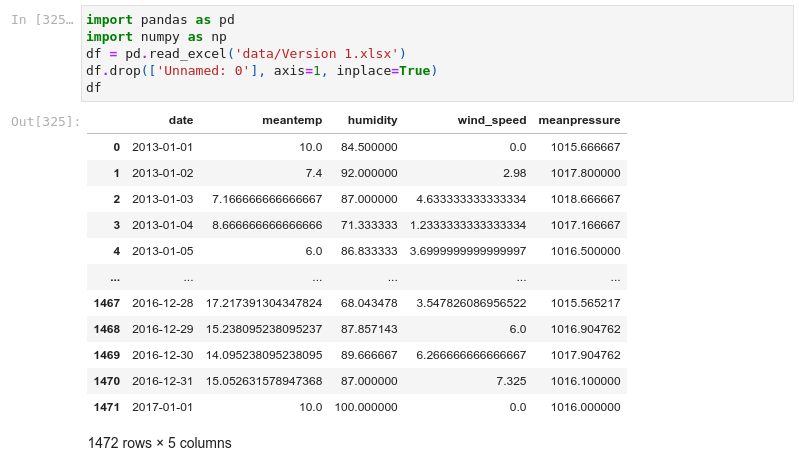
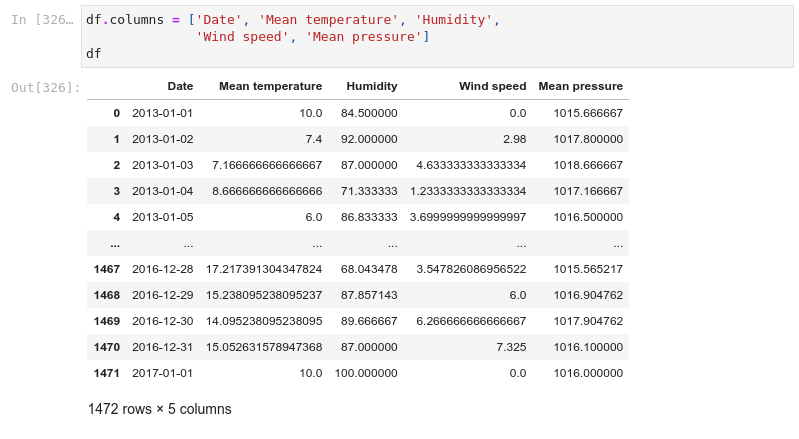
# Виконання

## Зчитування файлу та зміна назви стовпців

Для початку імпортуємо модуль pandas та застосуємо функцію pandas.read\_excel для зчитування .xlsx-файлів. Видалимо колонку "Unnamed: 0", оскільки індекси за замовчуванням створюються.

  
  
Рисунок 3.1 - Зчитування .xlsx-файлу в датафрейм

Змінимо назви стовпців.

  
  
Рисунок 3.2 - Зміна назв стовпців

## Знаходження проблем з даними та виконання попередньої обробки даних для усунення цих проблем

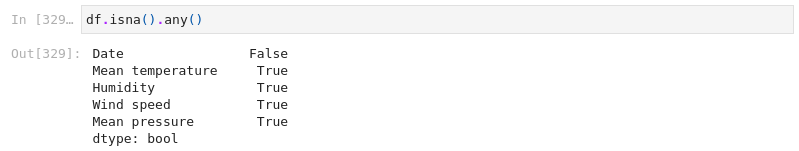
Перевіримо типи даних в стовпцях.

  
  
Рисунок 3.3 - Типи даних

Приведемо всі типи в датафреймі в тип object.

  
  
Рисунок 3.4 - Приведення до типу object

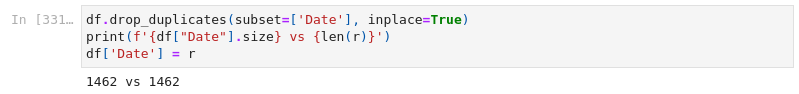
Перевіримо, чи є пропущенні значення у кожному стовпці. для цього застосуємо метод isna, який повертає для кожного стовпця вектор булевих значень, що позначають, чи елемент пустий. Далі застосуємо метод any, для того щоб дізнатися, чи є хоча б одне пропущене значення.

