Nonvisual Data Exploration and Representation (Nonvisual Data Science Workshop #3)

(0:00 - 4:01)

So this is the third workshop in the non-visual data science workshop series, welcome all. This one is, you know, it has kind of almost like two names, but I've kind of just updated it to a slightly more descriptive name, which is exploring and representing data, but originally I called it, which is a more exciting title, talking to your data, which I think is still irrelevant. I'm able to hear, and in this workshop we will be exploring ways to represent different types of data, such as, we'll get to it in a minute, but categorical data, different kinds of data, time data, accounts data, and binary data, all different kinds of data, and we'll be thinking about different ways to represent those non-visually, you know, traditionally our most, you know, in data science in general, often people will resort to visualizations to represent these kinds of data.

We'll be trying to, we'll be doing it non-visually, and we'll also be talking a little bit about how to replace visual, specific visualizations. So we'll kind of be trying to stay primarily in this non-visual realm, but I will speak sometimes to the practical elements of like, well, what if my colleagues ask for a bar chart, what if my colleagues ask for a line chart, something like that. I want to get a few administrative items out of the way.

This is the midpoint in the workshop series. This is the third workshop. This is the last workshop that will be taught by me, and my colleague Sarah Kane, who's a Marshall Fellow at Cambridge in astronomy, will be taking over for the next two sessions to show you all sonification, which I'm very excited about.

So we'll be switching gears a little, but we'll still be drawing on a lot of what we've learned in these first three workshops, and using, you know, IPython and some of these other tools that we have kind of been using all along. Okay, let me go through there. Since it's kind of the midpoint, I want to go a little more in depth in some of the administrative stuff.

There's been some questions that have come up a couple of times from people, so I have a list here. Okay, yeah, so first of all, I'd like to thank those of you, or those, you know, you've kind of maybe come to know them a little bit, the helpers who are hanging out in the chat answering questions. So I just want to thank again Alex, Elizabeth, Sarah, Stephen, and also Paul and Monica, who I think Steve, Paul, and Monica aren't here today, but, you know, they really help things run smoothly and keep things humming along in the chat while people have questions.

So thank you so much to the helpers. I'd also like to thank, there's been some participants who have been sort of very actively contributing, and I'd like to thank Nikhil Vohra, who made some pretty extensive contributions to the first of the workshop series,

the first curriculum for the workshop series, and he did that using GitHub, so he contributed using GitHub. He, you know, GitHub is basically, you can think of it as a social media site for programmers, but where people store their source code, download source code.

It's sort of social media meets infrastructure, and he used features on GitHub to collaborate and contributed to the workshops, and that's something you can do. You can just sort of do open source contributions to the curriculum that is currently hosted on GitHub if that interests you. You can also send ideas or updates or anything like that by email.

(4:02 - 8:27)

I will say that, you know, if you're interested in entering this field, it can be really useful to have a bit of a history of contributions on GitHub, and so this is one way to kind of get started with that, so you can either create an issue, which is basically like you telling me that there's something that could be changed, and then I'll change it, or you can even do what's called a pull request, which is a little bit more complicated, but basically where you write up your own suggestions for how things should be changed, and I can accept it, pull it in, and then once, if you do that, you'll actually be a contributor on the curriculum, just like Nikhil is now, so that's something, an option for you. If you're interested in that, you can drop me a line at Patrick at IotaSchool.com, just if you wanted a little information on how to get started with those kinds of contributions, or just have an interesting view. Okay, and then we also had another contributor, Peter Zigmund, who is following along asynchronously, so probably not in the room, and Peter has been translating the curriculum to Hungarian, which is really exciting, and I'm excited to post that when he's completed it, so that's a very cool contribution as well, and I'd also just like, I think, to thank Chancey here, because I think, Chancey, you've helped me out with some of the ins and outs of using Zoom, and you just pointed out that I have a broken link on the curriculum and everything, so I really do appreciate everyone who's contributed so far.

Let's see, one thing people have asked us about a lot is a mailing list for VI data scientists and people looking to get into data science or learning it, and I think that's a great idea. We have some, a lot of people signed up for this workshop series, and I think it's the basis of a good mailing list, you know, so I am looking into setting up a, or self-hosting a mailing list. If that doesn't go well, I guess I'll use free lists or something, but that, you know, it's a little bit technically involved, but I'm hoping to set it up this week, so since a bunch of people have asked for that, I do think it's a good idea, and I think we're going to go ahead and set that up, so excited for that, and I will send an email to the, to everyone who registered for this workshop when that is live, and also post on some of the main mailing lists, like Python Viz and so on.

Something else people have asked about is, I think there are some frustrations with GitHub, I think maybe rightfully so, and some people would find the raw Markdown files that the curriculum is based on to be more useful. Some people would also kind of prefer to download the videos directly onto their computers, which I totally get, so we're going to try to facilitate that all by making, we're going to basically make sure the home that is a little more user-friendly, we're going to do that toward the end of this workshop series, or right after the series concludes, and we also, I also want to talk with the folks at Pandas to see if they want to host the curriculum, or if we should create a new website for it, but basically there will be a website for it, and there will be, you know, it will make it very easy for you to download the whole curriculum all at once if you prefer to have it on your computer, and also the videos to download directly, okay? So, that's not going to be right away, but it'll be probably shortly after this workshop series concludes in early March. Creating other tutorials, this is something I wanted to mention.

We have a lot of stakeholders, you know, stakeholders is a stupid word, but people who are very involved in various blind and visually impaired organizations registered for this, and people who are parts of other major organizations, whether it be academia, science, the, you know, the blindness community. I'd just like to say, you know, I think this has been a great partnership, and that NumFocus has allowed us to create this very cool curriculum, make it open to the public. I just want, you know, I haven't pestered you guys that much about my own organization, IOTA, but I will say I would love the chance to create more resources like this for the blind community, to teach the blind community some more of these technical skills, and to create new curriculum, you know, workshop series like this, recordings.

(8:28 - 11:30)

If you're part of an organization where you think they might be interested in collaborating to create these kinds of materials, reach out. It can be a, you know, I like, you know, in the best case scenario, it would be for the public. That's kind of where I like to be, but also, you know, I do work with organizations to create trainings specifically for your organization or resources specifically for your organization, too.

So, you know, drop me a line if you think your organization would be interested in working with IOTA, and I'll just say, like, two sentences about IOTA. So, IOTA is consultancy. I'm the head of principal consultant, and we do various things, including creating curriculum like this, teaching workshops, but also we consult on what I would say some tricky intersections between accessibility, writing, you know, like creating a curriculum, and technology, so programming, okay? So, anything that kind of fits into those things, making web applications that have some connection to those things, stuff like that.

That's where we kind of live. So, reach out, and if you have any projects in mind, okay?

And that's probably the only that's probably the only I mean, one more little plug at the end of this whole workshop series, but please, you know, let other people know about this IOTA or the kind of work we do, if it comes up in conversation, because that's very helpful. It allows us to create things like this.

That's just it. That's everything. Okay.

And then I would just finally like to thank Pandas, Numfocus, and Patrick Hoefler, core developer at Pandas or with Pandas, to, you know, for taking a chance on this workshop series and creating these materials, and they're really the reason that, you know, this is all that possible. So, all right. So, that's more administrative stuff than usual, but we'll, you know, it's the midpoint of this workshop series.

Okay. Let's go ahead and get started with the workshops. We are first I'm going to share my screen, and then we'll open up our IPython environment, and we will do a little discussion about different kinds of data.

Okay? So, I'm going to go ahead and share. Okay. I'm opening Anaconda prompt.

I'm pressing the Windows button. I'm going to type Anaconda prompt. I typed A and A, and that was a note, but you may need to type a little more in your case.

(11:30 - 11:43)

Remember, you want Anaconda prompt, not Anaconda navigator. And then, when you open Anaconda prompt, you have your command line. In Windows, we call this command line environment CMD.

(11:44 - 12:52)

There's also another one that some of you have been using called PowerShell, which will more or less work with this, but I find that the CMD environment is a little more stable for applications like IPython, and I'm going to maximize the screen. I'm also going to remove the title bar by pressing alt enter. Okay? So, now I have a nice just the command line.

And then, um, we need to start IPython. So, let's type IPython. That didn't update.

So, that probably means my review is off. So, let's double check that. Okay.

So, it didn't it didn't say the usual version. Let's just do it again. I quit IPython.

I'm typing the word exit to quit. And I'm running IPython again. So, you can hear how it sounds when you start it.

(12:56 - 14:09)

So, you'll hear and there's a whole bunch of other information, but you'll hear the

version. Just a little review and sort of make sure you're all in the right environment. Um, so, what are we doing today? So, a big part of data science is taking data in certain that takes certain forms and then representing it in various ways.

Okay? And we'll talk about the different forms and so on in a second. But I, you know, so, what are the ways you can present data? So, we'll talk if you follow another data science tutorial, chances are the main way that they will represent data is through a visualization. And I looked up visual definition of visualization on Wikipedia, and it was the most general sort of definition of all time.

It was basically like everything in the way they define everything as a visualization, like every picture, everything is a visualization. But basically, my definition is a visualization would be a it's a way to represent some kind of data visually. Okay? And the most common visualizations are things like charts and graphs, though there are other kinds of more complicated visualizations.

(14:11 - 16:12)

And we'll get into some of the, but, you know, the most common charts and graphs would be things like bar charts, pie charts, line graphs, and scatter plots. Okay? There's a whole bunch of other kinds. I would say maybe 15 to 25 common visualization types, and then a more, a bunch more esoteric ones.

But, you know, those are the ones you're going to encounter most frequently in the wild. And that's kind of the ones that people reach for most quickly. Okay? Now, what other ways are there to represent data? Since, you know, you and I and many people on this call don't, don't, you know, maybe we use visualizations, but we don't really benefit from them in the same way that sighted people do.

Maybe we have to use them to share with our colleagues or something like that, but we don't benefit the same way. You can use a sonification. So, we're going to talk about that next week with Sarah's workshop on an introduction to sonification.

