Intro to Statistical Analysis with Python

Sunny Fang, CSC Computing Fellow

Agenda

01

Statistical testing

A primer/review on statistical testing & why it is important

02

Power of visualizations

Why we love visualizations...not just numbers

03

One-sample t-test

ft. live coding!!!

04

Demo: Logistic Regression (if time permits)

*NOTE: Whenever you see `lets_code()`, it means to go back to the Python notebook!

01

Statistics 101

What is statistics?

There are two main tasks in statistics: **explanation/inference** vs. **prediction.**

Examples of inference:

- Estimating and interpreting the parameters of a model
- hypothesis tests and confidence intervals

Examples of prediction:

- classification problems: Random forest, logistic regression, etc

What is statistics?

There are two main tasks in statistics: **explanation/inference** vs. **prediction.**

Examples of inference:

- Estimating and interpreting the parameters of a model
- hypothesis tests and confidence intervals

Examples of prediction:

- classification problems: Random forest, logistic regression, etc

TODAY, we are going to focus on inference

Statistical Modeling: how

FIG. 2. *Steps in the statistical modeling process.*

Statistical Modeling: how

our focus today

FIG. 2. *Steps in the statistical modeling process.*

Note: if you are building a predictive model for your thesis, please feel free to let me know!

Statistical Modeling: how

our focus today

FIG. 2. *Steps in the statistical modeling process.*

the most crucial step

Note: if you are building a predictive model for your thesis, please feel free to let me know!

How do I choose the correct statistical test?

flowchart for guidance

tinyurl.com/choosing-stats-test-csc

interactive flowchart

www.statsflowchart.co.uk

How do I choose the correct statistical test?

Example heuristic:

1. How many variables am I analyzing? (univariate vs. bivariate or more)
2. What type of outcome and predictor is it?

   continuous (numerical) or **categorical (qualitative)?**
3. Is my data parametric or not?
   - ***parametric*** is a fancy way of saying "data follows a distribution (usually normal)"
   - Some examples of ***nonparametric*** data include: (1) descriptive statistics (e.g., median) but not the data points, (2) skewed data that does not follow the normal distribution

A quick note on “p-value”

- **What is p-value?**

A quick note on “p-value”

- **What is p-value?**
  Given any test, the p-value is *how likely it is to get a particular result from sample data,* ***assuming the null hypothesis is true.***
  Can also be written as **P(observed result | H_0 true)**

  "If we found a p-value of 7%, it would say 'if we were to take these data as just decisive against the null hypothesis, then in 7% of the cases in the long run in which the hypothesis is true, it will get rejected falsely'."
  - David Cox

- **p-value does not…**
  1. Tell you the probability of *hypothesis being tested is true* (or false)
  2. Quantify evidence

A quick note on “p-value”

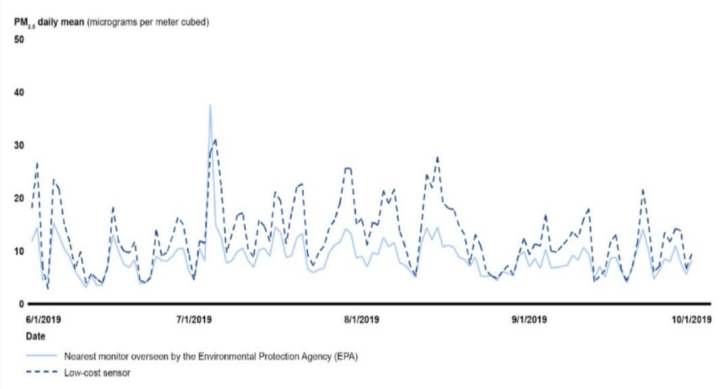
- **What is p-value?**
  Given any test, the p-value is *how likely it is to get a particular result from sample data if the null hypothesis is true*.

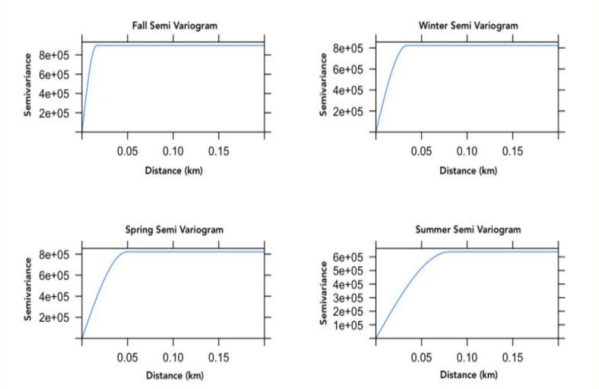
  It is important to remember that the significance level of 0.05 is *often arbitrary!*

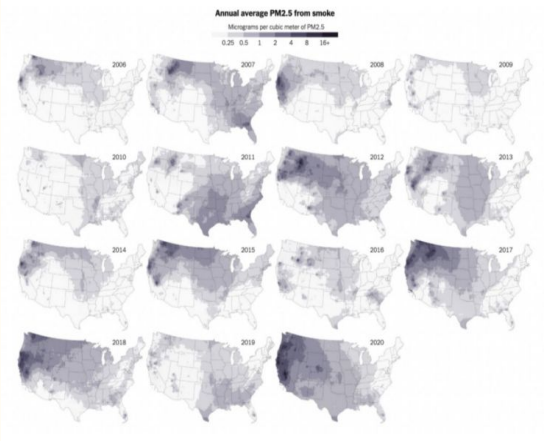
02

Why visualize?

some visuals from past theses







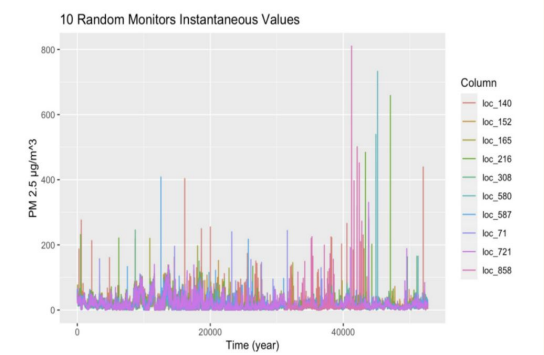


Figure 3: *10 randomly selected monitors of the 1058 total uses in study as an instantaneous time series with 10 minutes averages from September 1, 2021 to September 1, 2022.*

Activity!

