⋮

This is a graded discussion: 10 points possible

due -

31 44

What are the existing methods?

So, picking a good number of bins is critical when you create a histogram. And as you may have noticed already, when you draw a histogram using Python or R, they somehow pick the number of bins automatically! So, I'd like to ask you to research the following:

1. What are the existing methods for picking the number of histogram bins? And what are the key ideas?
2. Which one is employed in Python (numpy, matplotlib, etc.) or R?

Then share your thoughts on what would be the best ways to go about drawing a histogram. You may argue, for instance, that we should use Python default, or that we should use "Sturges' rule", or that we should ditch everything and try multiple bin sizes.

Search entries or author | Unread | ↑ | ↓ | ✓ Subscribed

↩ Reply

○

https:/

Sneha Satish (https://iu.instructure.com/courses/2165942/users/6679606)

Sep 25, 2023

⋮

Some methods I found to pick the number of bins are:

- Sturges' Rule: The idea is that the number of bins should be approximately 1 + log2(N) where N is the number of data points. It can be done in Python by defining num_bins and then plotting it.
- Square Root Rule: The idea is that the number of bins is set to the square root of the number of data point i.e. sqrt(N). It can be done in python by using sqrt and then plotting it.
- Freedman-Diaconis Rule: It takes into account the data's distribution and spead, then suggests a bin width of 2 * IQR/ (N^(1/3)). It can be done in Python.

Using defaults provided by libraries like Matplotlib can be a great starting point for people who are not well aware with these methods.

Reply

○

https:/

Dustin Cole (https://iu.instructure.com/courses/2165942/users/6701715)

Sep 25, 2023

I have never put much thought into binning. I have always added a large number of bins to see what the distribution would look like and then refined it to choose the number that seemed the most suitable for my purposes.

I just read that Python Numpy uses Sturges' by default. (I had to look that up as I am not familiar with it. Matplot lib uses the same rule by default. R uses Sturges' as well, so I would assume there is no single perfect answer but Sturges' seems to be popular.

Reply

○

https:/

Thomas Jablenski (https://iu.instructure.com/courses/2165942/users/6701599)

Sep 25, 2023

1. The methods that I found to choose bins were Sturge's rule, Freedman-Diaconis rule, Scott's Rule, Rice's rule, and Doane's rule. They all seem to relate in the sense that they take the number of bins and then multiply some values to it, some pertaining to the dataset (IQR or standard deviation), or taking logs of the number of observations. The former is the most common way to calculate bins.

2. In Python for numpy, the number of bins is defaulted to the number 10 but can take in a string to choose a different method.

When choosing the number of bins it isn't always certain what you're trying to look for in your chart. It sometimes takes some trial and error to find a good number of bins to choose. I vote for ditching everything and trying multiple bin sizes.

Edited by **Thomas Jablenski (https://iu.instructure.com/courses/2165942/users/6701599)** on Sep 25 at 9:31pm

Reply

○

https:/

Erik Gonzalez (https://iu.instructure.com/courses/2165942/users/6352173)

Sep 26, 2023

1. One method for selecting the number of histogram bins is Sturge's rule, which involves taking the base 2 log of a number and adding one to it. For example, a distribution with 32 records would lead to a result of 5 + 1, for a total of 6 bins. The Freedman-Diaconis rule considers the overall sample size while also taking into account the spread of data between the first and third quartile. Increases in sample size would increase the number of bins, while increases in the interquartile spread would decrease the number of bins (by increasing the bin width).

2. Numpy typically defaults to either Sturges or Freedman-Diaconis. It calculates both values and takes the one that returns the maximum amount of bins. Per the numpy documentation, this typically leads to Sturges being leveraged for smaller sampler sizes and FD for larger ones.

When drawing a histogram with no intuitive sense of the underlying data, I find it is typically good to start with the python defaults, but then to investigate multiple bin sizes. That being said, when considering multiple bin sizes, it is important to consider that you may accidentally "find" peaks in the data that are actually noise.

Reply

○

https:/

Carmen Galgano (https://iu.instructure.com/courses/2165942/users/6762945)

Wednesday

1. There are some mathematical ways to pick the number of histogram bins. Most consider factors such as sample size, the data range, and standard deviation. using square root choice, you would take the range and divide it by the IQR of the data multiplied by 1/square root of sample size. This is a good strategy because it compensates for size and distribution of the data. Another way is Sturges Rule, for normal distributions. This takes 1+log2(n) as the formula for bins. This is often the go to for selecting the number of bins.

2. Python's matplotlib uses Sturges rule. I would say that using Sturges rule is the best way to go about it because it simple and easy to understand. The only drawback is non normal data. If you have non normal data it might be best to opt for a different method for determining bins and to manually assign bins.

Reply

○

https:/

Yumeng Liang (https://iu.instructure.com/courses/2165942/users/6587577)

Wednesday

Generally speaking, there are several ways to pick bins. 1) Square Root Rule 2) Sturges' Formula 3) Rice Rule 4) Freedman-Diaconis Rule 5) Scott's Rule. For python, numpy and matplotlib often use simple methods like the Square Root Rule or Sturges' Rule by default, but they also allow users to specify the number of bins or use other methods, such as the Freedman-Diaconis Rule.

Reply

○

https:/

[Hymavathi Gummudala (https://iu.instructure.com/courses/2165942/users/6679250)](https://iu.instructure.com/courses/2165942/users/6679250)

Thursday

1. Methods for Picking Histogram Bins:
   - Square Root Choice (Freedman-Diaconis Rule): Based on data spread and less sensitive to outliers.
   - Sturges' Rule: Assumes bins = log2(n) + 1.
   - Scott's Rule: Bin width based on data's standard deviation.
   - Rice Rule: Suggests bins = 2n^(1/3).
   - Doane's Formula: Considers skewness and kurtosis.
2. Defaults in Python and R:
   - Python (matplotlib): Default bins determined automatically (combination of Sturges' and Scott's rules).-> `plt.hist()`
   - Python (numpy): Allows manual or default (similar to matplotlib). -> `numpy.histogram()`
   - R: Default bins determined by Sturges' rule but can be overridden
   - I would say its good to give multiple tries with different bins

Edited by [Hymavathi Gummudala (https://iu.instructure.com/courses/2165942/users/6679250)](https://iu.instructure.com/courses/2165942/users/6679250) on Sep 28 at 1:42am

Reply

○

https:/

[Shantanu Dixit (https://iu.instructure.com/courses/2165942/users/6684610)](https://iu.instructure.com/courses/2165942/users/6684610)

Thursday

1.

