

Документ подписан посредством электронной подписи  
Информация о владельце:  
ФИО: Шамрай-Курбатова Лидия Викторовна  
Должность: Ректор  
Дата подписания: 11.06.2026 14:05:43  
Уникальный программный ключ:  
b1e4399771b07e18f31755456972d73b2ccfc531

Автономная некоммерческая организация высшего образования  
«Волгоградский институт бизнеса»

## Рабочая программа учебной дисциплины

**Python: основные библиотеки для анализа данных**

(Наименование дисциплины)

**54.03.01 Дизайн, направленность (профиль) «Цифровой дизайн»**

(Направление подготовки / Профиль)

**Бакалавр**

(Квалификация)

**Кафедра разработчик**

**Экономики и управления**

**Год набора**

**2026**

Вид учебной деятельности	Трудоемкость (объем) дисциплины	
	Очная форма	Очно-заочная форма
	Д	В
Зачетные единицы	4	4
Общее количество часов	144	144
Аудиторные часы контактной работы обучающегося с преподавателями:	64	38
– Лекционные (Л)	16	10
– Практические (ПЗ)	16	10
– Лабораторные (ЛЗ)	32	18
– Семинарские (СЗ)		
Самостоятельная работа обучающихся (СРО)	80	106
К (Р-Г) Р (П) (+;-)		
Тестирование (+;-)		
ДКР (+;-)		
Зачет (+;-)	+	+
Зачет с оценкой (+;- (Кол-во часов))		
Экзамен (+;- (Кол-во часов))		

Волгоград 2026

## Содержание

Раздел 1. Организационно-методический раздел .....	3
Раздел 2. Тематический план.....	5
Раздел 3. Содержание дисциплины.....	6
Раздел 4. Организация самостоятельной работы обучающихся.....	13
Раздел 5. Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся.....	16
Раздел 6. Оценочные средства промежуточной аттестации (с ключами) .....	19
Раздел 7. Перечень учебной литературы, необходимой для освоения дисциплины .....	19
Раздел 8. Материально-техническая база и информационные технологии.....	23
Раздел 9. Методические указания для обучающихся по освоению дисциплины .....	25

## Раздел 1. Организационно-методический раздел

### 1.1. Цели освоения дисциплины

Дисциплина «Python: основные библиотеки для анализа данных» входит в перечень **Обязательных дисциплин** подготовки обучающихся по направлению **Дизайн, направленность (профиль) «Цифровой дизайн»**.

Целью дисциплины является формирование **компетенций** (в соответствии с ФГОС ВО и требованиями к результатам освоения основной профессиональной образовательной программы высшего образования (ОПОП ВО)):

**ОПК-2. Способен работать с научной литературой; собирать, анализировать и обобщать результаты научных исследований; оценивать полученную информацию; самостоятельно проводить научно-исследовательскую работу; участвовать в научно-практических конференциях.**

**Дескрипторы общепрофессиональных компетенций:**

**ОПК-2.1** Способен собирать, анализировать и обобщать результаты исследований в области графического и цифрового дизайна, включая данные пользовательской аналитики и А/В-тестирования, включая анализ целесообразности применения методов искусственного интеллекта, формирование требований к данным и метрикам качества моделей.

**ОПК-2.2** Способен проводить прикладные исследования (аудит пользовательских сценариев, конкурентный анализ), выполнять постановку целей, разработку концепции системы, разработку технического задания на создание цифрового продукта, в том числе с помощью ИИ-систем; участвовать в научных и профессиональных конференциях по дизайну и цифровым инновациям.

Перечисленные компетенции формируются в процессе достижения **индикаторов компетенций**:

Обобщенная трудовая функция/ трудовая функция	Код и наименование дескриптора компетенций	Код и наименование индикатора достижения компетенций (из ПС)
<b>ОПК-2.</b> Способен работать с научной литературой; собирать, анализировать и обобщать результаты научных исследований; оценивать полученную информацию; самостоятельно проводить научно-исследовательскую работу; участвовать в научно-практических конференциях.	<b>ОПК-2.1</b> Способен собирать, анализировать и обобщать результаты исследований в области графического и цифрового дизайна, включая данные пользовательской аналитики и А/В-тестирования, включая анализ целесообразности применения методов искусственного интеллекта, формирование требований к данным и метрикам качества моделей  <b>ОПК-2.2</b> Способен проводить прикладные исследования (аудит пользовательских сценариев, конкурентный анализ), выполнять постановку целей, разработку концепции системы, разработку технического задания на создание цифрового продукта, в	<b>Знание:</b> ИД-1 ОПК-2.1 Методика поиска, сбора и анализа информации, необходимой для разработки проектного задания на создание объектов визуальной информации, идентификации и коммуникации В/01.6 ИД-2 ОПК-2.2 Подходы к постановке целей, разработке концепции системы и технического задания на создание цифрового продукта, в том числе с помощью ИИ-систем (без привязки к профессиональному стандарту) <b>Умения:</b> ИД-3 ОПК-2.1 Использовать специальные компьютерные программы для проектирования объектов визуальной информации, идентификации и коммуникации В/02.6 ИД-4 ОПК-2.2 Использовать подходы к постановке целей, разработке концепции системы и технического задания на создание цифрового продукта, в том числе с помощью ИИ-систем (без привязки к профессиональному стандарту) <b>Навыки и (или)опыт деятельности:</b> ИД-5 ОПК-2.1 Изучение информации, необходимой для работы над дизайн-проектом

	том числе с помощью ИИ-систем; участвовать в научных и профессиональных конференциях по дизайну и цифровым инновациям	объекта визуальной информации, идентификации и коммуникации В/02.6 ИД-6 ОПК-2.2 Владение навыками постановки целей, разработки концепции системы и технического задания на создание цифрового продукта, в том числе с помощью ИИ-систем (без привязки к профессиональному стандарту)
--	---	---

**1.2. Место дисциплины в структуре ОПОП ВО  
направления подготовки 54.03.01 Дизайн, направленность (профиль) «Цифровой дизайн»**

№	Предшествующие дисциплины (дисциплины, изучаемые параллельно)	Последующие дисциплины
1	2	3
1	Математический анализ и моделирование	Машинное обучение
2		Производственная практика (Проектно-технологическая практика)
3		Производственная практика (Преддипломная практика)
4		Подготовка к процедуре защиты и защита выпускной квалификационной работы

*Последовательность формирования компетенций в указанных дисциплинах может быть изменена в зависимости от формы и срока обучения, а также преподавания с использованием дистанционных технологий обучения.*

**1.3. Нормативная документация**

Рабочая программа учебной дисциплины составлена на основе:

- Федерального государственного образовательного стандарта высшего образования по направлению подготовки **54.03.01 Дизайн**;
- Учебного плана направления подготовки **54.03.01 Дизайн, направленность (профиль) «Цифровой дизайн»** 2026 годов набора;
- Образца рабочей программы учебной дисциплины (приказ № 113-О от 01.09.2021 г.).

## Раздел 2. Тематический план

### Очная форма обучения (полный срок)

№	Тема дисциплины	Трудоемкость				Код индикатора и дескриптора достижения компетенций
		Всего	Аудиторные занятия		СРО	
			Л	ПЗ (ЛЗ, СЗ)		
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>
1	Введение в экосистему анализа данных на Python: NumPy и основы векторизации	18	2	6	10	ИД-1 ОПК- 2.1 ИД-2 ОПК- 2.2
2	Работа с табличными данными: библиотека Pandas (Series, DataFrame, индексация, фильтрация)	18	2	6	10	ИД-1 ОПК- 2.1 ИД-2 ОПК- 2.2
3	Очистка и подготовка данных с Pandas: обработка пропусков, дубликатов, выбросов и преобразование типов	18	2	6	10	ИД-3 ОПК- 2.1 ИД-4 ОПК- 2.2
4	Агрегирование, группировка и слияние данных: groupby, merge, concat, pivot_table	18	2	6	10	ИД-3 ОПК- 2.1 ИД-4 ОПК- 2.2
5	Визуализация данных: Matplotlib и Seaborn (основные типы графиков, настройка стилей, интерпретация)	18	2	6	10	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2
6	Работа с временными рядами: DatetimeIndex, ресемплинг, сдвиги, скользящие окна	18	2	6	10	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2
7	Введение в статистический анализ и моделирование с использованием SciPy и Statsmodels	18	2	6	10	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2
8	Построение пайплайнов обработки данных и базовых моделей с Scikit-learn	18	2	6	10	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2
<b>Вид промежуточной аттестации (Зачет)</b>						
<b>Итого</b>		<b>144</b>	<b>16</b>	<b>48</b>	<b>80</b>	

