1_BasicDataManipulation2

August 27, 2024

[18]:
```
import pandas as pd;
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# This file includes basic data manipulation
# The objective is to determine the minimum, maximum, and average values, as␣
 ↪well as the pairwise correlation coefficients for all numerical variables␣
 ↪among individuals who have been diagnosed with chronic kidney disease (CKD)

file_path = 'kd.csv' # Fill here the correct filepath
df = pd.read_csv(file_path)

# Replace '?' value with <NA>
df.replace('?', pd.NA, inplace=True)

# Convert specific numeric columns to numeric data type
numeric_int_cols = ['age', 'bp', 'bgr', 'bu', 'sod', 'pcv', 'wbcc']
numeric_float_cols = ['sc', 'pot', 'hemo', 'rbcc']

# Convert values to numeric values (int or float)
df[numeric_int_cols] = df[numeric_int_cols].apply(pd.to_numeric,␣
 ↪errors='coerce')
df[numeric_float_cols] = df[numeric_float_cols].apply(pd.to_numeric,␣
 ↪errors='coerce')

# Fill missing values with the mean for numeric columns
df[numeric_int_cols] = df[numeric_int_cols].fillna(df[numeric_int_cols].mean())
df[numeric_float_cols] = df[numeric_float_cols].fillna(df[numeric_float_cols].
 ↪mean())

# Convert non-numeric columns to category
non_numeric_cols = df.columns.difference(numeric_int_cols + numeric_float_cols)
df[non_numeric_cols] = df[non_numeric_cols].astype('category')

# Filtering ckd patients
affected_patients_df = df[df['class'] == 'ckd']
```

```
# Printing basic statistics
res = affected_patients_df.describe()
print(res)
```

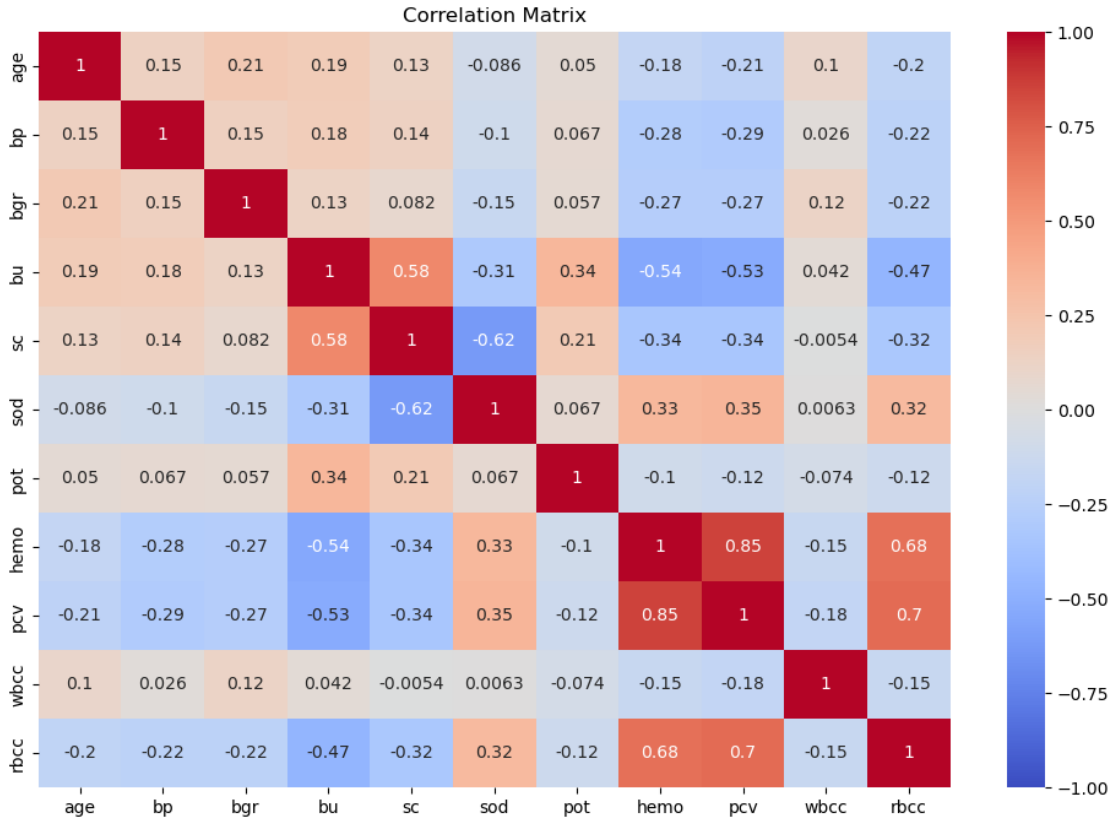
```
              age          bp         bgr          bu          sc         sod  \
count  248.000000  248.000000  248.000000  248.000000  248.000000  248.000000
mean    54.330109   79.575366  171.312047   71.857800    4.364998  135.088136
std     17.135049   14.946593   85.642513   57.257842    6.811876   10.329330
min      2.000000   50.000000   22.000000    1.500000    0.500000    4.500000
25%     47.750000   70.000000  110.750000   32.000000    1.500000  134.000000
50%     58.500000   80.000000  148.036517   55.000000    2.450000  137.528754
75%     65.000000   90.000000  210.750000   90.000000    4.325000  137.528754
max     90.000000  180.000000  490.000000  391.000000   76.000000  163.000000

              pot        hemo         pcv          wbcc        rbcc
count  248.000000  248.000000  248.000000    248.000000  248.000000
mean     4.797424   10.992297   34.541842   8811.718236    4.329091
std      3.544640    2.108740    6.712680   2806.307787    0.715413
min      2.500000    3.100000    9.000000   2200.000000    2.100000
25%      4.100000    9.800000   30.000000   7900.000000    3.900000
50%      4.627244   11.300000   36.000000   8406.122449    4.707435
75%      4.627244   12.526437   38.884498   9600.000000    4.707435
max     47.000000   16.100000   52.000000  26400.000000    8.000000
```

