**Сравнение различных библиотек для визуализации данных: Matplotlib, Seaborn и Plotly**: Создать набор визуализаций с использованием Matplotlib, Seaborn и Plotly, сравнить их функциональность и удобство использования.

**Выполнил студент группы Python-разработчик от 27.05:**

**Шагжиева Вероника Андреевна**

**2024**

**СОДЕРЖАНИЕ**

1. Введение……………………………………………………………………..3 стр  
2. Основные понятия и определения …………………………………………4 стр  
3. Описание библиотек.......................................................................................5 стр

3.1 Matplotlib.................................................................................................5-6 стр

3.2 Seaborn....................................................................................................7-8 стр

3.3 Plotly......................................................................................................... 9 стр

4. Функциональность и удобство использования библиотек........................10 стр

5. Сравнение библиотек...............................................................................11-17 стр

6. Результат сравнения библиотек...................................................................18 стр

7. Заключение…………………………………………………………………19 стр

**ВВЕДЕНИЕ**

“Один рисунок стоит 1000 слов” (Роберт Дж. Барнард)

Эта фраза как нельзя лучше говорить о важности визуализации для специалиста по данным или любого, кто работает с данными. Набор данных, маленький и большой, будучи представленным в виде изображения, графика или диаграммы, позволяет намного больше понять специалисту по анализу другим людям. Визуализация данных играет важную роль в появлении как малых, так и больших объемов данных.

Один из ключевых навыков специалиста по данным - это умение убедительно рассказать историю. Сперва для написания этой истории нужно изучить данные, поэтому аналитик визуализирует данные во многих формах, пока необработанные данные не начнут обретать какой-то смысл. Паттерны в данных лучше видны именно при визуализации

Следовательно, изучение программных инструментов для визуализации данных позволяет нам извлекать информацию, лучше понимать данные и принимать более эффективные решения.

В Python имеется богатый набор инструментов для задач в визуализации данных всех видов и мастей. Существует много библиотек, которые позволяют создавать всевозможные изображения и диаграммы.

В этой работе, я рассмотрю некоторые из библиотек для визуализации данных.

Цель работы:

Провести сравнение различных библиотек для визуализации данных.

Задачи

1. Изучить библиотеки для визуализации данных
2. Создать визуализации с использованием каждой библиотеки
3. Провести сравнительный анализ созданных визуализаций
4. Провести сравнение их функциональности и удобства использования

**2.Основные понятия и определения**

**Визуализация данных** — это представление информации в графическом формате с целью сделать ее более доступной для анализа или интерпретации.

**Matplotlib** — наиболее распространённая и стандартная библиотека в Python для создания статических графиков. Она предлагает широкий набор инструментов для построения различных графиков и диаграмм, включая линейные, столбчатые, круговые, точечные, гистограммы, графики распределений и другие.

**Seaborn** — библиотека, основанная на Matplotlib, которая предоставляет дополнительные функции и улучшенный дизайн графиков. Seaborn упрощает создание сложных визуализаций и предлагает встроенные темы, которые делают графики более привлекательными.

**Plotly** — библиотека для создания интерактивных визуализаций данных. Она позволяет создавать графики, которые пользователи могут исследовать, изменять и делиться с другими. Plotly поддерживает множество типов графиков, включая линейные графики, столбчатые диаграммы, круговые диаграммы и тепловые карты.

**3.Описание библиотек**

**3.1 Matplotlib**

Matplotlib — это одна из самых популярных библиотек для визуализации данных в Python. Она позволяет создавать разнообразные графики и диаграммы, которые помогают лучше понять и интерпретировать данные. Эта библиотека особенно полезна для новичков, так как она интуитивно понятна и имеет богатую документацию. Визуализация данных играет ключевую роль в анализе данных, так как позволяет быстро выявить тенденции, аномалии и взаимосвязи, которые могут быть неочевидны при просмотре сырых данных.

Matplotlib часто используется в сочетании с другими библиотеками, такими как NumPy и Pandas, что делает её мощным инструментом для анализа данных. NumPy предоставляет поддержку для работы с массивами и матрицами, а Pandas — для работы с таблицами данных.

Типы графиков и диаграмм в Matplotlib:

* Диаграмма рассеяния
* Столбчатая диаграмма
* Круговая диаграмма
* Линейный график
* 3D-графики

#### Плюсы

1. **Четко отображены свойства данных**

При анализе данных возможность быстро посмотреть распределение может быть очень полезной.

#### Минусы

Matplotlib может создать любой график, но с его помощью может быть сложно построить или подогнать сложные графики, чтобы они выглядели презентабельно.

**3.2 Seaborn**

Seaborn — это библиотека для визуализации данных на языке Python, построенная на основе matplotlib. Она предоставляет высокоуровневый интерфейс для рисования привлекательных и информативных статистических графиков. Seaborn позволяет легко создавать сложные визуализации с минимальными усилиями, что делает его отличным инструментом для анализа данных. В отличие от matplotlib, Seaborn предлагает более удобные и эстетически приятные способы представления данных, что особенно важно для аналитиков и исследователей данных.

Seaborn также интегрируется с библиотекой pandas, что позволяет легко работать с DataFrame и другими структурами данных. Это делает его идеальным выбором для тех, кто уже знаком с pandas и хочет расширить свои возможности в области визуализации данных. Кроме того, Seaborn включает в себя множество встроенных функций для статистического анализа, таких как регрессия и распределение данных, что делает его мощным инструментом для комплексного анализа данных.

Типы графиков в Seaborn:

* Линейные графики

### Гистограммы

### Диаграммы рассеяния

### Ящик с усами (Boxplot)

#### Плюсы

1. **Меньше кода**

Предоставляет интерфейс более высокого уровня для построения похожих графиков. Другими словами, seaborn обычно строит графики, аналогичные matplotlib, но с меньшим количеством кода и более красивым дизайном

#### **2. Делает стандартные графики красивее**

Многие люди выбирают seaborn для создания широко используемых графиков, таких как столбчатые и прямоугольные диаграммы, расчетные графики, гистограммы и т. д., но не только потому, что это потребует меньше кода, они еще и визуально приятнее.

#### Минусы

Seaborn более ограничен и не имеет такой широкой коллекции графиков, как matplotlib.

**3.3 Plotly**

Plotly — это библиотека визуализации данных с открытым исходным кодом, предоставляющая широкий спектр возможностей интерактивного построения графиков на языке Python. Plotly позволяет создавать интерактивные визуализации, включая линейные графики, точечные диаграммы, гистограммы, тепловые карты, трехмерные графики и многое другое. Благодаря своей гибкости и простоте использования библиотека широко используется в различных областях: анализ данных, научные исследования, финансы, веб-разработка и других.

