

УДК 004.921

ВИЗУАЛИЗАЦИИ ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА PYTHON И ЕГО БИБЛИОТЕКИ MATPLOTLIB.PYPLOTT

Марчук К.А.

Научный руководитель – старший преподаватель Гецман Е.М.

Визуализация данных — это наглядное представление массивов различной информации. Визуальная информация лучше воспринимается и позволяет быстро донести до зрителя собственные мысли и рассуждения. Физиологически, восприятие визуальной информации является основной для человека, так как 90% всей информации человек воспринимает через зрение.

Существует огромное количество способов визуализации данных, но самым простым, а потому и самым распространенным методом визуализации являются графики.

Построение графиков — один из главных этапов обработки данных. Современные компьютерные программы, предоставляющие функцию построения графиков, условно можно разделить на две составляющие:

- программы с визуальным интерфейсом выполняющие различные функции, например: MS Excel, OpenOffice/LibreOffice Calc, Statistica, Grapher.
- программы предназначены только для создания двухмерных и трёхмерных графиков, например: MATLAB и SciLab и различные библиотеки вроде PGPlot и PLPlot.

Основное преимущество программ предназначенные только для создания двухмерных и трёхмерных графиков, заключается в том, что можно встроить построение графика в программу, производящую вычисления. Кроме того, скрипты позволяют легко перестраивать графики с новыми данными, автоматизировать построение графиков, а также дают почти неограниченный контроль над точностью позиционирования и размером графиков [1]. Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Модуль matplotlib — специализированная библиотека для языка Python. Основное её преимущество в простоте и скорости использования, а также она позволяет делать графики очень высокого качества (рис. 1).

В данной работе продемонстрированы следующие способы представления информации на языке Python:

- графика функции
- совокупности графиков функции
- диаграмма
- круговая диаграмма
- создание 3d каркасов

Графиком функции называется множество точек плоскости таких, что абсциссы и ординаты этих точек удовлетворяют уравнению $y=f(x)$.

Построим график нагрузочных потерь электроэнергии за год для линий 10кВ и 0,38кВ по таблице 1. В Республике Беларусь в целом сети 0,38 кВ

составляют около 40% от суммарной протяженности линий электропередач общей протяженностью 202,509 км (воздушные) и 39,923 км (кабельные). От надежности работы сетей 0,38 кВ и их загрузки решающим образом зависят надежность, качество и экономичность электроснабжения потребителей, а от качества расчетов технических потерь в сетях 0,38 кВ – точность определения коммерческих потерь.

Таблица 1 – отери электроэнергии в распределительных электрических сетях 0,38-10 кВ РЭС

Расчетный период (месяц)/ тыс.кВтч	1	2	3	4	5	6	7	8	9	10	11	12
Нагрузочные 0,38 кВ	401,8	279,2	239,7	224,5	195,3	163,8	178,1	180,4	165,1	237,7	251,0	263,5
Нагрузочные 10 кВ	46,9	45,8	33,2	30,1	27,1	22,7	23,4	24,9	20,3	30,2	37,3	36,9

```
import matplotlib.pyplot as plt#импорт библиотеки
import numpy as np#импорт библиотеки
fig=plt.subplots(figsize=(15,10))# создание фигуры, а также задание размера фигуры
y=np.array([401.8,279.2,239.7,224.5,195.3,163.8,178.1,180.4,165.1,237.7,251.0,263.5])#потери электроэнергии в тыс.кВт*ч
y1=np.array([46.923,45.796,33.229,30.085,27.054,22.729,23.408,24.864,20.312,30.171,37.269,36.926])#потери электроэнергии в тыс.кВт*ч
x=np.arange(1,13,1)#массив numpy со значениями от 1 до 12
plt.plot(x,y1,'ro',linestyle='-', label='Линия 10кВ')#построение графика
plt.plot(x,y,'bo',linestyle='-', label='Линия 0.38кВ')#построение графика
plt.title("График нормативных потерь электроэнергии в распределительных электрических сетях 0,38-10 кВ РЭС за год") # Оглавление
plt.xlabel('Месяц')
plt.ylabel("Потери электроэнергии в тыс. кВт*ч")
plt.grid() # нанесение сетки на наш график
plt.legend() # подпись графика
plt.show() # отображение информации
```

