Therefore, we can classify variables into two categories:

- Variables whose values we aim to predict.
- Variables used as inputs to inform our prediction.

Response vs. Predictor Variables



TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9

Response vs. Predictor Variables

$X = X_1, \dots, X_p$

$X_j = x_{1j}, \dots, x_{ij}, \dots, x_{nj}$

predictors

features

covariates

independent variable

$\boldsymbol{y} = y_1, \dots, y_i, \dots, y_n$

outcome

response variable

dependent variable

TV	radio	newspaper	sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	9.3
151.5	41.3	58.5	18.5
180.8	10.8	58.4	12.9

n observations

p predictors

Response vs. Predictor Variables

This is called X: a.k.a.
The Design Matrix

n observations

TV	radio	newspaper
230.1	37.8	69.2
44.5	39.3	45.1
17.2	45.9	69.3
151.5	41.3	58.5
180.8	10.8	58.4

$\boldsymbol{y}$:
The response variable

sales
22.1
10.4
9.3
18.5
12.9

Response vs. Predictor Variables

This is called X: a.k.a.
The Design Matrix

y:
The response variable

n observations

TV	radio	newspaper
230.1	37.8	69.2
44.5	39.3	45.1
17.2	45.9	69.3
151.5	41.3	58.5
180.8	10.8	58.4

sales
22.1
10.4
9.3
18.5
12.9

Capital letters mean matrices,

Response vs. Predictor Variables

This is called X: a.k.a.
The Design Matrix

n observations

TV	radio	newspaper
230.1	37.8	69.2
44.5	39.3	45.1
17.2	45.9	69.3
151.5	41.3	58.5
180.8	10.8	58.4

y
The response variable

sales
22.1
10.4
9.3
18.5
12.9

Capital letters mean matrices, lower case letters mean vectors

Sklearn expects certain dimensions

```
>>> X.shape
(n, p)
```

```
>>> y.shape
(n,) OR (n, 1)
```

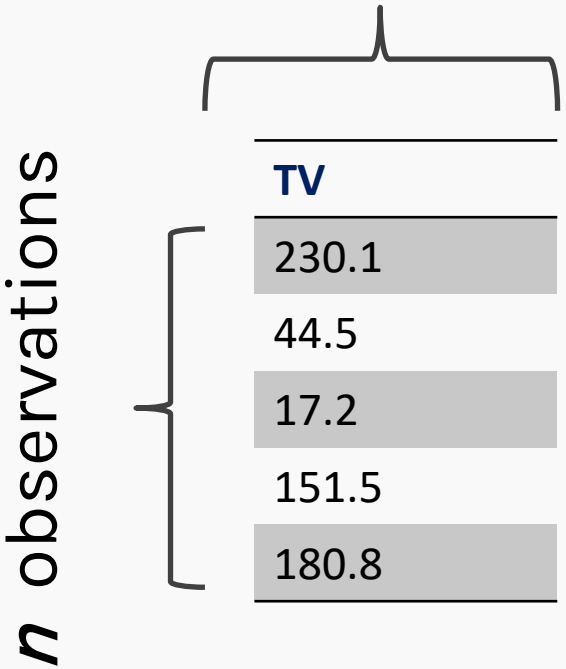
n observations

TV	radio	newspaper
230.1	37.8	69.2
44.5	39.3	45.1
17.2	45.9	69.3
151.5	41.3	58.5
180.8	10.8	58.4

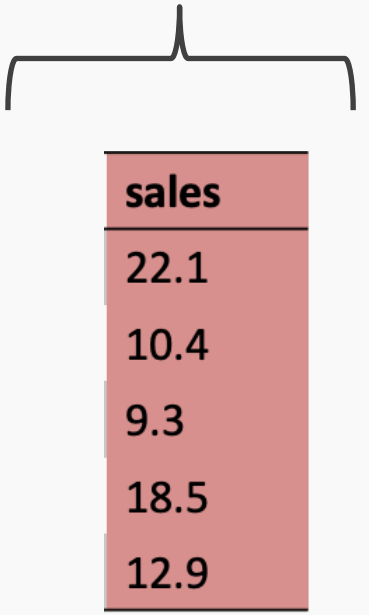
sales
22.1
10.4
9.3
18.5
12.9

Sklearn expects certain dimensions

```
>>> X.shape
(n,) OR (n,1)
```




```
>>> y.shape
(n,) OR (n,1)
```



Pandas Review

df[['x']] vs df['x']

Which of the statements below is correct?

Options:

A. df[['x']] `returns` a pd.Series object whereas df['x'] returns a pd.DataFrame.

B. df[['x']] `is invalid operation.`

C. df[['x']] returns a pd.DataFrame whereas df['x'] `returns` a pd.Series object.

D. df['x'] `is invalid operation.`

Statistical Model

True vs. Statistical Model

- Imagine an ice cream cone so perfect that it captures every flavor, topping, and swirl of deliciousness. That is what a ***true model*** is.

True vs. Statistical Model

- Imagine an ice cream cone so perfect that it captures every flavor, topping, and swirl of deliciousness. That is what a ***true model*** is.

- But reality is like an ice cream shop with infinite flavors and toppings. Trying to fit all of them into one cone is impossible.

True vs. Statistical Model

- Imagine an ice cream cone so perfect that it captures every flavor, topping, and swirl of deliciousness. That is what a ***true model*** is.

- But reality is like an ice cream shop with infinite flavors and toppings. Trying to fit all of them into one cone is impossible.

- This is why we use ***statistical models***: instead of trying to scoop the impossible sundae, we craft a tasty treat from the flavors we have.

True vs. Statistical Model

We assume that the response variable, y, is related to the predictor variables, X, through an unknown function which can be generally expressed as:

$$y = f(X) + \varepsilon$$

Here, f represents the unknown function expressing an underlying rule for relating y to X. ε represents the random amount (unrelated to X) that y differs from the rule $f(X)$.

A ***statistical model*** is any algorithm used to estimate f. We denote the estimated function as $\widehat{f}$.

Prediction vs. Estimation

Inference Problems:

- The primary focus is on obtaining $\hat{f}$, which is an estimate of the true function f
- Objective: Understand the form and characteristics of $\hat{f}$.

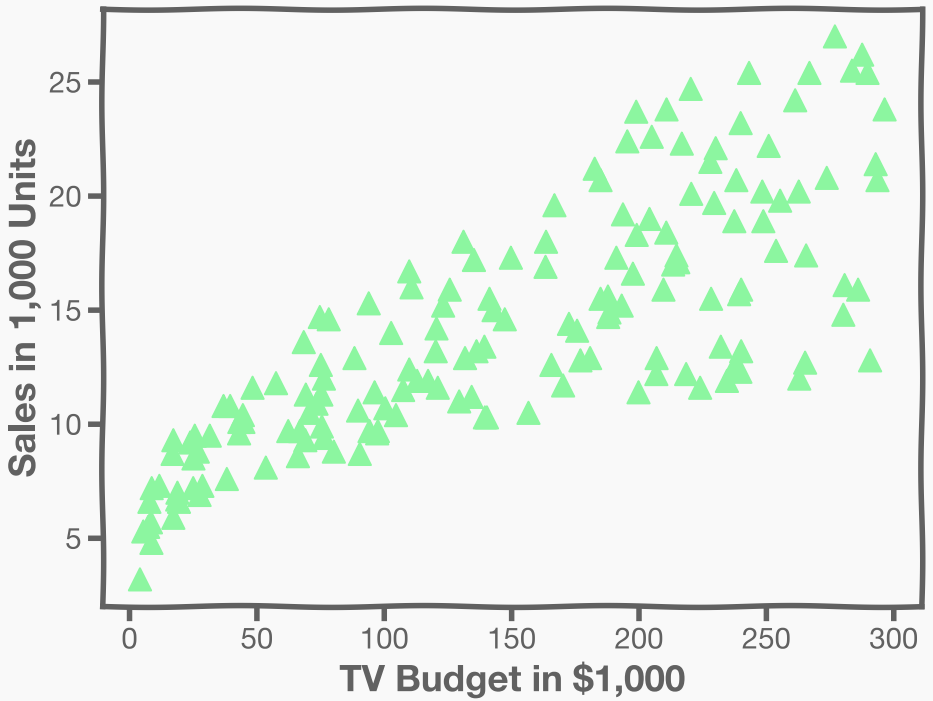
Prediction Problems

- The specific form of $\hat{f}$ is less important than the accuracy of the predictions.
- Objective: Minimize the difference between predicted values $\hat{y}$ and observed values y.

Example: Predicting sales

Motivation: Predict Sales

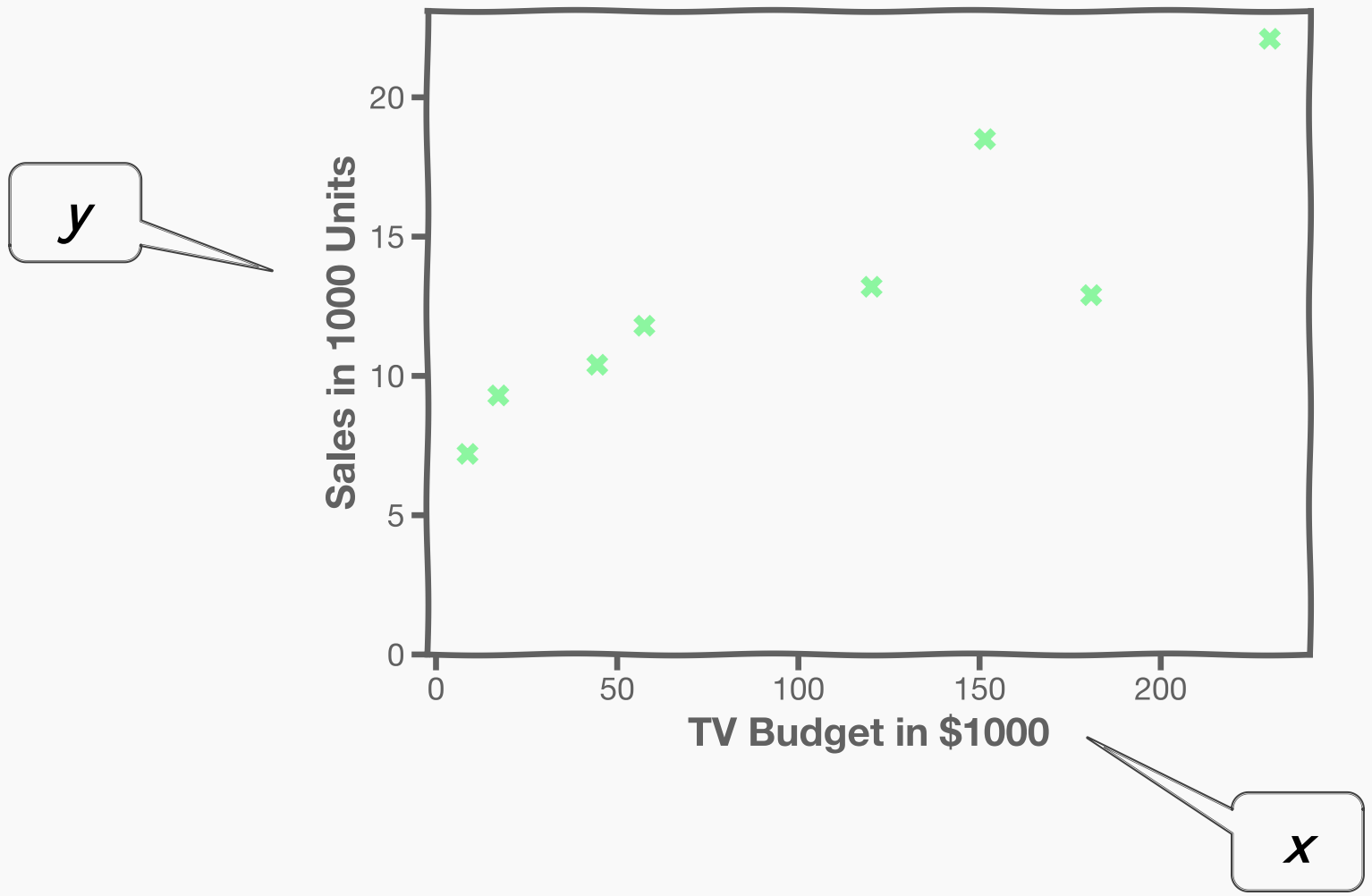
Build a model to **predict** sales based on TV budget



The response, y, is the sales.

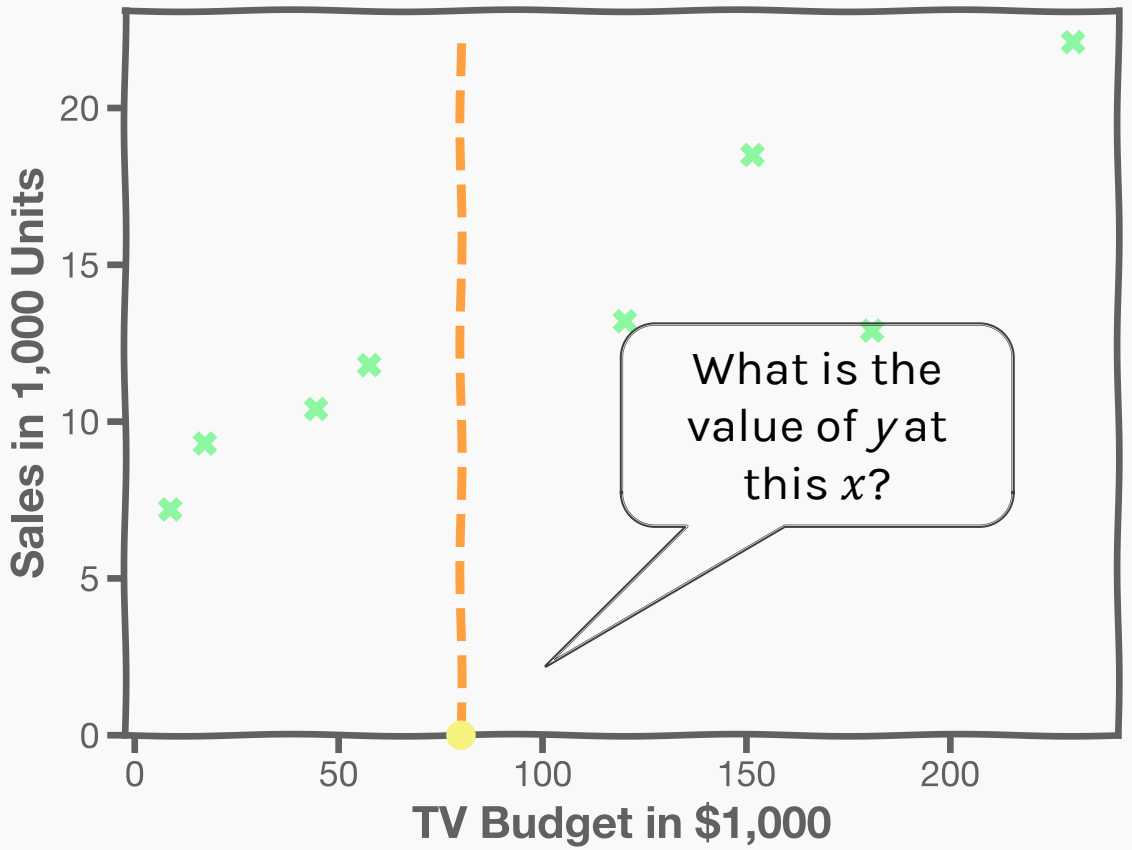
The predictor, x, is TV budget.

