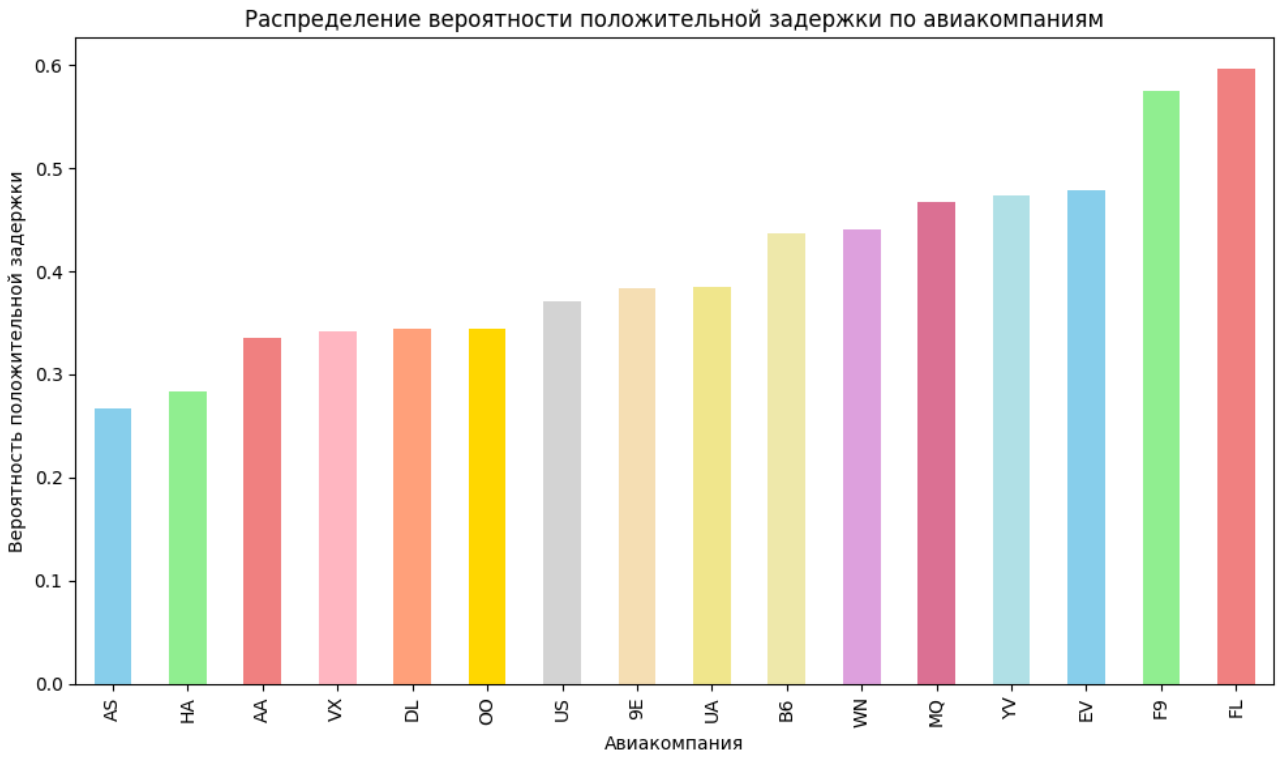
1. Определение вероятности положительной задержки для авиакомпаний.

Анализ:

Анализ проводится с помощью следующих функций:

- df.groupby('carrier'): Группирует данные по столбцу ‘carrier’ (авиакомпания).
- ['Положительная задержка'].mean(): Вычисляет среднее значение столбца ‘Положительная задержка’ (вероятность положительной задержки) для каждой группы авиакомпаний.
- sort_values(ascending=True): Сортирует полученные значения по возрастанию.
- plot(kind='bar', ...): Строит столбчатую диаграмму с указанными цветами.
- d='bar', ...): Строит столбчатую диаграмму с указанными цветами.

Строится столбчатая диаграмма, где ось X - авиакомпания, ось Y - вероятность положительной задержки:



Вывод: в результате анализа мы получили распределение вероятности положительной задержки для всех компаний. Из граика видно, что некоторые компании имеют большую вероятность задержки чем другие.

