1. **Определяем цель**

Возьмем вымышленный сайт онлайн-магазина, торгующего виджетами.

Отдел маркетинга хочет увеличить продажи, рассылая рекламные электронные письма с кодом купона на скидки для виджетов. Это изменение является потенциальным обновлением бизнес-модели, поскольку компания ранее не предлагала купоны. Однако сотрудник компании недавно прочитал о том, что после добавления кода купона некий

предприниматель потерял значительную долю прибыли, а также о том, что отказ от кодов купонов рассматривают как положительный пример на GUI.org. С учетом этих внешних данных возникли опасения, что добавление поля для кода купона снизит доход, даже если купонов нет; т. е. сам факт упоминания о купоне замедлит работу пользователей и заставит их искать коды или даже отказаться от покупки.

Мы хотим оценить влияние простого добавления поля для ввода кода купона. Наша цель просто оценить влияние этого поля кода купона на доход и проверить обоснованность

опасений, что это будет отвлекать людей от оформления заказа. Поскольку это простое изменение, мы протестируем две реализации пользовательского интерфейса.

1. **Определяем метрики**

Чтобы измерить влияние изменения, нам нужно определить показатели цели или показатели успеха: - очевидным показателем успеха для этого эксперимента может быть доход.

Не нужно использовать в качестве показателя саму сумму дохода, поскольку она зависит от количества пользователей в каждом варианте.

Возможное решение: Необходимо нормализовать ключевые показатели по фактическим размерам выборки, чтобы получить доход на пользователя.

Следующий важный вопрос – решить, каких пользователей следует учитывать в знаменателе при вычислении дохода на пользователя:

А) всех пользователей, посетивших сайт. Это допустимый вариант, однако он дает размытую выборку, потому что в нее входят пользователи, которые никогда не начинали оформление заказа в варианте, где было внесено изменение.

Б) только пользователей, завершивших процесс заказа. Этот выбор неверен, поскольку предполагает, что изменение повлияет на сумму покупки, а не на процент пользователей, совершивших покупку.

В) только пользователей, которые начинают оформление заказа. Это лучший выбор. Мы включаем всех потенциально затронутых пользователей, но исключаем незатронутых (пользователей, которые никогда не начинают оформление заказа), искажающих наши результаты.

**3. Формулируем гипотезу**

Первоначальная гипотеза такова: «Добавление поля кода купона на страницу оформления заказа снизит доход».

Исходя из вышеперечисленных шагов уточняем гипотезу: Добавление поля кода купона на страницу оформления заказа снизит доход на пользователя для пользователей, которые начинают оформление заказа.

**4. Настраиваем эксперимент**

Частотный метод: метод значения p и метод кривой roc:

Вероятность того, что каждый пользователь отображается как Идея A или B, равна 0,5, и преобразование выполняется напрямую с неизвестной вероятностью p\_A = 0,04 или p\_B = 0,05.

import scipy.stats as stats

# Истинная вероятность

p\_A = 0.05

p\_B = 0.04

#User Flow

n\_users = 13500

n\_A = stats.binom.rvs(n=n\_users, p=0.5, size=1)[0]

n\_B = n\_users - n\_A

#Conversion Strategy

conversions\_A = stats.bernoulli.rvs(p\_A, size=n\_A)

conversions\_B = stats.bernoulli.rvs(p\_B, size=n\_B)

print("creative A was observed {} times and led to {} conversions".format(n\_A, sum(conversions\_A)))

print("creative B was observed {} times and led to {} conversions".format(n\_B, sum(conversions\_B)))

В нашем случае точный тест Фишера и тест Барнарда кажутся наиболее релевантными, поскольку нас интересует коэффициент конверсии, полученный из биномиального распределения.

Мы будем использовать тест Фишера, который обладает характеристиками точного теста. Причина, по которой он называется так, заключается в том, что его p-значение может быть точно рассчитано.