Statistical Model



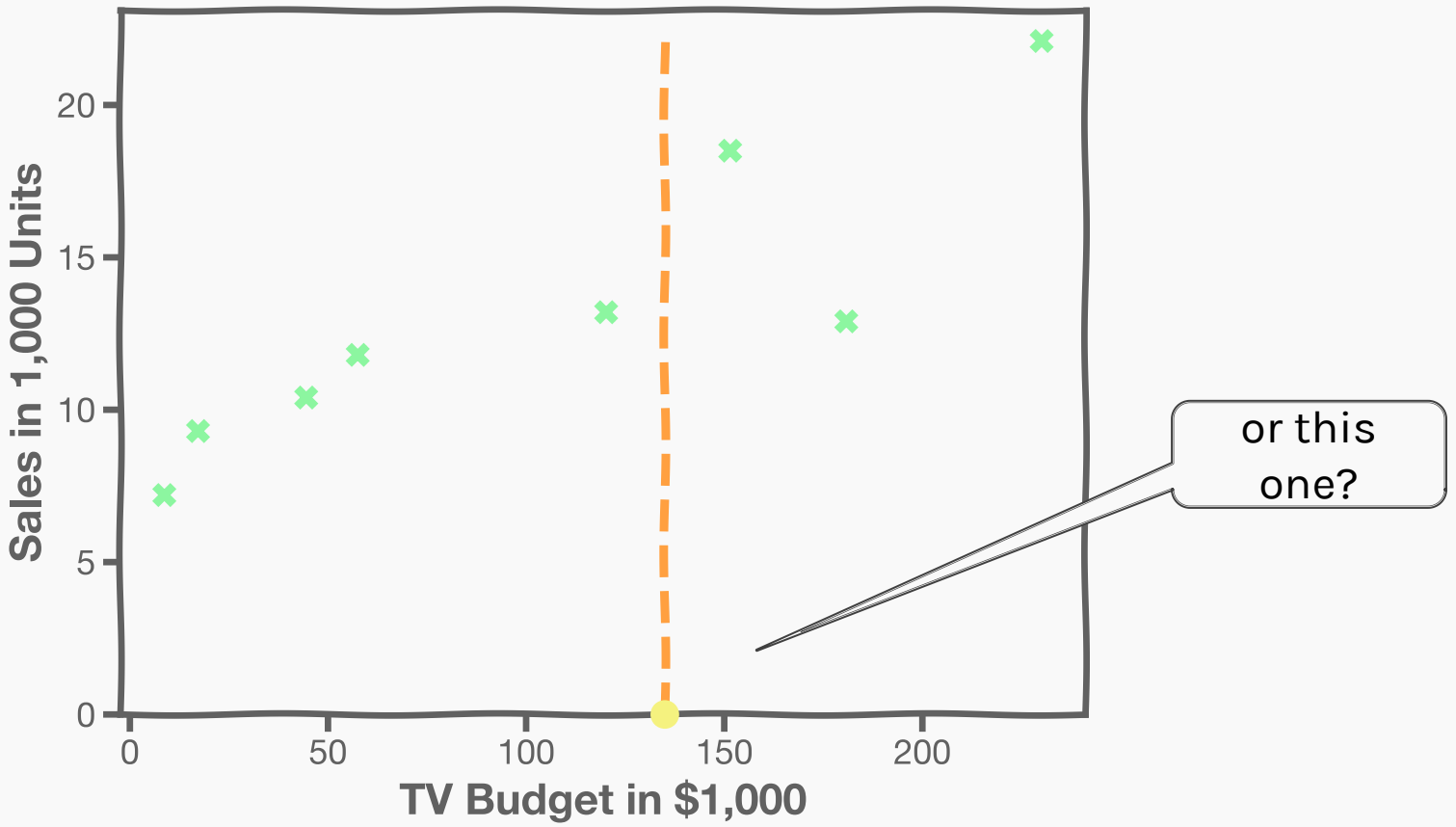
Statistical Model

How do we predict y for some x?



Statistical Model

How do we predict y for some x?



Choices for model

Which of the following methods could be used to predict the value of y given x?

Options:

A. Utilize a Convolutional Neural Network (CNN).

B. Use a Linear Regression Model with a slope of 3 and an intercept of 2.

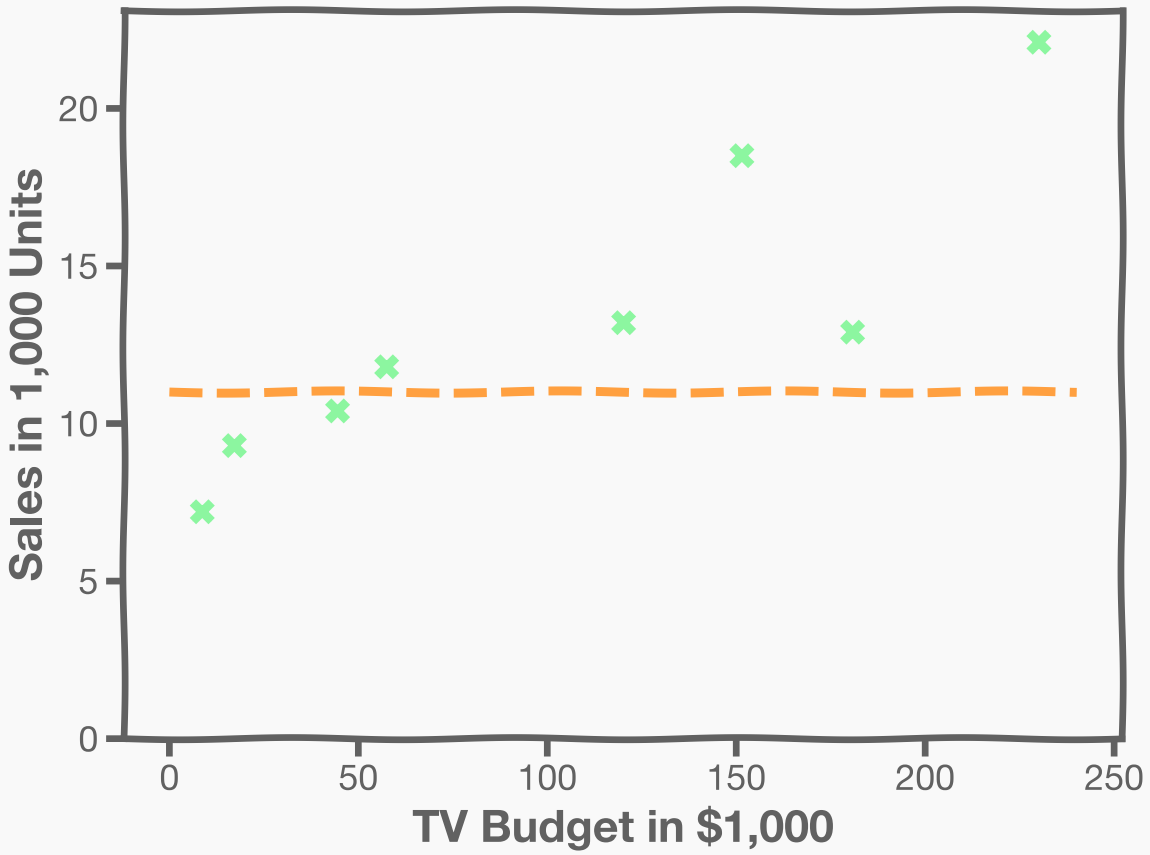
C. Identify examples that closely resemble the input data point.

D. Consult a TF during office hours for the answer.

E. Calculate the average value of y from the available data points.

Statistical Model

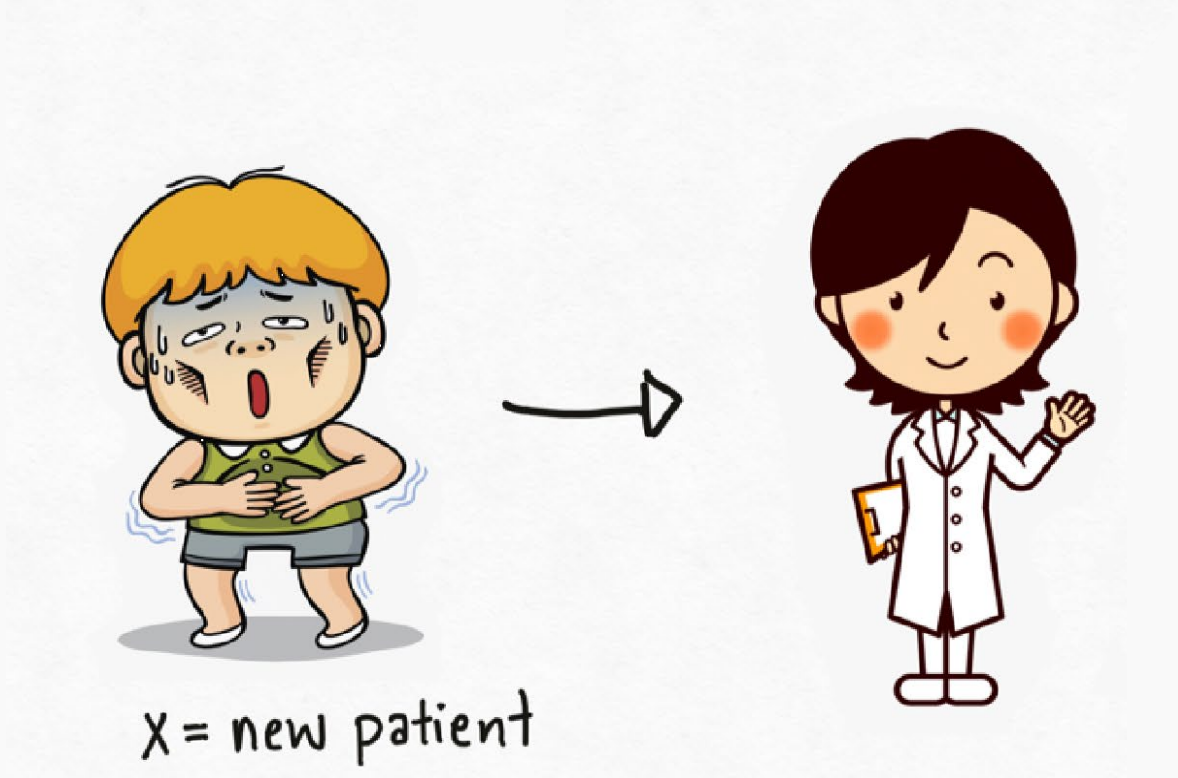
A simple idea is to take the mean of all y's: $\frac{1}{n}\sum_{i=1}^{n} y_i$



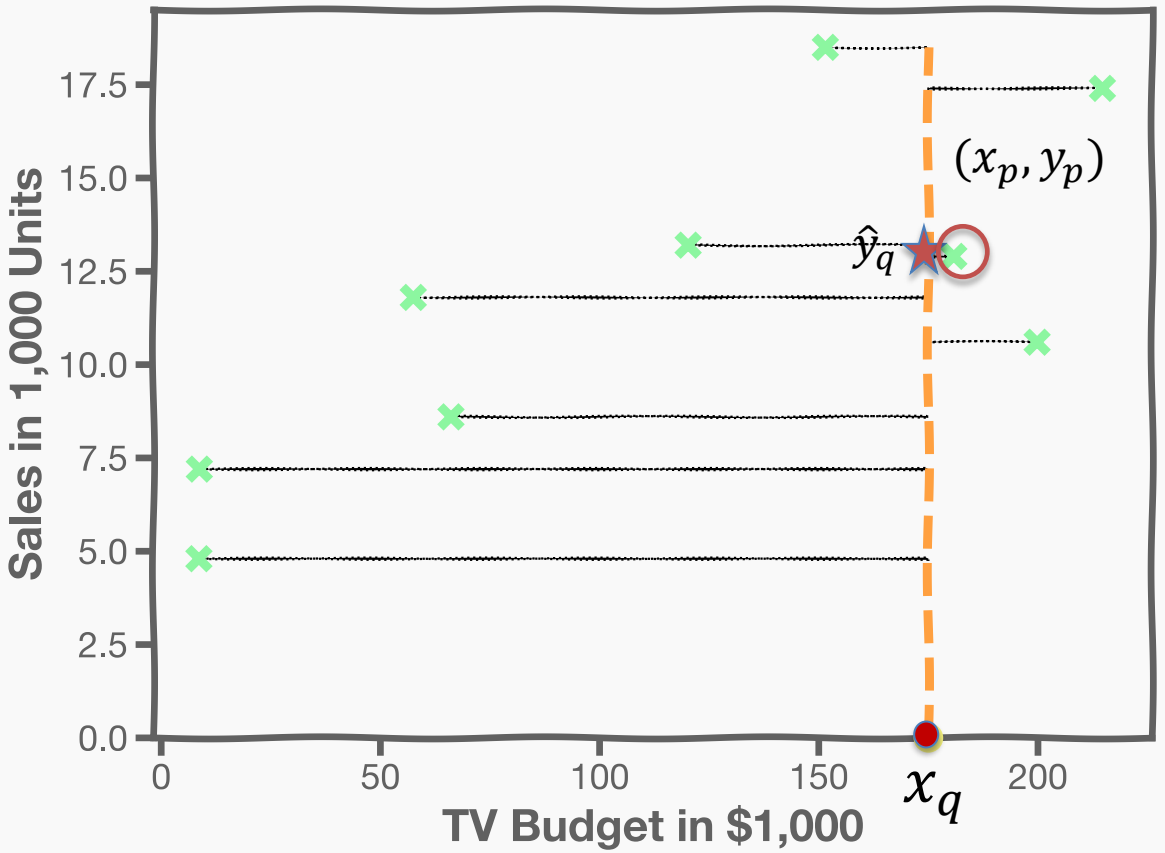
$$\frac{1}{n}\sum_{i=1}^{n} y_i$$

K-Nearest Neighbors

k-Nearest Neighbors – kNN



k-Nearest Neighbors – kNN



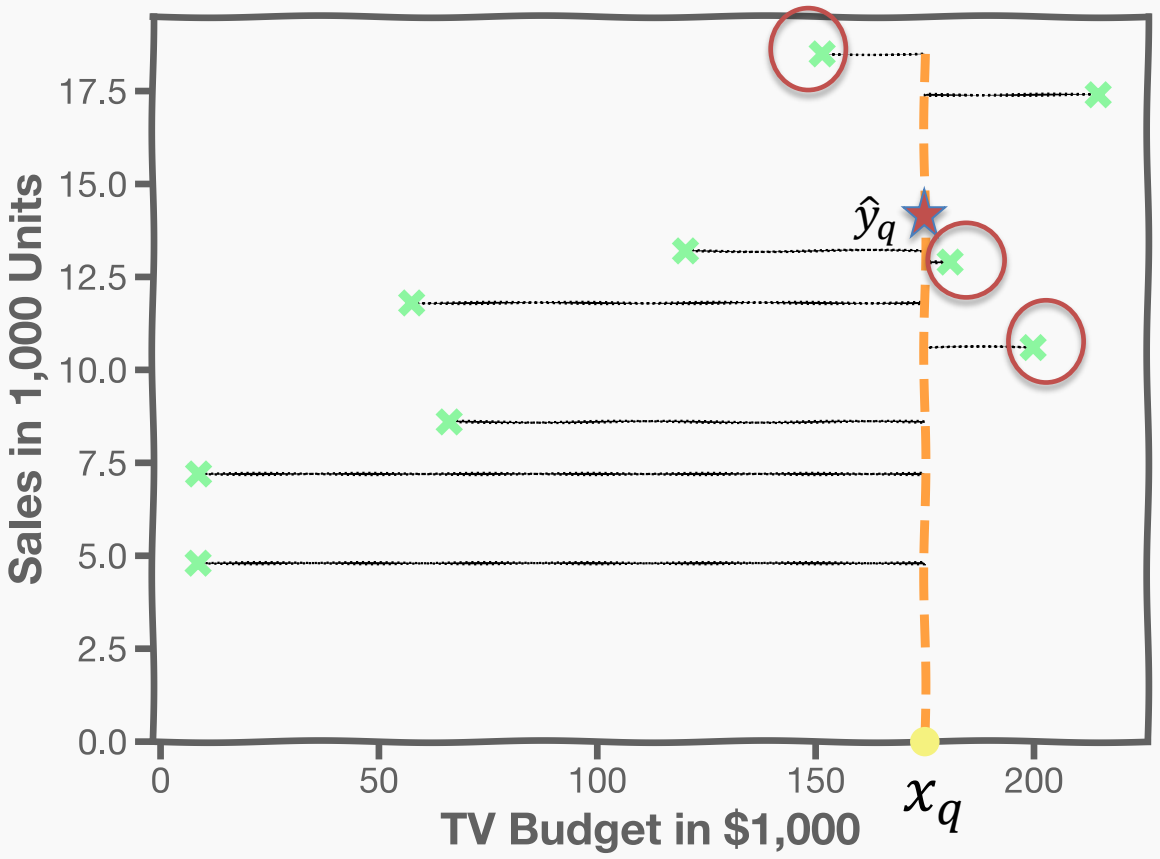
What is $\hat{y}_q$ at some x_q ?

Find distances to all other points $D(x_q, x_i)$

Find the nearest neighbor, (x_p, y_p)

Predict $\hat{y}_q = y_p$

k-Nearest Neighbors – kNN



What is $\hat{y}_q$ at some x_q ?