### Очно-заочная форма обучения (полный срок)

№	Тема дисциплины	Трудоемкость				Код индикатора и дескриптора достижения компетенций
		Всего	Аудиторные занятия		СРО	
			Л	ПЗ (ЛЗ, СЗ)		
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>
1	Введение в экосистему анализа данных на Python: NumPy и основы векторизации	18	1	4	13	ИД-1 ОПК- 2.1 ИД-2 ОПК- 2.2
2	Работа с табличными данными: библиотека Pandas (Series, DataFrame, индексация, фильтрация)	18	1	4	13	ИД-1 ОПК- 2.1 ИД-2 ОПК- 2.2
3	Очистка и подготовка данных с Pandas: обработка пропусков, дубликатов, выбросов и преобразование типов	18	1	4	13	ИД-3 ОПК- 2.1 ИД-4 ОПК- 2.2

4	Агрегирование, группировка и слияние данных: <code>groupby</code> , <code>merge</code> , <code>concat</code> , <code>pivot_table</code>	18	1	4	13	ИД-3 ОПК- 2.1 ИД-4 ОПК- 2.2
5	Визуализация данных: <code>Matplotlib</code> и <code>Seaborn</code> (основные типы графиков, настройка стилей, интерпретация)	18	1	4	13	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2
6	Работа с временными рядами: <code>DatetimeIndex</code> , ресемплинг, сдвиги, скользящие окна	18	1	4	13	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2
7	Введение в статистический анализ и моделирование с использованием <code>SciPy</code> и <code>Statsmodels</code>	18	2	2	14	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2
8	Построение пайплайнов обработки данных и базовых моделей с <code>Scikit-learn</code>	18	2	2	14	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2
<b>Вид промежуточной аттестации (Зачет)</b>						
<b>Итого</b>		<b>144</b>	<b>10</b>	<b>28</b>	<b>106</b>	

### Раздел 3. Содержание дисциплины

#### 3.1. Содержание дисциплины

**Тема 1. Введение в экосистему анализа данных на Python: NumPy и основы векторизации.** Обзор экосистемы Python для анализа данных: ключевые библиотеки (`NumPy`, `Pandas`, `Matplotlib`, `SciPy`, `Scikit-learn`) и их взаимодействие. Понятие многомерного массива `ndarray` в `NumPy`: атрибуты (`shape`, `dtype`, `size`, `ndim`), создание массивов (`array`, `zeros`, `ones`, `arange`, `linspace`, `random`). Индексирование и срезы массивов, булево индексирование. Основные операции над массивами: поэлементная арифметика, универсальные функции (`ufunc`). Понятие векторизации вычислений и её преимущества перед циклами. Правила трансляции (`broadcasting`) для массивов разной формы. Основы линейной алгебры в `NumPy`: умножение матриц, вычисление определителя, обращение матриц. Генерация случайных чисел и работа с `random`-модулем.

**Тема 2. Работа с табличными данными: библиотека Pandas (Series, DataFrame, индексация, фильтрация).** Введение в `Pandas`: назначение и основные структуры данных. `Series`: создание, атрибуты, индексация, операции. `DataFrame`: создание из словарей, списков, CSV-файлов. Методы просмотра данных: `head`, `tail`, `info`, `describe`. Работа с индексами: установка, сброс, переименование. Доступ к данным: `loc` (по меткам) и `iloc` (по целочисленным позициям). Булева фильтрация строк по условиям. Добавление, удаление и изменение столбцов. Применение функций к данным: `map`, `apply`, `applymap`. Сортировка данных по значениям и индексам. Базовые статистические методы `DataFrame`: `mean`, `median`, `std`, `min`, `max`, `quantile`.

**Тема 3. Очистка и подготовка данных с Pandas: обработка пропусков, дубликатов, выбросов и преобразование типов.** Выявление и обработка пропущенных данных: методы `isnull`, `notnull`, `dropna` (удаление строк/столбцов с пропусками), `fillna` (заполнение константой, средним, медианой, методом `ffill/bfill`). Интерполяция пропусков (`interpolate`). Работа с дубликатами: `duplicated`, `drop_duplicates`. Выявление выбросов: методы IQR (межквартильный размах) и Z-оценка. Преобразование типов данных: `astype`, `to_numeric`, `to_datetime`. Категориальные данные: тип `category`, преимущества использования. Замена значений: `replace`, `map`. Удаление столбцов и строк с низкой вариативностью. Нормализация имен столбцов. Объединение нескольких `DataFrame`.

**Тема 4. Агрегирование, группировка и слияние данных: groupby, merge, concat, pivot\_table.** Группировка данных: метод `groupby`, принцип `split-apply-combine`. Итерация по группам. Агрегирующие функции: `mean`, `sum`, `count`, `min`, `max`, `std`. Применение нескольких функций

одновременно с `agg`. Трансформация и фильтрация групп (`transform`, `filter`). Создание сводных таблиц (`pivot_table`) и кросс-таблиц (`crosstab`). Объединение `DataFrame`: `concat` (конкатенация по строкам и столбцам), `merge` (слияние по ключам, типы соединений: `inner`, `left`, `right`, `outer`). Использование параметров `on`, `left_on`, `right_on`, `how`, `suffixes`. Объединение по индексу (`join`). Применение группировки и слияния для решения аналитических задач.

**Тема 5. Визуализация данных: Matplotlib и Seaborn (основные типы графиков, настройка стилей, интерпретация).** Философия визуализации данных. Библиотека `Matplotlib`: фигура (`figure`) и оси (`axes`). Основные типы графиков: `line plot` (линейный), `scatter plot` (точечная диаграмма), `bar chart` (столбчатая диаграмма), `histogram` (гистограмма), `boxplot` (ящик с усами). Настройка внешнего вида: заголовки, подписи осей, легенда, сетка, цвета, стили линий, маркеры. Создание нескольких подграфиков (`subplots`). Библиотека `Seaborn`: высокоуровневый интерфейс. Установка стилей и цветовых палитр. Основные функции: `histplot`, `kdeplot`, `boxplot`, `violinplot`, `swarmplot`, `scatterplot`, `heatmap` (корреляционная матрица), `pairplot` (матрица парных графиков). Интерпретация визуализаций для анализа распределений, зависимостей и выбросов.

**Тема 6. Работа с временными рядами: DatetimeIndex, ресемплинг, сдвиги, скользящие окна.** Представление дат и времени в `Pandas`: тип `datetime64`. Создание `DatetimeIndex`. Парсинг дат при загрузке данных (`parse_dates`). Основные методы доступа к компонентам даты: `dt.year`, `dt.month`, `dt.day`, `dt.dayofweek`. Индексирование и фильтрация по временным меткам и диапазонам. Ресемплинг временных рядов: изменение частоты дискретизации с помощью `resample`, методы агрегации (`mean`, `sum`, `asfreq`, `ffill`, `bfill`). Сдвиги во времени: `shift` (сдвиг значений), `tshift` (сдвиг индекса). Разности между наблюдениями: `diff`. Скользящие окна (`rolling`): вычисление скользящего среднего, скользящей суммы, скользящего стандартного отклонения. Расширяющиеся окна (`expanding`). Применение для сглаживания и анализа трендов.

**Тема 7. Введение в статистический анализ и моделирование с использованием SciPy и Statsmodels.** Библиотека `SciPy`: модуль `stats` для статистических расчетов. Основные описательные статистики: `describe`, `skew`, `kurtosis`. Проверка статистических гипотез: `t`-критерий Стьюдента (`ttest_ind`, `ttest_1samp`), критерий хи-квадрат (`chisquare`), критерий Манна-Уитни (`mannwhitneyu`). Корреляционный анализ: расчет коэффициентов корреляции Пирсона и Спирмена (`pearsonr`, `spearmanr`), визуализация корреляционной матрицы. Библиотека `Statsmodels`: построение статистических моделей. Простая и множественная линейная регрессия (`OLS`). Интерпретация результатов: коэффициенты, `p`-значения, `R`-квадрат, `F`-статистика. Диагностика регрессионных моделей: анализ остатков.

**Тема 8. Построение пайплайнов обработки данных и базовых моделей с Scikit-learn.** Введение в `Scikit-learn`: единый API для предобработки и моделирования. Разделение данных на обучающую и тестовую выборки (`train_test_split`). Предобработка признаков: масштабирование (`StandardScaler`, `MinMaxScaler`), нормализация (`Normalizer`). Кодирование категориальных признаков (`OneHotEncoder`, `OrdinalEncoder`). Создание пайплайнов (`Pipeline`) для последовательной обработки данных и обучения модели. Применение `ColumnTransformer` для раздельной обработки разных типов признаков. Базовые алгоритмы машинного обучения: линейная регрессия, логистическая регрессия, дерево решений, метод `k`-ближайших соседей. Оценка качества моделей: метрики для регрессии (`MSE`, `MAE`, `R2`) и классификации (`accuracy`, `precision`, `recall`, `F1`). Кросс-валидация (`cross_val_score`). Сохранение и загрузка моделей (`joblib`, `pickle`).

