Лекция 4

# Множественные тесты

Курс “Практическая аналитика данных”, 2021

образование

# План лекции

1. **Введение**. Множественные тесты: когда и зачем
2. **Дизайн эксперимента**. Специфика множественных тестов
3. **Оценка результатов эксперимента**. Виды ошибок при множественном тестировании. Таблица истинности для множественных статистических тестов
4. **Методы контроля ошибок при множественном тестировании.** FWER и методы его контроля. FDR и методы его контроля.
5. **Практическая часть.** Программная реализация поправок для множественного тестирования. Пример оценки результатов и формирования выводов.

# 1. Множественные тесты: когда и зачем

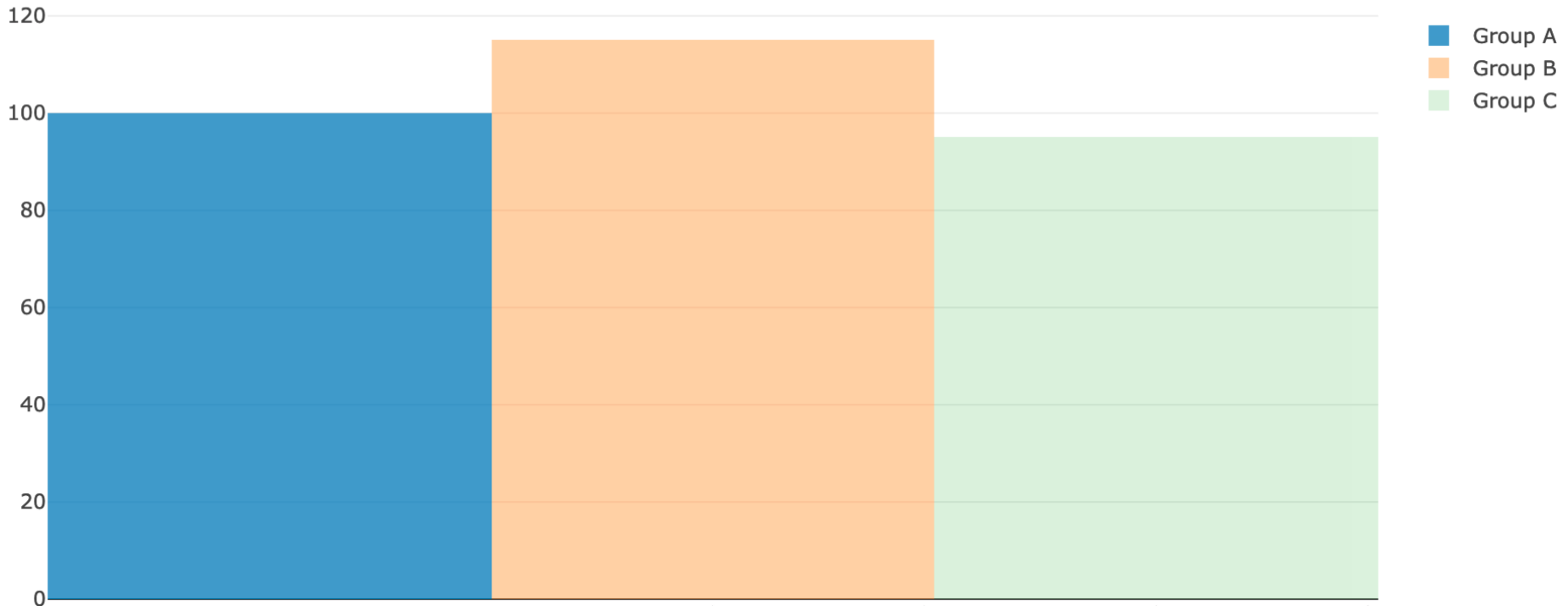
# Когда использовать множественные тесты

Основные цели использования:

- быстро протестировать влияние разных изменений (нескольких номиналов промокодов, нескольких вариантов экрана главной, несколько вариантов корзины);
- протестировать изменения, воздействие которых необходимо оценивать только совместно (несколько вариантов ступенек для лесенок тарифов, разные тарифы на размещение, разные варианты сбора за лид/целевого пользователя/проданный товар или покупку, совершенную у партнёра на вашем сайте, разные варианты выдачи и модели ранжирования);
- на подгруппы действуют различающиеся факторы, влияющие на результат (разные гео-зоны внутри города, разные минимальные расстояния от стартовой позиции курьера, на которого назначается заказ, до точки сбора заказа и/или до адреса покупателя, разные города);
- глобальный контроль и A/A-тесты .

