```
!pip install pandas
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/d
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/pyt
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/di
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-p
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, ca
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
from tqdm import tqdm
import pandas as pd
import numpy as np
import os
import time
import os, time, warnings
from tqdm import tqdm
from multiprocessing import Pool, cpu_count
warnings.filterwarnings("ignore", category=RuntimeWarning)
```

```
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.float_format', '{:.2f}'.format)
```

```
df1=pd.read_csv('/content/drive/MyDrive/FE/FE462.csv')
```

```
<ipython-input-21-efb793941f81>:1: DtypeWarning: Columns (14,15,16,17,19,20
  df1=pd.read_csv('/content/drive/MyDrive/FE/FE462.csv')
```

```
df=pd.read_csv('/content/drive/MyDrive/FE/@20.csv')
```

```
total_cells = df.size
missing_cells = df.isnull().sum().sum()
missing_ratio = (missing_cells / total_cells) * 100

print(f" Total cells        : {total_cells:,}")
print(f" Missing cells      : {missing_cells:,}")
print(f" Missing percentage: {missing_ratio:.2f}%")
```

Show hidden output

```
missing_report = df.isnull().sum().to_frame(name='Missing Count')
missing_report['Total Rows'] = len(df)
missing_report['Missing %'] = (missing_report['Missing Count'] / missing_report

missing_report = missing_report[missing_report['Missing Count'] > 0]
missing_report = missing_report.sort_values(by='Missing %', ascending=False)


pd.set_option('display.float_format', lambda x: '%.2f' % x)
missing_report
```

	Missing Count	Total Rows	Missing %
Urobilinogen	900000	900000	100.00
Amorphous Elements	900000	900000	100.00
Ketones	900000	900000	100.00
Blood and Haemoglobin	900000	900000	100.00
W.B.Cs / HPF	900000	900000	100.00
Concentration	899996	900000	100.00
Consistancy	899995	900000	100.00
R.B.Cs / HPF	899995	900000	100.00
24 Hour Urine Volume (263)	899993	900000	100.00
Albumin in Urine (263)	899992	900000	100.00
Cystatin C	899992	900000	100.00
W. B. Cs / HPF	899947	900000	99.99
Platelet Count	899908	900000	99.99
Leucocyte esterase	899794	900000	99.98

Epithelial Cells / HPF	899794	900000	99.98
Nitrite	899794	900000	99.98
Colour(Urine Physical Examination)	899794	900000	99.98
Aspect(Urine Physical Examination) Ordinal Encoding	899794	900000	99.98
Protein(Urine Physical Examination)	899794	900000	99.98
Specific Gravity	899794	900000	99.98
Bilirubin_Numeric	899794	900000	99.98
Amorphous_Numeric	899794	900000	99.98
pH(Urine Physical Examination)	899794	900000	99.98
Glucose(Urine Physical Examination)	899794	900000	99.98
Red cell count	899790	900000	99.98
RDW	899787	900000	99.98
MCH	899784	900000	99.98
MCV	899784	900000	99.98
Total Leucocytic Count	899784	900000	99.98
Lymphocytes absolute count	899784	900000	99.98
Basophils absolute count	899784	900000	99.98
Eosinophils absolute count	899784	900000	99.98
MCHC	899784	900000	99.98
Monocytes absolute count	899784	900000	99.98
Neutrophils absolute count	899784	900000	99.98
Hematocrit	899781	900000	99.98
Hemoglobin	899781	900000	99.98
T. Cholesterol/HDL_Numeric	899758	900000	99.97
T. Cholesterol/HDL	899758	900000	99.97
LDL / HDL	899757	900000	99.97
Rheumatoid Factor (quantitative)	899498	900000	99.94
Lead in blood	899290	900000	99.92
Transferrin	898281	900000	99.81

Titre on Hep2 cells	897668	900000	99.74
Microalbuminuria (24 h urine)	890469	900000	98.94
C-Reactive Protein (CRP) quantitative	888923	900000	98.77
Magnesium (Mg) in Serum	882887	900000	98.10
Prostatic Specific Antigen (PSA) Total	780763	900000	86.75
Testosterone (Total)	776737	900000	86.30
Erythrocyte Sedimentation Rate(ESR)	755744	900000	83.97
Ferritin In Serum	749229	900000	83.25
Iron (Fe) in Serum	746343	900000	82.93
Urea in Serum	687717	900000	76.41
BUN/Creatinine Ratio	624730	900000	69.41
Globulin in Serum	624329	900000	69.37
Total Protein in Serum	618897	900000	68.77
Alkaline Phosphatase	611452	900000	67.94
Chloride in Serum	607973	900000	67.55
Diastolic Pressure	607167	900000	67.46
Systolic Pressure	607167	900000	67.46
Estimated Glomerular Filtration Rate(eGFR)	601105	900000	66.79
Sodium (Na) in Serum	597269	900000	66.36
HDL Cholesterol	594795	900000	66.09
Potassium (K) in Serum	593976	900000	66.00
CRP H.S	586365	900000	65.15
Bilirubin (Direct)	581936	900000	64.66
Bilirubin (Total)	581615	900000	64.62
Triglycerides (TG) in Serum	556776	900000	61.86
Cholesterol	551734	900000	61.30
Mean of blood glucose	549685	900000	61.08
Hb A1c %	549093	900000	61.01
Blood Urea Nitrogen (BUN)	538755	900000	59.86
Free T4	510513	900000	56.72

