Andrew Garcia
2/15/2022

ECE 5984 SP22 – Prof. Jones – HW2

1. Print to the Python console a more complete set of statistics on the data. You may use numpy operations on a numpy array, or operations on a Pandas DataFrame (or Series) for this functionality. The output that I want is as follows (not all columns are shown):

Python Console Output

```
In [80]: runfile('C:/Users/agarc/OneDrive/Documents/GitHub/Virginia_Tech_Masters/ECE_5984_Appl_Machine_Learning_SP22/Homework_2/simple_stats.py', wdir='C:/Users/agarc/OneDrive/
Documents/GitHub/Virginia_Tech_Masters/ECE_5984_Appl_Machine_Learning_SP22/Homework_2')
Reloaded modules: stats_report
File C:\Users\agarc\OneDrive\Documents\GitHub\Virginia_Tech_Masters\ECE_5984_Appl_Machine_Learning_SP22\Homework_2\Heart Disease.xlsx is of size (303, 16)
          stat   member         age  sex          cp    trestbps      chol         fbs     restecg  thalach       exang     oldpeak       slope          ca        thal   bt      target
0  cardinality      301   41.000000    2    4.000000   49.000000     152    2.000000    3.000000       87    2.000000   40.000000    3.000000    5.000000    4.000000    5    2.000000
1         mean  48332.7   54.366337  N/A    0.970199  131.615894  246.59    0.152027    0.529801  149.131    0.324503    1.043046    1.398671    0.741611    2.313531  N/A    0.544554
2       median    48340   55.000000  N/A    1.000000  130.000000   241.5    0.000000    1.000000      152    0.000000    0.800000    1.000000    0.000000    2.000000  N/A    1.000000
3  n_at_median        1    8.000000  N/A   50.000000   36.000000       0  251.000000  152.000000        7  204.000000   13.000000  139.000000  170.000000  166.000000  N/A  165.000000
4         mode      N/A   58.000000  N/A    0.000000  120.000000     N/A    0.000000    1.000000      N/A    0.000000    0.000000    2.000000    0.000000    2.000000  N/A    1.000000
5    n_at_mode      N/A   19.000000  N/A  142.000000   37.000000     N/A  251.000000  152.000000      N/A  204.000000   98.000000  141.000000  170.000000  166.000000  N/A  165.000000
6       stddev  877.941    9.082101  N/A    1.032257   17.566716  51.9697    0.359655    0.525849  22.5955    0.468966    1.161452    0.616872    1.026753    0.612277  N/A    0.498835
7          min    46820   29.000000  N/A    0.000000   94.000000     126    0.000000    0.000000       71    0.000000    0.000000    0.000000    0.000000    0.000000  N/A    0.000000
8          max    49840   77.000000  N/A    3.000000  200.000000     564    1.000000    2.000000      202    1.000000    6.200000    2.000000    4.000000    3.000000  N/A    1.000000
9        nzero        0    0.000000    0  142.000000    0.000000       0  251.000000  146.000000        0  204.000000   98.000000   21.000000  170.000000    2.000000    0  138.000000
10     nmissing        2    0.000000    1    1.000000    1.000000       3    7.000000    1.000000       35    1.000000    1.000000    2.000000    5.000000    0.000000    5    0.000000
```

2. Write a similar report to an Excel workbook. I used operations on a pandas DataFrame for this functionality. Note: "cardinality" is the number of distinct values. The output that I want in the spreadsheet is as follows (note, yours will have numbers 😊😊):

Report Excel File:

