
INFO 2950: Intro to Data Science

Lecture 8
2023-09-18

Agenda

1. Admin
2. Reshaping
3. Regression Review
4. Transformations
 - a. What are they
 - b. Interpretations

Admin

- Phase 1 is due this Thursday 9/21
 - Each team will receive an email from your TA grader on Tuesday containing their Github username so you can add them to your repo

Admin

- Phase 1 is due this Thursday 9/21
 - Each team will receive an email from your TA grader on Tuesday containing their Github username so you can add them to your repo
- HW3 is due Tuesday 9/26, with the FINAL late submission date of Friday 9/29 so that we can release solutions before the weekend

Admin

- Phase 1 is due this Thursday 9/21
 - Each team will receive an email from your TA grader on Tuesday containing their Github username so you can add them to your repo
- HW3 is due Tuesday 9/26, with the FINAL late submission date of Friday 9/29 so that we can release solutions before the weekend
- Midterm is 10/2; let us know early if you need to take a make-up exam

HW3 errata

#A2 output should look like this:

```
(1008, 6)
```

	ParkName	States	Region	Acres	Year	Visitors
0	Gates of the Arctic	Alaska	Alaska	7523897.45	2021	7362.0
1	Wrangell-St. Elias	Alaska	Alaska	8323146.48	2021	50189.0
2	Katmai	Alaska	Alaska	3674529.33	2021	24764.0
3	Glacier Bay	Alaska	Alaska	3223383.43	2021	89768.0
4	North Cascades	Washington	Pacific West	504780.94	2021	17855.0

HW3 hint

#A3 might give a hint if your #B2 is different from our output!

HW3 hint

#A3 might give a hint if your #B2 is different from our output!

If you're on a machine that only stores 4 bytes instead of 8 bytes (int32 instead of int64), you might get different values when very very big multiplication or very very small multiplication happens

HW3 hint

#A3 might give a hint if your #B2 is different from our output!

If you're on a machine that only stores 4 bytes instead of 8 bytes (int32 instead of int64), you might get different values when very very big multiplication or very very small multiplication happens



INFO2950_Lec2_20230823

File Edit View Insert Format

Interview Question: data types

- What does `print(1.81e308)` display?
inf
- Why? **System overflow at 2^{1024}**

HW3 errata

- Notes on the above hints / errata are already updated on Canvas (but the content of the questions are all the same; you can turn in your ipynb based on either download)
- Going forward, we will consolidate fixes from Ed discussions into Canvas / lecture announcements!
- We are human and make mistakes; please let us know if you catch other issues & we will fix them ASAP!

Our end goal: everyone aces 2950!

- Most of your 2950 grade comes from two places:
 - Prelim & Final Exam
 - Final Project

Our end goal: everyone aces 2950!

- Most of your 2950 grade comes from two places:
 - Prelim & Final Exam
 - ^-- practice using **whiteboards**
 - Final Project
 - ^-- practice using **homework**

What is the point of whiteboards?

- Whiteboard questions are direct practice for prelim & final exam questions
 - We can't ask you to code for INFO 2950 exams: they're pencil & paper exams!

What is the point of whiteboards?

- Whiteboard questions are direct practice for prelim & final exam questions
 - We can't ask you to code for INFO 2950 exams: they're pencil & paper exams!
- Whiteboards help you stay engaged in class & help us see if we should re-cover material
- Interviews can often involve “whiteboarding”!

What is the point of homework?

- Homework is direct practice for your final project
- This is where you get hands-on Python experience
 - These are skills to add to your portfolio for job applications


What is the point of homework?

- Homework is direct practice for your final project
- This is where you get hands-on Python experience
 - These are skills to add to your portfolio for job applications
- **Lecture slides + using Python documentation + trial & error (going to “the gym”) = how you get better at homework**

What is the point of homework?

- Homework is direct practice for your final project
- This is where you get hands-on Python experience
 - These are skills to add to your portfolio for job applications
- **Lecture slides + using Python documentation + trial & error (going to “the gym”) = how you get better at homework**

Everyone goes at a different pace for this! Take your time & don't be shy in going to Student Hours!



Attendance!



Pre-2950



Post-2950



tinyurl.com/3m5p3ye3

Reshaping

SEMINAR SERIES

My fourteen year fight with data reshaping

Friday, January 10th, 2020
11:30AM – 12:30PM
UTHealth SPH E-101 - Auditorium

Presenter:

Dr. Hadley Wickham
Chief Scientist
RStudio



Abstract

The reshape package appeared on CRAN in 2005, followed by reshape2 in 2010, and tidyr in 2014. Across all three packages, there have been over 40 releases over the last 14 years. What makes reshaping/tidying data so hard? Why has it taken so many attempts to get it right? (And is it really right, or just right for now?) I'll use reshape-reshape2-tidyr as a lens think about the co-evolution of data structures and code, managing change, interface design, and the intertwined nature of R as a programming language and environment for interactive data analysis.

Reshaping dataframes: what?

- Sometimes your data will come to you in a form where you either wish the columns were rows, or vice versa

Reshaping dataframes: why?

- Sometimes your data will come to you in a form where you either wish the columns were rows, or vice versa
- Getting it into the format you want can help you do aggregations (e.g. group by, sum), mutations (make a new column using old columns), run regressions (using x and y as columns), etc.

Which is better for... Plotting? Regression?

input_x	is_high	output_y
2023-09-19	high	77
2023-09-19	low	58
2023-09-20	high	73
2023-09-20	low	55
2023-09-21	high	80
2023-09-21	low	57



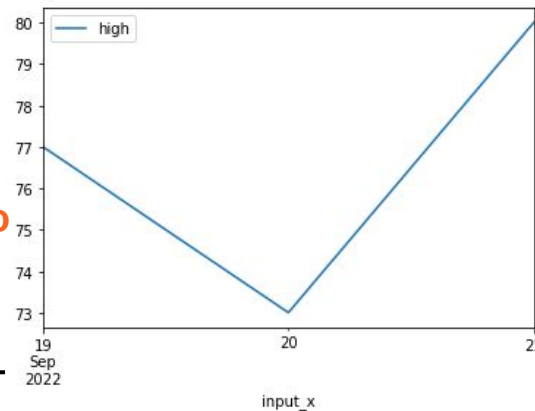
input_x	high	low
2023-09-19	77	58
2023-09-20	73	55
2023-09-21	80	57

input_x	is_high	output_y
2023-09-19	high	77
2023-09-19	low	58
2023-09-20	high	73
2023-09-20	low	55
2023-09-21	high	80
2023-09-21	low	57



input_x	high	low
2023-09-19	77	58
2023-09-20	73	55
2023-09-21	80	57

`df.plot('input_x', 'high')`



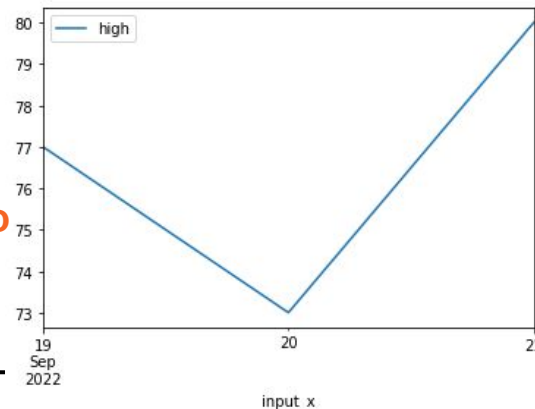
Easier to plot a subset of data, or do $high \sim input_x$

input_x	is_high	output_y
2023-09-19	high	77
2023-09-19	low	58
2023-09-20	high	73
2023-09-20	low	55
2023-09-21	high	80
2023-09-21	low	57



input_x	high	low
2023-09-19	77	58
2023-09-20	73	55
2023-09-21	80	57

```
df.plot('input_x', 'high')
```



Easier to plot a
subset of data, or do
 $high \sim input_x$

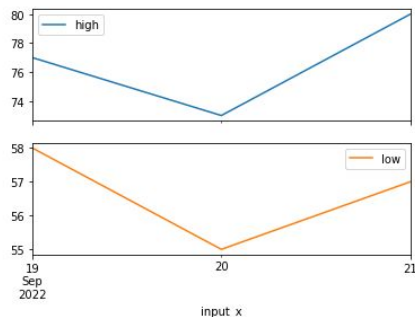
*note: x's need to be numeric in a regression, so you'd likely convert input_x to be # days since a certain date

input_x	is_high	output_y
2023-09-19	high	77
2023-09-19	low	58
2023-09-20	high	73
2023-09-20	low	55
2023-09-21	high	80
2023-09-21	low	57

input_x	high	low
2023-09-19	77	58
2023-09-20	73	55
2023-09-21	80	57



Easier to plot grouped data, or
do multivariable regression



```
g = sns.FacetGrid(df, row="is_high")
g.map(sns.lineplot, "input_x", "output_y")
```

When to reshape?

input_x	is_high	output_y
2023-09-19	high	77
2023-09-19	low	58
2023-09-20	high	73
2023-09-20	low	55
2023-09-21	high	80
2023-09-21	low	57



input_x	high	low
2023-09-19	77	58
2023-09-20	73	55
2023-09-21	80	57

should be same units
for interpretability

input_x	time_of_day	output_y
2023-09-19	high	77
2023-09-19	avg	67.5
2023-09-19	low	58
2023-09-20	high	73
2023-09-20	avg	64
2023-09-20	low	55
2023-09-21	high	80
2023-09-21	avg	68.5
2023-09-21	low	57

Which one is...
“Wide” vs. “Long”*?

***long a.k.a. tidy, skinny, tall**

input_x	high	low	avg
2023-09-19	77	58	67.5
2023-09-20	73	55	64.0
2023-09-21	80	57	68.5

input_x	time_of_day	output_y
2023-09-19	high	77
2023-09-19	avg	67.5
2023-09-19	low	58
2023-09-20	high	73
2023-09-20	avg	64
2023-09-20	low	55
2023-09-21	high	80
2023-09-21	avg	68.5
2023-09-21	low	57

“Long”



“Wide”



input_x	high	low	avg
2023-09-19	77	58	67.5
2023-09-20	73	55	64.0
2023-09-21	80	57	68.5

Reshaping: how?

- When in doubt, Google search for how to convert from “long to wide” or “wide to long” with Pandas

Reshaping: how?

- When in doubt, Google search for how to convert from “long to wide” or “wide to long” with Pandas
- Many methods, some fancier (e.g. multiple columns at a time)

Long to wide	Wide to long
<code>pivot()</code> <code>unstack()</code> <code>long_to_wide()</code>	<code>melt()</code> <code>stack()</code> <code>wide_to_long()</code>

Reshaping: how?

- When in doubt, Google search for how to convert from “long to wide” or “wide to long” with Pandas
- Many methods, some fancier (e.g. multiple columns at a time)



Long to wide	Wide to long
<code>pivot()</code> <code>unstack()</code> <code>long_to_wide()</code>	<code>melt()</code> <code>stack()</code> <code>wide_to_long()</code>

Reshaping: how?

- When in doubt, Google search for how to convert from “long to wide” or “wide to long” with Pandas
- Many methods, some fancier (e.g. multiple columns at a time)



Long to wide	Wide to long
<code>pivot()</code> <code>unstack()</code> <code>long_to_wide()</code>	<code>melt()</code> <code>stack()</code> <code>wide_to_long()</code>

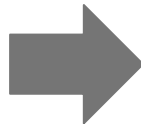
Start with 'wide_df'

input_x	high	low
2023-09-19	77	58
2023-09-20	73	55
2023-09-21	80	57

To make this long, what do we use: melt or pivot?

‘wide_df’ to long: melt

input_x	high	low
2023-09-19	77	58
2023-09-20	73	55
2023-09-21	80	57



input_x	is_high	output_y
2023-09-19	high	77
2023-09-19	low	58
2023-09-20	high	73
2023-09-20	low	55
2023-09-21	high	80
2023-09-21	low	57

id_vars = a list of column names
that should remain the same

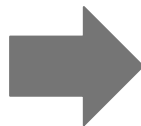
value_vars = a list of column names
that are going to be reshaped

'wide_df' to long: melt

input_x	high	low
2023-09-19	77	58
2023-09-20	73	55
2023-09-21	80	57

id_vars = a list of column names that should remain the same

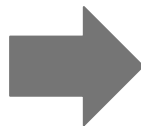
value_vars = a list of column names that are going to be reshaped



input_x	is_high	output_y
2023-09-19	high	77
2023-09-19	low	58
2023-09-20	high	73
2023-09-20	low	55
2023-09-21	high	80
2023-09-21	low	57

'wide_df' to long: melt

input_x	high	low
2023-09-19	77	58
2023-09-20	73	55
2023-09-21	80	57



input_x	is_high	output_y
2023-09-19	high	77
2023-09-19	low	58
2023-09-20	high	73
2023-09-20	low	55
2023-09-21	high	80
2023-09-21	low	57

id_vars = a list of column names
that should remain the same

value_vars = a list of column names
that are going to be reshaped

‘wide_df’ to long: melt

input_x	high	low
2022-09-19	77	58
2022-09-20	73	55
2022-09-21	80	57

id_vars = a list of column names
that should remain the same

value_vars = a list of column names
that are going to be reshaped

```
wide_df.melt(  
    id_vars = 'input_x',  
    value_vars = ['high', 'low']  
)
```

‘wide_df’ to long: melt

input_x	high	low
2022-09-19	77	58
2022-09-20	73	55
2022-09-21	80	57

	input_x	variable	value
0	2022-09-19	high	77
1	2022-09-20	high	73
2	2022-09-21	high	80
3	2022-09-19	low	58
4	2022-09-20	low	55
5	2022-09-21	low	57

id_vars = a list of column names
that should remain the same

value_vars = a list of column names
that are going to be reshaped

```
wide_df.melt(  
    id_vars = 'input_x',  
    value_vars = ['high', 'low']  
)
```

‘wide_df’ to long: melt

input_x	high	low
2022-09-19	77	58
2022-09-20	73	55
2022-09-21	80	57

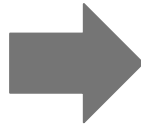
	input_x	variable	value
0	2022-09-19	high	77
1	2022-09-20	high	73
2	2022-09-21	high	80
3	2022-09-19	low	58
4	2022-09-20	low	55
5	2022-09-21	low	57

Default column names:

- “**variable**” contains what used to be the column name (in the list of value_vars)
- “**value**” is the contents of the cells in the columns within value_vars
- You can change these column names in the same melt() statement using **var_name** and **value_name** options

'wide_df' to long: melt

input_x	high	low
2022-09-19	77	58
2022-09-20	73	55
2022-09-21	80	57



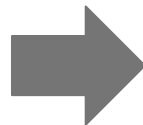
input_x	is_high	output_y
2022-09-19	high	77
2022-09-19	low	58
2022-09-20	high	73
2022-09-20	low	55
2022-09-21	high	80
2022-09-21	low	57

```
tall_df = wide_df.melt(  
    id_vars = 'input_x',  
    value_vars = ['high', 'low'],  
    var_name = 'is_high',  
    value_name = 'output_y')  
tall_df.sort_values(by=['input_x', 'is_high'])
```


Fill in the code

Table name: *wide_df*

input_x	2022-09-19	2022-09-20	2022-09-21
high	77	73	80
low	58	55	57



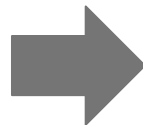
	input_x	variable	value
0	high	2022-09-19	77
1	low	2022-09-19	58
2	high	2022-09-20	73
3	low	2022-09-20	55
4	high	2022-09-21	80
5	low	2022-09-21	57

```
wide_df.melt(  
    id_vars = _____,  
    value_vars=[_____])
```

Fill in the code

Table name: *wide_df*

input_x	2022-09-19	2022-09-20	2022-09-21
high	77	73	80
low	58	55	57



	input_x	variable	value
0	high	2022-09-19	77
1	low	2022-09-19	58
2	high	2022-09-20	73
3	low	2022-09-20	55
4	high	2022-09-21	80
5	low	2022-09-21	57

```
wide_df.melt(  
    id_vars = 'input_x',  
    value_vars=['2022-09-19',  
                '2022-09-20', '2022-09-21'])
```

Remember your column
name quotation marks!!

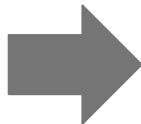
Start with 'tall_df'

input_x	is_high	output_y
2022-09-19	high	77
2022-09-19	low	58
2022-09-20	high	73
2022-09-20	low	55
2022-09-21	high	80
2022-09-21	low	57

To make this wide, what do we use: melt or pivot?