Free T4	510513	900000	56.72
Albumin in Serum	468039	900000	52.00
Glucose in Plasma (Fasting)	376147	900000	41.79
Calcium in Serum (Total)	361075	900000	40.12
Thyroid Stimulating Hormone (TSH)	355299	900000	39.48
Uric Acid in Serum	351487	900000	39.05
Vitamin D (25 OH-Vit D -Total)	331532	900000	36.84
Aspartate Aminotransferase (AST)	325714	900000	36.19
Alanine Aminotransferase (ALT)	320871	900000	35.65
Creatinine in Serum	306509	900000	34.06

```
df.dtypes
```

	0
Unnamed: 0	int64
RESEARCH_ID	object
SAMPLE_ID	object
COLLECTYEAR	int64
REGN_DATE	object
AGE_YEARS	float64
AGE_DAYS	int64
AGE_MONTHS	int64
HEIGHT	int64
WEIGHT	float64
BMI	float64
Thyroid Stimulating Hormone (TSH)	float64
Uric Acid in Serum	float64
Alanine Aminotransferase (ALT)	float64
Ferritin In Serum	float64
Blood Urea Nitrogen (BUN)	float64
Lymphocytes absolute count	float64

R. B. Cs / HPFs	float64
Aspect(Urine Physical Examination) Ordinal Encoding	float64
Eosinophils absolute count	float64
Vitamin D (25 OH-Vit D -Total)	float64
C-Reactive Protein (CRP) quantitative	float64
Transferrin	float64
Red cell count	float64
Basophils absolute count	float64
Crystals(Urine Microscopic Examination :)	float64
Protein(Urine Physical Examination)	float64
Colour(Urine Physical Examination)	float64
Nitrite	float64
LDL Cholesterol	int64
LDL / HDL	float64
24 Hour Urine Volume (263)	float64
Hemoglobin	float64
Total Leucocytic Count	float64
Hematocrit	float64
MCV	float64
Glucose(Urine Physical Examination)	float64
Urea in Serum	float64
Prostatic Specific Antigen (PSA) Total	float64
Testosterone (Total)	float64
Alkaline Phosphatase	float64
Total Protein in Serum	float64
Estimated Glomerular Filtration Rate(eGFR)	float64
Anti CCP Abs	int64
BUN/Creatinine Ratio	float64
Ketones	float64
MCHC	float64

MCHC	float64
pH(Urine Physical Examination)	float64
Amorphous Elements	float64
Blood and Haemoglobin	float64
Epithelial Cells / HPF	float64
Casts(Urine Microscopic Examination :)	int64
Chloride in Serum	float64
Cholesterol	float64
T. Cholesterol/HDL	float64
Urobilinogen	float64
R.B.Cs / HPF	float64
Erythrocyte Sedimentation Rate(ESR)	float64
Glucose in Plasma (Fasting)	float64
Hb A1c %	float64
Mean of blood glucose	float64
Microalbuminuria (24 h urine)	float64
Bilirubin (Total)	float64
Florescence Pattern	int64
Lead in blood	float64
Monocytes absolute count	float64
Consistancy	float64
Neutrophils absolute count	float64
Specific Gravity	float64
W. B. Cs / HPF	float64
Aspartate Aminotransferase (AST)	float64
Calcium in Serum (Total)	float64
Free T4	float64
Potassium (K) in Serum	float64
Albumin in Serum	float64
Iron (Fe) in Serum	float64