# 2. Дизайн эксперимента

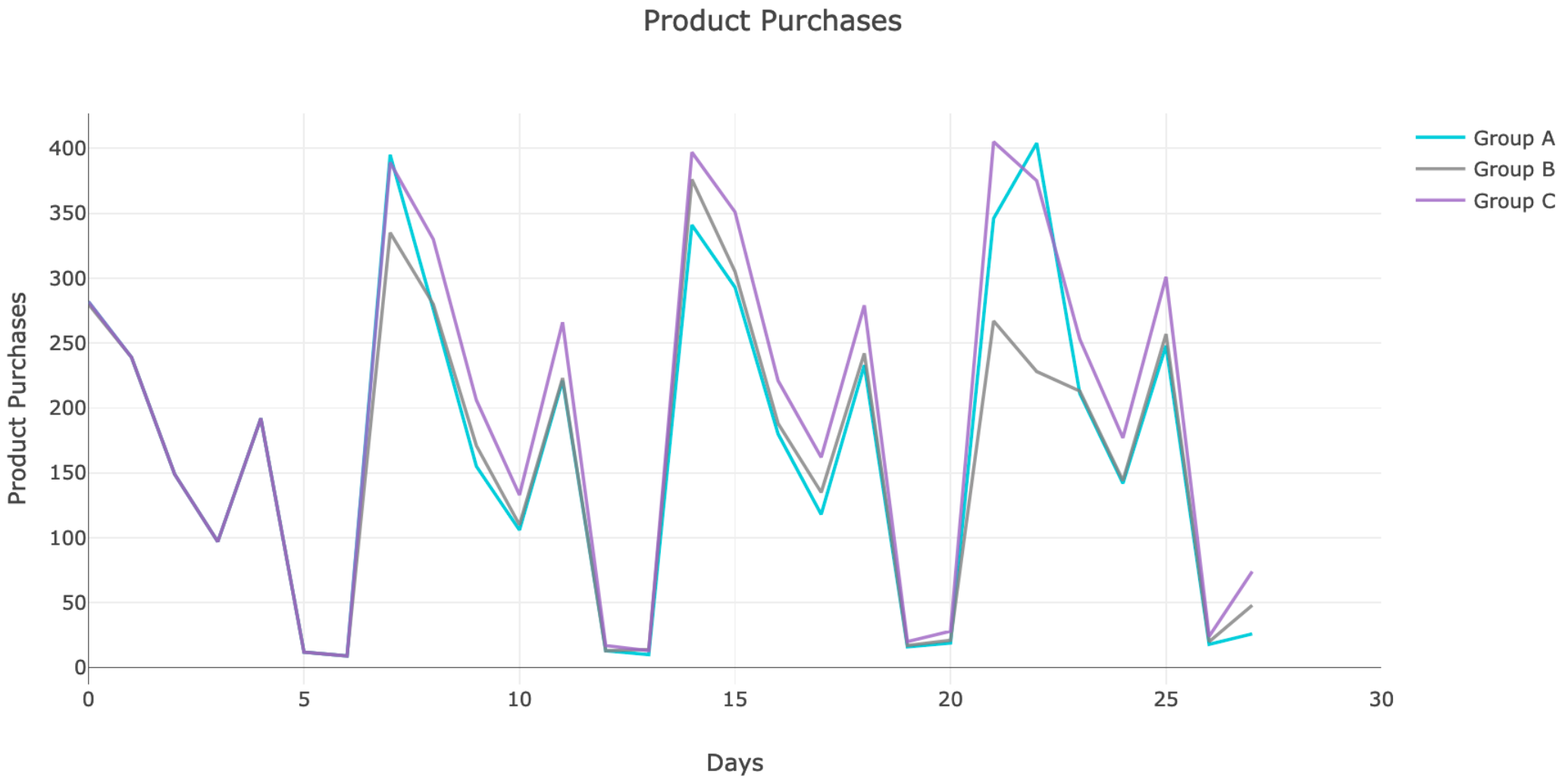
# Разбиение на