Existing methods for picking Bins:

Square Root Rule: The number of bins is the square root of data points.

Sturges’ Rule: Useful for small datasets. Uses a formula based on data point count.

Rice’s Rule: A bit like Sturges’ but slightly different formula.

Freedman-Diaconis Rule: Uses data spread to decide bin size.

Scott's Rule: Uses data's standard deviation to determine bin size.

2.

Python: Usually uses Freedman-Diaconis Rule but offers other choices.

R: It starts with Sturges' rule but you can pick others too.

I think I would go with the default. If it looks odd, try other methods and pick the best for my data.

Edited by **Shantanu Dixit (https://iu.instructure.com/courses/2165942/users/6684610)** on Sep 28 at 1:42pm

Reply

○

https:/

Sangzun Park (https://iu.instructure.com/courses/2165942/users/6703376)

Thursday

In general, there are Sturge's law and Freedman-Diaconis rule. The sturge rule is to set the number of bins considering the data size. This Sturge law is best suited to continuous data that are normally distributed and symmetric. However, on the other hand, it is not suitable for asymmetric data and numerous observations. freedman-Diaocnis is a rule that considers not only the sample size but also the distribution of the sample. Other examples include Doane's law, which is an improved version of the Sturge estimator that works better on non-normal data, Scottd's law, which considers data variability and size, and Rice's law, which only considers data size.

Python basically has auto, sturges, freedman, doane, scott, rice, and sqrt as bins options. R also provides a similar number of default options. However, since both can be used by specifying a formula by the user, there seems to be no rule or formula that cannot be implemented. I think the default option, auto, would be best for a first try. And the next method to consider is sturge. This is because normally distributed data of appropriate size are quite commonly used.

Reply

○

(https:/

Andi Mai (https://iu.instructure.com/courses/2165942/users/6705680)

Thursday

I usually use by default and if it's not proper, I'll set the number of bins manually to see which is better.

Existing method:

1. **Square Root Choice (Sturges' Formula):**
   - **Key Idea:** The number of bins is chosen as the square root of the number of data points.
   - **Use in Python and R:** This method is available in libraries like Matplotlib in Python (`plt.hist`) and base R (`hist` function).
2. **Scott's Rule:**
   - **Key Idea:** It considers the sample size and the variance of the data to estimate an optimal bin width and, consequently, the number of bins.
   - **Use in Python and R:** Scott's rule is commonly available in libraries like Matplotlib and Seaborn in Python and base R.
3. **Freedman-Diaconis Rule:**
   - **Key Idea:** Similar to Scott's Rule, it considers the sample size and the interquartile range (IQR) to estimate an optimal bin width and the number of bins.
   - **Use in Python and R:** This rule is often available in Python libraries like Matplotlib and Seaborn and in R.
4. **Doane's Formula:**
   - **Key Idea:** It accounts for skewness in the data by considering the number of data points, the standard error of skewness, and the kurtosis of the data.
   - **Use in Python and R:** Some libraries may implement this method, but it's less commonly used compared to the others.
5. **Manual Selection:**
   - **Key Idea:** You can also manually specify the number of bins based on domain knowledge or the specific requirements of your analysis.
   - **Use in Python and R:** Most histogram plotting functions in Python (e.g., Matplotlib) and R allow you to specify the number of bins manually.

Reply

○

https:/

[Gary Croke (https://iu.instructure.com/courses/2165942/users/6706306)](https://iu.instructure.com/courses/2165942/users/6706306)

Thursday

The default number of bins in matplotlib is 10, which seems like a reasonable value without analyzing the dataset. In R the default is 30, which seems pretty large.

There are a number of rules of thumb, or algorithms to determine the number of bins for a given dataset, including Doane's rule, the Rice rule and Square Root rule. One of the most popular methods is Sturge's rule, which sets the number of bins as the base 2 log of the size of the dataset, plus 1, rounding up. For example, for 200 observations, the number of bins would be ceil(log2(200) + 1) = 9. This method works well for a lot of datasets, including normal distributions, but suffers if the data has a lot of outliers.

Ultimately it is best to thoughtfully decide on the optimal number of bins after examining a dataset. If a one-size-fits-all approach must be selected as a first attempt, Sturge's rule seems okay, and better than simply selecting a fixed number for all datasets.

Reply

○

https:/

[Onur Tekiner (https://iu.instructure.com/courses/2165942/users/6758180)](https://iu.instructure.com/courses/2165942/users/6758180)

Friday

There are many of them, and the most common are;

Square root Choice (Freedman-Diaconis Rule),
Sturges' Rule,
Scott's Rule,
Rice's Rule,
Doane's Formula,
Freedman-Diaconis Rule.

The methods usually calculate the bin size out of the total sample size, distributions, standard deviation, etc.

Python uses Freedman-Diaconis rule.

In my opinion, there is no one correct answer for bin size. It depends on variables, values, and sample size. It also depends on how closely we need to investigate the data. If the details are so important, we can make higher bin sizes and vice versa.

Since the Freedman-Diaconis rule uses IQR, I would try this one for visualization.

Reply

https:/

Shreedeep Sadasivan Nair (*he/him/his*) (https://iu.instructure.com/courses/2165942/users/6813278)

Friday

There are number of ways of calculating bins for a histogram , the methods depends on the data.

1)A)Square Root Choice:

key idea: In this we use square root of the number of data points as the bins

B)Struge's Formula:

The number of bins is determined by the formula: k=1+log(N) where N is the number of data points

C) Scotts Rule**:**

The bin width is determined by the formula: h=3.5 * (std deviation)/(N^(1/3)), where N is the number of data points.