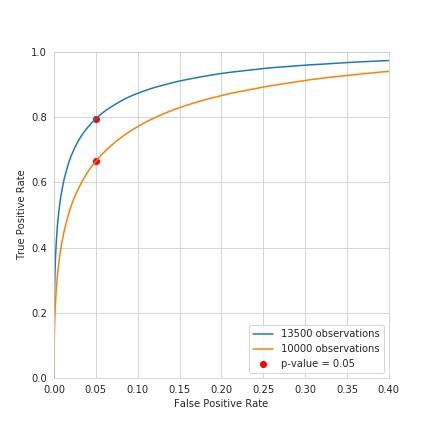
H0: эффект от добавления поля аналогичен (p\_A = p\_B)

H1: одно изменение на сайте лучше другого

Хорошая практика требует от нас определения статистических возможностей теста. Его можно рассчитать на основе результатов моделирования, разработанного в данной среде.

Наша цель - 0,8, и используемые нами наблюдения будут откалиброваны для достижения этого порога.

Для удобства мы рассматриваем только альтернативную гипотезу H1 - «альтернативная гипотеза p\_A = 0,04 или p\_B = 0,05», которая позволит нам легко рассчитать эффект от теста.



Как правило, α = 0,05 является пределом приемлемости области отклонения значения **p** (т. е. Частоты ложных тревог). Для этого порога потребуется не менее 13 500 наблюдений.

Предварительная интерпретация результатов теста:

Зададим вопрос: какие выводы мы можем сделать на основании результатов наших тестов и значения **p**?

Можем ли мы оценить вероятность того, что ситуация А лучше, чем ситуация Б?

Если да, можем ли мы оценить прогресс между ними?

**Ответ на оба вопроса - нет.**

Мы узнали, что при предположении (H0) вероятность того, что p-значение является просто наблюдением, по крайней мере экстремальна. Но это также подчеркивает наиболее важное ограничение p-значения: Значение **p** не является мерой размера эффекта! Даже если получен положительный результат, он не дает информации о том, что A лучше, чем B.

Чтобы извлечь информацию о размере эффекта, вы можете использовать другой более мощный инструмент: доверительный интервал.

Доверительный интервал представляет собой разумный диапазон значений неизвестного параметра, относящийся к уверенности в том, что истинный параметр находится в рекомендованном диапазоне.

Это дает нам больше понимания и глубокого понимания, а также дает нам информацию о вероятности и неопределенности наших оценок.

Однако у них есть и свои недостатки:

Нам нужно выбрать разные методы для определения этих интервалов, в зависимости от некоторых предположений: доверительного интервала биномиального распределения.

Доверительный интервал значений p\_A и p\_B не будет напрямую преобразован в доверительный интервал разницы между p\_A и p\_B!

К счастью, есть методы прямого вычисления доверительного интервала разницы, но нам все равно нужно выбирать из десятков методов Один вид. Например, мы можем использовать «простейший» метод Вальда, не требующий коррекции непрерывности:

Используем опыт 𝑝̂, чтобы оценить p\_A и p\_B, а z соответствует α / 2-процентилю нормального распределения. Следовательно, для 95% доверительного интервала (то есть α = 0,05) соответствующее значение z будет 1,96.

**Числовое приложение даст нам интервал [0,0009; 0,0171] с центром в 0,009, а не перекрывается с 0, как мы надеялись!**

**5. Проводим эксперимент**

Байесовский метод

В этой части мы будем использовать библиотеку PyMC3 Python, которая позволяет легко строить байесовские непараметрические модели.

Важно помнить, что наша цель состоит в том, чтобы получить точные и актуальные результаты, и, что не менее важно, поколение может быть разделено с любыми другими заинтересованными сторонами (даже с ненаучными заинтересованными сторонами) И понятные KPI. Он будет нести как можно больше информации. Байесовский метод предоставляет нам инструменты, необходимые для выполнения этой операции. Он позволяет нам точно вычислить то, что нам нужно: апостериорное распределение p\_A и p\_B, а именно P (p\_A | X) и P (p\_B | X) и P (p\_A - p\_B> 0 | X), то есть вероятность того, что креатив А принесет больше конверсий, чем креатив Б. Выберем неинформированный унифицированный априор для p\_A и p\_B.

import pymc3 as pm

with pm.Model() as model:

n\_users = 10000

# Определить случайные и детерминированные переменные (построить сеть)

#Количество пользователей

n\_A = pm.Binomial("n\_A", n\_users, 0.5)

n\_B = pm.Deterministic("n\_B", n\_users - n\_A)