Basically, it's a way to represent data through changes in sound, in qualities of sound. So, a quality of sound would be something like the pitch. Sarah might have her own definition, but that's my probably not as good definition.

Other ways to represent data might be text descriptions. So, I mean, we sometimes say alt text in certain contexts, but I would also, the more general term would be a text description. So, maybe the data is in some ways described in natural language, you know? So, if I say to you, oh, the data is two columns, and it's times, one column is times, one column is integers, and the integers are between five and 100, and they represent people's grades.

You know, like that's, you know, that's a way to represent data, right? I'm telling you

what the data is, gives you a general idea of the data. Maybe it's not quite as specific, but, you know. Then there are other ways to represent data, but those are, I would say, are some of the big ones.

(16:12 - 21:08)

The other one that I really want to mention is tactile graphics, which is a really cool direct way to experience forms similar to visualizations, but they're really, you access them through raised surfaces or other kinds of tactile interfaces, such as, you know, dots raised in a digital, you know, some kind of refreshable braille display, and basically it lets you use your hands or, if you don't have hands, another one of your, another part of your body to experience the data manually or through tactician, you know, through feeling. You know, so, you know, for example, you might feel a line chart. I know, I'm chanting not to call you out again, but I know you did a bunch of work with tactile graphics early in the pandemic, and once people are flattening the curve, so, the, you know, and when you have that available to you, you can feel things like a line chart, which is pretty cool, and a very direct way to experience something like people usually use with the visualization.

Now, today we're going to do something completely different, really, and this is, there's not really a word for this. I call it talking to your data, okay, and I would also say it's an interactive data representation is another, maybe, that would be a more fancy way to say it. I kind of like talking to your data, okay, or a data conversation, and basically what that is is we load our data into a manipulatable structure, you know, like our data frame in Pandas, like a series in Pandas, and then we use programming to query the data, poke the data, transform the data, explore the data, and represent the data, do all sorts of things with the data, and clean the data is also another thing that you can do, and do all sorts of stuff with the data, and it has, there is pluses and minuses.

We'll talk at the end about some of the advantages and disadvantages of this approach. I think, in many ways, it's a very exciting approach, and we're going to jump in with that now. So, what are we going to be representing in this workshop? So, let's just really quickly talk about types of data that we'll be working with, okay, and these are not formal types.

Remember, in the first workshop, we talked about Python data types, so those were like things like a string, a list, an integer, which is a number, a float, which is a number, things like that, okay. Those are formalized because there's only a certain number of them. They exist in Python.

They have certain formal ways that you interact with them, okay, through programming. These categories I'm going to explain to you now, that I'm going to say types of data, which sounds a lot like data types. I'm talking about something different.

I'm talking about very general ideas, okay. So, things like saying, if I say time data, that's, there's nothing formalized about that. It can take a lot of different forms.

It could be anything from a string telling you January 6, 1995, with a comma and everything in it, to a number, which is the number of seconds since 1970, which is, surprisingly, that is actually something programmers do. They'll represent time as a number of seconds since a very specific moment in 1970, and that moment is called the epoch, which is pretty wild. Basically, it's almost like computer history started in a particular second in 1978.

It can't from that point. It's one way to represent time. So, that could be an integer, and the other one could be a string, and other ones could be, there's other ways to represent it, more specific data types that are specifically for date and time.

However, when we say time data, we mean data that represent points in time. So, it's a more general concept, okay. What are the ones we're going to work with today? The main ones are numeric, which you've already worked with before in this workshop series.

So, for example, our prices of the Airbnb listings from last time were numeric data. What else is there? There's the other really big one that you'll hear a lot about, and this is as close to a formalized one as you'll get, is categorical data, okay. Categorical data represents a sorting into various categories, or kind of like buckets.

It will sort each entity in a data set into one of a small set of categories, okay. And we will talk, we will make that a little less abstract in a minute when we work with it, okay. There is also time data.

I just discussed that as representation of points in time. And one other useful one to keep an eye out for is descriptive, or if you want to use a fancy word, nominative data. Basically, that's data that is a little more like a unique value.

(21:08 - 22:42)

So, something like someone's email address in a data set, or Airbnb listings that we have, or an ID number, something like that. Those things, they tend to be unique, or a little closer to being unique, and they're usually used to identify things, okay. And for it, it doesn't usually make sense to order it, and it doesn't usually make sense to do things like get a mean or a median.

Like, if you get a mean or a median of ID numbers, that is literally, it's mean, right, okay. So, basically, it's just descriptive of an entity, but not useful in the same way, like number, numeric data, or time data, or something like this, okay. Then I will throw in two other ones, and these are a little bit different because you will most often use these as an intermediate step when you work with data.

And these, but you will sometimes find them in a data set directly, and those are, the ones we're going to talk about today are count, so when you count things up, and then also binary data, so whether something, whether a thing about the entities is true or false. So, for example, if we were to say in our Airbnb data set, it is underscore Brooklyn. We could make a new column called is Brooklyn, and that would be to either say true if the listing was in Brooklyn, and false if it was anywhere else, okay.

That would be binary data, all right. And we'll work with most of these. We might not do so much with descriptive or nominative data in this workshop, but I wanted you to keep an eye out for it.

(22:43 - 26:11)

But we will work with the other ones pretty directly in this workshop. Hopefully, we can get to most of this. There's definitely a lot.

All right. So, let's jump right in and say, let's work with categorical data a little bit, because this is one that I would say it's probably the most, one of the most important to wrap your head around outside of straight numeric data, which we worked with last week and doing things like the mean and the median. And so, knowing how to work with categorical data, categorical data is really critical for data science.

And even, especially when you're like starting an analysis or something like that. So, you may be like, okay, I didn't really get your explanation of categorical data. Let's actually pull some categorical data from our data set, and then we will talk about what categorical data it is.

Again, when we actually have an example in front of us, so it won't be as confusing and abstract. We'll actually have an example in front of us, okay. So, the first thing we'll do is we will pull in our data set.

And it's going to be the same as last week. So, I'll tell you exactly what to type. I will give you instructions for people who participated last week.

I'll give you a little bit of a shortcut, okay. So, you can type, you may be able to do this. df = pandas

And then you can type a little bit of pandas, P-A-N. "P-A-N." And you can, if you press the up arrow, it may fill in our URL and everything that we had before.

So, we fill in a bit of what you wrote before, and then you can press the up arrow. So, it did fill it all. I'll go through that.

Don't worry if you didn't catch that. I'm going to press enter. Sorry, I skipped a step.

You need to first import pandas. So,"I-M-P-O-R-T. Space. P-A-N-D-A-S." P-A-N-D-A-S. Okay.

And then we'll run our DF equals.

And I will go through it again. So, in left. Okay.

It worked for me. So, what you want to type, if you don't have it stored in your history, I tried to tell you a way to access this line from your history from last week. If you don't have it in your history, maybe you're doing this for the first time.

Maybe you can't detect it for some reason. Go ahead and type this. DF space.

That's D as in dog, F as in foxtrot. And then space equals space pandas, P-A-N-D-A-S, dot read underscore CSV. Open parenthesis.

(26:12 - 27:05)

And now you want to pin in a URL as a string. So, we do a double quote. Then do HTTP colon forward slash forward slash bit.ly, B-I-T, period, L-Y, forward slash NYC BNB.

So, that's Bravo November Bravo, you know? New NYC for New York City. And then BNB. Bravo November Bravo.

B as in boy, N as in Nancy. And then finish the double quotes, do a right parenthesis, and then run it. Hopefully, helpers, maybe you can also paste it in so people have the option to copy it.

(27:06 - 28:23)

And it's also present in the written curriculum that we're going to follow along. Okay. Once you have it loaded in, you should be able to do DF.

Just type DF by itself, and you have your data. That's enough. It's staying the number of rows, and there's a whole lot of information.

It's probably read for like 20 minutes if we, there's a ton of information. But basically, it's our big data set. We could do, you know, you could do things like on it like we did last week.

We can do len on it to see how many rows. DF.shape to see how many columns and rows. You could do all the things that we did before.

DF.columns to see the columns. These are things we all did in the last workshop. You could do them, this is the same data as we used last week in this data frame object.

Okay. Now we have our data in front of us. Let's pull out some categorical data and think about it.

Okay. And explore it. So, I'm going to clear my screen because it helps me out

cognitively to clear my screen before I do something new.

(28:24 - 33:38)

So, I did control L, and I heard, you know, in left bracket, five, right bracket. We're entering the fifth line that we've entered so far. Let's do "DF.neig." And I'm going to let it fill it in by pressing tab.

So, go ahead and press tab. Neighbourhood. Because it was filling in the rest of the word neighborhood.

I say to do this because for some reason this data set, it spells, which is fine, but some of us are Americans, we don't use the U in neighborhood. Okay. So, in British English, they say a neighborhood with a U. In American English, we don't do the U. So, this is a point of confusion.

But in this data set, it's spelled the British way. Okay. So, I recommend using the tab and just letting the computer fill it in.

There will be no confusion, no problems. Okay. I recommend to do that whenever you can.

Okay. So, we want to do DF.neighbourhood, and then I'm going to do a period. So, remember, it's neighborhood with U, dot, dot, head, "in left bracket, H, E, A, D, head, left paren, right paren."

So, I did head, left parenthesis, right parenthesis. So, it's got DF.neighborhood.head. Okay. And we should get the first five items from this column.

Okay. The neighborhood column in our Airbnb data set. All right.

We're getting "the first five items. Out left bracket, five, right bracket, colon, zero, Kensington, one, Midtown." Oh, I'm sorry.

I meant to do a... This is fine, actually. So, this is our data for individual neighborhoods. What I actually wanted to use is neighborhood group, which will tell us something a little more specific about this categorical data.

So, if you want, you can type it again. That's probably the easiest way. You can also press up and backspace your answer.

I'll type it again. So, I'll do DF.neighbourhood. I'm letting it fill it in. And then I'm just going to go to underscore and tap in and then fill in group.

So, it's neighbourhood underscore group. DF.neighbourhood underscore group. That's what we want.