Виды графиков в Plotly:

* Линейный график
* Гистограмма
* Пузырьковая диаграмма
* Круговая диаграмма
* Поверхностная диаграмма
* Карты

#### Плюсы

1. **Похож на R**

Если вы поклонник графиков в R и вам не хватает его функционала при переходе на Python, Plotly даст вам такое же качество графиков с использованием Python!

**2. Простота создания интерактивных графиков**

Plotly также упрощает создание интерактивных графиков. Интерактивные графики не только красиво выглядят, но и позволяют публике более внимательно изучить каждую точку на графике.

**3. Легко делать сложные графики**

С помощью Plotly достаточно легко создавать сложные графики.

**4. Функциональность и удобство использования библиотек**

**Matplotlib**.

* Позволяет быстро строить диаграммы и графики разных видов, настраивать их оформление.
* С помощью Matplotlib можно строить 2D-фигуры, формировать на основе данных линейные, точечные, столбчатые, круговые и другие диаграммы, рисовать контурные графики, формировать поля векторов и спектрограммы.

**Seaborn**.

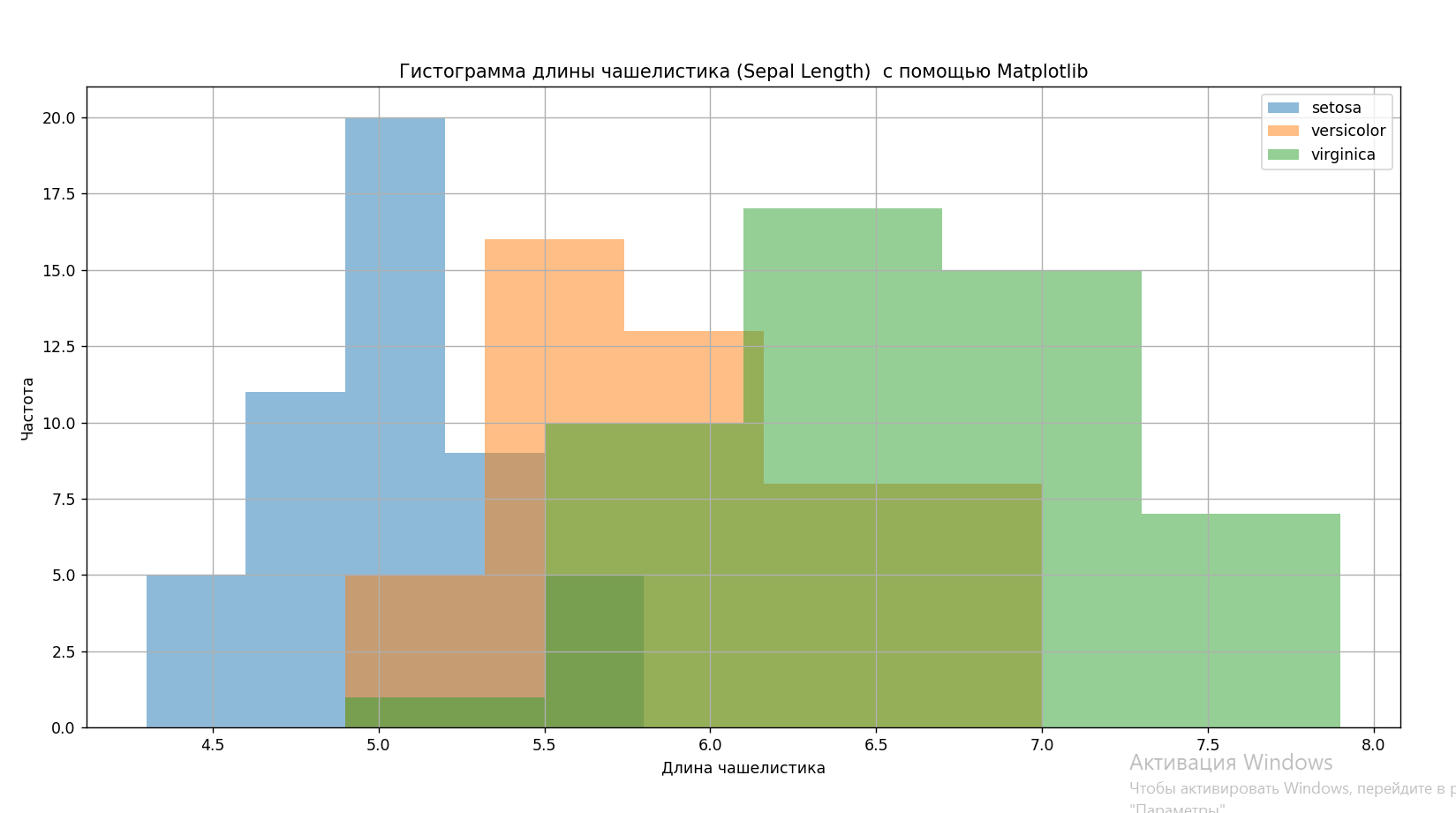
* Библиотека, основанная на Matplotlib, предоставляет дополнительные функции и улучшенный дизайн графиков.
* Seaborn упрощает создание сложных визуализаций и предлагает встроенные темы, которые делают графики более привлекательными.
* С помощью Seaborn можно вычислять и визуализировать корреляции, анализировать и сравнивать данные между собой, строить графики по этому анализу.