группы



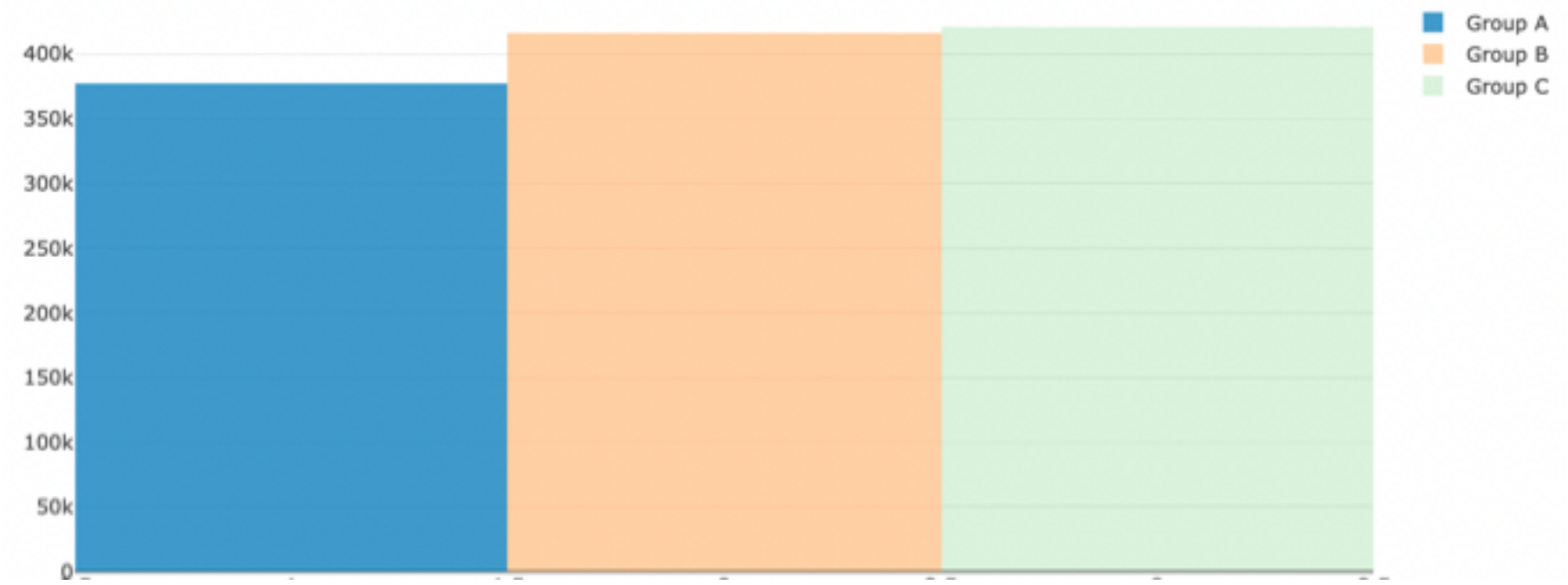

A/B/C-тест – поиск оптимальной цены:

A: 0%, B: +15%, C: -5%

# Динамика покупок



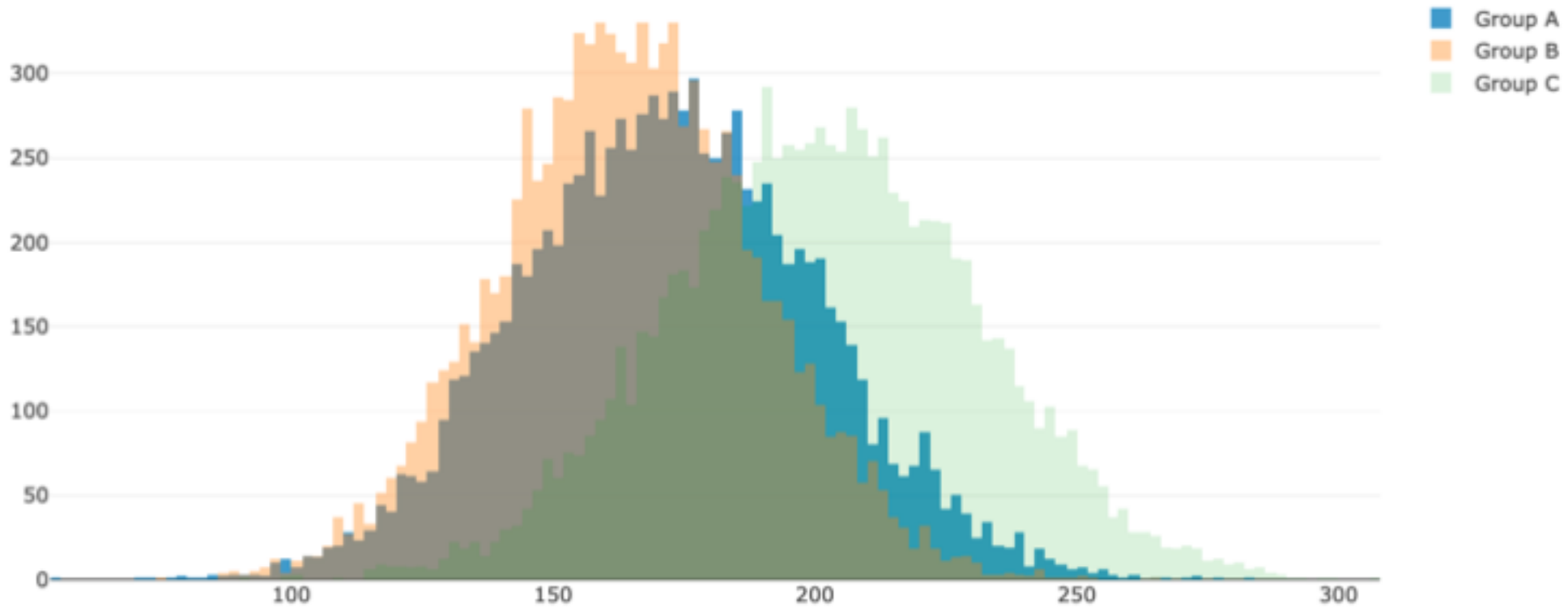
# Динамика покупок: выручка



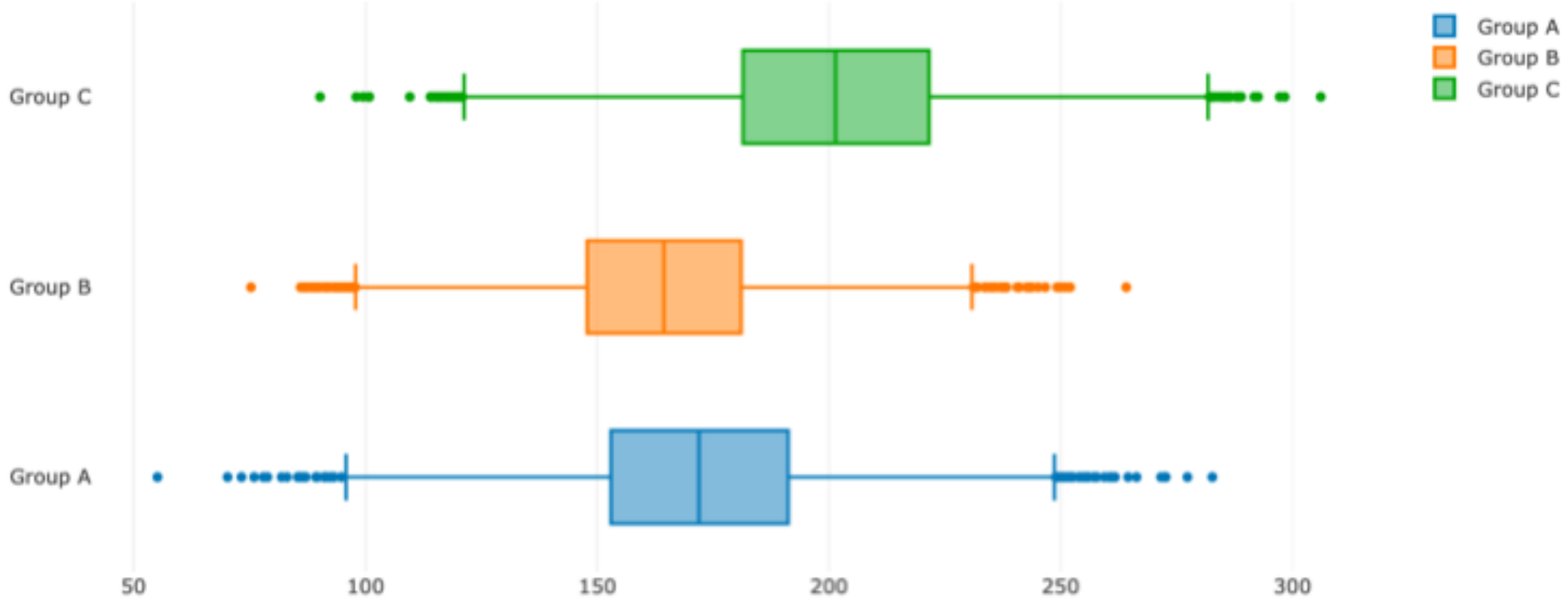

A/B/C-тест – поиск оптимальной цены:

A: 100%, B: +10.01%, C: +11.34%

# Распределение средних значений количества покупок в подгруппах



# Диаграмма размаха средних значений количества покупок в подгруппах



# 3. Оценка результатов эксперимента

# Оценка результатов

- Рассмотрим $k$ гипотез $H_0$:

$$H0i \quad vs. \quad H1i, \quad i = 1, \ldots, k$$

| | H0 *не отклонена* | H0 *отклонена* | Total |
|---|---|---|---|
| H0 *верна* | U | V | $k_0$ |
| H0 *неверна* | T | S | $k_1$ |
| Total | k-R | R | $k$ |

- $k0$ – количество верных нулевых гипотез,
- R – количество отклоненных нулевых гипотез.

# Оценка результатов

- Ошибка при множественном тестировании:

$$\mathbb{P}(\text{significant}) = 1 - (1 - \alpha)^k$$

- При $k = 3$ и $\alpha = 0.05$, $\mathbb{P} = 0.14$.

- Поправки на множественное тестирование – контроль ошибок:
  - $FWER$: $FWER = \mathbb{P}(V \geq 1)$
  - $FDR$: $FDR = \mathbb{E}(V/R|R > 0)$