And essentially, you'll see in a second what this corresponds to if you're familiar with New York City, the different boroughs in New York City. So, Manhattan, Queens, Staten Island, the Bronx, and Brooklyn. Okay? So, let's... And that's our column.

Let's do head so we only get the first five. So, it should be DF.neighborhood underscore group dot... Open parenthesis. Okay.

So, open parenthesis, close parenthesis. DF.neighborhood underscore group dot head open parenthesis, close parenthesis. You'll notice something.

And this is what I wanted to show you. And this is why the neighborhood wouldn't really help us with this. This is the main characteristic of categorical data is that the values are repeated.

Okay? So, this column consists of a relatively small set of values. So, Manhattan, the Bronx, Queens, Brooklyn, and Staten Island. And they're repeated over and over again.

And when they appear in this column, they tell us that the item in the row, the entity, which in our case, it represents an Airbnb listing from 2019. Okay? Each row in this data set is an Airbnb listing from 2019. And when one of these items appears in this column, it tells us, hey, that listing is from the Bronx.

It's from Queens. It's from Manhattan. Okay? And you'll notice the two here that are appearing most frequently are Manhattan and Brooklyn.

Okay? We can confirm that in a minute. But when you see repeated data like this, when you see repeated information, and it could be numbers too, but if it's a small number of items that are repeated, then that gives you an indication that it's categorical data. Okay? Also, if you think about, hey, this could basically be considered a category.

So, something like neighborhood group, that sounds like a category. And neighborhood sounds like a category. Okay? Then it's going to be categorical data.

It sorts the data, the entities in the data set into various little buckets. Okay? Now, what do you do with categorical data? How do we represent it? How do we explore it? So, the first thing you should do with categorical data typically is to get a set. Okay? So, there's two things you should do right away with categorical data.

(33:38 - 36:05)

To get a set of all the categories and then to count the various items. Okay? These are the two things we're going to do in order. Okay? So, let's do the line we had before, but now we're going to put the we're going to put the function set around all of it.

Okay? So, I'll type it again. Set. "Set."

Open parenthesis. And then df.neighborhood underscore group. That's the line in underscore.

Group. So, it's neighborhood group, underscore group, dot. And let's not do the head.

It doesn't really matter here. So, we could do set df.neighborhood group. Set open parenthesis df.neighborhood group.

Those are our five items. "Bronx, Brooklyn, Staten Island, Manhattan, and Queens." Okay? Those are the five items.

Those are our categories. And we use the set function. It basically says give us the unique values from this data set.

Okay? And what it returns here is a list-like object, which is a set object. But it's very similar to a list. Okay? So, that tells us how many categories we have to work with.

The other thing that I recommend doing with a with categorical data, once you know that it is categorical data, even if you suspect it's categorical data, is to count it. And I would also say that set is very good at telling you, is this categorical data? So, if you run the set function on a column, then you will learn, you will see, oh, is it a small number of categories of repeated values? If it's a small number of items that come back, then that's a strong indication that where you're working with is categorical data. But if it's almost as many items as we have rows in the data set, then it's probably not categorical data.

It's probably something like descriptive or nominative data. If you ran a set on email addresses, it will pretty much be the same number of items as the full data set. Okay? So, let's do let's now let's do a count.

(36:06 - 36:20)

Okay? And we'll talk about counts. So, let's do df.neighborhood group again. df.neighborhood group.

Remember, I'm making a lot of use of tab here. This is where tab is going to come in useful. We're going to be writing longer lines in this session.

(36:22 - 41:49)

So, we're going to be doing I'm going to use tab a lot more. Hopefully, you're getting a little bit used to this environment. I will also say, you know, if I'm moving sometimes a little bit fast in these workshops, you know, which is kind of the nature of some of these more online workshops.

In-person workshops, they tend to go a lot slower, and there's time to work. But, you know, we're kind of doing this for the recording a little bit. And you can review afterward.

And there's a raised hand. We'll get to that in one second. There's a recording, and there's a written curriculum.

So, if you're feeling a little stress at any point in this workshop, just remember that those resources that they're available to you. And Sarah and I are available. We're just an email away.

And there's also the office hours on Thursday. So, a lot of resources available to you if you're feeling a little learning stress. We had a raised hand.

Is that an intentional raised hand? I mean, we can, if you want to get on the mic and ask a question, you're welcome. I hit that button all the time myself, too. So, if you do, you know, if it was a question you wanted to ask, just feel free to get on the mic.

Okay. So, DF.Neighborhood group. And now, so, we had to DF.Neighborhood underscore group.

And now, we're going to use a new method. And we're going to use this method a lot. Okay.

So, try to remember what it's called because we're going to be using it a lot. Okay. But you'll get a little reinforcement as this goes on.

Let's do DF.Neighborhood group. Value underscore counts. So, it's DF.Neighborhood underscore group.

Value underscore counts. Open parenthesis. Close parenthesis.

I kind of hate the name of this method because it's hard to type. It has an underscore in it. It's kind of long.

And I use it a lot. It's one of the methods I use the most when I'm doing a data analysis. Okay.

So, and we're going to use it a lot, too. So, remember that tab is your friends. In this case, it might not be that useful to you.

But, you know. All right. Let's run it and see what we get.

Out left bracket eight right bracket colon "Manhattan 21,661. Brooklyn 20,104." So, okay.

What are we listening to here? We'll zoom in a second. We're hearing a series which has labels. Remember from last week, what we have the series is represented with two columns.

For us, screen reader users, it's that each line has on the left first a label. Okay. Which is

the index.

And on the right, it has a value. In this case, it's a number. Okay.

So, we're listening to a series. The labels are the categories. Okay.

So, Manhattan, Bronx, Brooklyn, et cetera that we got from our set. These are the unique values in this column. Okay.

And then the right most column or the right most item in each line is the value. Okay. Which in this case, because we're performing a count, we created a count, it's the count.

It's how many times each of these items appears in the dataset. Okay. So, we heard Manhattan was something like 21,000 something something.

Brooklyn was 20,000 something something. So, a little bit less. Let's listen to one more.

So, Queens, there's a bit of a drop, 5,660 something. Bronx is another drop. Okay.

So, you're hearing the values. It's pretty intuitive. It tells us how many items in our dataset.

20,000 some odd are Manhattan. 20,000 some odd are Brooklyn. About 5,000 are the Queens and about something like 1,000 are Bronx.

And a very small number presumably are Staten Island. I guess we could look into that. "Staten Island 373."

Only 373. Very few Airbnbs on Staten Island. So, if you want to run an Airbnb, don't open it on Staten Island.

Maybe open it on Manhattan. I'm not in that business. Don't take my advice.

And then at the end, it'll say the D type. Remember, series always tells you the D type. "Name colon neighborhood line group.

dtype colon int." D type int. So, it's integers.

The items in this are integers. Okay. They're counts.

Now, okay. Let's talk about this a little bit. First of all, this is already very useful.

You pretty much if you've ever been familiar with a bar chart, some of us are, you know, you have partial vision. Some of us used to be sighted or more sighted. Maybe you're familiar with visualizations.

You know what a bar chart looks like. I'll just quickly describe it. A bar chart basically it represents data with how long different lines are.

Okay. So, if a line is longer, it has more items in it. Okay.

(41:51 - 44:13)

Basically, what we have here is basically it's a bar chart. Okay. It tells us how many items there are.

And it's almost it's almost like if we were going to take a bar chart created by a sighted data scientist and represent it in some format that is accessible to the visually impaired, we might actually pick a format like this and it actually would be pretty good. You might have your own preferred format, but this one is actually not terrible. Okay.

It's like it's pretty it has a label. It has a number. Each each pair is on a line.

It's actually pretty usable. Okay. So, basically what you have here is a bar chart.

Okay. Now, how is it different from what what could a sighted person do a little more advantageously with a bar chart that you couldn't do with this? I would my answer to that is, first of all, not that much. But if I had to say one thing, it would be that the sighted person might get very intuitively to their mind the difference between the different items.

Okay. So, instead of hearing the or seeing the very specific numbers, they might be like, oh, okay. They might get an insight like, for example, Brooklyn and Manhattan are really close together.

There's a pretty big drop off between those two and Queens and then another big drop off between Queens and the Bronx. Okay. Now, to be honest, you can get that from these numbers pretty easily.

Okay. But let's say let's make it even more like a bar chart and actually and also very similar to a pie chart because a pie chart represents data by it draws a circle. So, if you make a circle with your hand and then different slices on the circle represent their proportion of each item in the data set.

Okay. So, we have about 20 percent of the items are Brooklyn. So, 20,000 divided by some 40 something thousand would be something like 40 something percent.

Okay. Just to do some back of the napkin. So, we would take out a slice of the pie that would be 40 something percent.

That is basically how a pie chart works. Data scientists actually hate pie charts and they look down on people who use pie charts, but in my experience people like many bosses and stuff that I've had like pie charts. So, but, you know, mostly things that you could do with a pie chart you could do with a bar chart and probably better.

(44:14 - 44:33)

But so, basically what we're doing here is we're thinking of ways to replace both pie and bar charts. Okay. But what we're going to do now is even more like a pie chart and it also will give you a little more of that intuition of the differences in the sizes.

Okay. So, let's use the second technique here. So, you already basically have replaced bar charts, I think.

(44:33 - 46:52)

And bar charts are one of the easiest to replace non-visually in my opinion. But let's make it even better. Let's make our representation or non-visual representation even more like intuitive to us.

Okay. Or even more useful. So, let's do this.

First, let's save our counts to a variable. Okay. And I'm going to tell you how to do that with some hot keys.

Okay. If that's okay. So, if you prefer to type it all out, that's fine.

We'll go over that as well. But let's press the up button. That loads in the last line we ran.

So, that will be the one that where we get our counts. Okay. Now, press control A. And you won't hear anything.

Just press control A to go to the beginning of the line. And then type let's just call it neighborhood counts. Okay.