Visit **one** of the three links, skim through the article and pay attention to the data visualization. Note one good thing and one thing you'd change from it.

Unequal access to water

tinyurl.com/water-csc

Planet or Plastic?

tinyurl.com/planetorplastic-csc

(email required)

"A Good Life for All"

tinyurl.com/goodlife-csc

How to choose the best graph?

- Before we jump into coding...**pause and think about:**
  1. What is the question I am trying to answer?
  2. What type of visualization suits my question the most?

     ⇒ tinyurl.com/choose-best-graph
  3. What are my x and y-axes?
  4. What do I want to highlight in my graph? (Do I need a caption? Annotations?)

Quick review: data exploration

Reading file

```
import pandas as pd


df = pd.read_csv("your_file_name.csv")


df.head(5) # first n rows, default 5
```

Reading file

```
import pandas as pd
```

since read_csv is a function in pandas, we need to explicate the library we are using

```
df = pd.read_csv("your_file_name.csv")
```

This is a different case where “df” is your variable, and “head” is one of the many methods you can access from a DataFrame object

```
df.head(5) # first n rows, default 5
```

Understanding the Dataset

```
print(df.shape) # -> rows, columns
print(f"The dataset has {df.shape[0]} rows and
{df.shape[1]} columns\n")


print(df.info()) # -> col_names, values, memory
```

Exploratory Data Analysis

- **Exploratory Data Analysis (EDA)** is a process that summarize their main characteristics of a dataset
  - involves *descriptive statistics* (median, mean, mode and measures of variability)
  - often utilizes data visualization techniques
- Data types
  - **Numeric**: magnitude (e.g., temperature)
  - **Categorical**: no order (e.g., states)
  - **Ordinal**: ordered, (e.g., education level)

Data Cleaning

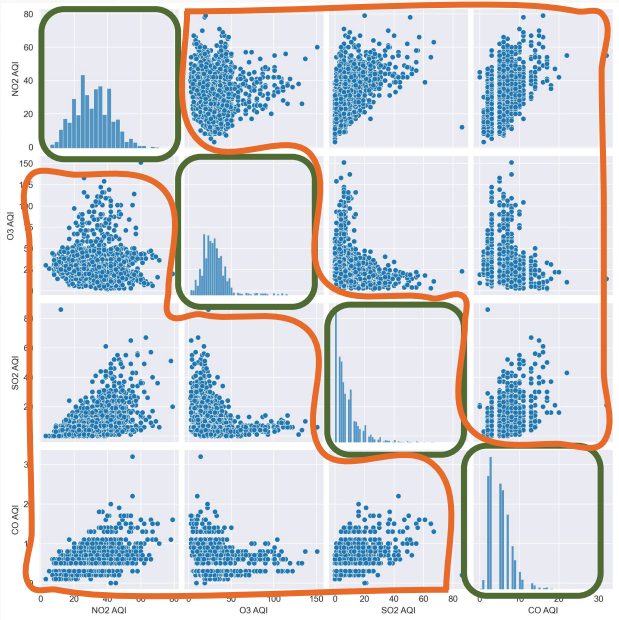
- `df.dropna(inplace=True)` drops all NA values in the dataset
- `df.info()` tells you the data type for each column
  - if int or float ⇒ numerical
  - otherwise ⇒ usually *ordinal* or *categorical*

Exploratory Data Analysis: Numeric

- `df.describe()` returns the count, mean, std, min, 25%, 50%, 75%, and max of **ALL** numerical columns; or you can get individual metrics by:
  - **Get central tendency measures**
    - `df['col_name'].mean()` ⇒ mean
    - `df['col_name'].median()` ⇒ median
    - `df['col_name'].median()` ⇒ mode
    - `df['col_name'].quantile([n])` ⇒ returns the n*100% percentile
  - **Get spread**
    - `df['col_name'].std()` ⇒ standard deviation
    - `df['col_name'].var()` ⇒ variance

Exploratory Data Analysis: Numeric

- `sns.pairplot()` returns (1) the **univariate distribution** and (2) the **bivariate scatterplot** between two variables



Data Slicing

Syntax:

- `df.loc[row_name, col_name]`
- `df.iloc[row_index, col_index]`

subset by row

```
df.loc[0:4]
```

```
df.iloc[0:4]
```

	State	County	City	Date Local	NO2 Units	NO2 Mean
0	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
1	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
2	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
3	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
4	Arizona	Maricopa	Phoenix	2000-01-02	Parts per billion	22.958333
...	...	...	...	...	...	...
1746656	Wyoming	Laramie	Not in a city	2016-03-30	Parts per billion	1.083333
1746657	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130
1746658	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130
1746659	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130
1746660	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130

Data Slicing

Syntax:

- `df.loc[row_name, col_name]`
- `df.iloc[row_index, col_index]`
- `(col only) df[col_name]`

subset by columns

```
df.loc[:,"County"]
```

"all rows"

```
df.iloc[:,2]
```

```
df["County"]
```

	State	County	City	Date Local	NO2 Units	NO2 Mean
0	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
1	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
2	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
3	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
4	Arizona	Maricopa	Phoenix	2000-01-02	Parts per billion	22.958333
...	...	...	...	...	...	...
1746656	Wyoming	Laramie	Not in a city	2016-03-30	Parts per billion	1.083333
1746657	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130
1746658	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130
1746659	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130
1746660	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130