2)A) In python normally Square Root Choice or Struges Formula is used.

Edited by **Shreedeep Sadasivan Nair (https://iu.instructure.com/courses/2165942/users/6813278)** on Sep 29 at 12:23pm

Reply

https:/

Andrea Chung (https://iu.instructure.com/courses/2165942/users/6443321)

Friday

One of the common methods is square root rule. The number of bins should be the square root of the number of data points. The method can be implemented in Python by using numpy library and matplotlib library. Another method is Sturges' rule, where number of bins is 1 + log2(N) and N is number of observations. Lastly, one of the existing methods is Freedman-Diaconis Rule, where interquartile range is used to calculate the bin width. I believe that best methods can change by what kind of dataset we have. Square root rule is the most common methods but Sturges' method can be helpful when the dataset has bell-shaped distribution.

Reply

○

https:/

Prem Amal (https://iu.instructure.com/courses/2165942/users/6684842)

Saturday

Square Root Choice: This method takes the square root of the number of data points (n) to determine the number of bins. It is a simple method and works well for small datasets.

Reference:

https://www.qimacros.com/histogram-excel/how-to-determine-histogram-bin-interval/#:~:text=Calculate%20the%20number%20of%20bins%20by%20taking%20the%20square%20root,)%20by%20the%20%23%20of%20bins **(https://www.qimacros.com/histogram-excel/how-to-determine-histogram-bin-interval/#:~:text=Calculate%20the%20number%20of%20bins%20by%20taking%20the%20square%20root,)%20by%20the%20%23%20of%20bins)** .

Sturges' Formula: Sturges' formula suggests a number of bins based on the logarithm of the data size. It calculates the number of bins as 1+log2(n), where n is the number of data points. This method tends to produce fewer bins for smaller datasets and more bins for larger datasets.

Rice Rule: The Rice Rule determines the number of bins as 2*n^(1/3). It's similar to Sturges' formula but typically results in a larger number of bins for the same dataset size.

Scott's Normal Reference Rule: Scott's rule takes into account the standard deviation and the number of data points. It calculates the bin width as 3.5*std/n^(1/3) and then uses this width to calculate the number of bins. This method can handle datasets with varying levels of variance.

Freedman-Diaconis' Rule: This rule considers the interquartile range (IQR) and the number of data points. It calculates the bin width as 2*IQR/n^(1/3) and then determines the number of bins based on the width. This method is robust to outliers.

Reference:

https://www.statisticshowto.com/choose-bin-sizes-statistics/ **(https://www.statisticshowto.com/choose-bin-sizes-statistics/)**

Which one is employed in Python (numpy, matplotlib, etc.) or R?

In Python - Matplotlib, when we create a histogram using the plt.hist() function, it often uses a default automatic bin selection method . However, wecan specify the number of bins explicitly. The methods that can be used for automatic bin selection are:

'auto', 'fd', 'doane', 'scott', 'stone', 'rice', 'sturges', or 'sqrt'.

Reference:

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html
(https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html)

Using the automatic bin selection provided by Python or R is a good starting point. These libraries often choose reasonable bin sizes based on the data's characteristics. If we have domain knowledge or understand the data's underlying distribution, then we may choose to adjust the binning method accordingly.Also, experimenting with different bin sizes and methods to visually inspect the histograms can be done.

Edited by **Prem Amal (https://iu.instructure.com/courses/2165942/users/6684842)** on Sep 30 at 10:38am

Reply

○

https:/

Vedant Tapadia (https://iu.instructure.com/courses/2165942/users/6678810) ⋮

Saturday

Sturges' Formula: Number of Bins = 1 + $\log_2$(N), where N is the number of data points.

This formula is best suited for a normal distribution and would minimize the impact of outliers.

Square Root Choice: Number of Bins = √(N), where N is the number of data points.

The most simple method that can find you the right number of bins.

Rice's Rule: Number of Bins ≈ 2 * N^(1/3).

This rule is a simple method that encourages a larger number of bins.

Scott's Normal Reference Rule: Bin Width = 3.5 * (Sample Standard Deviation) / (N^(1/3)), where N is the number of data points. Then, the Number of Bins = (Data Range) / (Bin Width).

This rule takes into account the sample size and the standard deviation of the data to determine the number of bins.

The default number of bins for Python using matplotlib is 10 and the default for R is Sturges' formula.

In my opinion we should try multiple methods for deciding number of bins as one method might be suitable for one dataset and the other might be more suitable for the other dataset. There is no one single method that would fit all datasets and give the best histogram for all cases.

Reply

○

(https:/

Mothi Gowtham Ashok Kumar (*he/him/his*) (https://iu.instructure.com/courses/2165942/users/6683278)

Saturday

- **Sturges' rule:** This rule uses the number of data points to calculate the recommended number of bins. The formula for Sturges' rule is:

```
k = 1 + log2(n)
```

where k is the number of bins and n is the number of data points.

- **Freedman-Diaconis rule:** This rule uses the interquartile range (IQR) to calculate the recommended number of bins. The formula for the Freedman-Diaconis rule is:

```
k = 2(IQR)^(3/5) / n^(1/5)
```

where k is the number of bins, IQR is the interquartile range, and n is the number of data points.

- **Scott's rule:** This rule also uses the IQR to calculate the recommended number of bins. The formula for Scott's rule is:

```
k = (3.49 * IQR / n^(1/3))
```

where k is the number of bins, IQR is the interquartile range, and n is the number of data points.

In Python, the default method used to determine the number of histogram bins is **Sturges' rule**. This can be changed using the bins argument of the hist () function.

In R, the default method used to determine the number of histogram bins is **Scott's rule**. This can be changed using the bins argument of the hist () function.

Thoughts:
When using Python (matplotlib), it is fine to use if the dataset is large
when using R, I would consider Sturges' rule instead of Scott's rule, if the dataset is small
If we are unsure how many bins to use, we can try multiple bin sizes and see which one gives the most informative and visually appealing histogram.