	stat	member	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	bt	target
0	cardinality	301	41	2	4	49	152	2	3	87	2	40	3	5	4	5	2
1	mean	48332.66	54.36634	N/A	0.970199	131.6159	246.59	0.152027	0.529801	149.1306	0.324503	1.043046	1.398671	0.741611	2.313531	N/A	0.544554
2	median	48340	55	N/A	1	130	241.5	0	1	152	0	0.8	1	0	2	N/A	1
3	n_at_med	1	8	N/A	50	36	0	251	152	7	204	13	139	170	166	N/A	165
4	mode	N/A	58	N/A	0	120	N/A	0	1	N/A	0	0	2	0	2	N/A	1
5	n_at_mod	N/A	19	N/A	142	37	N/A	251	152	N/A	204	98	141	170	166	N/A	165
6	stddev	877.9405	9.082101	N/A	1.032257	17.56672	51.96965	0.359655	0.525849	22.59548	0.468966	1.161452	0.616872	1.026753	0.612277	N/A	0.498835
7	min	46820	29	N/A	0	94	126	0	0	71	0	0	0	0	0	N/A	0
8	max	49840	77	N/A	3	200	564	1	2	202	1	6.2	2	4	3	N/A	1
9	nzero	0	0	0	142	0	0	251	146	0	204	98	21	170	2	0	138
10	nmissing	2	0	1	1	1	3	7	1	35	1	1	2	5	0	5	0

3. From this DQR, determine a few things about each column in the data set. For every column in the data, tell me each of the following:
    a. The type of the feature (ID, target, or feature – and what type of feature – continuous, binary, interval, categorical, etc.)

Member = ID
Sex = Binary
Age = Numeric
CP = Categorical
Trestbps = Numeric
Chol = Numeric / Categorical
FBS = Binary
Restecg = Categorical
Thalach = Numeric
Exang = Binary
Oldpeak = Numeric
Slope = Categorical

Ca = Categorical
Thal = Categorical
Bt = Categorical
Target = Binary

b. How many values are missing and how many are invalid?
Total Missing Values: 65
Total Invalid Stats: 22 (not including stats that don't make sense)
Member = 2
Age = 0
Sex = 1
CP = 1
Trestbps = 1
Chol = 3
FBS = 7
Restecg = 1
Thalach = 35
Exang = 1
Oldpeak = 1
Slope = 2
Ca = 5
Thal = 0
Bt = 5
Target = 0

c. What should be done about the missing values – BE SPECIFIC (don't just say "replace missing values" but tell me how)
For numeric features such as Age, Blood Pressure, or Cholesterol, it would be best to replace those missing values with the average. For categorical and binary features, it would be best to replace those missing values with the mode. And for ID features such as Member, it would be best to not replace those values because there is no meaningful statistic that would replace an ID.

d. Whether the feature contains a significant number of outliers
Member: None
Age: None
Sex: None
CP: None
Trestbps: None
Chol: None:
Fbs: None
Restecg: None
Thalach: None
Exang: None
Oldpeak: None
Slope: None

CA: Has cardinality of 5 however it should be cardinality of 4 based on the website.
Thal: Has cardinality of 4 however it should be cardinality of 3 based on the website.
Bt: None
Target: None

e. Whether the feature should be ignored in modeling (by removing the column).
Member: Yes
Age: No
Sex: No
CP: No
Trestbps: No
Chol: No
Fbs: No
Restecg: No
Thalach: No
Exang: No
Oldpeak: No
Slope: No
CA: No
Thal: No
Bt: No
Target: No

4. Calculate and write to Excel workbooks the covariance and correlation matrices for the numeric values in this data set. Create data frames with a row and a column for each numeric value; the entries in the cells are the covariances and the correlations for each pair of numeric features. Include the target value if it's numeric.

Covariance Output:

	member	age	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
member	770779.6	1485.512	-368.049	1703.156	849.032	-5.55443	-8.75953	-8177.45	147.9313	308.3186	-150.858	332.2046	138.1584	-378.277
age	1485.512	82.48456	-0.61136	44.66247	102.9632	0.384963	-0.54518	-73.3305	0.451123	2.244928	-0.95879	2.565035	0.378139	-1.02134
cp	-368.049	-0.61136	1.065554	0.829635	-4.48857	0.034763	0.024086	6.56436	-0.19806	-0.18546	0.077503	-0.19194	-0.10033	0.22554
trestbps	1703.156	44.66247	0.829635	308.5895	110.9536	1.129367	-1.04833	-15.4046	0.589535	3.956998	-1.31787	1.82032	0.672724	-1.27577
chol	849.032	102.9632	-4.48857	110.9536	2700.845	0.370144	-4.26464	16.46495	1.733934	3.545596	-0.00628	3.706157	2.959699	-2.07552
fbs	-5.55443	0.384963	0.034763	1.129367	0.370144	0.129352	-0.0158	-0.12661	0.004093	0.002277	-0.01334	0.051147	-0.00828	-0.00346
restecg	-8.75953	-0.54518	0.024086	-1.04833	-4.26464	-0.0158	0.276518	0.261412	-0.01746	-0.03644	0.029866	-0.03816	-0.00266	0.035159
thalach	-8177.45	-73.3305	6.56436	-15.4046	16.46495	-0.12661	0.261412	510.5559	-3.79813	-8.98838	5.357866	-3.98486	-1.15117	4.667393
exang	147.9313	0.451123	-0.19806	0.589535	1.733934	0.004093	-0.01746	-3.79813	0.219929	0.15093	-0.07291	0.05683	0.058129	-0.10147
oldpeak	308.3186	2.244928	-0.18546	3.956998	3.545596	0.002277	-0.03644	-8.98838	0.15093	1.348971	-0.4159	0.275694	0.148872	-0.25217
slope	-150.858	-0.95879	0.077503	-1.31787	-0.00628	-0.01334	0.029866	5.357866	-0.07291	-0.4159	0.380532	-0.05575	-0.03958	0.106722
ca	332.2046	2.565035	-0.19194	1.82032	3.706157	0.051147	-0.03816	-3.98486	0.05683	0.275694	-0.05575	1.054222	0.090254	-0.1975
thal	138.1584	0.378139	-0.10033	0.672724	2.959699	-0.00828	-0.00266	-1.15117	0.058129	0.148872	-0.03958	0.090254	0.374883	-0.10508
target	-378.277	-1.02134	0.22554	-1.27577	-2.07552	-0.00346	0.035159	4.667393	-0.10147	-0.25217	0.106722	-0.1975	-0.10508	0.248836

Correlation Output:

Andrew Garcia
2/15/2022

	member	age	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
member	1	0.185808	-0.408	0.110312	0.018508	-0.01795	-0.01901	-0.40954	0.358983	0.300974	-0.27801	0.374141	0.25639	-0.86329
age	0.185808	1	-0.06525	0.279508	0.217346	0.117935	-0.11407	-0.36278	0.106317	0.212657	-0.17089	0.275932	0.068001	-0.22544
cp	-0.408	-0.06525	1	0.045633	-0.08363	0.09284	0.04435	0.288622	-0.41076	-0.15448	0.121477	-0.18119	-0.15881	0.437885
trestbps	0.110312	0.279508	0.045633	1	0.121418	0.176945	-0.1133	-0.03798	0.071432	0.193315	-0.12144	0.100169	0.06247	-0.14555
chol	0.018508	0.217346	-0.08363	0.121418	1	0.019904	-0.1559	0.013848	0.07104	0.058719	-0.0002	0.069296	0.093505	-0.08
fbs	-0.01795	0.117935	0.09284	0.176945	0.019904	1	-0.08327	-0.01537	0.024151	0.005434	-0.06024	0.139262	-0.03761	-0.01924
restecg	-0.01901	-0.11407	0.04435	-0.1133	-0.1559	-0.08327	1	0.021833	-0.07092	-0.05962	0.092035	-0.0705	-0.00827	0.134078
thalach	-0.40954	-0.36278	0.288622	-0.03798	0.013848	-0.01537	0.021833	1	-0.36053	-0.34325	0.385144	-0.16838	-0.08212	0.413508
exang	0.358983	0.106317	-0.41076	0.071432	0.07104	0.024151	-0.07092	-0.36053	1	0.279031	-0.25202	0.117729	0.20253	-0.4339
oldpeak	0.300974	0.212657	-0.15448	0.193315	0.058719	0.005434	-0.05962	-0.34325	0.279031	1	-0.57906	0.231374	0.20909	-0.43539
slope	-0.27801	-0.17089	0.121477	-0.12144	-0.0002	-0.06024	0.092035	0.385144	-0.25202	-0.57906	1	-0.08905	-0.10453	0.346633
ca	0.374141	0.275932	-0.18119	0.100169	0.069296	0.139262	-0.0705	-0.16838	0.117729	0.231374	-0.08905	1	0.143738	-0.38511
thal	0.25639	0.068001	-0.15881	0.06247	0.093505	-0.03761	-0.00827	-0.08212	0.20253	0.20909	-0.10453	0.143738	1	-0.34403
target	-0.86329	-0.22544	0.437885	-0.14555	-0.08	-0.01924	0.134078	0.413508	-0.4339	-0.43539	0.346633	-0.38511	-0.34403	1

5. From these workbooks, determine the three feature values (predictors) that are most highly correlated with the target; list them and their correlation. Note that either a large positive or a large negative correlation with the target indicates a good predictor. Also, find the three predictors that are the most highly correlated with each other; list them and their cross-correlation.