And you can spell neighbourhood any way you want. I'm going to spell it the British way, I guess, because that's closer to the answer. "neighbourhood_counts = " And then I'm going to do a space and equals and a space. So, what I'm doing is I'm saving the last line to a variable. Now, you could also do a really quick way to do this is another cool way to do this is you could type neighborhood underscore counts equals underscore. Because underscore actually represents the last line you put in, which is pretty cool.

And that's one of the contributions Nick Hilvora made when he edited the first in this workshop series is he added a description of that underscore that Python lets you do, which is a pretty cool trick. So, you could do that. Or you could do what I did, which is to press up, move to the beginning of the line with control A, and then type it in.

Okay. Because who wants to type everything over and over again? Certainly not programmers. Programmers are lazy.

In left bracket ten right bracket colon. Okay. So, basically, what we did there is we saved

our previous counts to a variable.

So, we did neighborhood underscore counts equals df.neighborhood underscore group.value underscore counts. Okay. And we didn't want to type it all.

(46:59 - 47:10)

If someone wants to get on the mic for the question, that's fine. I'm hearing raised hands. But maybe actually just type in the chat if you want to raise your hand because it's in maybe accidental hand raising, which is something I very frequently do in meetings.

(47:11 - 49:36)

Because I often want to look at the participant list and then I raise my hand by accident. Okay. So, we now have neighborhood underscore counts.

So, that's our counts that we were just looking at. The five items, each with the label of the borough, you know, Manhattan, Bronx, you know, the neighborhood group, and then the count, the number of times it occurs in the data set. So, that's what we have saved to the variable.

What we're going to do now is neighborhood underscore counts. Did it do it? Space. In left bracket.

Oh, it filled it all in. Neighborhood underscore counts. So, I used the tab to fill it in.

You can type it all out yourself too if you prefer. It's neighborhood underscore counts. That's the variable we created.

And now, what we're going to do is it's a new technique. We're going to get the proportions of the data. Okay.

And they're going to be kind of like decimal points, kind of like fractions that are going to represent how much of the data is each item. Okay. So, we're going to do divided by.

So, it's neighborhood underscore counts divided by. Len for length. "L-E-N.

Len. Left paren.

D-F. Right paren." Neighborhood underscore counts. Then, a space. A forward slash. A space.

Remember, forward slashes divide. And then, len. Open paren.

D-F. Close paren. And so, basically, the len D-F tells us the total number of items in the data set.

So, that's 40 something thousand. That's all the items, all the listings in this data set. And then, so, what this is going to do is going to divide each item in our counts by the total number in the data set.

So, Manhattan is 21,000 something. It's going to then divide that by 49,000 and something. Okay.

And what that will give us is a decimal that will tell us the proportion of the items in the data set that are that. Okay. And remember, we said it would be something like 40 something.

.44 or .43. It'll be something like that. Okay. And that will tell us that 44, 43 percent or whatever of the items are Manhattan.

(49:36 - 51:46)

But anyway, let's run it and see what it sounds like. Okay. "Out left bracket 10 right bracket colon.

Manhattan 0.44." So, you hear that. 0.44. Okay. So, that's about 44 percent of the items in this column are Manhattan.

Okay. So, 44 percent of the items of the Airbnb listings in 2019 were in Manhattan. "3011.

Brooklyn 0.4111." Brooklyn 0.41. Now, you can tell even more intuitively, these are very close to each other. So, 44 percent are Manhattan. Three percent less, 41 percent, something like that, are Brooklyn.

"67. Queens 0.115." Queens 0.11. That's 11 percent. "Bronx 0.022." Bronx 0.02. Only 2 percent.

.313. "Staten Island 0.00s." Oh, Staten Island is tiny. "0.00762." So, less than 1 percent. .7 percent, less than a percent of the items are Staten Island.

Sorry, Staten Island. Okay. And then, so, that's pretty cool, right? So, we're actually getting proportions.

So, now, not only replace the bar chart, we kind of replace the pie chart too, you know, because now the pie chart tells you the proportions of each item, and what this does, this tells you the proportions. Okay. And let's do one more useful thing.

Okay. And I would say sometimes you might just want to leave it there at proportions. And this will tell you very, pretty much tell you the same data that someone using a bar or pie chart would.

And honestly, it's very usable. It's good. Like, it's pretty straightforward to replace a bar or pie chart non-visually.

And even there's actually some slight advantages to this, because sometimes it can be a little hard to read a pie chart if there's a lot of items or so on. It's not a problem. That's another thing about bar and pie charts is that usually sighted people can only put so many items in them, usually a relatively small number, probably not more than like 20, and usually more like 5 or 10.

(51:46 - 52:00)

And a pie chart really does very poorly with anything more than, in most cases, 5, 10 items. That's about as many as you can put in a pie chart, because just reasons, you know, it just doesn't look good. It looks bad, and it's hard to read if there's too many items.

(52:01 - 53:05)

So, actually, you don't really lose out that much with this, because already sighted people can only intuitively grasp a relatively small number of items in a bar or pie chart. And so, we're actually not at that much of a disadvantage with this type of data. Let's do one more thing, which I think is pretty cool.

It might even kind of be in some ways better than a pie chart or bar chart. Well, certainly a pie chart, but maybe in some ways even a little better than a bar chart or show something kind of cool. So, let's do this.

We'll do our neighborhood counts. "in left bracket 11 right bracket colon neighborhood line counts." So, neighborhood underscore counts.

Remember, that's our number of counts of each item that we created. And then let's do dots. And this one always is hard to remember, but I think I've got a grasp on it.

Dot P-C-T. That's short for percent.

I always want to say it should be P-E-R, but whatever. I didn't get to name these. So, they didn't ask my input when they created all this stuff for some reason.

(53:07 - 54:18)

P-C-T underscore. "P-C-T. Line."

Line. Remember, line is underscore. "C-H colon A-N-G-E."

Okay. That was hard to understand. But basically, it's neighborhood underscore counts, our variable, dot P-C-T underscore change.

That's for percent change. And then do an open parenthesis and a close parenthesis. "Left paren.

Right paren." And then let's listen to this. And we will see what we get.

I think this one's pretty cool. "Out left bracket 11 right bracket colon. Out left bracket 11 right bracket colon. Manhattan N-A-N." Okay. It said Manhattan N-A-N. I'll explain that in a minute. Basically, no data. "Brooklyn minus 0.0718." Brooklyn minus 0.07 something something. Okay. So, that is actually, what that's telling you is, between Manhattan and Brooklyn, it decreased by 0.07 percent. Seven percent.

Okay. Remember that these decimals are another way to represent percentages. Okay.

That says it decreased by seven percent. So, from Manhattan to Brooklyn decreased by seven percent. Let's hear the next one.

(54:18 - 56:37)

"880. Queens minus 0.7181." Okay. Queens was minus 0.71. So, Queens is almost a 70 percent drop from the previous.

So, this tells you, what this tells you is the percentage change from the previous item. So, it only changed seven percent between Manhattan and Brooklyn. So, those are pretty similar.

Then, between Brooklyn and Queens, it dropped 70 percent. So, that's a big drop. So, this is telling you, now you're getting exactly what a sighted person would get from the bar chart, that intuition about the change.

Okay. And the groupings. Okay.

That you can get, oh, that changed a lot. Okay. Let's just hear the rest.

"66. Bronx minus 0.807." Bronx, down 80 percent. Okay.

So, an even bigger difference from between Brooklyn and Queens. Okay. And that's something that's hard to get just from the numbers.

Okay. And what we're getting is actually even a little more precise, you know, because you're going to get the exact numbers. You get an intuition with the sighted visualization, but you won't get the exact numbers.

So, actually, I think this is cool. Let's just hear the Staten Island. "7448.

Staten Island minus 0.658." Staten Island down 65 percent. Okay. So, the biggest drop is between Queens and, between Queens and the Bronx.

Okay. But also, another big drop was between Brooklyn and Queens. Okay.

So, this tells us a very, it's a very intuitive way of telling what you would get in a bar graph, which is how big are the changes. Okay. That's what you would get with a bar graph, difference in lines.

So, this is a very, let's say a very adequate or actually pretty good replacement for a bar or pie chart. Okay. Which are, I will say, those are probably the arguably the most used visualizations would be a bar chart.

Maybe you could say a line chart. That would, I would say it's kind of a toss up between bar and line chart are the most commonly used visualizations. We've kind of already sort of have a pretty adequate replacement for the bar chart.

Okay. We are going to move on to the next section. I'm going to do a time check.

(56:37 - 1:00:16)

And then if people maybe have a question. "2 colon 00 PM." We're exactly at the halfway point, which is exactly what I want to do.

Love that. All right. Do people have any questions or do the helpers want to kind of kick up any questions that were coming up a lot before we move on to the next section? Okay.

We already replaced bar and pie charts. Admittedly, those are kind of the most easy to replace non-visually. Okay.

And we're going to kind of get into some harder to replace them, but we already have a win under our belts. Okay. It's been remarkably quiet in the chat.

So, I think anyone has, I know. So, if anyone has questions. You know, I would love, I'm going to do a long wait.

I would love if someone were to ask a question at this time. So, you're doing me a favor. So, if you feel so inclined, just go ahead and turn on your mic and ask a question.

Would sorting the data first, the data sets first give you a better indication of their differences? I would say one thing is that the counts are automatically sorted. Okay. Usually from greatest to least.

Okay. Now, I will say we will be using sorting more before we're through here and sorting is really useful, but in this case, you kind of get the sorting free with the, with the counts. So, it's not really, I don't think necessary.

You can sort the categorical data, but really what you're going to get if you sort it is

something like Manhattan, Manhattan, Manhattan, Manhattan, Manhattan, Manhattan, Manhattan, Manhattan, Manhattan, Manhattan. It'll sort it into alphabetical order and then you'll just get a lot of repeated values. You'll be like the Bronx, the Bronx, the Bronx, the Bronx, the Bronx.

So, I would say sorting the column by itself doesn't really tell you that much. But sorting the counts is very useful and you get that kind of for free. So, usually by default, it gives you from the biggest value to the smallest value.

So, they call that descending order. It gives you descending order. Okay.

Great question. Great question. Thank you for asking it.