**Plotly**.

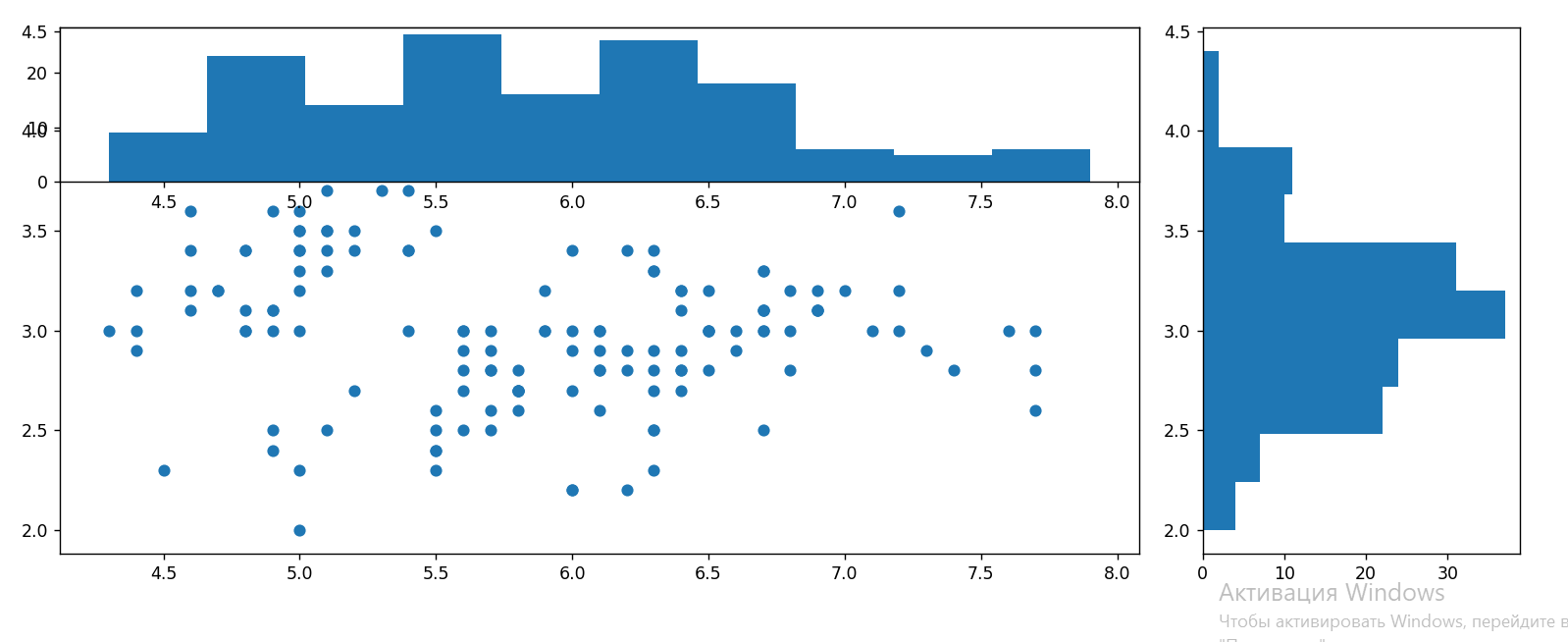
* Библиотека для создания интерактивных визуализаций данных.
* Она позволяет создавать графики, которые пользователи могут исследовать, изменять и делиться с другими.
* Plotly поддерживает множество типов графиков, включая линейные графики, столбчатые диаграммы, круговые диаграммы и тепловые карты.

**5.Сравнение библиотек Matplotlib, Seaborn и Plotly на примере датасет "Iris"**

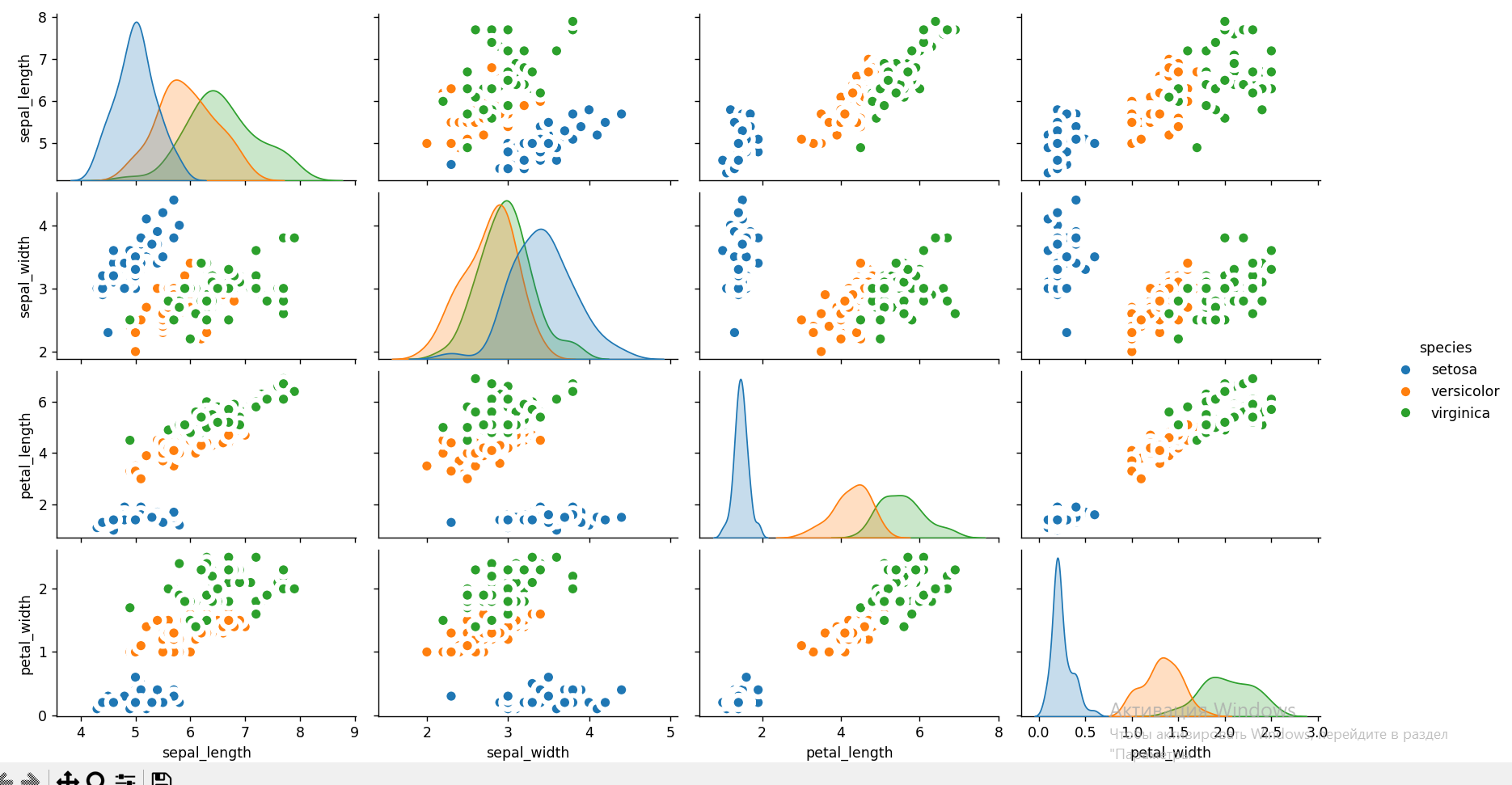
# Импорт необходимых библиотек  
import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt  
import plotly.express as px  
import numpy as np  
from matplotlib.gridspec import GridSpec  
# Загрузка датасета Iris  
iris = sns.load\_dataset('iris')  
iris  
# Визуализация с Matplotlib  
# Гистограмма  
plt.figure(figsize=(5, 5))  
for species in iris['species'].unique():  
 subset = iris[iris['species'] == species]  
 plt.hist(subset['sepal\_length'], alpha=0.5, bins=5, label=species)  
plt.title('Гистограмма длины чашелистика (Sepal Length) c помощью Matplotlib')  
plt.xlabel('Длина чашелистика')  
plt.ylabel('Частота')  
plt.legend()  
plt.grid(True)  
plt.show()



