Семинар 1

Бизнес-аналитика и ИИ как инструмент эффективного управления

Что сегодня будем делать

- Познакомимся
- Решим орг. вопросы (группа в TG / формула оценки / инструменты)
- Будем учиться разведочному анализу данных (EDA)

Группа в TG

Пару правил:

1. Общаемся культурно
2. Бесконтрольный флуд переводим в ЛС
3. Не спамим рекламой «1ХБет»

Группа для вас, чтобы вы могли быстро получить нужную информацию, спросить, уточнить

Семинарская группа

Формула оценки

Текущий контроль			Экзамен
Активность (А)	**Домашние задания (ДЗ)**	**Отчет по проекту (О)**	**Защита проекта (З)**
Активность в работе на семинарах	*Домашние и др. практические задания* *(команда 3–5 чел)*	*В форме презентации, других материалов по итогам мини-проекта, подкрепленные использованными аналитическими инструментами*	*Презентация и защита по результатам мини-проекта по бизнес-аналитике* *(команда 3–5 чел)*
0-10 баллов	*0-10 баллов*		*0-10 баллов*
Результирующая оценка (Р)			
$Р = 10\% * А + 25\% * ДЗ\#1 + 25\% * ДЗ\#2 + 20\% * О + 20\% * З$			

Формула оценки

Для сдачи решения домашних заданий и финального проекта
вам нужно будет завести команду:

- **Количество:**
  3-5 человек в команде
- **Участники**:
  Только из своей группы
- **Дедлайн регистрации**:
  14.11
- **Способ регистрации**:
  Мы направим вам ссылку на google-таблицы, где можно будет вступить в команду

Где будем работать

 Python (Google Colab или Jupyter Notebook)

 Orange

Yandex DataLens

Про виртуального друга

Использовать ChatGPT и аналоги для написания кода – **ок**

Но! Обязательно вчитывайтесь в то, что он пишет, и *сами* осмысливайте результаты.

Главное в домашках и в проекте – получить ценный, осмысленный результат. И вы должны *сами* уметь объяснять графики / таблицы, которые будете показывать.

Статья от OpenAI по тому, как правильно промптить GPT-5

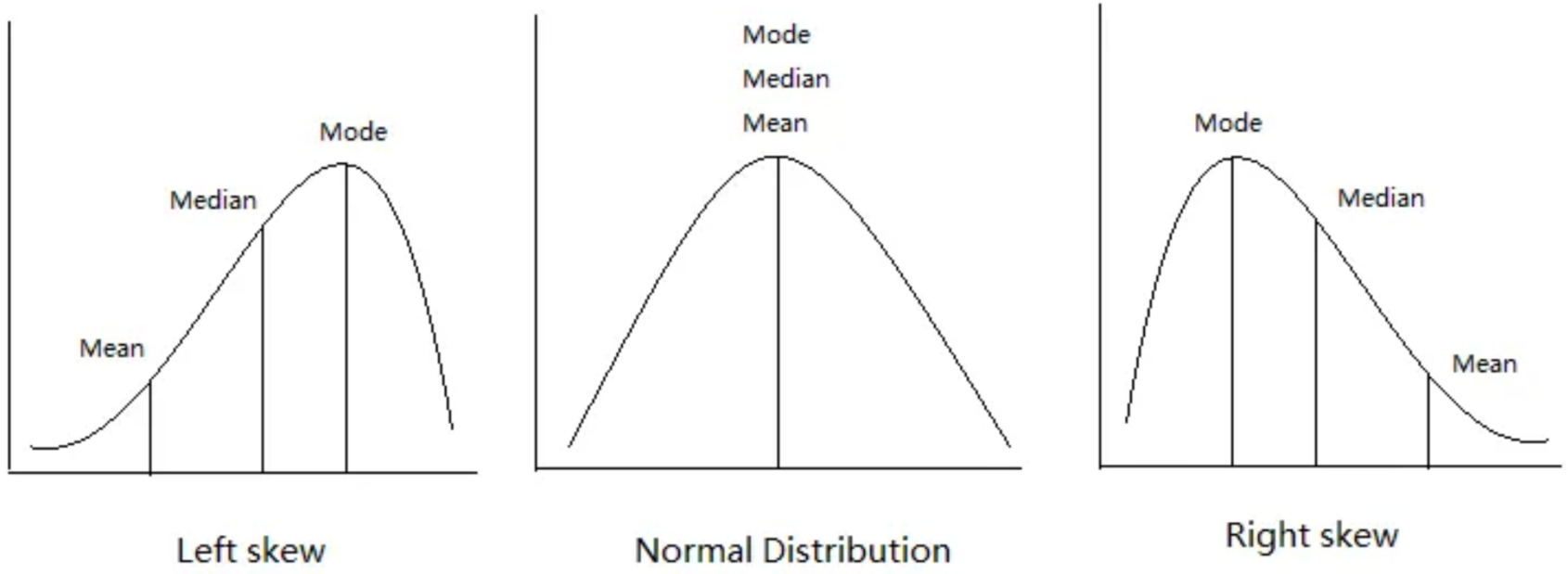
1. Давайте четкую цель, полный контекст, и максимально подробное «ТЗ».
2. *Hint:* GPT-5 и другие современные модели очень хорошо ориентируются в тегах. Например, хороший промпт может выглядеть так:
   <context>… </context>
   <goal> … </goal>
   <limitations>…</limitations>

Переходим к занятию!

Вспоминаем статистику

- Среднее (Mean)
- Медиана (Median)
- Мода (Mode)