Data Slicing

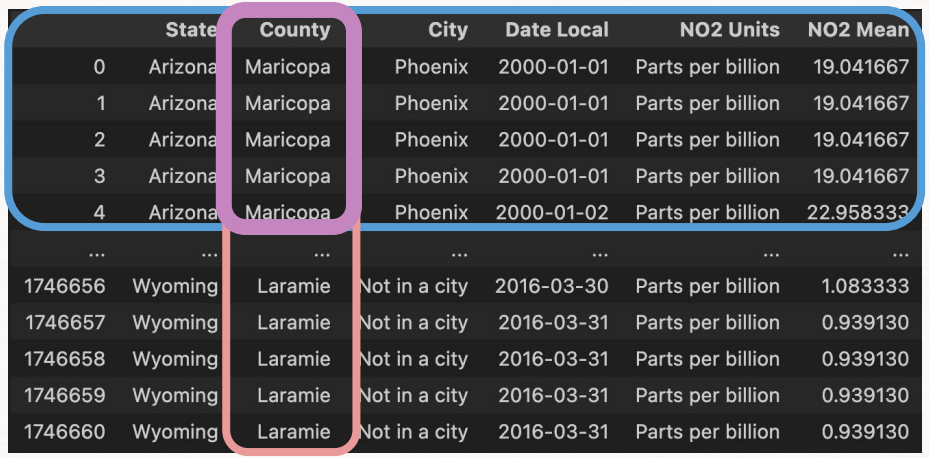
Syntax:

- `df.loc[row_name, col_name]`
- `df.iloc[row_index, col_index]`

subset by both

```
df.loc[0:4,"County"]
```

```
df.iloc[0:4, 2]
```



	State	County	City	Date Local	NO2 Units	NO2 Mean
0	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
1	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
2	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
3	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
4	Arizona	Maricopa	Phoenix	2000-01-02	Parts per billion	22.958333
...	...	...	...	...	...	...
1746656	Wyoming	Laramie	Not in a city	2016-03-30	Parts per billion	1.083333
1746657	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130
1746658	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130
1746659	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130
1746660	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130

Data Slicing

Syntax: `df.loc[row_name, [col_names]]`

subset by *multiple* columns

```
df.loc[:,["County","Date Local"]]
(or)
df[["County","Date Local"]]
```

this is a list object!!

	State	County	City	Date Local	NO2 Units	NO2 Mean
0	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
1	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
2	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
3	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
4	Arizona	Maricopa	Phoenix	2000-01-02	Parts per billion	22.958333
...	...	...	...	...	...	...
1746656	Wyoming	Laramie	Not in a city	2016-03-30	Parts per billion	1.083333
1746657	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130
1746658	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130
1746659	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130
1746660	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130

	County	Date Local
0	Maricopa	2000-01-01
1	Maricopa	2000-01-01
2	Maricopa	2000-01-01
3	Maricopa	2000-01-01
4	Maricopa	2000-01-02
...	...	...
1746656	Laramie	2016-03-30
1746657	Laramie	2016-03-31
1746658	Laramie	2016-03-31
1746659	Laramie	2016-03-31
1746660	Laramie	2016-03-31

Boolean Masking

Slicing data based on given conditions - recall *operators*!

1. find rows with NO2 mean > 20

```
df[df.loc[:,"NO2 Mean"] > 20]
```

Selects all rows of the column “NO2 Mean”

	State	County	City	Date Local	NO2 Units	NO2 Mean
0	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
1	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
2	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
3	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667
4	Arizona	Maricopa	Phoenix	2000-01-02	Parts per billion	22.958333
...	...	...	...	...	...	...
1746656	Wyoming	Laramie	Not in a city	2016-03-30	Parts per billion	1.083333
1746657	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130
1746658	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130
1746659	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130
1746660	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130

Boolean Masking

Slicing data based on given conditions - recall *operators*!

1. find rows with NO2 mean > 20

```
df[df.loc[:,"NO2 Mean"] > 20]
```

Selects the column "NO2 Mean"

Goes through each record and see if it satisfies the condition

	State	County	City	Date Local	NO2 Units	NO2 Mean	
0	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667	False
1	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667	False
2	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667	False
3	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667	False
4	Arizona	Maricopa	Phoenix	2000-01-02	Parts per billion	22.958333	True
...	...	...	...	...	...	...	...
1746656	Wyoming	Laramie	Not in a city	2016-03-30	Parts per billion	1.083333	False
1746657	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130	False
1746658	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130	False
1746659	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130	False
1746660	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130	False

Boolean Masking

Slicing data based on given conditions - recall *operators*!

1. find rows with NO2 mean > 20

```
df[df.loc[:,"NO2 Mean"] > 20]
```

Only keep the rows that returns True

	State	County	City	Date Local	NO2 Units	NO2 Mean	
0	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667	False
1	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667	False
2	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667	False
3	Arizona	Maricopa	Phoenix	2000-01-01	Parts per billion	19.041667	False
4	Arizona	Maricopa	Phoenix	2000-01-02	Parts per billion	22.958333	True
...	...	...	...	...	...	...	...
1746656	Wyoming	Laramie	Not in a city	2016-03-30	Parts per billion	1.083333	False
1746657	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130	False
1746658	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130	False
1746659	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130	False
1746660	Wyoming	Laramie	Not in a city	2016-03-31	Parts per billion	0.939130	False

03

Live-coding: T-test

Let's do some live coding!

tinyurl.com/cf-es-stats

Go to `Files` → `Save a copy in Drive`

`lets_code()` After creating the notebook on Colab/your local IDE, run Sections 0. Setup and 1. Data Processing

Python libraries

Pandas provides functionalities that are crucial to data analysis and is suitable for handling multiple file types (e.g., Comma-separated values, or csv, files)

Matplotlib and **Seaborn** are both libraries for data visualization. Seaborn is built on top of matplotlib

SciPy is a library for scientific computing in Python. Built on NumPy, it has several built-in functions for statistical tests.

