Accessing Data

1. Write a PROC CONTENTS step to generate a report of the storm_summary.sas7bdat table.

```
proc contents data="FILEPATH/storm_summary.sas7bdat";
run;
```

2. How many observations are in the table?

3118

3. How is the table sorted?

by Season, Name

4. Double-click the **storm_summary.sas7bdat** SAS table to view it. How are missing character and numeric values represented in the data?

A blank is stored for a character missing value and a period is stored for a numeric missing value.

5. Examine the length of the **Basin** column. Could *East Pacific* be properly stored as a data value in the **Basin** column?

No. **Basin** has a length of 2 bytes, so *East Pacific* would be truncated, and the value would be *Ea*.

6. Create a library named PG1.

```
libname pg1 base "/home/u48709362/EPG194/data";
```

7. Write a LIBNAME statement to create a library named **NP** that reads **np_info.xlsx** in the course data as follows:

```
libname np xlsx "/home/u48709362/EPG194/data/np_info.xlsx";
```

8. Navigate to your list of libraries and open the **NP** library. How many tables are there in the **NP** library?

There are three tables in the **NP** library: **Parks**, **Species**, and **Visits**.

9. Write an OPTIONS statement to ensure that column names follow SAS naming conventions.

```
options validvarname=v7;
```

10. Write a PROC CONTENTS step to read the **Parks** table in the **NP** library.

```
proc contents data=np.parks;
run;
```

11. Run the program and examine the log. Which column names have been modified to follow SAS naming conventions?

Park Code changed to **Park_Code** and **Park Name** changed to **Park_Name.**

12. Import storm_damage.tab file.

```
proc import datafile="/home/u48709362/EPG194/data/storm_damage.tab"
        dbms=tab
        out=storm_damage replace;
run
```

13. Complete the PROC IMPORT step to read **eu_sport_trade.xlsx**. Create a SAS table named **eu_sport_trade** and replace the table if it exists.

```
proc import datafile="/home/u48709362/EPG194/data/eu_sport_trade.xlsx"
        dbms=xlsx
        out=eu_sport_trade
        replace;
run;
```

14. Create PROC CONTENTS code to display the descriptor portion of the **eu_sport_trade** table.

```
proc contents data=eu_sport_trade;
run;
```