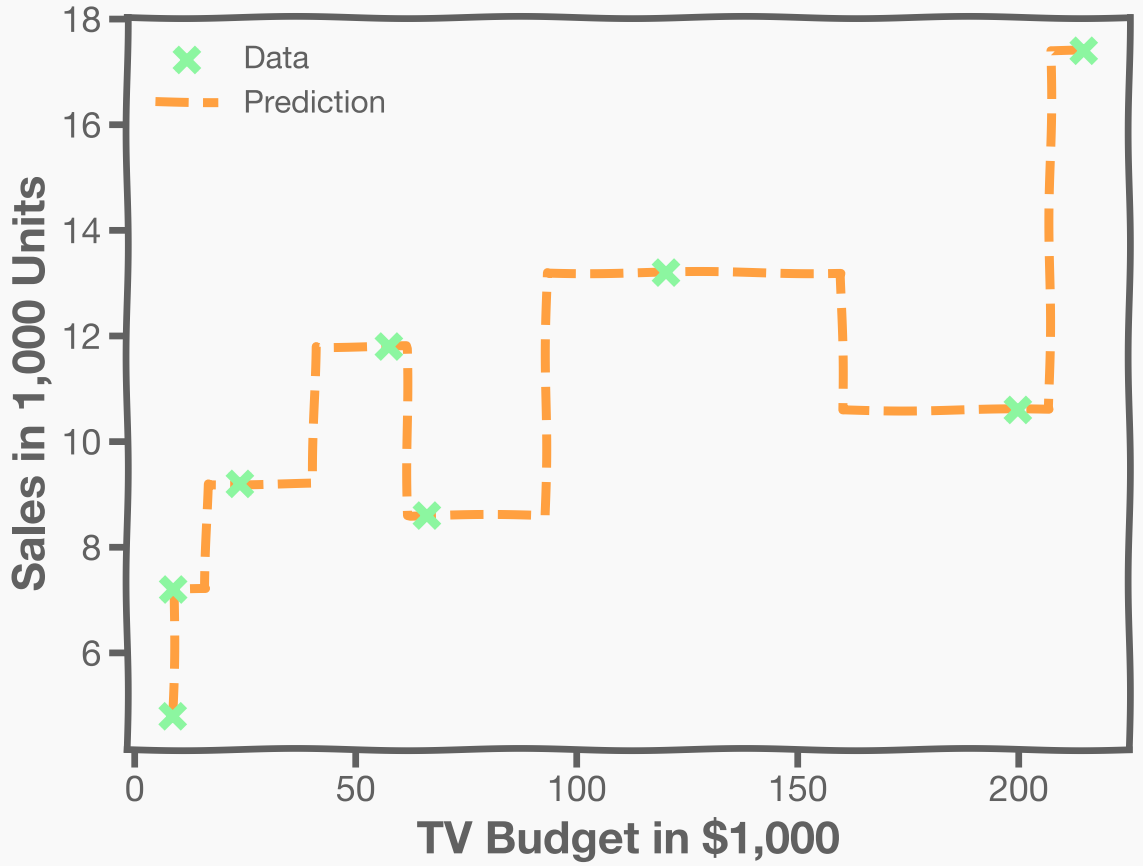
Find distances to all other points $D(x_q, x_i)$

Find the k-nearest neighbors, $x_{q_1}, \ldots, x_{q_k}$

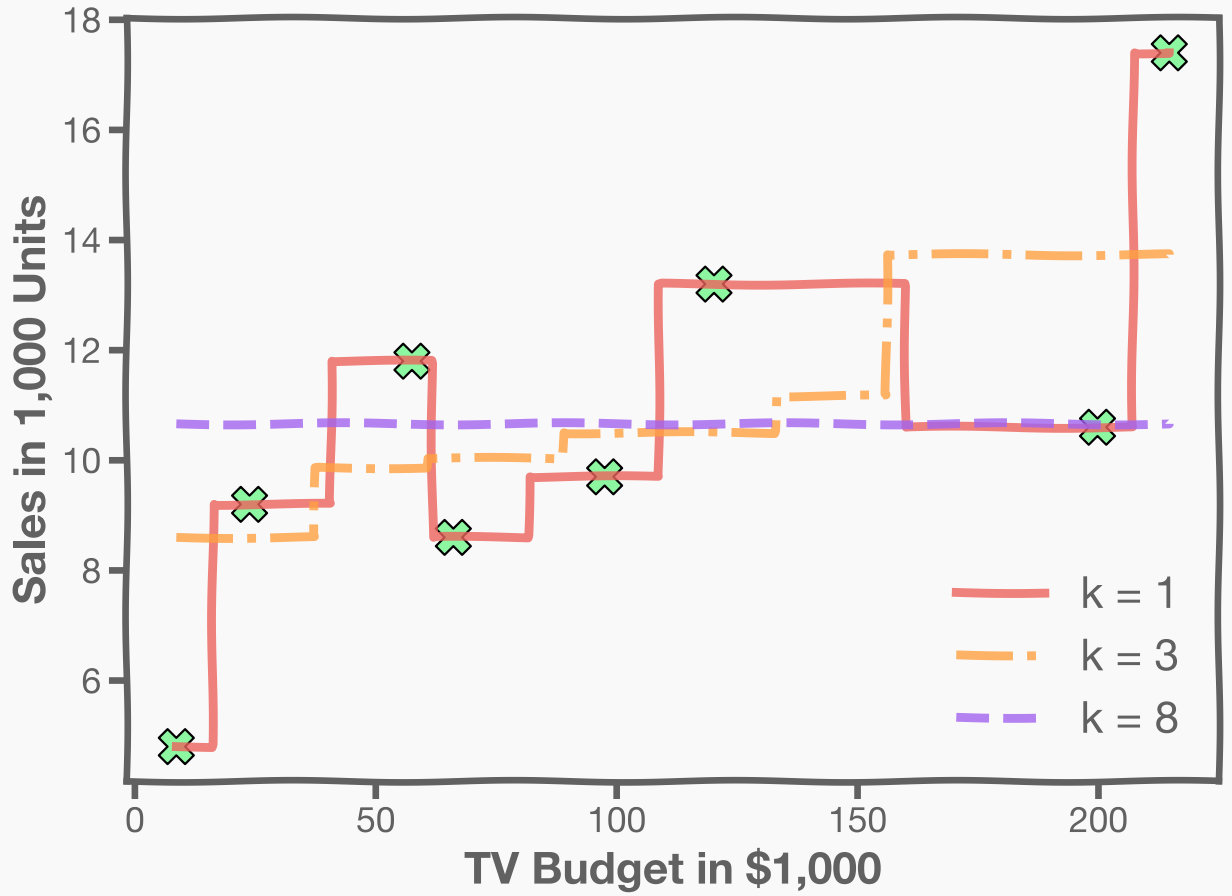
Predict $\hat{y}_q = \frac{1}{k}\sum_i^k y_{q_i}$

k-Nearest Neighbors – kNN

Do the same for "all" $x's$

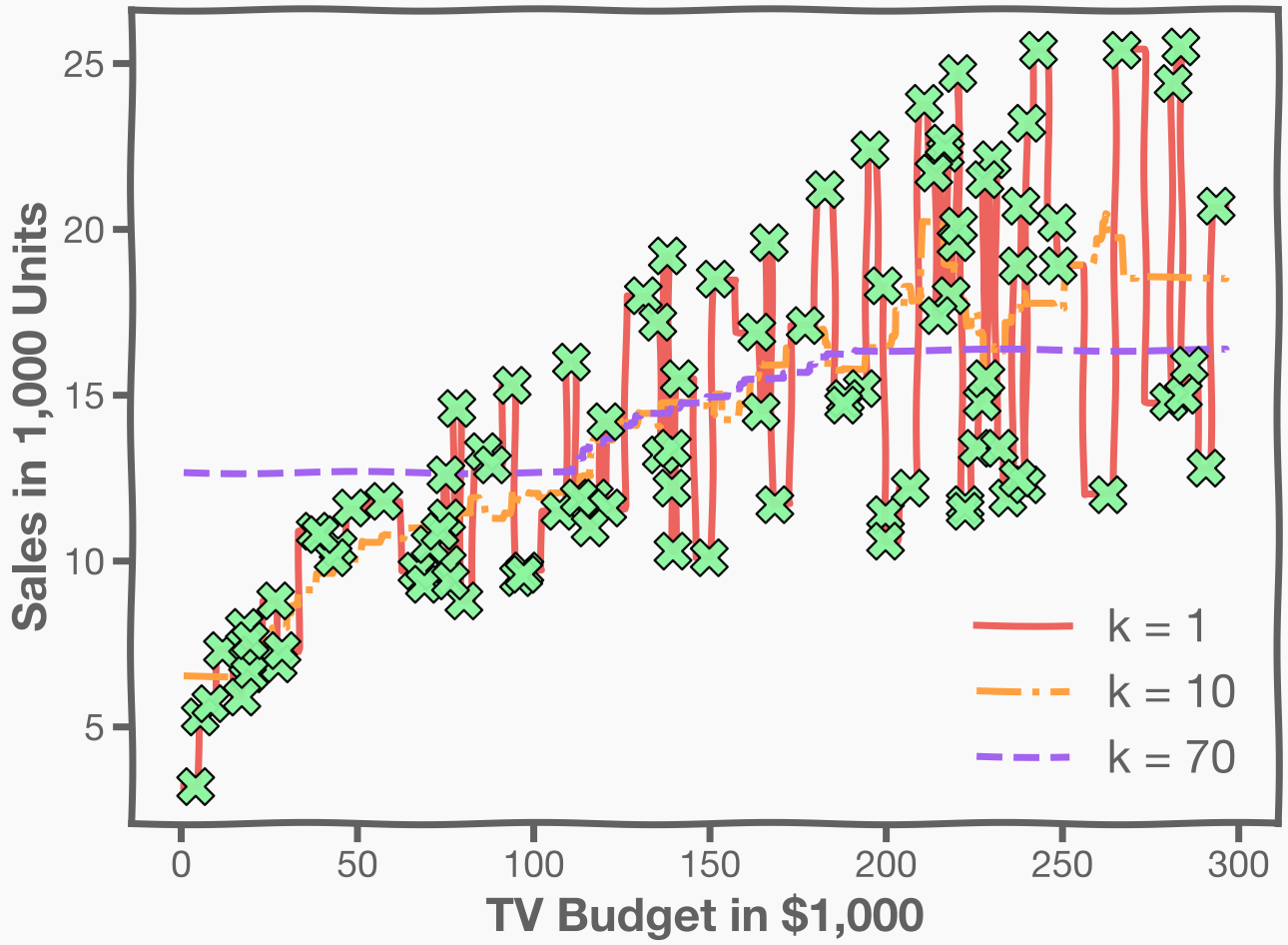


k-Nearest Neighbors – kNN



k-Nearest Neighbors – kNN

We can try different k-models on more data



Lecture Outline

Part A: Statistical Modeling

k-Nearest Neighbors (kNN)

Part B: Error Evaluation and Model Comparison

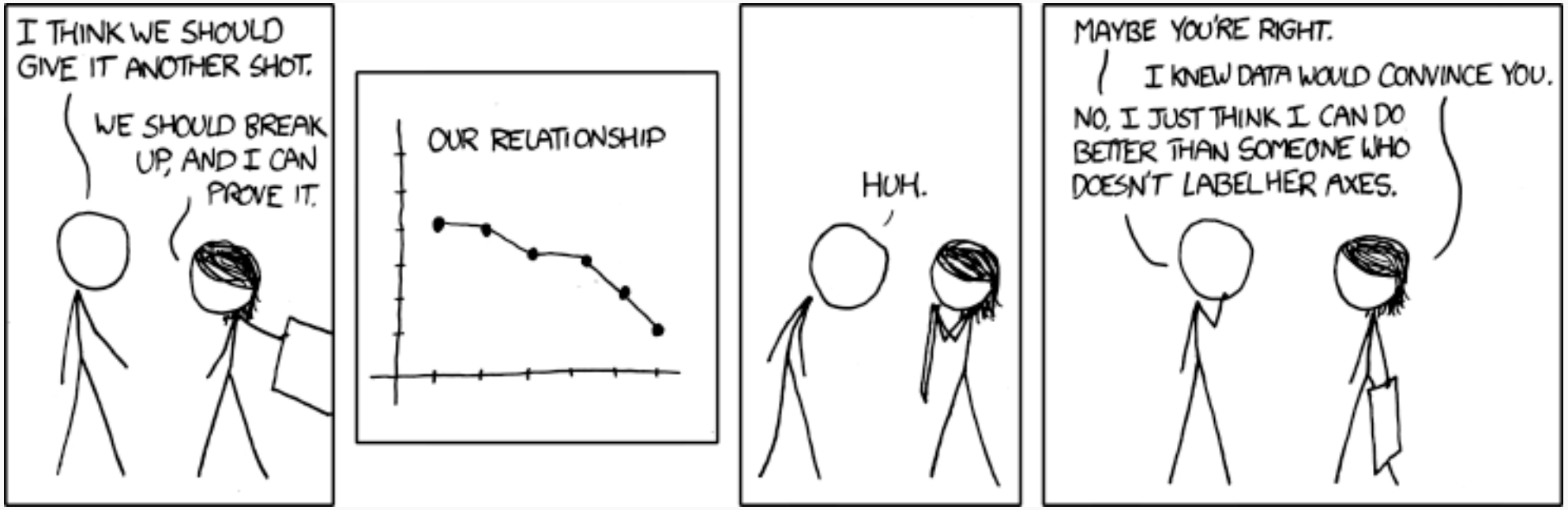
How do we evaluate our model?

How do we choose from two different models?

Part C: Linear Models

Linear Regression

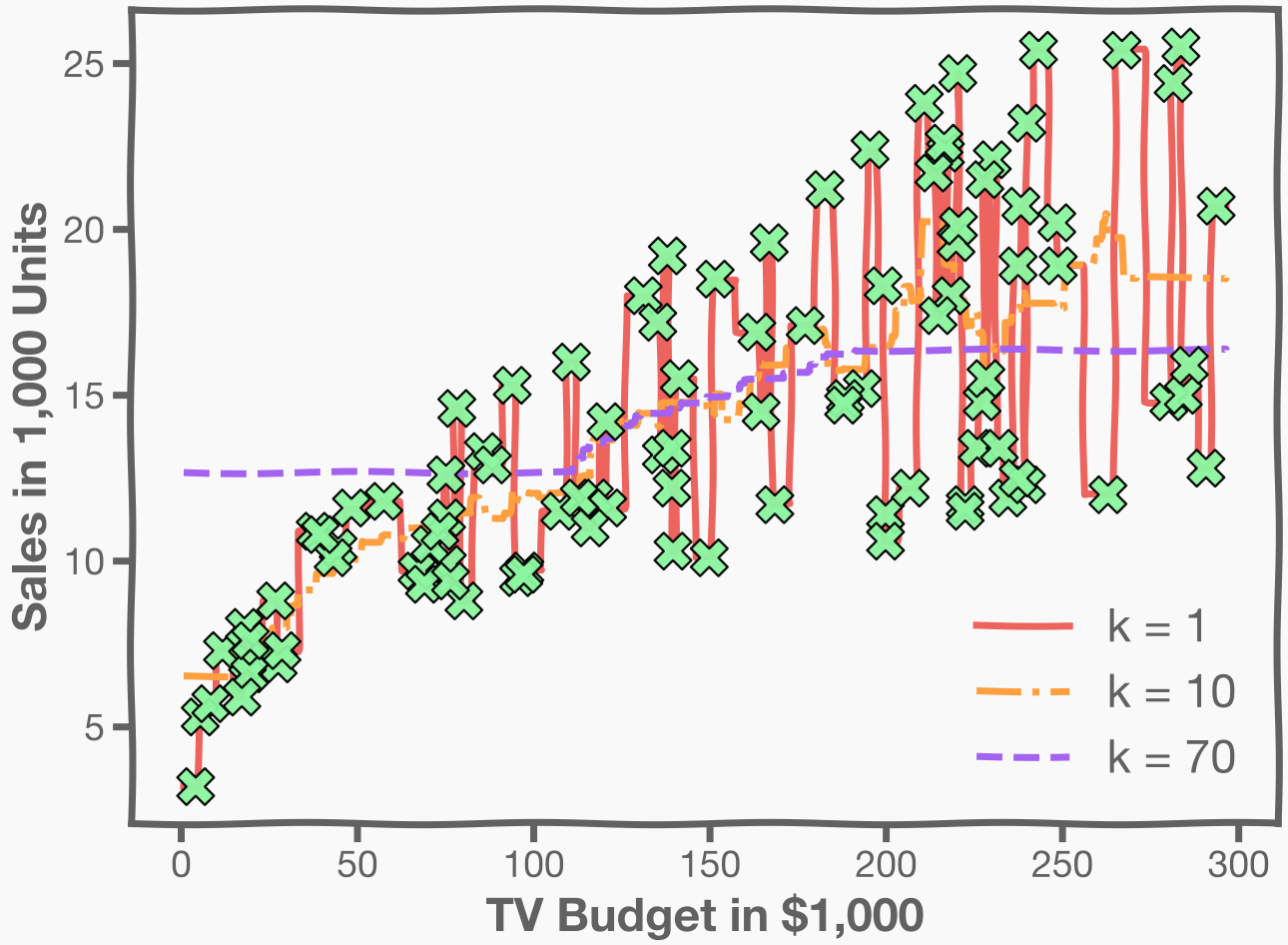
Multi-linear regression



https://xkcd.com/833/

k-Nearest Neighbors – kNN

We have tested various models using different k-values on the data.



Choices for model

Which model do you think is the best?