Hypothesis testing: steps

STEP 1 Define the problem and data

STEP 2 Define null and alternative hypothesis

STEP 3 Define α (significance level)

STEP 4 Check assumptions

STEP 5 (Optional) Data visualization

STEP 6 Statistical Testing

STEP 7 Report Findings

Hypothesis testing: steps

STEP 1 Define the problem and data

STEP 2 Define null and alternative hypothesis

STEP 3 Define α (significance level)

STEP 4 Check assumptions

STEP 5 (Optional) Data visualization

STEP 6 Statistical Testing

STEP 7 Report Findings

Step 1: Define the problem and data

On October 1, 2015, the Environmental Protection Agency strengthened the National Ambient Air Quality Standards: "areas will meet the standards if the 4th highest daily maximum 8-hour ozone concentration per year, averaged over three years, is **equal to or less than 70 ppb** (or 0.07 ppm)

Here, we want to test whether or not the **state of California (CA)** meets the standard. For the purpose of this demonstration, we are going to loosen the "4th highest daily maximum 8-hour ozone concentration per year" assumption. Instead, we are going to see if the **average of the 1st max O3 value from 2011 to 2013*** meet the standards.

* i.e., take the max O_3 value daily for three years, and find the mean

Step 1: Define the problem and data

On October 1, 2015, the Environmental Protection Agency strengthened the National Ambient Air Quality Standards: "areas will meet the standards if the 4th highest daily maximum 8-hour ozone concentration per year, averaged over three years, is **equal to or less than 70 ppb** (or 0.07 ppm)

We'll revisit the problem in step 2!

Here, we want to test whether or not the **state of California (CA)** meets the standard. For the purpose of this demonstration, we are going to loosen the "4th highest daily maximum 8-hour ozone concentration per year" assumption. Instead, we are going to see if the **average of the 1st max O3 value from 2011 to 2013*** meet the standards.

* i.e., take the max O_3 value daily for three years, and find the mean

Step 1: Define the problem and **data**

Now that we know the data, let's go ahead and subset our data (review)!

We are looking for: Air quality dataset where

boolean masking

- WHERE State == 'California' AND
- WHERE Year >= 2011 AND
- WHERE Year <= 2013

subsetting

- Keeping columns 'O3 1st Max Value' and 'Date Local'

* Note: order of operation here doesn't matter too much; you can also **subset the column first** THEN **apply the conditions**

`lets_code()` Let's go back to step 1 under **#2. T-test example: one-sample t-test** in the Colab Notebook and subset the data together!

Hypothesis testing: steps

STEP 1 Define the problem and data

STEP 2 Define null and alternative hypothesis

STEP 3 Define α (significance level)

STEP 4 Check assumptions

STEP 5 (Optional) Data visualization

STEP 6 Statistical Testing

STEP 7 Report Findings

Hypothesis testing: steps

STEP 1 Define the problem and data

STEP 2 Define null and alternative hypothesis

STEP 3 Define α (significance level)

STEP 4 Check assumptions

STEP 5 (Optional) Data visualization

STEP 6 Statistical Testing

STEP 7 Report Findings

Step 2: Define null and alternative hypothesis

1. Recall the problem (question) we want to solve.

On October 1, 2015, the Environmental Protection Agency strengthened the National Ambient Air Quality Standards: "areas will meet the standards if the 4th highest daily maximum 8-hour ozone concentration per year, averaged over three years, is **equal to or less than 70 ppb** (or 0.07 ppm)

Here, we want to test whether or not the **state of California (CA) meets the standard.** For the purpose of this demonstration, we are going to loosen the "4th highest daily maximum 8-hour ozone concentration per year" assumption. Instead, we are going to see if the **average of the 1st max O3 value from 2011 to 2013*** meet the standards.

Step 2: Define null and alternative hypothesis

1. Recall the problem (question) we want to solve.
2. Formulate a research question based on the problem, such as: *Does the mean value of the daily max ozone concentration in **California exceed the EPA standard, 0.07**?*
3. Understand what "null" and "alternative" means:
   - Null: the perceived "status quo"
     (i.e., CA's mean O_3 concentration is less than or equal to EPA standard)
   - Alternative: the claim you are making
     (i.e., CA's mean O_3 concentration is greater than EPA standard)
4. Translate the above into your null and alternative hypothesis

source: https://resources.nu.edu/statsresources/hypothesis

Step 2: Define null and alternative hypothesis

Therefore, the hypotheses are defined as follows:

H_0: The mean of the daily 1st max ozone conc. in CA is **less than or equal to** 0.07.

*** NOTE:** The null hypothesis MUST ALWAYS CONTAIN an **"equality statement" (=, ≤, or ≥).** The choice among the three matters for the interpretation, but the hypothesis testing procedure remains the same!

H_A: The mean of the daily 1st max ozone conc. in CA is **greater than** 0.07.

*** NOTE:** Conversely, the alternative hypothesis MUST ALWAYS CONTAIN an **"inequality statement"(≠, <, or >).** The choice here **MATTERS (!!!!)** because it will affect the statistical test (e.g., two vs. one-tailed) you use.

Step 2: Define null and alternative hypothesis

What does it mean to reject null?