### 3.2. Содержание практического блока дисциплины

#### Очная форма обучения (полный срок)

№	Тема практического (семинарского, лабораторного) занятия
1	2
ПЗ 1	Установка и настройка среды: Anaconda, Jupyter Notebook, VS Code. Импорт библио-

	<p>теки NumPy. Создание одномерных и многомерных массивов с использованием <code>array</code>, <code>zeros</code>, <code>ones</code>, <code>arange</code>, <code>linspace</code>, <code>random.randn</code>. Изучение атрибутов массивов: <code>shape</code>, <code>dtype</code>, <code>size</code>, <code>ndim</code>. Выполнение операций индексирования и срезов, извлечение подмассивов. Булево индексирование: фильтрация элементов по условию. Выполнение поэлементных арифметических операций над массивами. Применение универсальных функций (<code>ufunc</code>): <code>sqrt</code>, <code>exp</code>, <code>sin</code>, <code>log</code>. Сравнение скорости векторизованных вычислений с поэлементными циклами на больших массивах. Решение задач на трансляцию (<code>broadcasting</code>): сложение массива с числом, сложение строки с матрицей. Вычисление матричных операций: умножение матриц (<code>dot</code>, <code>matmul</code>), вычисление определителя (<code>linalg.det</code>), обращение матриц (<code>linalg.inv</code>). Генерация выборок случайных чисел (<code>normal</code>, <code>uniform</code>, <code>binomial</code>). Визуализация сгенерированных данных с помощью <code>Matplotlib</code>.</p>
ПЗ 2	<p>Импорт библиотеки <code>Pandas</code>. Создание <code>Series</code> из списка, словаря и массива NumPy. Создание <code>DataFrame</code> из словаря списков, из списка словарей, из CSV-файла (<code>read_csv</code>). Использование методов <code>head</code>, <code>tail</code>, <code>info</code>, <code>describe</code> для первичного исследования данных. Установка и сброс индекса (<code>set_index</code>, <code>reset_index</code>). Выборка данных с помощью <code>loc</code> (по меткам индекса) и <code>iloc</code> (по целочисленным позициям). Фильтрация строк по условиям с использованием булевых масок (например, <code>df[df['age'] &gt; 30]</code>). Добавление новых столбцов на основе существующих. Удаление столбцов и строк (<code>drop</code>). Применение функций к данным: <code>map</code> для преобразования значений в столбце, <code>apply</code> для строк/столбцов <code>DataFrame</code>, <code>applymap</code> для поэлементного применения. Сортировка данных по значениям (<code>sort_values</code>) и по индексу (<code>sort_index</code>). Вычисление базовых статистик: <code>mean</code>, <code>median</code>, <code>std</code>, <code>min</code>, <code>max</code>, <code>quantile</code>, <code>corr</code>.</p>
ПЗ 3	<p>Загрузка реального «грязного» датасета (например, <code>Titanic</code>, <code>Housing</code>). Обнаружение пропусков с помощью <code>isnull</code>, <code>info</code>, суммирование пропусков. Удаление строк с пропусками (<code>dropna</code>) с различными параметрами (<code>how</code>, <code>thresh</code>, <code>subset</code>). Заполнение пропусков константой, средним, медианой, модой (<code>fillna</code>). Использование метода <code>ffill</code> и <code>bfill</code> для заполнения предыдущими/следующими значениями. Интерполяция пропусков (<code>interpolate</code>). Обнаружение дубликатов (<code>uplicated</code>) и их удаление (<code>drop_duplicates</code>). Выявление выбросов: визуализация с помощью <code>boxplot</code>, расчёт IQR (межквартильного размаха), фильтрация выбросов по условию. Преобразование типов данных: <code>astype</code>, <code>pd.to_datetime</code>, <code>pd.to_numeric</code>. Работа с категориальными данными: преобразование в тип <code>category</code>, методы <code>cat.codes</code>, <code>cat.categories</code>. Замена значений с помощью <code>replace</code> и <code>map</code>. Приведение имен столбцов к единому стилю (<code>lower</code>, <code>strip</code>, <code>replace</code>). Сохранение очищенного датасета в CSV.</p>
ПЗ 4	<p>Группировка данных с помощью <code>groupby</code> по одному и нескольким столбцам. Вычисление агрегирующих статистик (<code>mean</code>, <code>sum</code>, <code>count</code>, <code>min</code>, <code>max</code>, <code>std</code>) для групп. Применение нескольких агрегирующих функций одновременно с <code>agg</code> (например, <code>agg(['mean', 'std'])</code>). Использование <code>transform</code> для добавления агрегированных значений обратно в исходный <code>DataFrame</code>. Фильтрация групп с помощью <code>filter</code>. Создание сводных таблиц (<code>pivot table</code>) с указанием значений, индексов, столбцов и агрегирующей функции. Использование <code>crosstab</code> для построения таблиц сопряженности. Конкатенация <code>DataFrame</code> по строкам (<code>concat</code> с <code>axis=0</code>) и столбцам (<code>axis=1</code>). Слияние <code>DataFrame</code> по ключам с помощью <code>merge</code>: <code>inner join</code>, <code>left join</code>, <code>right join</code>, <code>outer join</code>. Практика на примере объединения информации о клиентах и заказах. Решение аналитической задачи: расчёт выручки по категориям товаров с использованием <code>groupby</code> и <code>merge</code>.</p>
ПЗ 5	<p>Импорт <code>Matplotlib</code> (<code>pyplot</code>) и <code>Seaborn</code>. Построение линейного графика (<code>plot</code>) для временного ряда. Создание точечной диаграммы (<code>scatter</code>) для визуализации зависимости между двумя переменными. Построение столбчатой диаграммы (<code>bar</code>) для сравнения категориальных данных. Создание гистограммы (<code>hist</code>) и ящика с усами (<code>boxplot</code>) для анализа распределения. Настройка внешнего вида: добавление заголовка (<code>title</code>), подписей осей (<code>xlabel</code>, <code>ylabel</code>), легенды (<code>legend</code>), сетки (<code>grid</code>). Изменение цветов, стилей линий, маркеров. Создание нескольких подграфиков в одной фигуре (<code>subplots</code>). Использование <code>Seaborn</code>: установка стиля (<code>set_style</code>) и цветовой палитры (<code>set_palette</code>).</p>

	Построение распределений: <code>histplot</code> , <code>kdeplot</code> . Визуализация взаимосвязей: <code>scatterplot</code> , <code>pairplot</code> (матрица парных графиков). Визуализация корреляционной матрицы с помощью <code>heatmap</code> . Создание сложных многопанельных графиков с <code>FacetGrid</code> . Сохранение графика в файл ( <code>savefig</code> ).
ПЗ 6	Загрузка временного ряда (например, данные о погоде, курсы валют, биржевые котировки). Преобразование столбца с датами в тип <code>datetime</code> с помощью <code>pd.to_datetime</code> . Установка столбца с датами в качестве индекса ( <code>set_index</code> ). Фильтрация данных по временным диапазонам (например, <code>df['2023-01-01':'2023-12-31']</code> ). Извлечение компонентов даты: <code>dt.year</code> , <code>dt.month</code> , <code>dt.day</code> , <code>dt.dayofweek</code> . Ресемплинг временного ряда с изменением частоты ( <code>resample</code> ): переход с дневных данных на месячные ( <code>mean</code> , <code>sum</code> ), на недельные. Использование методов <code>asfreq</code> (простое изменение частоты без агрегации), <code>ffill</code> , <code>bfill</code> для заполнения пропусков при ресемплинге. Сдвиг значений ( <code>shift</code> ) для создания лаговых признаков. Вычисление разностей между последовательными наблюдениями ( <code>diff</code> ). Расчёт скользящего среднего ( <code>rolling(window=7).mean()</code> ) для сглаживания. Расчёт скользящей суммы и стандартного отклонения. Визуализация исходного ряда и сглаженного на одном графике. Расчёт расширяющихся статистик ( <code>expanding</code> ).
ПЗ 7	Импорт модуля <code>stats</code> из <code>SciPy</code> . Вычисление описательных статистик: <code>skew</code> (коэффициент асимметрии), <code>kurtosis</code> (коэффициент эксцесса). Проведение t-критерия Стьюдента для одной выборки: проверка гипотезы о равенстве среднего заданному значению ( <code>ttest_1samp</code> ). Проведение t-критерия для двух независимых выборок ( <code>ttest_ind</code> ). Проведение критерия Манна-Уитни ( <code>mannwhitneyu</code> ) как непараметрической альтернативы. Расчёт коэффициента корреляции Пирсона ( <code>pearsonr</code> ) и Спирмена ( <code>spearmanr</code> ) между двумя переменными, интерпретация p-значения. Импорт <code>statsmodels.api</code> . Построение простой линейной регрессии с использованием OLS: формула $y \sim X$ , обучение модели, получение сводки результатов ( <code>summary</code> ). Интерпретация коэффициентов, p-значений, R-квадрата. Построение множественной линейной регрессии с несколькими предикторами. Анализ остатков: построение графиков остатков, проверка на нормальность ( <code>probplot</code> ). Добавление категориальных переменных с использованием <code>C()</code> .
ПЗ 8	Импорт модулей <code>Scikit-learn</code> . Загрузка встроенного датасета (например, <code>iris</code> , <code>boston</code> , <code>diabetes</code> ). Разделение данных на обучающую и тестовую выборки ( <code>train_test_split</code> ) с параметрами <code>test_size</code> , <code>random_state</code> , <code>stratify</code> . Масштабирование признаков: обучение <code>StandardScaler</code> на обучающей выборке, трансформация обучающей и тестовой выборок. Создание <code>Pipeline</code> для последовательного применения трансформеров (масштабирование) и модели (логистическая регрессия). Использование <code>ColumnTransformer</code> для применения разных трансформеров к разным типам признаков (числовые — масштабирование, категориальные — <code>OneHotEncoder</code> ). Обучение модели линейной регрессии ( <code>LinearRegression</code> ) и оценка качества с помощью <code>mean_squared_error</code> , <code>r2_score</code> . Обучение модели логистической регрессии ( <code>LogisticRegression</code> ) для классификации, оценка с помощью <code>accuracy_score</code> , <code>classification_report</code> . Кросс-валидация с помощью <code>cross_val_score</code> , анализ результатов. Подбор гиперпараметров с помощью <code>GridSearchCV</code> в составе пайплайна. Сохранение обученной модели в файл с помощью <code>joblib.dump</code> и загрузка модели с помощью <code>joblib.load</code> . Применение загруженной модели для предсказания на новых данных.