Options:

A. $k = 1$

B. $k = 10$

C. $k = 70$

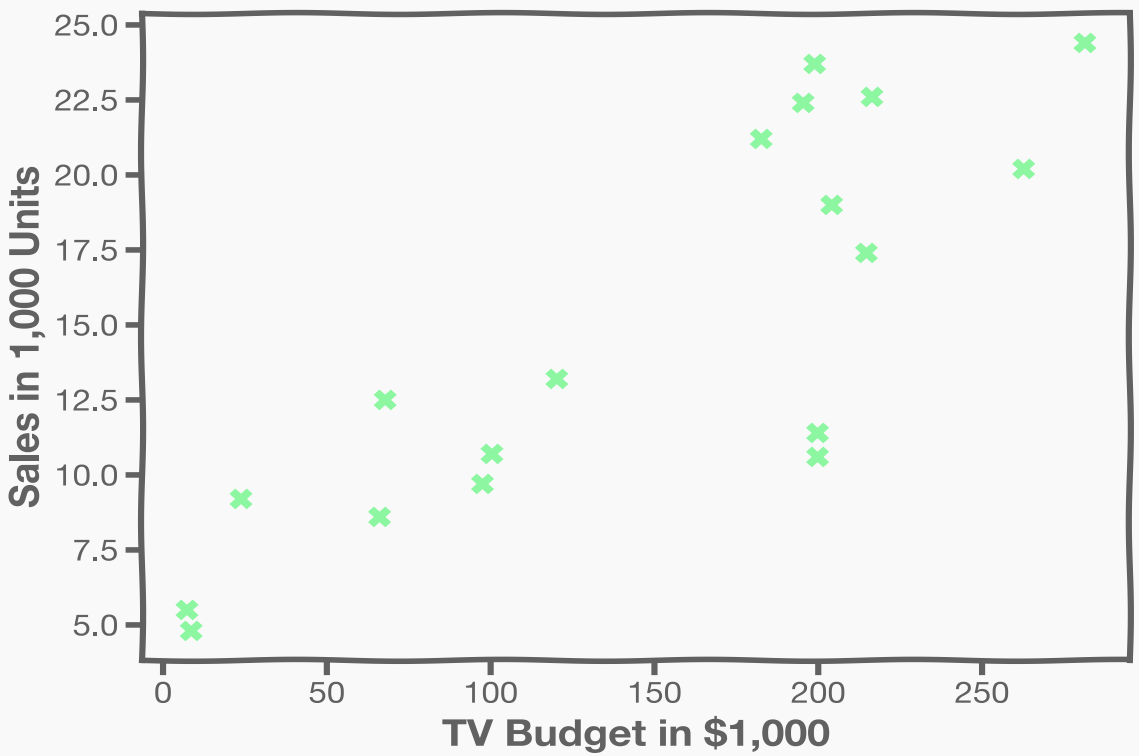
D. $k = 15$

E. 🤷‍♂️

Error Evaluation

Error Evaluation

We need to define what we mean by *best*. To do so, we start with our data.



Error Evaluation

We first **withhold** a portion of the data from the model; this process is called **train-test** split.

Train Set

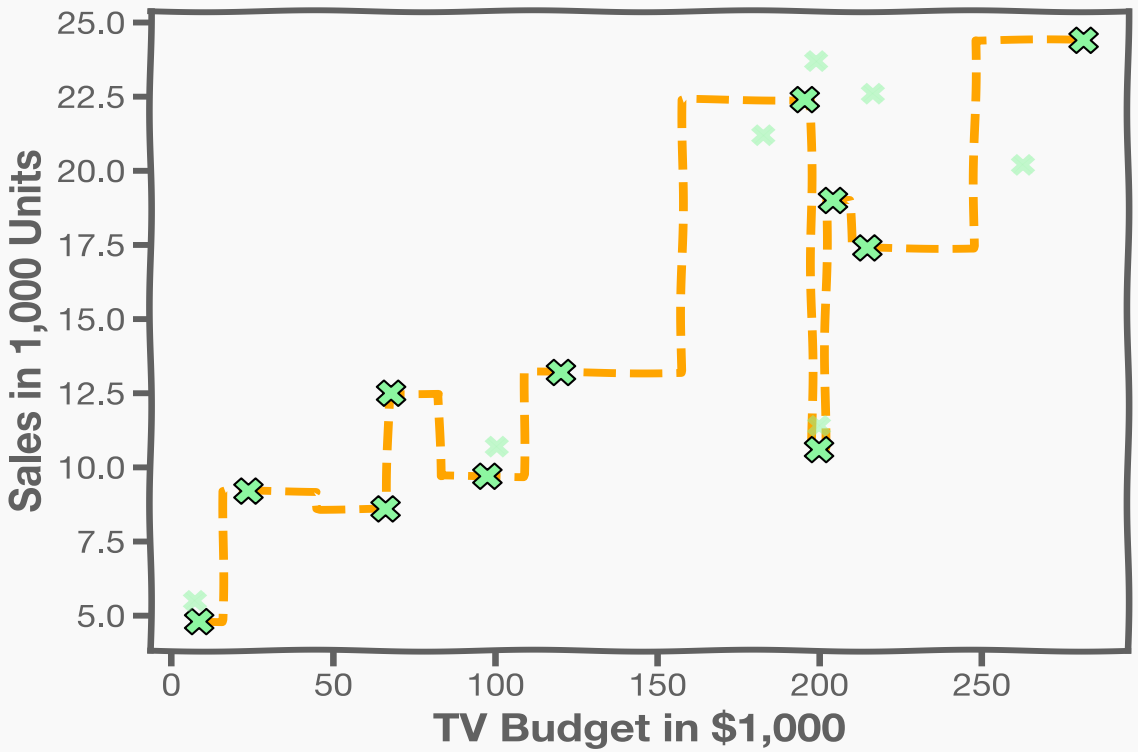
The data that we use to train our model to estimate, $\hat{y}$.

Test Set

The data that we use to evaluate our model's performance.

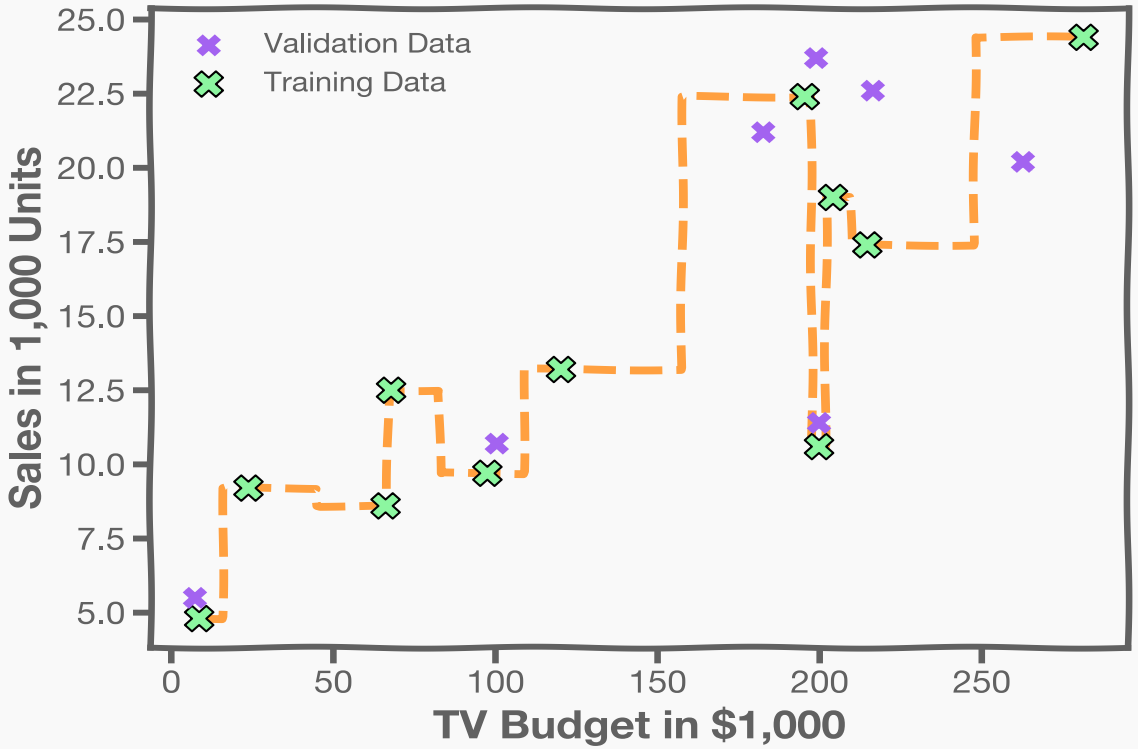
Error Evaluation

Estimate $\hat{y}'s$ values for all the data points in the training set when *k=1*.



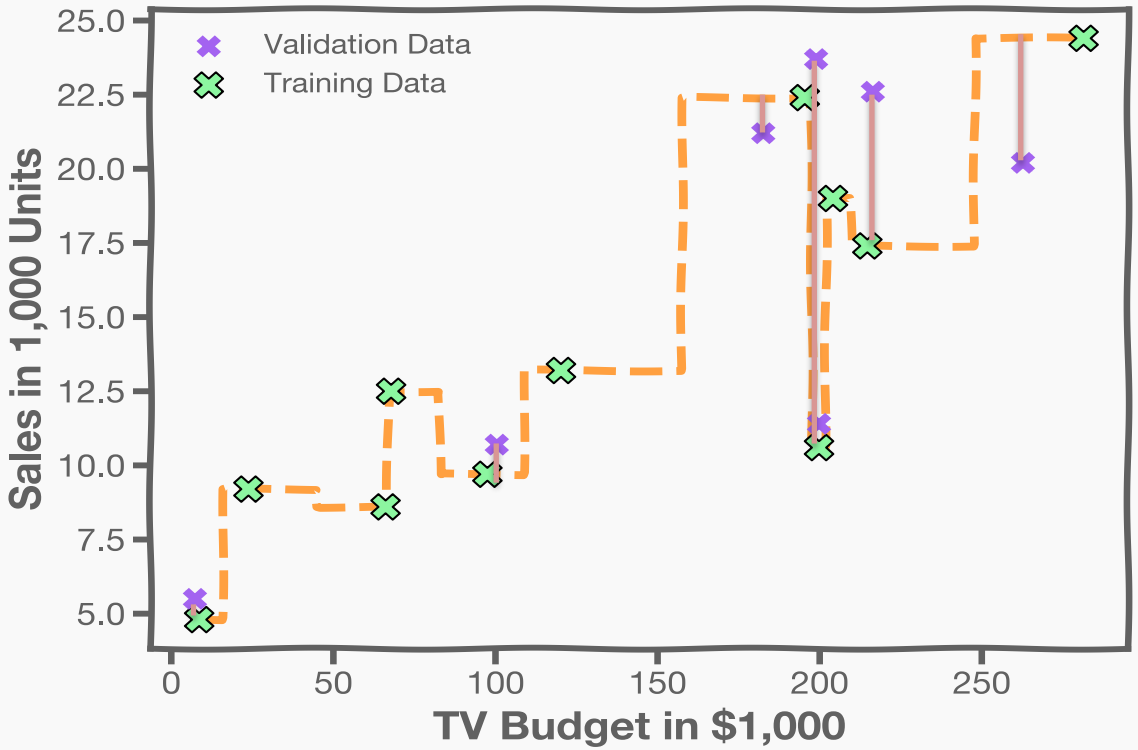
Error Evaluation

Now, we examine the data that was not used for estimating $\hat{y}$, the **test data** represented by purple crosses.



Error Evaluation

And we calculate the **residuals** $(y_i - \hat{y}_i)$.



For each observation (x_n, y_n), the absolute residuals, $r_i = |y_i - \hat{y}_i|$ quantify the error at each observation point.

Error Evaluation

To quantify the performance of a model, we aggregate the errors. This aggregated value is commonly referred to as the *loss*, *error*, or *cost function*.

A widely used loss function for quantitative outcomes is the **Mean Squared Error (MSE):**

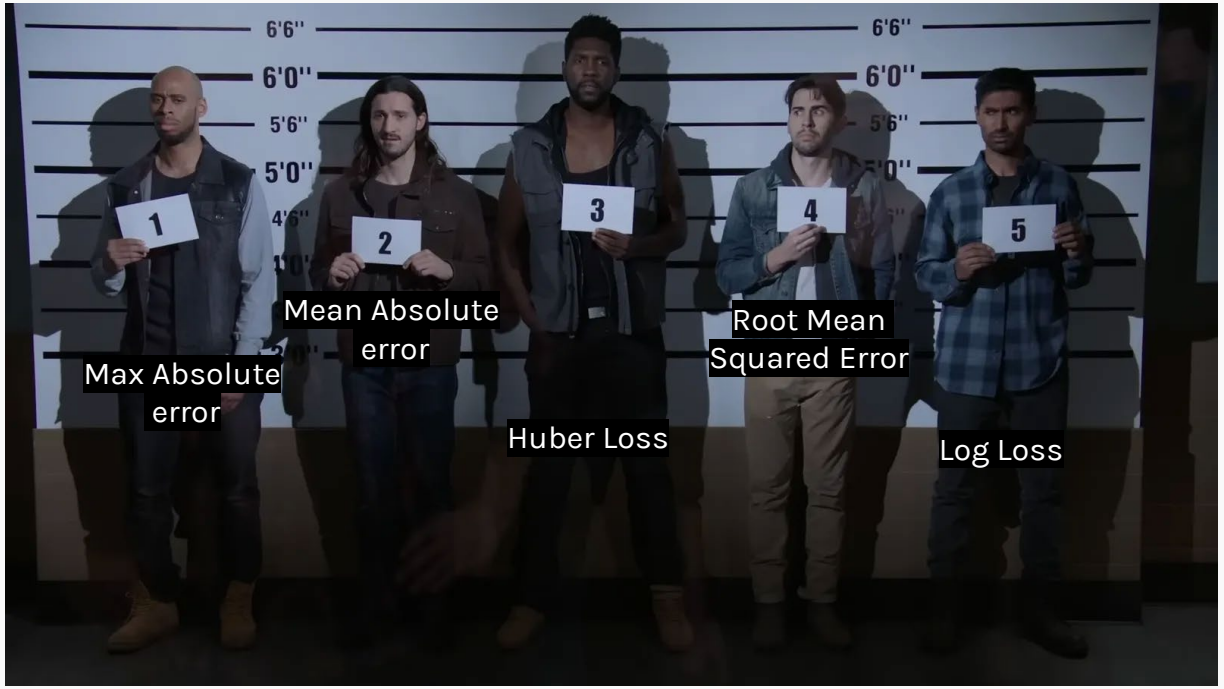
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2$$

Note: Loss and cost function refer to the same thing. Cost usually refers to the total loss where loss refers to a single training point.

Error Evaluation

Caution: MSE is not the only valid, or necessarily the best, loss function for all scenarios.

There are many different types of loss functions:



Note: The square **R**oot of the **M**ean of the **S**quared **E**rrors (RMSE) is also commonly used.

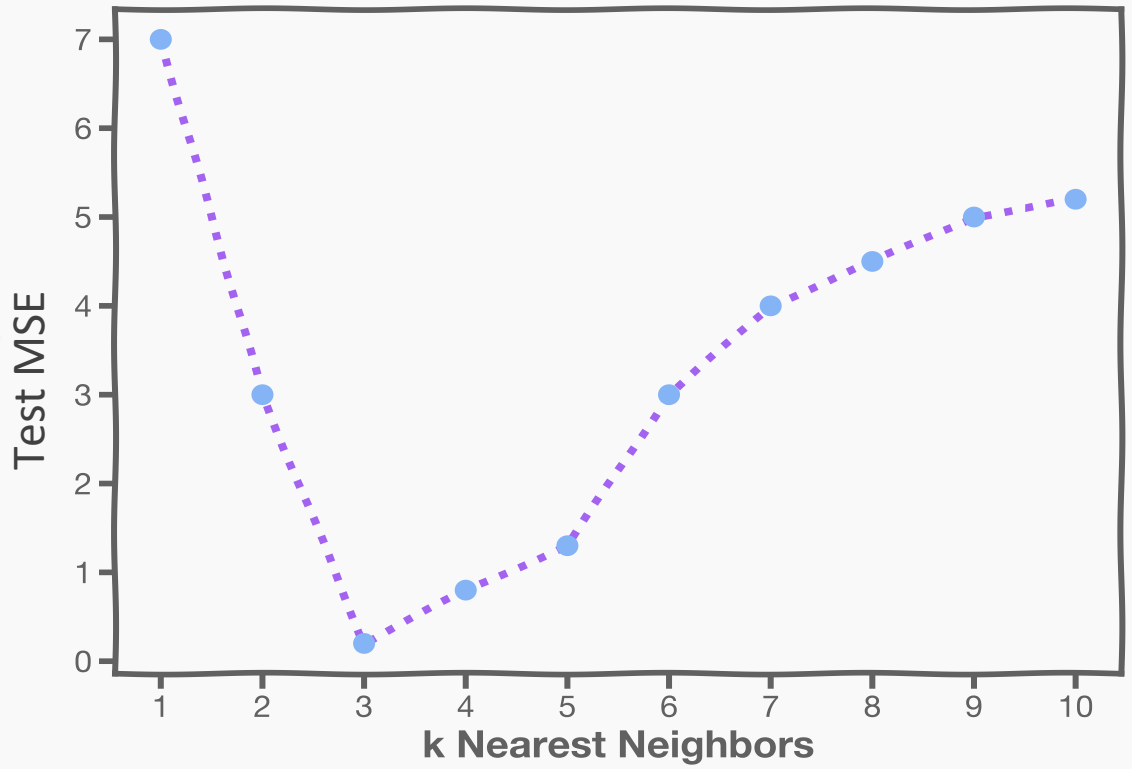
$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Model Comparison

Model Comparison

We repeat this process for all values of *k* and compare the MSEs on the test set.