'tall_df' to wide: pivot

input_x	is_high	output_y
2022-09-19	high	77
2022-09-19	low	58
2022-09-20	high	73
2022-09-20	low	55
2022-09-21	high	80
2022-09-21	low	57



input_x	high	low
2022-09-19	77	58
2022-09-20	73	55
2022-09-21	80	57

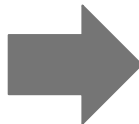
index = (a list of) column names that should remain the same

columns = (a list of) variable to split into separate columns

values = values to fill the new columns

'tall_df' to wide: pivot

input_x	is_high	output_y
2022-09-19	high	77
2022-09-19	low	58
2022-09-20	high	73
2022-09-20	low	55
2022-09-21	high	80
2022-09-21	low	57



input_x	high	low
2022-09-19	77	58
2022-09-20	73	55
2022-09-21	80	57

index = (a list of) column names that should remain the same

columns = (a list of) variable to split into separate columns

values = values to fill the new columns

'tall_df' to wide: pivot

input_x	is_high	output_y
2022-09-19	high	77
2022-09-19	low	58
2022-09-20	high	73
2022-09-20	low	55
2022-09-21	high	80
2022-09-21	low	57



input_x	high	low
2022-09-19	77	58
2022-09-20	73	55
2022-09-21	80	57

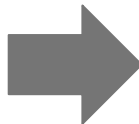
index = (a list of) column names that should remain the same

columns = (a list of) variable to split into separate columns

values = values to fill the new columns

'tall_df' to wide: pivot

input_x	is_high	output_y
2022-09-19	high	77
2022-09-19	low	58
2022-09-20	high	73
2022-09-20	low	55
2022-09-21	high	80
2022-09-21	low	57



input_x	high	low
2022-09-19	77	58
2022-09-20	73	55
2022-09-21	80	57

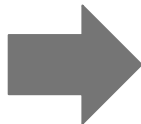
index = (a list of) column names that should remain the same

columns = (a list of) variable to split into separate columns

values = values to fill the new columns

'tall_df' to wide: pivot

input_x	is_high	output_y
2022-09-19	high	77
2022-09-19	low	58
2022-09-20	high	73
2022-09-20	low	55
2022-09-21	high	80
2022-09-21	low	57



is_high	high	low
input_x		
2022-09-19	77	58
2022-09-20	73	55
2022-09-21	80	57

```
tall_df.pivot(  
    index = 'input_x',  
    columns='is_high',  
    values='output_y')
```


'tall_df' to wide: pivot

input_x	is_high	output_y
2022-09-19	high	77
2022-09-19	low	58
2022-09-20	high	73
2022-09-20	low	55
2022-09-21	high	80
2022-09-21	low	57



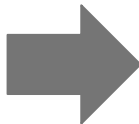
	is_high	high	low
input_x			
2022-09-19		77	58
2022-09-20		73	55
2022-09-21		80	57

Pivot tables in pandas:
`wide_df.shape` still returns (3,2)
but now includes extra headers

Fill in the code

Table name: *tall_df*

input_x	is_high	output_y
2022-09-19	high	77
2022-09-19	low	58
2022-09-20	high	73
2022-09-20	low	55
2022-09-21	high	80
2022-09-21	low	57



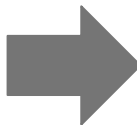
input_x	2022-09-19	2022-09-20	2022-09-21
is_high			
high	77	73	80
low	58	55	57

```
tall_df.pivot(  
    index=_____,  
    columns = _____,  
    values = _____)
```

Fill in the code

Table name: *tall_df*

input_x	is_high	output_y
2022-09-19	high	77
2022-09-19	low	58
2022-09-20	high	73
2022-09-20	low	55
2022-09-21	high	80
2022-09-21	low	57



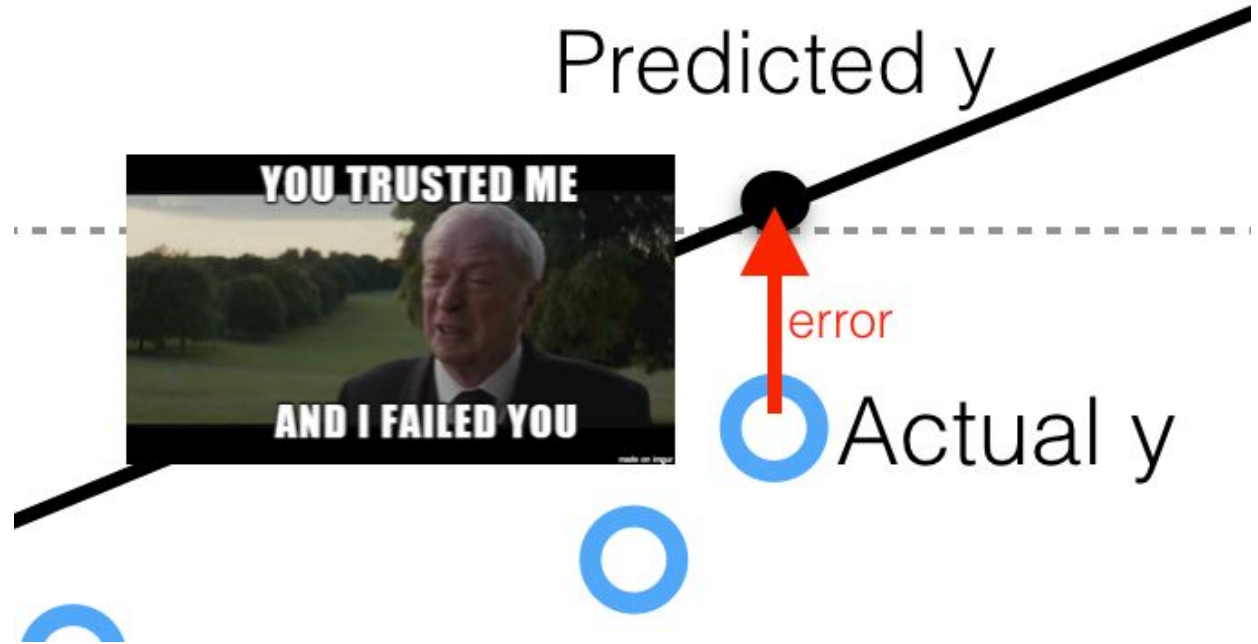
input_x	2022-09-19	2022-09-20	2022-09-21
is_high			
high	77	73	80
low	58	55	57

```
tall_df.pivot(  
    index='is_high',  
    columns = 'input_x',  
    values = 'output_y')
```

Reshaping takeaways

- Wide to long: **melt**
- Long to wide: **pivot**
- Search for pandas documentation when you need to use reshaping functions

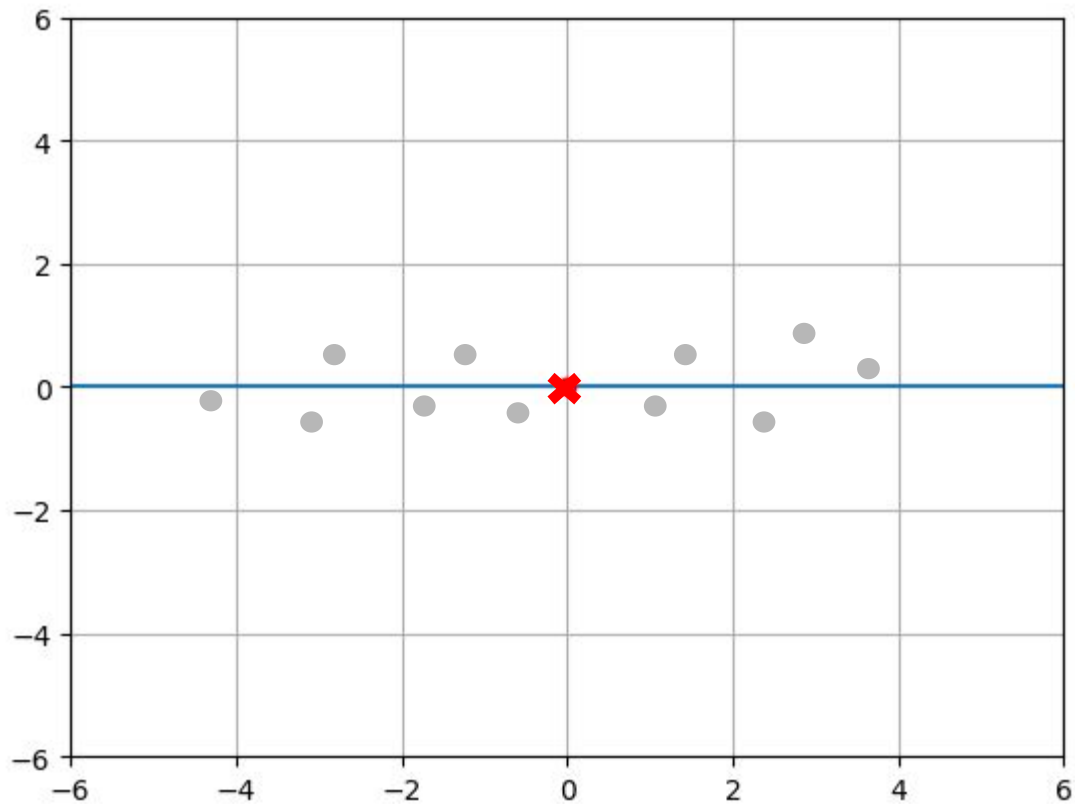
1 min break



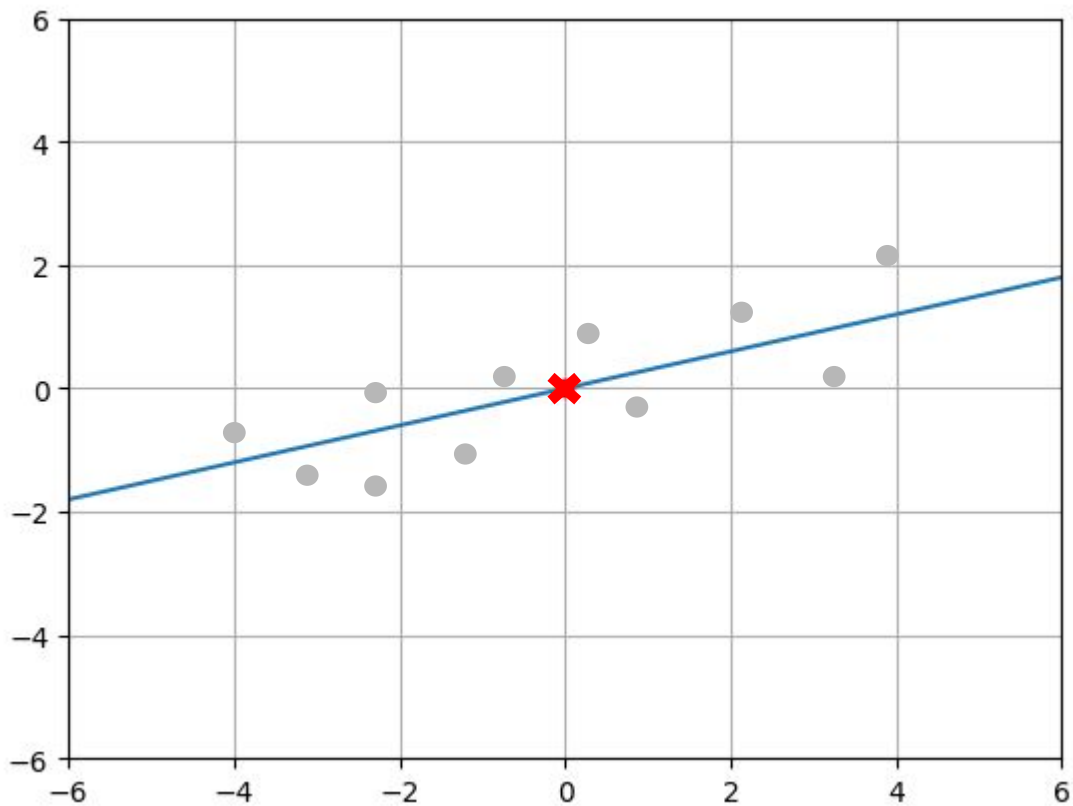
**Back to thinking about
regressions!**

**Let's develop a visual
intuition for different
regression coefficients
and values**

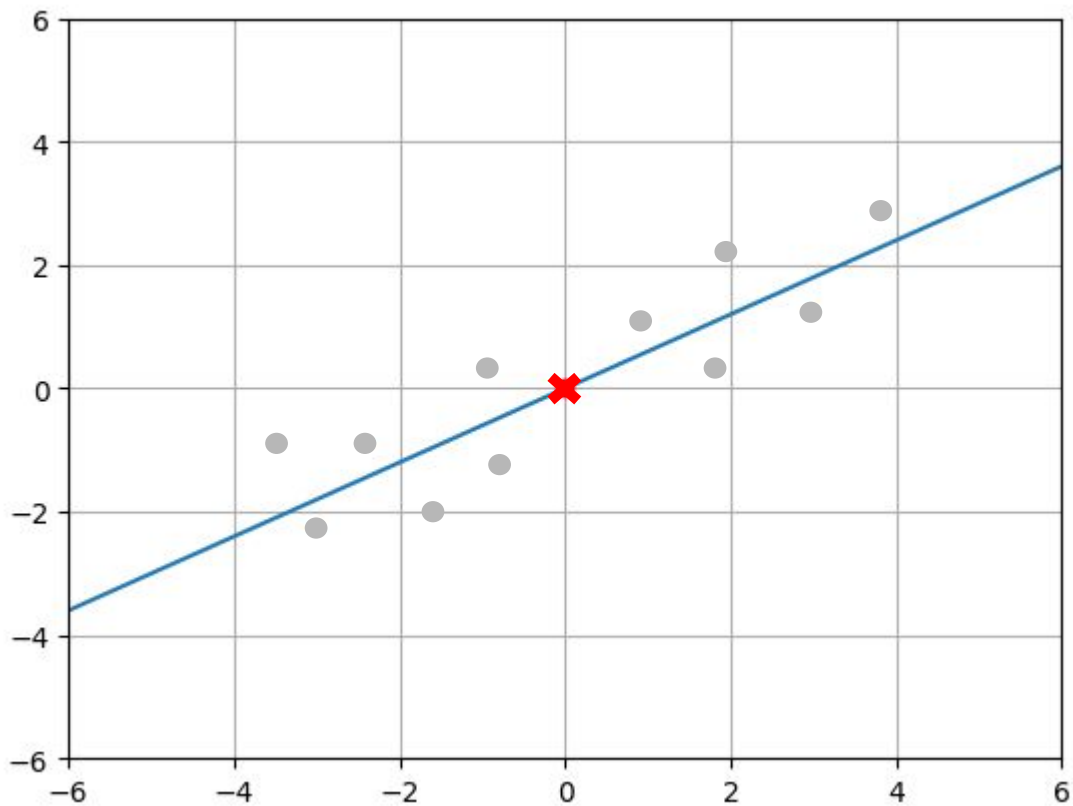
Slope β	0
Intercept α	0
X	0
Error ε	0



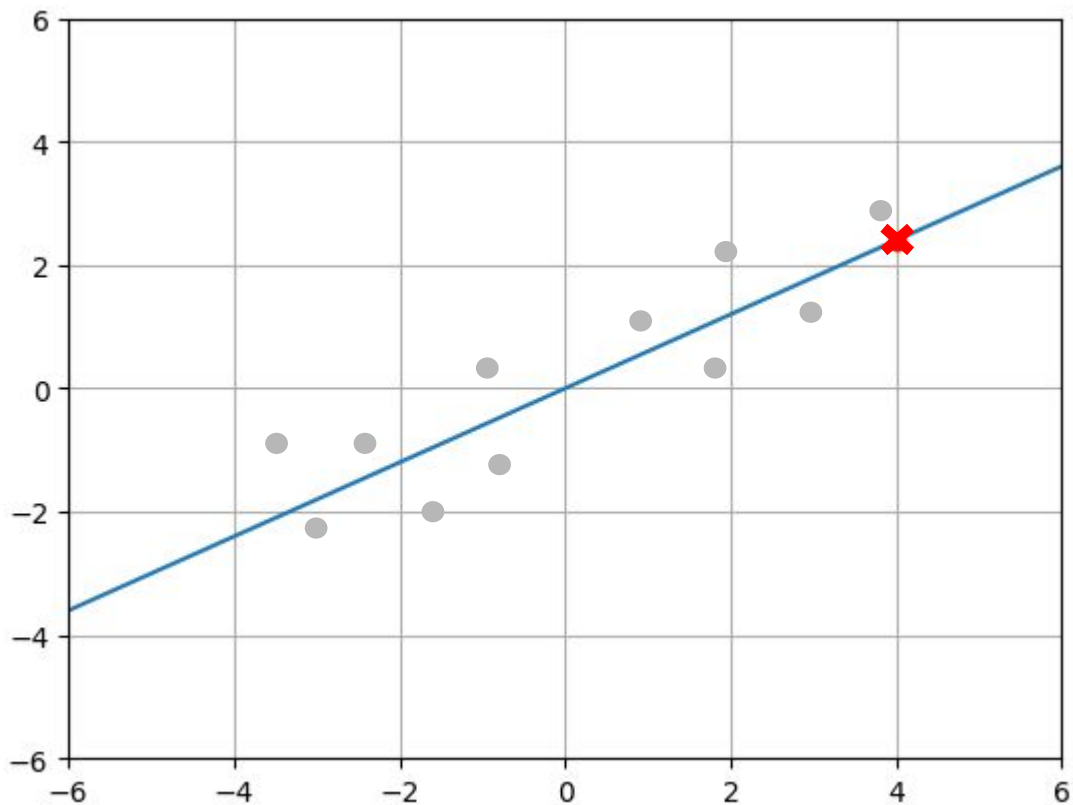
Slope β	0
Intercept α	0
X	0
Error ϵ	0



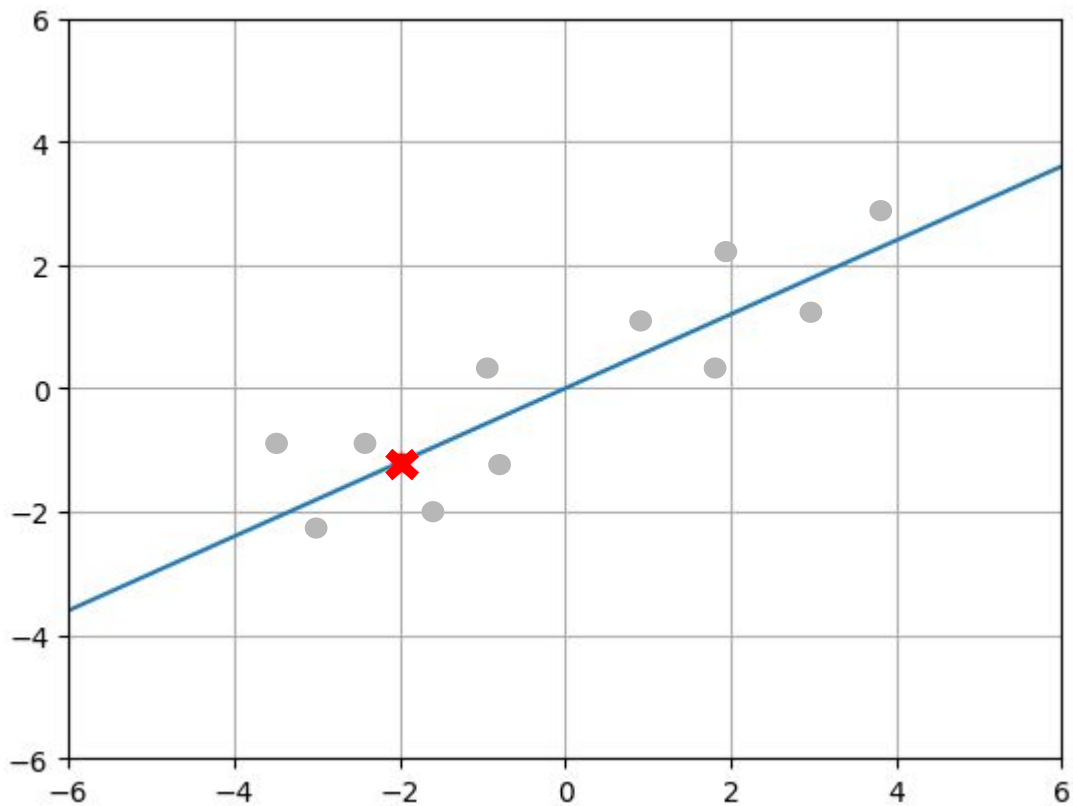
Slope β	0.30
Intercept α	0
X	0
Error ϵ	0