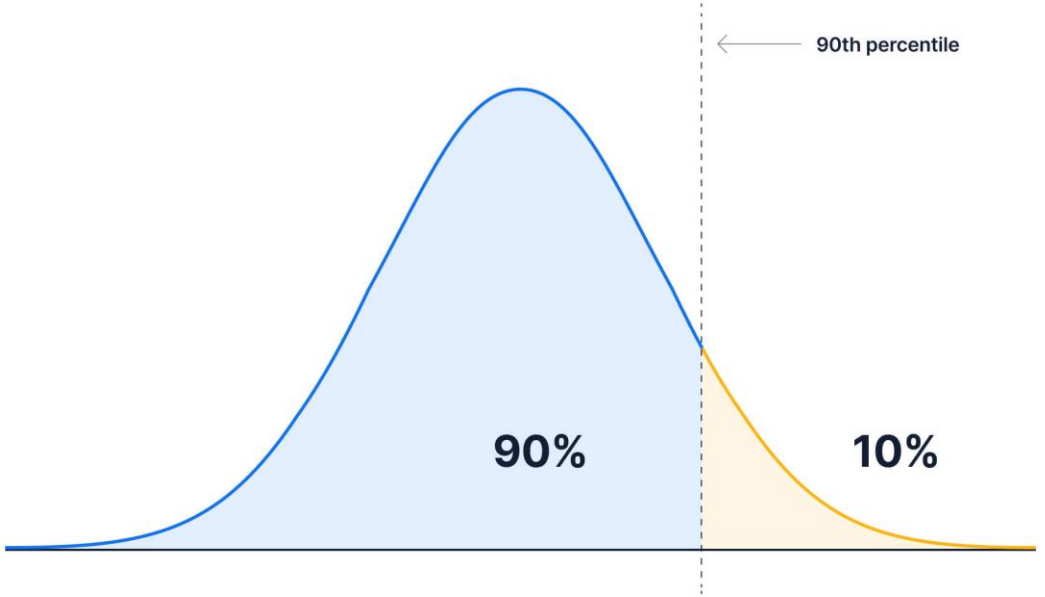
Когда считаете среднее в данных, на всякий случай всегда проверяйте медиану. Если они сильно отличаются – это знак, что у вас в данных выбросы (которые иногда очень важно чистить / правильно обработать), или ваше распределение сильно скошено (skewed)



Вспоминаем статистику

- *Перцентили (10% / 50% / 90% / 99% / 99.99%)*

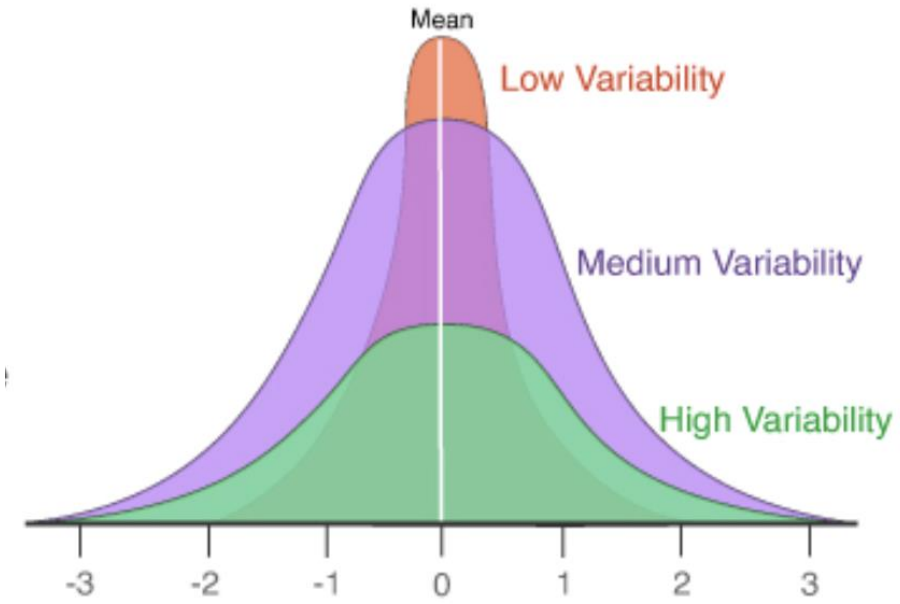
Пример использования: вам нужно ввести ограничение на количество контактов (позвонить / написать), которое может сделать пользователь на Авито в течение дня. Как просто подобрать правильное ограничение? Подобрать такое число контактов, чтобы 99% пользователей не заметили бы изменений. То есть вам нужно найти 99-й перцентиль по количеству контактов за день.



Вспоминаем статистику

- *Дисперсия*
- *Стандартное отклонение*

Отвечаем на вопрос: насколько «разбросаны» наши данные? Все наблюдения находятся рядом с каким-то значением, или между наблюдениями есть очень большая разница?



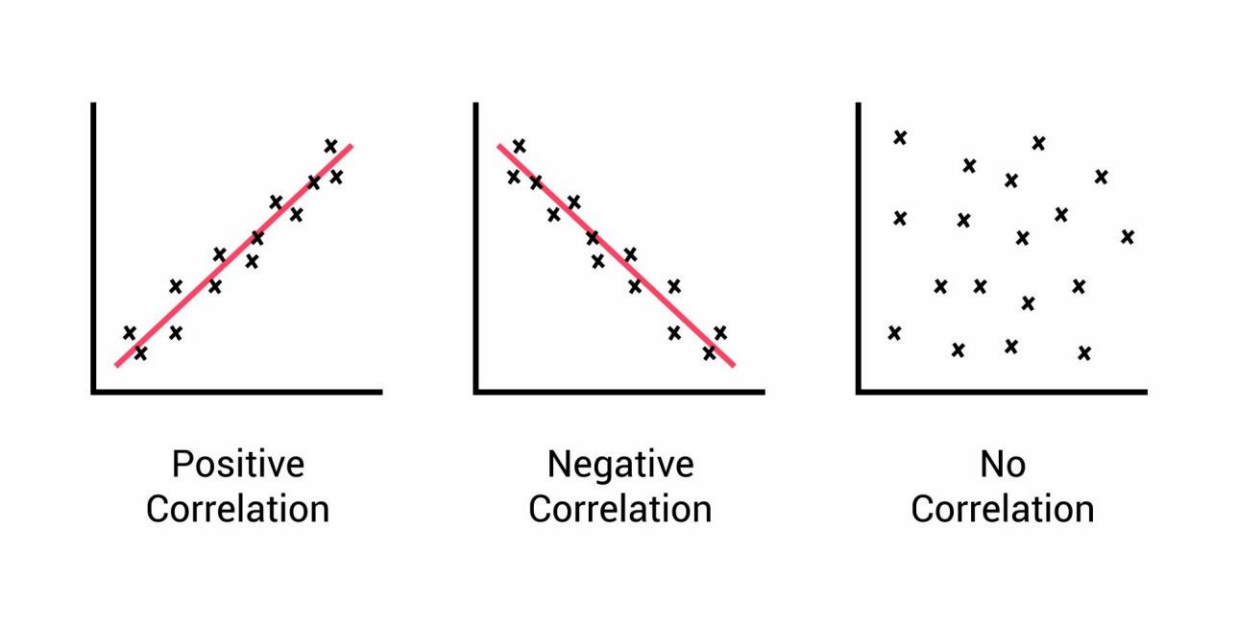
$$\sigma^2 = \frac{\sum (X - \mu)^2}{N}$$