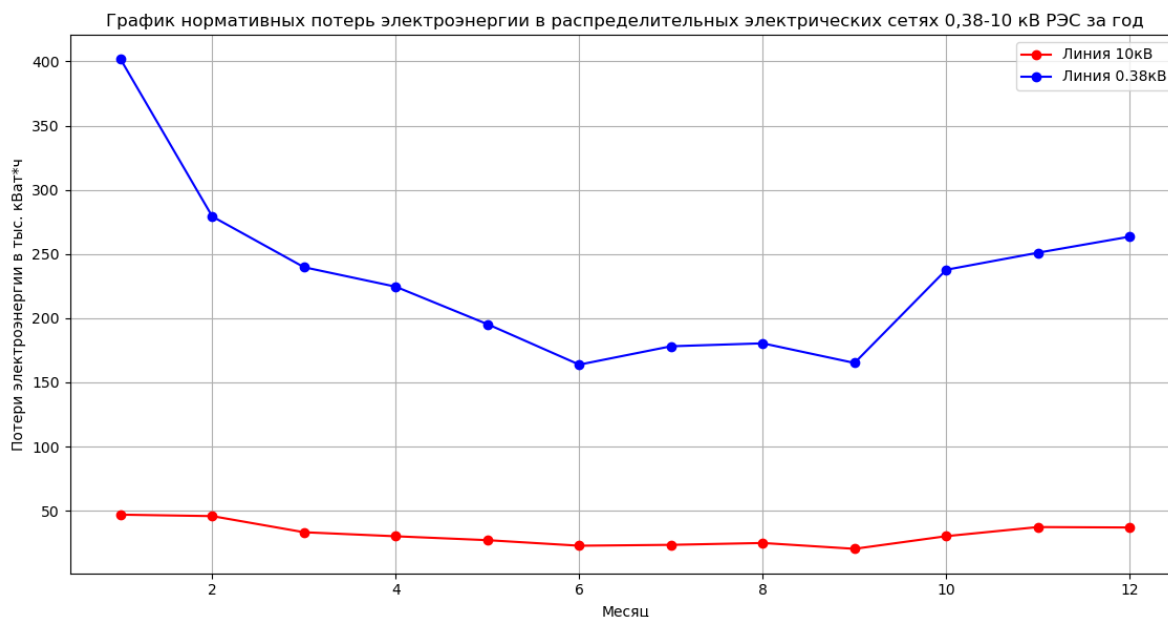


Рисунок 1 – Пример построения графика нормативных потерь электроэнергии при помощи Python

Наглядное построение из рис.1 видно, что нагрузочные потери в сетях 0,38 кВ во много раз больше, чем нагрузочные потери в линиях 10 кВ.Что говорит о проведении мероприятий по общей оптимизации сети для снижения потерь электроэнергии в электрических сетях.