We will introduce validation set soon



Which model is the best?

Question

Which model do you think is the best now

Options:

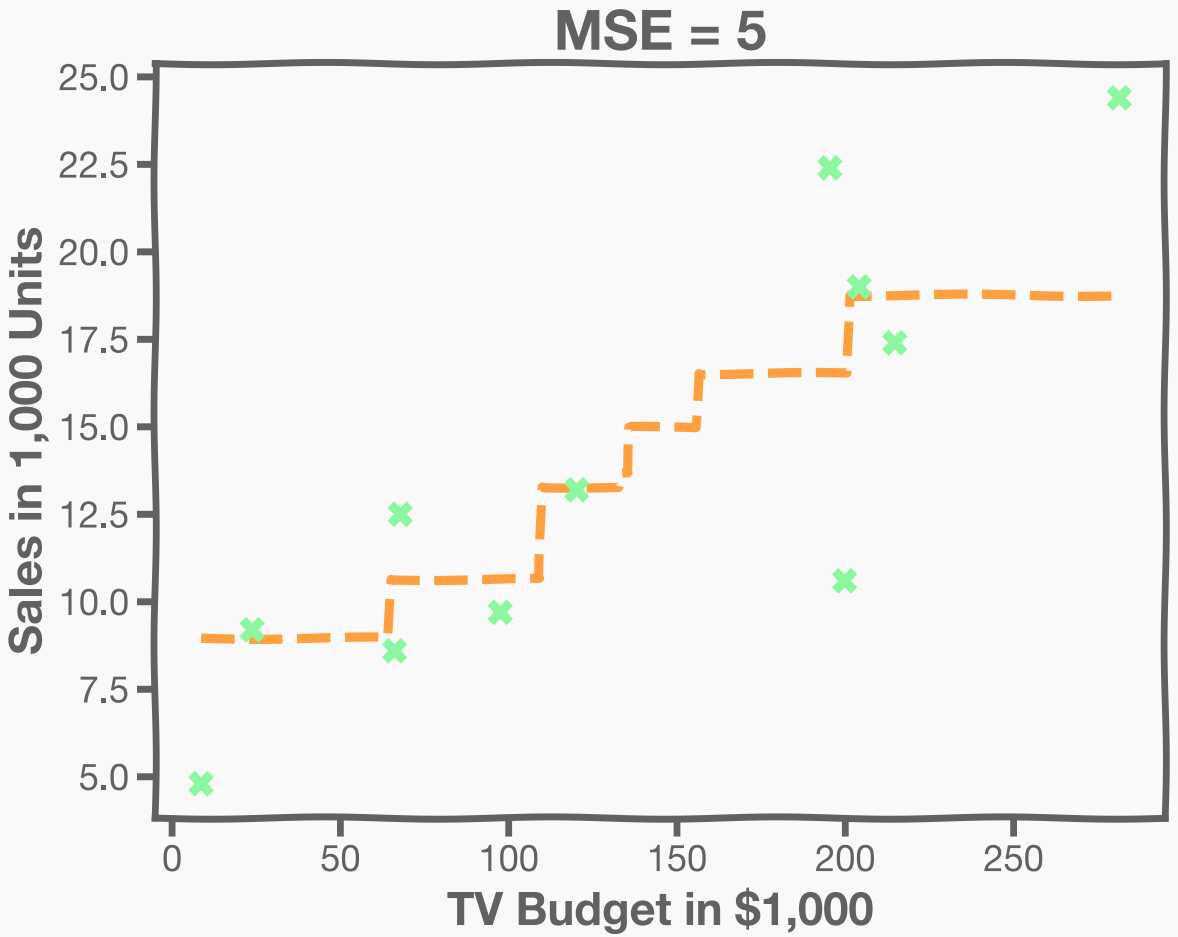
$A.$ $k = 3$

$B.$ $k = 3$ but why don't we experiment with different train/test splits?

Model Fitness

Model fitness

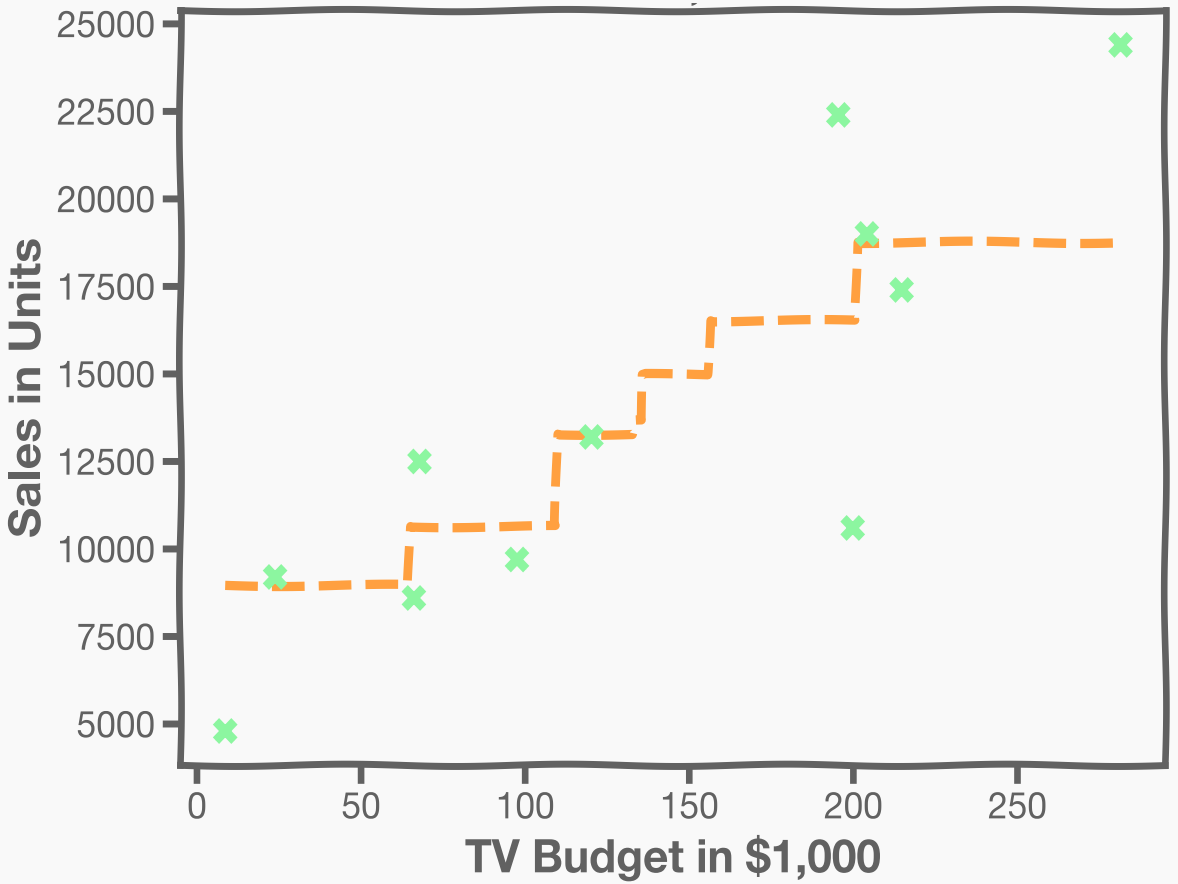
Calculate the MSE for $k = 3$ using a subset of the data.



Is MSE=5.0 good enough?

Model fitness

What would happen if we measure the *Sales* in single units instead of 1000 units?



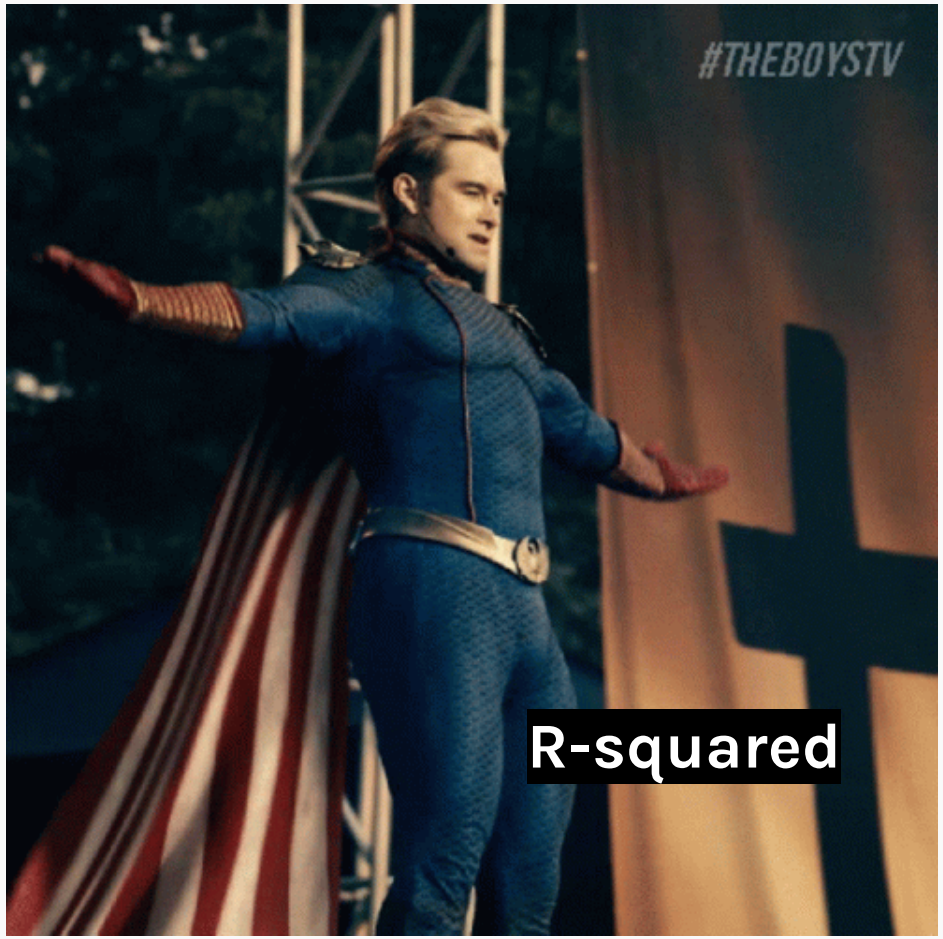
MSE is now 5,004,930.

Is that good?

Model fitness

It would be more meaningful to compare it to a benchmark or a known value.

A benchmark that isn't affected by the scale of the data.



R-squared

Though it is called R-squared, it is not the square of a quantity R as given in this formula.

We will use two reference models for comparison:

1. The simplest model, often considered the worst possible, where the predicted value $\hat{y}$ is the mean of all observations:

$$\hat{y} = \bar{y} = \frac{1}{n}\sum_i y_i$$

2. The ideal or best possible model, where the predicted value $\hat{y}$ is identical to the actual value y.

Using these two reference models, we define the R^2 (R-squared) value as:

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\bar{y} - y_i)^2}$$

R-squared

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\bar{y} - y_i)^2}$$

- If our model is as good as the mean value, $\bar{y}$, then $R^2 = 0$
- If our model is perfect, then $R^2 = 1$
- R^2 can be negative if the model is worse than the average. This can happen when we evaluate the model in the test set.

Lecture Outline

Part A: Statistical Modeling

k-Nearest Neighbors (kNN)

Part B: Error Evaluation and Model Comparison

How do we evaluate our model?

How do we choose from two different models?

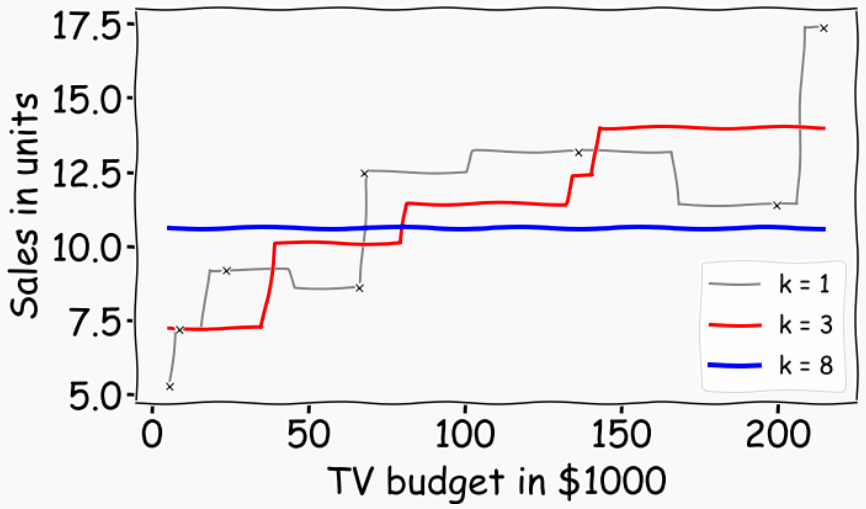
Part C: Linear Models

Linear Regression

Multi-linear regression

Linear Models

kNN model



Note that in building our kNN model for prediction (non-parametric), we did not compute a closed form for $\hat{f}$.

What if we ask the question:

"how much more sales do we expect if we double the TV advertising budget?"

Linear Regression

Linear Models

We can build a model by first assuming a simple form of f:

$$f(x) = \beta_0 + \beta_1 X$$

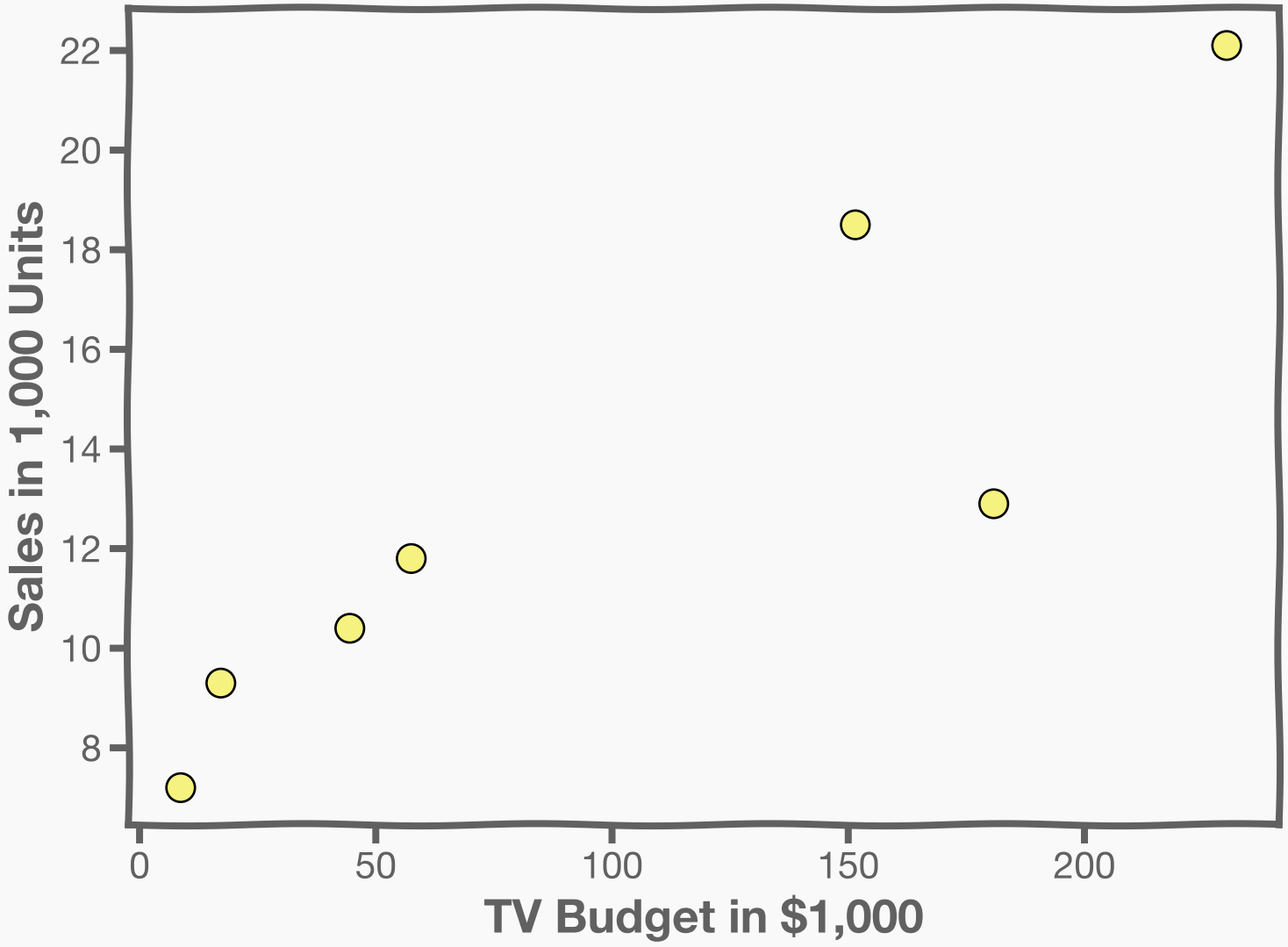
… then it follows that our estimate is:

$$\hat{y} = \hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 X$$

where $\hat{\beta}_1$ and $\hat{\beta}_0$ are **estimates** of β_1 and β_0 respectively, that we compute using observations.