Did Sarah, was it Sarah or someone was starting to say something there or? The exact same question. So, nothing further here. Well, thank you.

Is that Prima? Thank you. I really appreciate the question. Okay.

So, if people have other questions, please place them in the chat. You know, I love those long, awkward pauses when I do remote teaching. So, I really, I live for that.

So, I will maybe ask more questions later in the workshop. Okay. Okay.

So, we started out with counting, which is one of the favorite tools in my toolbox. And I also showed you a couple of other cool tricks, which are getting the proportions, those little percentages, and the percent drop or the percent change, which is quite useful. I will say, I'm just going to gesture toward the written tutorial here because we don't have time to cover all of this.

But I anticipated a couple of frequently asked questions at this point. And those are, one, say I want to work with sighted colleagues and I want to make an actual visual bar chart or an actual visual pie chart. How do I do that? Okay.

Now, why, you know, why, no, I mean, I'm just thinking, no. I mean, we live in a sighted world. Last time I checked.

(1:00:18 - 1:00:56)

And, you know, we work with sighted colleagues and sometimes it's useful to actually create the visualization. So, I did give instructions for creating a visualization of our counts data in the written tutorial. So, you can check that out.

Another thing is, question is, can you get actual percentages? And I give instructions for that too. If you don't want the decimals, there's a way to actually get nice looking percentages. It's a little bit involved, not terribly involved, so we're not going to cover it here.

But that information is there in the thing. I had another one, FAQ, but let me check. Max, Patrick, email, I search, creating bar chart, creating percentages.

(1:00:56 - 1:01:33)

Oh, you're right. And also, one useful thing is that I won't cover here, but it's in the tutorial and the written tutorial is you can, a way to skip the step where you divide by len data frame, there's actually a way to skip that step and to do, just change the value counts a little, function a little bit. You add normalize equals true into the parentheses.

And that will let you skip that divide step and it will give you those percentage, those proportions directly. So, check those out in the written tutorial. I'm not going to cover them here, but those are my guesses and things you would want to do with this categorical data at this point.

(1:01:34 - 1:01:55)

And you can always send me email, so I'll add to that. All right. So, I'm going to now show you maybe, in my personal opinion, the next kind of really cool tool in the toolbox, in this data exploration and representation toolbox, which is indexing.

(1:01:56 - 1:03:00)

Another way I would say this is using binary data. Okay. So, what we're going to do here is that we're going to look at our full data set.

So, we have a full data set, which of 47,000 rows. Now, let's say we want to answer a question, not about the full data set, but only about a specific portion of the data. So, for example, the example we're going to use is, say we only want to analyze rooms that are really expensive, so over $1,000.

Okay. Maybe you're deciding if you want to eat the rich or something like that. So, we'll look only at the expensive rooms.

So, what we can do, and this is what I'm going to teach you, is you can make a new data set that is a subset of the old data set for which specific thing is true, for which a specific statement that we make is true. For example, you can say, okay, I want to make a subset of the data where the price of each listing is over $1,000. Okay.

(1:03:00 - 1:04:15)

$1,001 or more. So, we're going to do that now. So, let's clear our screen, clear our minds, control L. So, we've already entered 11 lines of code.

We're on line 12 here. And now let's do our indexing. And I would say this is probably maybe one of my top most used techniques when I'm doing this, is creating these

subsets of the data.

Okay. So, this is a pretty cool one, and very flexible. So, let's do, we're going to do this in two steps.

First, we create a series that is, it's going to be a series containing binary data. Another way of saying that is it's a Boolean series. Okay.

Boolean, say kind of the same as binary. True or false. We're going to create a series that only has true or false values.

And those are going to correspond to what we ask for. So, we're going to say, create a Boolean series where it'll be true if the price for the row is over $1,000. And false if the price of the row is less than $1,000.

(1:04:15 - 1:06:20)

So, that's our first step. And we're going to go and save it to a variable write off. Okay.

So, we'll do, I'll call it expensive underscore bools for Booleans. Okay. I did an underscore there.

Expensive underscore bools. That's going to be our variable name for our Boolean series of true false values.

Bools equals space. And let's do df.price. And this is where we specify what we want. So, I'm going to say greater than $1,000.

I did three zeros there. Zero, zero, zero. So, what we put in is expensive underscore bools equals, for assigning the variable, df.price greater than sign $1,000.

You can leave off the spaces if you're not feeling it. I put the spaces in because that's what they tell you to do in programmer school, which I never went to. Okay.

So, I'm going to press enter. "In left bracket, 13 right bracket colon." Okay.

So, remember when you assign variables, you don't get any output. You just hear the input line again. Remember, the way of thinking about that is that the data that we would normally hear with the screen reader or see, perceive, depending on how you're using it, goes into the variable.

It doesn't come out into our command line environment. That's one way to think about it. So, let's check out that series.

(1:06:21 - 1:08:01)

So, let's do what did I call it? Expensive underscore bools. Expensive underscore bools

dots head. And I just use head a lot because I don't want to hear too much output.

Okay. Remember, head will get you the first five items. So, head is very useful to just get a little bit of the data so you can kind of get a sense of it.

"Out left bracket, 13 right bracket colon. Zero false. One false.

Two false. Three false. Four false."

So, that was pretty boring because they're all false. But most of the listings are not over $1,000. Right? I mean, you know, thank the universe, you know, because, like, you know, I can't afford $1,000 a year.

But we could do DF. So, now we have our Booleans. But now how do we tell if there's actually any trues in it? So, the two things you can do with the Boolean series by itself that I recommend are here's a little trick that I really like.

Okay. I haven't really seen this anywhere. I don't know where I picked this up.

Maybe from a colleague or something. But you could do price or expensive underscore bools. Expensive underscore bools dot mean.

So, get the mean. Get the mean of the Boolean series. And what this will do is it will tell you the percentage there's percent that are true.

(1:08:08 - 1:09:19)

So, 0.004. About half a percent of them are over $1,000. That's still a good number. Okay.

So, we did expensive underscore bools dot mean. And that told us the percentage in a decimal of the items that are true. Okay.

That's a cool trick. So, we do know there are some in there. And now let's find out exactly how many.

So, we can also use remember we said we used our value we had our method value counts. You can also count the true and false values. So, let's do "EXP line bools."

So, expensive underscore bools dot remember using the tab. That's why it's saying it that way. Dot.

Expensive underscore bools dot value underscore counts. "V-A-L-U-E. Value.

Line. C-O-U-N-T-S. S. Left paren.

Right paren." Okay. So, we're using the value counts method on our Boolean series to

count the values.

(1:09:19 - 1:11:42)

"Out left bracket 15 right bracket colon. False 48,656." That's 48,000 something.

That's a lot. That's not all of them. "True 239."

So, we have 239 values in here that are true. That means there's 239 rows in this data set where the price of the room is greater than $1,000.

Okay. Now, let's do the real useful thing, which is we're going to create a new data frame. We're going to call it expensive underscore df.

Expensive data frame. It's a new data frame. We can't just keep using df1, df2, df3.

Don't do that. Don't do that to yourself. Once you go beyond using one data frame, you want to give it a name that's kind of somewhat useful.

We'll call it expensive underscore df. Then, what we'll do, what we'll put in it is only the items where the room is, where only the rows where the price is over $1,000. Okay.

So, we do expensive. Expensive underscore df space equals space. Let's do df, our original data frame. Now, we open square bracket. "Df left bracket." Now, we pass in, we give it our bool, boolean series. "Expensive line bools. Right bracket."

So, we did expensive underscore df equals df open square bracket. Expensive underscore bools, our boolean series variable. Right square bracket.

You can kind of think of this as a fancy, remember we used slicing syntax on our lists in the first tutorial to pull out specific items from a list. It's very similar to that syntax, but we're using our boolean series to tell pandas which rows to pull out. We'll go over that again.

(1:11:43 - 1:14:33)

I'm going to press enter. We saved a variable so we didn't get anything as output. Let's take a look at that variable.

Let's do length on this variable. Expensive line. Df.

So, I'm putting expensive underscore df inside len. 239 items.

We have a new data frame and it's 239 items. You can look at it manually, but now what we have in here is all the items are expensive. Every item in this new data set had a room that was over a thousand dollars.

So, we can do things like, what would be the mean of the prices of these rooms? It would probably be more, but let's just try it out. So, we could do... I did too much.

So, I did expensive underscore df dot price dot mean. So, it's expensive underscore df dot price dot mean, open parenthesis, close parenthesis. So, if you only take the items that are over a thousand dollars, then the mean is really large.

It's like $2,500, which doesn't tell us that much. But you could do other things. Maybe I want to know what neighbourhoods are the most expensive. So, we could do expensive underscore df dot neighborhood. So, I did expensive underscore df, I filled it in, dot neighbourhood. You did the math to fill things in.

So, it's expensive underscore df dot neighborhood. And then now let's do value counts, just to count them up. Because we don't want to go through all that data.

We just want to count it up. So, it's expensive underscore df dot neighborhood. That tells us which neighborhood that each item is in.

(1:14:33 - 1:15:16)

Dot value underscore counts, open parenthesis, close parenthesis. So, what essentially this is telling us is it's going to tell us which neighborhoods have Airbnbs that are listings that are over a thousand dollars. Okay? "File quote lesson Python." Expensive df dot neighborhood. Dot price. Oh, no.

Dot value counts. "Right. Riverdale one.

East Flatbush one. Sheep's Head Bay one. Flushing one. So, this is kind of boring. Fort Green one. So, we actually want to sort this.

(1:15:16 - 1:18:05)

Cypress Hills one." Oh, you know what it's doing. It's printing.

It's because I'm so zoomed in that it prints out stuff later. So, we actually want to use head as well. So, I'm going to press up, colon, and do dot head.

So, actually we're chaining together a lot of things. And we'll go over it one more time. Dot h e a d. Let's hear the output and I'll go over it one more time.

"Head. Right paren." So, this prints the first five items.

We're hearing items from the middle, which is not that good. "Colon. Value line counts left paren. Right paren. Dot head left paren. Right paren.

Out left bracket 21 right bracket colon. Upper west side 23." Upper west side 23.