2. Построение гистограммы расстояния перелета distance, определение коротких, средних и далеких перелетов, определение направления дальних перелетов и определение времени задержки в каждой группе.

Анализ:

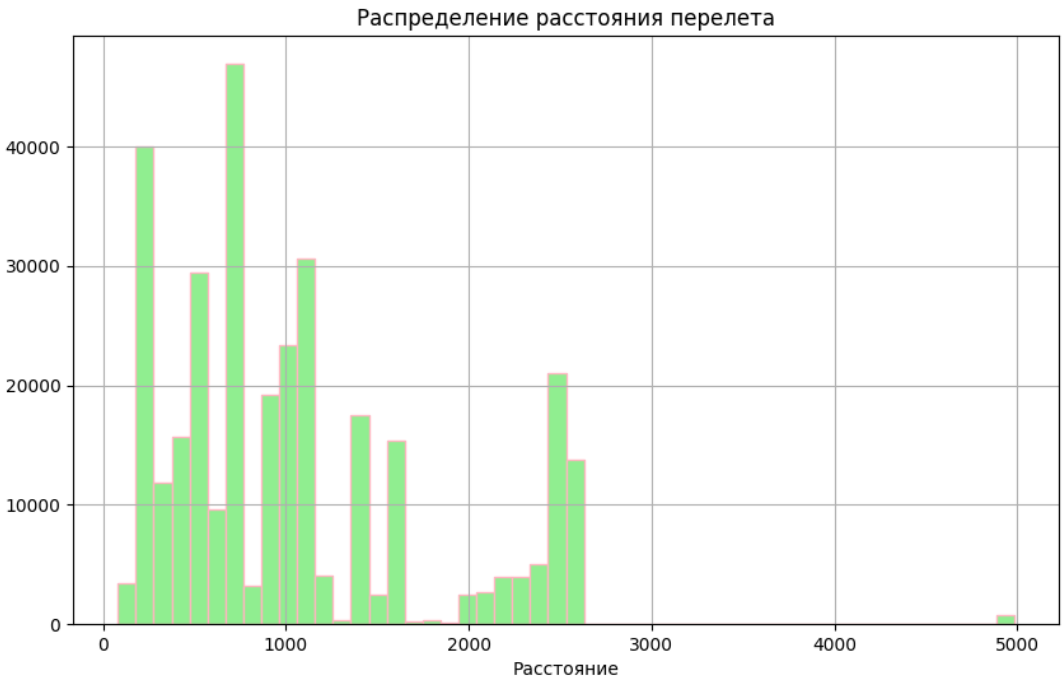
plt.hist(df['distance'], ...): Строит гистограмму распределения расстояния ‘distance’.

df['distance'].quantile([0.25, 0.5, 0.75]): Вычисляет квартили (25%, 50% и 75%) для распределения расстояния.

pd.cut(df['distance'], ...): Создает новый столбец ‘Категория перелета’, разделяя данные по расстоянию на три группы (короткое, среднее, длинное) на основе полученных квартилей.

df[df['Категория перелета'] == 'Длинный']['dest'].unique(): Выводит список уникальных пунктов назначения (‘dest’) для рейсов, отнесенных к категории ‘Длинный’.

df.groupby('Категория перелета', observed=False)['dep_delay'].mean(): Группирует данные по категориям расстояния и вычисляет среднее время задержки вылета (‘dep_delay’) для каждой группы.



Определяются квантили: 25-й, 50-й (медиана) и 75-й.