### Очно-заочная форма обучения (полный срок)

№	Тема практического (семинарского, лабораторного) занятия
1	2
ПЗ 1	Установка и настройка среды: Anaconda, Jupyter Notebook, VS Code. Импорт библиотеки NumPy. Создание одномерных и многомерных массивов с использованием <code>array</code> , <code>zeros</code> , <code>ones</code> , <code>arange</code> , <code>linspace</code> , <code>random.randn</code> . Изучение атрибутов массивов: <code>shape</code> , <code>dtype</code> , <code>size</code> , <code>ndim</code> . Выполнение операций индексирования и срезов, извлечение под-

	<p>массивов. Булево индексирование: фильтрация элементов по условию. Выполнение поэлементных арифметических операций над массивами. Применение универсальных функций (ufunc): sqrt, exp, sin, log. Сравнение скорости векторизованных вычислений с поэлементными циклами на больших массивах. Решение задач на трансляцию (broadcasting): сложение массива с числом, сложение строки с матрицей. Вычисление матричных операций: умножение матриц (dot, matmul), вычисление определителя (linalg.det), обращение матриц (linalg.inv). Генерация выборок случайных чисел (normal, uniform, binomial). Визуализация сгенерированных данных с помощью Matplotlib.</p>
ПЗ 2	<p>Импорт библиотеки Pandas. Создание Series из списка, словаря и массива NumPy. Создание DataFrame из словаря списков, из списка словарей, из CSV-файла (read_csv). Использование методов head, tail, info, describe для первичного исследования данных. Установка и сброс индекса (set_index, reset_index). Выборка данных с помощью loc (по меткам индекса) и iloc (по целочисленным позициям). Фильтрация строк по условиям с использованием булевых масок (например, df[df['age'] &gt; 30]). Добавление новых столбцов на основе существующих. Удаление столбцов и строк (drop). Применение функций к данным: map для преобразования значений в столбце, apply для строк/столбцов DataFrame, applymap для поэлементного применения. Сортировка данных по значениям (sort_values) и по индексу (sort_index). Вычисление базовых статистик: mean, median, std, min, max, quantile, corr.</p>
ПЗ 3	<p>Загрузка реального «грязного» датасета (например, Titanic, Housing). Обнаружение пропусков с помощью isnull, info, суммирование пропусков. Удаление строк с пропусками (dropna) с различными параметрами (how, thresh, subset). Заполнение пропусков константой, средним, медианой, модой (fillna). Использование метода ffill и bfill для заполнения предыдущими/следующими значениями. Интерполяция пропусков (interpolate). Обнаружение дубликатов (duplicated) и их удаление (drop_duplicates). Выявление выбросов: визуализация с помощью boxplot, расчёт IQR (межквартильного размаха), фильтрация выбросов по условию. Преобразование типов данных: astype, pd.to_datetime, pd.to_numeric. Работа с категориальными данными: преобразование в тип category, методы cat.codes, cat.categories. Замена значений с помощью replace и map. Приведение имен столбцов к единому стилю (lower, strip, replace). Сохранение очищенного датасета в CSV.</p>
ПЗ 4	<p>Группировка данных с помощью groupby по одному и нескольким столбцам. Вычисление агрегирующих статистик (mean, sum, count, min, max, std) для групп. Применение нескольких агрегирующих функций одновременно с agg (например, agg(['mean', 'std'])). Использование transform для добавления агрегированных значений обратно в исходный DataFrame. Фильтрация групп с помощью filter. Создание сводных таблиц (pivot_table) с указанием значений, индексов, столбцов и агрегирующей функции. Использование crosstab для построения таблиц сопряженности. Конкатенация DataFrame по строкам (concat с axis=0) и столбцам (axis=1). Слияние DataFrame по ключам с помощью merge: inner join, left join, right join, outer join. Практика на примере объединения информации о клиентах и заказах. Решение аналитической задачи: расчёт выручки по категориям товаров с использованием groupby и merge.</p>
ПЗ 5	<p>Импорт Matplotlib (pyplot) и Seaborn. Построение линейного графика (plot) для временного ряда. Создание точечной диаграммы (scatter) для визуализации зависимости между двумя переменными. Построение столбчатой диаграммы (bar) для сравнения категориальных данных. Создание гистограммы (hist) и ящика с усами (boxplot) для анализа распределения. Настройка внешнего вида: добавление заголовка (title), подписей осей (xlabel, ylabel), легенды (legend), сетки (grid). Изменение цветов, стилей линий, маркеров. Создание нескольких подграфиков в одной фигуре (subplots). Использование Seaborn: установка стиля (set_style) и цветовой палитры (set_palette). Построение распределений: histplot, kdeplot. Визуализация взаимосвязей: scatterplot, pairplot (матрица парных графиков). Визуализация корреляционной матрицы с помощью heatmap. Создание сложных многопанельных графиков с FacetGrid. Сохране-</p>

	ние графика в файл (savefig).
ПЗ 6	Загрузка временного ряда (например, данные о погоде, курсы валют, биржевые котировки). Преобразование столбца с датами в тип datetime с помощью <code>pd.to_datetime</code> . Установка столбца с датами в качестве индекса ( <code>set_index</code> ). Фильтрация данных по временным диапазонам (например, <code>df['2023-01-01':'2023-12-31']</code> ). Извлечение компонентов даты: <code>dt.year</code> , <code>dt.month</code> , <code>dt.day</code> , <code>dt.dayofweek</code> . Ресемплинг временного ряда с изменением частоты ( <code>resample</code> ): переход с дневных данных на месячные ( <code>mean</code> , <code>sum</code> ), на недельные. Использование методов <code>asfreq</code> (простое изменение частоты без агрегации), <code>ffill</code> , <code>bfill</code> для заполнения пропусков при ресемплинге. Сдвиг значений ( <code>shift</code> ) для создания лаговых признаков. Вычисление разностей между последовательными наблюдениями ( <code>diff</code> ). Расчёт скользящего среднего ( <code>rolling(window=7).mean()</code> ) для сглаживания. Расчёт скользящей суммы и стандартного отклонения. Визуализация исходного ряда и сглаженного на одном графике. Расчёт расширяющихся статистик ( <code>expanding</code> ).
ПЗ 7	Импорт модуля <code>stats</code> из <code>SciPy</code> . Вычисление описательных статистик: <code>skew</code> (коэффициент асимметрии), <code>kurtosis</code> (коэффициент эксцесса). Проведение t-критерия Стьюдента для одной выборки: проверка гипотезы о равенстве среднего заданному значению ( <code>ttest_1samp</code> ). Проведение t-критерия для двух независимых выборок ( <code>ttest_ind</code> ). Проведение критерия Манна-Уитни ( <code>mannwhitneyu</code> ) как непараметрической альтернативы. Расчёт коэффициента корреляции Пирсона ( <code>pearsonr</code> ) и Спирмена ( <code>spearmanr</code> ) между двумя переменными, интерпретация p-значения. Импорт <code>statsmodels.api</code> . Построение простой линейной регрессии с использованием OLS: формула $y \sim X$ , обучение модели, получение сводки результатов ( <code>summary</code> ). Интерпретация коэффициентов, p-значений, R-квадрата. Построение множественной линейной регрессии с несколькими предикторами. Анализ остатков: построение графиков остатков, проверка на нормальность ( <code>probplot</code> ). Добавление категориальных переменных с использованием <code>C()</code> .