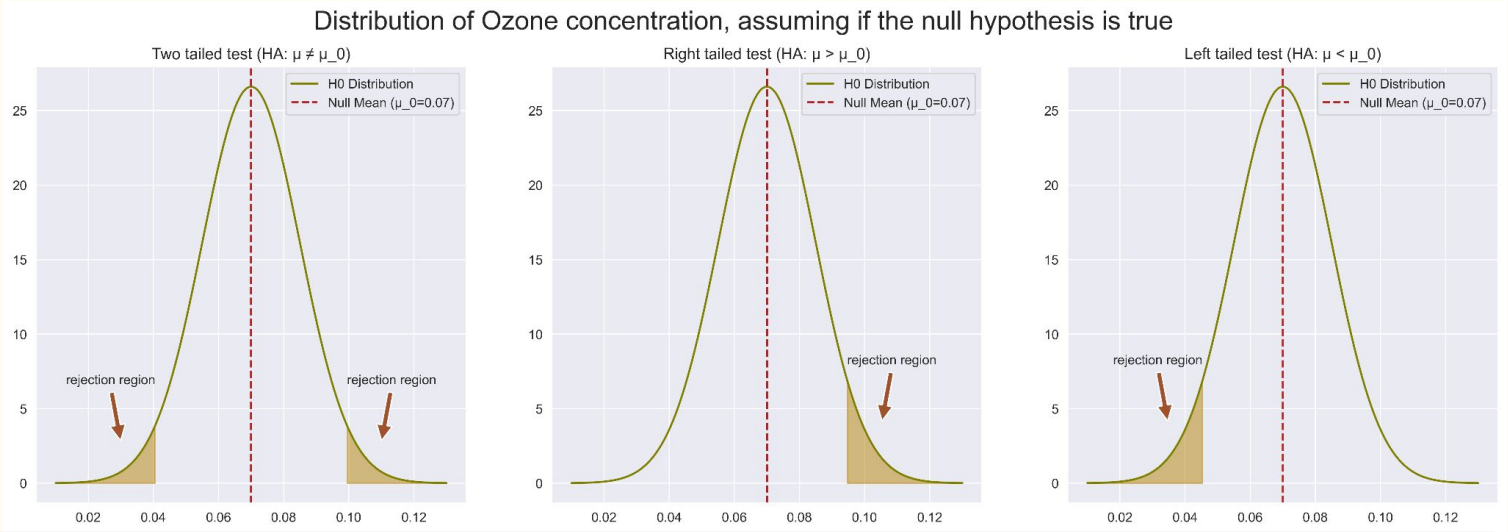
Think about the following three null-alternative hypothesis pairs:

1. Two-tailed test
   - **H_0:** The mean of the daily 1st max ozone conc. in CA is 0.07.
   - **H_A:** The mean of the daily 1st max ozone conc. in CA is **not equal to** 0.07.
2. Right-tailed test (what we're doing!)
   - **H_0:** The mean of the daily 1st max ozone conc. in CA is 0.07.
   - **H_A:** The mean of the daily 1st max ozone conc. in CA is **greater than** 0.07.
3. Left-tailed test
   - **H_0:** The mean of the daily 1st max ozone conc. in CA is 0.07.
   - **H_A:** The mean of the daily 1st max ozone conc. in CA is **less than** 0.07.

What does it mean to reject null?

Here, we see the null distribution (i.e., assuming the null hypothesis is true ⇒ the distribution is centered around the **null mean μ_0 = 0.07**).

Note: since we don't know the standard deviation, we use the sample std as proxy

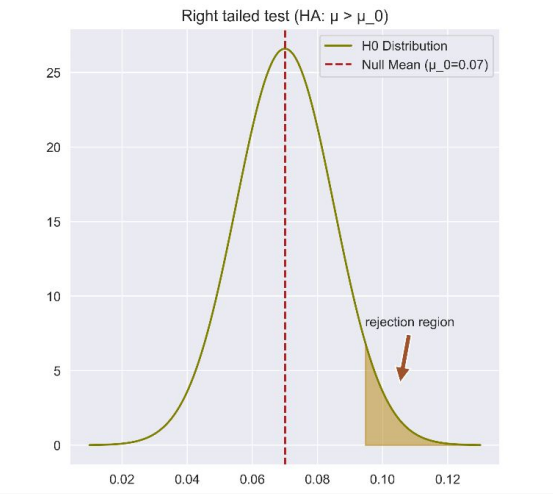


null distribution			
t-test type	**Two-tailed**	**Right-tailed**	**Left-tailed**
alternative hypothesis	The mean of the daily 1st max ozone conc. in CA is not equal to 0.07.	The mean of the daily 1st max ozone conc. in CA is **greater than 0.07.**	The mean of the daily 1st max ozone conc. in CA is **less than 0.07.**

What does it mean to reject null?

Intuitively, think of rejecting the null as having an **observed sample point estimate** (sample **mean** in this case) *so extreme* that it falls under the rejection region.

Let's look at the right-tailed test in more detail since this is what we are doing

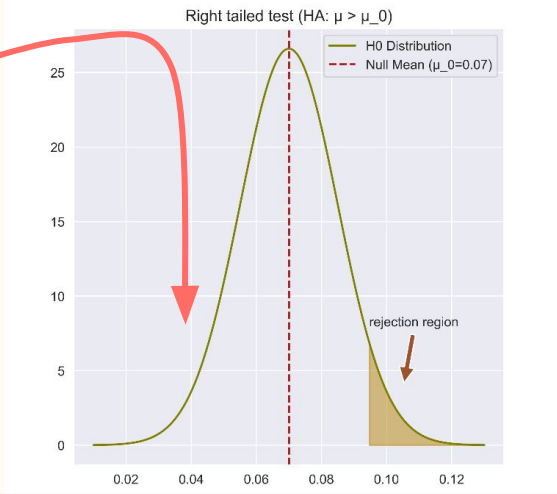


null distribution	
t-test type	**Right-tailed**
alternative hypothesis	The mean of the daily 1st max ozone conc. in CA is **greater than 0.07.**

What does it mean to reject null?

Intuitively, think of rejecting the null as having an **observed sample point estimate** (sample **mean** in this case) *so extreme* that it falls under the rejection region.

Our sample mean is around 0.039, which will be all the way over here



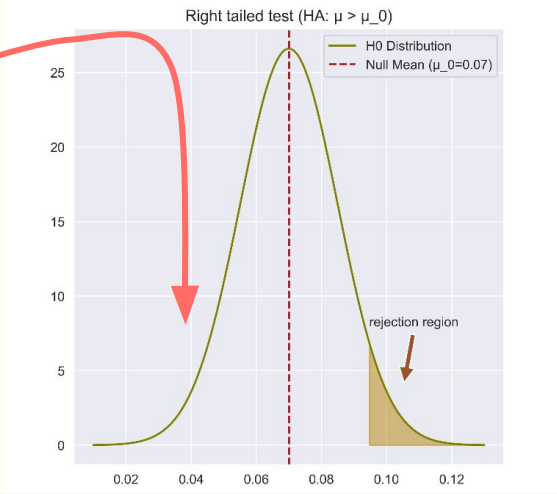
null distribution	
t-test type	**Right-tailed**
alternative hypothesis	The mean of the daily 1st max ozone conc. in CA is **greater than 0.07.**

What does it mean to reject null?