[20]:
```
# Filtering strings out because cannot handle them when creating matrix
isNum = df.select_dtypes(include=[np.number])

# Calculate pairwise correlation coefficients
correlationMatrix = isNum.corr()

# Visualize the correlation matrix with Seaborn
plt.figure(figsize=(12, 8))
sns.heatmap(correlationMatrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Matrix')
plt.show()
```


Correlation Matrix
age 1 0.15 0.21 0.19 0.13 -0.086 0.05 -0.18 -0.21 0.1 -0.2
bp 0.15 1 0.15 0.18 0.14 -0.1 0.067 -0.28 -0.29 0.026 -0.22
bgr 0.21 0.15 1 0.13 0.082 -0.15 0.057 -0.27 -0.27 0.12 -0.22
bu 0.19 0.18 0.13 1 0.58 -0.31 0.34 -0.54 -0.53 0.042 -0.47
sc 0.13 0.14 0.082 0.58 1 -0.62 0.21 -0.34 -0.34 -0.0054 -0.32
sod -0.086 -0.1 -0.15 -0.31 -0.62 1 0.067 0.33 0.35 0.0063 0.32
pot 0.05 0.067 0.057 0.34 0.21 0.067 1 -0.1 -0.12 -0.074 -0.12
hemo -0.18 -0.28 -0.27 -0.54 -0.34 0.33 -0.1 1 0.85 -0.15 0.68
pcv -0.21 -0.29 -0.27 -0.53 -0.34 0.35 -0.12 0.85 1 -0.18 0.7
wbcc 0.1 0.026 0.12 0.042 -0.0054 0.0063 -0.074 -0.15 -0.18 1 -0.15
rbcc -0.2 -0.22 -0.22 -0.47 -0.32 0.32 -0.12 0.68 0.7 -0.15 1
age bp bgr bu sc sod pot hemo pcv wbcc rbcc
1.00 0.75 0.50 0.25 0.00 −0.25 −0.50 −0.75 −1.00