So, there are 23 items or there are 23 listings that are more expensive than a thousand dollars in the upper west side. That's a pretty seems like a pretty pricey neighborhood. "Midtown 19.

Midtown 19. West village 14." West village 14. Okay. Okay. So, those are very expensive neighborhoods.

Any other ones? "Upper east side 13. Chelsea 13." Those all sound like very expensive neighborhoods in New York. Okay. So, what we did there was we have our new data frame that we created that has rooms that are only over the $1,000 price point, but it has all the other data in it, right? So, it's like a new data frame, but the only rows in it are items that are over the prices over a certain amount, but it still has all the other data in it. So, we can pull out the names.

We can pull out other items. We can pull out the neighborhood. We can pull out the neighborhood group from this new data frame.

So, we just then do what we did before, which is pull out some categorical data. So, we pull out the neighborhood, then we count it, value counts, and then I did head because we had, you know, it was printing out too many things. Okay.

So, we just pulled out the first five items, and it told us the five most expensive neighborhoods. Okay. So, that was expensive underscore df dot neighborhood dot value counts, count the data, dot head.

Okay. And that tells us the five neighborhoods that have the most rooms over $1,000 in the data. You can see this is a very flexible method.

Anything you can think about that you could pose as a question about something greater than, something less than, you can make a new data set out of it. Okay. And then explore only that data set.

Okay. So, it's a very flexible tool in your toolbox. You make subsets of your data.

Okay. I'm going to do a quick little time check because there's one thing, something I would like to show you with this, but we may not have time for it. I might just have to gesture to it.

(1:18:11 - 1:21:17)

222. We can very quickly go through it. I think we might have time for it.

So, I live in New York, as we talked about, maybe briefly before, and I grew up in Queens, here in Queens, and I live here in Queens. Okay. And I say, actually, I didn't get very far in life because I grew up in Queens and I live in Queens.

But, literally, in terms of distance. But, you know, so, but I like my neighborhood. It's called Woodside.

It's here in Queens. And I want to know some stuff about, let's say I want to know the average price of an Airbnb in Woodside. Okay.

In my neighborhood. So, we'll use something a little bit, we're using the exact same technique. We're going to create a Boolean series where if it's, if the neighborhood is Woodside, it will be true.

The row will be true. And if it's not Woodside, it'll be false. Then we're going to use that Boolean series to create a new data frame.

And the data frame will only be items in Woodside. Okay. Only be items in the neighborhood Woodside.

Okay. So, instead of working with numeric data, instead of using numeric data to create a subset, you know, greater than, less than, whatever, like we did with the expensive data frame, we're going to actually use categorical data. So, we'll say the data has to be in the, in terms of the neighborhood as a categorical data.

And then we'll say, okay, the data has to be in Woodside. Okay. So, it has to be in a specific category in the neighborhood column, which is Woodside.

Okay. And you can tell I'm in Queens because someone's driving by. Hopefully you guys don't hear that.

Maybe you do. Let's do control L to clear our screen. Okay.

I guess I'll do the math. Focus is the worst thing in computers. Okay.

So, I'm doing DF, or I'm sorry, I'm doing, I'm going to call this Woodside underscore bools. I'm using the same technique. I create a Boolean series.

I'm going to call it Woodside bools. I'm going to go a little faster with this because we already did this. I'm just showing you a slightly different technique.

Woodside bools equals DF.neighbourhood equals equals is equal to, and then I'll use the string, quote, "W-O-O-D-S-I-D-E." I have to use a capital letter for Woodside because that's how the data is constituted, I think, quote. Okay.

So, we have our Boolean series. I'm just going to double check it by doing the mean to make sure there's something in it so that it works. So, Woodside underscore bools dot mean.

(1:21:25 - 1:21:42)

So, there is something in it. It wasn't just a zero. So, it's not a lot of the data.

It's a very small percentage of the data, but it's there. And then we're immediately going to create a new data frame from our Booleans. So, I'll say Woodside underscore DF.

(1:21:47 - 1:22:29)

DF. Say a new data frame. If you want, you can just sit back and relax during this part.

You kind of already did this. And then equals, Woodside DF equals, and then DF, open parenthesis, no, open square bracket, DF, and we pass in our Woodside DF. This is exactly what we did before.

You know, to create the new series, we do DF, open the square bracket, and we pass in our Booleans. Okay? And then it pulls out all the rows for which a value is true. Okay? For which, in this case, for which the neighborhood was equal to Woodside.

(1:22:30 - 1:22:49)

So, Woodside underscore bools. So, here I have Woodside underscore DF equals DF square bracket, Woodside underscore bools. Okay? And what this does is it creates a new data set.

(1:22:49 - 1:23:08)

It's a subset of our data where only the neighborhoods that are Woodside, or only the items in the neighborhood column that are equal to Woodside are put into the new data frame. Okay? That's a terrible way of saying it. It's a subset of the data for which the neighborhood is Woodside.

(1:23:11 - 1:24:40)

So, now I have a new variable, Woodside DF. Okay? And let's just quickly, we'll get the mean price. So, we'll do Woodside underscore DF.

This is just like we did in the last workshop. So, remember, if you kind of remember from the last time, I think if we do it again, "df dot price dot mean()." The mean price in the full data set was.

So, in Woodside, it's only like $80 or $85, which is pretty cheap. It's relatively cheap. You know, so, you know, come to Woodside, I guess.

And then one other fun thing is you could pull out the, don't follow along with this, but I'm going to show you the flexibility of this approach. When I was working with the data, just preparing this tutorial, I noticed that there's some very funny room names for the cheap rooms in Woodside. So, just for a laugh, I'll pull that out.

(1:24:40 - 1:25:41)

So, you don't have to follow along with this if you don't want to. But what I'll do is I'll do Woodside DF. In fact, I'm going to sort my new data frame on price.

So, I'm going to get cheap and expensive. So, I'm going to do Woodside DF dot sort values. Okay.

And then I'm going to do dot name. So, it's Woodside DF, our variable, dot sort underscore values. And then into that, I give price.

So, I basically say sort the data frame on price from, I think it's from cheapest to most expensive. And then I did another dot, and I'm going to say I want the names. Okay.

(1:25:43 - 1:26:10)

The name column. And then I'll just get the top most, the head. Don't worry too much about following along with this.

I'm just doing this to entertain you because they're funny. But that is useful. So, basically, I take my new data frame, which is only Woodside, my neighborhood.

Then I'm going to sort by price. Okay. Sort the whole data frame by price.

(1:26:11 - 2:05:20)

Then I'm going to pull out the top, the five first results, which I believe will be the cheapest. ... ... So, there's a funny one. It has some emoji, which isn't printing out correctly, I guess. they're not hearing it and it says wow exclamation mark cozy exclamation mark so that's one of the rooms and then another one of the rooms was it only has the word you in it like you like letter u can get to Manhattan and that's literally the cheapest room and with that it was $28 when I looked at it and it was you capital letter can get to Manhattan and you know or you can use a certain number of train lines I thought that was kind of funny and then another room says just wow cozy and that costs like 30 bucks or something like that so the ones that are the cheapest are kind of the most eccentric and I thought that was kind of a little bit entertaining okay and that kind of also shows the flexibility of combining some of these things so what we did was we created a data set that was only Woodside then we sorted on the price to get the cheapest to most expensive and then we pulled out the names from the data frame and then we got the first five which is the cheap five cheapest rooms you can also do doctail and get the last which would be room five rooms would be the most expensive okay so once you're gonna get a little bit used to these methods you can start mixing and matching and it becomes a very flexible okay so we now have about half an hour I believe okay "2 colon 32 p.m." exactly okay so what I want to introduce is one more concept which is correlation and then I may just kind of gesture toward how you would

work with create line charts as well but we may not have time to fully engage with that but there are materials in the written curriculum as well so we might just have to do that a little bit of an abbreviated version of the of creating a line chart it's a lot to take in anyway so I think we're kind of covering a lot but okay so we we've learned indexing that's creating subsets of the data I would say if you're gonna learn two techniques for data for exploring data for representing data then the two you want to learn or make a count that's incredibly useful to just do a count of a categorical of categorical data will tell you a lot the second thing I'll tell you is do that learn to do that indexing learn to use those boolean series which will allow you to create subsets of the data they also allow you to create answer specific questions so you can learn oh what our question well here was what was the proportion of the rooms that are over a thousand dollars and the answer was it's about half a percent you know we got the boolean series and then we do dot mean and it tells us okay how many of these items is is this proposition true and that's a very flexible we can also say well how many rooms are less than $30 how many rooms are under $100 or we could say things like how many rooms have the minimum night greater than that you have to stay greater than a week 30 than seven days anything that's a number you can use that greater or less than and answer questions about it say what's the proportion of rooms for which you know the other let me think of other numeric items that the number of rooms for which the minimum nights you can stay is less than a week you know and it will tell you the percentage that's very useful and once you create a subset of the data a new data frame based on that you can answer other questions like you can say you know of the rooms where you can only stay for one night what's the mean price and is it greater than or less than the full data set or greater and less than rooms where you can stay for more than one these are you can answer very specific questions with these two tools only two tools in counting and indexing these are I would say if you're gonna learn two things you can get very far with just those two tools okay very flexible and you also see it combined well with sorting getting the head and the head and the tail those are the old and we're really only using a small number of methods here to do some very flexible stuff okay you learn those three or four methods count values head tail sort sort values and and then to get that indexing method that I showed you where you use greater than less than or equal equal to and then that's a small number of methods that lets you do a lot okay so let's do something a little more data sciencey and this is a you know we're kind of coming up on the end this probably the last thing we fully we managed to fully engage with we'll kind of just put a line charts at the end and we do have some written curriculum but I think we may not be able to get to it so let's clear our clear our minds clear our screens control L "in left bracket 30 right bracket colon" and we're up to 30 we've entered 30 lines of Python we're up to line 30 here and what we're gonna talk about now is we're gonna try to replace another common visualization type a little bit but we it's not it's gonna be a little more challenging it's not quite as easy to replace but it is possible to get kind of a certain percentage of the way there and that is that what we're gonna learn to do is we're going to learn to correlate to if we have two variables so we have two columns