Intuitively, think of rejecting the null as having an **observed sample point estimate** (sample **mean** in this case) *so extreme* that it falls under the rejection region.

Our sample mean is around 0.039, which will be all the way over here

Intuitively, since it is far from the rejection region, it is likely that we will fail to reject the null



However, it is important that we still conduct the actual statistical test!!

null distribution	
t-test type	**Right-tailed**
alternative hypothesis	The mean of the daily 1st max ozone conc. in CA is **greater than 0.07.**

Hypothesis testing: steps

STEP 1 Define the problem and data

STEP 2 Define null and alternative hypothesis

STEP 3 **Define α (significance level)**

STEP 4 Check assumptions

STEP 5 (Optional) Data visualization

STEP 6 Statistical Testing

STEP 7 Report Findings

Step 3: Define α (significance level)

What exactly is significance level, or alpha (α)? Why is the "default" always 0.05? (TL; DR: it's a mutually agreed upon cutoff)

The definition is a bit out of scope for this workshop but in a nutshell...

- α = probability that we incorrectly reject the null hypothesis when the it is actually true, AKA **Type I error** or "false positives"
- i.e., α = 0.05 means we are allowing a 5% chance of *incorrectly* stating mean O_3 concentration in CA exceeds EPA standard when it **doesn't**.
- By lowering α, you are lowering the chance of making such error

source: https://www.statisticsteacher.org/2017/01/05/why-0-05-two-examples-that-put-students-in-the-role-of-decision-maker/

Hypothesis testing: steps

STEP 1 Define the problem and data

STEP 2 Define null and alternative hypothesis

STEP 3 Define α (significance level)

STEP 4 Check assumptions

STEP 5 (Optional) Data visualization

STEP 6 Statistical Testing

STEP 7 Report Findings

Hypothesis testing: steps

STEP 1 Define the problem and data

STEP 2 Define null and alternative hypothesis

STEP 3 Define α (significance level)

STEP 4 Check assumptions

STEP 5 (Optional) Data visualization

STEP 6 Statistical Testing

STEP 7 Report Findings

Step 4: Check Assumptions

Assumption	Explanation	How to check/remediate
Independence of samples	Data are (1) not influenced by implicit factor & (2) not correlated over time*	Some solutions here
Identically distributed	All samples are drawn from the sample probability distribution	for two-sample: Kolmogorov-Smirnov Test **OR** QQ-plots to compare distribution
Normality	Samples should be normally distributed OR have n > 30	Check if sample size > 30 **OR** QQ-plots to see if data points generally lie on 45° line **OR** if not normal, try data transformation
Equal Variance	Samples should have equal variances	for two-sample: QQ-plots **OR** box plots to compare the variance

Source: https://www.datacamp.com/tutorial/an-introduction-to-python-t-tests#t-test-assumptions-under

Step 4: Check Assumptions

For one-sample t-test, this is the one we're focusing on:

Assumption	Explanation	How to check/remediate
Independence of samples	Data are (1) not influenced by implicit factor & (2) not correlated over time*	Some solutions here
Identically distributed	All samples are drawn from the sample probability distribution	for two-sample: Kolmogorov-Smirnov Test **OR** QQ-plots to compare distribution
Normality	Samples should be normally distributed OR have n > 30	Check if sample size > 30 **OR** QQ-plots to see if data points generally lie on 45° line **OR** if not normal, try data transformation
Equal Variance	Samples should have equal variances	for two-sample: QQ-plots **OR** box plots to compare the variance

Source: https://www.datacamp.com/tutorial/an-introduction-to-python-t-tests#t-test-assumptions-under

Step 4: Check Assumptions

Assumption	Explanation	How to check/remediate
Independence of samples	Data are (1) not influenced by implicit factor & (2) not correlated over time*	Some solutions here
Identically distributed	All samples are drawn from the sample probability distribution	for two-sample: Kolmogorov-Smirnov Test **OR** QQ-plots to compare distribution
Normality	Samples should be normally distributed OR have n > 30	Check if sample size > 30 **OR** QQ-plots to see if data points generally lie on 45° line **OR** if not normal, try data transformation
Equal Variance	Samples should have equal variances	for two-sample: QQ-plots **OR** box plots to compare the variance

For one-sample t-test, this is the one we're focusing on:

Since n > 30, we can assume this is met

Source: https://www.datacamp.com/tutorial/an-introduction-to-python-t-tests#t-test-assumptions-under

Step 4: Check Assumptions

⚠️ **important disclaimer:** the data we are looking at is temporal (i.e., changes over time), so it violates the independence assumption, but for the purpose of this demo,, let's assume it satisfies all assumptions..

Assumption	Explanation	How to check/remediate
Independence of samples	Data are (1) not influenced by implicit factor & (2) not correlated over time*	Some solutions here
Identically distributed	All samples are drawn from the sample probability distribution	for two-sample: Kolmogorov-Smirnov Test **OR** QQ-plots to compare distribution
Normality	Samples should be normally distributed OR have n > 30	Check if sample size > 30 **OR** QQ-plots to see if data points generally lie on 45° line **OR** if not normal, try data transformation
Equal Variance	Samples should have equal variances	for two-sample: QQ-plots **OR** box plots to compare the variance

Source: https://www.datacamp.com/tutorial/an-introduction-to-python-t-tests#t-test-assumptions-under

Hypothesis testing: steps

STEP 1 Define the problem and data

STEP 2 Define null and alternative hypothesis

STEP 3 Define α (significance level)

STEP 4 Check assumptions

STEP 5 (Optional) Data visualization

STEP 6 Statistical Testing

STEP 7 Report Findings

Hypothesis testing: steps

STEP 1 Define the problem and data

STEP 2 Define null and alternative hypothesis

STEP 3 Define α (significance level)

STEP 4 Check assumptions

STEP 5 (Optional) Data visualization

STEP 6 Statistical Testing

STEP 7 Report Findings

Step 5: Data Visualization

Data visualization helps

- Checking assumptions (see step 4)
- Understand the observed sample
  - e.g., check the “distribution” of the sample with a histogram
- Explore other data trends that may not be answered using a statistical test
  - e.g., when is it more likely to have O_3 conc. above standard?

