PART 1: DATA ANALYSIS USING SAS

/* Task 1 Import the data from amphibians.xlsx to SAS */

```
/* The code aims to import datafilr from excel to SAS*/


/* VALIDVARNAME option to v7 to allow variable names up to 7 characters */
OPTIONS VALIDVARNAME=v7;
    /* PROC IMPORT statement to import the data from the specified Excel file */
    PROC IMPORT DATAFILE='/home/u63290591/sasuser.v94/amphibians.xlsx' DBMS=xlsx
    OUT=work.amphibians REPLACE;
        /* Specify that the first row of the Excel file contains variable names */
        GETNAMES=yes;
    /* RUN statement indicates the end of the PROC IMPORT step */
    RUN;

/* Use the PROC PRINT statement to display the imported data */
PROC PRINT DATA=work.amphibians;
    /* RUN statement indicates the end of the PROC PRINT step */
RUN;

/* By running this code, the excel file will be imported */
```

/* Task 2 Analyse the seven amphibian variables. For each of the species: find the number of sites where the species was observed */

```
/* The given code analyses the seven amphibian variables in the dataset. It computes the number of times each species was observed */

/* PROC MEANS analysis VARIABLE 'green_frogs' in the dataset 'work.amphibians */
/* SUM option calculated the total amount of 'green_frogs' */
/* MAXDEC=0 means no decimal places displayed for the calculated values */
/* nolabels ensures that no labels are displayed */
PROC MEANS DATA=work.amphibians SUM MAXDEC=0 nolabels;
        VAR green_frogs;
        /* OUTPUT statement saves the sum of 'green_frogs' in a dataset named 'work.greenfrogs_sum' */
        OUTPUT OUT=work.greenfrogs_sum
        /* SUM= option calculated the total sum of 'green_frogs' should be saved in a variable named 'Greenfrogs_sum' within the output dataset */
        SUM=Greenfrogs_sum;
        /* RUN statement indicates the end of the PROC MEANS step */
RUN;

/* By running this code, will obtain the sum of 'green_frogs' */

/* Similar to the previous step, this PROC MEANS calculates the sum of 'brown_frogs' and saves it in 'work.brownfrogs_sum' */
PROC MEANS DATA=work.amphibians SUM MAXDEC=0 nolabels;
        VAR brown_frogs;
        OUTPUT OUT=work.brownfrogs_sum
        SUM=Brownfrogs_sum;
RUN;
```

```
/* Calculates the sum of 'common_toad' and saves it in 'work.commontoad_sum' */
PROC MEANS DATA=work.amphibians SUM MAXDEC=0 nolabels;
	VAR common_toad;
	OUTPUT OUT=work.commontoad_sum
	SUM=Commontoad_sum;
RUN;

/* Calculates the sum of 'fire_bellied_toad' and saves it in 'work.Firebelliedtoad_sum' */
PROC MEANS DATA=work.amphibians SUM MAXDEC=0 nolabels;
	VAR fire_bellied_toad;
	OUTPUT OUT=work.firebelliedtoad_sum
	SUM=Firebelliedtoad_sum;
RUN;

/* Calculates the sum of 'tree_frog' and saves it in 'work.treefrog_sum' */
PROC MEANS DATA=work.amphibians SUM MAXDEC=0 nolabels;
	VAR tree_frog;
	OUTPUT OUT=work.treefrog_sum
	SUM=Treefrog_sum;
RUN;

/* Calculates the sum of 'common_newt' and saves it in 'work.commonnewt_sum' */
PROC MEANS DATA=work.amphibians SUM MAXDEC=0 nolabels;
	VAR common_newt;
	OUTPUT OUT=work.commonnewt_sum
	SUM=Commonnewt_sum;
RUN;
```

```
/* Calculates the sum of 'great_crested_newt' and saves it in 'work.greatcrestednewt_sum' */
PROC MEANS DATA=work.amphibians SUM MAXDEC=0 nolabels;
        VAR great_crested_newt;
        OUTPUT OUT=work.greatcrestednewt_sum
        SUM=Greatcrestednewt_sum;
RUN;

/* A new dataset named 'work.sites' is created using the DATA statement */
/* DROP statement excludes variables (type and freq) from the 'work.sites' dataset */
DATA work.sites(DROP=_type_ _freq_);
        /* SET statement combines datasets of each species that created previously */
        SET work.greenfrogs_sum work.brownfrogs_sum work.commontoad_sum
            work.firebelliedtoad_sum work.treefrog_sum work.commonnewt_sum
            work.greatcrestednewt_sum;
        /* Number_Of_Sites variable created using the SUM function, which sums up the
        counts for each species */
        Number_Of_Sites=SUM(Greenfrogs_sum, Brownfrogs_sum, Commontoad_sum,
                            Firebelliedtoad_sum, Treefrog_sum, Commonnewt_sum,
                            Greatcrestednewt_sum);
        /* RUN statement indicates the end of the DATA step */
RUN;

/* PROC PRINT  print the contents of the 'work.sites' dataset, which includes the
Number_Of_Sites variable */
PROC PRINT DATA=work.sites;
        /* RUN statement indicates the end of the PROC PRINT step */
RUN;

/* The final dataset, 'work.sites', contains the total number of sites where each species was
observed */
```