Edited by Mothi Gowtham Ashok Kumar (https://iu.instructure.com/courses/2165942/users/6683278) on Sep 30 at 12:18pm

Reply

○

https:/

Olufisola Oladipo (https://iu.instructure.com/courses/2165942/users/6469527)

Saturday

The existing methods for picking the number of histogram bins are:

1. Square Root rule - the number of bins should be the square root of the number of data points

2. Sturges' rule - the number of bin should be 1 + log2(n), n being the data points

3. Freedman-Diaconis rule - uses interquartile range (IQR) to calculate the bin width.

2. The square root rule is the default method for Matplotlib in Python.

I think one needs to understand the data points in context and use one's discretion to choose the appropriate method that will accurately and meaningfully represent the data.

Reply

○

https:/

Robert Perez (*he/him/his*) (https://iu.instructure.com/courses/2165942/users/6701521)

Saturday

1. It does seem like picking a number of bins is somewhere between art and science. A bit of research on the question reveals a lot of debate and a lot of examples supporting different bin choices. However, most of the methods involve counting the number of observations in your

distribution and performing some calculation on them to settle on a number of bins. One example came from David Scott in 1979 and is as follows:

$R(n^{(1/3)})/(3.49\sigma)$

where

- R is the range of data (in your case $R = 3-(-3)= 6$),
- n is the number of samples,
- σ is your standard deviation.

(Source: **https://academic.oup.com/biomet/article-abstract/66/3/605/232642?redirectedFrom=fulltext** (https://academic.oup.com/biomet/article-abstract/66/3/605/232642?redirectedFrom=fulltext))

2. From what I can tell, MatPlotLib uses a default of 10 bins.

(Source: **https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.hist.html#:~:text=With%20Numpy%201.11%20or%20newer,bins%22%5D%20%3D%2010%20** (https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.hist.html#:~:text=With%20Numpy%201.11%20or%20newer,bins%22%5D%20%3D%2010%20) .)

This number seems fairly arbitrary, so perhaps I'm missing some nuance where a different method is really being run behind the scenes to arrive at a bin number.

Knowing myself, I would probably start with the default and then make adjustments in an iterative way, not using any particular formula, per se, until the visualization reflects the story I want my data to tell.

Reply

○

https:/

Mukul Gharpure (https://iu.instructure.com/courses/2165942/users/6678592)

Saturday

Choosing the right bin count for histograms is very important. Both Python and R have default methods to determine bin numbers:

- Methods for picking histogram bins:
  - Square-root rule
  - Rice Rule
  - Sturges' rule

        - Freedman-Diaconis
    - Python's default: Often uses the Freedman-Diaconis rule, though it can vary between libraries.
    - R's default: Typically uses Sturges' rule.
    - Recommendation**:** While defaults work in many cases, it's important to understand the data and potentially experiment with multiple bin sizes for a comprehensive view.

Reply

○

(https:/

Madhuri Patibandla (*she/her/hers*) (https://iu.instructure.com/courses/2165942/users/6760559)

Saturday

1. What are the existing methods for picking the number of histogram bins? And what are the key ideas?

Count the number of data points.

Calculate the number of bins by taking the square root of the number of data points and round up.

Calculate the bin width by dividing the specification tolerance or range (USL-LSL or Max-Min value) by the # of bins.

Which one is employed in Python (NumPy, matplotlib, etc.) or R?:

in Python we can use the NumPy. Histogram and matplotlib.hist packages to plot the histogram and when we use this package then bins will calculate automatically and display the graph.
In R also we can create a histogram with R. hist function and call the vector, then based on thte given data points then R will display the histogram.

the best way to use the histograms to use the Sturges rule to identify the bins for histogram.
The Square-root Rule: Number of bins = $\lceil \sqrt{n} \rceil$
The Rice Rule: Number of bins = $\lceil 2 * 3\sqrt{n} \rceil$
The Freedman-Diaconis' Rule: Number of bins = (2*IQR) / $3\sqrt{n}$ where IQR is the interquartile range.

Reply

https:/

Tejo Kotikalapudi (https://iu.instructure.com/courses/2165942/users/6591977)

Saturday

There are multiple ways in which the bin size is determined.

1. Sturge's Rule: $k = \lceil \log_2 n \rceil + 1,$
where:
K = number of class intervals (bins).
N = number of observations in the set.
log = logarithm of the number.

for continuous data that is normally distributed and symmetrical. This is not good for severely skewed, non symmetric data sets, or for an extremely large number of observations

2. Doane's rule to Choose Bin Sizes
This modified version of Sturge's rule may also lead to over-smoothing:

$$k = 1 + \log_2(n) + \log_2\left(1 + \frac{|g_1|}{\sigma_{g_1}}\right)$$

where $\text{�}_1 g_1$ is the estimated 3rd-moment-**skewness (https://en.wikipedia.org/wiki/Skewness)** of the distribution and

$$\text{�}\text{�}_1 = \sqrt{\frac{6(\text{�} - 2)}{(\text{�} + 1)(\text{�} + 3)}}\sigma_{g_1} = \sqrt{\frac{6(n-2)}{(n+1)(n+3)}}$$

Scott's Rule

Scott's rule to choose bin sizes is based on the standard deviation(σ) of the data. $h = \frac{3.49\hat{\sigma}}{\sqrt[3]{n}},$

Rice's Rule: simple alternative to Sturges' rule.
Rice's rule is defined as: (cube root of the number of observations) * 2.

Freedman-Diaconis's Rule
This formula uses the interquartile range (IQR): The interquartile range is a measure of where the "middle fifty" is in a data set. Where a range is a measure of where the beginning and end are in a set, an interquartile range is a measure of where the bulk of the values lie IQR = Q3 –

Q1.

$$h = 2\frac{\text{IQR}(x)}{\sqrt[3]{n}},$$

2. Which one is employed in Python (numpy, matplotlib, etc.) or R?

In numpy, passing bins parameter in numpy.histogram will provide optimal bin width and consequently the number of bins by using are calculated. The parameters can be passed as

By default, auto is used which returns maximum of the 'sturges' and 'fd' Freedman Diaconis Estimator estimators. Provides good all around performance.

matplotlib also uses `numpy.histogram_bin_edges` **(https://numpy.org/doc/stable/reference/generated/numpy.histogram_bin_edges.html#numpy.histogram_bin_edges)** : 'auto', 'fd', 'doane', 'scott', 'stone', 'rice', 'sturges', or 'sqrt'., by default auto.