Good resource to have ⇒ tinyurl.com/choose-best-graph

Step 5: Data Visualization

`lets_code()` Let's go back to step 5 under **#2. T-test example: one-sample t-test** in the Colab Notebook and show how we can visualize O_3 distribution and change over time!

Hypothesis testing: steps

STEP 1 Define the problem and data

STEP 2 Define null and alternative hypothesis

STEP 3 Define α (significance level)

STEP 4 Check assumptions

STEP 5 (Optional) Data visualization

STEP 6 Statistical Testing

STEP 7 Report Findings

Hypothesis testing: steps

STEP 1 Define the problem and data

STEP 2 Define null and alternative hypothesis

STEP 3 Define α (significance level)

STEP 4 Check assumptions

STEP 5 (Optional) Data visualization

STEP 6 Statistical Testing

STEP 7 Report Findings

Step 6: Statistical Testing

- Like coding, you don't necessarily need to memorize the formula of a t-statistic, but it's important to gain intuition
- What is t-statistic?
  - Ratio between the difference of observed ($\bar{x}$) and expected (μ) to the standard error

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

- p-value, in relation to the t-statistic, is the probability of observing a t-statistic as extreme as your observed value, **assuming if the null hypothesis is true**

Good video if you'd like to learn more: https://www.youtube.com/watch?v=tI6mdx3s0zk

Step 6: Statistical Testing

`lets_code()` Go to step 6 under #2. T-test example: one-sample t-test in the Colab Notebook. Feel free to jump to **`conducting t-tests using scipy stats`** to conduct the t-test directly.

Step 7: Report findings

- If your p-value < significance level (α), you reject the null hypothesis (fail to reject otherwise)
- We never say we "accept" the hypothesis, why?
  - **"Accepting"** means we are confident that the hypothesis is *correct*.
  - However, remember that p-value **NEVER tells you the probability of hypothesis being tested is true.** It only gives the likelihood of observing the result *conditioned* on the null hypothesis being true.
- Aside from saying we reject / fail to reject the null, it is always a good practice to add context (e.g., rejecting the null ⇒ **there is sufficient evidence** to suggest that the O_3 concentration is above threshold)

Step 7: Report findings

- If your p-value < significance level (α), you reject the null hypothesis (fail to reject otherwise)
- We never say we "accept" the hypothesis, why?
  - **"Accepting"** means we are confident that the hypothesis is *correct*.
  - However, remember that p-value **NEVER tells you the probability of hypothesis being tested is true.** It only gives the likelihood of observing the result *conditioned* on the null hypothesis being true.
- Aside from saying we reject / fail to reject the null, it is always a good practice to add context (e.g., rejecting the null ⇒ **there is sufficient evidence** to suggest that the O_3 concentration is above threshold)

Important language because again, we cannot assert something we cannot quantify!

Step 7: Report findings

`lets_code()` Go to step 7 under #2. T-test example: one-sample t-test in the Colab Notebook. We're not coding this time, but we are asking you to write **your conclusion given the t-test results.**

04

Coding demo: Model-fitting

What is **model fitting?**

STEP 1 define a function that takes in a set of parameters and returns a predicted data set.

STEP 2 define an 'error function' that provides a number representing the difference between your data and the model's prediction for any given set of model parameters. e.g., sum of squared error (SSE)

STEP 3 find the parameters that minimize this difference

Adopted from https://courses.washington.edu/matlab1/ModelFitting.html

What is **model fitting?**

STEP 1 define a function that takes in a set of parameters and returns a predicted data set.

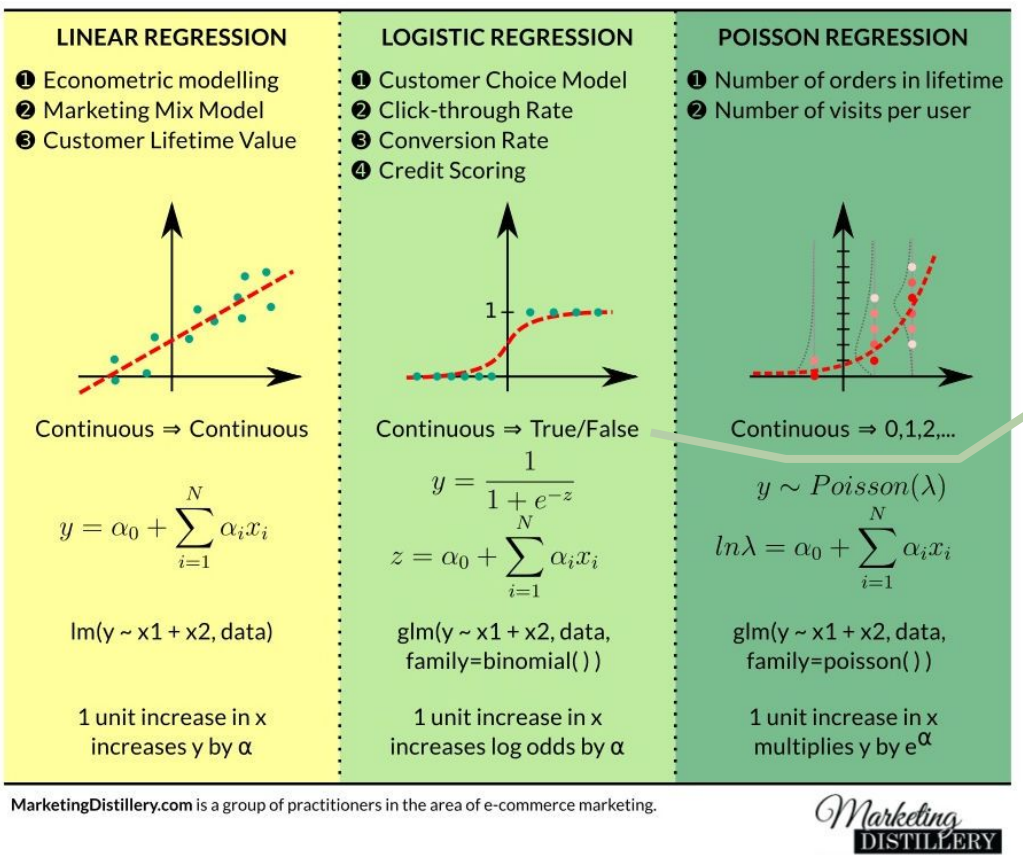
Thanks to libraries such as statsmodels and scikit-learn, we usually only need to do STEP 1!!