Квантили делят данные на три группы:

Короткие: Расстояние до 25-го квантиля. (509 миль)

Средние: Расстояние от 25-го до 50-го квантиля. (888 миль)

Длинные: Расстояние от 50-го квантиля. (1389 миль)

Определяются направления для рейсов в группе длинные:

'IAH' 'MIA' 'BOM' 'FLL' 'MCO' 'PBI' 'TPA' 'LAX' 'SFO' 'DFW''LAS' 'MSP'

'RSW' 'SJU' 'PHX' 'DEN' 'SNA' 'MSY''SLC' 'XNA' 'SEA''SRQ' 'MEM' 'SAN'

‘JAC’ 'HNL' 'AUS' 'STT' 'EGE' 'HOU' 'LGB' 'BUR' 'MC! 'ЗАТ' 'PDX' 'SJC'

'OMA' 'OAK' 'SMF' 'DSM' 'PЗE' 'TUL' 'OKC' 'HDN' 'BZN' 'MTJ' 'EYW' 'PЗP ' ABQ' 'STL' 'AMC'

Рассчитывается среднее время задержки вылета (dep_delay) для каждой группы перелетов.

- короткий: ~13 мин

- средний: ~14,2 мин

-длинный: ~11,5 мин

Вывод:

Из исследования видно, что

1. Большинство перелетов относятся к коротким

2. Время задержки зависит от длины перелета и для далеких перелетов оно меньше чем для коротких, это говорит о том, что далекие перелеты менее подвержены задержкам.

3. Дальные перелеты направлены в разные города, что говорит о том что это не имеет или имеет малое статистическое значение.

3. Построение графика времени задержки перелетов по месяцам, проверка гипотезы о равенстве средних в январе и феврале на уровне значимости 0.05 и на уровне значимости 0.01.

Анализ:
Для анализа использовались следующие функции:
pd.to_datetime(df['month'], format='%m').dt.month_name(): Преобразует столбец ‘month’ в формате месяца (01, 02, …) в формат полного названия месяца (January, February, …).
sns.pointplot(data=df, ...): Строит точечный график среднего времени задержки вылета по месяцам.

errorbar=('ci', 95): Добавляет 95% доверительные интервалы к точкам на графике.
stats.ttest_ind(january_data, february_data): Проводит t-тест для сравнения средних значений задержек в январе и феврале.

Результаты статистического теста:

На уровне значимости 0,05 гипотеза о равенстве средних времен задержки вылета в январе и феврале отвергается. Это означает, что существует статистически значимое различие между средним временем задержки в этих двух месяцах.

На уровне значимости 0,01 эта же гипотеза не отвергается. Это означает, что на данном уровне значимости данные не предоставляют достаточных доказательств для вывода о различии средних значений задержек в январе и феврале.

Вывод: используя тест стьюдента мы показали, что на уровне значимости 0,05 существуют статистически значимые отличие между средним временем задержек в январе и феврали. Но на уровне 0,01 нельзя заявить о различии между временем задержки в этих месяцах.

4.Нахождение коэффициента корреляции между временем полета и пройденным расстоянием, поиск коэффициентов линейной регрессии.

Анализ:

df['distance'].corr(df['air_time']): Вычисляет коэффициент корреляции между расстоянием ‘distance’ и временем полета ‘air_time’.

sns.scatterplot(data=df, ...): Строит точечную диаграмму в осях distance (x) и air_time (y).

stats.linregress(df['distance'], df['air_time']): Вычисляет коэффициентылинейной регрессии.

plt.plot(x_values, y_values, ...): Рисует линию регрессии на графике.

Коэффициент корреляции: ~0,99

Коэффициенты линейной регресии:

Slope ~0,126

Intrcept ~18,47

Вывод:

Коэффициент корреляции: ~0,99 что говорит о сильной зависимости между временем и длиной полета.

Линия линейной регрессии показывает, что с увеличением расстояния время полета также увеличивается.

Интерпретация параметров :

- Свободный член(Intrcept): Этот параметр модели представляет собой ожидаемое время в полете при нулевом расстоянии

- Наклон линии регрессии (Slope): Этот параметр определяет, насколько увеличивается (или уменьшается) время в полете при увеличении расстояния на единицу

5. Построение нормированной гистограммы распределения задержки прилета по тем рейсам, которые вылетели в пределах +/-15 минут от времени в расписании. Предположение о том, каким распределением может описываться полученная гистограмма, оценка параметров этого распределения и нанесние графика плотности на график с гистограммой.