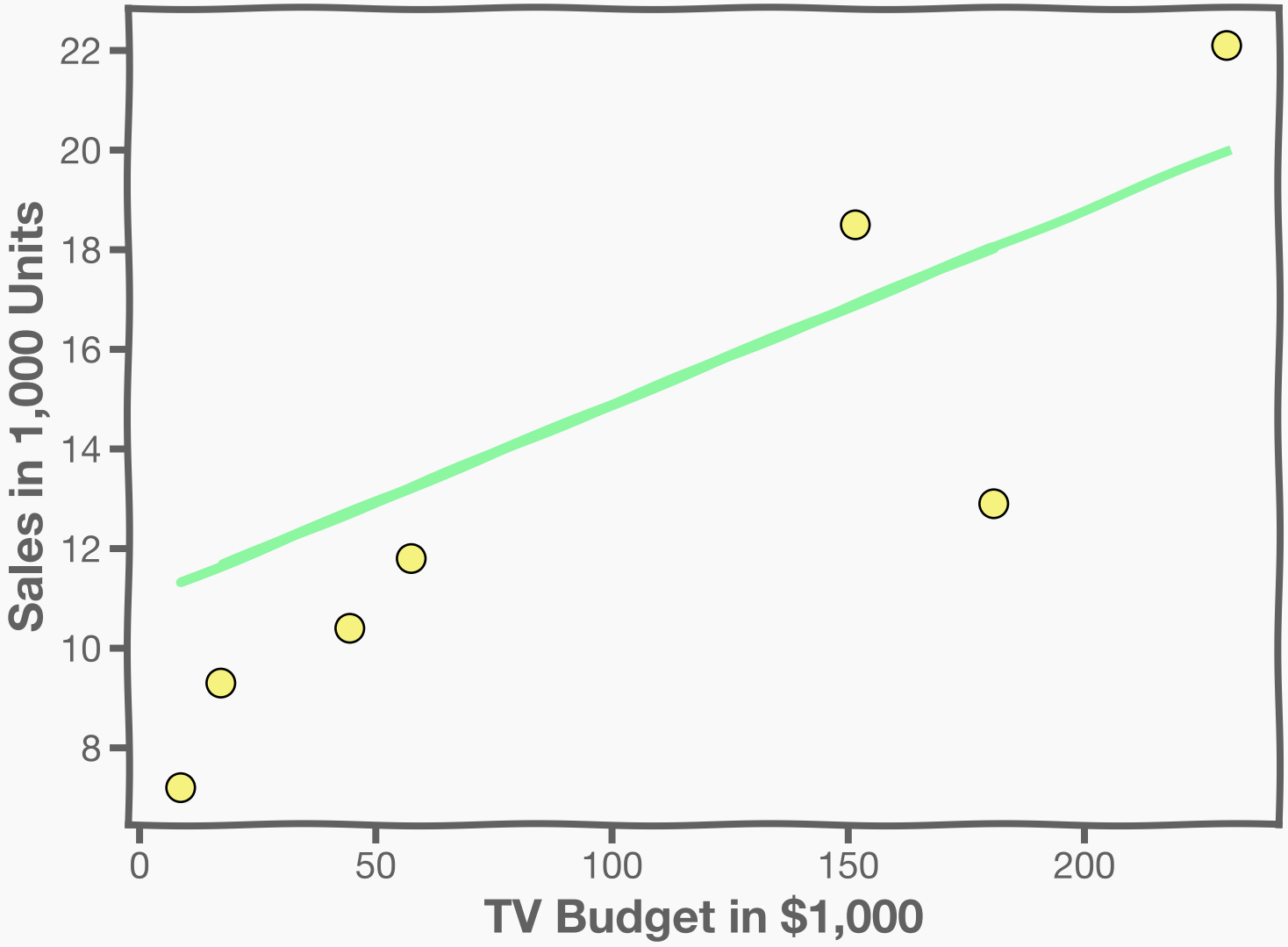
Estimate of the regression coefficients

For a given data set



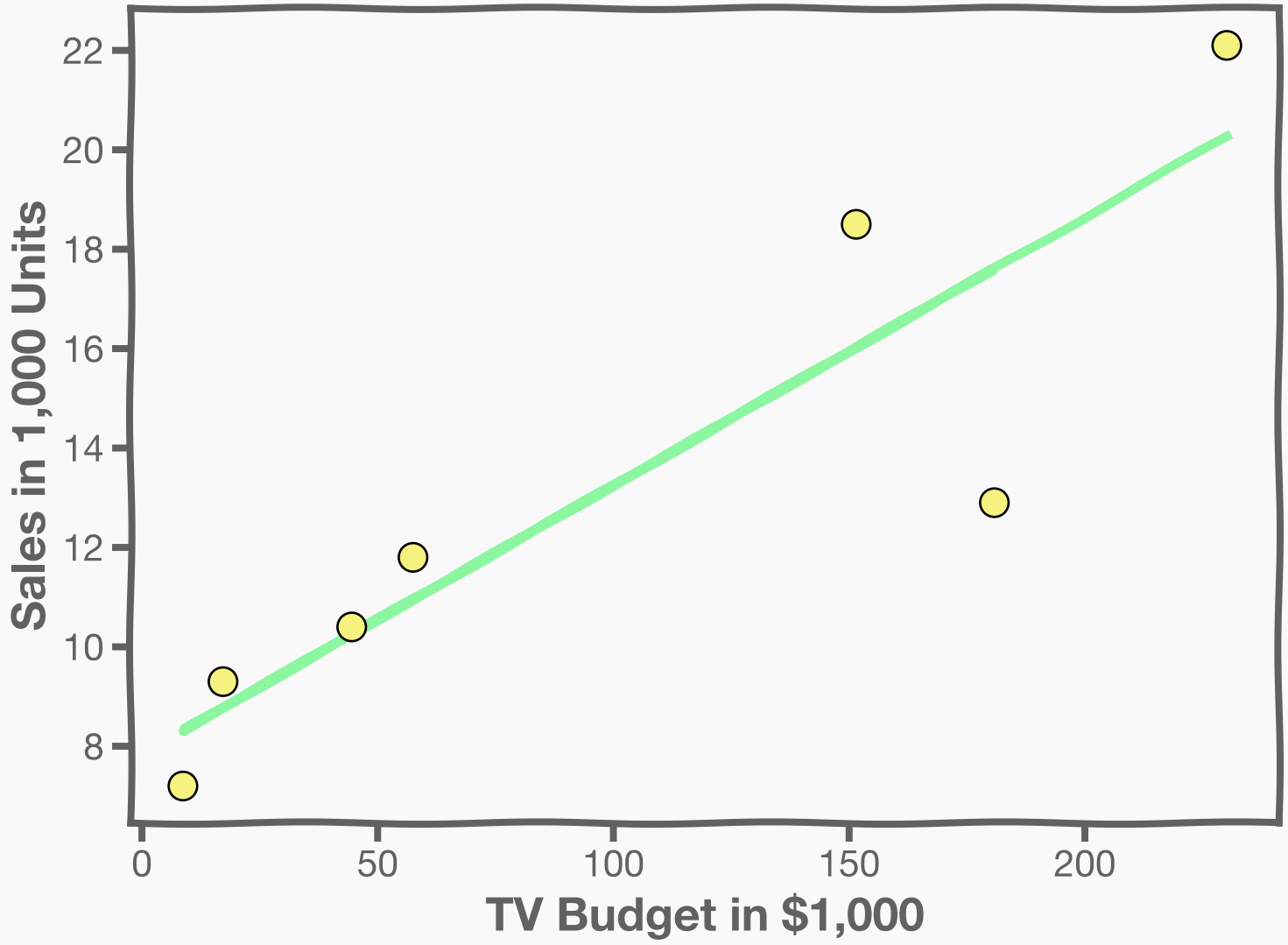
Estimate of the regression coefficients (cont)

Is this line good?



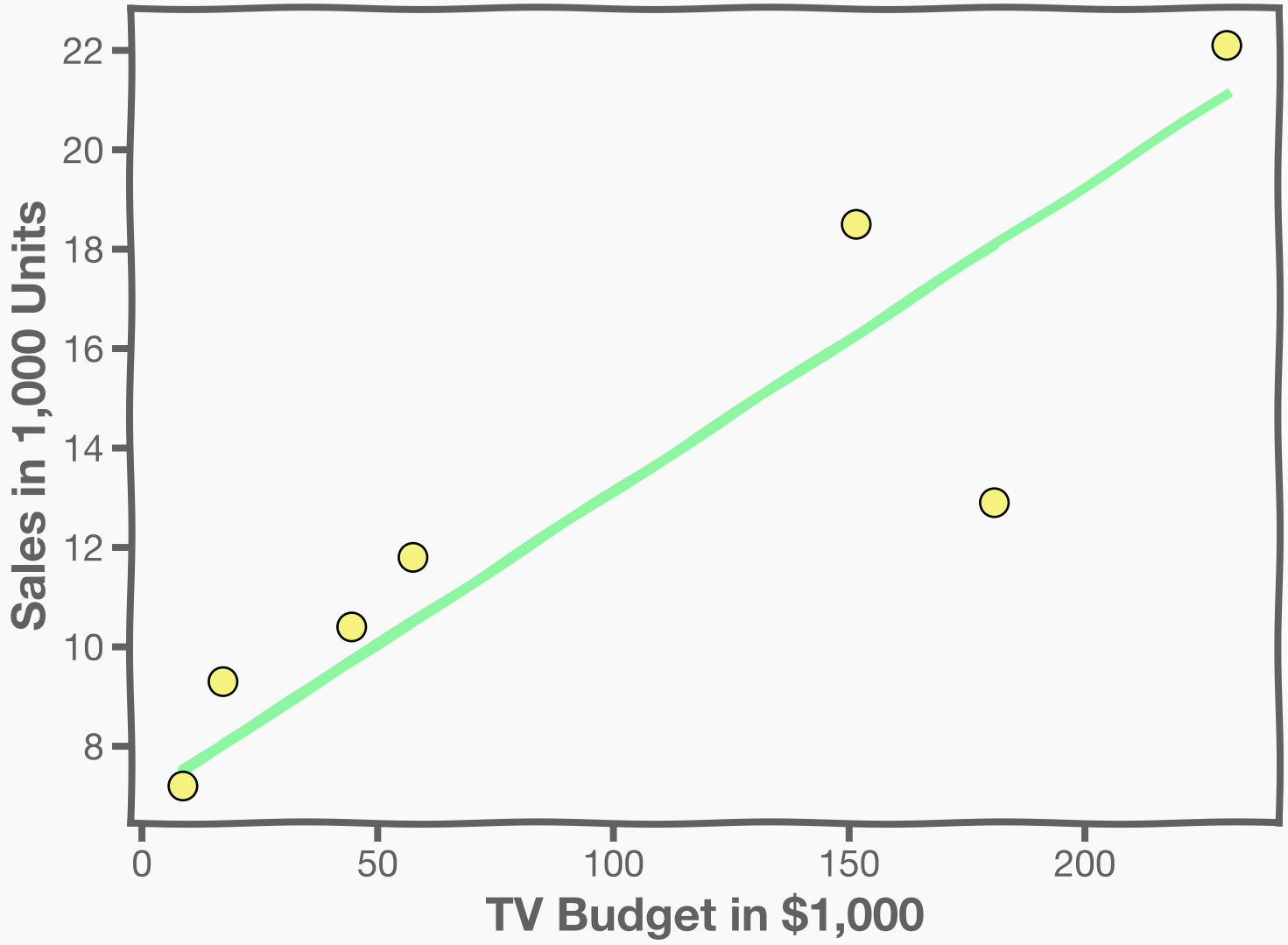
Estimate of the regression coefficients (cont)

Maybe this one?



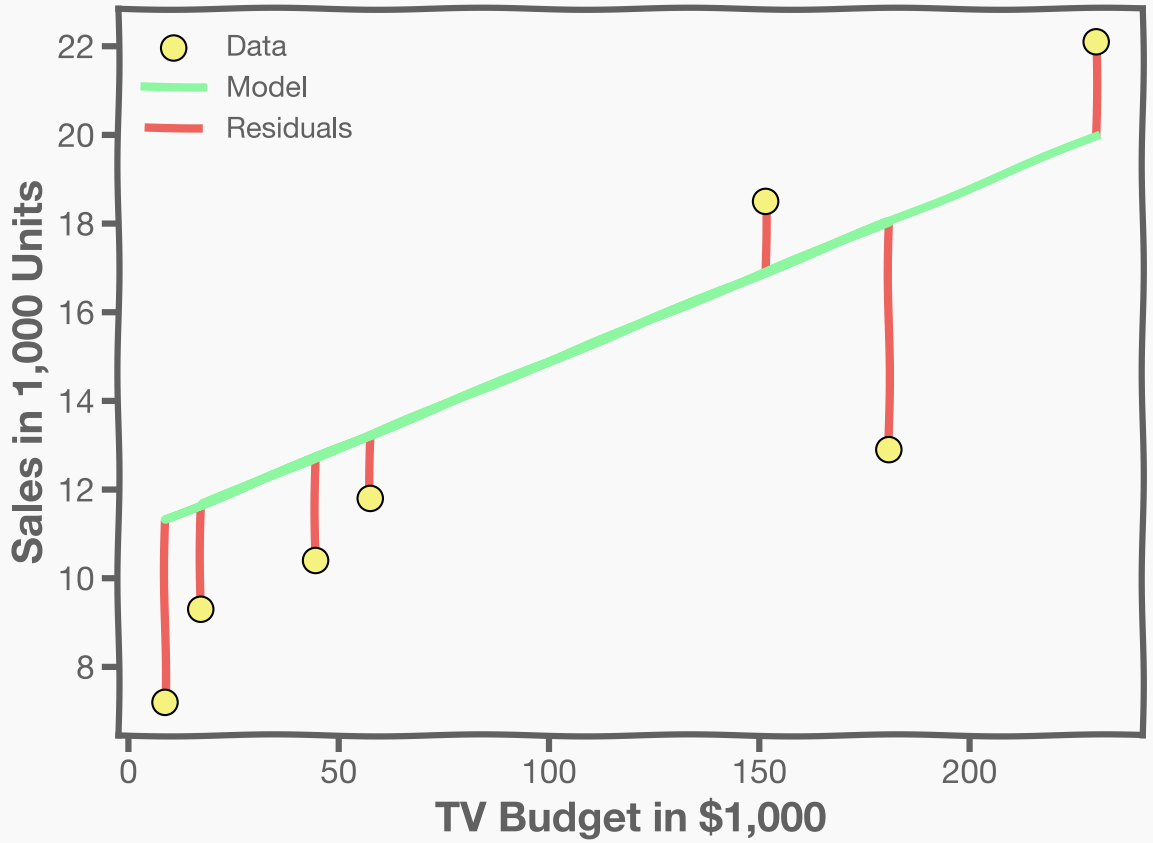
Estimate of the regression coefficients (cont)

Or this one?



Estimate of the regression coefficients (cont.)

Question: Which line is the best?



As before, for each observation (x_n, y_n), the absolute residuals, $r_i = |y_i - \hat{y}_i|$ quantify the error at each observation.

Estimate of the regression coefficients (cont.)

AGAIN, we use the MSE as our **loss function**,

$$L\left(\beta_0, \beta_1\right) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2$$

We choose β_1 and β_0 that minimizes the predictive errors made by our model, i.e., minimize our loss function.