CRP H.S	float64
Triglycerides (TG) in Serum	float64
Rheumatoid Factor (quantitative)	float64
Platelet Count	float64
Albumin in Urine (263)	float64
MCH	float64
RDW	float64
W.B.Cs / HPF	float64
Leucocyte esterase	float64
Concentration	float64
Creatinine in Serum	float64
Sodium (Na) in Serum	float64
Bilirubin (Direct)	float64
Magnesium (Mg) in Serum	float64
Titre on Hep2 cells	float64
HDL Cholesterol	float64
Globulin in Serum	float64
Cystatin C	float64
RESEARCH_ID_int	int64
GENDER_BINARY	int64
CITY_NAME_ENCODED	int64
BDL	int64
Florescence Pattern	int64
Systolic Pressure	float64
Diastolic Pressure	float64
Amorphous_Numeric	float64
Bilirubin_Numeric	float64
T. Cholesterol/HDL_Numeric	float64

dtype: object

```
object_columns_df = df.select_dtypes(include=['object'])
object_columns_df.dtypes
```

	0
RESEARCH_ID	object
SAMPLE_ID	object
REGN_DATE	object

dtype: object

```
df.duplicated().sum()
```

```
np.int64(0)
```

```
df.shape
```

```
(900000, 104)
```

```
df.columns
```

```
Index(['Unnamed: 0', 'RESEARCH_ID', 'SAMPLE_ID', 'COLLECTYEAR',
'REGN_DATE',
       'AGE_YEARS', 'AGE_DAYS', 'AGE_MONTHS', 'HEIGHT', 'WEIGHT',
       ...
       'RESEARCH_ID_int', 'GENDER_BINARY', 'CITY_NAME_ENCODED', 'BDL',
       'Florescence Pattern ', 'Systolic Pressure', 'Diastolic Pressure',
       'Amorphous_Numeric', 'Bilirubin_Numeric', 'T.
Cholesterol/HDL_Numeric'],
      dtype='object', length=104)
```

```
columns_zero_is_missing = [
    "LDL Cholesterol",
    "HDL Cholesterol",
    "LDL / HDL",
    "Triglycerides (TG) in Serum",
    "Cholesterol",
    "T. Cholesterol/HDL",
    "T. Cholesterol/HDL_Numeric",
    "Bilirubin (Total)",
    "Bilirubin (Direct)",
    "Iron (Fe) in Serum",
    "Ferritin In Serum",
    "Uric Acid in Serum",
```

```
    "Albumin in Serum",
    "Globulin in Serum",
    "Total Protein in Serum",
    "Calcium in Serum (Total)",
    "Magnesium (Mg) in Serum",
    "Sodium (Na) in Serum",
    "Potassium (K) in Serum",
    "Creatinine in Serum",
    "Urea in Serum",
    "Blood Urea Nitrogen (BUN)",
    "Estimated Glomerular Filtration Rate(eGFR)",
    " BUN/Creatinine Ratio",
    "Cystatin C",
    "Thyroid Stimulating Hormone (TSH)",
    "Free T4",
    "Prostatic Specific Antigen (PSA) Total",
    "Testosterone (Total)",
    "Vitamin D (25 OH-Vit D -Total)",
    "C-Reactive Protein (CRP) quantitative",
    "CRP H.S",
    "Rheumatoid Factor (quantitative)",
    "Anti CCP Abs",
    "Transferrin",
    "Aspartate Aminotransferase (AST)",
    "Alanine Aminotransferase (ALT)",
    "Alkaline Phosphatase",
    "Hemoglobin",
    "Total Leucocytic Count",
    "Platelet Count",
    "MCV",
    "MCH",
    "MCHC",
    "RDW",
    "Mean of blood glucose",
    "Hb A1c %",
    "Glucose in Plasma (Fasting)",
    "Microalbuminuria (24 h urine)",
    "Albumin in Urine (263)",
    "Lead in blood",
    "Casts(Urine Microscopic Examination :)",
]
```

```
for col in columns_zero_is_missing:
    if col in df.columns:
       df[col] = df[col].replace(0, np.nan)
```

```
df.isnull().sum().sum()
```

```
df.duplicated().sum().sum()
```

```
df_sorted = df.sort_values(by=["RESEARCH_ID", "SAMPLE_ID", "REGN_DATE"])
df_fast = df_sorted.copy()

columns_to_fill = df_fast.columns[df_fast.isnull().any()].tolist()

total_filled = 0
start_time = time.time()


for col in tqdm(columns_to_fill, desc="Filling NaNs", unit="column"):
    before = df_fast[col].isna().sum()
    df_fast[col] = df_fast.groupby(["RESEARCH_ID", "SAMPLE_ID"])[col].ffill()
    df_fast[col] = df_fast.groupby(["RESEARCH_ID", "SAMPLE_ID"])[col].bfill()
   after = df_fast[col].isna().sum()
    total_filled += (before - after)


end_time = time.time()
print(f"\n✅ Total filled values: {total_filled:,}")
print(f"⏱️ Time elapsed: {end_time - start_time:.2f} seconds.")


output_folder = "/content/drive/MyDrive/XXXX/knowledge_project"
os.makedirs(output_folder, exist_ok=True)


output_path = os.path.join(output_folder, "cleaned_dataset.csv")
df_fast.to_csv(output_path, index=False)