### 3.3. Образовательные технологии Очная форма обучения (полный срок)

№	Тема занятия	Вид учебного занятия	Форма / Методы интерактивного обучения	% учебного времени
1	2	3	4	5
1	Введение в экосистему анализа данных на Python: NumPy и основы векторизации	ПЗ	Дискуссионные технологии, Работа в малых группах, Взаимопроверка, Мозговой штурм, Интерактивные тренажёры	25
2	Работа с табличными данными: библиотека Pandas (Series, DataFrame, индексация, фильтрация)	ПЗ	Дискуссионные технологии, Групповое решение проблемных задач, Кейс-стади, Взаимообучение, Проектно-ориентированное обучение	25
3	Очистка и подготовка данных с Pandas: обработка пропусков, дубликатов, выбросов и преобразование типов	ПЗ	Работа в парах, Деловая игра, Дискуссионные технологии, Мозговой штурм, Интерактивная визуализация с коллективным обсуждением	25
4	Агрегирование, группировка и слияние данных: <code>groupby</code> , <code>merge</code> , <code>concat</code> , <code>pivot_table</code>	ПЗ	Семинар-дискуссия, Работа в группах с презентацией, Ролевая игра, Интерактивная доска, Кейс-стади	25
5	Визуализация данных:	ПЗ	Групповое решение задач, Кон-	25

	Matplotlib и Seaborn (основные типы графиков, настройка стилей, интерпретация)		курс, Интерактивная визуализация, Мозговой штурм, Проектно-ориентированное обучение	
6	Работа с временными рядами: DatetimeIndex, ресемплинг, сдвиги, скользящие окна	ПЗ	Дискуссионные технологии, Работа в группах, Кейс-стади, Проектно-ориентированное обучение, Взаимооценка	25
7	Введение в статистический анализ и моделирование с использованием SciPy и Statsmodels	ПЗ	Проектно-ориентированное обучение	25
8	Построение пайплайнов обработки данных и базовых моделей с Scikit-learn	ПЗ	Проектно-ориентированное обучение	25
<b>Итого</b>				<b>25%</b>

### Очно-заочная форма обучения (полный срок)

№	Тема занятия	Вид учебного занятия	Форма / Методы интерактивного обучения	% учебного времени
1	2	3	4	5
1	Введение в экосистему анализа данных на Python: NumPy и основы векторизации	ПЗ	Дискуссионные технологии, Работа в малых группах, Взаимопроверка, Мозговой штурм, Интерактивные тренажёры	25
2	Работа с табличными данными: библиотека Pandas (Series, DataFrame, индексация, фильтрация)	ПЗ	Дискуссионные технологии, Групповое решение проблемных задач, Кейс-стади, Взаимообучение, Проектно-ориентированное обучение	25
3	Очистка и подготовка данных с Pandas: обработка пропусков, дубликатов, выбросов и преобразование типов	ПЗ	Работа в парах, Деловая игра, Дискуссионные технологии, Мозговой штурм, Интерактивная визуализация с коллективным обсуждением	25
4	Агрегирование, группировка и слияние данных: groupby, merge, concat, pivot_table	ПЗ	Семинар-дискуссия, Работа в группах с презентацией, Ролевая игра, Интерактивная доска, Кейс-стади	25
5	Визуализация данных: Matplotlib и Seaborn (основные типы графиков, настройка стилей, интерпретация)	ПЗ	Групповое решение задач, Конкурс, Интерактивная визуализация, Мозговой штурм, Проектно-ориентированное обучение	25
6	Работа с временными рядами: DatetimeIndex, ресемплинг, сдвиги, скользящие окна	ПЗ	Дискуссионные технологии, Работа в группах, Кейс-стади, Проектно-ориентированное обучение, Взаимооценка	25
7	Введение в статистический анализ и моделирование с использованием SciPy и Statsmodels	ПЗ	Проектно-ориентированное обучение	25
<b>Итого</b>				<b>25%</b>

## Раздел 4. Организация самостоятельной работы обучающихся

### 4.1. Организация самостоятельной работы обучающихся

№	Тема дисциплины	№ вопро-сов	№ рекомендуемой литерату-ры
1	2	3	4
1	Введение в экосистему анализа данных на Python: NumPy и основы векторизации	1-5	1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15
2	Работа с табличными данными: библиотека Pandas (Series, DataFrame, индексация, фильтрация)	6-10	1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15
3	Очистка и подготовка данных с Pandas: обработка пропусков, дубликатов, выбросов и преобразование типов	11-15	1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15
4	Агрегирование, группировка и слияние данных: groupby, merge, concat, pivot_table	16-20	1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15
5	Визуализация данных: Matplotlib и Seaborn (основные типы графиков, настройка стилей, интерпретация)	21-25	1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15
6	Работа с временными рядами: DatetimeIndex, ресемплинг, сдвиги, скользящие окна	26-30	1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15
7	Введение в статистический анализ и моделирование с использованием SciPy и Statsmodels	31-35	1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15
8	Построение пайплайнов обработки данных и базовых моделей с Scikit-learn	35-40	1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15

#### Перечень вопросов, выносимых на самостоятельную работу обучающихся

1. Какие основные библиотеки Python используются для анализа данных и какие задачи решает каждая из них?
2. Что такое массив ndarray в NumPy? Перечислите его основные атрибуты (shape, dtype, size, ndim).
3. Какие способы создания массивов NumPy вы знаете? Приведите примеры использования arau, zeros, ones, arange, linspace.
4. Что такое булево индексирование в NumPy? Как с его помощью выполнить фильтрацию элементов массива по условию?
5. Сформулируйте правила трансляции (broadcasting) в NumPy. Приведите пример, когда трансляция работает, и пример, когда возникает ошибка.
6. Что такое Series и DataFrame в Pandas? В чём разница между этими структурами данных?
7. Какие способы создания DataFrame существуют? Как загрузить данные из CSV-файла?
8. В чём разница между методами доступа loc и iloc в Pandas? Приведите примеры использования каждого.
9. Как выполнить фильтрацию строк DataFrame по условию? Приведите пример с использованием булева массива.
10. Какие методы в Pandas используются для первичного просмотра и описания данных (head, info, describe)?
11. Как обнаружить пропущенные значения в DataFrame? Какие методы используются для их под-

счёта и визуализации?

12. Какие стратегии заполнения пропусков существуют в Pandas? Приведите примеры использования `fillna` с разными параметрами.
13. Как обнаружить и удалить дублирующиеся строки в `DataFrame`? Какие методы для этого предназначены?
14. Какие методы используются для выявления выбросов в данных? Опишите метод межквартильного размаха (IQR).
15. Как преобразовать типы данных в Pandas? Что делает метод `to_datetime` и для чего он применяется?
16. Опишите принцип работы метода `groupby` в Pandas. Как работает концепция `split-apply-combine`?
17. Как применить несколько агрегирующих функций одновременно при группировке? Что делает метод `agg`?
18. В чём разница между методами `concat` и `merge` в Pandas? Когда какой метод предпочтительнее?
19. Какие типы соединений (`joins`) поддерживает метод `merge`? Опишите `inner`, `left`, `right`, `outer` join.
20. Что такое сводная таблица (`pivot_table`) в Pandas? Для решения каких задач она применяется?
21. Какие основные типы графиков поддерживает библиотека `Matplotlib`? Для визуализации каких данных подходит каждый тип?
22. Как создать несколько подграфиков (`subplots`) в `Matplotlib`? Приведите пример композиции из двух графиков.
23. Что такое `Seaborn` и в чём его преимущества перед `Matplotlib`? Как изменить стиль оформления графиков?
24. Как визуализировать корреляционную матрицу с помощью тепловой карты (`heatmap`) в `Seaborn`?
25. Что показывает диаграмма рассеяния (`scatterplot`) и матрица парных графиков (`pairplot`)? Как их интерпретировать?
26. Как преобразовать столбец с датами в формат `datetime` и установить его в качестве индекса `DataFrame`?
27. Как выполнить фильтрацию временного ряда по диапазону дат? Приведите пример.
28. Что такое ресемплинг временного ряда? Как изменить дневные данные на месячные с помощью `resample`?
29. Что делают методы `shift` и `diff` при работе с временными рядами? Для чего они применяются?
30. Что такое скользящее среднее (`rolling mean`)? Как его вычислить и для чего оно используется при анализе временных рядов?
31. Какие статистические критерии доступны в модуле `scipy.stats`? Опишите t-критерий Стьюдента.
32. Как рассчитать коэффициент корреляции Пирсона между двумя переменными с помощью `SciPy`? Как интерпретировать полученное значение?
33. В чём разница между корреляцией Пирсона и корреляцией Спирмена? Когда какой метод предпочтительнее?
34. Как построить модель линейной регрессии с помощью `statsmodels`? Что показывает метод `summary`?
35. Как интерпретировать коэффициенты линейной регрессии,  $p$ -значения и коэффициент детерминации  $R^2$ ?
36. Как разделить данные на обучающую и тестовую выборки в `Scikit-learn`? Какую функцию для этого используют?
37. Для чего используется масштабирование признаков? В чём разница между `StandardScaler` и `MinMaxScaler`?
38. Что такое `Pipeline` в `Scikit-learn`? Какие преимущества даёт использование пайплайнов?
39. Для чего предназначен `ColumnTransformer`? Как он используется для обработки разнотипных признаков?
40. Как оценить качество модели классификации с помощью метрик `accuracy`, `precision`, `recall` и `F1-мера`?