STEP 2 define an 'error function' that provides a number representing the difference between your data and the model's prediction for any given set of model parameters. e.g., sum of squared error (SSE)

STEP 3 find the parameters that minimize this difference

Adopted from https://courses.washington.edu/matlab1/ModelFitting.html

Which regression function should I use?



Check out here to see what you are trying to do ⇒

* note that this is a non-exhaustive list, but a good place to start

Can also be binary categories!

Source: https://www.slideshare.net/slideshow/the-three-regressions/41916377

Python libraries for model fitting

NumPy is a great numerical computing tool with functions ranging from finding descriptive stats to manipulating N-dimensional arrays (and much more!)

statsmodels

statsmodels contains various built-in different statistical models for parameter estimation. `statsmodels.formula.api` allows for R-style functions

scikit learn

scikit-learn is a machine learning library that provides several built-in regression models

:)

Before we go....

Before we go...some tips on coding

```
# 1a: subset the dataset, keeping California only, name it 'ca'
ca = df[df['State'] == 'California']

# 1b: subset 'ca' to keep data where year is after 2011 (inclusive)
ca_2011 = ca[ca['Date Local'].dt.year >= 2011]

# 1c: subset 'ca_2011' to keep data where year is before 2013 (inclusive)
ca_2011_2013 = ca_2011[ca_2011['Date Local'].dt.year <= 2013]

# 1d: last but not least, we can just keep the columns we want
ca_2011_2013 = ca_2011_2013[['Date Local', 'O3 1st Max Value']]

# optional but highly recommended: rename columns for easier access
ca_2011_2013 = ca_2011_2013.rename(columns={"Date Local": "date", "O3 1st Max Value": "o3"})

# always a good idea to check the data
display(ca_2011_2013.head())

# save file
ca_2011_2013.to_csv("../data/ca_2011_2013.csv")
```

Document, document, document! The MAIN person you are writing comments for is **YOUR FUTURE SELF!**

Before we go...some tips on coding

```
# 1a: subset the dataset, keeping California only, name it 'ca'
ca = df[df['State'] == 'California']

# 1b: subset 'ca' to keep data where year is after 2011 (inclusive)
ca_2011 = ca[ca['Date Local'].dt.year >= 2011]

# 1c: subset 'ca_2011' to keep data where year is before 2013 (inclusive)
ca_2011_2013 = ca_2011[ca_2011['Date Local'].dt.year <= 2013]

# 1d: last but not least, we can just keep the columns we want
ca_2011_2013 = ca_2011_2013[['Date Local', 'O3 1st Max Value']]

# optional but highly recommended: rename columns for easier access
ca_2011_2013 = ca_2011_2013.rename(columns={"Date Local": "date", "O3 1st Max Value": "o3"})

# always a good idea to check the data
display(ca_2011_2013.head())

# save file
ca_2011_2013.to_csv("../data/ca_2011_2013.csv")
```

When naming variables, make it intuitive (e.g., would it make sense to name the new df “x”?)

Before we go...some tips on coding

```
# 1a: subset the dataset, keeping California only, name it 'ca'
ca = df[df['State'] == 'California']

# 1b: subset 'ca' to keep data where year is after 2011 (inclusive)
ca_2011 = ca[ca['Date Local'].dt.year >= 2011]

# 1c: subset 'ca_2011' to keep data where year is before 2013 (inclusive)
ca_2011_2013 = ca_2011[ca_2011['Date Local'].dt.year <= 2013]

# 1d: last but not least, we can just keep the columns we want
ca_2011_2013 = ca_2011_2013[['Date Local', 'O3 1st Max Value']]

# optional but highly recommended: rename columns for easier access
ca_2011_2013 = ca_2011_2013.rename(columns={"Date Local": "date", "O3 1st Max Value": "o3"})

# always a good idea to check the data
display(ca_2011_2013.head())

# save file
ca_2011_2013.to_csv("../data/ca_2011_2013.csv")
```

Print statements/ Display are your best friend for sanity checks!

Before we go...some tips on coding

```
# 1a: subset the dataset, keeping California only, name it 'ca'
ca = df[df['State'] == 'California']

# 1b: subset 'ca' to keep data where year is after 2011 (inclusive)
ca_2011 = ca[ca['Date Local'].dt.year >= 2011]

# 1c: subset 'ca_2011' to keep data where year is before 2013 (inclusive)
ca_2011_2013 = ca_2011[ca_2011['Date Local'].dt.year <= 2013]

# 1d: last but not least, we can just keep the columns we want
ca_2011_2013 = ca_2011_2013[['Date Local', 'O3 1st Max Value']]

# optional but highly recommended: rename columns for easier access
ca_2011_2013 = ca_2011_2013.rename(columns={"Date Local": "date", "O3 1st Max Value": "o3"})

# always a good idea to check the data
display(ca_2011_2013.head())

# save file
ca_2011_2013.to_csv("../data/ca_2011_2013.csv")
```

After you make edits to a dataframe (e.g., subset, merge, etc), save it as a new file for easy access!

Questions?