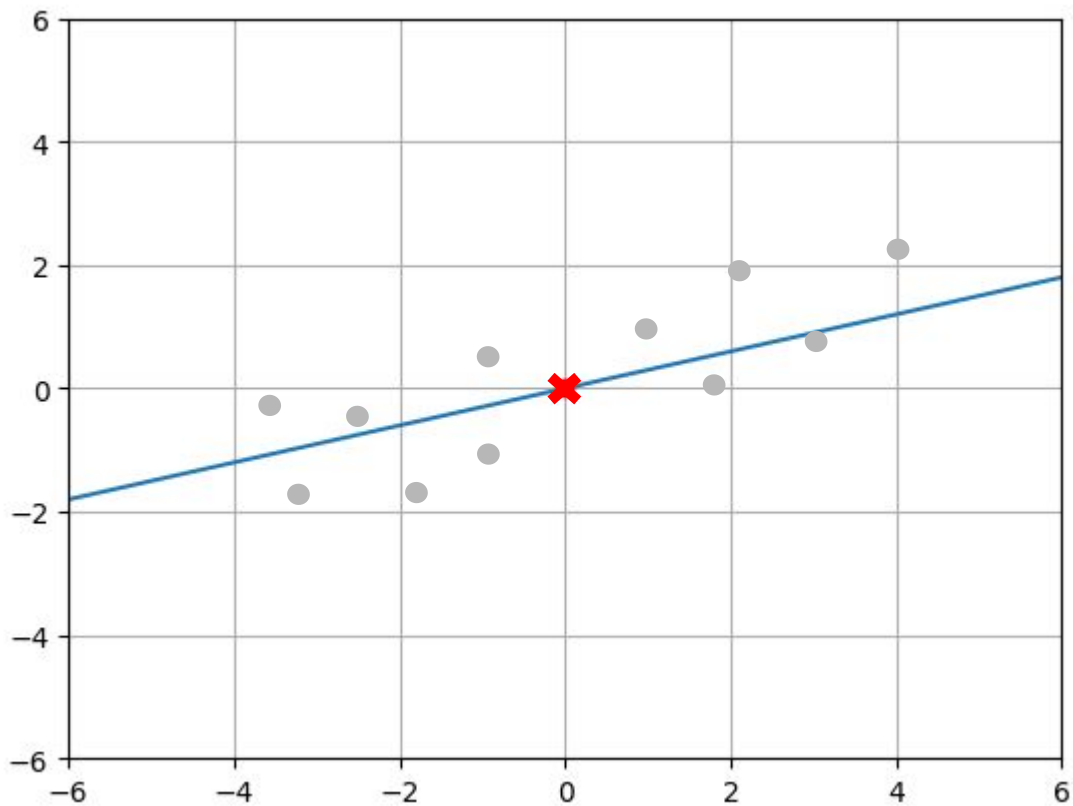
Slope β	0.60
Intercept α	0
X	0
Error ϵ	0



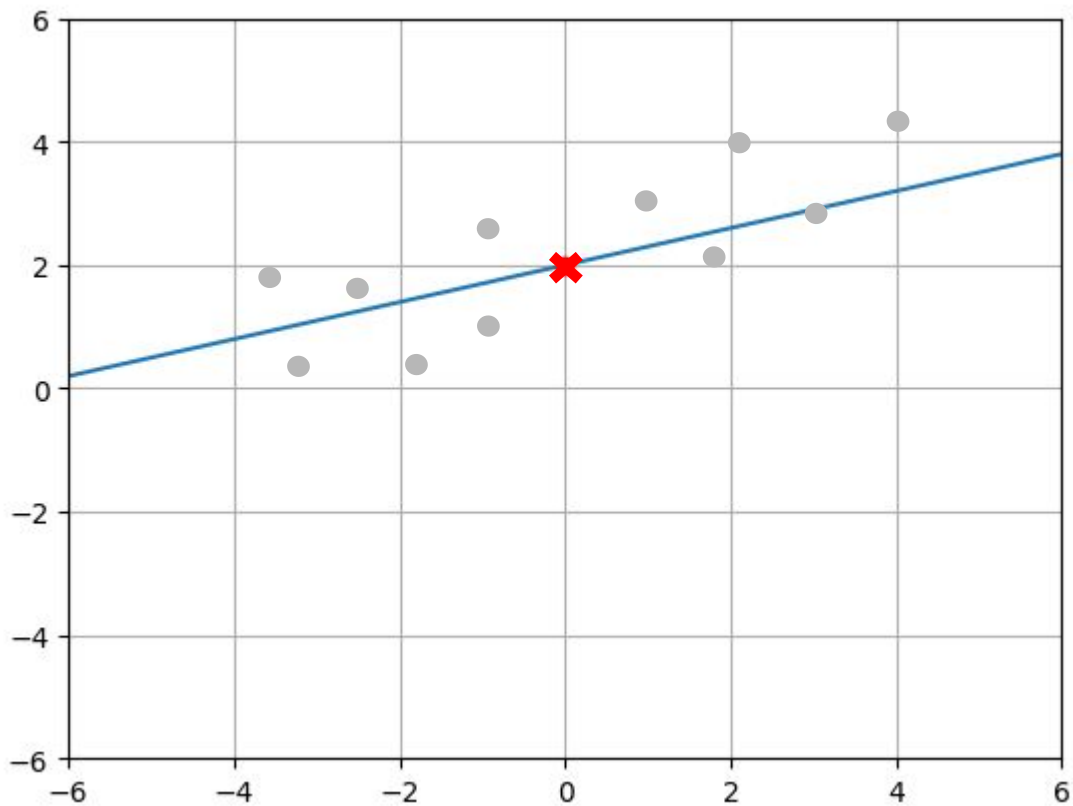
Slope β	0.60
Intercept α	0
X	4.0
Error ϵ	0



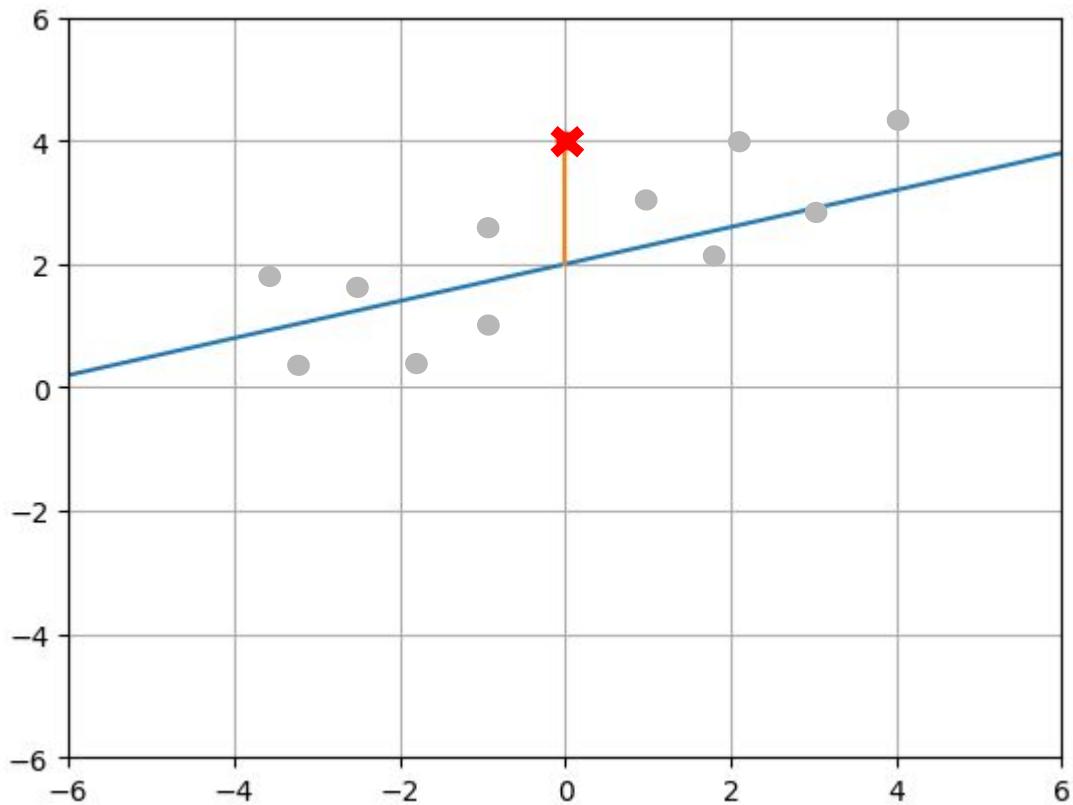
Slope β	0.60
Intercept α	0
X	-2.0
Error ϵ	0



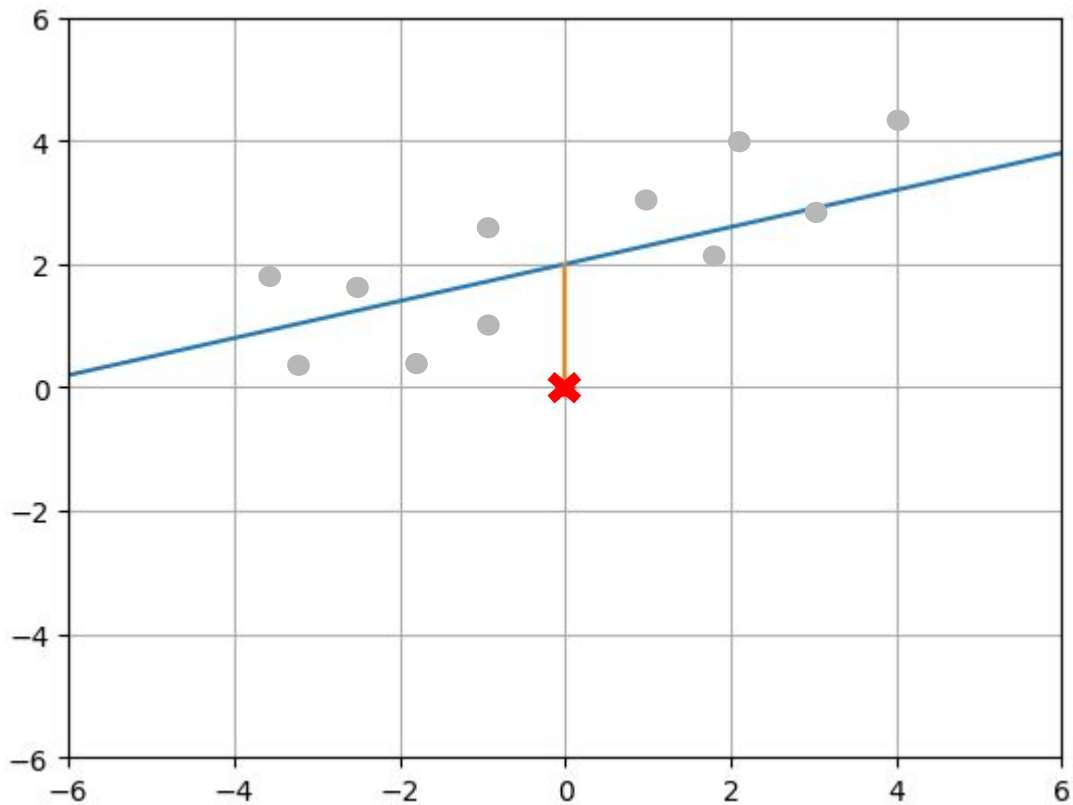
Slope β	0.30
Intercept α	0
X	0
Error ϵ	0



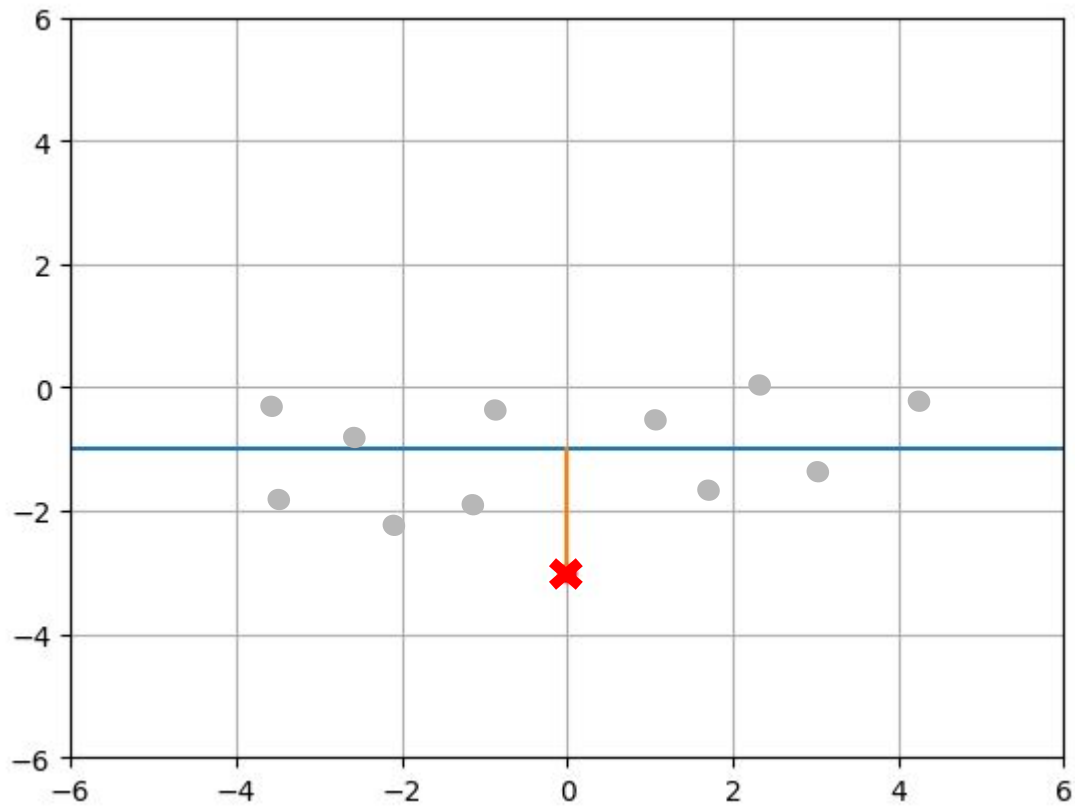
Slope β	0.30
Intercept α	2.0
X	0
Error ϵ	0



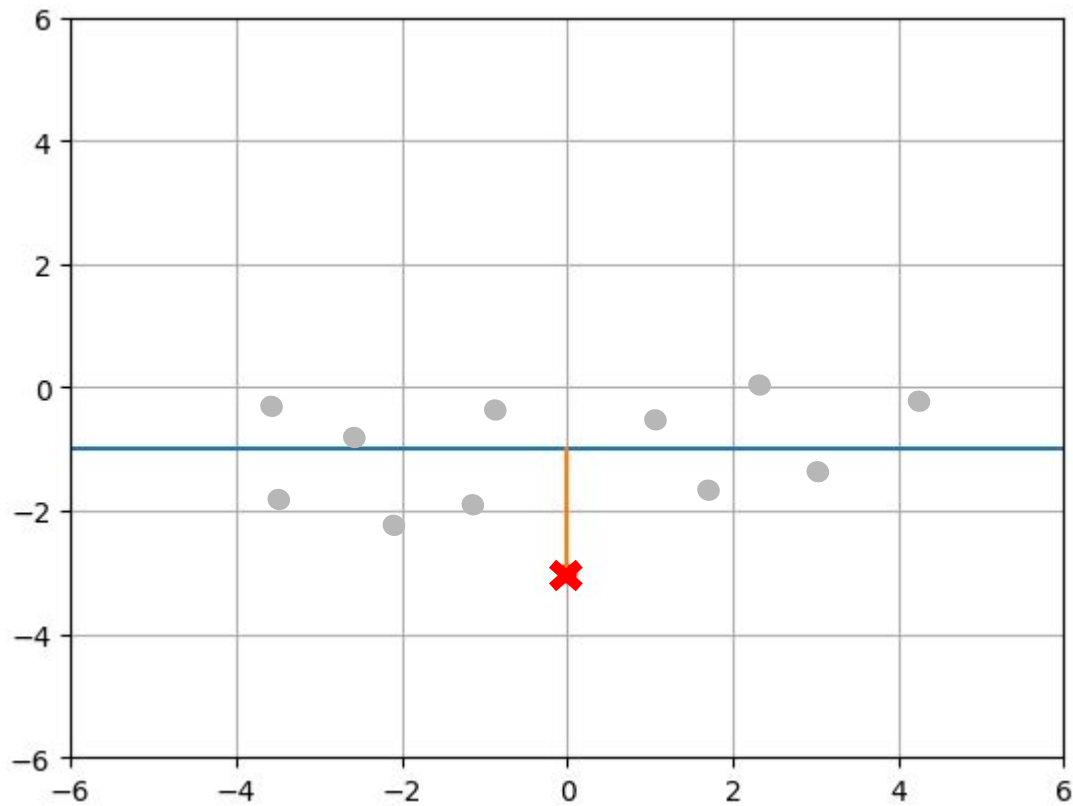
Slope β	0.30
Intercept α	2.0
X	0
Error ϵ	2.0