#### **4.2. Перечень учебно-методического обеспечения самостоятельной работы обучающихся**

Самостоятельная работа обучающихся обеспечивается следующими учебно-методическими материалами:

1. Указаниями в рабочей программе по дисциплине (п.4.1.)
2. Лекционные материалы в составе учебно-методического комплекса по дисциплине
3. Заданиями и методическими рекомендациями по организации самостоятельной работы обучающихся в составе учебно-методического комплекса по дисциплине.
4. Глоссарием по дисциплине в составе учебно-методического комплекса по дисциплине.

## Раздел 5. Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся

Фонд оценочных средств по дисциплине представляет собой совокупность контролирующих материалов, предназначенных для измерения уровня достижения обучающимися установленных результатов образовательной программы. ФОС по дисциплине используется при проведении оперативного контроля и промежуточной аттестации обучающихся. Требования к структуре и содержанию ФОС дисциплины регламентируются Положением о фонде оценочных материалов по программам высшего образования – программам бакалавриата, магистратуры.

### 5.1. Паспорт фонда оценочных средств

#### Очная форма обучения (полный срок)

№	Контролируемые разделы (темы) дисциплины	Оценочные средства			
		Л	ПЗ (ЛЗ, СЗ)	СРО	Код индикатора и дескриптора достижения компетенций
1	2	3	4	5	6
1	Введение в экосистему анализа данных на Python: NumPy и основы векторизации	УО	33, МШ	ПРВ	ИД-1 ОПК- 2.1 ИД-2 ОПК- 2.2
2	Работа с табличными данными: библиотека Pandas (Series, DataFrame, индексация, фильтрация)	УО	33, Д	ПРВ	ИД-1 ОПК- 2.1 ИД-2 ОПК- 2.2
3	Очистка и подготовка данных с Pandas: обработка пропусков, дубликатов, выбросов и преобразование типов	УО	33, Д, МШ	ПРВ	ИД-3 ОПК- 2.1 ИД-4 ОПК- 2.2
4	Агрегирование, группировка и слияние данных: groupby, merge, concat, pivot_table	УО	33, Д, МП	ПРВ	ИД-3 ОПК- 2.1 ИД-4 ОПК- 2.2
5	Визуализация данных: Matplotlib и Seaborn (основные типы графиков, настройка стилей, интерпретация)	УО	33, МШ	ПРВ	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2
6	Работа с временными рядами: DatetimeIndex, ресемплинг, сдвиги, скользящие окна	УО	33, Д	ПРВ	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2
7	Введение в статистический анализ и моделирование с использованием SciPy и Statsmodels	УО	33, МШ	ПРВ	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2
8	Построение пайплайнов обработки данных и базовых моделей с Scikit-learn	УО	33, Д	ПРВ	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2

#### Очно-заочная форма обучения (полный срок)

№	Контролируемые разделы (темы) дисциплины	Оценочные средства			
		Л	ПЗ (ЛЗ, СЗ)	СРО	Код индикатора и дескриптора достижения компетенций
1	2	3	4	5	6
1	Введение в экосистему анализа данных на Python: NumPy и основы векторизации	УО	33, МШ	ПРВ	ИД-1 ОПК- 2.1 ИД-2 ОПК- 2.2
2	Работа с табличными данными: библиотека Pandas (Series, DataFrame, индексация,	УО	33, Д	ПРВ	ИД-1 ОПК- 2.1 ИД-2 ОПК- 2.2

	фильтрация)				
3	Очистка и подготовка данных с Pandas: обработка пропусков, дубликатов, выбросов и преобразование типов	УО	33, Д, МШ	ПРВ	ИД-3 ОПК- 2.1 ИД-4 ОПК- 2.2
4	Агрегирование, группировка и слияние данных: <code>groupby</code> , <code>merge</code> , <code>concat</code> , <code>pivot table</code>	УО	33, Д, МП	ПРВ	ИД-3 ОПК- 2.1 ИД-4 ОПК- 2.2
5	Визуализация данных: <code>Matplotlib</code> и <code>Seaborn</code> (основные типы графиков, настройка стилей, интерпретация)	УО	33, МШ	ПРВ	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2
6	Работа с временными рядами: <code>DatetimeIndex</code> , ресемплинг, сдвиги, скользящие окна	УО	33, Д	ПРВ	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2
7	Введение в статистический анализ и моделирование с использованием <code>SciPy</code> и <code>Statsmodels</code>	УО	33, МШ	ПРВ	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2
8	Построение пайплайнов обработки данных и базовых моделей с <code>Scikit-learn</code>			ПРВ	ИД-5 ОПК- 2.1 ИД-6 ОПК- 2.2

### Условные обозначения оценочных средств (Столбцы 3, 4, 5):

**33** – защита выполненных заданий (творческих, расчетных и т.д.), представление презентаций;

**ПРВ** – проверка рефератов, отчетов, рецензий, аннотаций, конспектов, графического материала, эссе, переводов, решений заданий, выполненных заданий в электронном виде и т.д.;

**МШ** – Метод мозгового штурма;

**Д** – Дискуссия, полемика, диспут, дебаты;

**МП** – Метод проектов.

### 5.2. Тематика письменных работ обучающихся

1. Сравнительный анализ производительности векторизованных вычислений в NumPy и циклических операций на Python.
2. Исследование методов обработки пропущенных значений в Pandas: удаление, заполнение статистиками, интерполяция.
3. Применение группировки и сводных таблиц для анализа продаж интернет-магазина.
4. Визуализация распределений и выявление выбросов с использованием `boxplot` и гистограмм в `Matplotlib` и `Seaborn`.
5. Анализ временного ряда финансовых данных: ресемплинг, скользящие средние и выявление трендов.
6. Сравнение методов масштабирования данных (`StandardScaler`, `MinMaxScaler`, `RobustScaler`) и их влияние на качество моделей.
7. Разработка пайплайна предобработки данных для задачи классификации с использованием `ColumnTransformer`.
8. Исследование корреляционных зависимостей между признаками с визуализацией тепловой карты и парных графиков.
9. Применение t-критерия Стьюдента и критерия Манна-Уитни для сравнения выборок в реальном датасете.
10. Построение и интерпретация модели множественной линейной регрессии с использованием `Statsmodels`.
11. Обработка категориальных признаков: сравнение `OneHotEncoder`, `OrdinalEncoder` и целевого кодирования.
12. Анализ датасета с пропусками и выбросами: полный цикл очистки и подготовки данных.

13. Визуализация многомерных данных с помощью метода главных компонент (PCA) и интерпретация результатов.
14. Исследование распределений случайных величин с использованием гистограмм, ядерной оценки плотности и Q-Q графиков.
15. Сравнение эффективности различных стратегий заполнения пропусков при прогнозировании временных рядов.
16. Применение скользящих окон и экспоненциального сглаживания для прогнозирования временного ряда.
17. Исследование влияния выбросов на коэффициент корреляции Пирсона и методы их обнаружения.
18. Разработка пайплайна для автоматической обработки и визуализации данных с использованием Pipeline и FunctionTransformer.
19. Сравнение моделей линейной регрессии, дерева решений и случайного леса на реальном датасете.
20. Анализ сезонности и трендов временного ряда с использованием декомпозиции и визуализации компонентов.
21. Применение кросс-валидации для оценки стабильности моделей машинного обучения.
22. Исследование влияния дисбаланса классов на метрики классификации и методов его коррекции.
23. Построение дашборда для разведочного анализа данных с использованием библиотек визуализации.
24. Анализ текстовых данных с применением Bag of Words, TF-IDF и визуализации облака слов.
25. Сравнительный анализ библиотек визуализации: Matplotlib, Seaborn и Plotly Express.