print(f"\n File saved successfully at: {output_path}")
```

```
df_fast=pd.read_csv('/content/drive/MyDrive/XXXX/knowledge_project/cleaned_data
```

```
df_fast.duplicated().sum().sum()
```

```
np.int64(0)
```

```
df_fast.isnull().sum().sum()
```

```
np.int64(61344978)
```

```
df_fast["REGN_DATE"] = pd.to_datetime(df_fast["REGN_DATE"], errors='coerce')
```

```
df_fast["REGN_DATE"].dtype
```

```
dtype('<M8[ns]')
```

```
df_fast = df_fast.sort_values(by="RESEARCH_ID")
```

```
df_marg=df_fast
```

```
df_marg.isnull().sum().sum()
```

```
np.int64(61344978)
```

```
def smart_impute_group_numpy_rid(group_tuple):
    _, group = group_tuple
    group = group.copy()
    dates = group["REGN_DATE"].values.astype("datetime64[ns]")
    excluded_cols = ["RESEARCH_ID", "SAMPLE_ID", "REGN_DATE"]
    columns_to_fill = [col for col in group.columns if col not in excluded_cols

    for col in columns_to_fill:
        values = group[col].values
        mask_nan = np.isnan(values)
        if not np.any(mask_nan): continue
        known_idx = np.where(~mask_nan)[0]
        known_dates = dates[known_idx]
        known_values = values[known_idx]

        for idx in np.where(mask_nan)[0]:
            current_date = dates[idx]
            same_date_mask = known_dates == current_date
            if np.any(same_date_mask):
                values[idx] = known_values[same_date_mask][0]
                continue
            if known_dates.size > 0:
                time_deltas = np.abs(known_dates - current_date)
                values[idx] = known_values[np.argmin(time_deltas)]
                continue
            if known_values.size > 0:
```

```
                mean_val = np.nanmean(known_values)
                if not np.isnan(mean_val):
                    values[idx] = mean_val
        group[col] = values
    return group


all_groups = list(df_marg.groupby("RESEARCH_ID"))
batch_size = 50000
total_batches = (len(all_groups) + batch_size - 1) // batch_size
output_folder = "/content/drive/MyDrive/XXXX/rid_batches"
os.makedirs(output_folder, exist_ok=True)

start_time = time.time()


for i in range(total_batches):
    batch_path = f"{output_folder}/batch_rid_{i+1}.csv"
    if os.path.exists(batch_path):
        print(f"✅ Skipping batch {i+1} (already processed).")
        continue


    batch_groups = all_groups[i * batch_size : (i+1) * batch_size]
    results = []

    with Pool(cpu_count() - 1) as pool:
        for result in tqdm(pool.imap(smart_impute_group_numpy_rid, batch_groups)
                           total=len(batch_groups),
                           desc=f"Batch RID {i+1}",
                           unit="group",
                           dynamic_ncols=True):
            results.append(result)

    df_batch = pd.concat(results, ignore_index=True)
    df_batch.to_csv(batch_path, index=False)
    print(f" Saved: {batch_path}")

print("\n Merging all RID batches...")
merged_df = pd.concat(
    [pd.read_csv(f"{output_folder}/batch_rid_{i+1}.csv") for i in range(total_ba
    ignore_index=True
)


final_path = "/content/drive/MyDrive/XXXX/filled_final_by_rid_batches.csv"
merged_df.to_csv(final_path, index=False)

end_time = time.time()
```

```
end_time = time.time()
minutes, seconds = divmod(int(end_time - start_time), 60)


print(f" RID Imputation Completed in {total_batches} Batches")
print(f" Final file saved at: {final_path}")
print(f" Remaining missing values: {merged_df.isnull().sum().sum():,}")
print(f"⏱ Time taken: {minutes} min {seconds} sec")
```

Show hidden output

```
df4=pd.read_csv('/content/drive/MyDrive/XXXX/filled_final_by_rid_batches.csv')
```

```
df4.isnull().sum().sum()
```

```
np.int64(55841104)
```

```
df4.shape
```

```
(900000, 104)
```

```
df_marg2=pd.read_csv("/content/drive/MyDrive/XXXX/filled_final_by_rid_batches.c
```

```
df_marg2.isnull().sum().sum()
```

```
np.int64(55841104)
```

```
df_marg2.duplicated().sum().sum()
```

```
np.int64(0)
```

Start coding or generate with AI.