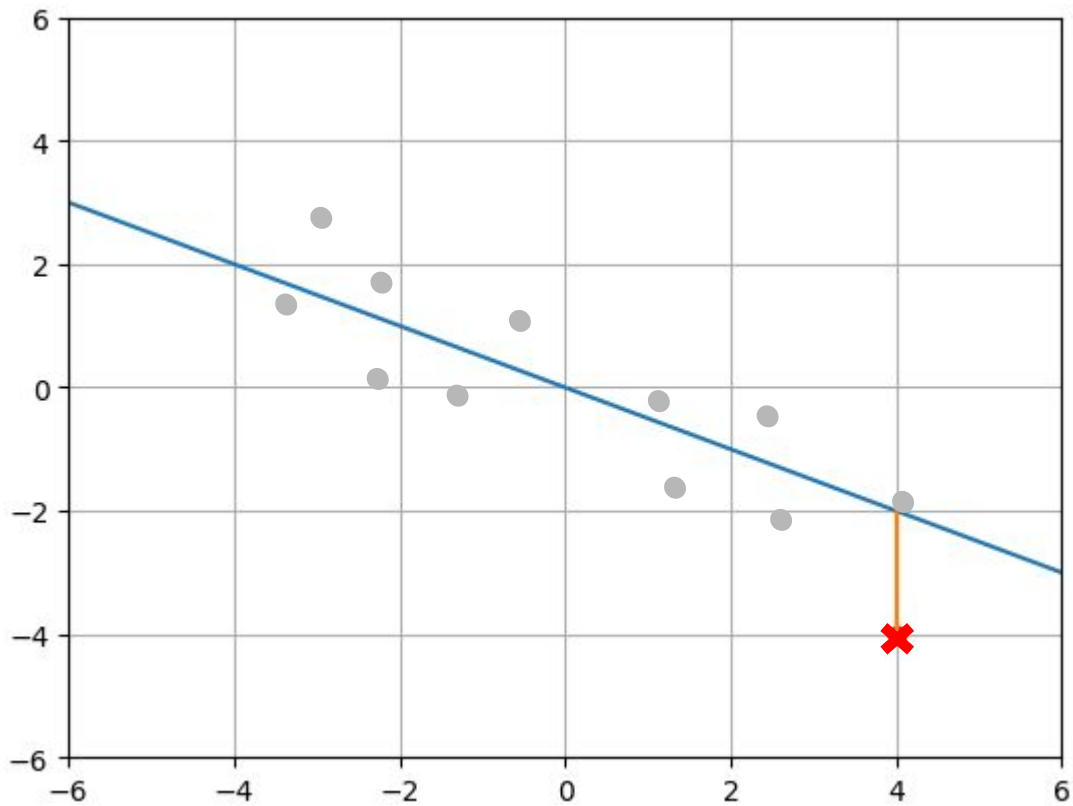
Slope β	0.30
Intercept α	2.0
X	0
Error ϵ	-2.0



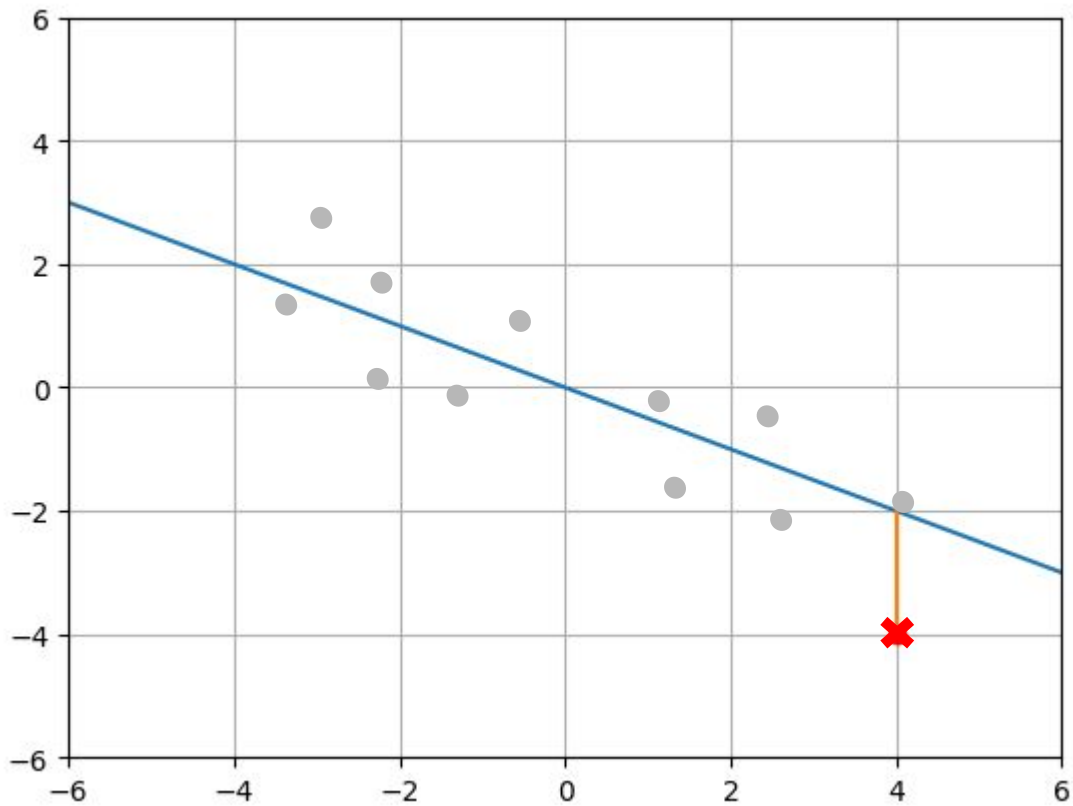
Slope β	?
Intercept α	?
X	?
Error ϵ	?



Slope β	0
Intercept α	-1
X	0
Error ϵ	-2



Slope β	?
Intercept α	?
X	?
Error ϵ	?

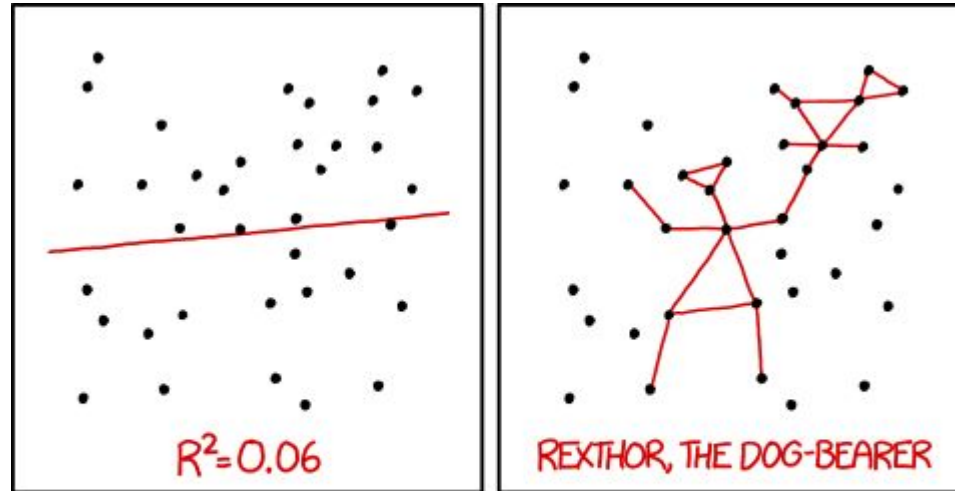


Slope β	-0.5
Intercept α	0
X	4.0
Error ϵ	-2.0

Regression recap

- We find intercept α and slope β by minimizing the sum of squared errors ε
 - We can do this using math, but Python has packages that will automatically give us the numbers α and β

1 min break



I DON'T TRUST LINEAR REGRESSIONS WHEN IT'S HARDER
TO GUESS THE DIRECTION OF THE CORRELATION FROM THE
SCATTER PLOT THAN TO FIND NEW CONSTELLATIONS ON IT.

Simple linear model: summarize relationships

- Model: $y = \alpha + \beta x$
- 1 unit increase in x corresponds to a β unit increase/decrease in y

Simple linear model: summarize relationships

- Model: $y = \alpha + \beta x$
- 1 unit increase in x corresponds to a β unit increase/decrease in y
- Today: regression transformations that have different relationships

What is a transformation?

- $x \rightarrow f(x)$
- Transformation functions are things like...
 - Squaring
 - Logarithms
 - Exponentiating
 - Adding/subtracting a constant
 - ...

What is a transformation?

- $x \rightarrow f(x)$
- Transformation functions are things like...
 - Squaring: if $x=5$, $f(x) = 25$
 - Logarithms: if $x=5$, $f(x) = [\text{unclear, what base are we using?}]$
 - Exponentiating
 - Adding/subtracting a constant
 - ...

Can we use transformations with regressions?

- Original model: $y = \alpha + \beta x$
- Transformed model:
 - $f(y) = \alpha + \beta x$
 - $y = \alpha + \beta * f(x)$
 - $f(y) = \alpha + \beta * f(x)$

Can we use transformations with regressions?

- Original model: $y = \alpha + \beta x$
- Transformed model:
 - $f(y) = \alpha + \beta x$
 - $y = \alpha + \beta * f(x)$
 - $f(y) = \alpha + \beta * f(x)$
- If $f()$ is not linear (e.g. logarithmic, quadratic), are these still linear regressions?

Can we use transformations with regressions?

- If $f()$ is not linear (e.g. logarithmic, quadratic), are these still linear regressions? YES!

Can we use transformations with regressions?

- If $f()$ is not linear (e.g. logarithmic, quadratic), are these still linear regressions? YES!
- Pretend like you are Python. If I tell you to run the regression $Z \sim X$, is this a linear regression?

X		Z
1		25
2		100
3		64
...		...

Can we use transformations with regressions?

- If $f()$ is not linear (e.g. logarithmic, quadratic), are these still linear regressions? YES!
- Pretend like you are Python. If I tell you to run the regression $Z \sim X$, is this a linear regression?

X		Z
1		25
2		100
3		64
...		...



Can we use transformations with regressions?

X	Y	Z
1	-5	25
2	10	100
3	8	64
...

- If $f()$ is not linear (e.g. logarithmic, quadratic), are these still linear regressions? YES!
- Pretend like you are Python. If I tell you to run the regression $Z \sim X$, is this a linear regression?
- Yes! You'd have no idea that Z was secretly Y^2 !

Can we use transformations with regressions?

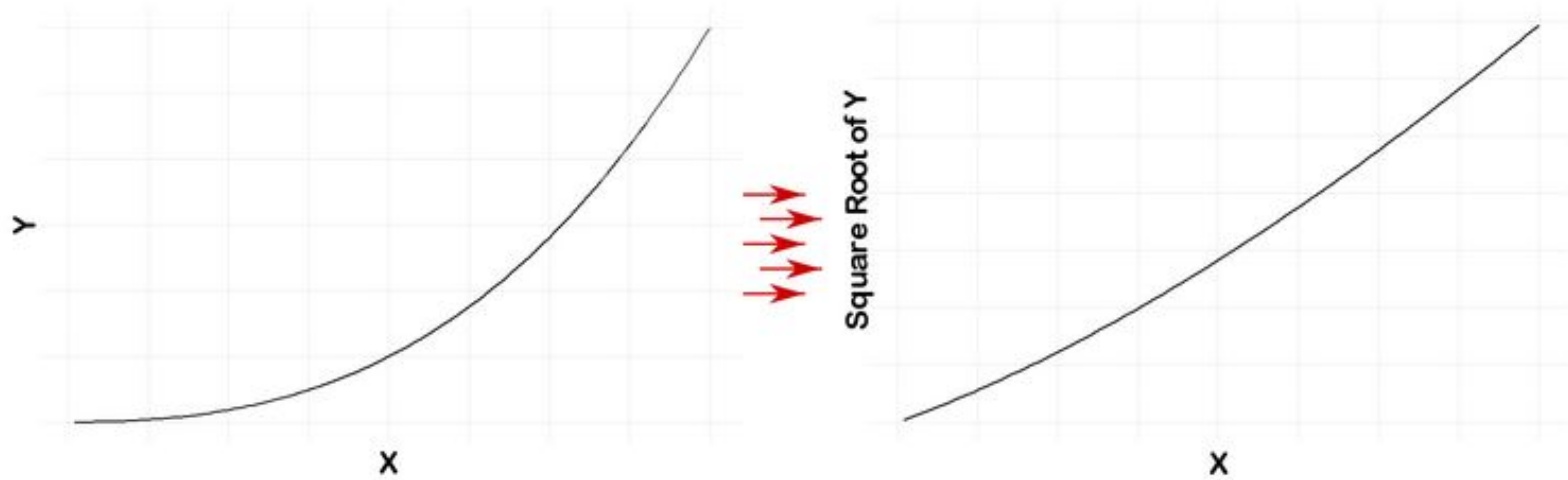
X	Y	Z
1	-5	25
2	10	100
3	8	64
...

- If $f()$ is not linear (e.g. logarithmic, quadratic), are these still linear regressions? YES!
 - Pretend like you are Python. If I tell you to run the regression $Z \sim X$, is this a linear regression?
 - Yes! You'd have no idea that Z was secretly Y^2 !
 - It doesn't matter whether there's a (nonlinear) relationship among variables: from the perspective of the regression, they're just numbers.
-

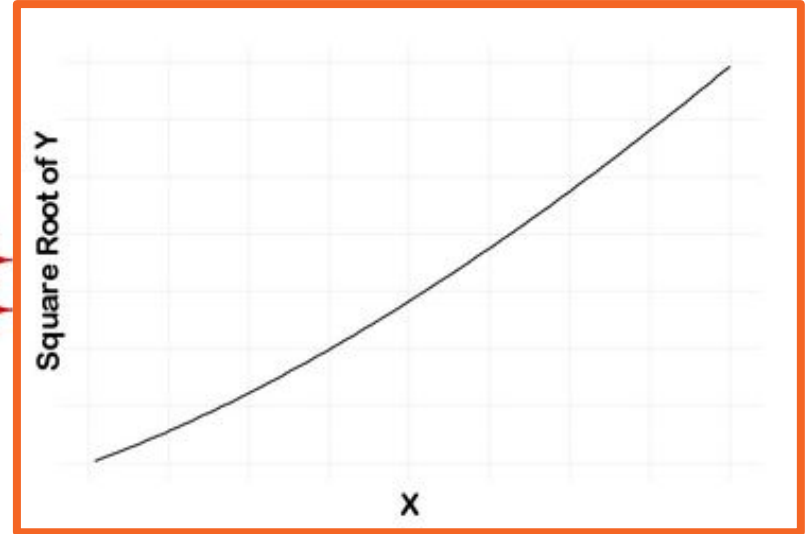
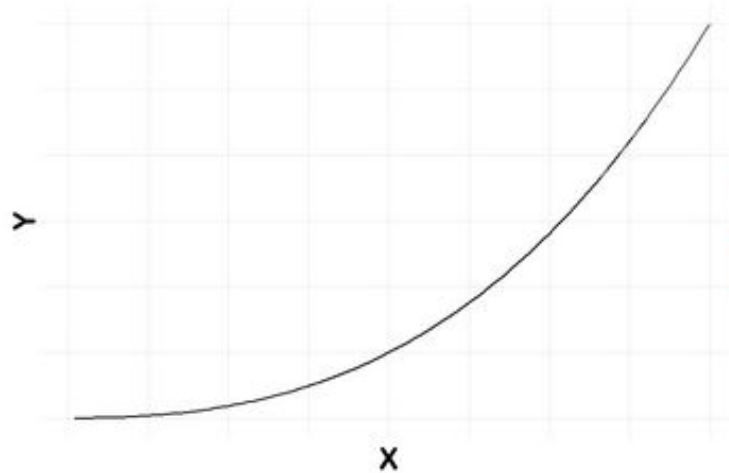
Why would we use transformations in regressions?

- Transformations (including non-linear ones) are used to **achieve greater linearity** in a linear regression model
- Transformations can allow for easier interpretability of your model and data

Which of these has better linear fit?



Which of these has better linear fit?



After the transformation, the relationship looks linear enough to run a linear regression

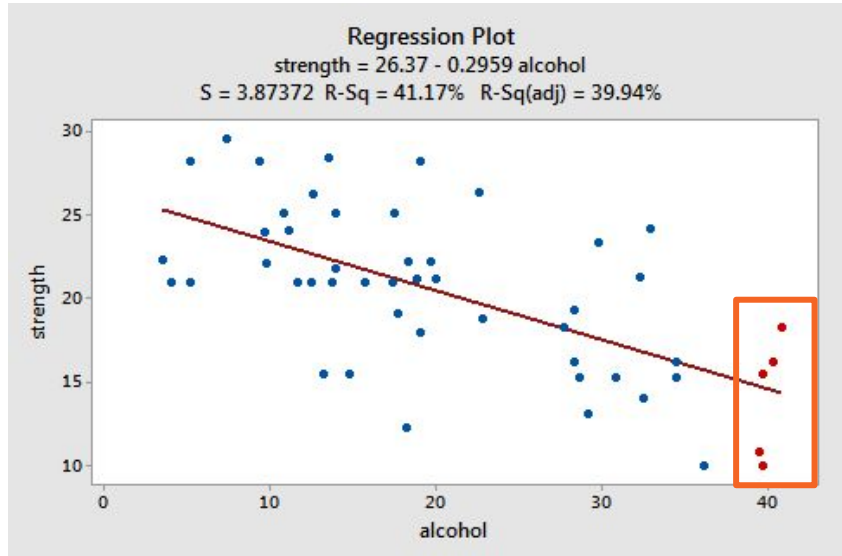
How do you know if you need to transform your data?

1. Calculate your predicted errors ϵ_i (residuals) from model $y = \alpha + \beta x$

How do you know if you need to transform your data?