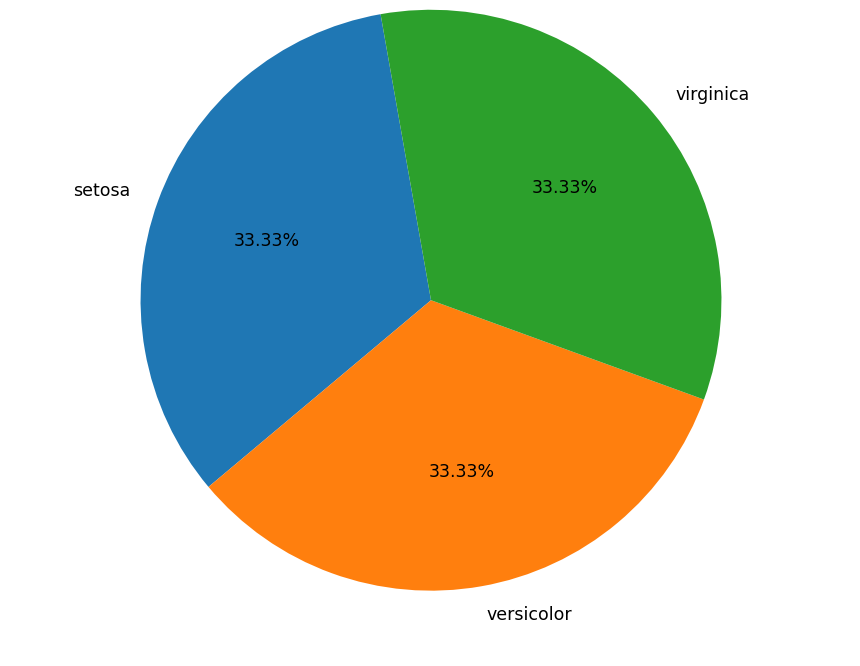
# Диаграмма рассеивания   
fig = plt.figure()  
gs = GridSpec(4, 4)  
ax\_scatter = fig.add\_subplot(gs[1:4, 0:3])  
ax\_hist\_x = fig.add\_subplot(gs[1, 0:3])  
ax\_hist\_y = fig.add\_subplot(gs[1:4, 3])  
ax\_scatter.scatter(iris['sepal\_length'], iris['sepal\_width'])  
ax\_hist\_x.hist(iris['sepal\_length'])  
ax\_hist\_y.hist(iris['sepal\_width'], orientation='horizontal')  
plt.show()



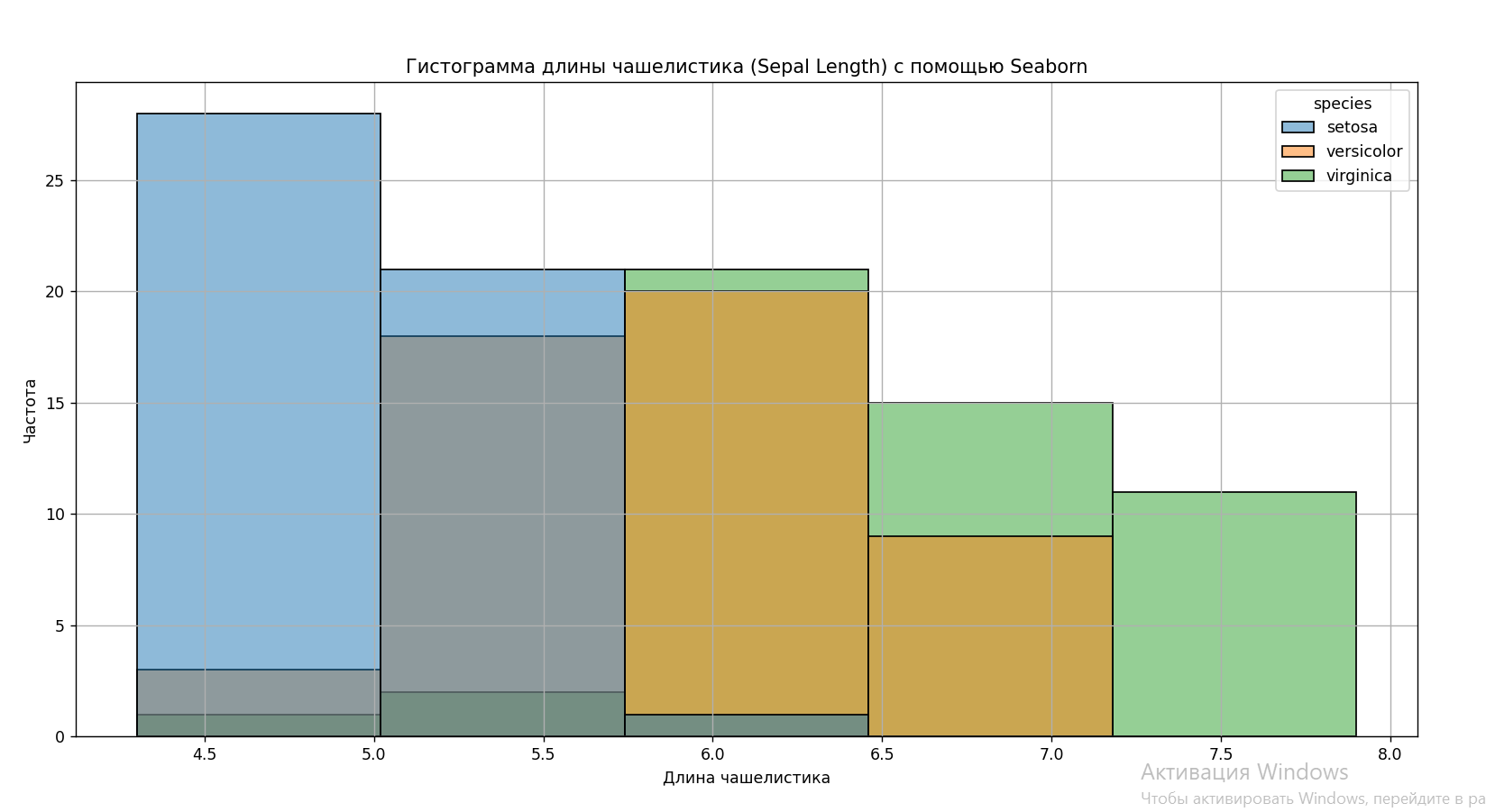
# Парный график  
plt.figure(figsize=(12,6), dpi= 100)  
sns.pairplot(iris, kind="scatter", hue="species", plot\_kws=dict(s=100, edgecolor="white", linewidth=3.5))  
plt.show()