Then the optimal values, $\hat{\beta}_0$ and $\hat{\beta}_1$, should be:

$$\widehat{\beta}_0, \widehat{\beta}_1 = \underset{\beta_0,\beta_1}{\operatorname{argmin}} L(\beta_0, \beta_1).$$

WE CALL THIS **FITTING** OR **TRAINING** THE MODEL

Introducing...

sklearn.linear_model.LinearRegression

Methods

fit(X, y[, sample_weight])	Fit linear model.
get_params([deep])	Get parameters for this estimator.
predict(X)	Predict using the linear model.
score(X, y[, sample_weight])	Return the coefficient of determination R^2 of the prediction.

```
>>> from sklearn.linear_model import LinearRegression
>>> reg = LinearRegression()
>>> reg.fit(X, y)
>>> reg.coef_
array([1.2])
>>> reg.intercept_
3.2
>>> reg.predict(np.array([[3]]))
array([16.])
```

RECAP: Exercise

```
>>> from sklearn.linear_model import LinearRegression
>>> df = pd.read_csv('Advertising.csv')
>>> X= df[['TV']].values
>>> y = df['Sales'].values
```

RECAP: Exercise

```
>>> from sklearn.linear_model import LinearRegression
>>> df = pd.read_csv('Advertising.csv')
>>> X= df[['TV']].values
>>> y = df['Sales'].values
>>> reg = LinearRegression()
>>> reg.fit(X, y)
```

RECAP: Exercise

```
>>> from sklearn.linear_model import LinearRegression
>>> df = pd.read_csv('Advertising.csv')
>>> X= df[['TV']].values
>>> y = df['Sales'].values
>>> reg = LinearRegression()
>>> reg.fit(X, y)
>>> reg.coef_
array([[0.04665056]])
>>> reg.intercept_
array([7.08543108])
>>> reg.predict(np.array([[100]]))
array([[11.75048733]])
```

```
>>> reg.fit(X, y)
```

What is happening here?

Derivative definition

A derivative is the instantaneous rate of change of a single valued function. Given a function f(x) the derivati

$$f'($$

Quiz

If you have to guess someone's height, would you rather be told

Options:

A. Their weight, only
B. Their weight and biological sex
C. Their weight, biological sex, and income
D. Their weight, biological sex, income, and favorite number

Multi-linear Regression

Multi-Linear Regression

Of course, you'd always want as much data about a person as possible. Even though height and favorite number may not be strongly related, at worst you could just ignore the information on favorite number.

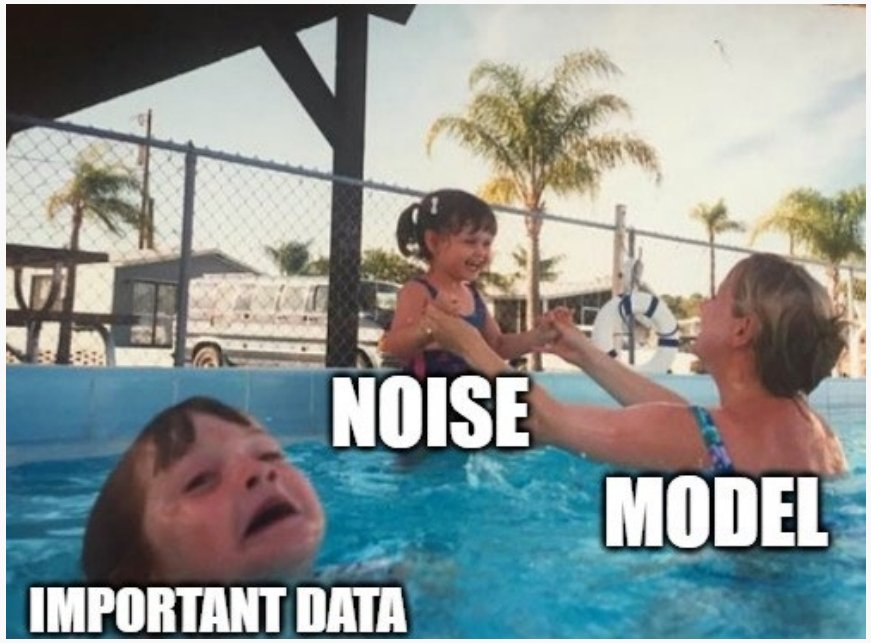
We want our models to be able to take in lots of data as they make their predictions.

This approach brings up a few questions.

Multi-Linear Regression

Data Noise

- Can too much irrelevant data introduce noise and make pattern detection difficult?



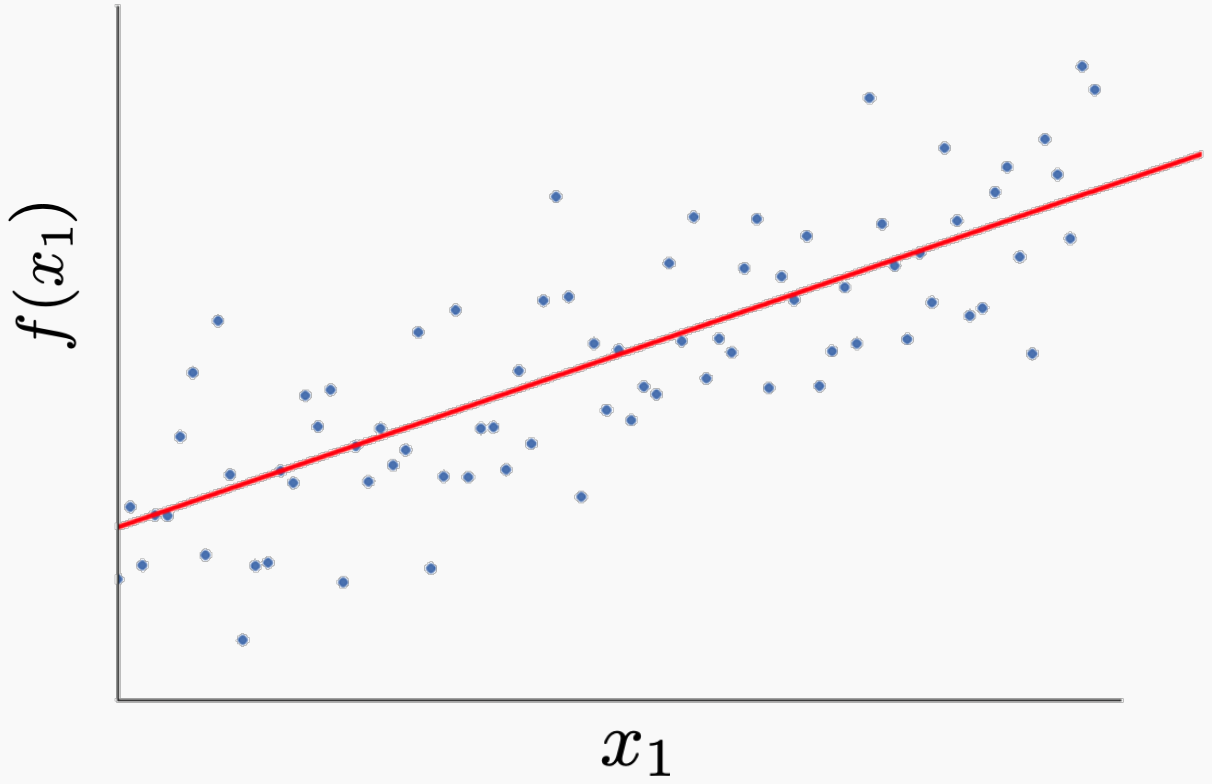
Ethical Considerations

- Are there privacy concerns related to collecting more data than needed?

Simple Linear Regression

In simple linear regression, we assume a simple basic form for f:

$$f(x) = \beta_0 + \beta_1 x$$

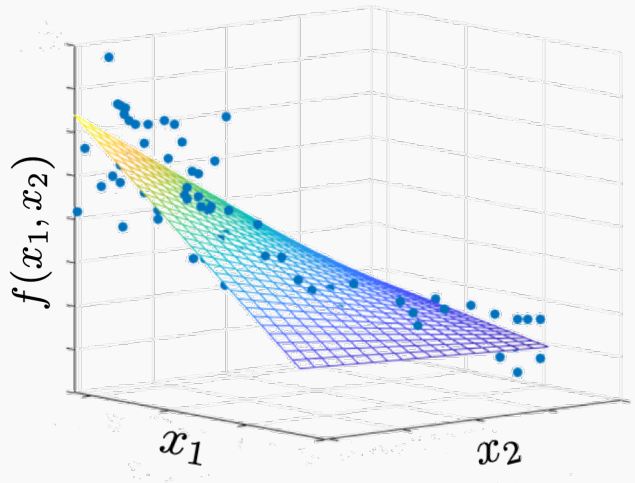


Linear Regression in n-D

In practice, it is unlikely that any response variable y depends solely on one predictor x. Rather, we expect that y is a function of multiple predictors $x_1, x_2, \dots, x_p$.

Using the notation:

$$\mathbf{y} = y_1, \dots, y_n, \qquad X = \boldsymbol{x}_1, \dots, \boldsymbol{x}_p \qquad \text{and} \qquad \boldsymbol{x}_j = x_{1j}, \dots, x_{nj}$$



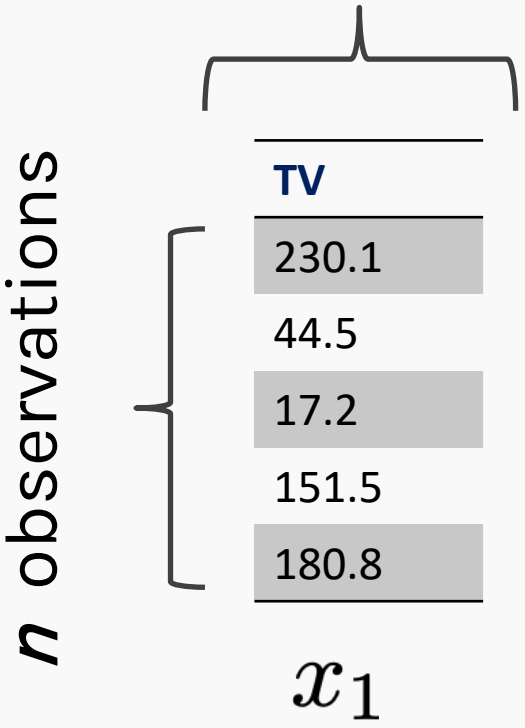
PROTOPAPAS

In multiple linear regression, we assume a similar form for f as in simple linear regression. We can assume a simple form for f a multilinear form:

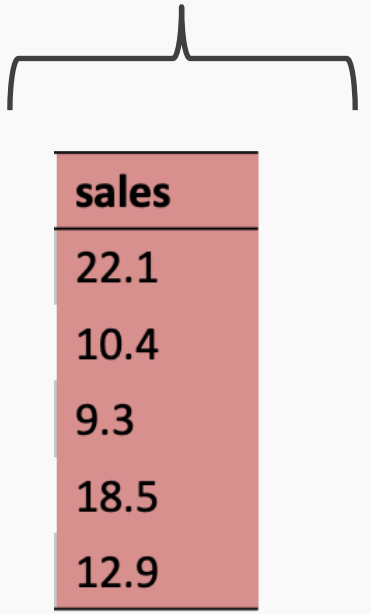
$$f(\boldsymbol{x}_1, \dots, \boldsymbol{x}_p) = \beta_0 + \beta_1 \boldsymbol{x}_1 + \cdots + \beta_p \boldsymbol{x}_p$$

Response vs. Predictor Variables

The Design Matrix



y:
The response variable



Response vs. Predictor Variables

The Design Matrix

n observations

TV	radio	newspaper
230.1	37.8	69.2
44.5	39.3	45.1
17.2	45.9	69.3
151.5	41.3	58.5
180.8	10.8	58.4

x_1 x_2 x_3

y:
The response variable

sales
22.1
10.4
9.3
18.5
12.9

Multi-Linear Regression, example

For our data

$$Sales = \beta_0 + \beta_1 \times TV + \beta_2 \times Radio + \beta_3 \times Newspaper$$

In linear algebra notation

$$\boldsymbol{Y} = \begin{pmatrix} Sales_1 \\ \vdots \\ Sales_n \end{pmatrix}, \boldsymbol{X} = \begin{pmatrix} 1 & TV_1 & Radio_1 & News_1 \\ \vdots & & \vdots & \vdots \\ 1 & TV_n & Radio_n & News_n \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_3 \end{pmatrix}$$

Multi-Linear Regression, example

For our data

$$Sales = \beta_0 + \beta_1 \times TV + \beta_2 \times Radio + \beta_3 \times Newspaper$$

In linear algebra notation

$$\boldsymbol{Y} = \begin{pmatrix} Sales_1 \\ \vdots \\ Sales_n \end{pmatrix}, \boldsymbol{X} = \begin{pmatrix} 1 & TV_1 & Radio_1 & News_1 \\ \vdots & & \vdots & \vdots \\ 1 & TV_n & Radio_n & News_n \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_3 \end{pmatrix}$$

Multi-Linear Regression, example

For our data

$$Sales = \beta_0 + \beta_1 \times TV + \beta_2 \times Radio + \beta_3 \times Newspaper$$

In linear algebra notation

$$\boldsymbol{Y} = \begin{pmatrix} Sales_1 \\ \vdots \\ Sales_n \end{pmatrix}, \boldsymbol{X} = \begin{pmatrix} 1 & TV_1 & Radio_1 & News_1 \\ \vdots & & \vdots & \vdots \\ 1 & TV_n & Radio_n & News_n \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_3 \end{pmatrix}$$

$$Sales_1 = (1 \quad TV_1 \quad Radio_1 \quad News_1) \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_3 \end{pmatrix}$$

Multi-Linear Regression, example

$$Sales_1 = (1 \quad TV_1 \quad Radio_1 \quad News_1)\begin{pmatrix}\beta_0 \\ \vdots \\ \beta_3\end{pmatrix}$$

$$Sales = \beta_0 + \beta_1 \times TV + \beta_2 \times Radio + \beta_3 \times Newspaper$$

$$Y = X\beta$$

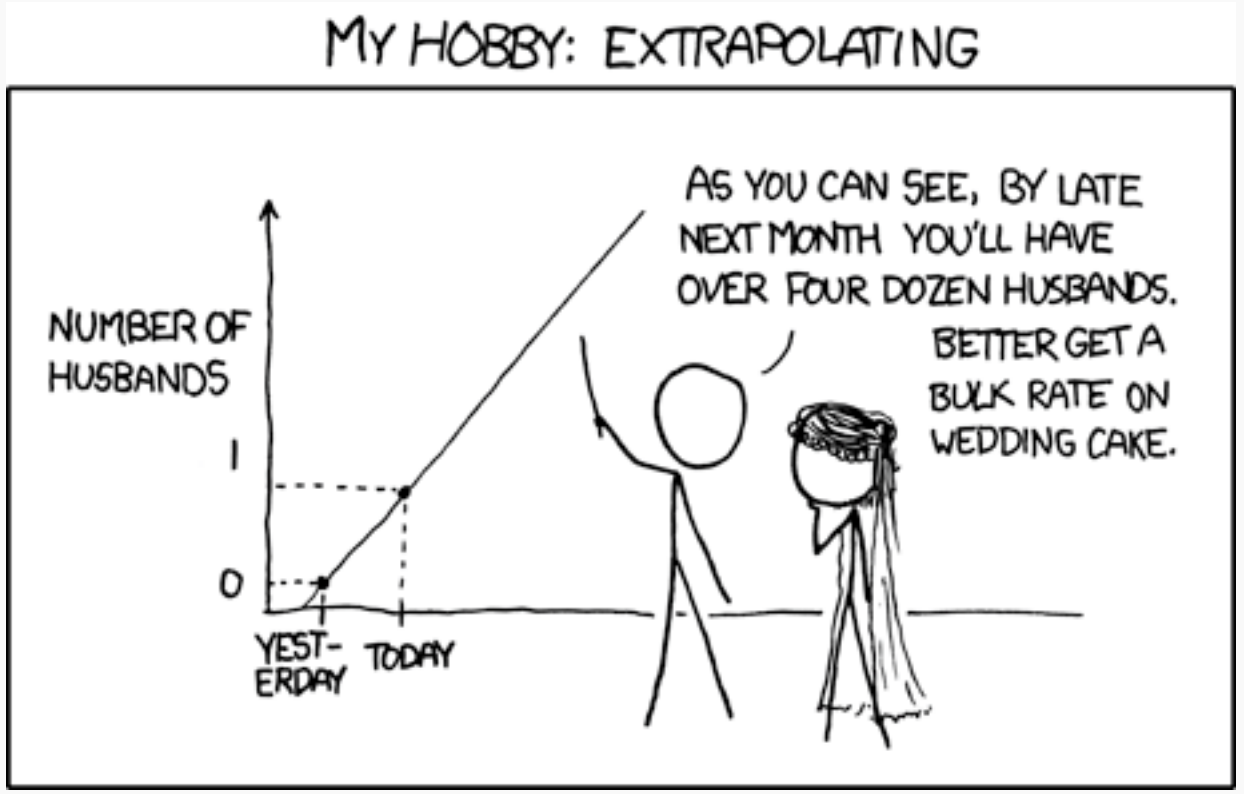
Multi-Linear Regression

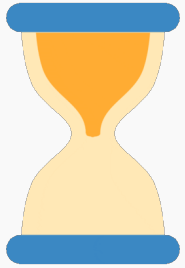
The model takes a simple algebraic form: $Y = X\beta$

We will again choose the **MSE** as our loss function, which can be expressed in vector notation as

$$MSE(\boldsymbol{\beta}) = \frac{1}{n}\|Y - X\beta\|^2$$

$$MSE(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - x_{i1}\beta_1 - x_{i2}\beta_2)^2$$





Digestion Time

Interpreting Model Parameters

Quiz question

In a simple linear regression model, you have the equation $Y=5+3X$. What does the coefficient 3 represent?