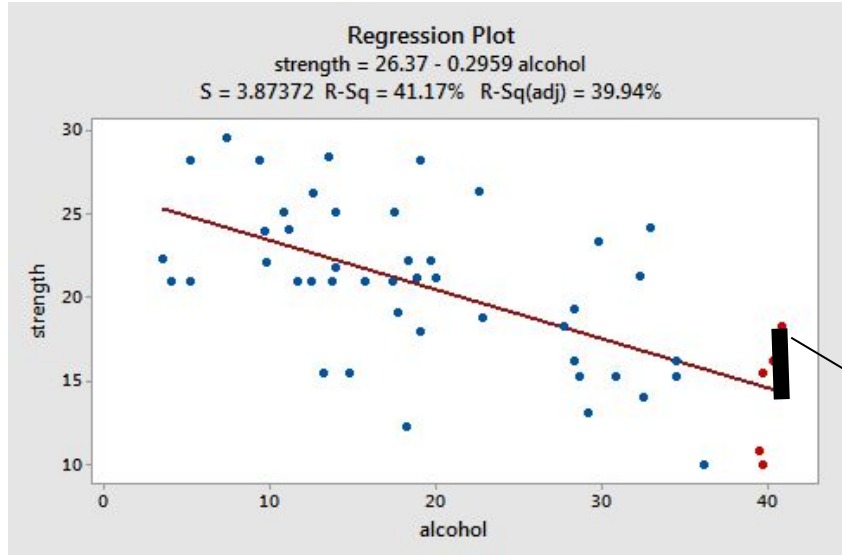
1. Calculate your predicted errors ϵ_i (residuals) from model $y = \alpha + \beta x$
2. Plot the residuals on y-axis against estimated y-hats on x-axis:
 - a. If they seem random, the data are ~linear
 - b. If not random, data are non-linear and need to be transformed

Plot → Residual Plot



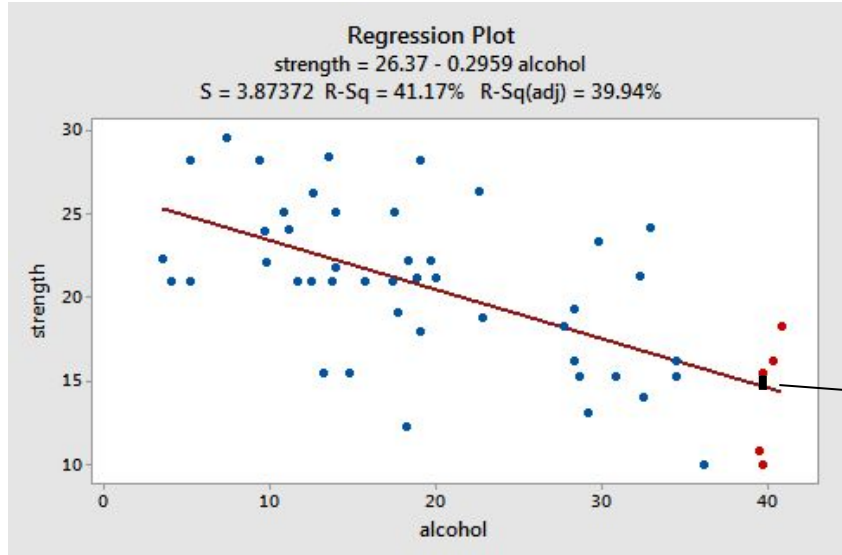
Let's look at the residuals (difference between data points and regression line)

Plot → Residual Plot



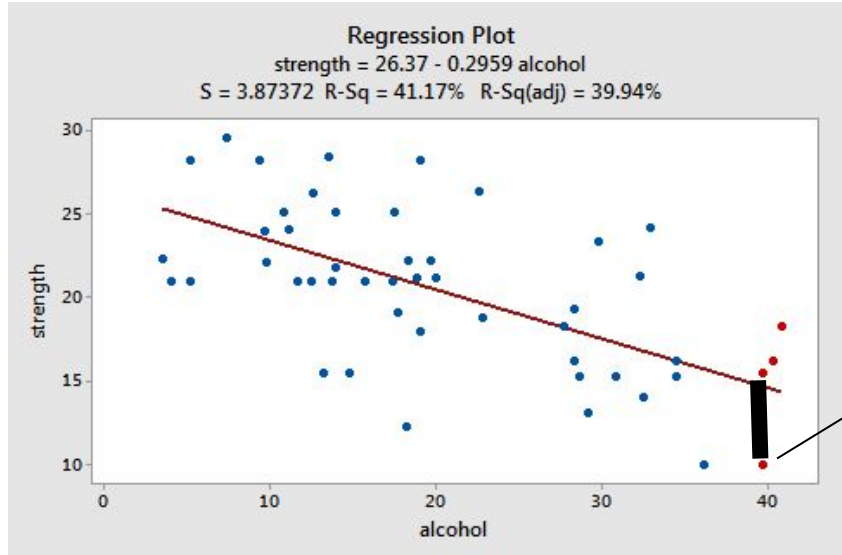
Residual ~ 5 “strength”

Plot → Residual Plot



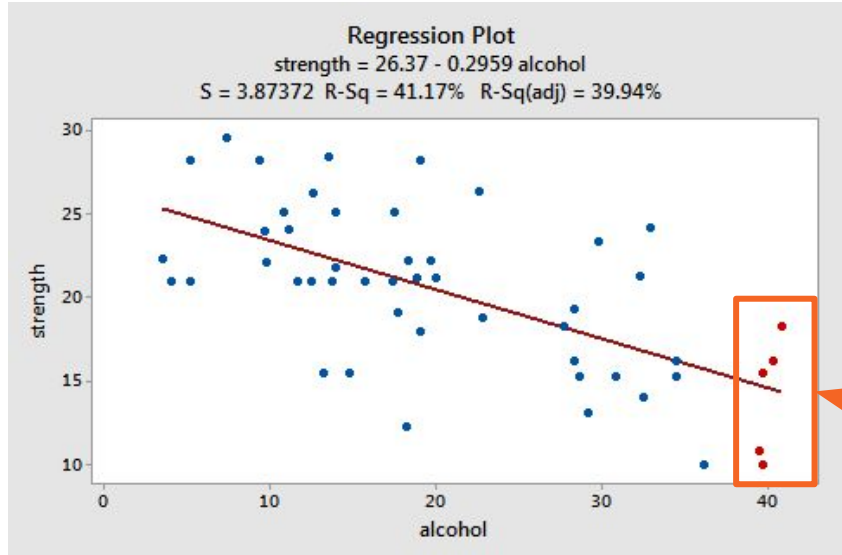
Residual ~ 1 “strength”

Plot → Residual Plot



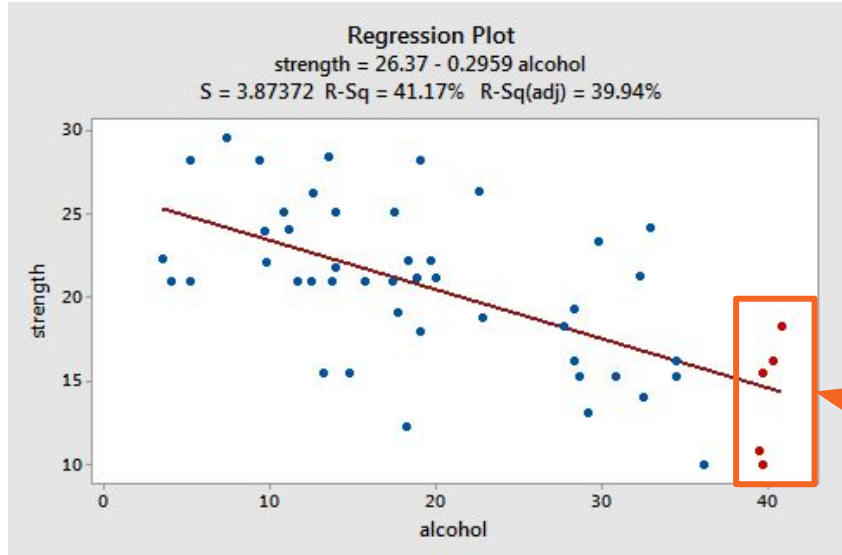
Residual ~ -5 “strength”

Plot → Residual Plot



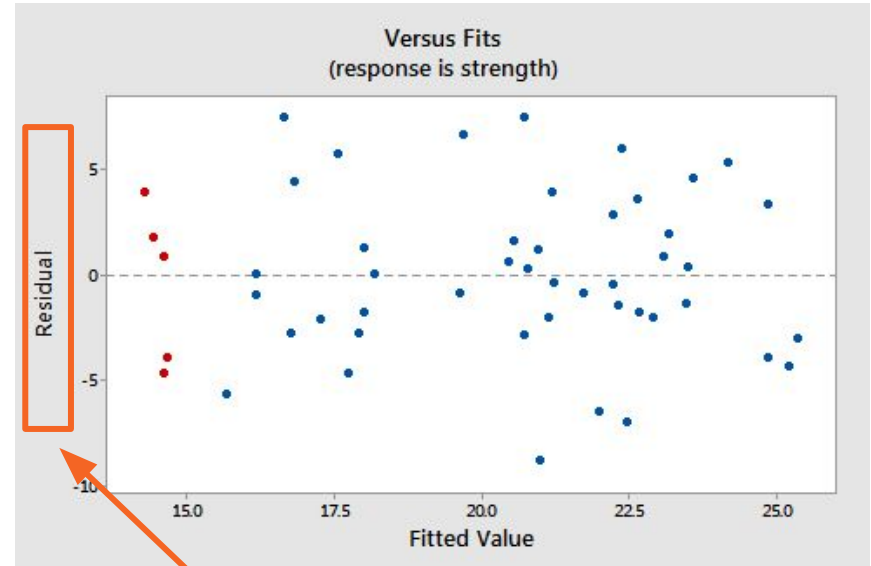
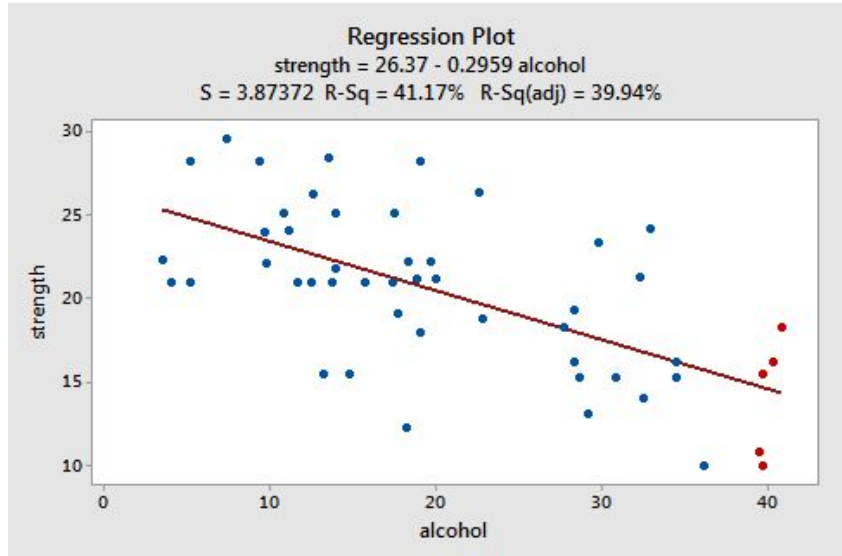
How many of the red dots will have positive residual?
How many negative?

Plot → Residual Plot



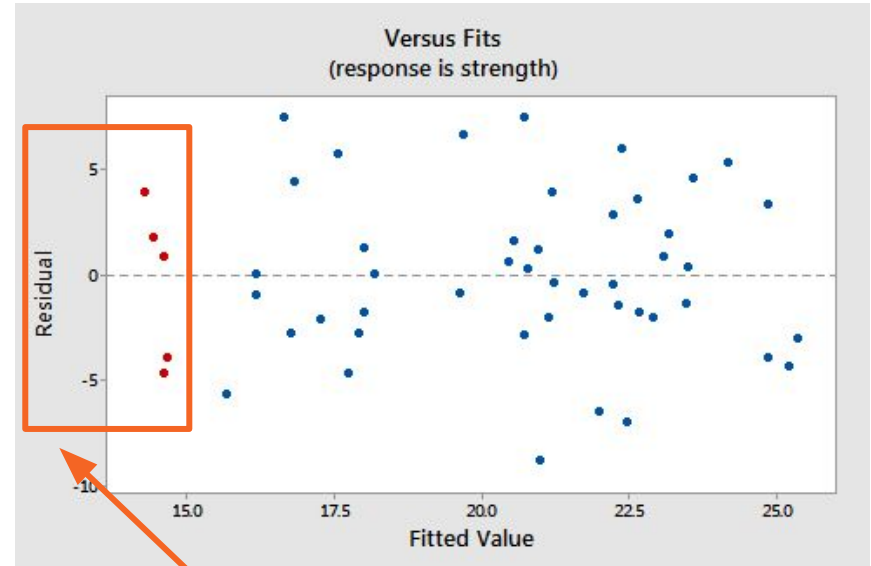
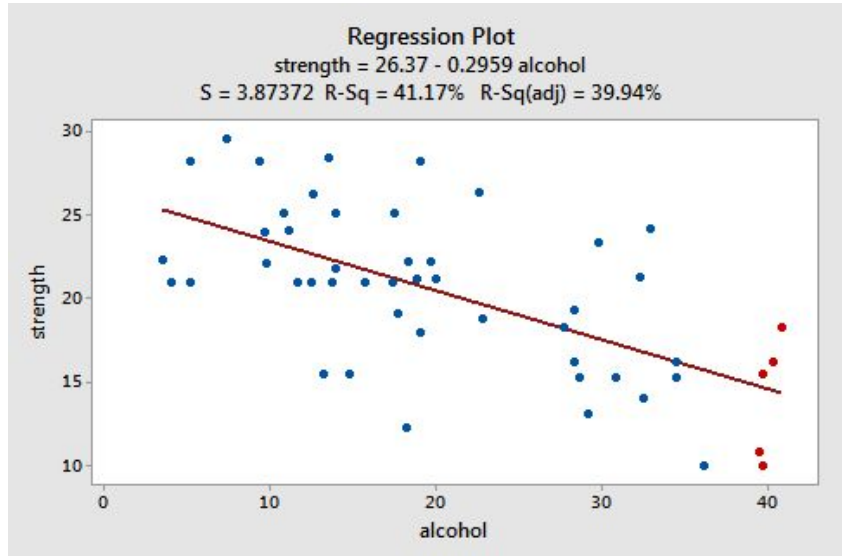
3 red dots will have positive residual,
2 will have negative residual

Plot → Residual Plot



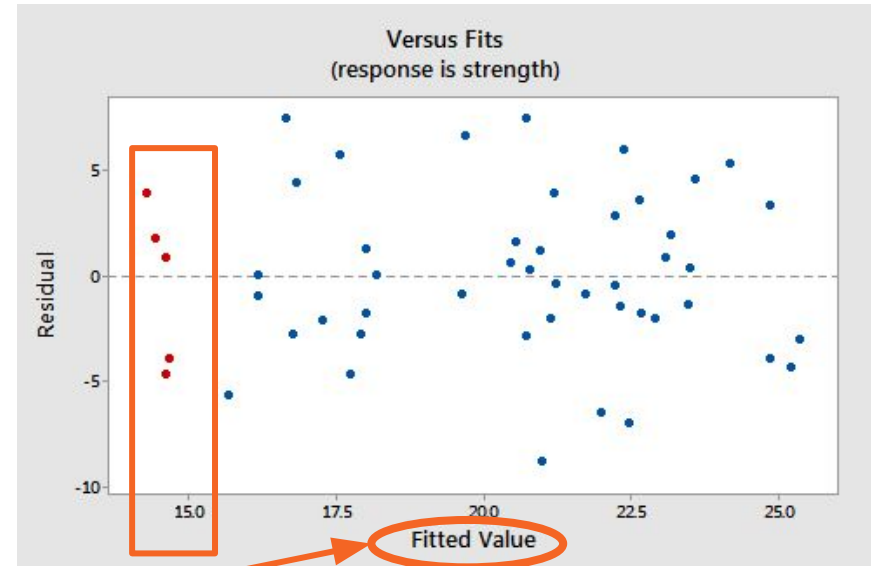
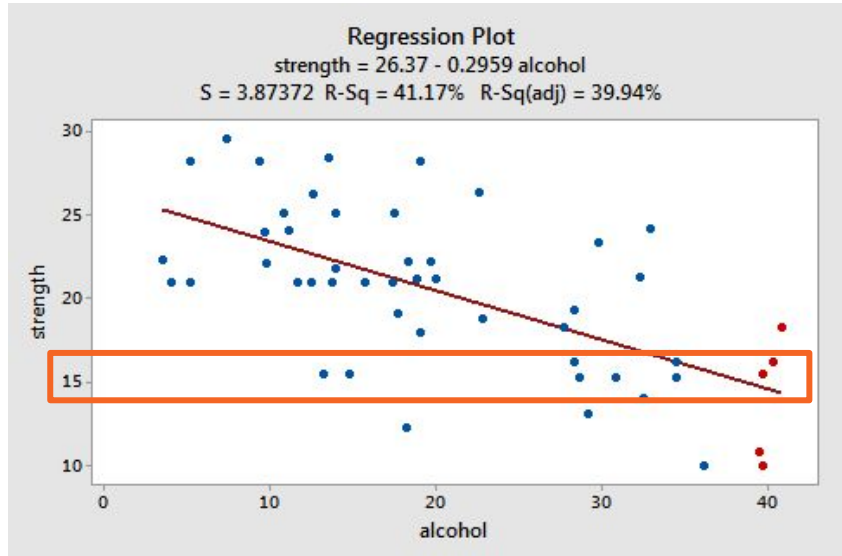
Residual plot y-axis shows the residuals of the original plot

Plot → Residual Plot



Residual plot y-axis shows the residuals of the original plot (red dots: 3 positive, 2 negative)

Plot → Residual Plot



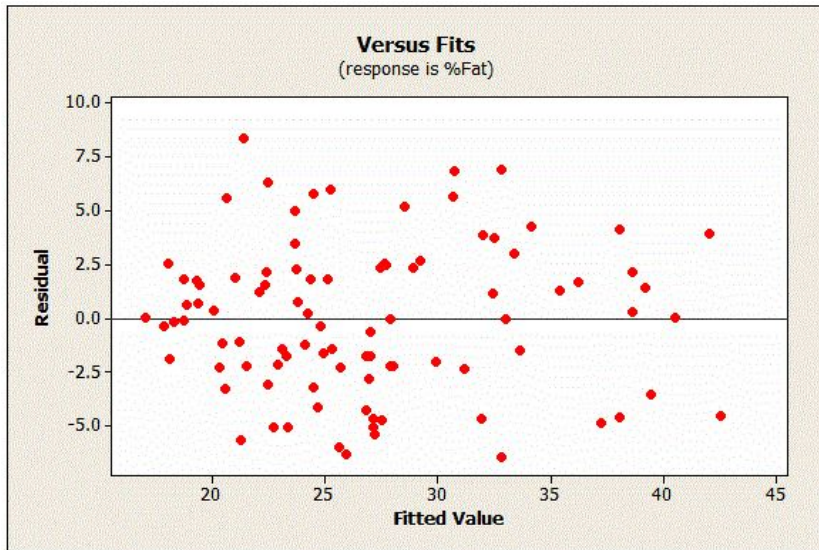
Residual plot x-axis shows \hat{y} (predicted “strength” if you trace the red dots to the regression line)

When to use transformations with regressions?

