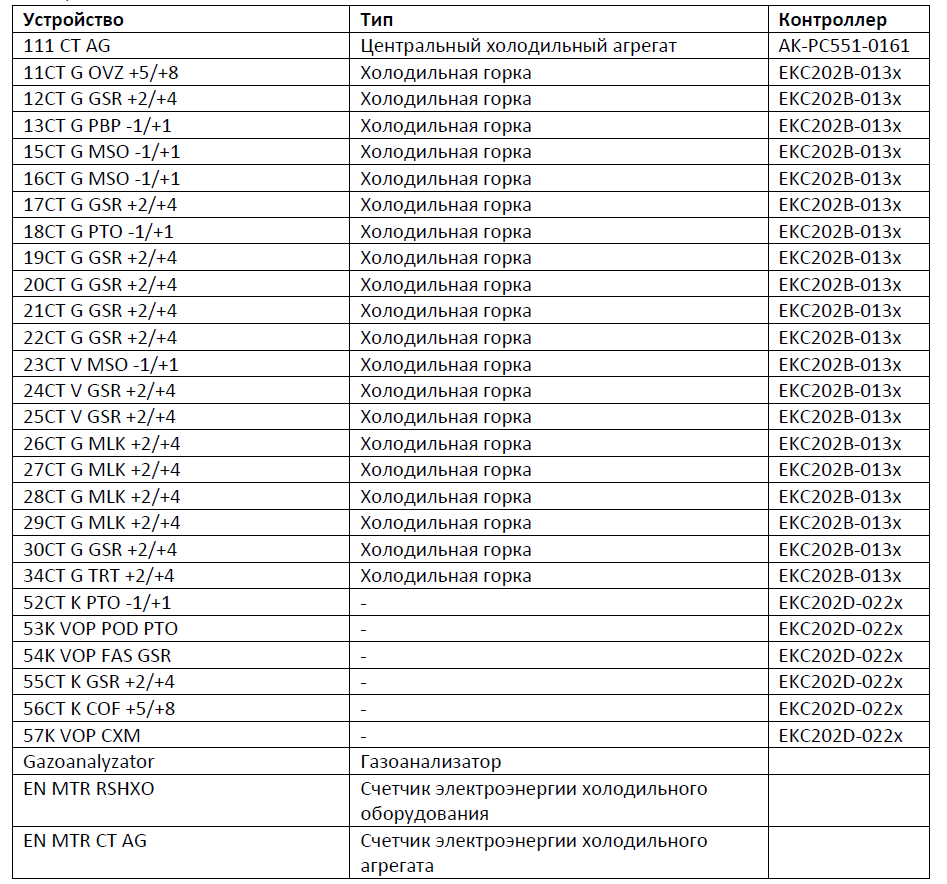
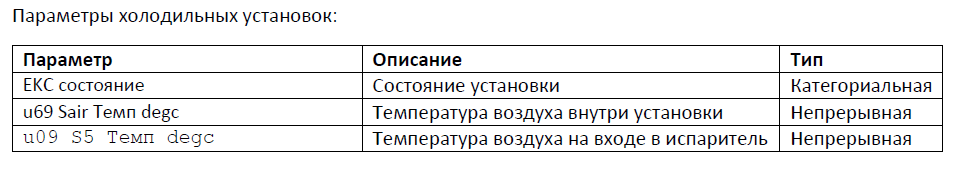
# Анализ данных. Домашнее задание

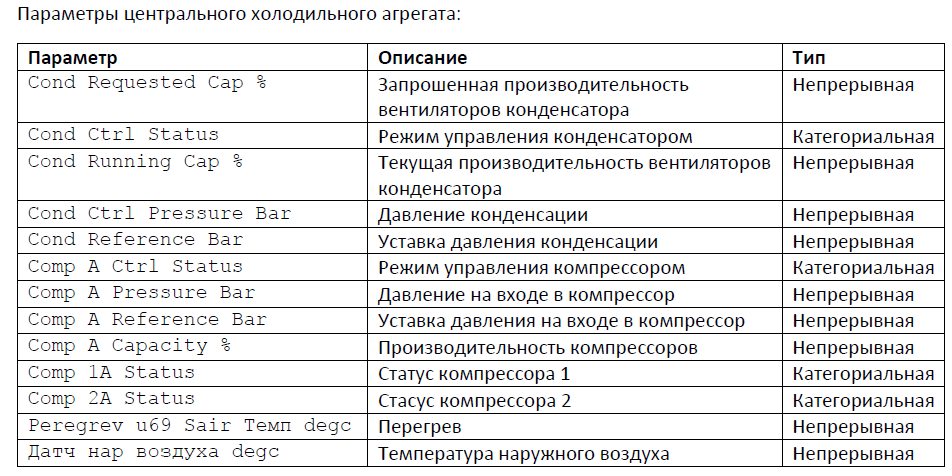
Задача состоит в работе с реальными данными о работе системы кондиционирования одного из распределительных центров крупной торговой сети.