  
  
Рисунок 3.5 - Інформація про присутність пустих значень у стовпцях

Розберемося з колонкою дат. При роботі я побачив цікаву деталь, що застосувавши pandas.date\_range я отримував 1462 рядки, а в колонці дат рядків 1472.

  
  
Рисунок 3.6 - Підозра на наявність дублікатів в датах

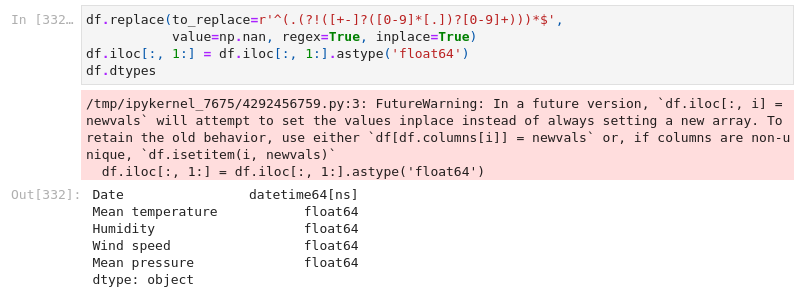
Приберемо дублікати та перевіремо ще раз.

  
  
Рисунок 3.7 - Прибирання дублікатів

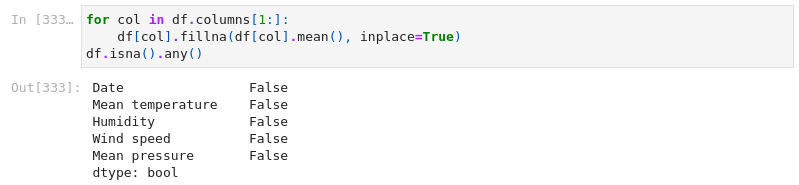
У стовпцях з числами можуть бути символи, що не відповідають загальному виду числа. Приберемо їх на пусті клітинки за допомогою регулярних виразів.

Пояснимо регулярний вираз. Його основу складає дана частина: "[+-]?([0-9]\*[.])?[0-9]+". Вона перевіряє рядок на те, що це або int, або float. Далі треба застосувати інверсію умови, тобто: (.(?!(умова))).

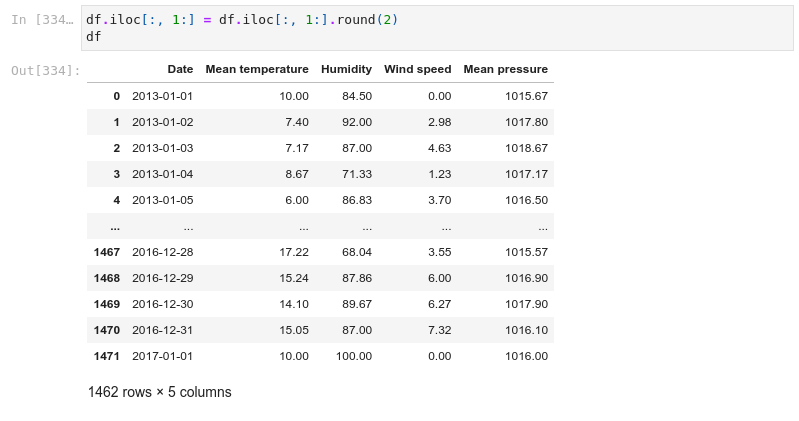
У кінці приведемо усі стовпчики до типу float64.