В некоторых случаях необходимо визуализировать информацию для нескольких случаев для дальнейшего анализа и последующего сравнения данных либо же разделить график на отдельные функции(рис. 2).

```
import numpy as np
import matplotlib.pyplot as plt
y=np.array((851.57,499.052,500.127,232.822,388.608,297.269,361.311,274.226,344.345,
262.697,232.822,510.638))#потери электроэнергии в тыс.кВат*ч
x=np.arange(1,13,1)#массив numpy со значениями от 1 до 12
y1=np.array((651.57,799.052,590.127,390.32,699.608,297.269,361.311,274.226,589.39,
262.697,281.8,568.638))#потери электроэнергии в тыс.кВат*ч
y2=np.array((901.34,906.052,890.127,456.872,407.63,207.261,391.331,274.226,678.3,
391.627,223.98,458.638))#потери электроэнергии в тыс.кВат*ч
y3=np.array((890.23,744.12,540.112,310.823,633.18,227.221,333.322,474.226,539.1,
432.697,301.8,559.421))#потери электроэнергии в тыс.кВат*ч
fig,ax1=plt.subplots(2,2,figsize=(13,8))# создаем фигуру, а также задаем ее размеры
ax1[0][0].plot(x,y,'-ro')#Построение графика
ax1[0][0].set_xlabel("Месяц")#подпись оси x
ax1[0][0].set_ylabel("Потери электроэнергии в тыс. кВат*ч")#подпись оси y
ax1[0][0].set_title("График потерь электроэнергии за 2019")# подпись графика
ax1[0][1].plot(x,y1,'-bo')
ax1[0][1].set_xlabel("Месяц")
ax1[0][1].set_ylabel("Потери электроэнергии в тыс. кВат*ч")
ax1[0][1].set_title("График потерь электроэнергии за 2018")
ax1[1][0].plot(x,y2,'-yo')
ax1[1][0].set_xlabel("Месяц")
ax1[1][0].set_ylabel("Потери электроэнергии в тыс. кВат*ч")
ax1[1][0].set_title("График потерь электроэнергии за 2017")
ax1[1][1].plot(x,y3,'-go')
ax1[1][1].set_xlabel("Месяц")
ax1[1][1].set_ylabel("Потери электроэнергии в тыс. кВат*ч")
ax1[1][1].set_title("График потерь электроэнергии за 2016")
fig.tight_layout()#автоматически настраивает параметры подзаголовка
plt.show()# отображение информации
```

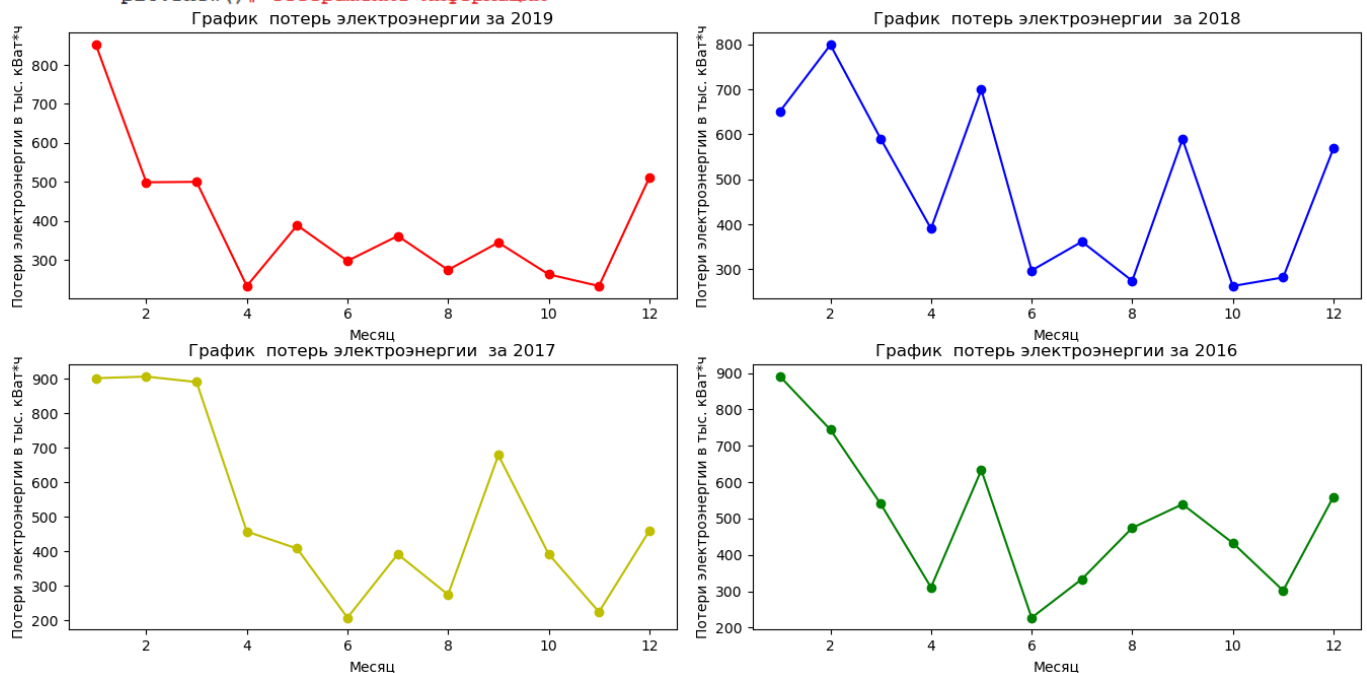


Рисунок 2 – Пример построения графиков нормативных потерь электроэнергии за разные промежутки времени при помощи Python