/* Task 3 Which species was observed most often, i.e. at the largest number of sites? */

```
/* The code aims to identify the species that was observed most often, meaning the species that appeared at the largest number of sites */

/* PROC SORT  sort the 'work.sites' dataset */
PROC SORT DATA=work.sites;
    /* BY statement specifies to sort the dataset in descending order of the variable 'Number_Of_Sites' */
    BY DESCENDING Number_Of_Sites;
    /* RUN statement indicates the end of the PROC SORT step */
RUN;

/* PROC PRINT print the contents of the 'work.sites' dataset */
/* (OBS=1) option is used to limit the proc print steps to print only the first observation, which means the species with the highest number of sites */
PROC PRINT DATA=work.sites (OBS=1);
    /* TITLE statement provide a title for the output, stating "The Species That Observed Most Often" */
    TITLE The Species That Observed Most Often;
    /* RUN statement indicates the end of the PROC PRINT step */
RUN;

/* By running this code, will obtain the species that was observed most often, as indicated by the highest number of sites in the dataset */
```

/* Task 4 Which species was observed least often, i.e. at the smallest number of sites? */

```
/* The code aims to identify the species that was observed least often, meaning the species that appeared at the smallest number of sites */

/* PROC SORT procedure is used to sort the 'work.sites' dataset */
PROC SORT DATA=work.sites;
    /* BY statement specifies to sort the dataset in ascending order of the variable 'Number_Of_Sites' */
    BY Number_Of_Sites;
    /* RUN statement indicates the end of the PROC SORT step */
RUN;

/* PROC PRINT print the contents of the 'work.sites' dataset */
/* (OBS=1) option is used to limit the output to only the first observation, which will correspond to the species with the smallest number of sites */
PROC PRINT DATA=work.sites (OBS=1);
    /* TITLE statement provides a title for the output, stating "The Species That Observed Least Often" */
    TITLE The Species That Observed Least Often;
    /* RUN statement indicates the end of the PROC PRINT step */
RUN;

/* By running this code, we obtain the species that was observed least often, as indicated by the smallest number of sites in the dataset */
```