  
  
Рисунок 3.8 - Заміна невідповідних клітинок у стовпчиках на -1, приведення типів до float64

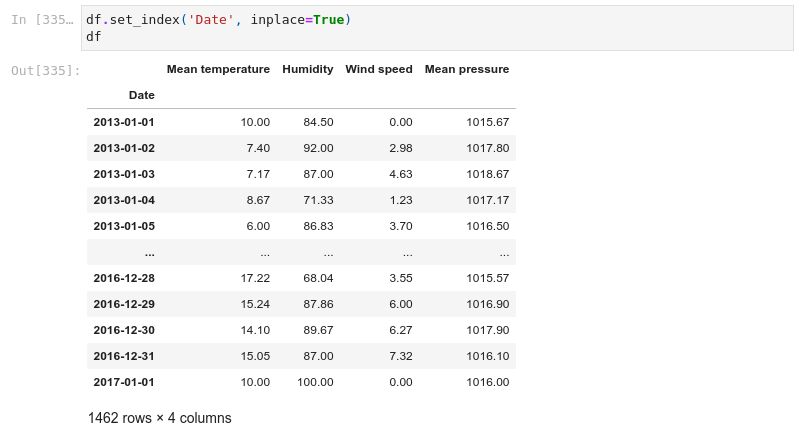
Заповнимо пусті значення на середнє значення для кожного стовпчика відповідно. Застосуємо метод fillna.

  
  
Рисунок 3.9 - Заміна пустих значень на середнє для кожного стовпчика

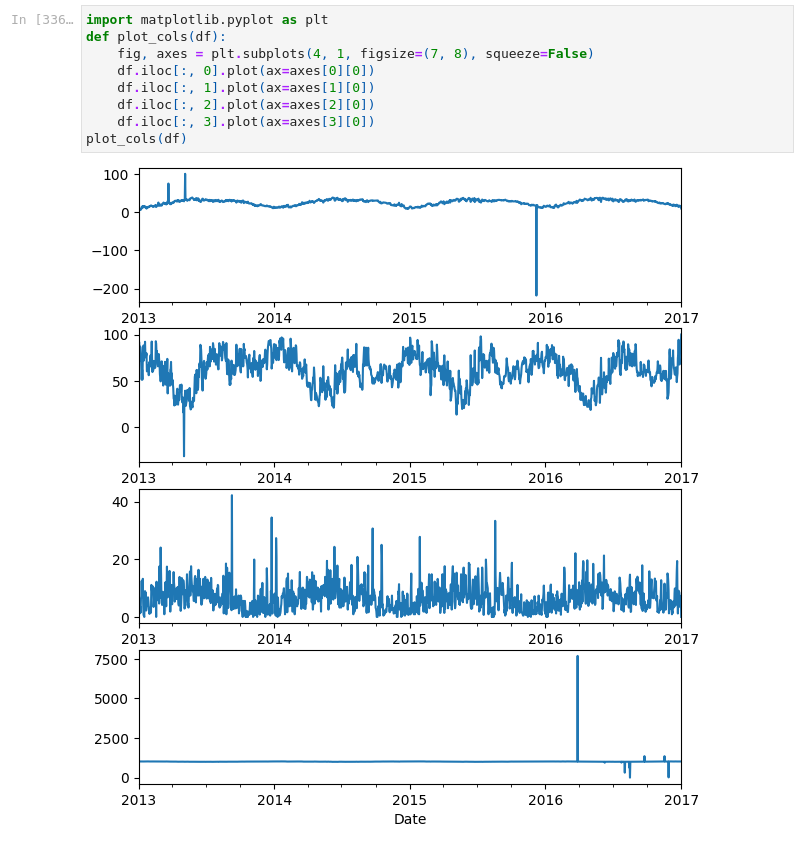
Округлимо до сотих значення. Застосуємо метод round.

  
  
Рисунок 3.10 - Округлення чисел у датафреймі

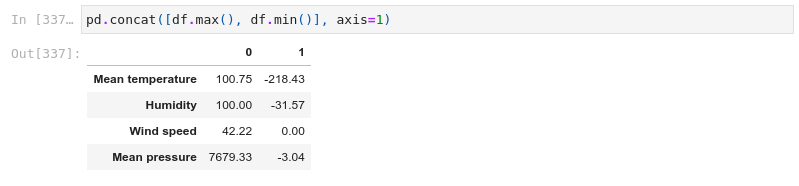
Зробимо колонку дат індексом за допомогою методу set\_index.