Диаграмма - один из способов представления статистических данных в графическом виде – в виде столбчатой диаграммы. Диаграммы полезны для представления различных изменений данных за определенный промежуток времени, а также для наглядного сравнения различных величин(рис. 3).

```
import numpy as np
import matplotlib.pyplot as plt
x=np.array((851.57,499.052,500.127,232.822,388.608,297.269,361.311,274.226,344.345,
262.697,232.822,510.638))#потери электроэнергии в тыс. кВт*ч
labels=['Январь','Февраль','Март','Апрель','Май','Июнь','Июль',
'Август','Сентябрь','Октябрь','Ноябрь','Декабрь']
fig,ax=plt.subplots(figsize=(13,8))#задаем размер нашей фигуры
ax.bar(labels,x)#создае гистограмму
ax.set_ylabel("Потери электроэнергии в тыс. кВт*ч")
ax.set_title("Диаграмма нормативных потерь электроэнергии в распределительных электрических сетях 0,38-10 кВ РЭС за год")
plt.show()
```

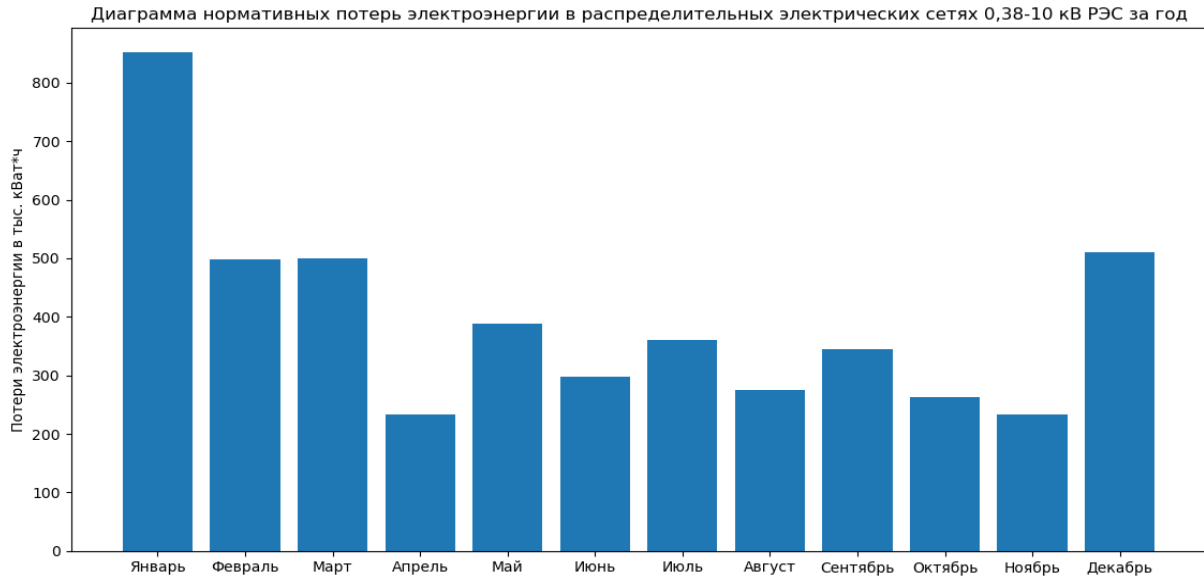


Рисунок 3 – Пример построения диаграммы потерь электроэнергии при помощи Python

Круговые диаграммы — это круглые графики, поделенные на секторы, каждый из которых представляет размер какой-либо связанной части данных(рис. 4).



```
import numpy as np
import matplotlib.pyplot as plt
fig=plt.subplots(figsize=(13,8))#задаем размер фигуры
ap=np.array([121,146,189])#потери в кВт*ч
name=["Потери на гистерезис","Потери вихревых токов","Нагрузочные потери"]#подписи на диаграмме
plt.pie(ap,labels=name,shadow=1,explode=(0,0,0.1),autopct="%1.1f%%")#создаем круговую диаграмму
plt.title("Потери трансформатора")#подпись диаграммы
plt.show()#отображение информации
```