/* Task 5 Create a new variable (species) as a sum of the seven amphibian variables. This will be the number of different species observed at the site */

```
/* The code creates a new variable named 'Species'.  'Species' variable represents the total count of different amphibian species observed at each site */

/* DATA statement specifies the dataset 'work.amphibians' as input and output dataset */
DATA work.amphibians;
    /* SET statement received the data from 'work.amphibians' for further processing */
    SET work.amphibians;
    /* Species variable is created using the SUM function to sum up the values of a list of the seven amphibian variables,  resulting in the total count of different species observed at each site */
    Species=SUM(of Green_frogs Brown_frogs Common_toad Fire_bellied_toad
                   Tree_frog Common_newt Great_crested_newt);
    /* RUN statement indicates the end of the DATA step */
RUN;

/* PROC PRINT procedure print the updated 'work.amphibians' dataset with the newly added 'Species' variable */
PROC PRINT DATA=work.amphibians;
    /* RUN statement indicates the end of the PROC PRINT step */
RUN;

/* By running this code, we create a new variable 'Species' in the 'work.amphibians' dataset that represents the total count of different amphibian species observed at each site */
```

/* Task 6 Produce the frequencies of all values of the variable species by motorway */

```
/* The code is to determine the frequencies of all values of the 'Species' variable for each unique value of the 'Motorway' variable */

/* PROC FREQ procedure is used to calculate the frequencies of variable combinations */
/* DATA statement specifies the dataset 'work.amphibians' as the input dataset for the frequency analysis */
PROC FREQ DATA=work.amphibians;
    /* TABLES statement specifies to analyze the frequencies of all combinations of 'Species' and 'Motorway' */
    /* NOPERCENT exclude the percentages from the output */
    /* NOROW and NOCOL options are used to suppress the display of row and column percentages in the output */
    TABLES Species*Motorway/NOPERCENT NOROW NOCOL
    /* OUT statement save the resulting frequency table in a new dataset named 'work.motorway' */
    OUT=work.motorway;
    /* RUN statement indicates the end of the PROC FREQ step */
RUN;

/* By running this code, we obtain a frequency table that shows the frequencies of all values of the 'Species' variable for each unique value of the 'Motorway' variable */
```

/* Task 7 What was the largest number of different species observed at any A1 site? */

```
/* The code is  to determine the largest number of different species observed at any A1 site */

/* DATA step specifies the dataset 'work.motorwayA1' as both the input and output dataset */
DATA work.motorwayA1;
    /* SET statement recieved the data from the 'work.motorway' dataset */
    SET work.motorway;
    /* WHERE statement  filtering only the observations where the 'Motorway' variable is
    equal to 'A1'  */
    WHERE Motorway='A1';
    /* RUN statement indicates the end of the DATA step */
RUN;

/* PROC MEANS procedure is used to calculate summary statistics */
/* DATA stept specifies the dataset 'work.motorwayA1' as the input dataset for analysis */
/* MAX option calculate the maximum value of the 'species' variable */
/* MAXDEC=0 means that no decimal places are displayed for the calculated values */
/* nolabels ensure the output is displayed without any labels */
PROC MEANS DATA=work.motorwayA1 MAX MAXDEC=0 nolabel;
    /* VAR statement specifies the variable 'species' as the variable for which the
    maximum value should be calculated */
    VAR species;
    /* TITLE statement provides a title for the output, stating "The Largest Number Of
    Different Species Observed At Any A1 Site" */
    TITLE The Largest Number Of Different Species Observed At Any A1 Site;
    /* RUN statement indicates the end of the PROC MEANS step */
RUN;

/* By running this code, we obtain the largest number of different species observed at any A1
site */
```

/* Task 8 What was the largest number of different species observed at any S52 site? */

```
/* The code is  to determine the largest number of different species observed at any S52 site */

/* DATA step specifies the dataset 'work.motorwayS52' as  input and output dataset */
DATA work.motorwayS52;
    /* SET statement received the data from the 'work.motorway' dataset */
    SET work.motorway;
    /* WHERE statement filtering only the observations where the 'Motorway' variable is
    equal to 'S52' */
    WHERE Motorway='S52';
    /* RUN statement indicates the end of the DATA step */
RUN;

/* PROC MEANS procedure is used to calculate summary statistics */
/* DATA statement specifies the dataset 'work.motorwayS52' as the input dataset  */
/* MAX option calculate the maximum value of the 'species' variable */
/* MAXDEC=0 means that no decimal places are displayed for the calculated values */
/* nolabels ensure the output is displayed without any labels */
PROC MEANS DATA=work.motorwayS52 MAX MAXDEC=0 nolabel;
    /* VAR statement specifies the variable 'species' as the variable for which the
    maximum value should be calculated */
    VAR species;
    /* TITLE statement provides a title for the output, stating "The Largest Number Of
    Different Species Observed At Any S52 Site" */
    TITLE The Largest Number Of Different Species Observed At Any S52 Site;
    /* RUN statement indicates the end of the PROC MEANS step */
RUN;

/* By running this code, we obtain the largest number of different species observed at any S52
site */
```