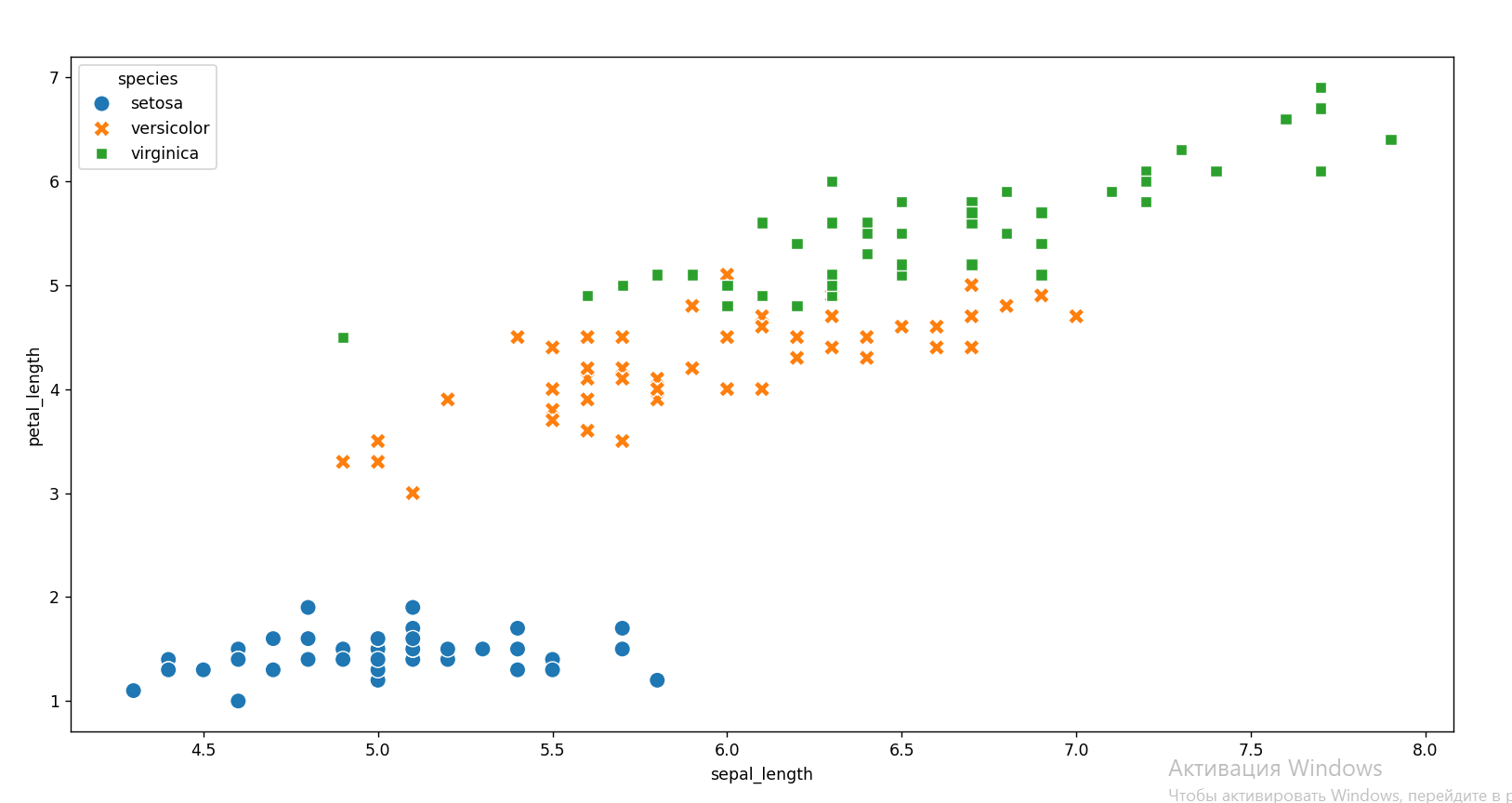
# Круговая диаграмма  
species\_counts = iris['species'].value\_counts()  
plt.figure(figsize=(5,5))  
plt.pie(species\_counts,labels=species\_counts.index, autopct='%2.2f%%', startangle=100)  
plt.show()