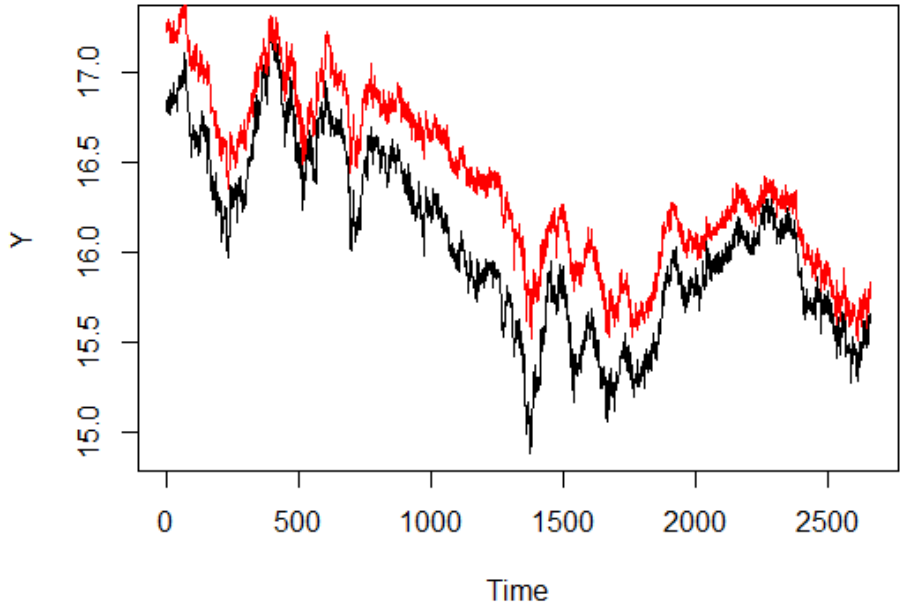
$$\sigma = \sqrt{\frac{\sum (X - \mu)^2}{n}}$$

Вспоминаем статистику

- *Корреляция*

Взаимосвязь между двумя или более случайными величинами, показывающая, насколько их изменения сопутствуют друг другу





Correlation is not causation!

Метрики

Метрика – это ***измеряемый*** показатель, который используется для отслеживания *чего-либо.*

Зачем метрики нужны в бизнесе:

1. Понимать, что вообще происходит с нашей компанией / продуктом
   (каждый день смотрим на операционные метрики – проверяем, что все ок)
2. Понимать, хорошо или плохо поработали отдельные люди / команды
   (ставим KPI / проводим A/B тесты)
3. Понимать, какие инициативы могут сделать наш продукт лучше, и решать, что делать в первую очередь
   (оцениваем инициативы по конкретным метрикам)

Метрики. Типы метрик

Когда мы что-то тестируем / меняем, то обычно выделяем три типа метрик:

- **Целевые**
- **Прокси**
- **Guardrail («Защитные»)**

Одна и та же метрика может выступать в любой из этих категорий в зависимости от конкретного кейса.

Пример.
Хотим улучшить checkout в нашем интернет магазине, чтобы у нас было больше заказов
(допустим, сейчас плохой дизайн / много шагов надо сделать / ...)

Целевая метрика: кол-во оплаченных заказов
Прокси метрики: конверсия на чекауте; скорость прохождения чекаута; ...
Guardrail: технические ошибки на чекауте / обращения в поддержку по оплате / ...

Метрики. North Star

North Star Metric – это ключевая метрика, который наилучшим образом отражает основную **ценность** продукта для **клиента** и при этом отражает стратегию / цели **компании**

У разных бизнесов – разные NSM:

- *Отельный бизнес* – кол-во подтвержденных ночей / проживаний
- *Маркетплейсы* – кол-во доставленных заказов
- *Авито* – целевые баеры (пользователи с договоренностью о сделке)
- *…*

Можно ли в качестве NSM просто взять общую выручку / прибыль компании?

Метрики. Дерево метрик

Дерево метрик позволяет «разложить» целевую метрику (не обязательно North Star) на *составляющие* части. Здесь мы раскладываем целевую метрику по формулам:

(Прибыль = Доход - расход)

*(Доход = Кол-во оплаченных курсов * средний чек за курс * наша комиссия)*

Что такое EDA

EDA (Exploratory Data Analysis) – «разведочный анализ данных». Процесс, в котором мы изучаем *форму* и *содержание* наших данных и делаем на основе анализа какие-то выводы.

Задачи EDA

1. Проверка качества данных (пропуски, выбросы, дубликаты)
2. Анализ распределений признаков
3. Связи / корреляции между переменными
4. Интерпретирование данных. Гипотезы / выводы / рекомендации

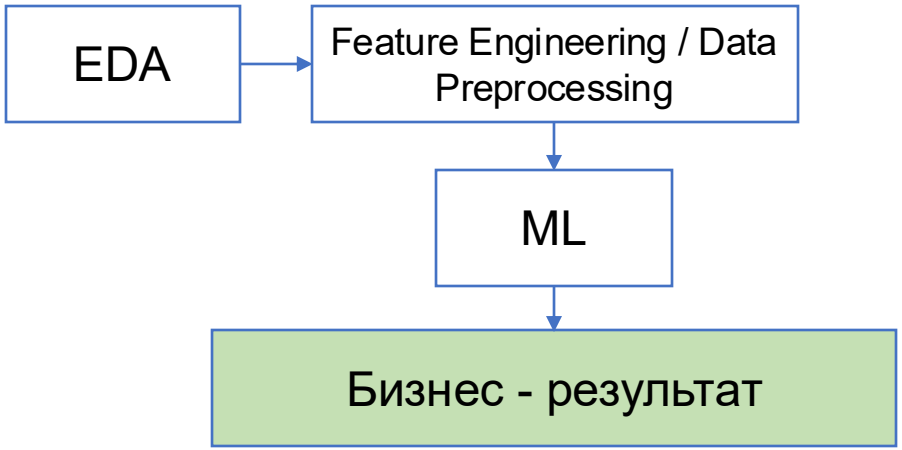
Как применяется EDA

Вариант 1. EDA – часть пайплайна машинного обучения

Здесь EDA – это первый шаг перед созданием ML модели, для которой нужны:

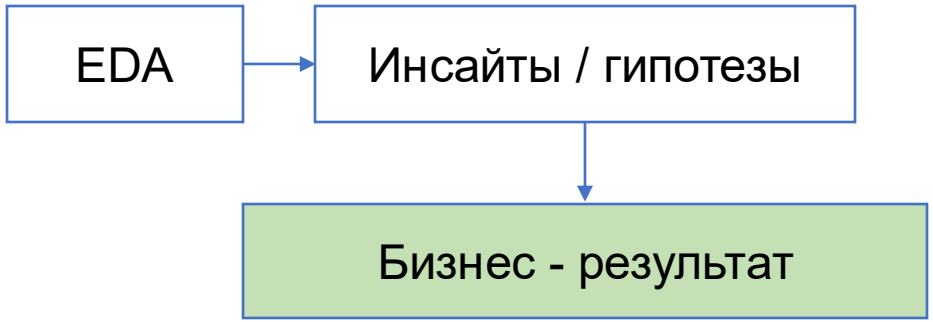
1. Хорошее качество данных
2. Важные признаки (возможно, созданные на основе других признаков)

Эти два пункта мы хотим «закрыть» на этапе EDA



Вариант 2. EDA – самодостаточная аналитика

Здесь EDA – это процесс, на основании которого мы можем **напрямую** (без использования ML) дать бизнесу полезные инсайты / рекомендации.