From the above, I feel we should use the Python, as it takes the maximum of the 'sturges' and 'fd' Freedman Diaconis Estimator estimators, also it is mentioned in the offiicial numpy documentation that this way provides good all around performance.

However, the choice of method to determine the number of bins should depend upon the nature of the data and the visual planned. One can view the visualisation for different bin sizes also to see which works best and its better representation.

Reply

○

https:/

Jeevan Deep Mankar (https://iu.instructure.com/courses/2165942/users/6644229) ⋮

Saturday

There are some existing methods for picking histogram bins -
Square root rule: This rule simply states that the number of bins should be equal to the square root of the number of data points.
Sturge's rule: This rule takes into account the range of the data as well as the number of data points. The formula for Sturge's rule is:
Number of bins = 1 + 3.322 * log10(n)
where n is the number of data points.
Freedman-Diaconis rule: This rule takes into account the quartiles of the data as well as the number of data points. The formula for the Freedman-Diaconis rule is:
Number of bins = 2 * (IQR / IQR_hat)

where IQR is the interquartile range and IQR_hat is an estimate of the IQR based on the standard deviation of the data.
By default, both Python and R use the square root rule to determine the number of histogram bins. However, both languages also provide ways to specify the number of bins manually or to use other algorithms, such as Sturge's rule or the Freedman-Diaconis rule.
Best way to draw histogram -
Start with a reasonable number of bins. The square root rule is a good starting point, but you may want to use more or fewer bins depending on the size and distribution of your data set.
Experiment with different bin sizes. Once you have a reasonable number of bins, try plotting the histogram with different bin sizes to see how the results change.
Consider using a histogram density plot. A histogram density plot is a type of histogram that shows the estimated probability density function of the data. Histogram density plots can be useful for visualizing the distribution of data with a large number of bins.

Reply

○

(https:/

Sydney Dicks (https://iu.instructure.com/courses/2165942/users/6819877)

Saturday

1. A few existing methods used for picking the number of histogram bins I found include Sturge's Rule, the Rice criterion, the Freedman Diaconis rule, and just playing around with different sizes until a suitable one is found. The use of these methods should vary depending on the distribution of the data and involve the use the statistics about, size of, and/or the distribution of the data.

2. Python has a default bin size of 10. R has a default bin size of 30.

I would argue that one of the defined rules should be used, if possible, and then the bin size can be altered from there. So start out with using a rule to find the general bin number range you should be in, then play around with bin numbers in that range to see what best fits your data.

Reply

○

(https:/

Jash Shah (https://iu.instructure.com/courses/2165942/users/6684840)

Sunday

Referencing https://www.statisticshowto.com/choose-bin-sizes-statistics/ (https://www.statisticshowto.com/choose-bin-sizes-statistics/)

In Python's `NumPy`, the default method for determining the number of bins is often based on the Freedman-Diaconis rule. This method is designed to provide a balance between too few and too many bins, making it a reasonable choice for many scenarios. However, you can also specify the number of bins manually using the bins parameter when creating a histogram.

The best approach can be trial and error, using experimentation, and using domain knowledge and visualization goals in place to handle this.

The key ideas in selecting the number of bins for a histogram are to strike a balance between granularity and smoothness, avoid under-smoothing or over-smoothing the data, and consider the underlying distribution characteristics. Here are the key ideas:

1. **Granularity vs. Smoothness:** The number of bins determines the granularity of the histogram. More bins provide finer detail but can result in noise or overfitting to the data, while fewer bins result in smoother representations but may hide important features.
2. **Avoiding Under-Smoothing:** If you use too few bins, the histogram may oversimplify the data distribution and important patterns or variations might be missed. Under-smoothing can lead to a loss of information.
3. **Avoiding Over-Smoothing:** On the other hand, using too many bins can lead to over-smoothing, making the histogram sensitive to noise or small fluctuations in the data. This can result in a misleading representation.

Edited by **Jash Shah (https://iu.instructure.com/courses/2165942/users/6684840)** on Oct 1 at 12:36am

Reply

(https:/

Simon Driver (https://iu.instructure.com/courses/2165942/users/6818242)

Sunday

1. There are a variety of existing methods for choosing the right number of bins; there is Sturges' rule, which is 1 + log2(n) bins, where n is the number of data points. There is the Freedman-Diaconis rule, which is based off of the spread and interquartile range of the dataset. There is also the square root rule, which is just taking the sqrt of the number of datapoints. Each of these methods, and other methods, is fundamentally based on trying to find a number that relates to the total number of observations via some equation based on something such as the square root or spread of the data.

2. In matplotlib, the default is the square root rule. In R, the default is Sturges' rule.

3. I would argue that we should probably try multiple approaches. It takes very little time and effort to try and implement a handful of these approaches, and we may glean some useful insights into the best way to present our data by trying out a variety of numbers of bins.

Source: https://saturncloud.io/blog/how-to-choose-bins-in-matplotlib-histogram-a-guide-for-data-scientists/

Reply

○

https:/

Sarah Biggs (https://iu.instructure.com/courses/2165942/users/5667580)

Sunday

1. There are many ways to choose bin size. I'll try to summarize below:
    1. Square root of n - the number of bins is the square root integer of the sample size. Only accounts for sample size and not spread.
    2. Sturge's rule - take log, base 2, of the sample size (as an integer), and add one. Only accounts for sample size and not spread.
    3. Freedman-Diaconis rule - takes into account sample size and spread of data by first finding bin width, which is the interquartile range multiplied by two and then divided by the cubed root of the sample size. Then you find the number of bins by subtracting the minimum of your x range from the maximum of your x range and dividing that by the bin width you previously calculated.
    4. Rice's rule - alternative to Sturges, found by multiplying the cubed root of the sample size by two.
    5. Doane's formula - also alternative to Sturge's but helps with non-normal data. Describing how it's calculated is too complicated here, but it aids with non-normal data by taking into account an estimated skewness of the data.
    6. Scott's normal reference rule - also best for normal data. However, it incorporates the sample standard deviation in order to minimize MISE.
    7. Varying bin widths - choosing varying bin widths based on the data.
2. Python choose 10 bins by default in Numpy, Matplotlib, and Pandas.