### 5.3. Перечень вопросов промежуточной аттестации по дисциплине

#### Вопросы к экзамену:

1. Опишите экосистему Python для анализа данных. Какие основные библиотеки (NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn) решают какие задачи?
2. Что такое массив ndarray в NumPy? Перечислите его основные атрибуты (shape, dtype, size, ndim) и способы создания массивов.
3. Что такое векторизация вычислений в NumPy? В чём её преимущество перед циклами? Приведите примеры универсальных функций (ufunc).
4. Сформулируйте правила трансляции (broadcasting) в NumPy. Приведите примеры успешной трансляции и случаи, когда возникает ошибка.
5. Что такое Series и DataFrame в Pandas? Какие способы создания DataFrame существуют? Как загрузить данные из CSV-файла?
6. В чём разница между методами доступа loc и iloc в Pandas? Приведите примеры использования каждого для выборки данных.
7. Как выполнить фильтрацию строк DataFrame по условию? Опишите использование булевых масок и метода query.
8. Какие методы в Pandas используются для обнаружения и обработки пропущенных значений? Опишите dropna, fillna, interpolate.
9. Как обнаружить и удалить дублирующиеся строки в DataFrame? Какие методы duplicated и drop\_duplicates для этого используются?
10. Какие методы используются для выявления выбросов в данных? Опишите метод межквартильного размаха (IQR) и его реализацию в Pandas.
11. Опишите принцип работы метода groupby в Pandas. Как работает концепция split-apply-combine? Приведите пример агрегации с agg.

12. В чём разница между методами `concat` и `merge` в `Pandas`? Какие типы соединений (`inner`, `left`, `right`, `outer`) поддерживает `merge`?
13. Что такое сводная таблица (`pivot_table`) в `Pandas`? Для решения каких задач она применяется? Чем отличается от `crosstab`?
14. Какие основные типы графиков поддерживает `Matplotlib`? Для визуализации каких данных подходит каждый тип? Как создать несколько подграфиков (`subplots`)?
15. Что такое `Seaborn` и в чём его преимущества перед `Matplotlib`? Как визуализировать корреляционную матрицу с помощью `heatmap`?
16. Как преобразовать столбец с датами в формат `datetime` и установить его в качестве индекса `DataFrame`? Как выполнить фильтрацию по диапазону дат?
17. Что такое ресемплинг временного ряда? Как изменить дневные данные на месячные с помощью `resample`? Какие методы агрегации можно использовать?
18. Что делают методы `shift` и `diff` при работе с временными рядами? Для чего применяется скользящее среднее (`rolling mean`)?
19. Какие статистические критерии доступны в модуле `scipy.stats`? Опишите t-критерий Стьюдента для одной и двух выборок.
20. Как рассчитать коэффициент корреляции Пирсона между двумя переменными с помощью `SciPy`? Как интерпретируются значение коэффициента и `p-value`?
21. Как построить модель линейной регрессии с помощью `Statsmodels`? Что показывает метод `summary`? Как интерпретируются коэффициенты и  $R^2$ ?
22. Как разделить данные на обучающую и тестовую выборки в `Scikit-learn`? Для чего используется параметр `random_state` и `stratify`?
23. Для чего используется масштабирование признаков? В чём разница между `StandardScaler`, `MinMaxScaler` и `RobustScaler`?
24. Что такое `Pipeline` в `Scikit-learn`? Какие преимущества даёт использование пайплайнов? Для чего предназначен `ColumnTransformer`?
25. Какие метрики используются для оценки качества моделей классификации? Дайте определение `accuracy`, `precision`, `recall`, F1-мере и матрице ошибок.

## Раздел 6. Оценочные средства промежуточной аттестации (с ключами)

1. Прочитайте текст и выберите один правильный ответ. Какой оператор используется для вывода данных на экран в `Python`?
  - а) `input()`
  - б) `print()`
  - в) `echo()`
  - г) `write()`
  - д) `display()`Правильный ответ: б)
2. Прочитайте текст и выберите один правильный ответ. Какой тип данных НЕ является изменяемым в `Python`?
  - а) `list`
  - б) `dict`
  - в) `set`
  - г) `tuple`
  - д) `bytearray`Правильный ответ: г)
3. Прочитайте текст и выберите один правильный ответ. Что вернет выражение `len("Python")`?
  - а) 5
  - б) 6
  - в) 7
  - г) 8

д) Error

Правильный ответ: б)

4. Прочитайте текст и выберите один правильный ответ. Какой символ используется для обозначения комментария в Python?

а) //

б) /\*

в) #

г) --

д) %%

Правильный ответ: в)

5. Прочитайте текст и выберите два правильных ответа. Какие из перечисленных методов используются для работы со списками в Python?

а) append()

б) insert()

в) push()

г) add()

д) enqueue()

Правильный ответ: а), б)

6. Прочитайте текст и выберите два правильных ответа. Какие ключевые слова используются для создания функции в Python?

а) function

б) def

в) return

г) lambda

д) func

Правильный ответ: б), г)

7. Укажите правильную последовательность этапов работы с файлом в Python:

1) Закрытие файла

2) Открытие файла

3) Чтение/запись данных

4) Создание объекта файла

Правильный ответ: 2-4-3-1

8. Укажите правильную последовательность приоритетов операций в Python (от высшего к низшему):

1) Сложение/вычитание

2) Скобки

3) Умножение/деление

4) Возведение в степень

Правильный ответ: 2-4-3-1

9. Установите соответствие между типами данных и их примерами:

А) int

Б) float

В) str

Г) bool

1) 3.14

2) "Hello"

3) True

4) 42

Правильный ответ: А-4, Б-1, В-2, Г-3

10. Установите соответствие между методами строк и их назначением:

А) upper()

Б) split()

В) strip()

1) Удаляет пробелы с концов строки

2) Преобразует строку к верхнему регистру

3) Разбивает строку на список

4) Преобразует строку к нижнему регистру

Правильный ответ: А-2, Б-3, В-1

11. Прочитайте текст и запишите правильный ответ. Ответ следует записать с маленькой буквы. Как называется структура данных, которая хранит элементы в порядке "последним пришел - первым вышел"?

Правильный ответ: стек

12. Прочитайте текст и запишите правильный ответ. Ответ следует записать с маленькой буквы. Какое ключевое слово используется для создания цикла с перебором элементов последовательности?

Правильный ответ: for

13. Прочитайте текст и запишите развернутый обоснованный ответ. Ответ следует записать с маленькой буквы. Дайте определение понятию "функция в Python" – это ...

Правильный ответ: именованный блок кода, который выполняет определенную задачу и может быть вызван многократно из разных мест программы; функция может принимать аргументы и возвращать значение (ответ студента может быть представлен в интерпретации, эквивалентной приведенному правильному ответу)

14. Прочитайте текст и запишите развернутый обоснованный ответ. Ответ следует записать с маленькой буквы. Дайте определение понятию "исключение в Python" – это ...

Правильный ответ: объект, который представляет ошибку, возникающую во время выполнения программы; исключения позволяют обрабатывать ошибочные ситуации и предотвращать аварийное завершение программы (ответ студента может быть представлен в интерпретации, эквивалентной приведенному правильному ответу)

## **Раздел 7. Перечень учебной литературы, необходимой для освоения дисциплины**

### **7.1. Основная литература**

1. Маккинли У. Python и анализ данных : практическое пособие. 3-е изд. Саратов : Профобразование, 2024. 482 с. (Книга создателя библиотеки pandas, охватывает IPython, NumPy, Pandas, Matplotlib. Третье издание обновлено под Python 3.10 и pandas 1.4) .
2. Маккинни У. Использование Python для анализа данных : руководство. Пер. с англ. А. Слинкина. Москва : ДМК Пресс, 2024. 536 с. (Авторитетный справочник по обработке наборов данных, содержит практические примеры эффективного решения задач анализа данных) .
3. Маккинни У. Python и анализ данных. Пер. с англ. А. Слинкина. 3-е изд. Москва : ДМК Пресс, 2024. 482 с. (Современное практическое введение в разработку научных приложений на Python, ориентированных на обработку данных) .
4. Маккинни У. Python и анализ данных : первичная обработка данных с применением pandas, NumPy и Jupyter. Москва : ДМК Пресс, 2024. 536 с. (Авторитетный справочник по реформатированию, очистке и обработке наборов данных на Python, переработанный с учетом версий Python 3.10 и pandas 1.4) .

5. Маккинни У. Использование Python для анализа данных. 3-е изд. Санкт-Петербург : Питер, 2024. 560 с. (Полное руководство по библиотекам Pandas, NumPy и IPython с множеством практических примеров) .

## 7.2. Дополнительная литература

1. Вандер Плас Дж. Python для сложных задач: наука о данных и машинное обучение. Санкт-Петербург : Питер, 2018. 576 с. (Книга охватывает NumPy, Pandas, Matplotlib, Scikit-learn и основы машинного обучения) .
2. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn, Keras и TensorFlow. 3-е изд. Москва : Диалектика, 2021. 848 с. (Продвинутое руководство, включающее подготовку данных и feature engineering на основе Pandas и NumPy) .
3. Маккинли У. Python для анализа данных. 2-е изд. Москва : Вильямс, 2020. 480 с. (Предыдущее издание книги-бестселлера, актуально для знакомства с основами) .
4. Агалаков А.А. Python — от базовых конструкций до обработки данных : учебное пособие для СПО. Саратов : Профобразование, 2025. 109 с. (Пособие охватывает работу с популярными библиотеками для анализа данных) .
5. Чубукова И.А. Data Mining : учебное пособие. Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. 469 с. (Пособие описывает методы анализа данных с применением современных инструментов) .

