**Author: Mark Maxwell**

**Project: Chapter 2 Answer Key for 'Data Science Using Python and R'**

**Python 3.7.0**

**Last Modified 02/09/2020**

**Chapter 2 - Working with the Data, Page 26-27**

**11. Download the program and open the compiler. What is contained in the bottom‐right window? The left (for Python) or top‐left (for R)?**

The text editor is defaulted to the left side pane in the Spyder IDE.

**12. Type a comment stating that you are working on Chapter 2 exercises.**

#I am working on Chapter 2 exercises

**13. Locate the “Run” button and note whether there is a keyboard shortcut.**

The run selection or current line button uses the F9 keyboard shortcut by default.

**14. Execute the comment from the previous exercise. What is the output? Explain your answer.**

The comment is not compiled by Python so no output is returned.

**15. Import the following packages:**

**a. For Python, import the pandas and numpy packages. Rename the pandas package “pd” and rename the numpy package “np.”**

**b. For R, import the ggplot2 package. Make sure you both install and open the package.**

**import** pandas as pd

**import** numpy as np

**16. Import the bank\_marketing\_training data set and name it bank\_train.**

bank\_train = pd.read\_csv("/Users/markmaxwell/Desktop/Website Data Sets/bank\_marketing\_training")

The above code uses an absolute path or a relative path can be set using the os package

**import** os

os.chdir("/Users/markmaxwell/Desktop/Website Data Sets/")

bank\_train = pd.read\_csv("bank\_marketing\_training")

**17. Create a contingency table of the variables response and previous\_outcome from the bank\_train data set. Do not save the output from the code.**

pd.crosstab(bank\_train['previous\_outcome'], bank\_train['response'])

﻿response no yes

previous\_outcome

failure 2390 385

nonexistent 21176 2034

success 320 569

**18. Rerun the code from the previous exercise, this time saving the output as crosstab\_01 (for Python code) or t1 (for R code).**

crosstab\_01 = pd.crosstab(bank\_train['previous\_outcome'], bank\_train['response'])

**19. After saving the output in the previous exercise, display the output using the name of the saved output.**

crosstab\_01

﻿response no yes

previous\_outcome

failure 2390 385

nonexistent 21176 2034

success 320 569

**20. Save the contingency table under a different name. This time, use your last name and favorite number as the name; for example, larose42.**

mark7 = crosstab\_01

**21. Save the first nine records of the bank\_train data set as their own data frame.**

first\_nine = bank\_train.loc[0:8]

**22. Save the age and marital records of the bank\_train data set as their own data frame.**

age\_marital = bank\_train[['age','marital']]

**23. Save the first three records of the age and marital variables as their own data frame.**

first\_three\_age\_marital = bank\_train.loc[0:2,['age','marital']]

**Chapter 2 - Hands on Analysis, Page 27-28**

**﻿24. Import the adult\_ch3\_training data set using the “Heading: Yes” setting. Rename the data set adult once it is imported.**

adult\_train = pd.read\_csv("/Users/markmaxwell/Desktop/Website Data Sets/adult\_ch3\_training")

**25. Write a comment explaining the change in the data set name.**

The name chosen is short but descriptive so the developer can remember what the variable contains. A shorter name is preferred since every time the data is needed the variable name needs to be typed in full.

**26. Import the following packages:**

**a. For Python, import the DecisionTreeClassifier command from the sklearn.tree package.**

**b. For R, import the rpart package. Make sure you both install and open the package.**

**from** sklearn.tree **import** DecisionTreeClassifier

**27. Create a contingency table of workclass and sex. Save the output as table01.**

table01 = pd.crosstab(adult\_train['workclass'],adult\_train['sex'])

**28. Create a contingency table of sex and marital status. Save the output as table02.**

table02 = pd.crosstab(adult\_train['marital-status'],adult\_train['sex'])

**29. Display the sex and workclass values of the person in the first record. What cell of table01 do they belong to? How many other records in the data set have the same sex and workclass values?**

adult\_train.loc[0,['sex','workclass']]

﻿sex Male

workclass Self-emp-not-inc

Name: 0, dtype: object

The first record is sex : Male, workclass : Self-emp-not-inc

table01 has this group as the seventh row and second column of the table

table01 shows that this group represents 992 records in the data

**30. Display the sex and marital status values of the people in records 6–10. Which cells of table02 do they belong to? How many other records in the data set have the same combinations of sex and marital status values?**

adult\_train.loc[5:9,['sex','marital-status']]

﻿ sex marital-status

5 Male Married-civ-spouse

6 Male Married-civ-spouse

7 Male Married-civ-spouse

8 Male Married-civ-spouse

9 Male Divorced

The 6-10 records of the data contain sex : Male, marital-status : ['Married-civ-spouse','Divorced']

sex : Male, marital-status : 'Married-civ-spouse' belongs to row 3 column 2 and contains 6,010 records

sex : Male, marital-status : 'Divorced' belongs to row 1 column 2 and contains 795 records

**31. Create a new data set that has only records whose marital status is “Married‐civ‐spouse” and name the data set adultMarried.**

adultMarried = adult\_train.loc[adult\_train['marital-status']=='Married-civ-spouse',]

**32. Recreate the contingency table of sex and workclass using the adultMarried data set. What differences do you notice between the sexes?**

pd.crosstab(adultMarried['workclass'],adultMarried['sex'])

﻿sex Female Male

workclass

? 67 224

Federal-gov 20 203

Local-gov 68 411

Never-worked 1 0

Private 491 3883

Self-emp-inc 24 347

Self-emp-not-inc 57 703

State-gov 33 237

Without-pay 0 2

The proportion of records for females dropped significantly more than for males indicating is more common for females to be out of the workforce within the 'Married-civ-spouse' marital-status.

**33. Create a new data set that has only records whose age value is greater than 40. Name the new data set adultOver40.**

adultOver40 = adult\_train.loc[adult\_train['age']>40,]

**34. Recreate the contingency table of sex and marital status using the adultOver40 data set. What differences do you notice?**

pd.crosstab(adultOver40['marital-status'],adultOver40['sex'])

﻿sex Female Male

marital-status

Divorced 720 448

Married-civ-spouse 305 3383

Married-spouse-absent 38 47

Never-married 256 315

Separated 133 74

Widowed 354 71

The 'Never-married' marital-status dropped significantly for the adults over 40 subset.