# 4. Методы контроля ошибок при множественном тестировании.

# Методы контроля ошибок FWER и FDR

- *FWER – Familywise Error Rate* – групповая вероятность ошибки первого рода:
  - Метод Бонферрони
  - Метод Холма

- *FDR – False Discovery Rate* – ожидаемая доля ложных отклонений гипотез или частота ложных срабатываний:
  - Метод Бенджамини-Хохберга

# *4.1. FWER* и методы его контроля

# FWER

- *Familywise Error Rate* – групповая вероятность ошибки первого рода:

$$FWER = \mathbb{P}(V \geq 1)$$

- Два типа методов контроля:
  - *Одношаговая* процедура: одновременно изменить все *p*-значения;
  - *Последовательная* процедура: последовательная корректировка $p$-значения и адаптивная реакция на результат.

# Метод Бонферрони

- Рассмотрим $k$ гипотез $H_0$:

$$H_{0i} \, vs \, H_{1i}, \; i = 1, \ldots, k$$

- $p_1, \ldots, p_k$ – величины $p\text{-}value$ проверок $k$ гипотез $H_{0i}$

- Для заданных $p_1, \ldots, p_k$ основная гипотеза $H_{0i}$ отклоняется, если

$$p_i/k \geq \alpha.$$

# Метод Бонферрони

- Высокая вероятность *ошибок 2 рода*.