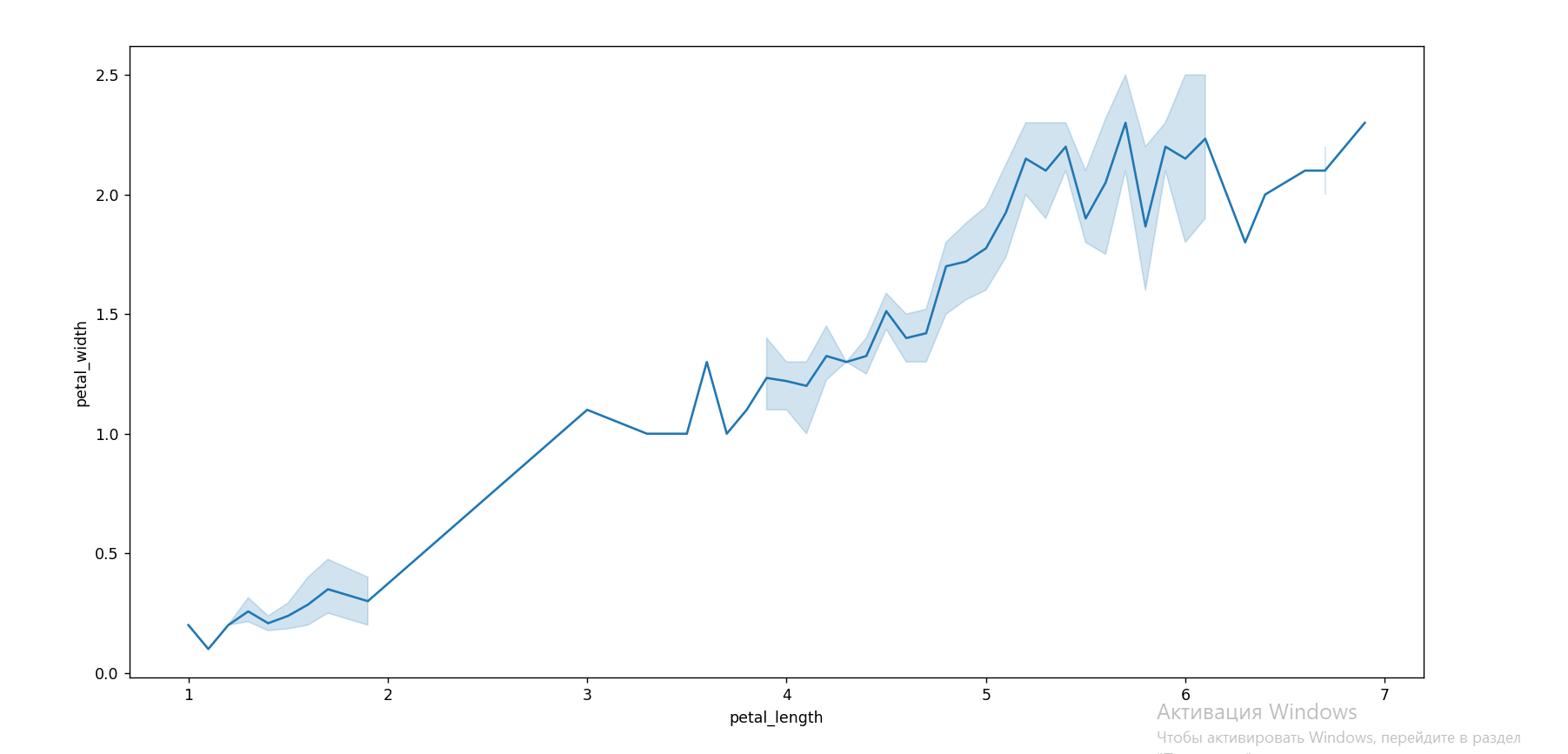
# Везуализация с Seaborn  
# Гистограмма  
plt.figure(figsize=(5, 5))  
sns.histplot(data=iris, x='sepal\_length', hue='species',alpha=0.5, bins=5, kde=False)  
plt.title('Гистограмма длины чашелистика (Sepal Length) с помощью Seaborn')  
plt.xlabel('Длина чашелистика')  
plt.ylabel('Частота')  
plt.grid(True)  
plt.show()



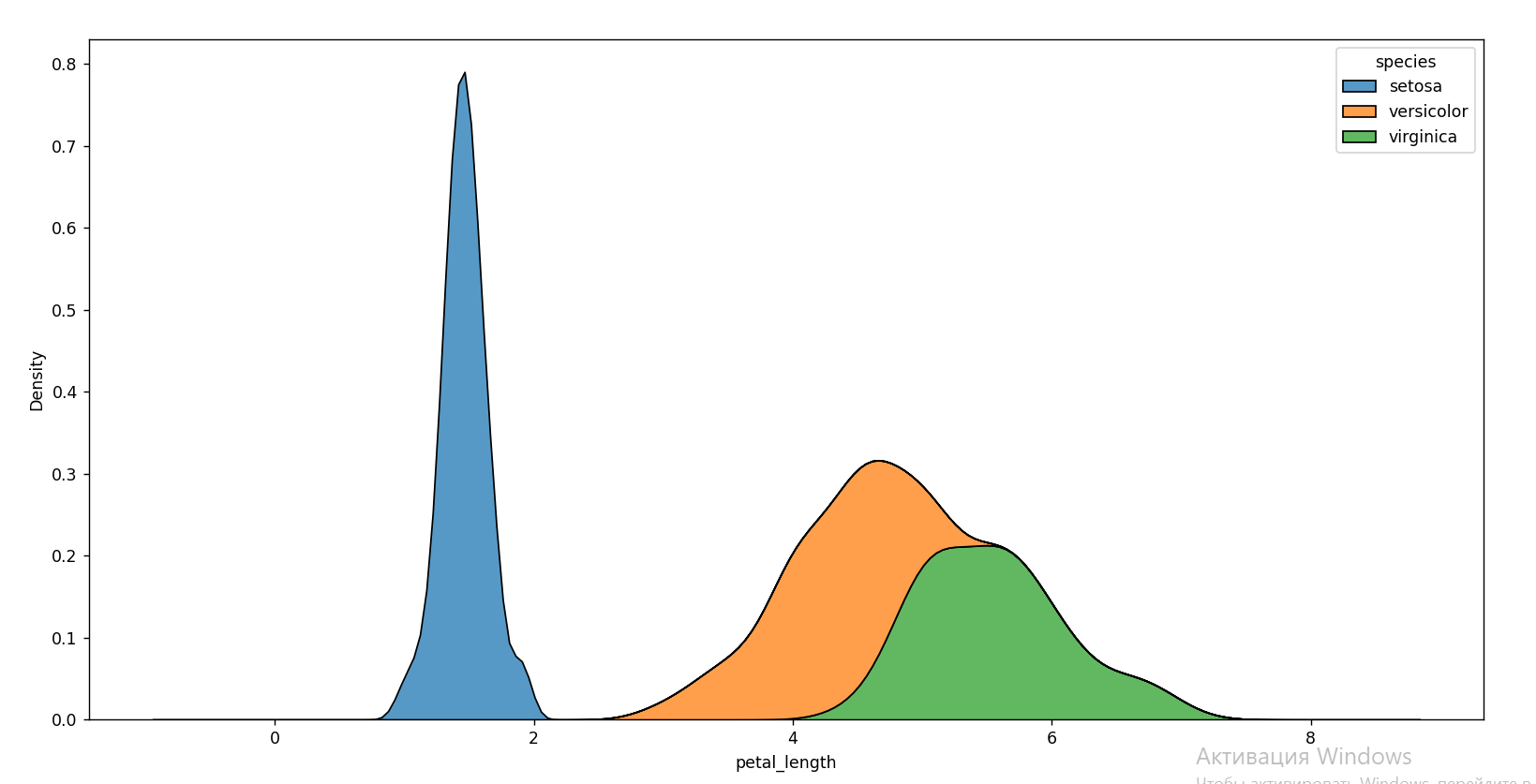
# Диаграмма рассеивания  
sns.scatterplot(x='sepal\_length',y='petal\_length',hue='species',style='species',s=100,data=iris)  
plt.show()