# Преобразование количества

conversions\_A = pm.Binomial("conversions\_A", n\_A, p\_A)

conversions\_B = pm.Binomial("conversions\_B", n\_B, p\_B)

observed\_conversions\_A = pm.Deterministic('observed\_conversions\_A', conversions\_A)

observed\_conversions\_B = pm.Deterministic('observed\_conversions\_B', conversions\_B)

p\_estimates = pm.Uniform("p\_estimates", 0, 1, shape=2)

delta = pm.Deterministic("delta", p\_estimates[1] - p\_estimates[0])

# Предоставлять данные наблюдений в сеть

obs\_A = pm.Binomial("obs\_A", n\_A, p\_estimates[0], observed=observed\_conversions\_A)

obs\_B = pm.Binomial("obs\_B", n\_B, p\_estimates[1], observed=observed\_conversions\_B)

# Запустить алгоритм MCMC

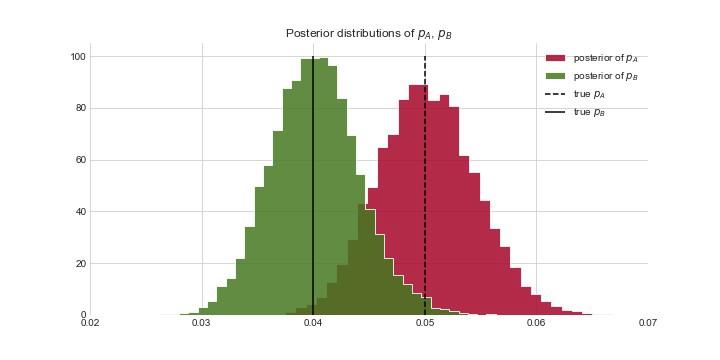
start = pm.find\_MAP()

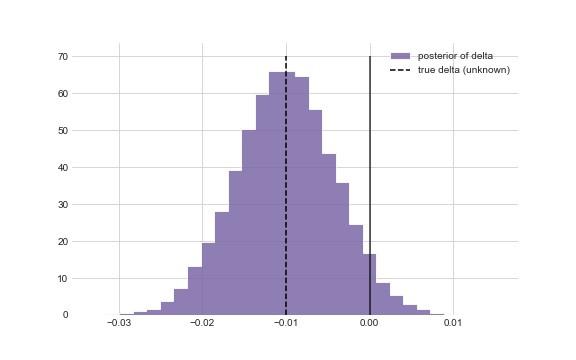
step = pm.Metropolis()

trace = pm.sample(50000, step=step)

burned\_trace = trace[1000:]

В сравнении с обычными методами А/Б тестирования, они дадут нам больше информации, потому что теперь мы можем нарисовать доверительный интервал для любого альфа-уровня путем непосредственной выборки из апостериорного объекта.





Важно отметить, что, поскольку p\_A и p\_B могут быть зависимыми, приращение не может быть вычислено из апостериорного значения p\_A и p\_B.

**6. Анализируем результаты**

Мы можем просто вычислить апостериорную вероятность меньше 0:

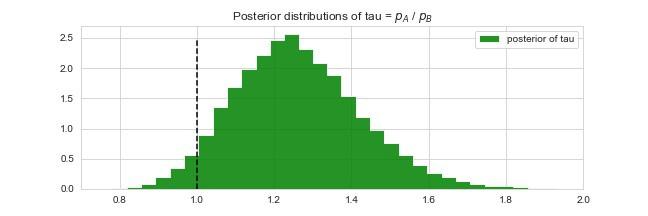
np.mean (delta\_samples < 0)

Получаем 0,956, что означает, что коэффициент конверсии креатива А на 96% выше, чем у креатива Б.

Можно пойти дальше. Представим, что переход с дизайна B на дизайн A стоит дорого, и только улучшение производительности не менее чем на 5% может быть прибыльным.

Просто вставим новую детерминированную переменную Tau = p\_A / p\_B в нашу сеть, а затем выберите апостериорное распределение.

tau = pm.Deterministic(“tau”, p\_estimates[0] / p\_estimates[1])



Апостериорное распределение тау, оцененное по выборке MCMC.

np.mean (tau\_samples > 1.05)

На этот раз мы получаем 0,91 ... не так решительно, как 95%, на которые мы обычно надеемся. Если нам нужно больше уверенности, нам просто нужно проводить A / B-тесты дольше.