EDA versus ML

Бизнесу далеко **не всегда нужны ML модели** для решения задач!

Иногда достаточно базовой (но качественной) аналитики на этапе EDA.

В некоторых задачах «бизнес - правила», сформированные на основании EDA, оказываются не хуже, чем тяжеловесные ML модели, которые нужно разрабатывать *N* недель.

Пример
Мы хотим проверить, имеет ли новая CRM-рассылка какой-то потенциал. Нужно собрать под нее сегмент пользователей, которые ранее хорошо реагировали на наши рассылки (которым мы отправим эту новую рассылку).

Вариант 1. Сделать на основе исторических данных ML модель, которая обучится на 100+ признаков клиентов, и будет предсказывать «горячесть» клиентов по отношению к нашим рассылкам
Вариант 2. На основе быстрого EDA собрать сегмент пользователей по 3-5 основным признакам, которые чаще среднего заходят к нам на площадку / к нам в категорию

Рассмотрим основные этапы EDA

1. Загрузка данных и первичный осмотр

1. Загружаем данные
   Источниками могут быть CSV / Excel / SQL-запрос / …
2. Смотрим на количество строк и столбцов
   Уже на этом этапе можно обнаружить странности в данных. Не слишком ли много / мало строк мы получили?
3. Быстро проверяем, какие переменные у нас есть
   df.head() / df.info() / df.describe()
   Ожидаемые ли у всех переменных типы данных? Есть ли признаки, которые должны быть числовыми (int / float), а отмечены они как строковые (object)? В каком формате признак даты (если он есть)?

1. Загрузка данных и первичный осмотр

Вспоминаем термины

Client_ID	Income	City	Age	…
131230482	150 000	Москва	35	
123232412	200 000	Екатеринбург	30	
021343000	70 000	Омск	26	

2. Качество данных

Пропуски / пустые значения

None / NaN значения.
- Есть ли у нас «отсутствующие» значения?
- Много ли их относительно размера датасета (0.01% / 1% / 10%)?
- В каких признаках встречаются пустые значения?

Виды пропусков в данных

- **MCAR (Missing Completely At Random)**
  Пропуски не зависят ни от чего (пропуски «полностью случайны»)
- **MAR (Missing At Random)**
  Пропуски зависят от наблюдаемых признаков. Например, у молодых пользователей чаще не указан доход. Можно «восстановить» признак дохода молодого человека, если взять средний доход у молодых в целом.
- **MNAR (Missing Not At Random)**
  Пропуски зависят от того, что мы пытаемся измерить. Например, люди с низким доходом не указывают свой доход, именно потому что он низкий. Такие пропуски требуют наибольшего внимания – «восстановить» такие пропуски наиболее проблемно (нужны допущения / доп. данные).

2. Качество данных

Дубликаты
Полностью повторяющиеся строки в датасете.
- Ожидаемо ли, что у нас в данных есть дубликаты?
- Если не ожидаемо, откуда дубликаты могли взяться?

Если дубликатов быть не должно, а они есть – часто проблема кроется в некорректном сборе данных (например, в SQL запросе в одном из JOIN данные задублировались).

2. Качество данных

Выбросы

Значения, которые значительно отличается от большинства других наблюдений в наборе данных. Если средний доход = 100 тыс. руб, а в данных у определенного объекта отмечен доход 10 млн. руб, то такое наблюдение (очень вероятно) можно назвать «выбросом».

Как обнаруживать выбросы

1. Проще всего – на глаз. Выбросы почти всегда видно на распределениях.
2. Через статистику. Например, через перцентили. 99-й перцентиль по доходу означает такой уровень дохода, который превышает всего 1% объектов (населения).

3. Анализ распределений и взаимосвязей переменных

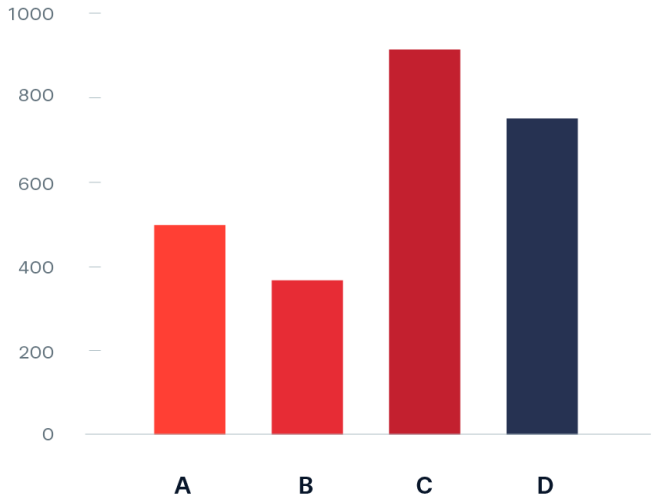
Далее мы переходим непосредственно к анализу данных.

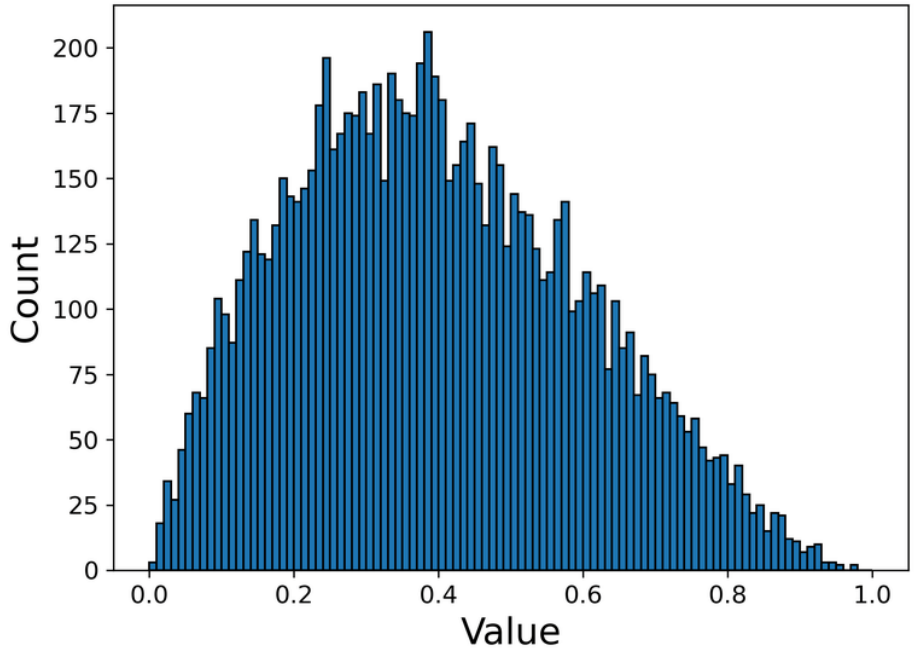
Для начала, вспомним визуальную составляющую анализа (типы графиков)...

