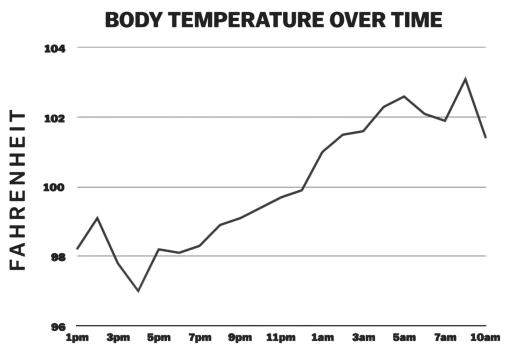
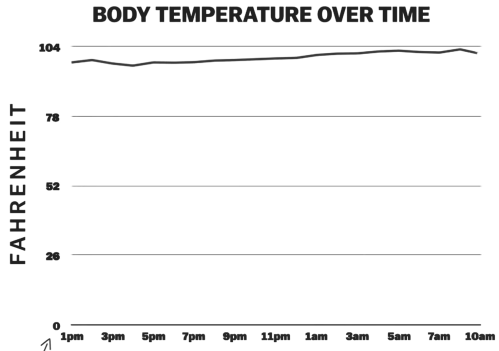
An extremely important aspect of data science is *visualizing* the data in a precise, consistent manner.

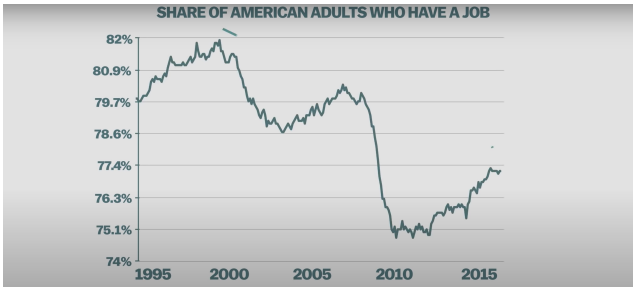
1. Visualizations

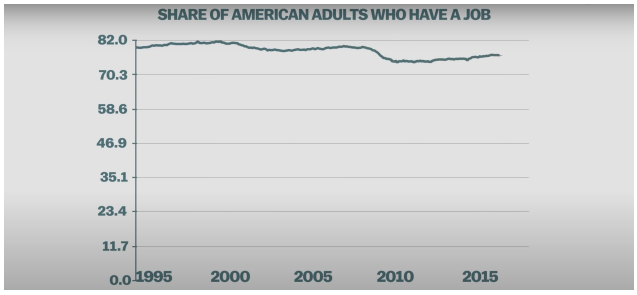
A common discussion people have when visualizing data revolves around the scale of the y-axis on a chart. Many state that every chart with a numerical y-axis ought to have its y-axis start at 0. After watching the video, consider the following questions and discuss:





STARTS AT ZERO





a. In your opinion, when would be a good time to start the y-axis at 0, and when would be a good time to not start it at 0?

b. In general, what are some considerations we want to take into account when making a data visualization?

2. Histograms

The table below shows the distribution of rents paid by students in Boston. The first column consists of ranges of monthly rent, in dollars. Ranges include the left endpoint but not the right. The second column shows the percentage of students who pay rent in each of the ranges.

a. Calculate the heights of the bars for the bins listed in the table, with correct units.

Dollars	Student (%)	Bar Height
250-350	25	
350-550	25	
550-950	25	
950-1350	25	

b. Draw a histogram of the data. You do not have to be precise with your drawing, but try your best! Make sure you label your axes!

c. True or False (Explain): If we combine the [250, 350) and [350, 550) bins together, the height of the new bin would be *greater than* the heights of both of the old bins.

3. Sheng Kee Fridays

Noah's favorite activity to celebrate Fridays is buying pastries at Sheng Kee before class. He stores his purchase data in a table, `pastries`, to keep track of his spending. Each row represents an individual purchase. The first few rows look like this:

item	category	price	satisfaction
Hot Dog Bun	Savory	2.75	8.5
Yudane Milk Bun	Sweet	2.99	9
Summer Romance	Sweet	2.79	10
Pineapple Bun	Sweet	2.45	7.75
Ham and Cheese Croissant	Savory	3.15	7.25

The table has 4 columns:

- **item**: string, name of the pastry
- **category**: string, whether the pastry is sweet or savory
- **price**: float, price of the pastry
- **satisfaction**: float, how satisfied (out of 10) Noah was after eating the pastry

a. Write a line of code to calculate the total amount Noah spent on pastries. Assume all of his pastry purchases are recorded in the table.

b. Write a line of code to calculate the average satisfaction Noah felt after eating sweet pastries.

```
__________(pastries._________(_____________________).column(__________________))
```

c. Noah's budget is getting tight, and he wants to buy pastries that will give him the most satisfaction per dollar. Write lines of code that will help us achieve this.

(i) Create an array that contains each purchase's satisfaction per dollar, then add a new column, "satisfaction per $", to the `pastries` table.
Hint: You can calculate a purchase's satisfaction per dollar by dividing its satisfaction score by its price.

```
score_array = pastries.__________(_____________) / pastries.__________(_____________)

pastries = _________.with_column(_____________________, __________________)
```

(ii) Noah is interested in the pastries he bought in the purchases with the top 3 satisfaction values per dollar. Write code that will output the names of the items from the top 3 purchases as an array.

```
pastries_sorted = pastries._________(_____________________, _____________________)

best_pastries = pastries_sorted.__________(_____________________).column(__________)
```

4. Fa18 Midterm Q2 Modified

A table `insurance` contains one row for each beneficiary that is covered by a particular insurance company:

age	bmi	smoker	region	cost
25	20.8	no	southwest	3208.79
25	30.2	yes	southwest	33900.71
62	32.1	no	northeast	1355.50

... (20198 rows omitted)

The table contains five columns:

- **age**: an int, the age of the beneficiary
- **bmi**: a float, the Body Mass Index (BMI) of the beneficiary
- **smoker**: a string, which indicates whether the beneficiary smokes
- **region**: a string, the region of the United States where the beneficiary lives
- **cost**: a float, the total amount of medical costs that the insurance company paid for this beneficiary last year

In each part below, fill in the blanks of the Python expression to compute the described value. **You must use ONLY the lines provided.** Do not write any code outside the blanks provided. The code in the line should evaluate to the value described.

a. A scatter plot comparing amount paid last year vs BMI (titles are usually written as Y vs. X) for only the beneficiaries whose cost exceeded $25,000 (i.e., with one dot per beneficiary whose cost last year exceeded $25,000).

```
high_cost = _____.____(_____,_____)
_____.____(_____,_____)
```

b. Write a function that takes an age as a parameter, and returns the average BMI of all beneficiaries of that age.

```
def average_bmi(age):
    right_age = insurance.where(_____,_____)
    bmis = right_age.____(_____)
    avg = sum(bmis) / len(bmis)
    _____
```