If my data are normal, I'd start with Sturge's rule. However, depending on what the data are showing, I'd likely try the F-D rule if the data are not normal. This doesn't seem to be a one-size-fits-all question though, and it might take some work to figure out what best displays the data and tells the story easily to an audience.

Reply

○

https:/

Maria Klein (https://iu.instructure.com/courses/2165942/users/5444499)

Sunday

It never occurred to me that there would be so many methods to try to optimize the number of histogram bins. In a quick search I identified:

- non-varying default, in which the same number of bins are always plotted regardless of data characteristics
- self-selected, which is any number of bins you think would be good between 5-20 (ideally which would be of the same interval size and your number of data points is evenly divisible by)
- variable bin width (where the number of samples in each bin is expected to be approximately equal)
- Sturge's rule
- Doane's rule
- Scott's rule
- Rice's rule
- Freedman-Diaconis's rule
- square root rule
- Shimazaki and Shinomoto's choice

The last seven are generally functions of some combination of sample size, measure of central tendency (mean or median), and measure of dispersion (sd, variance, or IQR). The method used by R depends on which code or package you use to create the histogram. The base hist function uses Sturge's rule; however, ggplot2 uses a default of 30 bins. In python, using matplotlib will result in a histogram with 10 bins by default.

I do think the number of bins should be informed by the data characteristics rather than a default integer, so I would be inclined to try one of the "rule" methods. I would probably avoid the variable bin width option unless necessary (due to a non-linear scale), because I think it's common to misinterpret the height rather than the area as the representation of frequency.

Reply

(https:/

Ao Zhang (https://iu.instructure.com/courses/2165942/users/6703098)

Sunday

There are several methods for picking the number of bins. A widely used one is Sturge's Rule. The formula is K = 1 + 3.322 x logN, where K is number of bins, N is number of observations. It is used when the data is continuous data and the data followed normally distributed. Meanwhile, there are also other rules for determining the number of bins. For example, Doane's Rule, Scott's Rule, Rice's Rule, and Freedom-Diacoin's Rule.

In Python, np.histogram uses a method similar to the Freedman-Diaconis rule by default. And plt.hist uses the Sturges' rule by default.

In R, hist function uses Sturges' Rule by default.

In my opinion, I will try the default method at first and see the output. If the number of bins could show the dataset clearly, I would use the figure based on the default method. If there are some misleadings, I will try to set the parameters by myself.

Reply

○

https:/

Adam Hume (https://iu.instructure.com/courses/2165942/users/6056428)

Sunday

After researching some of the methods to pick the number of bins, I have found the following:

- Square root choice
- Scott's Rule
- Rice rule
- Sturges' formula

The first method calculates the number of bins as the square root of n. This is an easy calculation method which makes it easy to use. The second, Scott's Rule, says to use 3.5 * (std deviation / n^(1/3)). This method takes into account the size of the data as well as the variability which makes it a more advanced and suitable option compared to the first. Rice rule suggests to use 2 * n^(1/3). This method is similar to the first in that it is easy to use and applicable across many different data sets. Finally, Sturges' formula is 1+log2(n). This method is recommended for use in large datasets that are normally distributed.

Reply

○

https:/

Shubham Agarwal (https://iu.instructure.com/courses/2165942/users/6682743)

Yesterday

1. Methods
    1. Square Root Choice: It calculates the number of bins as the square root of the total number of data points.
    2. Rice rule: It calculates the number of bins as twice the cube root of the total number of data points

2. Python and R (ggplot2) use heuristics and optimization techniques to select an optimal number of bins based on the data's characteristics.

I believe the number of bins should be carefully selected after understanding the data distribution and other characteristics, there is no one rule which can fit in every case

Reply

○

(https:/

Vaibhav Piyushkumar Lodhiya (https://iu.instructure.com/courses/2165942/users/6694681)

Yesterday

Sturges' Formula: Sturges' formula is a simple and widely used method that assumes a Gaussian distribution of data

Scott's Rule: Scott's rule considers the bandwidth (width of the bins) and is based on the standard deviation (σ) and number of data points (N).

Freedman-Diaconis Rule: The Freedman-Diaconis rule considers the interquartile range (IQR) and is less sensitive to outliers. It calculates the number of bins as:

Rice Rule:The Rice rule is a less common method that calculates the number of bins

Doane's Formula: Doane's formula is an extension of Sturges' formula and incorporates skewness in the data.

Reply

○

(https:/

Harsh Patel (*he/him/his*) (https://iu.instructure.com/courses/2165942/users/6825193)

Yesterday

I am not sure of many methods for picking the number of histogram bins. I only heard of square root choice.

Reply

○

(https://

Rohan Isaac (https://iu.instructure.com/courses/2165942/users/6694525)

Yesterday

1. The existing methods include Sturge's rule, Freedman-Diaconis' rule, using square root of number of samples, rice rule as well. The key idea remains that the bins are decided based on the number of data points in the sample. Some methods also consider the distribution of the sample.

2. Python uses Freedman Diaconis rule while R uses Sturge's rule.

I would say the best way to draw a histogram would be to use methods that also consider the distribution metrics of the sample as well but this can be biased as well since the metrics like standard deviation are mostly applicable to normal distributions.

Reply

○

(https://

Ram Kiran Devireddy (https://iu.instructure.com/courses/2165942/users/6677399)

Yesterday

Python libraries like Matplotlib and Numpy and R often use Sturges' Rule as a default because it's straightforward and widely known. But there are other methods as well, which these libraries offer, like the Scott's Rule, Freedman-Diaconis Rule etc.