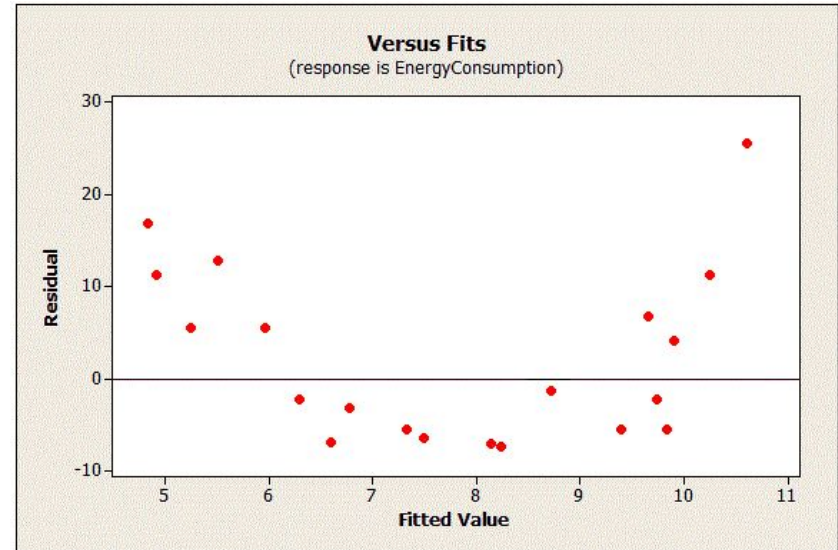
1. Calculate your predicted errors ϵ_i (residuals) from model $y = \alpha + \beta x$
2. Plot the residuals on y-axis against estimated y-hats on x-axis:
 - a. If they seem random, the data are ~linear
 - b. If not random, data are non-linear and need to be transformed

Residual Plots: good vs bad

Random Residuals ✓



Non-Random Residuals ✗




Why does residual randomness have to do with linearity?

Recall we have $y_i = \alpha + \beta x_i + \varepsilon_i$

Deterministic (no
randomness – just
plug into a function)

Why does residual randomness have to do with linearity?

Recall we have $y_i = \alpha + \beta x_i + \varepsilon_i$



Deterministic (no randomness – just plug into a function)

Stochastic: random and unpredictable (if it were predictable, it would've gone into the deterministic part of the model!)

Why does residual randomness have to do with linearity?

Recall we have $y_i = \alpha + \beta x_i + \varepsilon_i$

Residual plots
check if ε_i is
truly random



Stochastic: random
and unpredictable (if
it were predictable, it
would've gone into
the deterministic
part of the model!)

Why might a residual plot not be random?

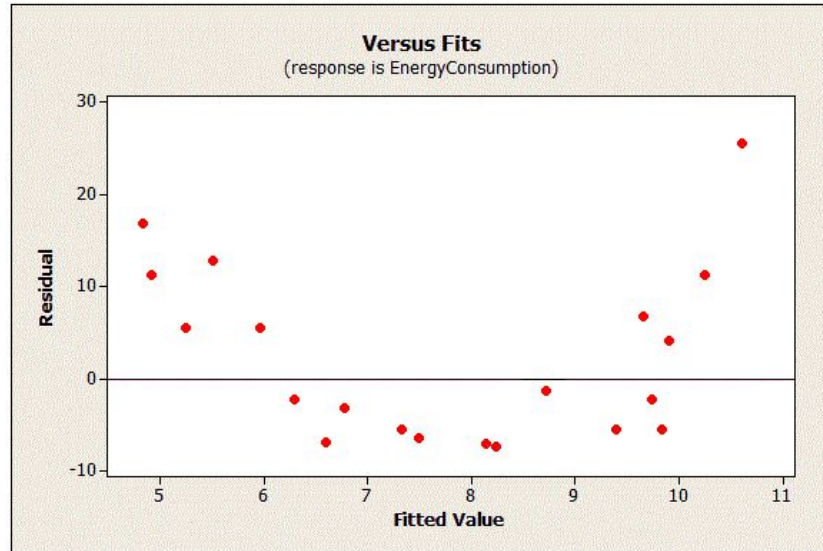
- The deterministic part of the model ($\alpha + \beta x$) is not capturing all of the information hidden in your data, which is leaking into your residual

Why might a residual plot not be random?

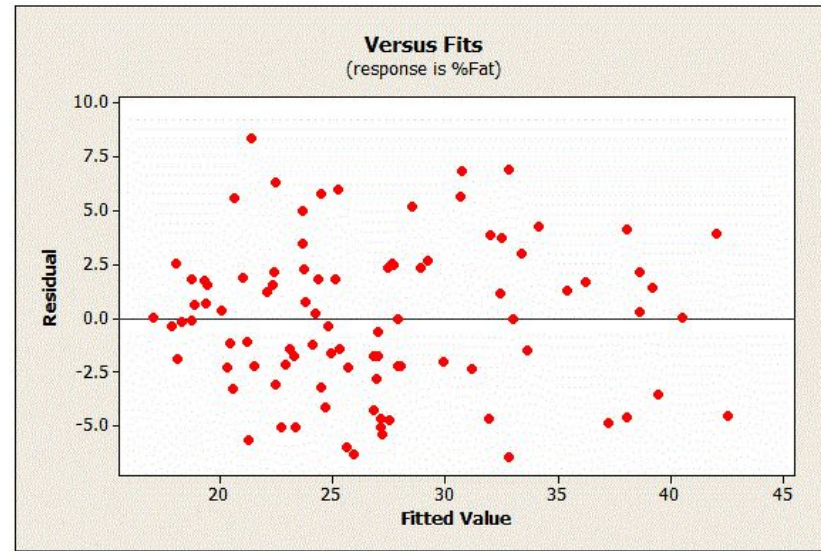
- The deterministic part of the model ($\alpha + \beta x$) is not capturing all of the information hidden in your data, which is leaking into your residual
 - Weird curvature – missing transformations
 - (for multivariable regressions) missing variables

Which of these residual plots indicates that the data need to be transformed?

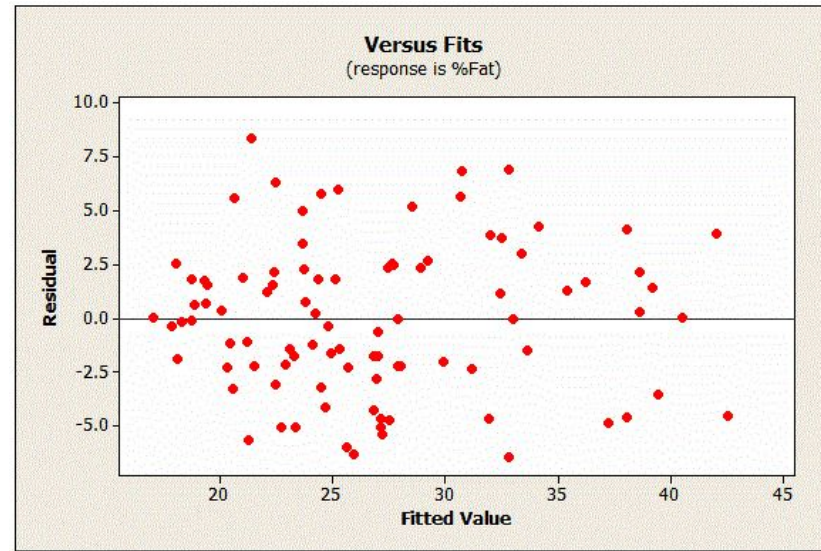
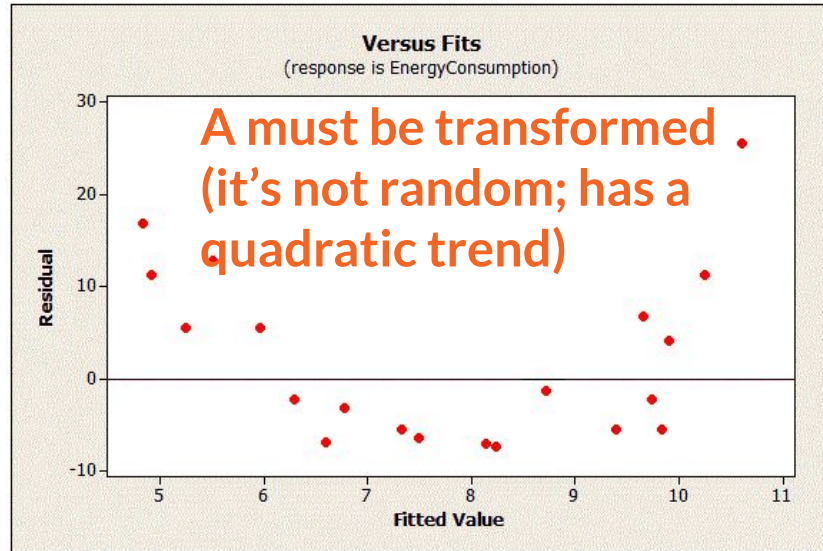
A



B



Which of these residual plots indicates that the data need to be transformed?



Caution: Heteroskedasticity

- “Heteroskedasticity” is when the **variance** of the residuals are unequal



**Variance is
not constant**



Heteroskedasticity



imgflip.com

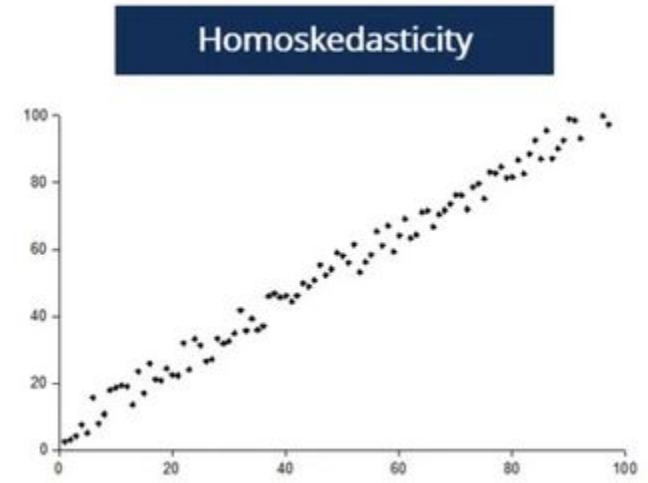
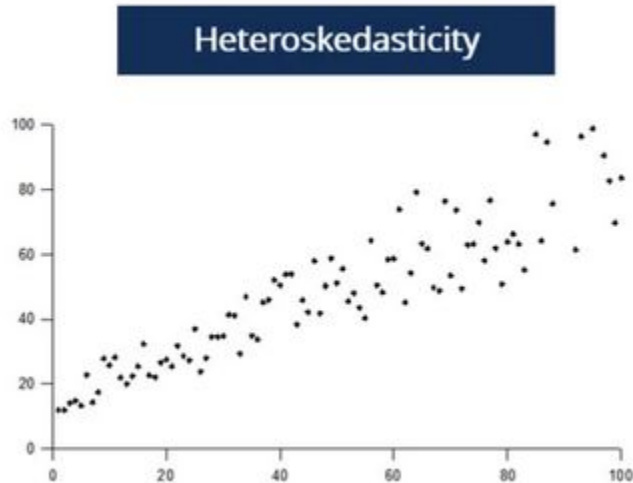
<https://twitter.com/vanikagrover/status/1374872137106870277>

Caution: Heteroskedasticity

- “Heteroskedasticity” is when the **variance** of the residuals are unequal
- This is a problem: OLS regressions have a baked-in “constant variance assumption” for residuals
 - **We shouldn't use OLS regressions if we clearly don't have constant variance**

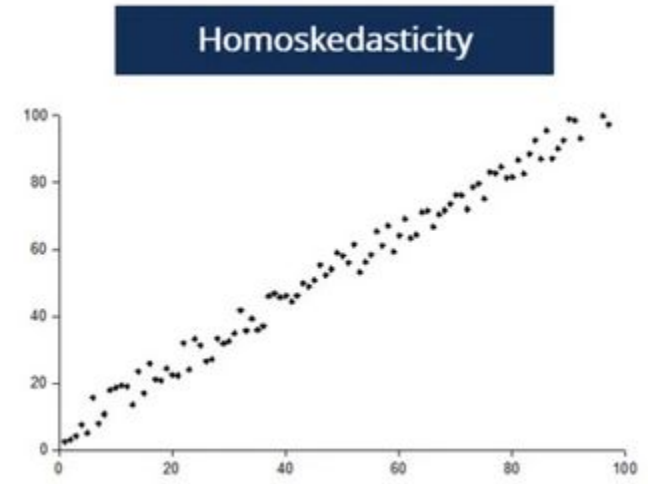
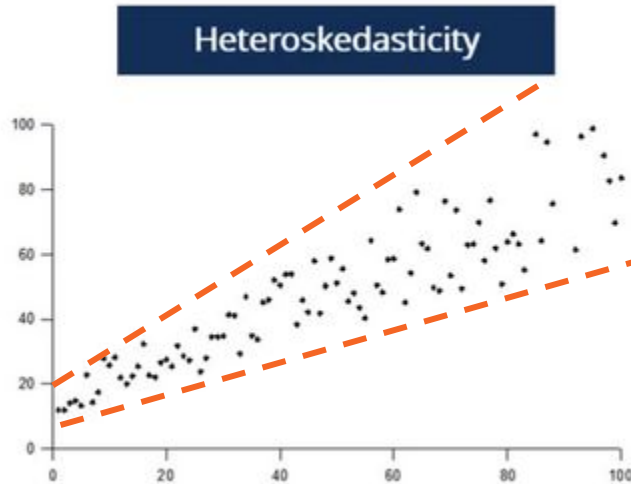
Caution: Heteroskedasticity

Residual plots



Caution: Heteroskedasticity

The classic tell:
cone/fan shape
in residual plot



Heteroskedasticity

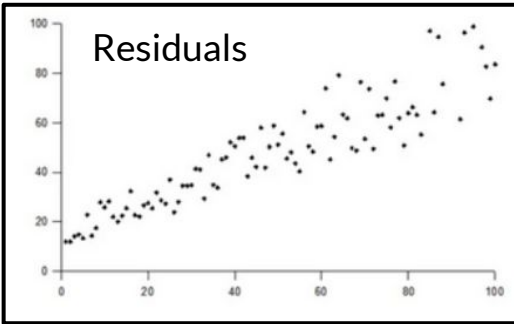
Homoskedasticity



The classic tell:
cone/fan shape
in residual plot

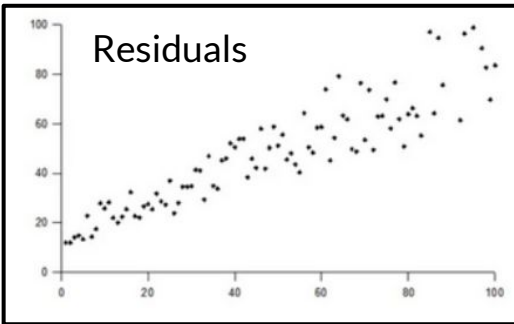
When does heteroskedasticity occur?

- It depends on the data. Some common examples:
 - a. In time series: values changing in the same way over time (**X = 1970-2023, Y = mobile phone sales**)

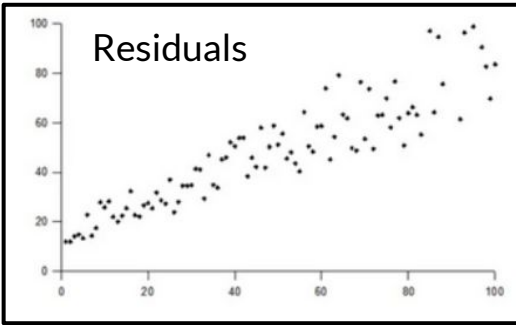


When does heteroskedasticity occur?

- It depends on the data. Some common examples:
 - a. In time series: values changing in the same way over time (**$X = 1970-2023$, $Y = \text{mobile phone sales}$**)
 - b. Massive range in values (\$1 vs. \$100000000) (**$X = \text{income}$, $Y = \text{food expenditures}$**)



When does heteroskedasticity occur?



- It depends on the data. Some common examples:
 - a. In time series: values changing in the same way over time (**X = 1970-2023, Y = mobile phone sales**)
 - b. Massive range in values (\$1 vs. \$100000000) (**X=income, Y = food expenditures**)
- In these examples: when X is low, everyone has about the same Y value (so variance is around 0). When X is high, there are more possible Y values (so variance is higher)

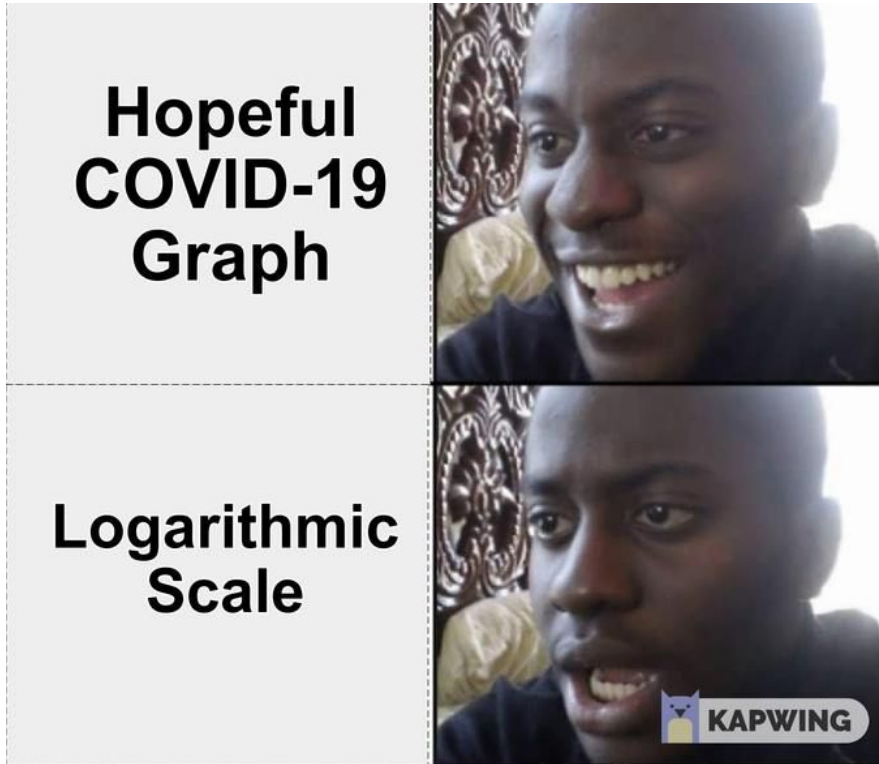
What do you do if you encounter heteroskedastic data?