- Быстрое снижение мощности теста при при росте $k$.

*Пример*: *10* тестов, $\alpha = 0.05$,

тогда необходимо получить

$$p = 5 \cdot 10^{-3} < 0.01, _i$$

чтобы сказать, что разница значимая.

# Метод Холма

- Развитие метода поправки Бонферрони;
- Метод предполагает последовательное изменение $p\text{-}value$:

1) нисходящая процедура – сортировка реальных $p\text{-}value$ по возрастанию:

$$p_1 \leq \cdots \leq p_k$$

2) нисходящая процедура – корректировка $\alpha$ :

$$\alpha'_i = \alpha / i$$

3) проверка до первой отвергнутой гипотезы $H_{0i}$ : $p_i \geq \alpha'_i$

отвергается $H_{0i}$ и все $H_{0j}, j > i$.

# *4.2. FDR* и методы его контроля

# FDR

- *False Discovery Rate* – ожидаемая доля ложных отклонений гипотез или частота ложных срабатываний:

$$FDR = \mathbb{E}(V/R|R > 0)$$

- Более строгий критерий:

$$\frac{\mathbb{E}(V)}{m} \leq FDR \leq FWER \leq \mathbb{E}(V)$$

# Метод Бенджамини-Хохберга

- Метод предполагает последовательное изменение $p\text{-}value$:

1) сортировка реальных $p\text{-}value$ по возрастанию:

$$p_1 \leq \cdots \leq p_k$$

2) восходящая процедура – корректировка $\alpha$:

$$\alpha'_i = i \cdot \alpha / k$$

3) проверка до первой отвергнутой гипотезы $H_{0i}$ : отвергается $H_{0i}$ и все $H_{0j}$, $j > i$.