Sturges' Rule:

This rule suggests the number of bins should be proportional to the logarithm of the number of data points. It's simple and often used.

Scott's Rule:

It considers the data's standard deviation and number of data points to determine the bin width.

Freedman-Diaconis Rule:

Similar to Scott's rule but considers the data's interquartile range (IQR) instead of standard deviation.

While Sturges' Rule is a good starting point for most of the datasets, there are still wide Varity of methods to explore. It really depends on the data. Sometimes, experimenting with a few methods or visualizing with different bin sizes can help you decide what works best for your specific dataset.

Reply

○

https:/

Shruthi Senthilmani (https://iu.instructure.com/courses/2165942/users/6688148)

1:23am

1. There are many existing methods for picking the number of histogram bins: In **Square Root Choice (Freedman-Diaconis Rule)**, the number of bins is determined based on the square root of the number of data points.

In **Sturges' Rule, t**he number of bins is determined using a logarithmic formula based on the number of data points: `1 + log2(n)` where 'n' is the number of data points.

In **Scott's Rule,** the number of bins is determined based on the standard deviation and the number of data points: `3.5 * (std_deviation / (n^(1/3)))` where 'n' is the number of data points and 'std_deviation' is the standard deviation of the data. This method can be explicitly specified as an argument (`bins='scott'`) when creating a histogram in matplotlib.

In Rice's Rule, the number of bins is determined using a formula that accounts for the cube root of the number of data points: `2 * n^(1/3)` where 'n' is the number of data points.

By Manual Selection, we can manually choose the number of bins based on our domain knowledge or by experimenting with different bin sizes to find the most informative representation.

2. In Python, by default, the maximum of the **Square Root Choice (Freedman-Diaconis Rule) and Sturges' Rule is used.** In R, by default, **Sturges' Rule is used.**

Ultimately, the best approach for drawing a histogram depends on the nature of the data and the insights that we aimed to extract. Python's default method seems like the most appropriate one as it takes the maximum of either Sturges and Square root choice, and this default choice can then be experimented with other methods like Scott's rule, etc. If we have the domain knowledge, manual selection would also be a good option to select the bin sizes.

Reply

○

https:/

Akash Patil (https://iu.instructure.com/courses/2165942/users/6699404)

1:23am

1. Square Root Choice aka Freedman-Diaconis Rule is based on the square root of the number of data points. Bins are calculated by taking the square root of the number of points.
2. Sturges' Rule, the number of bins is calculated as *bins = 1 + log2(n)*, where n is the number of data points.
3. Scott's Rule is based on the standard deviation of the data. The formula is: *bins = 3.5 * (std_deviation / (n^(1/3)))*.
4. Rice's Rule is calculated as *bins = 2 * (n^(1/3))*, where 'n' is the number of data points.
5. Manual Selection where bins are selected manually.

In Python, the maximum value between 'Square Root Choice (Freedman-Diaconis Rule)' and 'Sturges Rule' is used by default while in R 'Sturges Rule' is used by default.

Python's default method seems the most appropriate, but you can always start with the default and then experiment with other methods or manual selection. This can help you find the best method that suits your dataset specifically as there is no universal method that works for everything, every method has its advantages and disadvantages.

Reply

○

https:/

Sarthak Vivek Chawathe (*he/him/his*) (https://iu.instructure.com/courses/2165942/users/6688770)

12:19pm

There are several methods for picking the number of histogram bins, each with its own set of assumptions and considerations. Here are some of the commonly used methods, along with their key ideas:

1. **Square Root Choice (Freedman-Diaconis' Rule):**
   - **Key Idea:** The number of bins is calculated based on the square root of the number of data points. It's less sensitive to the sample size and aims to strike a balance between oversmoothing and over-detailing the data.
   - **Python/R Usage:** This method is often used in both Python (e.g., Matplotlib) and R.
2. **Sturges' Rule:**
   - **Key Idea:** Sturges' Rule assumes that the number of bins should be logarithmic with respect to the number of data points. It's a simple rule that works well for relatively small

datasets with normally distributed data.

- **Python/R Usage:** This method is also commonly implemented in both Python and R.

3. **Scott's Rule:**

   - **Key Idea:** Scott's Rule takes into account the sample standard deviation and sample size to determine the bin width. It aims to minimize the mean integrated square error of the density estimate.
   - **Python/R Usage:** Scott's Rule is used in some libraries, particularly in SciPy for Python.

4. **Doane's Formula:**

   - **Key Idea:** Doane's Formula takes skewness into account, assuming that data with different levels of skewness require different numbers of bins. It's considered a modification of Sturges' Rule.
   - **Python/R Usage:** This method is less commonly used but can be implemented manually in Python or R.

5. **Rice Rule:**

   - **Key Idea:** Rice Rule is another simple rule that rounds the number of bins up to the nearest power of 2.
   - **Python/R Usage:** It's not as widely used in Python or R but can be implemented manually.

6. **Automatic Methods (e.g., Freedman-Diaconis' Rule with Adaptive Kernel Density Estimation):**

   - **Key Idea:** Some libraries, like Seaborn in Python, use a combination of automatic methods. They first calculate the Freedman-Diaconis' Rule bin width and then use it to estimate the number of bins based on kernel density estimation.
   - **Python/R Usage:** Seaborn is an example of a library that employs this approach.

When choosing the best method for drawing a histogram, it depends on various factors, including the characteristics of your data, the purpose of the histogram, and personal preferences. Here are some considerations:

- **Data Distribution:** If your data is normally distributed or follows a specific known distribution, rules like Sturges' or Scott's may work well. If the data distribution is unknown or complex, Freedman-Diaconis' Rule or adaptive methods can be useful.

- **Sample Size:** Consider the size of your dataset. Some rules, like Sturges', may not perform well with very small or very large datasets.

- **Skewness:** If your data is highly skewed, consider methods like Doane's that account for skewness.
- **Visualization Tool:** The library or tool you're using may have its own default method for determining the number of bins. Consider using this default unless you have specific reasons to choose another method.
- **Experimentation:** Experiment with different binning methods and visualize the data with multiple bin sizes to see which one provides the most meaningful insights for your analysis.