- **Use transforms!** Specifically:
 - Log transform (← *most common*)
 - Square root transform
 - Cube root transform

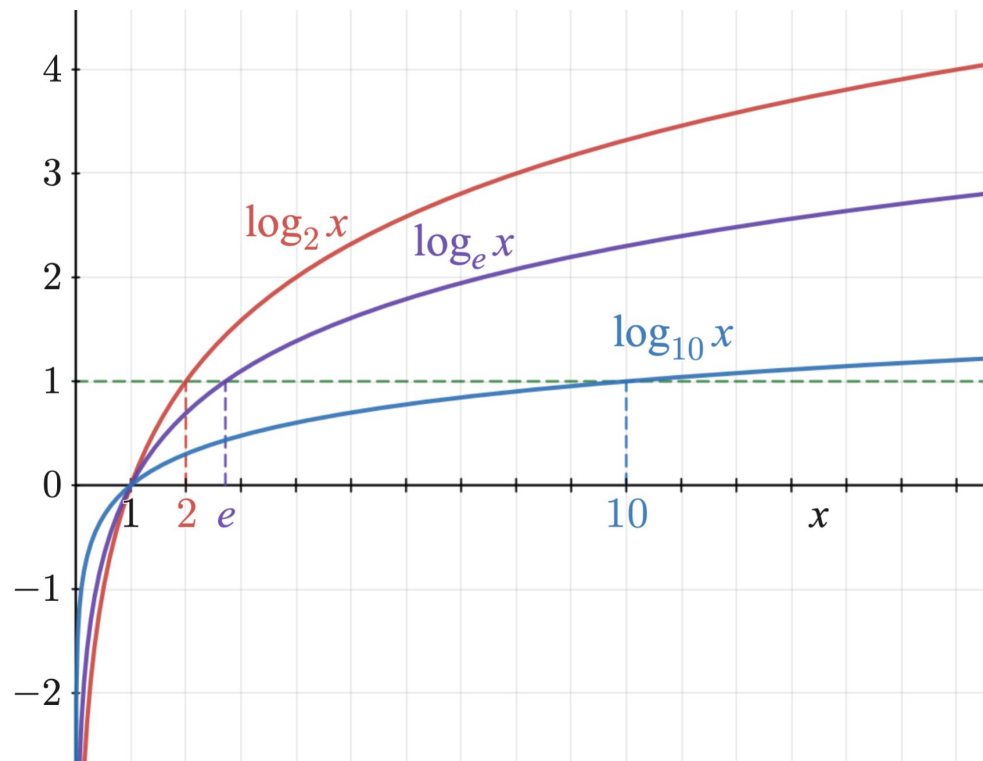
What do you do if you encounter heteroskedastic data?

- **Use transforms!** Specifically:
 - Log transform (← *most common*)
 - Square root transform
 - Cube root transform
- Why would a log transform help with heteroskedasticity?

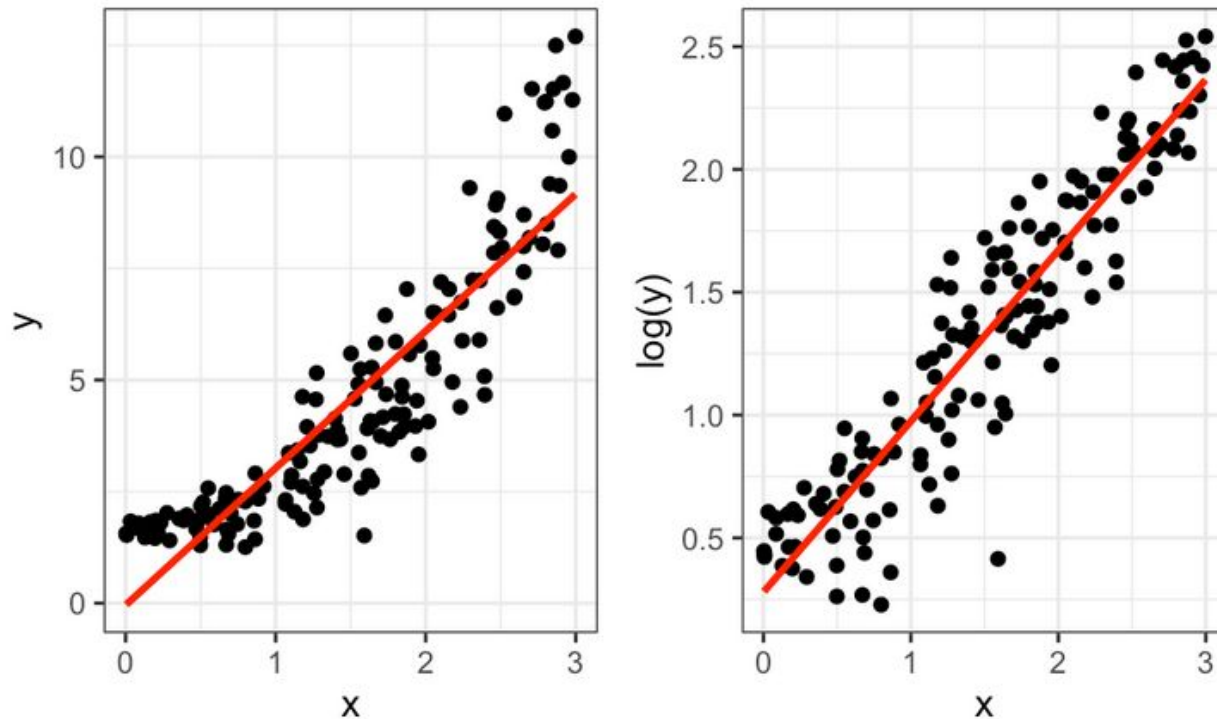
Logarithms smoosh big numbers



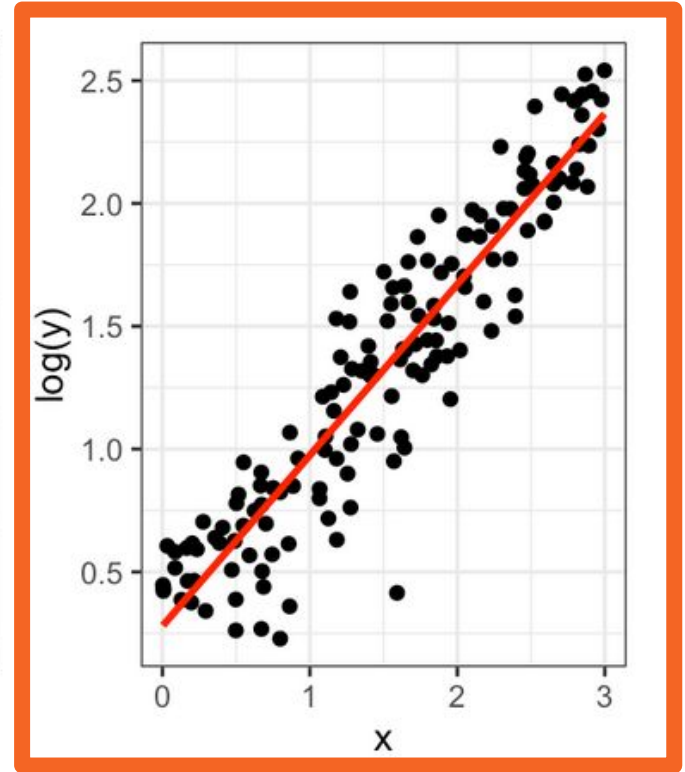
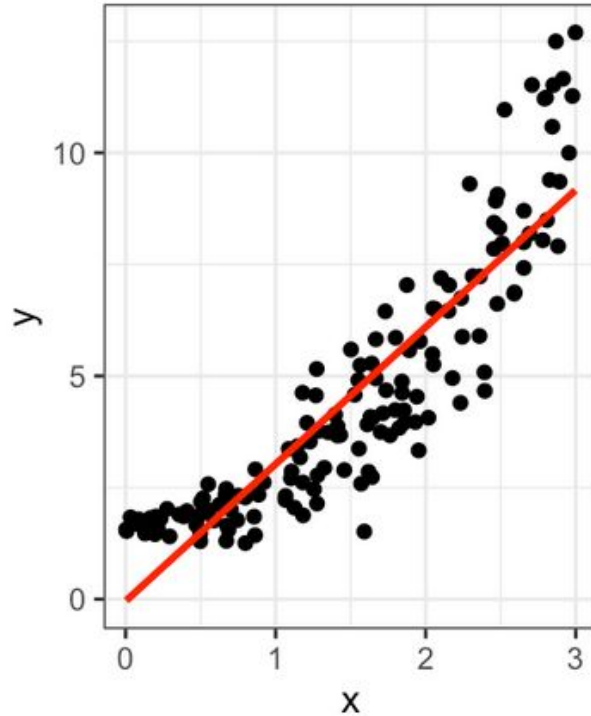
Logarithms smoosh big numbers



Which of these has better linear fit?



Taking the $\log(y)$ gives us something that's much better fit for a linear regression!



Moral of the story

- Make a residual plot to check if your data needs to be transformed (e.g., if it looks non-random and/or has a fan shape indicating heteroskedasticity)
- Transform your data accordingly before you run an OLS regression!!

Moral of the story

- Make a residual plot to check if your data needs to be transformed (e.g., if it looks non-random and/or has a fan shape indicating heteroskedasticity)
- Transform your data accordingly before you run an OLS regression!! **Do not run a regression on heteroskedastic data!**

When someone asks "who used OLS
for this heteroskedastic dataset?"



There are so many transformations that you can do! E.g. sqrt, squared, log, etc.

Method	Transform	Regression equation	Predicted value (\hat{y})
Standard linear regression	None	$y = b_0 + b_1x$	$\hat{y} = b_0 + b_1x$
Exponential model	DV = $\log(y)$	$\log(y) = b_0 + b_1x$	$\hat{y} = 10^{b_0 + b_1x}$
Quadratic model	DV = $\text{sqrt}(y)$	$\text{sqrt}(y) = b_0 + b_1x$	$\hat{y} = (b_0 + b_1x)^2$
Reciprocal model	DV = $1/y$	$1/y = b_0 + b_1x$	$\hat{y} = 1 / (b_0 + b_1x)$
Logarithmic model	IV = $\log(x)$	$y = b_0 + b_1\log(x)$	$\hat{y} = b_0 + b_1\log(x)$
Power model	DV = $\log(y)$ IV = $\log(x)$	$\log(y) = b_0 + b_1\log(x)$	$\hat{y} = 10^{b_0 + b_1\log(x)}$

Be careful with interpretations!

- Transforming your data is often necessary for your analysis, but the way you interpret the regression (especially the “summarize the relationship” step) will **change!**

Last time on **interpreting regressions**

1. Summarize relationship between variables
2. Make predictions
3. Inspect outliers and other oddities

Last time on interpreting regressions

1. Summarize relationship between variables
2. Make predictions
3. Inspect outliers and other oddities

If x increases by 1 unit, we expect y to _____?

Last time on interpreting regressions

1. Summarize relationship between variables
2. Make predictions
3. Inspect outliers and other oddities

If x increases by 1 unit, we expect y to increase/decrease by β

Interpreting regressions with logs (lns)

Model	Interpretation
Linear $y = \alpha + \beta x$	1 unit change in x is associated with a β unit change in y
Linear-log $y = \alpha + \beta \ln(x)$	If x is multiplied by e , we expect a β unit change in y
Log-linear $\ln(y) = \alpha + \beta x$	For a 1 unit change in x , we expect y to be multiplied by e^β
Log-log $\ln(y) = \alpha + \beta \ln(x)$	If x is multiplied by e , we expect y to be multiplied by e^β

This will be on the midterm!

Model	Interpretation
Linear $y = \alpha + \beta x$	1 unit change in x is associated with a β unit change in y
Linear-log $y = \alpha + \beta \ln(x)$	If x is multiplied by e , we expect a β unit change in y
Log-linear $\ln(y) = \alpha + \beta x$	For a 1 unit change in x , we expect y to be multiplied by e^β
Log-log $\ln(y) = \alpha + \beta \ln(x)$	If x is multiplied by e , we expect y to be multiplied by e^β

This is what we're used to

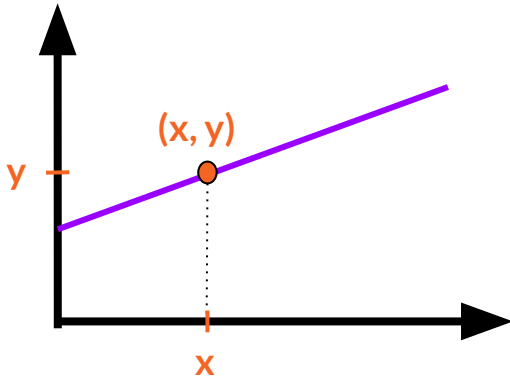
Model	Interpretation
Linear $y = \alpha + \beta x$	1 unit change in x is associated with a β unit change in y
Linear-log $y = \alpha + \beta \ln(x)$	If x is multiplied by e , we expect a β unit change in y
Log-linear $\ln(y) = \alpha + \beta x$	For a 1 unit change in x , we expect y to be multiplied by e^β
Log-log $\ln(y) = \alpha + \beta \ln(x)$	If x is multiplied by e , we expect y to be multiplied by e^β

Linear
 $y = \alpha + \beta x$

1 unit change in x is associated with a β unit change in y

How do we know this is true? ↗

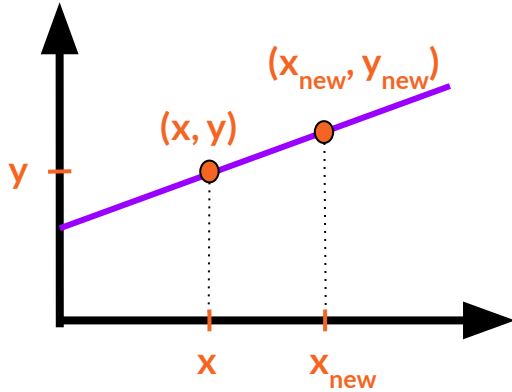
Let's say we start with our original value of x .



Linear
 $y = \alpha + \beta x$

1 unit change in x is associated with a β unit change in y

How do we know this is true? ↗



Let's say we start with our original value of x .

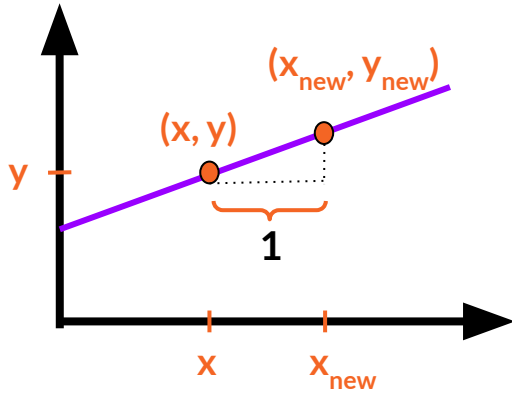
We can define a new value of x called x_{new} which represents a 1 unit change in x

$$x_{\text{new}} = x + 1$$

Linear
 $y = \alpha + \beta x$

1 unit change in x is associated with a β unit change in y

How do we know this is true? ↗



Let's say we start with our original value of x .

We can define a new value of x called x_{new} which represents a 1 unit change in x

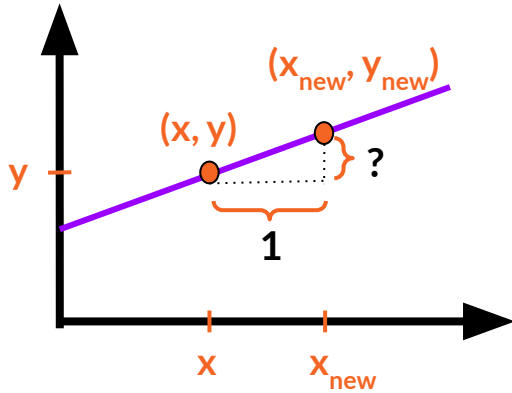
$$x_{\text{new}} = x + 1$$

Linear

$$y = \alpha + \beta x$$

1 unit change in x is associated with a β unit change in y

How do we know this is true? ↗



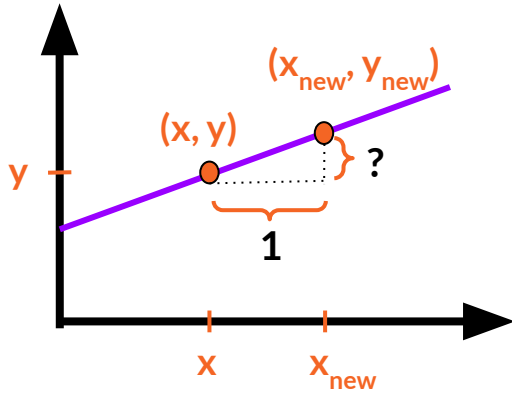
If we move from x to x_{new} we want to know how much the change is from y to y_{new}

Linear

$$y = \alpha + \beta x$$

1 unit change in x is associated with a β unit change in y

How do we know this is true? ↗



If we move from x to x_{new} we want to know how much the change is from y to y_{new}

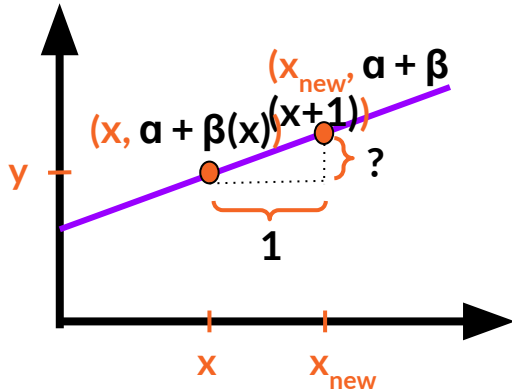
The purple line is linear, so $y_{\text{new}} = \alpha + \beta x_{\text{new}}$

Linear

$$y = \alpha + \beta x$$

1 unit change in x is associated with a β unit change in y

How do we know this is true? ↗



If we move from x to x_{new} we want to know how much the change is from y to y_{new}