each with numeric data then we what we're gonna learn to do is find a correlation between those two columns okay and what a correlation is it's a it shows a relationship between the two columns between the two variables okay and what do we mean by that and you know the fancy way to say this it's a linear relationship linear relationship basically means you could draw a line to show the relationship or you can get a number which is would be the slope of a line and then that's fancy instead the simple way to say it is it say you have two variables let's just imagine we have ice cream sales and temperature those are two variables we have the temperature in degrees Fahrenheit let's say I mean I'm in New York and a Fahrenheit is ridiculous but let's say we have temperature in degrees Fahrenheit and we have ice cream sales okay sales of ice cream per day and then at certain temperatures how much do the sales but how does that affect sales so we could use a correlation and that would suggest what is the relationship between the two variables the linear relationship so if one goes up does another one go up down or doesn't nothing happens that's basically what we're talking about if one variable goes up then what happens to the other one does it go up down or does nothing happen that's the direction of the relationship and then also how strong is the relationship so does it go up a lot does it go up a little does it go down a lot does it go down a little okay and that is basically what a correlation okay and what we're specifically going to talk about is called a Pearson correlation but it's the most common correlation type so let's not worry too much about that and the way we represent this is in a number from negative 1 to 1 okay so if the two numbers are if one number goes up the other one goes up we call that a positive correlation and we that number is and then the number that correlation number will be positive if if say one number goes up the other one goes down then the number or correlation number will be negative okay and then if there's no relationship then the number will be pretty close to 0 okay the correlation number will be pretty close to 0 okay so negative 1 means you have a negative relationship that the variables go in different directions 0 means the variables aren't related really they there's no association or close to 0 and then positive number means that there's some kind of relationship if one goes up the other one goes up and the closer the number is to 1 the stronger the relationship so if if a number goes up by 1 then the other one goes up by a fixed amount say if one number goes up by 1 then the other one goes up by 5 and that will be a relationship of 1 okay it's perfectly correlated okay and so 1 is perfectly correlated they're basically the same and 1 goes up the other one goes up straightforwardly linearly and then if negative 1 one goes up the other one will go down straightforwardly linearly okay so those are that's what we're gonna calculate and luckily pandas makes this in terms of programming this is pretty easy the hard part is just understanding what you're doing what the two what the number means okay basically just what I just explained to you that there's a direction positive negative or no relationship and that there's a strength whether it's strongly correlated or weakly correlated and then that number one number between negative 1 and 1 tells you the story of the correlation okay so let's correlate two variables in our data set or two columns and two numeric we only have a few numeric columns let's do number of

reviews and reviews per month those are two columns we have in this data set so reviews per month is how many reviews come in per month for that listing and the number of reviews is the total number of reviews okay so let's do DF dot DF number I'm gonna let it fill it in so number "line of line reviews" so it's DF dot number underscore of underscore reviews that's our column let's do dot C O R R dot correlation dot C O R R dot corr dot correlation that's short for correlation okay so it's DF dot number underscore of underscore reviews dot C O R R for correlation open parenthesis "C O R R left paren" dot and then now we pass in the other column so it's gonna be DF dot DF dot and then we're gonna do reviews per month "R E V I E W review per line month" okay and then close the parenthesis "right paren" so it's DF dot core sorry DF dot number underscore of underscore reviews dot core open parenthesis DF dot reviews underscore per underscore month right paren okay and those this will correlate the two columns and just give us one number between negative one and one "out left bracket 30 right bracket colon 0.5 4 9 8 6 7 5" 0.5 4 so that basically means first of all it's a positive number that means that the when one number goes up the other one goes up by some amount okay so that means there is a relationship between these two variables which kind of makes sense that the the number of reviews you have total should in some way correlate with the amount of reviews reviews you get per month because it's kind of obvious if you have a lot of if a if a listing has a lot of reviews they must have some amount you know of reviews per month it's probably going to be high and in this case we are a number it's positive so they have a positive relationship that one goes up the other one goes up by some amount and 0.54 that's a pretty strong relationship i would call that moderate to strong if it's somewhere around that middle mark like 0.4 0.5 you'd maybe call that moderately correlated and then if it's above 0.5 which is a little bit in this case you start calling it a strong correlation these are all kind of subjective but but you know it's fairly well agreed upon and then if it was say 0.3 0.2 0.1 that would be weakly correlated not very strongly correlated okay so in this case it is correlated um you know you might ask why aren't these numbers perfectly correlated if you know since you know the number of reviews you have total does depend on the number of reviews they have per month why are they perfectly correlated um the answer to that is that there's another variable at play here um uh many possibly others but the one i can think of is how long has the listing been okay and if the listing's been around a long time versus a short time then then that will affect how much the number of reviews has how much effect the number of reviews has on the um per month has on the total number of reviews okay um so uh so it's not the only variable that determines the other okay they're not perfectly correlated they're only somewhat correlated which makes a kind of intuitive sense i think that these would be correlated in some way um so we could do this we could try this on one other i'll just run quickly on one other variable let's try to see if the number of reviews and the price are correlated and you know i could see there being a maybe i could make a hypothesis i could say maybe rooms that are reviewed a lot have a higher price because who knows you know maybe they're popular um that's a guess but let's see if it's true or not we'll do df dot price dot c o r r for correlation ...open parenthesis

and then let's pass the other one and we'll do df dot number of reviews "line of line colon reviews right paren" so we have df dot price dot c o r r for correlation left parenthesis df dot number underscore of underscore reviews right paren okay let's see if these are correlated remember my hypothesis is maybe they are kind of correlated because i don't know maybe that if you have a lot of reviews you're popular maybe you can charge "out left bracket 31 right bracket colon minus 0.047" okay so i would say it's negative 0.047 okay so i would say that no my hypothesis is totally wrong and in this case it's first of all it's a negative number so it's if anything it's negatively correlated if one goes up the other one goes down but really what this is is uncorrelated it's very close to zero it's a very small number and so basically what these are is they're unrelated that the number of reviews doesn't seem to really affect the price in terms of this data set okay which is also another pretty interesting you know factoid or whatever okay so that's correlation it's um it's uh we're very commonly used like uh you know statistical method in data science um in general when people want to show correlations you'll often use uh they'll often use a visualization called a scatterplot and the scatterplot basically puts dots on a say you have a flat plane so you just imagine just the flat table and then imagine you take a pencil and put a whole bunch of dots on the table on the on the that surface and that's kind of the scatterplot and what a scatterplot lets you do if you're sighted or you can otherwise access it is that it shows you patterns like clusters and that will usually often also show you whether data is negatively correlated just exactly what we have here negatively correlated uncorrelated or positively correlated it may also show you useful things like how many outliers you have like if there's certain values that are way outside of the norm that we can't we haven't done we we haven't learned a way to do that with this non-visually okay so that's something a scatterplot can tell you that this won't but what this will tell you is the relationship in the data and that's about half of what a scatterplot or maybe more than half half to 70 percent of what a scatterplot will tell you is that relationship between the two very okay so we can get that from just making this correlation what it doesn't tell you is stuff about outliers or certain other kinds of more subtle relationships and clustering you can get those it's just quite complicated it's out of scope in this tutorial so you would want to look at things like gaussian functions which would allow you to it's a little bit mathematical but basically you can do you can look for outliers you could also just kind of look for really big numbers in the data set through methods that we've learned already which might tell you a certain amount of that okay but if you want to get fancy and kind of more formally look for outliers you can use gaussian functions okay which is i would say harder than what we've been doing here so but just know that it exists for non-div for our non-visual use okay um so i would say you know bar pie charts were a strong win with the non-visual approaches you can really strong pretty pretty well well replace a bar pie chart with non-visual methods um scatterplots definitely more difficult unfortunately scatterplots are also maybe you know Chancey is here she does a lot of tactile graphics scatterplots are also pretty hard are harder to get through tactician through feel on a tactile graphic line charts and stuff definitely work better i think and other kinds of

visualizations um but uh you know scatterplots are a little tricky um you might get some use out of it but um okay so we only have a few more minutes i wanted to do time "2:49 p.m" we have 10 minutes i'm just gonna quickly gesture toward how you could do time because we have a very limited amount of time and i want to kind of give you a little bit of the philosophy of all of this interaction at the end um but um essentially what you want to do in time is um and i will point you toward the written tutorial um but essentially what you want to do is um that there's special a special function in pandas that will allow you to take a time column and convert it into a form that you can use so it looks something like this pandas as pandas dot dot um pandas dot two underscore date time date time and then you can pass in a column and it will make it into a special format that will allow you to do time-based things with it um so it would be df the only time column on this data set is called last review when the last review came in "last_review" so here i did uh i did df uh our pandas dot two date time two underscore date time open paren and then i passed in our df dot last review column that was a it's time data um i'm going quickly with this i'm just going to give you a general idea of how to do it i wouldn't follow along now if you're if you're doing this live um and i'm going to save it to a variable i'm going to call the variable um i'll just call it date datess so i have now a column that's date--a lot of the a lot of them are empty but um uh some of them are not hopefully and then you can then use that you can do year uh date dot dt dot year "out left bracket 34 right bracket colon" and then we can count that in left bracket 30 dot value "2018.06" so now we have to we're hearing the output there let's run it again because "out left bracket 36 right bracket colon 2017.03 205 2016.0 2700" and essentially this is telling us how many items from each year um there are in the data set okay so this is kind of fairly useful time data and then you could um just sort this and then get an idea okay over time how how does something change okay this would be more useful if it was when the the listing was first posted or something like that but if you look at this and i i encourage you to take a look at the written curriculum that goes into more detail on how to perform this operation but basically what we're doing here is we're turning a column into a specific format that allows us to manipulate it in terms of time then we're using that to pull out the year then we're using our familiar value counts to count how many of each year we have and then if you sort it you'll get a sense of how things change over time and you can actually even add in that percent change um function uh or method that we used at the beginning and you'll get something very similar to a line chart where you get the idea of the percent went up the percent went you know the percent is going up by a certain amount per year or down a certain amount per year which is exactly what a line chart tells you um unfortunately i don't want to keep you guys longer than you you're you know are two hours um so if you're interested in how to replace these line charts i i recommend looking at the written curriculum uh and feel free to send me questions and stuff as well okay um i will say uh yeah so and that percent change it really does actually give you a fairly once you this is harder to make but once you have it if you if you you know get an account of all the different years um then uh then get the percent change you do get a sense of exactly