/* Task 9 Create a new variable (rich_site) that takes value 1 if there are more than three different species observed at the site and value 0 otherwise. This will be an indicator whether the site is rich in amphibians or not */

```
/* The code provided creates a new variable named 'Rich_Site'. The 'Rich_Site' variable serves as an indicator of whether a site is rich in amphibians or not */

/* DATA steps specifies the dataset 'work.amphibians' as both the input and output dataset */
DATA work.amphibians;
	/* SET statement received the data from 'work.amphibians' for further processing */
	SET work.amphibians;
	/* IF the variable 'Species' is greater than or equal to 3, THEN assigns the value "1" to the 'Rich_Site' variable, indicating that the site is rich in amphibians */
	IF Species>=3 THEN
		Rich_Site="1";
	/* ELSE assigns the value "0" to the 'Rich_Site' variable */
	ELSE
		Rich_Site="0";
	/* RUN statement indicates the end of the DATA step */
RUN;

/* PROC PRINT print the contents of the updated 'work.amphibians' dataset */
PROC PRINT DATA=work.amphibians;
	/* RUN statement indicates the end of the PROC PRINT step */
RUN;
/* By running this code, create a new variable 'Rich_Site' in the 'work.amphibians' dataset */
/* The 'Rich_Site' variable takes the value "1" if there are more than three different species observed at the site, indicating that the site is rich in amphibians */
/* If the condition is not met, the 'Rich_Site' variable takes the value "0", indicating that the site is not rich in amphibians */
```

/* Task 10 Produce the frequencies of all values of the variable rich_site */

```
/* The code is to determine the frequencies of all values of the 'Rich_Site' variable */


/* PROC FREQ calculate the frequencies of variable values */
/* DATA statement specifies the dataset 'work.amphibians' as the input dataset for the frequency analysis */
PROC FREQ DATA=work.amphibians;
        /* TABLES statement specifies the variable 'Rich_Site' to be analyzed */
        /* NOPERCENT exclude the percentages from the output */
        /* NOROW and NOCOL options are used to suppress the display of row and column percentages in the output */
        TABLES Rich_Site/NOPERCENT NOCUM
        /* OUT statement is used to save the resulting frequency table in a new dataset named 'work.richsite' */
        OUT=work.richsite;
        /* RUN statement indicates the end of the PROC FREQ step */
RUN;

/* By running this code, we obtain a frequency table that shows the frequencies of all values of the 'Rich_Site' variable */
```

/* Task 11 How many sites are rich in amphibians? */

/ The code aims to determine the number of sites that are rich in amphibians, based on the 'Rich_Site' variable */*

/ PROC PRINT print the contents of the 'work.richsite' dataset */*

/ DATA statement specifies 'work.richsite' as the input dataset for printing */*

```
PROC PRINT DATA=work.richsite NOOBS LABEL;
    /* WHERE statement is to specific to keep only the observations where the 'Rich_Site' variable is equal to '1' */
    WHERE Rich_Site='1';
    /* TITLE statement provides a title for the output, stating "The Number Of Sites Are Rich In Amphibians" */
    TITLE The Number Of Sites Are Rich In Amphibians;
    /* RUN statement indicates the end of the PROC PRINT step */
RUN;
```