# 5. Программная реализация

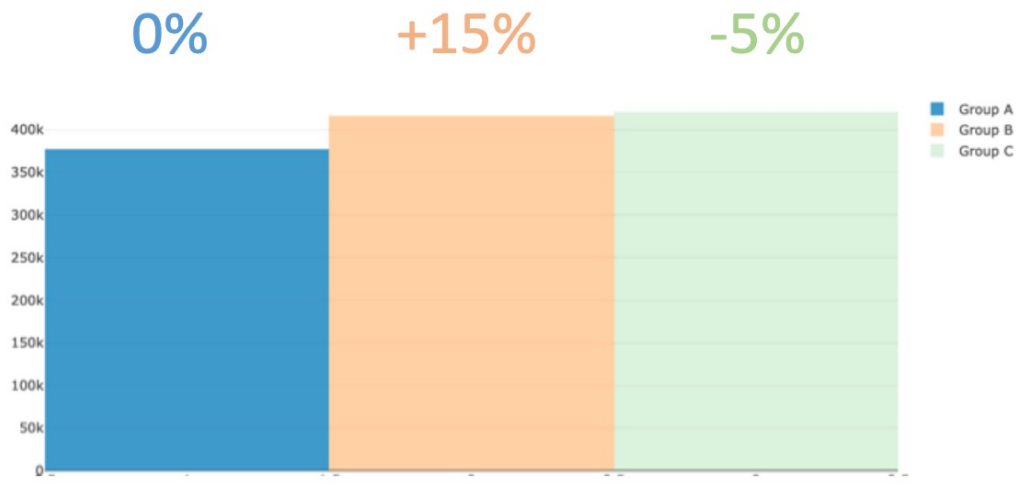
# Программная реализация

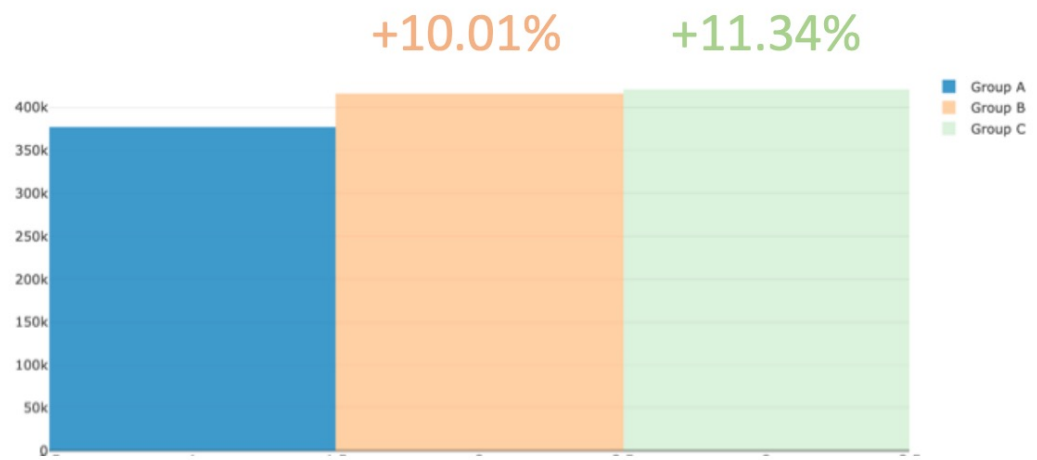
```
from scipy.stats import ttest_ind
from statsmodels.sandbox.stats.multicomp import multipletests
from bootstrapped import bootstrap as bs
from bootstrapped import compare_functions as bs_compare
from bootstrapped import stats_functions as bs_stats

# FWER: Бонферрони
bs_ab_estims = bs.bootstrap_ab(np.array(group_A), np.array(group_B), bs_stats.mean,
                               bs_compare.difference, num_iterations=5000,
                               alpha=0.05/3, iteration_batch_size=100, scale_test_by=1, num_threads=4)

bs_bc_estims = bs.bootstrap_ab(np.array(group_B), np.array(group_C), bs_stats.mean,
                               bs_compare.difference, num_iterations=5000,
                               alpha=0.05/3, iteration_batch_size=100, scale_test_by=1, num_threads=4)

bs_ac_estims = bs.bootstrap_ab(np.array(group_A), np.array(group_C), bs_stats.mean,
                               bs_compare.difference, num_iterations=5000,
                               alpha=0.05/3, iteration_batch_size=100, scale_test_by=1, num_threads=4)
```

# Программная реализация

```
from scipy.stats import ttest_ind
from statsmodels.sandbox.stats.multicomp import multipletests
from bootstrapped import bootstrap as bs
from bootstrapped import stats_functions as bs_stats


bs_data_a = bs.bootstrap(np.array(group_A), stat_func=bs_stats.mean,
                         num_iterations=10000, iteration_batch_size=300, return_distribution=True)
bs_data_b = bs.bootstrap(np.array(group_B), stat_func=bs_stats.mean,
                         num_iterations=10000, iteration_batch_size=300, return_distribution=True)
bs_data_a = bs.bootstrap(np.array(group_C), stat_func=bs_stats.mean,
                         num_iterations=10000, iteration_batch_size=300, return_distribution=True)

stat_ab, p_ab = ttest_ind(pd.DataFrame(bs_data_a), pd.DataFrame(bs_data_b))
stat_bc, p_bc = ttest_ind(pd.DataFrame(bs_data_b), pd.DataFrame(bs_data_c))
stat_ac, p_ac = ttest_ind(pd.DataFrame(bs_data_a), pd.DataFrame(bs_data_c))

print(sorted([p_ab, p_bc, p_ac]))

# FWER: Холм
print("FWER: " + str(multipletests(sorted([p_ab, p_bc, p_ac]), alpha=0.05, method='holm', is_sorted = True)))

# FDR: Бенджамини-Хохберг
print("FDR: " + str(multipletests(sorted([p_ab, p_bc, p_ac]), alpha=0.05, method='fdr_bh', is_sorted = True)))
```

# Пример интерпретации результатов




- Повышая цену в сегменте, мы можем повысить суммарную выручку, однако снизив цену можно увеличить выручку ещё больше за счет дополнительных покупок;
- Множественный тест, нивелировав недельные колебания спроса на продукт, позволил увидеть полезный insight: снижение цены приводит к большему увеличению выручки, чем её повышение.

# Спасибо за внимание ☺

Следующая лекция: временные ряды