The purple line is linear, so $y_{\text{new}} = \alpha + \beta x_{\text{new}}$

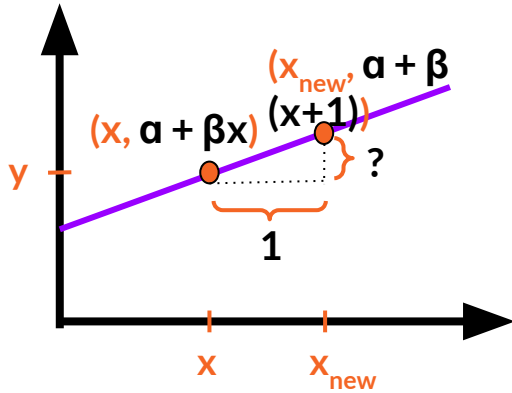
$$y_{\text{new}} = \alpha + \beta(x + 1)$$

Linear

$$y = \alpha + \beta x$$

1 unit change in x is associated with a β unit change in y

How do we know this is true? ↗



If we move from x to x_{new} we want to know how much the change is from y to y_{new}

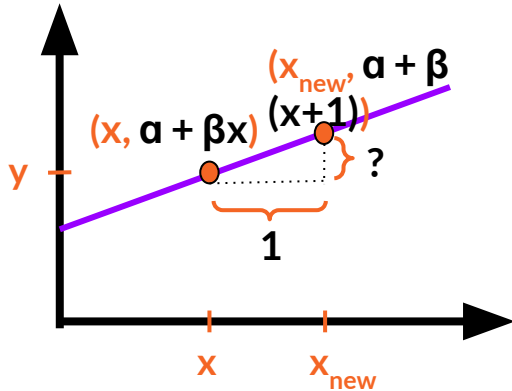
$$y_{\text{new}} - y = ?$$

Linear

$$y = \alpha + \beta x$$

1 unit change in x is associated with a β unit change in y

How do we know this is true? ↗



If we move from x to x_{new} we want to know how much the change is from y to y_{new}

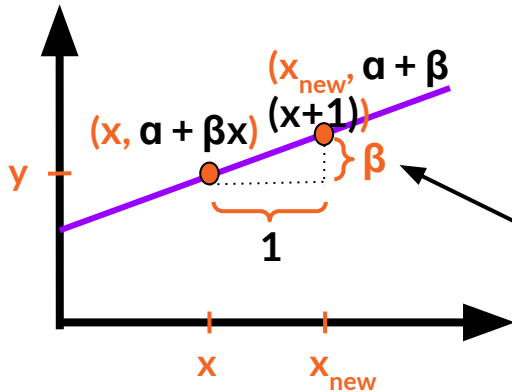
$$\begin{aligned} y_{\text{new}} - y &= [\alpha + \beta(x+1)] - [\alpha + \beta x] \\ &= [\alpha + \beta x + \beta] - [\alpha + \beta x] \\ &= \beta \end{aligned}$$

Linear

$$y = \alpha + \beta x$$

1 unit change in x is associated with a β unit change in y

How do we know this is true? ↗



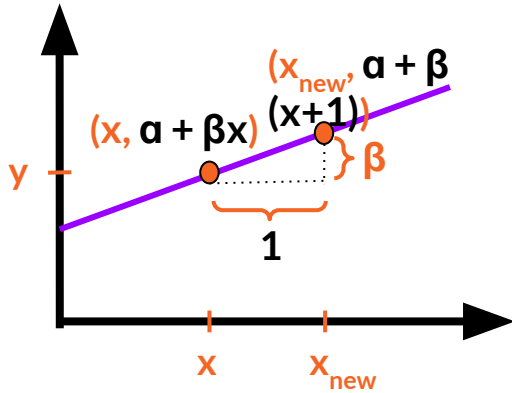
If we move from x to x_{new} we want to know how much the change is from y to y_{new}

$$\begin{aligned} y_{\text{new}} - y &= [\alpha + \beta(x+1)] - [\alpha + \beta x] \\ &= [\alpha + \beta x + \beta] - [\alpha + \beta x] \\ &= \beta \end{aligned}$$

Linear
 $y = \alpha + \beta x$

1 unit change in x is associated with a β unit change in y

How do we know this is true? ↗



For linear models, increasing x by 1 makes y increase by β

This makes sense visually, since β is our slope

But what if we use log-linear instead?

Model	Interpretation
Linear $y = \alpha + \beta x$	1 unit change in x is associated with a β unit change in y
Linear-log $y = \alpha + \beta \ln(x)$	If x is multiplied by e , we expect a β unit change in y
Log-linear $\ln(y) = \alpha + \beta x$	For a 1 unit change in x , we expect y to be multiplied by e^β
Log-log $\ln(y) = \alpha + \beta \ln(x)$	If x is multiplied by e , we expect y to be multiplied by e^β

Log-linear

$$\ln(y) = \alpha + \beta x$$

For a 1 unit change in x , we expect y to be multiplied by e^β

How do we know this is true? ↗

Log-linear

$$\ln(y) = \alpha + \beta x$$

For a 1 unit change in x , we expect y to be multiplied by e^β

How do we know this is true? ↗

We can do the same derivation with $x_{\text{new}} = x + 1$

Log-linear

$$\ln(y) = \alpha + \beta x$$

For a 1 unit change in x , we expect y to be multiplied by e^β

How do we know this is true? ↗

We can do the same derivation with $x_{\text{new}} = x + 1$

But this time, $\ln(y) = \alpha + \beta x$

This means that $y = e^{\alpha + \beta x}$

Log-linear

$$\ln(y) = \alpha + \beta x$$

For a 1 unit change in x , we expect y to be multiplied by e^β

How do we know this is true? ↗

Our goal is still to find the difference
between y_{new} and y

Log-linear

$$\ln(y) = \alpha + \beta x$$

For a 1 unit change in x , we expect y to be multiplied by e^β

How do we know this is true? ↗

Our goal is still to find the difference between y_{new} and y

We know $y = e^{\alpha + \beta x}$

We also have $y_{\text{new}} = e^{\alpha + \beta(x+1)}$

Log-linear

$$\ln(y) = \alpha + \beta x$$

For a 1 unit change in x , we expect y to be multiplied by e^β

How do we know this is true? ↗

Our goal is still to find the difference between y_{new} and y

We know $y = e^{\alpha + \beta x}$

We also have $y_{\text{new}} = e^{\alpha + \beta(x+1)}$

Exponentiation

rules: $e^{x+y} = e^x e^y$

Log-linear

$$\ln(y) = \alpha + \beta x$$

For a 1 unit change in x , we expect y to be multiplied by e^β

How do we know this is true? ↗

Our goal is still to find the difference between y_{new} and y

We know $y = e^{\alpha + \beta x}$

We also have $y_{\text{new}} = e^{\alpha + \beta(x+1)} = e^{\alpha + \beta x + \beta} = e^{\alpha + \beta x} e^\beta$

Exponentiation

rules: $e^{x+y} = e^x e^y$

Log-linear

$$\ln(y) = \alpha + \beta x$$

For a 1 unit change in x , we expect y to be multiplied by e^β

How do we know this is true? ↗

Our goal is still to find the difference between y_{new} and y

We know $y = e^{\alpha + \beta x}$

We also have $y_{\text{new}} = e^{\alpha + \beta(x+1)} = e^{\alpha + \beta x + \beta} = e^{\alpha + \beta x} e^\beta$

Log-linear

$$\ln(y) = \alpha + \beta x$$

For a 1 unit change in x , we expect y to be multiplied by e^β

How do we know this is true? ↗

Our goal is still to find the difference between y_{new} and y

We know $y = e^{\alpha + \beta x}$

We also have $y_{\text{new}} = e^{\alpha + \beta(x+1)} = e^{\alpha + \beta x + \beta} = e^{\alpha + \beta x} e^\beta$

$y_{\text{new}} = ?$ (in terms of y)

Log-linear

$$\ln(y) = \alpha + \beta x$$

For a 1 unit change in x , we expect y to be multiplied by e^β

How do we know this is true? ↗

Our goal is still to find the difference between y_{new} and y

We know $y = e^{\alpha + \beta x}$

We also have $y_{\text{new}} = e^{\alpha + \beta(x+1)} = e^{\alpha + \beta x + \beta} = e^{\alpha + \beta x} e^\beta$

$$y_{\text{new}} = y * e^\beta$$

Log-linear

$$\ln(y) = \alpha + \beta x$$

For a 1 unit change in x , we expect y to be multiplied by e^β

How do we know this is true? ↗

Our goal is still to find the difference between y_{new} and y

We know $y = e^{\alpha + \beta x}$

We also have $y_{\text{new}} = e^{\alpha + \beta(x+1)} = e^{\alpha + \beta x + \beta} = e^{\alpha + \beta x} e^\beta$

$$y_{\text{new}} = y * e^\beta$$

You can re-derive all of these interpretations with log or exp rules!

Model	Interpretation
Linear $y = \alpha + \beta x$	1 unit change in x is associated with a β unit change in y
Linear-log $y = \alpha + \beta \ln(x)$	If x is multiplied by e , we expect a β unit change in y
Log-linear $\ln(y) = \alpha + \beta x$	For a 1 unit change in x , we expect y to be multiplied by e^β
Log-log $\ln(y) = \alpha + \beta \ln(x)$	If x is multiplied by e , we expect y to be multiplied by e^β

What sort of model should we use?

Linear

$$y = a + \beta x$$

Linear-log

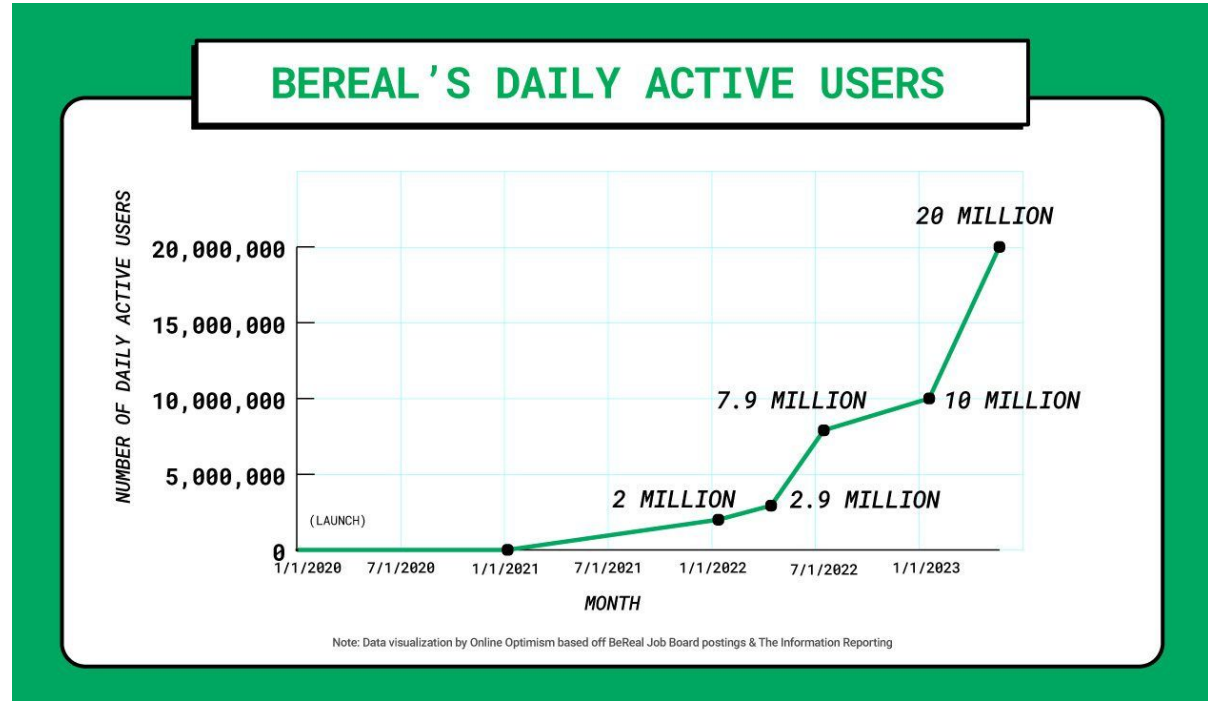
$$y = a + \beta \ln(x)$$

Log-linear

$$\ln(y) = a + \beta x$$

Log-log

$$\ln(y) = a + \beta \ln(x)$$



What sort of model should we use?

Linear

$$y = a + \beta x$$

Linear-log

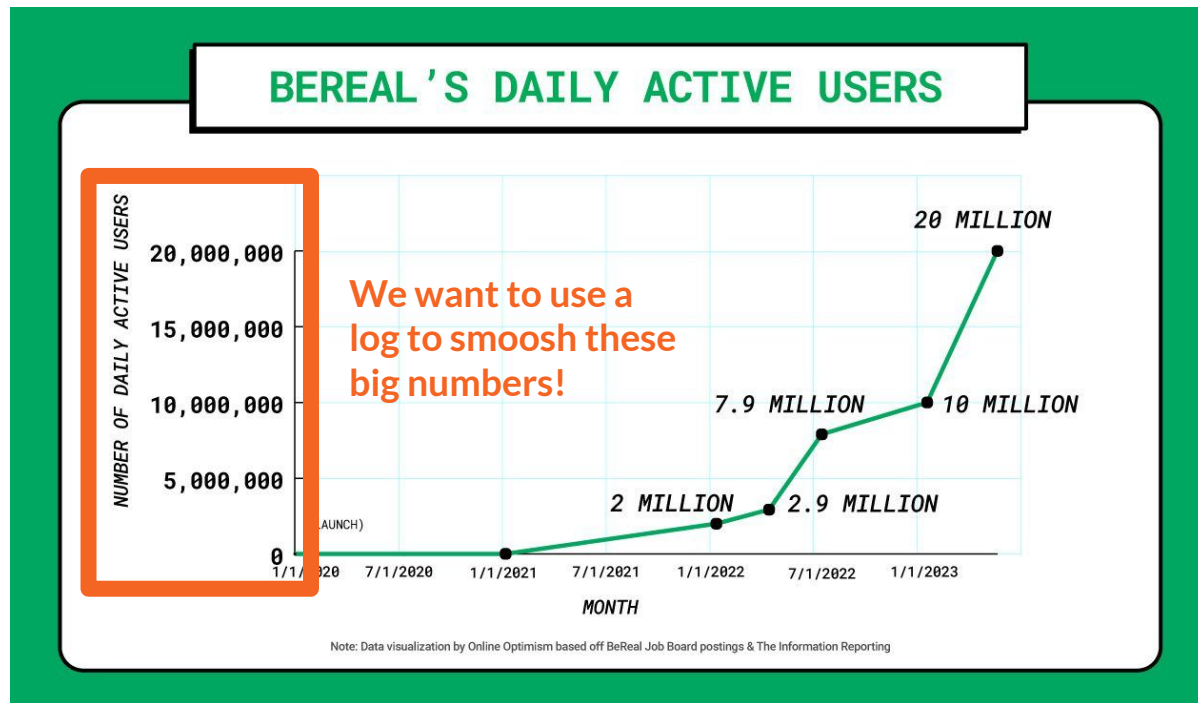
$$y = a + \beta \ln(x)$$

Log-linear

$$\ln(y) = a + \beta x$$

Log-log

$$\ln(y) = a + \beta \ln(x)$$



Regression interpretations: **summarize relationship**

x = millimeters of rainfall

y = umbrellas sold

$$y = -19 + 0.45x$$

Summarize relationship
between variables:

Our model shows a positive relationship between rain and sales of umbrellas; specifically, each additional mm of rain corresponds to an extra 0.45 umbrellas we expect to be sold.

x = quarters (from Q1-2020 to Q2-2022)

y = # BeReal app users

$$\ln(y) = -2.05 + 1.95x$$

Summarize relationship
between variables:

Log-linear

$$\ln(y) = \alpha + \beta x$$

For a 1 unit change in x , we expect y to be multiplied by e^β

x = millimeters of rainfall

y = umbrellas sold

$$y = -19 + 0.45x$$

Summarize relationship
between variables:

Our model shows a positive relationship between rain and sales of umbrellas; specifically, each additional mm of rain corresponds to an extra 0.45 umbrellas we expect to be sold.

x = quarters (from Q1-2020 to Q2-2022)

y = # BeReal app users

$$\ln(y) = -2.05 + 1.95x$$

Summarize relationship
between variables:

Log-linear

$$\ln(y) = \alpha + \beta x$$

For a 1 unit change in x , we expect y to be multiplied by e^β

x = millimeters of rainfall

y = umbrellas sold

$$y = -19 + 0.45x$$

Summarize relationship
between variables:

Our model shows a positive relationship between rain and sales of umbrellas; specifically, each additional mm of rain corresponds to an extra 0.45 umbrellas we expect to be sold.

x = quarters (from Q1-2020 to Q2-2022)

y = # BeReal app users

$$\ln(y) = -2.05 + 1.95x$$

Summarize relationship
between variables:

Our model shows a positive relationship between quarters and # BeReal app users; specifically, each additional quarter in time corresponds to $e^{1.95} = 7.03$ times more BeReal app users than the previous quarter

Regression interpretations: **make predictions**

x = millimeters of rainfall

y = umbrellas sold

$$y = -19 + 0.45x$$

Make predictions:

This model indicates that at the annual Ithaca average of 110mm of rainfall, we should expect to sell 30 umbrellas.

x = quarters (from Q1-2020 to Q2-2022)

y = # BeReal app users

$$\ln(y) = -2.05 + 1.95x$$

Make prediction for x=0:

Regression interpretations: **note oddities**

x = millimeters of rainfall

y = umbrellas sold

$$y = -19 + 0.45x$$

Inspect oddities / outliers:

We expect this model to hold for rainfall amounts between 80-170mm, but cannot extrapolate further.

x = quarters (from Q1-2020 to Q2-2022)

y = # BeReal app users

$$\ln(y) = -2.05 + 1.95x$$

Inspect oddities / outliers:

Become friends with this table!

Model	Interpretation
Linear $y = \alpha + \beta x$	1 unit change in x is associated with a β unit change in y
Linear-log $y = \alpha + \beta \ln(x)$	If x is multiplied by e , we expect a β unit change in y 1% change in x is associated with a $0.01 \cdot \beta$ unit change in y
Log-linear $\ln(y) = \alpha + \beta x$	For a 1 unit change in x , we expect y to be multiplied by e^β 1 unit change in x is associated with a $100 \cdot (\exp(\beta) - 1)\%$ change in y
Log-log $\ln(y) = \alpha + \beta \ln(x)$	If x is multiplied by e , we expect y to be multiplied by e^β 1% change in x is associated with a $\beta\%$ change in y (<i>elasticity</i>)