Рисунок 4 – Пример построения круговой диаграммы при помощи Python

Помимо построения графиков и зависимостей между двумя параметрами в библиотеке Matplotlib имеется возможность построения трехмерных объектов. Трёхмерная графика — раздел компьютерной графики, посвящённый методам создания изображений путём моделирования объектов в трёхмерном пространстве (рис. 5).

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
u, v = np.mgrid[0:2*np.pi:20j, 0:1*np.pi:20j] # создаем многомерную плотную сетку
x = np.cos(u)*np.sin(v) # многомерная сетка x со значениями cos(u)*sin(v)
y = np.sin(u)*np.sin(v) # многомерная сетка y со значениями sin(u)*sin(v)
z = np.cos(v)**2 # многомерная сетка z со значениями cos(v)^2
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d') # создаем оси
ax.plot_wireframe(x, y, z) # построение каркаса
plt.show()
```

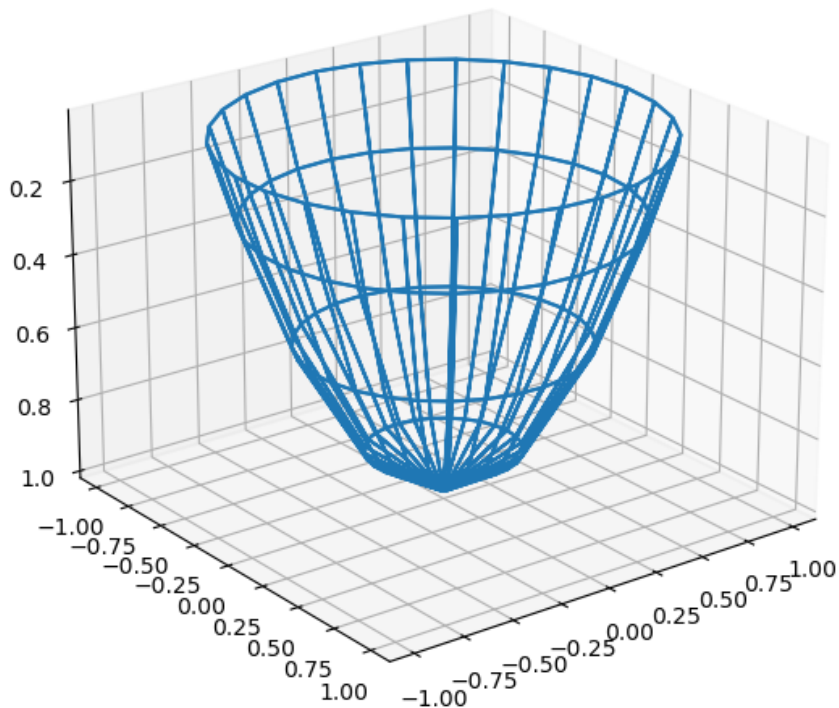


Рисунок 5 – Пример построения 3d-каркаса фигуры при помощи Python

Matplotlib - самая популярная библиотека Python для визуализации данных. Это библиотека для построения 2D-графиков, которой уже более 10 лет и которая поставляется с интерактивной платформой. Можно использовать данную библиотеку для различных целей, таких как создание графиков, гистограмм, спектров мощности, ствол-лист диаграмм, круговых диаграмм и многого другого. Самое лучшее в Matplotlib - это то, что нужно написать несколько строчек кода, а остальное обработается самостоятельно.

Литература

1. Сысоева М.В. Программирование для нормальных с нуля на языке Python / Сысоев И.В. -1-е изд.- Москва: Базальт СПО; МАКС Пресс, 2018. - 180 с.