# Линейный график  
sns.lineplot(x='petal\_length',y='petal\_width',data=iris)  
plt.show()



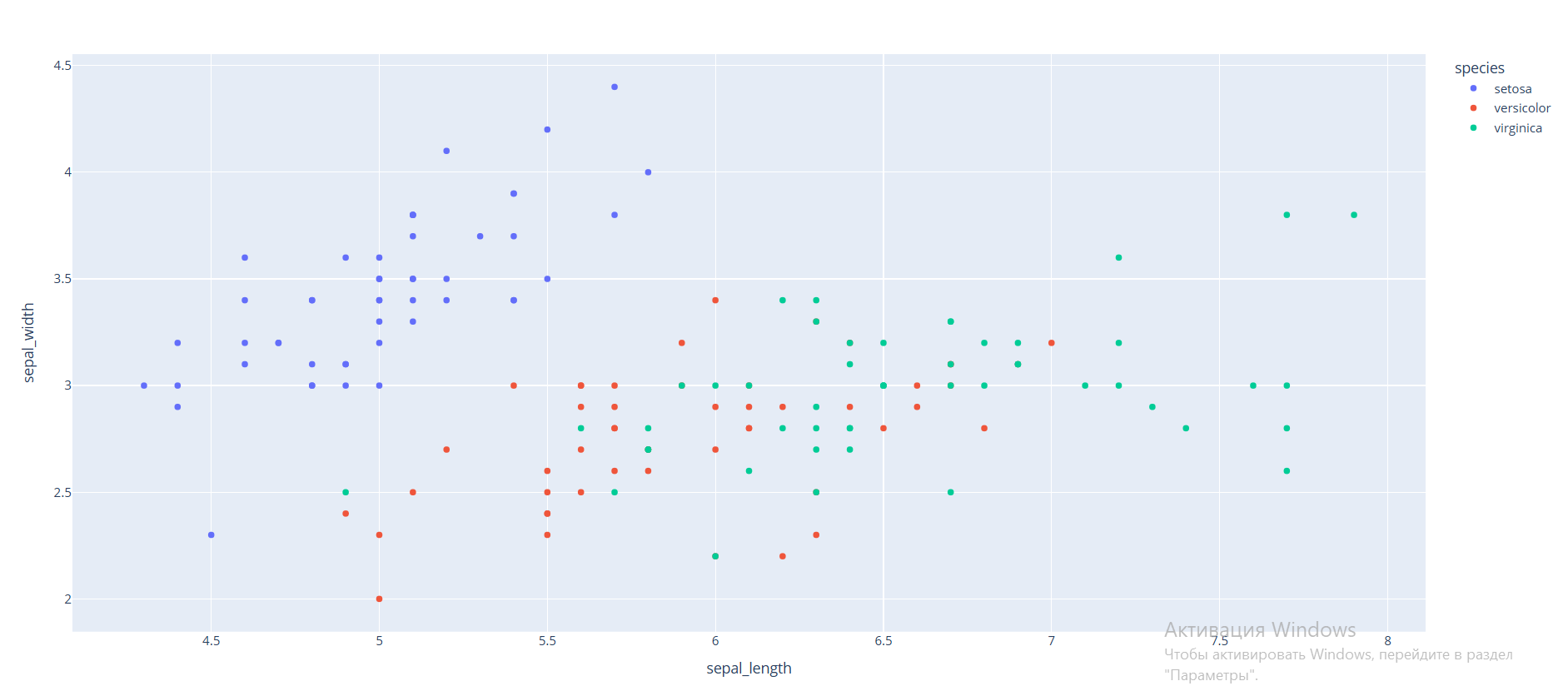
# График плотности  
sns.kdeplot(x='petal\_length',data=iris,hue='species',multiple='stack')  
plt.show()



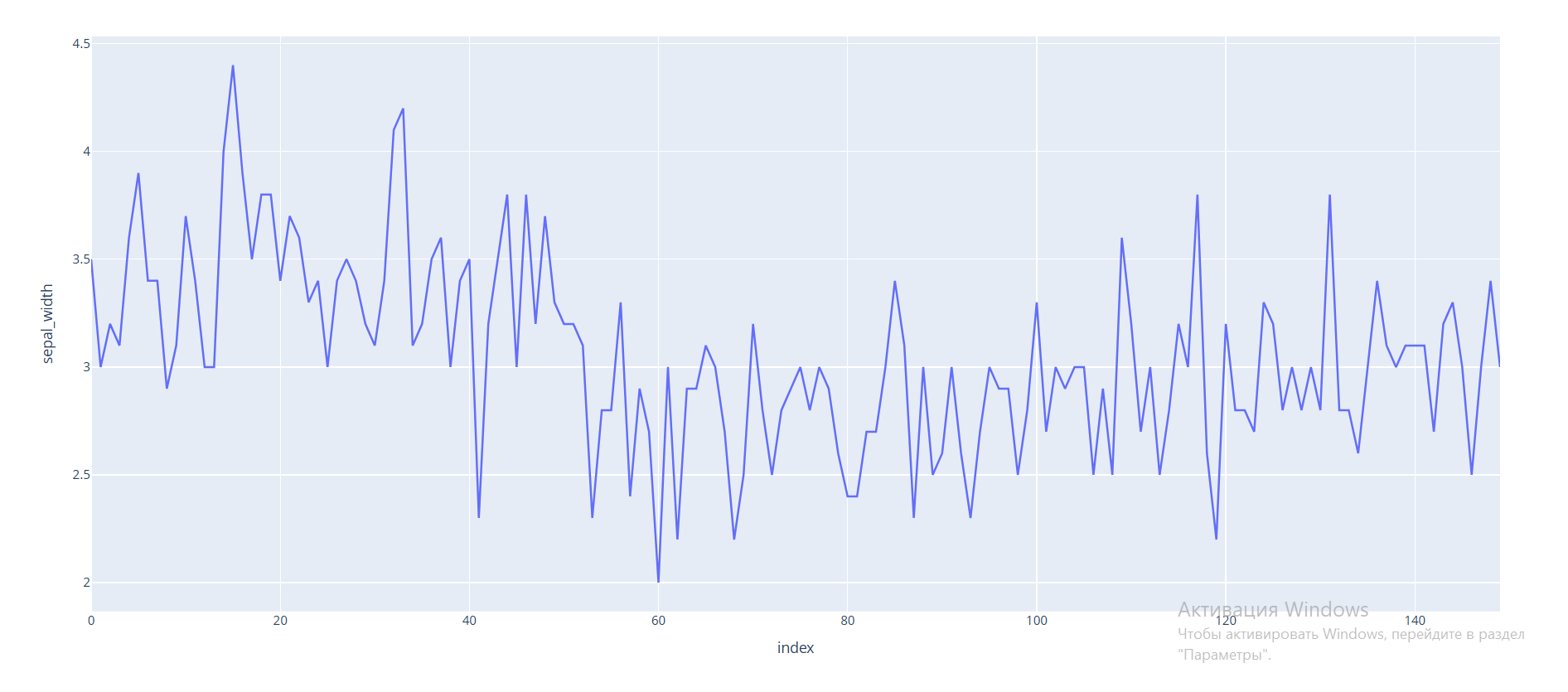
# Визуализация с Plotly  
# Гистограмма

fig = px.histogram(iris, x='sepal\_length', color='species', barmode='overlay',  
 title='Гистограмма длины чашелистика (Sepal Length) - Plotly',  
 labels={'sepal\_length': 'Длина чашелистика', 'count': 'Частота'})  
fig.update\_layout(xaxis\_title='Длина чашелистика', yaxis\_title='Частота')  
fig.show()