/ By running this code, we obtain a table that displays the observations from the 'work.richsite' dataset, filtered to include only the sites that are rich in amphibians (where 'Rich_Site' is equal to '1') */*

/* Task 12 Calculate the correlation coefficient between surface of water reservoir in m2 (SR) and number of water reservoirs in habitat (NR) */

```
/* The code provided calculates the correlation coefficient between two variables: 'SR' (surface of water reservoir in m2) and 'NR' (number of water reservoirs in habitat) */

/* PROC CORR procedure calculated correlation coefficients */
/* DATA statement specifies the dataset 'work.amphibians' as the input dataset for the correlation analysis */
PROC CORR DATA=work.amphibians;
    /* VAR statement specifies the variables to be included in the correlation analysis, which are 'SR' and 'NR' */
    VAR SR NR;
    /* RUN statement indicates the end of the PROC CORR step */
RUN;

/* By running this code, the correlation coefficient between 'SR' and 'NR' will be calculated */
```

/* Task 13 Is this correlation positive or negative? What is its strength (very weak/weak/moderate/strong/very strong)? */

/*
Regarding the direction of the correlation, a correlation coefficient of 0.65276 indicates a positive correlation between the two variables. This means that as one variable increases, the other variable also tends to increase.

In terms of strength, a correlation coefficient of 0.65276 falls within the range of moderate correlation. This suggests that there is a discernible relationship between the variables, but it is not exceptionally strong.
*/

/* Task 14 Develop a logistic regression model where rich_site is a dependent variable, and SR, NR, FR, and RR are independent variables */

```
/* The code aims to develop a logistic regression model using the 'work.amphibians' dataset. The dependent variable in the model is 'rich_site', while the independent variables are 'SR', 'NR', 'FR', and 'RR' */

/* PROC LOGISTIC perform logistic regression analysis */
/* DATA statement specifies the dataset 'work.amphibians' as the input dataset for the logistic regression */
PROC LOGISTIC DATA=work.amphibians;
    /* 'rich_site' is specified as the dependent variable, while 'SR', 'NR', 'FR', and 'RR' are specified as independent variables in the MODEL */
    MODEL rich_site=SR NR FR RR;
    /* OUTPUT statement is used to save the output from the logistic regression analysis */
    /* 'OUT' option specifies that the output should be saved in a dataset named 'work.logistic' */
    OUTPUT OUT=work.logistic;
    /* RUN statement indicates the end of the PROC LOGISTIC step */
RUN;

/* By running this code, a logistic regression model will be developed with 'rich_site' as the dependent variable and 'SR', 'NR', 'FR', and 'RR' as the independent variables */
```

/* Task 15 What is the AUC (c statistic) of this model? Based on the AUC, does the model separate the sites which are rich in amphibians well? */

/*
The area under the curve (AUC), also known as the concordance statistic or c-statistic, provides a measure of how well a model can accurately classify outcomes. It can be interpreted as follows:

If the AUC value is less than 0.5, the model is considered subpar, indicating that its predictive ability is worse than chance. In other words, the model performs poorly in distinguishing between different outcomes.

A value of 0.5 suggests that the model's predictive ability is equivalent to random chance. It means the model has no discriminative power to distinguish between outcomes and is essentially guessing.

The closer the AUC value is to 1, the more accurately the model can classify the outcomes. As the AUC increases, the model's ability to differentiate between different outcomes improves.

An AUC value of 1 indicates that the model can classify outcomes with absolute accuracy. However, it is important to note that achieving a perfect AUC is rare in most real-world scenarios.

Therefore, the AUC or c-statistic provides insights into how well a model can accurately classify outcomes, with higher values indicating better classification performance.

Based on the current analysis, the logistic regression model with an AUC (c-statistic) of 0.665 can somewhat distinguish between areas that have a high amphibian population and those that do not. However, the separation between the two groups is not particularly strong.

An AUC of 0.665 suggests that there is still a considerable overlap in the distributions of the predictor variables (SR, NR, FR, and RR) between areas with a high amphibian population and those without. This indicates that the model's ability to accurately predict amphibian-rich sites is limited.