3. Анализ распределений и взаимосвязей переменных

Barchart / Histogram

Показывает распределение по опредленному признаку. В случае Barchart смотрим на категориальную переменную, в случае Histogram – на числовую переменную

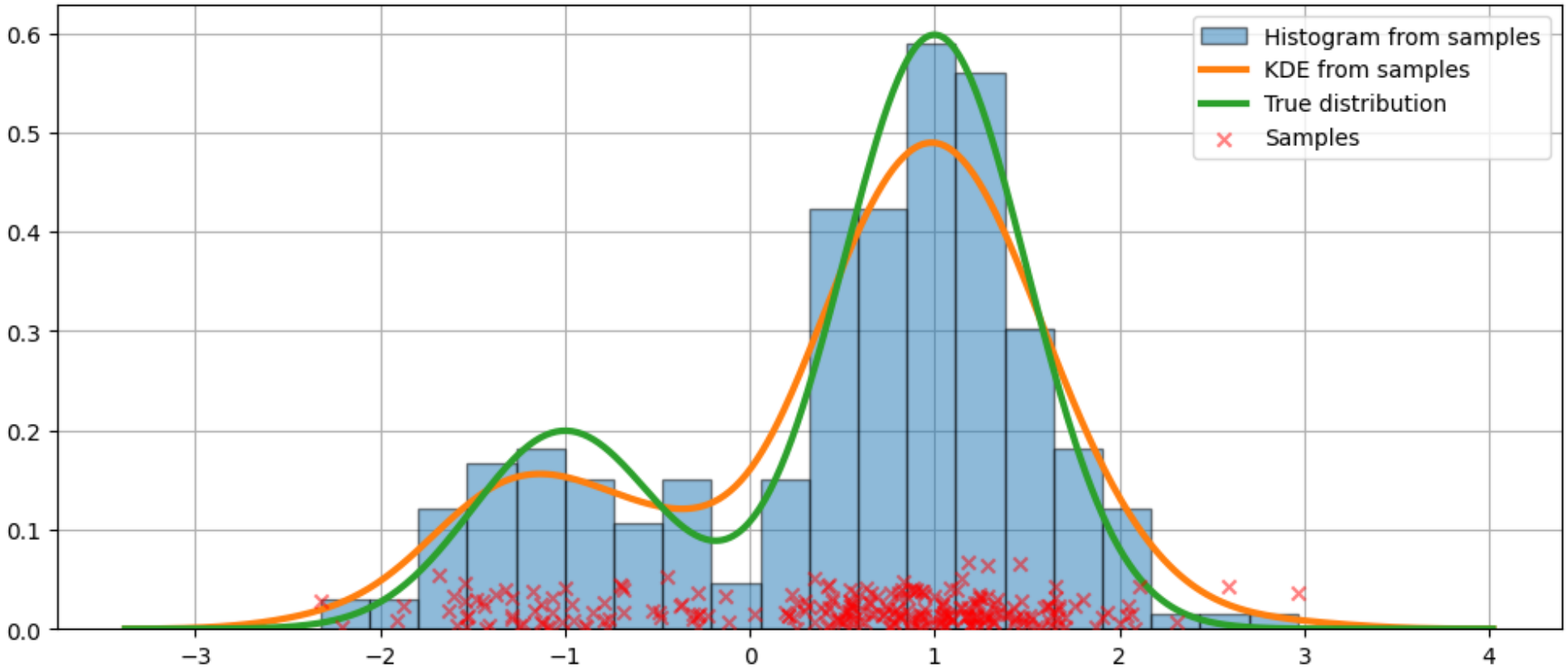




3. Анализ распределений и взаимосвязей переменных

Density Plot (KDE Plot)

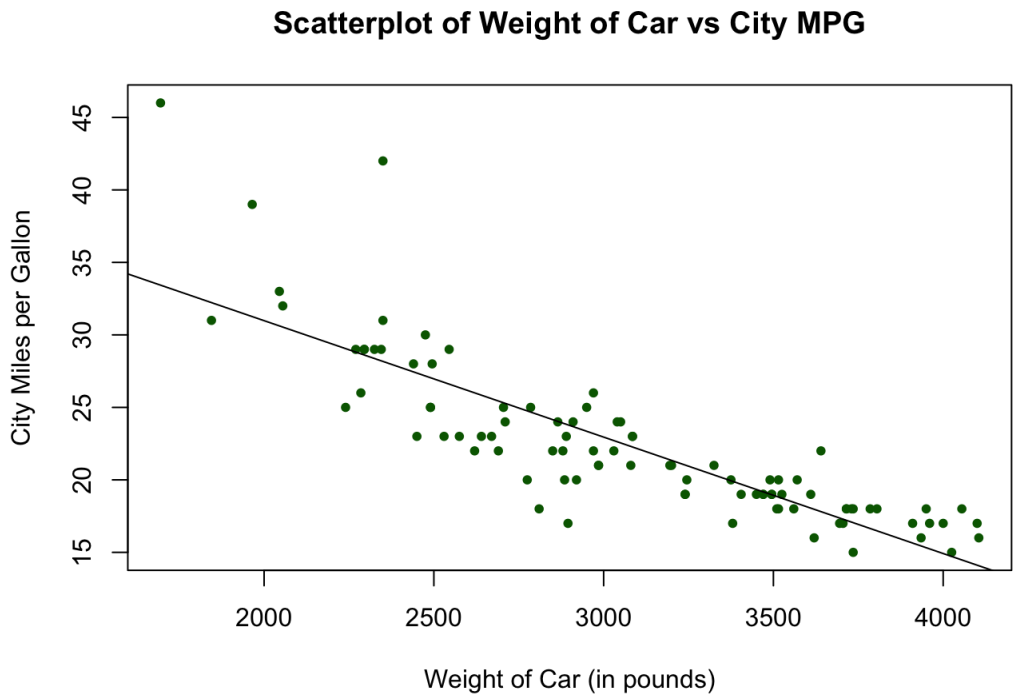
Показывает распределение числового признака. То же самое, что и Histogram, но в виде непрерывной аппроксимации (вместо отдельных столбиков смотрим на непрерывную линию)