In summary, the choice of binning method depends on the data and the specific context of your analysis. It's often a good practice to explore multiple bin sizes and methods to ensure that your histogram effectively communicates the underlying data distribution.

Reply

○

(https:/

Anudeep Devulapally (*he/him/his*) (https://iu.instructure.com/courses/2165942/users/6696028)

1:16pm

simple rule of thumb that can be used to calculate the optimal number of bins for a histogram. The formula for Sturge's rule is:

Number of bins = 1 + log2(n)

Both Python and R both use Sturge's rule and and often employ automated methods to determine the number of bins when creating histograms.

The choice of the best method for picking the number of bins depends on your data and the goals of your analysis.
Automated methods are often a good starting point, especially when you're exploring the data or creating quick visualizations.

Edited by Anudeep Devulapally (https://iu.instructure.com/courses/2165942/users/6696028) on Oct 3 at 1:17pm

Reply

○

(https:/

Yashada Nikam (*she/her/hers*) (https://iu.instructure.com/courses/2165942/users/6692441)

3:55pm

The default technique in Python commonly uses libraries like NumPy and Matplotlib, though this can change based on a specific function being called. For instance, the hist function in Matplotlib frequently uses Sturges' rule by default.

Depending on the plotting function, R's default behavior for deciding the number of bins can change. R frequently also employs Sturges' rule by default, though.

Reply

○

https:/

David Rosenthal (https://iu.instructure.com/courses/2165942/users/6762824)

4:03pm

1. There are many different ways to chose how many bins should be in a histogram including: Sturges' rule: Use the formula that uses the total number of data points.

square root choice: You can take the square root of the total number of data points.

rice rule: Here you round up the cube root of the total number of data points.

2. In python the typical default when using the hist function uses Sturges' rule but you can change this is you wnated.

I think that it all depends on the data you have, how many data points you have, the type of data you are showing, and what message you are trying to portray to the end user. I usually look at these factors and pick what I believe fits the graph the best to show a well rounded depiction. I also at time just use the default since it is easier and is usually pretty accurate with the number of bins that gets created.

Reply

○

https:/

Shreyas Aswar (*he/him/his*) (https://iu.instructure.com/courses/2165942/users/6676727)

4:19pm

- 1. Some of the methods :
  - Sturge's rule : It is quite simple and uses the following formula - Bins = log (number of observations) + 1
  - Sturge's rule only considers sample size
  - Freedman-Diaconis rule : This also considers spread of sample and not only the sample size.

    - Bayesian Blocks: It is a dynamic histogram drawing method which optimizes one of several possible fitness functions to determine an optimal binning for data, where the bins are not necessarily uniform width.
    - Scott's rule**:** This rule is based on minimizing the mean integrated squared error of the histogram.

  - 2. Python and R allows use of different of methods.
    - Both numpy and matplotlib uses by default 10 bins if the bins is an integer
    - If bins is a sequence it defines the bin edges based on left edge of first bin and right edge of last bin.
    - They are unequally spaced though.
    - 'auto', 'fd', 'doane', 'scott', 'stone', 'rice', 'sturges', or 'sqrt' these are the various options we can chose
    - Auto is maximum of Sturge's or FD
    - In R it is Sturge's method by default for hist() from graphics

I think using the default methods is good idea since they are tried and tested. However, it is important to understand the data and make decision based on the data. Sometimes it is also a good idea to experiment with multiple methods and pick one which seems to better than others.

References :
1.https://www.statology.org/sturges-rule/
2.https://medium.datadriveninvestor.com/how-to-decide-on-the-number-of-bins-of-a-histogram-3c36dc5b1cd8
3.https://ui.adsabs.harvard.edu/abs/2013ApJ...764..167S/abstract
4.https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html
5.https://numpy.org/doc/stable/reference/generated/numpy.histogram_bin_edges.html#numpy.histogram_bin_edges
6.https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/hist

Reply

○

(https:/

Jerin Easo Thomas (https://iu.instructure.com/courses/2165942/users/6688908)

4:21pm

1. Existing Methods for Determining the Number of Bins:
a. Square Root Rule: Calculates the number of data points by taking the square root of the number of data points.

b. Sturges' Rule: Depends on the dataset size in logarithm base 2.

c. Rice Rule: A middle-of-the-road method appropriate for larger datasets.

d. Freedman-Diaconis Rule: Determines bin width by taking data spread (IQR) into account.

e. Scott's Rule: Best for regularly distributed data, includes standard deviation.

2. Python and R Default Methods:
Python (e.g., matplotlib): Frequently implements a variant of the Freedman-Diaconis Rule.

R: By default, applies Sturges' rule.

Best Practice: Begin with the software's default bin size. Experiment with different bin sizes and binning algorithms to appropriately represent the data depending on its nature and distribution. Visual observation and comparison of histograms with different bin sizes can help you determine the most representative bin size for your particular dataset.

Reply

(https:/

Alan Varkey (https://iu.instructure.com/courses/2165942/users/6681532)

5:01pm

1. Sturges' Rule:

Suggests bins ≈ 1 + log2(N) for normal distribution.
Employed in Python/R: Default in matplotlib, often used in both.

2. Scott's Rule:

Recommends bin width ∝ N^(-1/3) * standard deviation.
Employed in Python/R: 'scott' option in matplotlib, R's default.

3. Freedman-Diaconis Rule:

Bin width ∝ 2 * IQR / (N^(1/3)).
Employed in Python/R: 'fd' option in matplotlib, 'FD' method in R.

4. Square Root Rule:

Suggests bins ≈ √N.
Not default but manually calculable in Python/R.

5. Auto Methods:

Offered by some libraries like seaborn in Python.

--My opinion on Selecting the Best Approach:--

Default Behavior: Use it as a starting point if data distribution is unknown.
Domain Knowledge: Choose based on expected data distribution.
Exploratory Analysis: Experiment with multiple bin sizes.
Auto Methods: Convenient for data-adaptive visualizations.

Reply