## Описание исходных данных

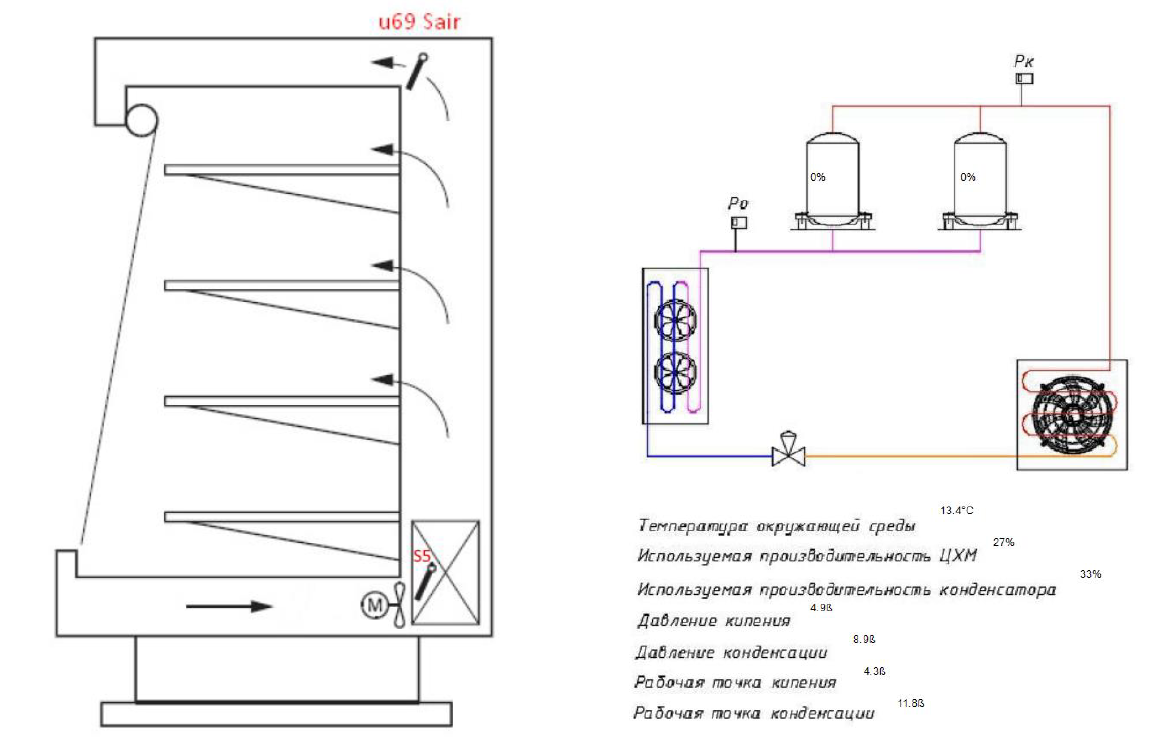
Система кондиционирования состоит множества практически одинаковых устройств, каждое из которых имеет определенный набор параметров.







В холодильных установках хранятся продукты, центральный холодильный агрегат обеспечивает установки хладагентом. Внешний вид холодильной установки представлен ниже. Это там, где лежит колбаса и йогурты с ценниками.



## Постановка задачи

Все визуализации должны быть подписаны, минимум заголовок и легенда (если требуется).