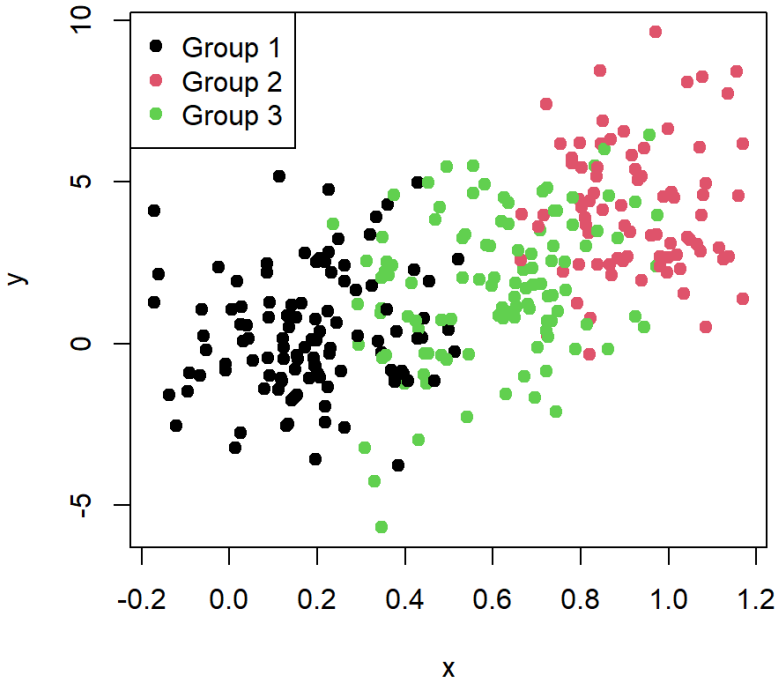


3. Анализ распределений и взаимосвязей переменных

Scatterplot

Показывает заимосвязь двух числовых (как правило) переменных

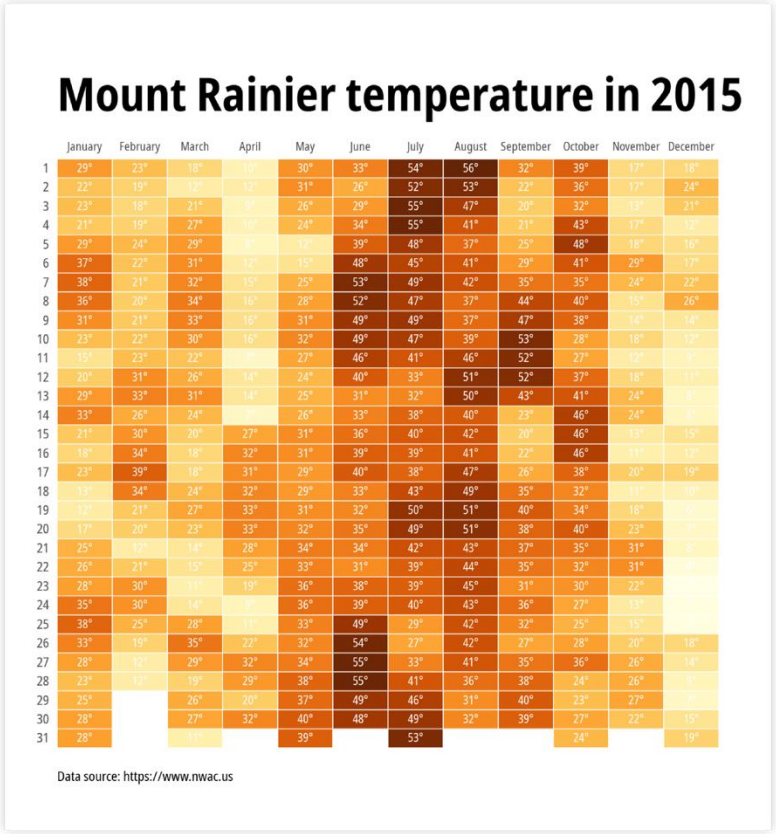


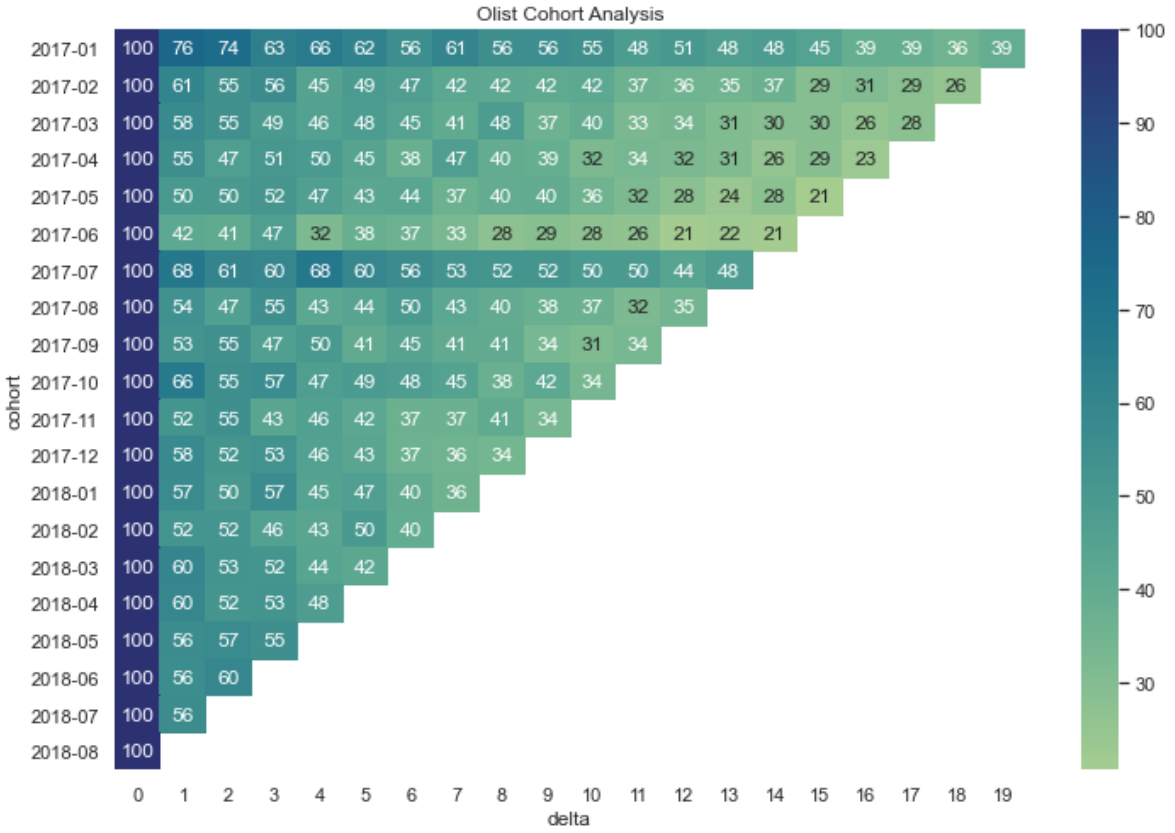


3. Анализ распределений и взаимосвязей переменных

Heatmap