Options

A. The predicted value of Y when X=0

B. The change in Y for a one-unit change in X

C. The amount by which Y varies randomly around the line

D. None of the above

Interpreting Model Parameters in Simple Linear Regression

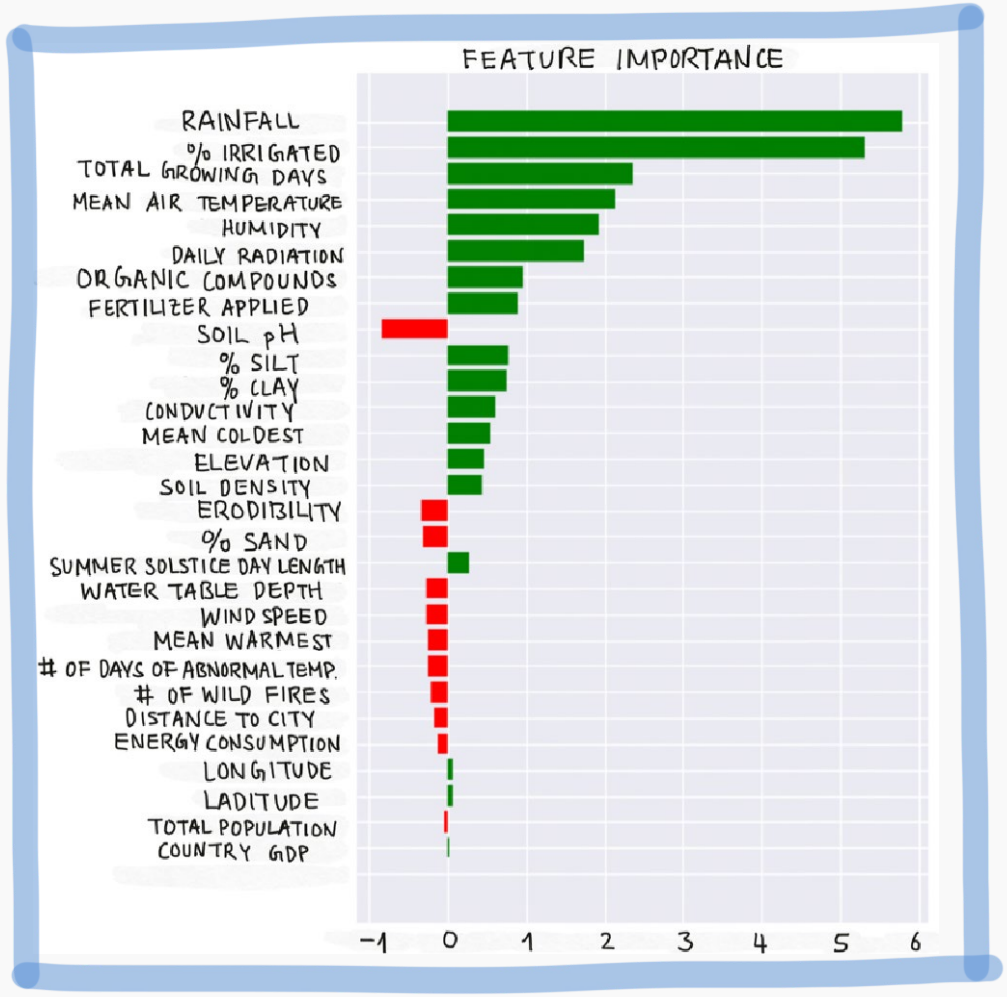
In the case of simple linear models, interpreting the model parameters is straightforward.

Interpretation

- β_0: Predicted value of *y* when *X*=0
- β_1: Change in *y* for a one-unit change in *X*

Interpreting multi-linear regression

In the case of simple linear models, interpreting the model parameters is straightforward.



When we have a large number of predictors: $X_1, \dots, X_J$, there will be a large number of model parameters, $\beta_1, \beta_2, \dots, \beta_J$.

Looking at the values of β's is impractical, so we visualize these values in a feature importance graph.

The feature importance graph shows which predictors has the most impact on the model's prediction.

Quiz question

In a multiple linear regression model, how does scaling the predictor variables affect the interpretation of feature importance based on the β coefficients?

Options

A. Scaling the predictors makes it easier to directly compare the importance of each feature based on their β coefficients.

B. Scaling the predictors makes all the features equally important.

C. Scaling the predictors increases the magnitude of β coefficients for less important features.

D. Scaling the predictors eliminates the need for β coefficients for feature importance.

Scaling

Understanding Scaling: Standardization & Normalization

Scaling transforms your data so that it fits within a specific range or distribution.

Standardization (Z-Score)

Transforms data to have mean = 0 and standard deviation = 1

$$\frac{X - mean}{std}$$

Normalization (Min-Max Scaling)

Rescales data to range between 0 and 1

$$\frac{X - min}{max - min}$$

Why Scale?

- Makes algorithms sensitive to feature scales perform better.
- Facilitates easier interpretation and analysis.

📚 **For More In-depth c**heck my notes and examples on EdStem!

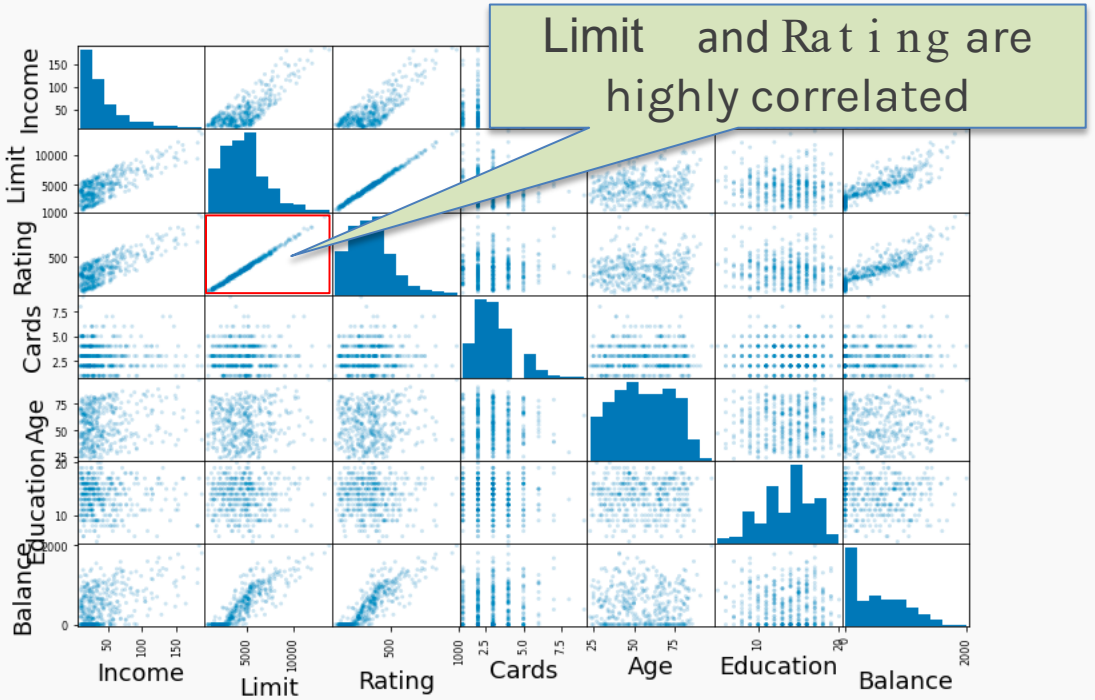
Collinearity

Collinearity

Collinearity refers to a situation where two or more predictors in a regression model are highly correlated with each other.

Collinearity affects our confidence in the estimated coefficients, making it challenging to assess the importance of individual predictors.

Collinearity



	Columns	Coefficients
0	Income	-7.802001
1	Limit	0.193077
2	Rating	1.102269
3	Cards	17.923274
4	Age	-0.634677
5	Education	-1.115028
6	Gender	10.406651
7	Student	426.469192
8	Married	-7.019100

	Columns	Coefficients
0	Income	-7.770915
1	Rating	3.976119
2	Cards	4.031215
3	Age	-0.669308
4	Education	-0.375954
5	Gender	10.368840
6	Student	417.417484
7	Married	-13.265344

Non-unique regression coefficients reduce model interpretability due to feature influence.

Positive coefficients for both `limit` and `rating` create ambiguity in attributing `balance` changes. Removing limit maintains model performance but alters coefficients.

Qualitative Predictors

Qualitative Predictors

So far, we have assumed that all variables are quantitative. But in practice, often some predictors are **qualitative**.

Example: The *credit data set* contains information about balance, age, cards, education, income, limit , and rating for a number of potential customers.

Income	Limit	Rating	Cards	Age	Education	Sex	Student	Married	Ethnicity	Balance
14.890	3606	283	2	34	11	Male	No	Yes	Caucasian	333
106.02	6645	483	3	82	15	Female	Yes	Yes	Asian	903
104.59	7075	514	4	71	11	Male	No	No	Asian	580
148.92	9504	681	3	36	11	Female	No	No	Asian	964
55.882	4897	357	2	68	16	Male	No	Yes	Caucasian	331

Qualitative Predictors

So far, we have assumed that all variables are quantitative. But in practice, often some predictors are **qualitative**.

Example: The *credit data set* contains information about balance, age, cards, education, income, limit , and rating for a number of potential customers.

Income	Limit	Rating	Cards	Age	Education	Sex	Student	Married	Ethnicity	Balance
14.890	3606	283	2	34	11	Male	No	Yes	Caucasian	333
106.02	6645	483	3	82	15	Female	Yes	Yes	Asian	903
104.59	7075	514	4	71	11	Male	No	No	Asian	580
148.92	9504	681	3	36	11	Female	No	No	Asian	964
55.882	4897	357	2	68	16	Male	No	Yes	Caucasian	331

Quiz question

You have a dataset with a column named 'Student' containing values 'Yes' and ' 'No'. How would you encode this column as a binary variable?

Options

A. Replace ‘No' with 0 and ‘Yes' with 1

B. Replace ‘No’ with 1 and ‘Yes' with 2

C. Replace ‘No’ with 1 and ‘Yes' with 0

D. Replace ‘No' with ‘N' and ‘Yes' with ‘Y'

Qualitative Predictors

If the predictor takes only two values, then we create an **indicator** or **dummy variable** that takes on two possible numerical values.

For example, for the sex column, we create a new variable:

$$x_i = \begin{cases} 1 & \text{if } i\,\text{th person is female} \\ 0 & \text{if } i\,\text{th person is male} \end{cases}$$

We then use this variable as a predictor in the regression equation.

$$y_i = \beta_0 + \beta_1 x_i = \begin{cases} \beta_0 + \beta_1 & \text{if } i \text{ th person is female} \\ \beta_0 & \text{if } i \text{ th person is male} \end{cases}$$

Quiz question

What is interpretation of β_0 and β_1?

Select all that apply.

$$y_i = \beta_0 + \beta_1 x_i = \begin{cases} \beta_0 + \beta_1 & \text{if } i \text{ th person is female} \\ \beta_0 & \text{if } i \text{ th person is male} \end{cases}$$

Options

A. β_0 represents the expected value of **balance** for males.

B. β_0 represents the difference in **balance** between males and females.

C. $\beta_0 + \beta_1$ represents the expected in **balance** for females

D. β_1 the average difference in **balance** between females and males.

More than two levels: One hot encoding

Why?

Often, the qualitative predictor takes more than two values (e.g. ethnicity in the credit data).

In this situation, a single dummy variable cannot represent all possible values.

We create additional dummy variable as:

$$x_{i,1} = \begin{cases} 1 & \text{if } i\text{th person is Asian} \\ 0 & \text{if } i\text{th person is not Asian} \end{cases}$$

$$x_{i,2} = \begin{cases} 1 & \text{if } i\text{th person is Caucasian} \\ 0 & \text{if } i\text{th person is not Caucasian} \end{cases}$$

More than two levels: One hot encoding

We then use these variables as predictors, the regression equation becomes:

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} = \begin{cases} \beta_0 + \beta_1 & \text{if } i \text{ th person is Asian} \\ \beta_0 + \beta_2 & \text{if } i \text{ th person is Caucasian} \\ \beta_0 & \text{if } i \text{ th person is AfricanAmerican} \end{cases}$$

Question: What is the interpretation of $\beta_0, \beta_1, \beta_2$?

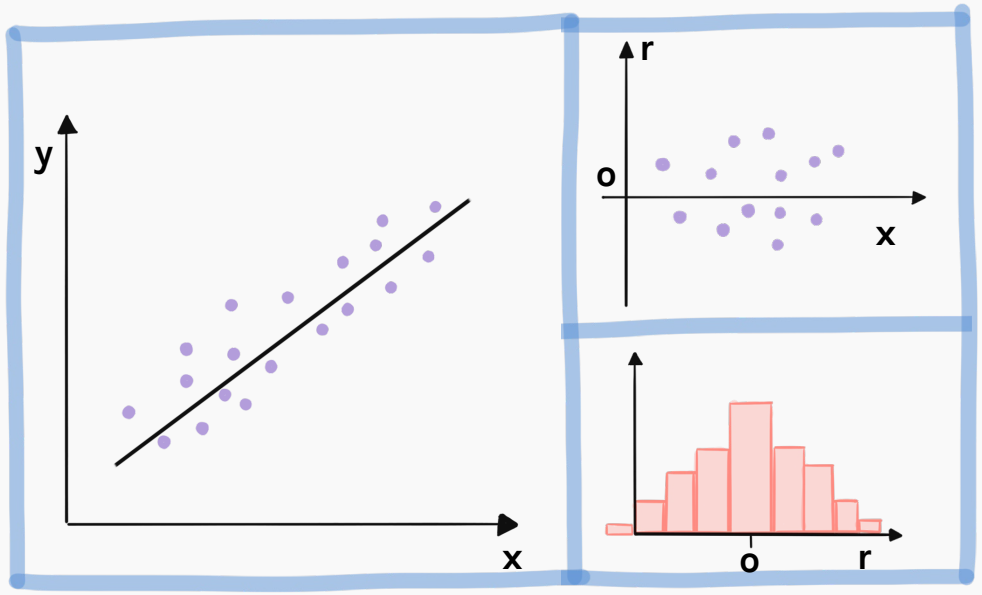
Beyond linearity

So far, we assumed:

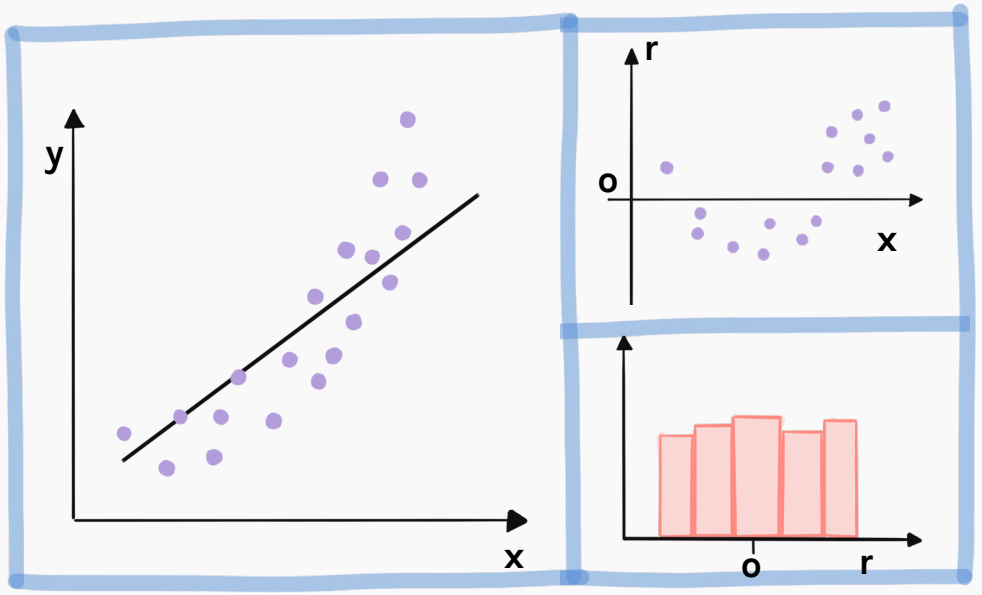
- linear relationship between X and Y
- the residuals $r_i = y_i - \hat{y}_i$ were uncorrelated (taking the average of the square residuals to calculate the MSE implicitly assumed uncorrelated residuals)

These assumptions need to be verified using the data. This is often done by **visually inspecting the residuals.**

Residual Analysis



<u>Linear assumption is correct.</u> There is no obvious relationship between residuals and x. Histogram of residuals is symmetric and normally distributed.



<u>Linear assumption is incorrect</u>. There is an obvious relationship between residuals and x. Histogram of residuals is symmetric but **not** normally distributed.

Note: For multi-regression, we plot the residuals vs predicted, $\hat{y}$, since there are too many x's and that could wash out the relationship.

Thank you!