  
  
Рисунок 3.11 - Приведення колонки дат до індексу

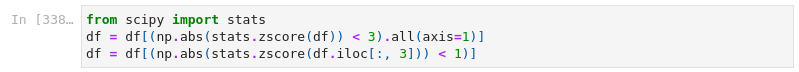
Перевіримо, чи є аномальні дані за допомогою графіків. Побачимо, що вони є і треба їх прибрати.

  
  
Рисунок 3.12 - Графіки стовпчиків

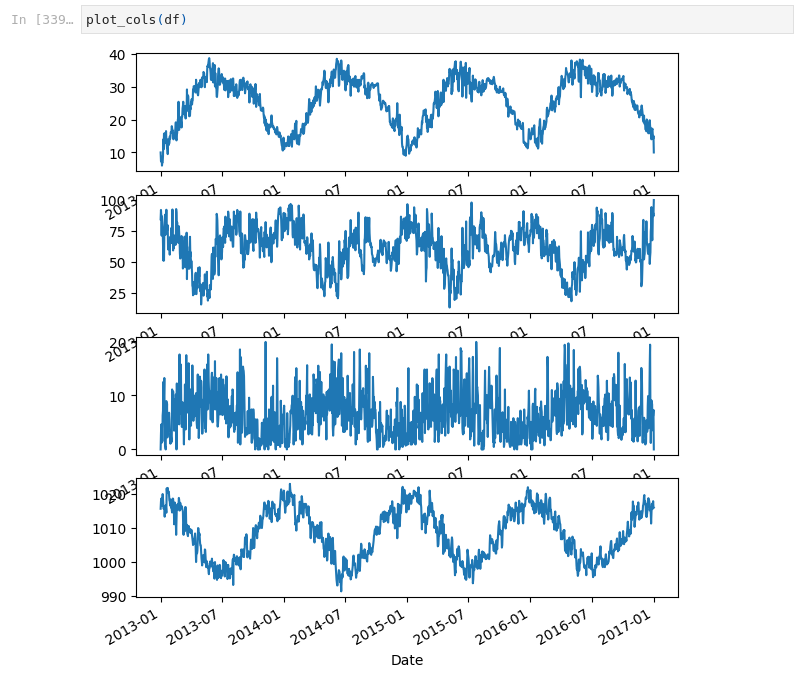
Виведемо для кожної колонки максимальне та мінімальне значення.

  
  
Рисунок 3.13 - Мінімальні та максимальні значення

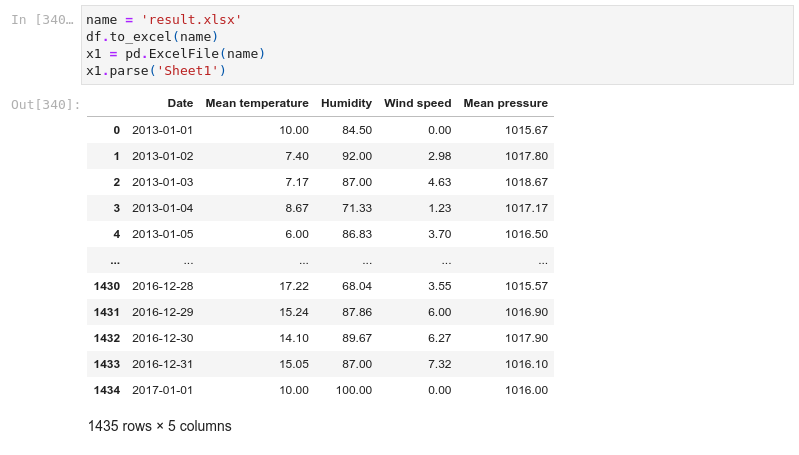
За допомогою функції zscore обрахуємо значенння відносно середнього та стандартного відхилення для кожної колонки. Оскільки напрямок значення не має, тов візьмемо модуль та потім порівняємо з певним віхиленням.

  
  
Рисунок 3.14 - Видалення аномалій

Побудуємо для наочності графіки. Як бачимо кількість аномалій зменшилася.

  
  
Рисунок 3.15 - Графіки стовпчиків

Експортуємо датафрейм до .xlsx-файлу за допомогою методу to\_excel.

  
  
Рисунок 3.16 - Збереження датафрейму до .xlsx-файлу