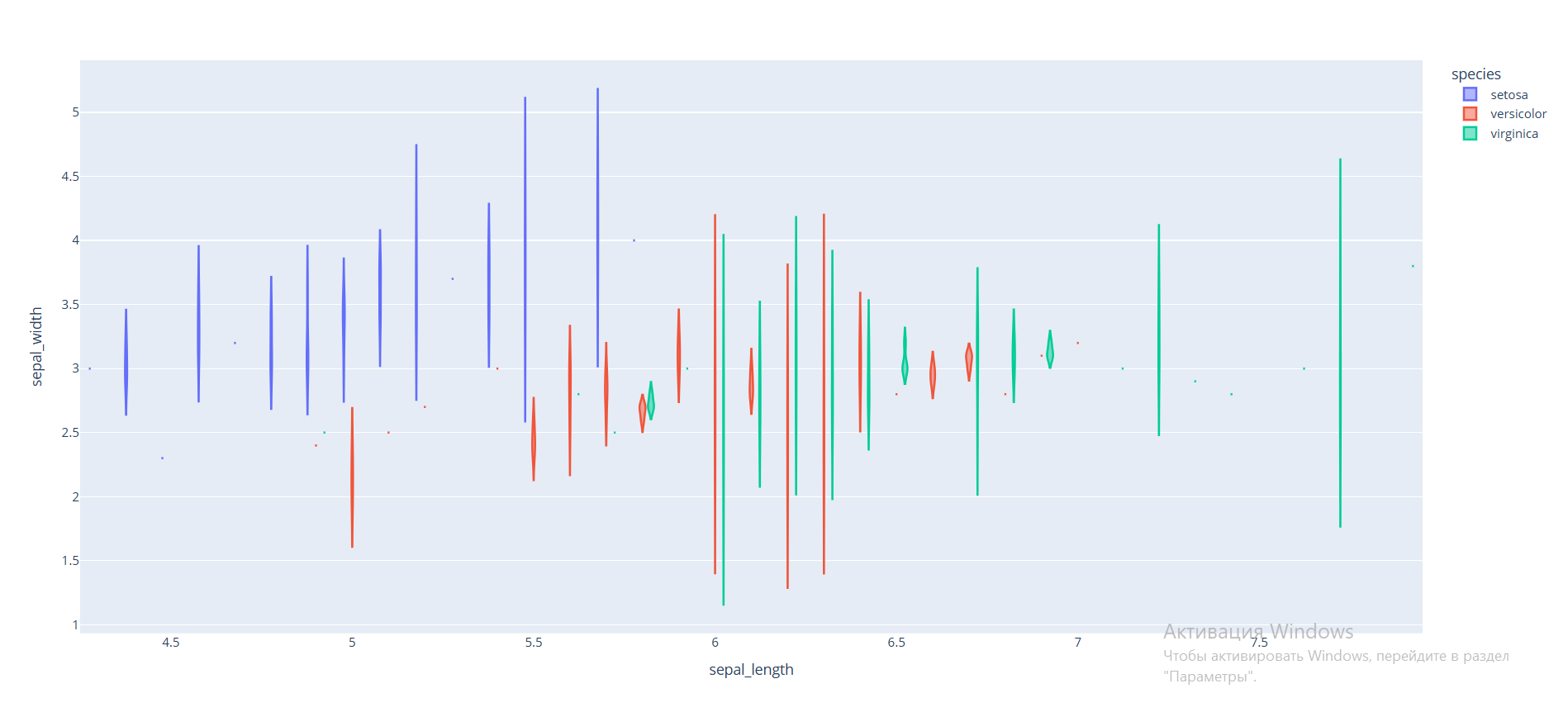
# Диаграмма рассеивания  
fig = px.scatter(iris, x="sepal\_length", y="sepal\_width", color="species")  
fig.show()



# Линейный график  
fig = px.line(iris, y="sepal\_width",)  
fig.show()



# График скрипки  
fig = px.violin(iris, x="sepal\_length", y="sepal\_width", color="species")  
fig.show()



**6.Результат сравнения библиотек**

Я провела сравнение библиотек про трём характеристикам:

* **Интерактивность**: предполагает ли библиотека интерактивные элементы
* **Синтаксис**: какой уровень контроля предлагает библиотека и следует ли она определенной парадигме
* **Основные преимущества и варианты использования**: к какой ситуации библиотека является лучшим выбором.

|  |  |  |  |
| --- | --- | --- | --- |
| Библиотека | Интерактивность | Синтаксис | Основные преимущества и варианты использования |
| **Matplotlib** | Ограничено | Низкоуровневый | Индивидуальный диаграммы |
| **Seaborn** | Ограничено | Высокоуровневым | Быстрый и презентабельные отчеты |
| **Plotly** | Да | Высокоуровневые и низкоуровневый | Коммерческие приложения и информационные панели |

**Заключение**

Проведя сравнение библиотек для визуализации, я сделала вывод что

с помощью Matplotlib можно создать что угодно, но для сложных графиков может потребоваться гораздо больше кода, чем другим библиотекам. Seaborn — это версия Matplotlib более высокого уровня. Несмотря на то, что коллекция графиков не настолько большая, как в Matplotlib, созданные с помощью seaborn широко используемые графики (например, столбчатая диаграмма, прямоугольная диаграмма, график пользовательской активности и т. д.), при меньшем количестве кода будет выглядеть визуально приятнее. Plotly отлично подходит для создания интерактивных и качественных графиков при помощи всего нескольких строк кода.

**Matplotlib, Seaborn и Plotly** — это основные библиотеки Python для визуализации данных. Каждая из них предлагает свои возможности и стиль визуализации, поэтому выбор зависит от предпочтений и задач пользователя.