Therefore, while the model does have some predictive value, it may not be reliable enough to make critical judgments or draw firm conclusions without additional research or improvement. It is important to conduct further studies or enhance the model to increase its accuracy and reliability before relying on it for important decision-making processes.
*/

/* Task 16 Create a pdf report (richest_in_amphibians.pdf) that contains the list of those sites where six or seven different species were observed. Include only the variables id, motorway, and species. Suppress the observation number column. Use the Ocean style and add a title ("Sites Richest in Amphibians") */

```
/* The code provided generates a PDF file named "richest_in_amphibians.pdf" using the ODS PDF statement in SAS. The PDF file will be created in the specified file path "/home/u63290591/Work/" and will use the "Ocean" style */

/* ODS PDF statement sets up the output destination as a PDF file */
/* FILE option specifies the file path and name of the PDF file to be created */
/* STYLE option specifies "Ocean" style template to be used */
ODS PDF FILE='/home/u63290591/Work/richest_in_amphibians.pdf' STYLE=Ocean;

    /* PROC PRINT procedure print the contents of the 'work.amphibians' dataset */
    /* DATA statement specifies the dataset to be printed */
    /* WHERE statement is used to filter only the sites where six or seven different species were observed */
    /* NOOBS option is used to suppress the observation number column in the output */
    /* LABEL option is included to display variable labels in the output */
    PROC PRINT DATA=work.amphibians (WHERE=(Species >=6)) NOOBS LABEL;
        /* VAR statement specifies 'id', 'motorway', and 'species' to be included in the print output */
        VAR id motorway species;
        /* TITLE statement a title for the PDF file, stating "Sites Richest in Amphibians" will be displayed at the top of the PDF file */
        TITLE "Sites Richest in Amphibians";
        /* RUN statement indicates the end of the PROC PRINT step */
    RUN;
/* ODS PDF CLOSE statement closes the PDF output destination, finalizing the PDF file */
```

ODS PDF CLOSE;

/ By running this code, a PDF report named "richest_in_amphibians.pdf" will be created in the specified file path */*

PART 2: SHORT REPORT

Based on several considerations, I would not recommend SAS as the software for business analytics to my future employer. Here are the reasons supporting this decision:

1. ***Cost:*** One significant drawback of SAS is its higher licensing and maintenance costs compared to other analytics software options. This can pose a financial burden, particularly for small businesses or organizations with limited financial resources. Allocating a substantial budget for software licensing may not be feasible or justifiable when there are more cost-effective alternatives available.
2. ***Programming Complexity:*** SAS utilizes its own programming language, which can be more challenging to learn and use compared to other analytical tools such as Python or R. Adopting SAS would require the organization to either hire professionals with a strong SAS programming background or invest significant time and effort into training existing employees. This additional requirement may slow down the implementation process and hinder the productivity of analytical teams, especially if they are not familiar with SAS.
3. ***User Interface Limitations:*** Many users find SAS's interface to be less intuitive and visually appealing compared to other modern analytical tools. This can lead to a less efficient and user-friendly experience for analysts and data professionals. In a rapidly evolving data analytics landscape, having an intuitive and user-friendly interface is crucial for maximizing productivity and enabling seamless collaboration across teams.
4. ***Availability of Alternatives:*** In recent years, there has been a proliferation of alternative analytics software options, such as Python, R, and other open-source languages. These alternatives offer comparable or even superior functionality while being more cost-effective, flexible, and widely adopted by the analytics community. Choosing a more widely used and supported tool can provide access to a larger talent pool and a vibrant user community for sharing knowledge and best practices.

Considering these factors, it is important to evaluate the specific needs, budget, and skill set of the organization before making a decision on the choice of analytics software. While SAS has its strengths and may be suitable for certain use cases or industries, the aforementioned limitations and the availability of more cost-effective and flexible alternatives make it less favorable for many organizations today.