Удобно позволяет смотреть «интенсивность» метрики. Используется в корреляционных матрицах, когортгном анализе.

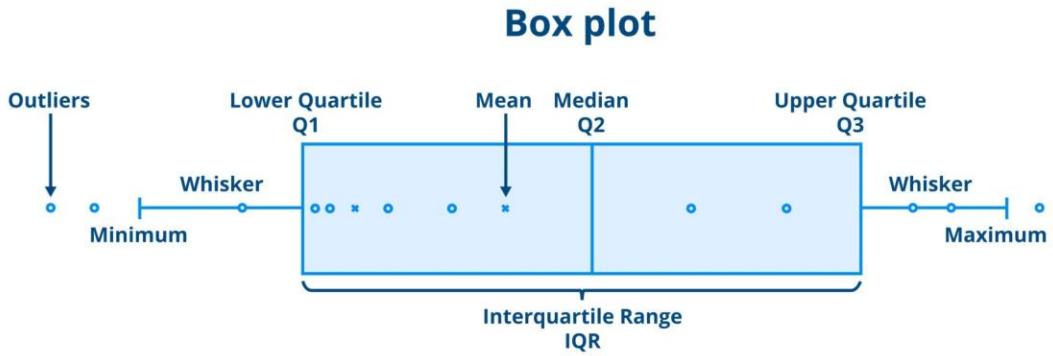


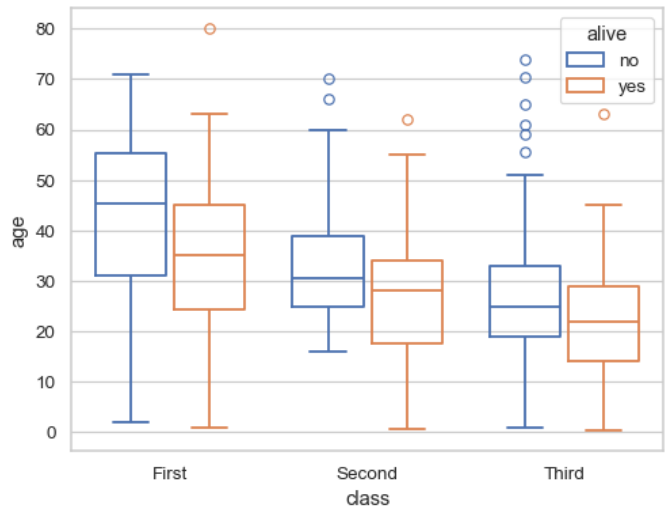


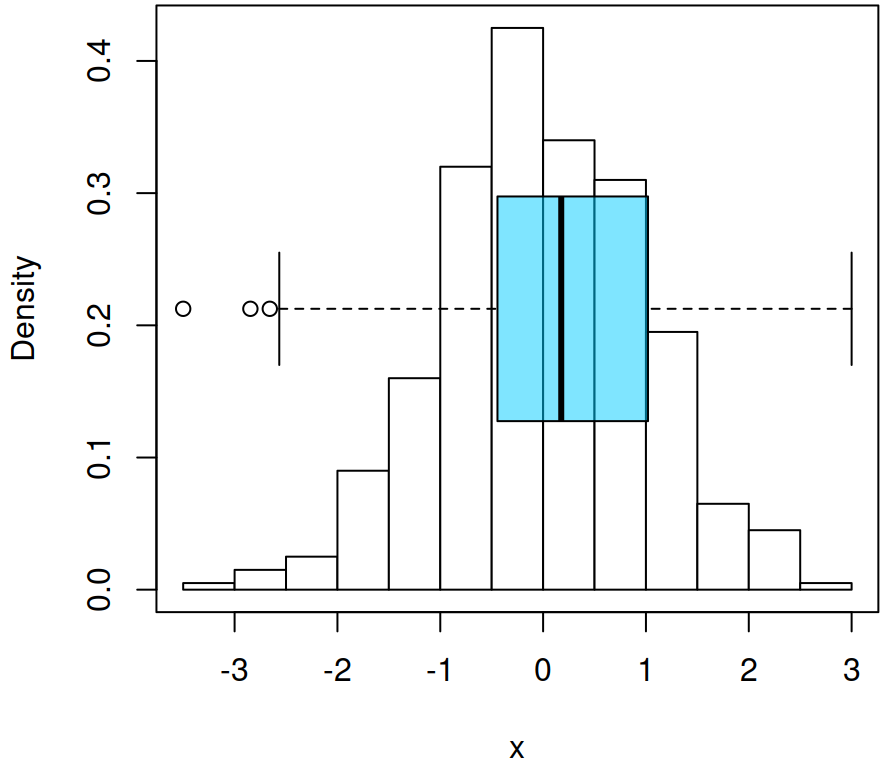
3. Анализ распределений и взаимосвязей переменных

Box Plot

”Ящик с усами”. Позволяет отобразить распределение (которое мы смотрели на Histogram) в один столбик.