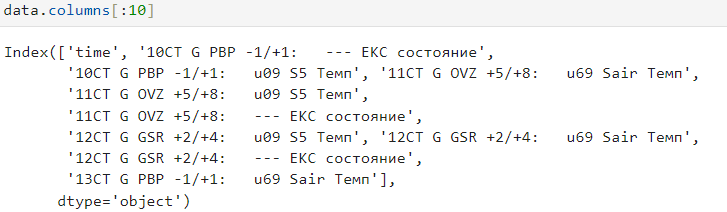
1. Загрузите исходный датасет

pd.read\_csv()

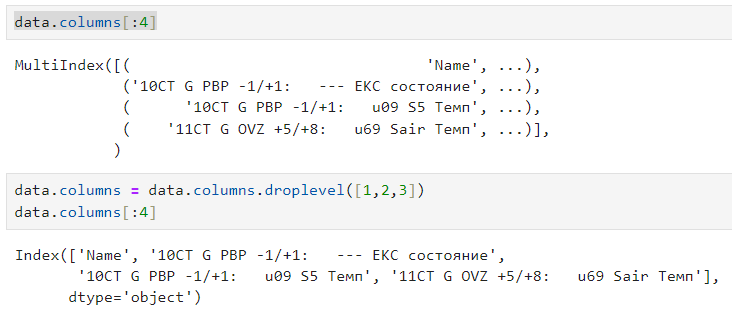
1. Сделайте необходимые преобразования при загрузке датасета, чтобы данные выглядели корректно.

<https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html>, корректно обработайте разделители, NA значения, кодировку, исключите ненужные строки, определите заголовок, при необходимости игнорируйте ошибки в кодировке или подправьте исходный файл.

1. Очистите заголовки колонок от ненужной информации, оставьте только имя устройства и название параметра:



Если при загрузке датасета у вас получились многоуровневые заголовки, можно удалить незначащие уровни методом df.columns.droplevel():



Для переименования индексов можно использовать метод df.rename(columns={<исходное имя>: <новое имя>})

1. Выведите в консоль информацию о датасете: его размер, название колонок, их типы данных

df.info()

1. Преобразуйте колонку со временем в формат datetime, сделайте ее индексом и отсортируйте (даже если датасет уже отсортирован)

pd.to\_datetime(), df.index = df[<колонка>], df.sort\_index()

1. Посчитайте количество пропусков в каждой колонке и визуализируйте их

df.isna().sum(), для визуализации можно использовать модуль missingno и метод matrix(df)

1. Как оптимизировать типы данных в колонках? Насколько меньше станет размер датафрейма после оптимизации типов?

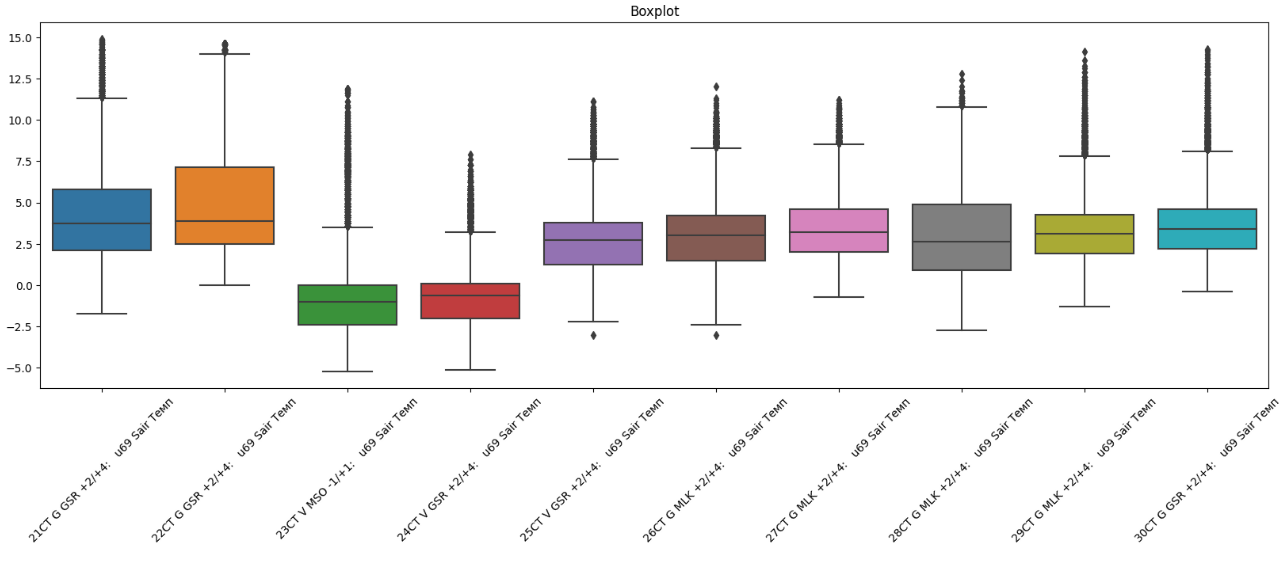
Приведите к типу int колонки с целыми значениями, используйте df.astype().