what you would get with a line chart which is how much does something how steep is the change how radical is the change between these different time periods the different years okay um this making these uh kinds of visualizations of over of time can get pretty uh tricky non-visually but once you make them you can get a pretty strong idea of the change in the data set okay um they're not as easy to make as the bar charts in terms of programming but but you do get a very strong sense just like you would with a line chart of the change over time um it's maybe not quite as direct or intuitive as what um sarah will show you uh in the coming two weeks with sonifications where you'll hear the change with the pitch but it is pretty okay it's pretty it's it's not terrible um it's it's easier to replace than the scatterplot um okay i'm clearing my screen in left we're pretty much at the end of the um of this workshop um i'll just want to give you a little bit of uh i want to point out some of the the things to keep in mind um and give you kind of the usual uh pep talk for that for the non-visual data analysis first of all is that um uh is that what was the the basic methodology we have here which is to create an interactive to to create interactive um objects that we can then manipulate so uh you know data frames series and then to ask for very specific things from them and to get very specific results back it really when you get fairly adept at it when you get even a little adept at it or when you get fairly adept at it it really feels like having a conversation with your data it really feels like talking to your data and you can get a very strong idea of the relationships the contents uh and so on uh of the data sets through this method um and i would say possibly even if you get quite good at it it's it you really honestly you won't be missing the visualizations that much you can really get a very strong sense of the data set through these methods um now are there downsides uh obviously uh i would say in some cases um the the main downside the main downsides i would say are if you're working with sighted colleagues um and you want to share uh a representation of the data i will say sighted people love charts you know um and you know they're they're often persuaded by them and so on so you if you want to work with sighted people it's helpful to actually make the the visualization if you can um you know it's obviously it's hard as an as a as a blind visual impaired person to make a visualization but if you're working with sighted people there's advantages um the other thing so if you're working with people that's one um important important reason that this this is not a great approach um specifically um the other one is uh uh the other disadvantage is that you do need to kind of know what you're doing um so i'm not gonna i'm trying to say this in a more diplomatic way but like um i would say often visualizations can be a little bit of a kludge um that people kind of just do them out of habit and then they look at the data and gives them some kind of general idea of relationships and so on um uh and you know a good visualization is a good visualization but often i feel like sometimes visualizations are a slight crutch for data scientists and unfortunately if you want to do things this way that's not a visual way you will kind of maybe need to know what you're doing in terms of the types of data you're working with how you're transforming them and how you're representing them in a more specific way you know doing things like the way we divided the um the data to get the proportions or to get the percent change they

do kind of or to get other useful things like the correlation um there's things we didn't even get into but um there's things like variance standard deviation and so on you really do need to know what those mean in order to make them useful to you um so you may need to get uh you know get a think about the math a little bit more or to think about the types of data that you have a little bit more um so unfortunately there's no real shortcuts for the non-visual stuff but if you when you get there it's a very powerful way to interact with data um and then i guess i'll leave you on um uh i mean just i wanted one say one more i'm looking at my notes you might you might need to actually know what you're doing you might need to actually know what you're doing you might you might need to act you might disadvantage not as immediate not as good for collaboration with sided or disadvantage not as immediate so i was good at collaboration some types of insights are harder to access or take longer we're going to make a we're going to make tonification for the next two workshops and those are much more direct um ways to experience the data okay um and they're much more shareable in a lot of ways okay um and i do think sonifications are going to be et cetera we'll talk more about them but they're going to be much bigger than they are in the coming years um and i do feel like they will help us out in these areas for when we're trying to create stuff to share with side of colleagues because what i've found is that while side of colleagues they maybe won't look want to look at these kind of like you know non-visual representations we have quite as much they really do often do like a sonification so some hope so i have high hopes for sonifications in the coming years i think it's good they're going to be bigger we're going to do more of them in the next two workshops as sarah takes over and i'll just say i've really enjoyed teaching these three workshops um it's kind of a privilege to teach workshops specifically for my fellow blind di people um i don't get that opportunity that often usually i teach sighted um people um and and it's quite nice to to teach you all how i actually do data science which is this in this non-visual mode so um so i will will end there except for questions and answers so thank you thank you all okay i'll leave the i will leave the recording on for questions and answers uh and then we will also then turn off the recording um if people you know if people want to ask off mic okay so we'll leave the recording on for now um and uh if anyone wants to get on the mic and ask a question this is the time or if the helpers have questions from the chat the only thing we've seen recurrent in the chat is a couple people having trouble um loading the data frame um from the url yeah um and we think upon upon some examination we believe that for these people it's probably either a problem with their network so it took too long to load because their internet's too slow and ipython timed out or maybe they have some sort of uh like proxy enabled so we encourage people to go look in their network settings um and make sure you don't have some like barrier like firewall i suppose between you and whatever you're downloading so um but if you're still having that issue we do encourage you to maybe come to office hours because this is not exactly an issue with the code so much as something something funky going on with your computer and one other additional thing i can so in the tutorial when i update the tutorial i will add a link to download the csv and then you can try to place the csv in your

home folder and import it that way the url is definitely a lot easier if you can access it okay um i would say you know the most likely if you're having a problem with the url 90 it's a problem with the the way you type the link um or in the copying and pasting uh i would say that's the most likely reason or the most common reason when i and and this is also when i do it with excited workshop participants it's that's almost always the reason and then the next reason is like sarah said network issues um now if you're on the zoom meeting chances are you know you're on the internet so hopefully you know you check that box i will say that in the last year or so github is getting more fussy with what ips they accept requests from and so on but they should accept for most um i am often on my um you know very suspicious vpn server and you know whatever it is rotterdam or something like that that a lot of websites don't like and cloudflare doesn't like and github still lets me look at these pages and import the data sets and all this kind of stuff so i would say triple quadruple and quintuple check your url remember it's http colon slash slash um bit.ly forward slash nyc bnb and i would also try both http and https just try both of those okay i did http because i think it's the most likely to work but you could also try https um as a possibility okay those are the things i've tried um okay i would hope someone asked a question just make me feel let me feel happy i love to hear your you know i love teaching a person and hearing people's voices and hearing the noise and questions and everything and that's you asking questions on the mic is the closest thing hey patrick it's marco hey marco how's it going hey it's going all right thanks again for this whole series here um i put i put this question in the chat but i just wanted to see if you your take on it uh so you know we talked a lot about the data sets that we have and um you know kind of understanding it from a text point of view within the terminal but i could take this airbnb data set you know grab a subset of something that i want to build as a bar chart or a line chart and then as long as i gain access to those values within a normal python data type like a dictionary um i could then funnel those values into like something that's going to write svg code for me to so i can actually create a tactile graphic out of that um so i was wondering like an elegant way of taking like a subset like if we grab price and the name i guess of the of you know the listing how to take that and create a dictionary straight from a subset and i think you know helper posted i guess um pandas has its own two underscore dicts uh or there's like there's a method that does that for you which is pretty awesome i was just wondering your thoughts on that well would it help to create an svg file directly like like an svg file of what side of people would would have would that be helpful or i mean i would be able to do that because that's that's right the whole thing that i've been working on is building svgs so you can do that already or do you want me to show you how to do that no i know how to do that yeah i teach people how to do that but it's just you want what you want is a representation which would be something like the lengths of the um like can you describe how the dictionary should look and then i could kind of maybe advise because the only thing is you know i've you've worked with svgs but in a very lazy way so you know okay yeah i mean like to let's say to create like a line chart you would just use the polyline svg shape and then you would the points um you would have to grab the you

have to kind of set it set up the axes of the canvas um take the data that you're getting from the data frame and uh you're kind of piping that in but then you have to massage the data in a certain way to get it to scale properly you probably need to we'll need to normalize it um i would say basically what you're talking about is something kind of fairly um it's it either has something existing for it in which case you know i probably wind up googling around the same as you and i encourage you actually to maybe add send me drop me an email and i'll do a quick google or to someone on the chat on the call no about this so um geopandas does this geopandas has a column for polygons and svgs um so the geospatial geojson community has already done some of these things um so if you go yeah if you go and check out geopandas you'll get some of this stuff and um alternatively um actually yeah no stick with that um that's probably the cheapest way to get svg um thank you i just want to say this is um the person who just answered there is tony fast who you know is a contributor to project jupiter and i had the pleasure of working with tony last year on a project with space telescope science institute to make their notebook outputs more accessible you can check out that project it's called notebooks for all and i will say tony is um he's been working on creating a version of jupyter notebooks that is um uh that is more readily accessible to us um and outputs things in more in more useful ways and so on so if you're interested in that maybe just send me an email i'll put you in touch with tony i know he's looking for testers and i heard that tony might be doing some kind of event coming up related to that as well so i'll let you guys know by email yeah i'm super interested in uh tactile svgs um especially on like tablets and phones so let's get in touch marco yeah definitely because i know the one thing is like to get the svg out but also make it human readable so we can get through the file and add our own things to it which is what i've been doing with like my own website um yeah yeah we can we can work on that stuff we just have to do some nasty things with pandas they're outside of the scope of here yeah i would say definitely possible it's just a matter of how to get from point a to point b in the in the easiest way and and uh and i'll put you guys does anyone have thanks does anyone else have any uh questions they'd like to get on the mic and ask and that was a fairly you know advanced question but you could ask a question about anything you know it doesn't have to be okay i live for those awkward pauses but you have to do them or otherwise people won't ask but we'll i think then we'll end the recording and i'll stay on for a minute or two if anyone has any other questions and we're looking forward to seeing you for um the uh the fourth workshop in the series that um sarah king will be leading on uh just the same time next tuesday so thank you all