Three features highly correlated to target: CP, Exang, and Oldpeak

Three predictors highly correlated:

Slope and Oldpeak: -0.579058

Slope and Thalach: 0.385144

Oldpeak and Thalach: -0.34325

Andrew Garcia
2/15/2022

Python Code:

simple_stats.py

```
import pandas

import stats_report as sr


filename = r"C:\Users\agarc\OneDrive\Documents\GitHub\Virginia_Tech_Masters\ECE_5984_Appl_Machine_Learning_SP22\Homework_2\Heart Disease.xlsx"

df = pandas.read_excel(filename) # read Excel spreadsheet

print('File {0} is of size {1}'.format(filename, df.shape))

labels = df.columns

report = sr.StatsReport()


# Create a simple data set summary for the console

for thisLabel in labels: # for each column, report stats

    thisCol = df[thisLabel]


    report.addCol(thisLabel, thisCol)


print(report.to_string())

report.statsdf.to_excel("Report_Andrew_Garcia.xlsx")


covariance = df.cov()

correlation = df.corr()

covariance.to_excel("Covariance_Andrew_Garcia.xlsx")

correlation.to_excel("Correlation_Andrew_Garcia.xlsx")
```

Andrew Garcia
2/15/2022

```
labels = correlation.columns

for thisLabel in labels:
    if thisLabel == "member":
        pass
    else:
        thisCol = correlation[thisLabel]
        v = thisCol.sort_values()
        max_v = v[-2]
        min_v = v[0]
        min_v_abs = abs(min_v)
        if max_v > min_v_abs:
            max_overall = max_v
            print(f"\n Label: {thisLabel} - Highest: {max_v}")
        elif min_v_abs > max_v:
            max_overall = min_v_abs
            print(f"\n Label: {thisLabel} - Highest: {min_v}")
```

stats_report.py

```
import pandas


class StatsReport:
    def __init__(self):
        self.statsdf = pandas.DataFrame()
        self.statsdf['stat'] = ['cardinality', 'mean', 'median', 'n_at_median', 'mode', 'n_at_mode', 'stddev',
'min', 'max', 'nzero', 'nmissing']
        pass


    def addCol(self, label, data):
```

```
        self.statsdf[label] = [self.cardinality_(data), self.mean_(data), self.median_(data),
self.n_at_median(data), self.mode_(data), self.n_at_mode(data), self.std_(data), self.min_(data),
self.max_(data), self.nzero_(data), self.nmissing_(data)]


    def to_string(self):
        return self.statsdf.to_string()


    def cardinality_(self, d):
        try:
            return d.nunique()
        except:
            return "N/A"


    def mean_(self, d):
        try:
            return d.mean()
        except:
            return "N/A"


    def median_(self, d):
        try:
            return d.median()
        except:
            return "N/A"


    def n_at_median(self, d):
        try:
            n = d == d.median()
            return n.sum()
```

```
        except:
            return "N/A"

    def mode_(self, d):
        try:
            return int(d.mode())
        except:
            return "N/A"

    def n_at_mode(self, d):
        try:
            n = d == int(d.mode())
            return n.sum()
        except:
            return "N/A"

    def std_(self, d):
        try:
            return d.std()
        except:
            return "N/A"

    def min_(self, d):
        try:
            return d.min()
        except:
            return "N/A"

    def max_(self, d):
```

```
        try:
            return d.max()
        except:
            return "N/A"


    def nzero_(self, d):
        try:
            n = d == 0
            return n.sum()
        except:
            return "N/A"


    def nmissing_(self, d):
        try:
            n = d.isna()
            return n.sum()
        except:
            return "N/A"
```