Приведите к типу category колонки с категориальными признаками (признаки, которые принимают ограниченное количество значений, например до 10).

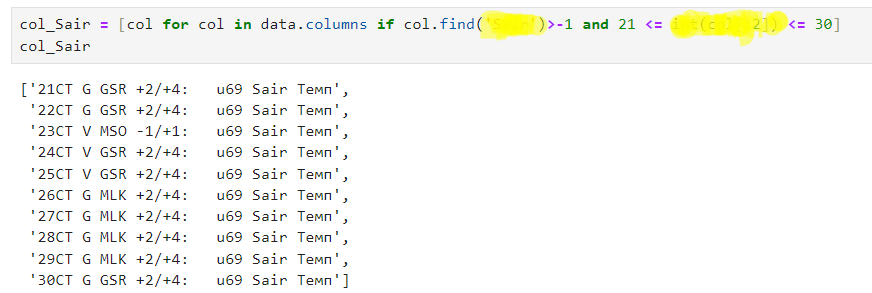
Сравните замеры получившихся датасетов, оригинальный датасет не перезаписывайте.

1. Дайте табличное и графическое статистическое описание признакам, содержащим параметр «Sair» для устройств 21CT, 22CT,…, 30CT.

df.describe(), sns.boxplot().



Можно создать отдельный список колонок, которые вы будете анализировать, например «col\_Sair». Для его создания можно использования list comprehension

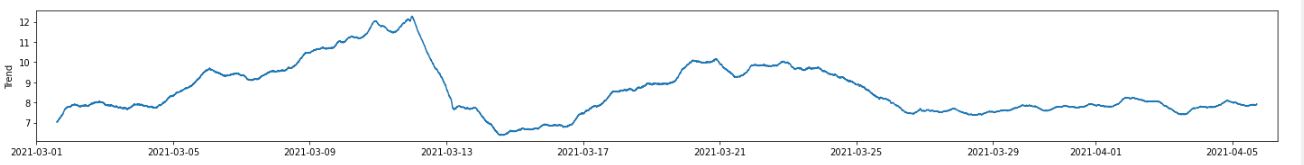


1. Ресемплируйте датасет по медианному значению за 4 минуты, отобразите на линейном графике значение признаков col\_Sair. График должен быть читабельным, иметь заголовок и легенду, можете выбрать другое значение ресэмплирования, чтобы график лучше читался.

Используйте любые модули (df.plot(), sns, plt, plotly). Ресемплированный датасет сохранять не нужно, просто отобразить.

1. Постройте для 2-3 признаков из col\_Sair сглаженный график поведения признаков во времени, чтобы был виден тренд

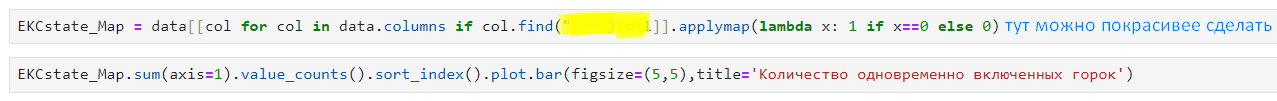
df.rolling()



1. Возьмите любой из параметров Sair и визуализируйте на единой координатной плоскости его поведение в зависимости от предобработки: оригинальный, ресемплированный и сглаженный.