Анализ:

df[(df['dep_delay'] >= -15) & (df['dep_delay'] <= 15)]: Фильтрует данные и оставляет рейсы с задержкой вылета в пределах ±15 минут.

sns.histplot(..., kde=True, stat='density', ...): Строит нормированную гистограмму распределения задержки прилета.

norm.fit(df_within_15_minutes['arr_delay']): Оценивает параметры нормального распределения (среднее значение и стандартное отклонение).

norm.pdf(xety, m, std): Вычисляет значение плотности вероятности нормального распределения.

plt.plot(xety, pwr, ...): Рисует график плотности на гистограмме.

Дополнительный вопрос**:** Для проверки гипотезы о выбранном распределении можно использовать тест Хи-квадрат:

Разбиение данных на интервалы:

intervals = np.linspace(df_within_15_minutes['arr_delay'].min(), df_within_15_minutes['arr_delay'].max(), 10): Создает 10 равномерно распределенных интервалов (bins) для данных о задержке прилета (df_within_15_minutes['arr_delay']). np.linspace генерирует равномерно распределенные числа от минимального значения arr_delay до максимального arr_delay, разделяя их на 10 интервалов.

Вычисление наблюдаемых и ожидаемых частот:

observed_frequencies = np.histogram(df_within_15_minutes['arr_delay'], bins=intervals)[0]: Подсчитывает количество наблюдений в каждом интервале (observed_frequencies). Функция np.histogram возвращает массив с подсчетом наблюдений в каждом интервале.

expected_frequencies = norm.cdf(intervals, loc=m, scale=std): Вычисляет кумулятивную функцию распределения (CDF) нормального распределения с параметрами m (среднее) и std (стандартное отклонение) для границ интервалов.

expected_frequencies = np.diff(expected_frequencies) * len(df_within_15_minutes): Преобразует CDF в ожидаемые частоты в каждом интервале, умножая на общее количество наблюдений (len(df_within_15_minutes)).

Вычисление статистики хи-квадрат:

chi2_statistic = np.sum((observed_frequencies - expected_frequencies)**2 / expected_frequencies): Вычисляет статистику хи-квадрат как сумму квадратов разностей между наблюдаемыми (observed_frequencies) и ожидаемыми (expected_frequencies) частотами, деленных на ожидаемые частоты.

Вычисление p-значения:

p_value = 1 - chi2.cdf(chi2_statistic, len(intervals) - 1): Вычисляет p-значение как вероятность получить такое же или более экстремальное значение статистики хи-квадрат, если данные действительно следуют нормальному распределению.

Проверка гипотезы о нормальном распределении:

al = 0.05: Устанавливает уровень значимости.if p_value < al: ... else: ...: Проверяет гипотезу о нормальном распределении на уровне значимости 0.05. Если p-значение меньше уровня значимости, то гипотеза отвергается, что означает, что данные вероятно не следуют нормальному распределению. В противном случае гипотеза не отвергается.

Предположение о распределении:

Согласно центральной предельной теореме, среднее значение большого количества независимых и одинаково распределенных случайных величин приближается к нормальному распределению, даже если исходное распределение не является нормальным.

В нашем случае, хотя отдельные задержки прилета могут иметь ненормальное распределение, средние значения множества таких задержек могут следовать нормальному распределению.

Параметры нормального распределения:
- средняя задержка μ ~ -8,25
- стандартное отклонение σ ~ 17,91

Вывод:

Анализ данных о задержках прилета показал, что распределение задержек в данной выборке можно приближенно описать нормальным распределением.

Среднее значение и стандартное отклонение определяют типичные отклонения от расписания.

Сравнение гистограммы с графиком плотности нормального распределения подтверждает соответствие данных этой модели.Также это подтверждает тест хи-квадрат.

Гистограмма демонстрирует, что вероятность возникновения задержек прилета уменьшается с увеличением их продолжительности как в сторону положительных, так и отрицательных значений.Это означает, что большинство рейсов прибывает с небольшими задержками, а вероятность значительных отклонений от расписания снижается.