## 7.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Маккинни У. Официальный сайт книги «Python for Data Analysis». Режим доступа: <https://wesmckinney.com/book> (дата обращения: 09.04.2026). (Открытый онлайн-ресурс, предоставляющий доступ к содержанию книги, файлам данных и коду примеров для актуальной версии Python 3.10) .
2. GitHub репозиторий книги «pydata-book». Режим доступа: <https://github.com/wesm/pydata-book> (дата обращения: 09.04.2026). (Официальный репозиторий с файлами данных, Jupyter notebook и дополнительными материалами для всех глав книги) .
3. Официальная документация библиотеки Pandas. Режим доступа: <https://pandas.pydata.org/docs/> (дата обращения: 09.04.2026). (Полное справочное руководство по всем функциям и методам Pandas с примерами использования) .
4. Официальная документация библиотеки NumPy. Режим доступа: <https://numpy.org/doc/stable/> (дата обращения: 09.04.2026). (Исчерпывающая документация по работе с многомерными массивами, универсальным функциям и линейной алгебре) .
5. Цифровой образовательный ресурс IPR SMART. Режим доступа: <https://www.iprbookshop.ru> (дата обращения: 09.04.2026). (Электронно-библиотечная система, предоставляющая доступ к актуальным учебным изданиям по Python и анализу данных) .

## Раздел 8. Материально-техническая база и информационные технологии

### Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине:

Материально-техническое обеспечение дисциплины «Python: основные библиотеки для анализа данных» включает в себя учебные аудитории для проведения занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, самостоятельной работы обучающихся.

Учебные аудитории укомплектованы специализированной мебелью и техническими средствами обучения. Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети Интернет.

Дисциплина может реализовываться с применением дистанционных технологий обучения. Специфика реализации дисциплины с применением дистанционных технологий обучения устанавливается дополнением к рабочей программе. В части не противоречащей специфике, изложенной в дополнении к программе, применяется настоящая рабочая программа.

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине с применением дистанционных образовательных технологий включает в себя:

Компьютерная техника, расположенная в учебном корпусе Института (ул. Качинцев, 63, кабинет Центра дистанционного обучения):

1. Intel i 3 3.4Ghz\ОЗУ 4Gb\500GB\RadeonHD5450
2. Intel PENTIUM 2.9GHz\ОЗУ 4GB\500GB

3. личные электронные устройства (компьютеры, ноутбуки, планшеты и иное), а также средства связи преподавателей и студентов.

Информационные технологии, необходимые для осуществления образовательного процесса по дисциплине с применением дистанционных образовательных технологий включают в себя:

- система дистанционного обучения (СДО) (Learning Management System) (LMS) Moodle (Modular Object-Oriented Dynamic Learning Environment);
- электронная почта;
- система компьютерного тестирования;
- Цифровой образовательный ресурс IPR SMART;
- система интернет-связи skype;
- телефонная связь;
- ПО для организации конференций.

Обучение обучающихся инвалидов и обучающихся с ограниченными возможностями здоровья осуществляется посредством применения специальных технических средств в зависимости от вида нозологии.

При проведении учебных занятий по дисциплине используются мультимедийные комплексы, электронные учебники и учебные пособия, адаптированные к ограничениям здоровья обучающихся.

Лекционные аудитории оборудованы мультимедийными кафедрами, подключенными к звуковым колонкам, позволяющими усилить звук для категории слабослышащих обучающихся, а также проекционными экранами, которые увеличивают изображение в несколько раз и позволяют воспринимать учебную информацию обучающимся с нарушениями зрения.

При обучении лиц с нарушениями слуха используется усилитель слуха для слабослышащих людей Cyber Ear модель НАР-40, помогающий обучаемым лучше воспринимать учебную информацию.

Обучающиеся с ограниченными возможностями здоровья, обеспечены печатными и электронными образовательными ресурсами (программы, учебники, учебные пособия, материалы для самостоятельной работы и т.д.) в формах, адаптированных к ограничениям их здоровья и восприятия информации:

#### для лиц с нарушениями зрения:

- в форме электронного документа;
- в форме аудиофайла;

**для лиц с нарушениями слуха:**

- в печатной форме;
- в форме электронного документа;

**для лиц с нарушениями опорно-двигательного аппарата:**

- в печатной форме;
- в форме электронного документа;
- в форме аудиофайла.

## Раздел 9. Методические указания для обучающихся по освоению дисциплины

Дисциплина включает практические занятия, самостоятельную работу обучающегося.

В ходе изучения дисциплины «Python: основные библиотеки для анализа данных» перед обучающимися стоит задача не только закрепить знания о сложных информационных явлениях, о чем свидетельствует содержание тематического плана, глубоко разобраться в объемном учебном материале, но и сформировать у себя на основе полученных компьютерных знаний соответствующие профессионально важные качества.

Практические занятия – один из самых эффективных видов учебных занятий, на которых обучающиеся учатся творчески работать с различной информацией, являются также действенной формой активизации самостоятельной работы обучающихся.

Целью практических занятий является закрепление полученных в ходе лекций, а также в ходе самостоятельной работы над учебной и специальной литературой, знаний, умений и навыков. На практических занятиях особо обращается внимание на умение обучающихся проявлять элементы творчества в процессе самостоятельной работы, применять полученные знания на практике.

Практические занятия занимают центральное место в учебном процессе, так как позволяют на завершающем этапе усвоения материала, после прослушанной лекции и самостоятельного поиска дополнительных сведений по рассматриваемой проблематике, окончательно уточнить, сформировать свои позиции в ходе работы в составе учебной группы.

Основное в подготовке и проведении практикума – это самостоятельная работа обучающегося над изучением темы лекционного материала. Практические занятия проводятся по специальным планам – заданиям, которые содержатся в материалах, подготовленных на кафедре. Обучающийся обязан точно знать план занятия либо конкретное задание к нему.

При подготовке к практическим занятиям следует чаще обращаться к справочной литературе, полнее использовать консультации (групповые и индивидуальные, устные и письменные) с преподавателями, которые читают лекции и проводят практикумы.

Таким образом, в процессе подготовке к практическому занятию рекомендуется:

- ознакомиться с вопросами плана;
- прочитать конспект лекции по изучаемой теме;
- прочитать соответствующие главы учебников, статьи;
- просмотреть перечень научных источников, предлагаемых в рабочей программе, выбрав несколько из них для углубленного изучения данной темы.

По каждому практическому заданию обучающиеся отчитываются преподавателю, оформляя письменный отчет, в котором сохраняют результаты своей работы в виде файлов. Результаты выполнения практических заданий оцениваются с учетом теоретических знаний по соответствующим вопросам дисциплины и уровнем владения практическими навыками при работе на компьютере.

Для углубленного изучения и освоения материала целесообразно выполнение практических работ, наряду с другими различными формами обучения обучающихся: тесты, задачи, упражнения, которые используются при проведении практических занятий, выполнении контрольных и аудиторных работ, а также при самостоятельном изучении данной дисциплины.

Одним из наиболее интенсивных способов изучения дисциплины является самостоятельное выполнение практических работ, на которых вырабатываются навыки по дисциплине «Python: основные библиотеки для анализа данных».

СРО позволяет глубже освоить теоретические и практические вопросы, понять принципы дисциплины «Python: основные библиотеки для анализа данных».

Основными задачами организации процесса самостоятельной работы по дисциплине являются:

- приобретение знаний по теоретическим основам дисциплины «Python: основные библиотеки для анализа данных», являющихся дополнением к материалу лекционных аудиторных занятий;
- приобретение практических навыков по дисциплине «Python: основные библиотеки для анализа данных».

Основные формы реализации СРО – изучение учебно-методической литературы по дисциплине.

плине «Python: основные библиотеки для анализа данных». В качестве базовой литературы можно использовать учебники и учебные пособия, согласно приведенному списку в разделе 6 рабочей программы, а также любые другие источники информации, такие как электронные учебники, обучающие и энциклопедические сайты, публикации журналов и конференций.

Обучающийся допускается к зачетному занятию по результатам успешного выполнения всех практических заданий и самостоятельной работы.

Учебно-методическое издание

Рабочая программа учебной дисциплины

---

**Python: основные библиотеки для анализа данных**

*(Наименование дисциплины в соответствии с учебным планом)*

**Скоробогатченко Дмитрий Анатольевич**

*(Фамилия, Имя, Отчество составителя)*

---