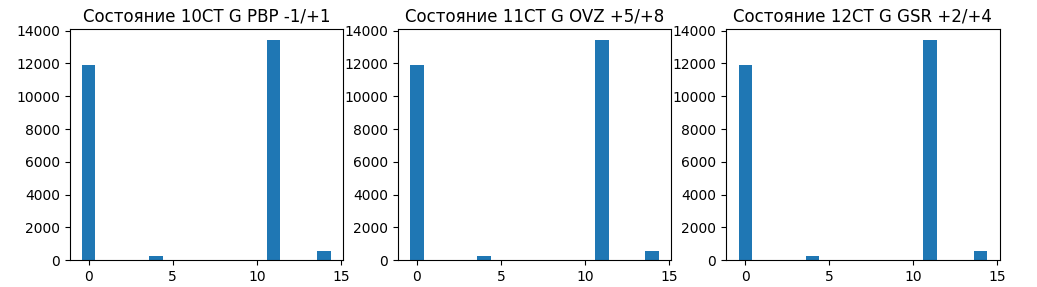
Параметры ресемплирования и сглаживания выберите любые. Подберите временной интервал, чтобы разница была видна.

1. Отобразите гистограмму по количеству одновременно включенных устройств. Признак состояния представлены в виде «<устройство>: --- EKC состояние», если признак принимает значение 0, то считаем устройство включенным.





1. Постройте для нескольких признаков состояния устройств гистограммы, которые покажут распределение этих устройств по состояниям.



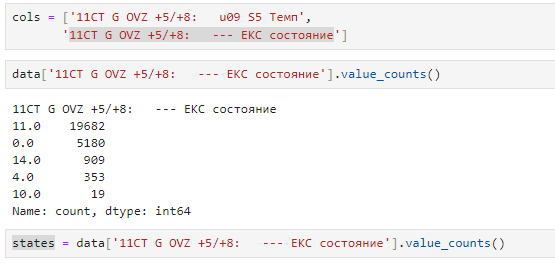
Используйте любой plt.bar или sns.barplot() и plt.subplots()

1. Постройте матрицу корреляции для признаков col\_Sair

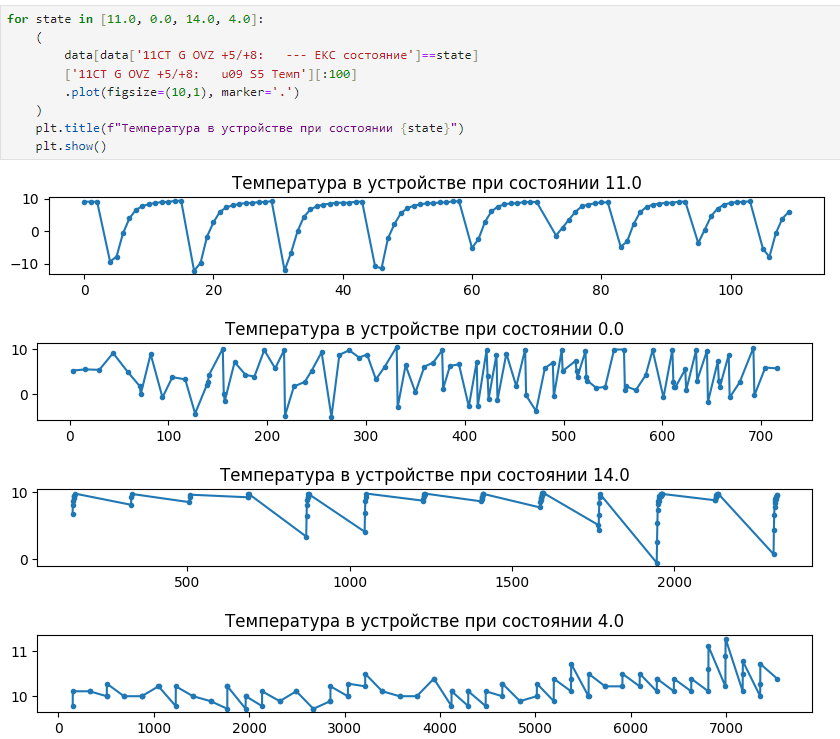
Получение таблицы с корреляцией: df.corr(), отрисовку удобно делать через sns.heatmap()

1. Творческое: сформулируйте гипотезу (вопрос) и проверьте ее (ответьте на него), подтвердив выводы визуализацией или статистиками.

Например, можно задаться вопросом, как ведет себя температура **u09 S5** в устройстве при разных состояниях устройства.



Только на основании графиков сделать выводы сложно:



Но статистика говорит о том, что температура в режимах, отличных от 0.0 или растет или меняется незначительно.