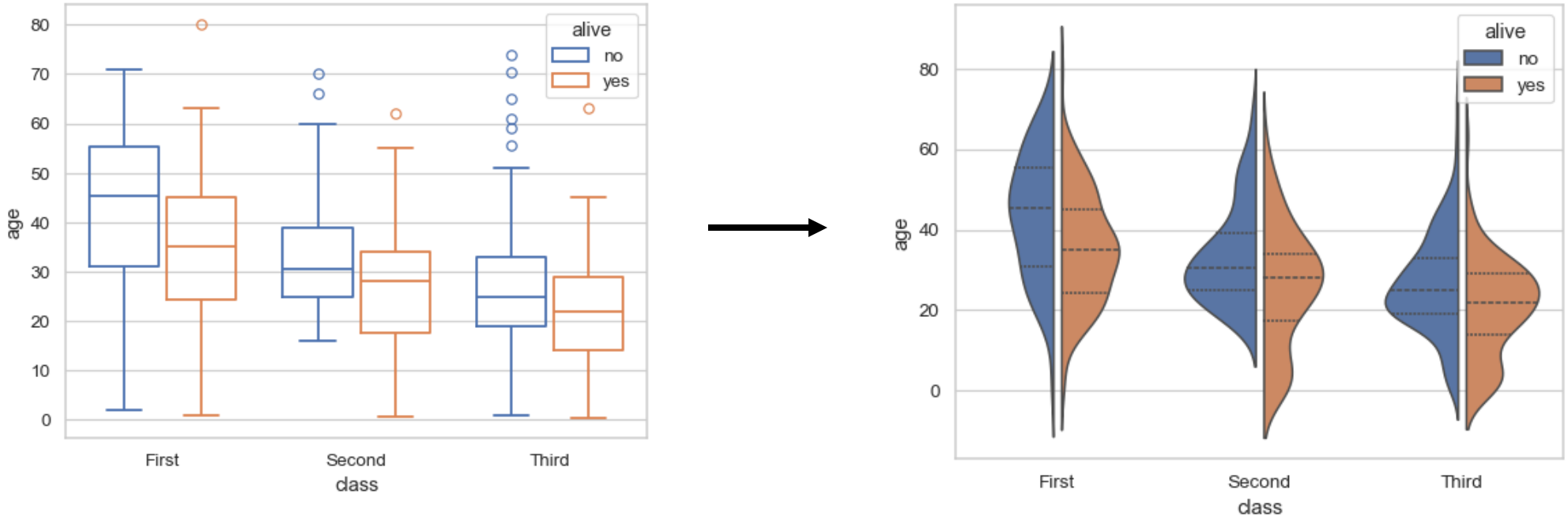




3. Анализ распределений и взаимосвязей переменных

Violin Plot

Почти то же самое, что и BoxPlot, только теперь в каждом “столбике” мы рисуем само распределение (вместо простых отметок о медиане / квантилях).



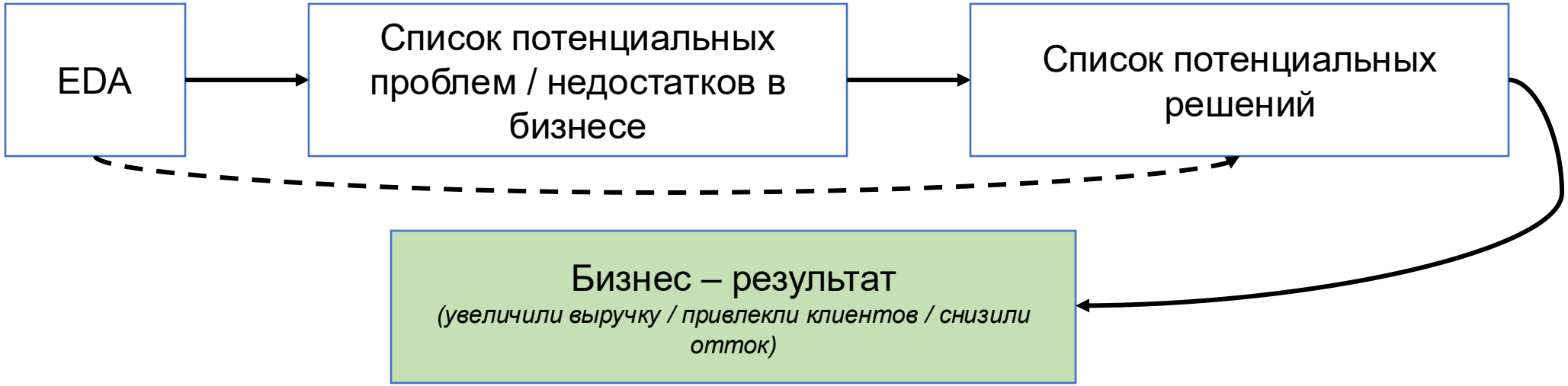
Возвращаемся к EDA

4. Формирование рекомендаций или выводы для ML

Мы построили много графиков: смотрели распределения / boxplot; посчитали много статистик (средние / медианы / перцентили); смотрели на взаимосвязи переменных.

Зачем?

1. Для того, чтобы сформировать список гипотез / рекомендаций для бизнеса

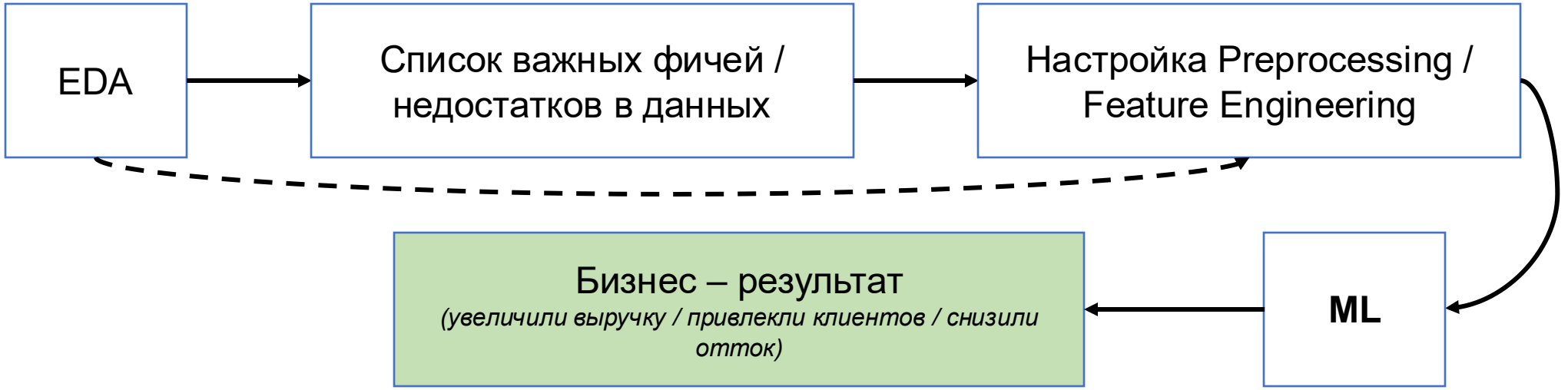


4. Формирование рекомендаций или выводы для ML

Мы построили много графиков: смотрели распределения / boxplot; посчитали много статистик (средние / медианы / перцентили); смотрели на взаимосвязи переменных.

Зачем?

2. Для того, чтобы подготовить pipeline ML модели



Переходим к практике!