**Homework 5**

**BIST/STAT 6494 Fall 2017**

**Haim Bar**

**Due Date: October 15, 2017**

**(by 23:59, via HuskyCT)**

1. **SAS:** Using the Hosp data set, compute the frequencies for the days of the week, months of the ear, and year, corresponding to the admission dates (variable AdmitDate).

Supply a format for the days of the week and months of the year. Use PROC FREQ to list these frequencies.

libname hw5 'P:\STAT-6494-Data Management in SAS and R\data';

**data** freq;

set hw5.hosp(keep=AdmitDate);

Day=weekday(AdmitDate);

Month=month(AdmitDate);

Year=year(AdmitDate);

**run**;

**proc** **format** ;

value days **1**='Sun' **2**='Mon' **3**='Tue' **4**='Wed' **5**='Thu' **6**='Fri' **7**='Sat';

value months **1**='Jan' **2**='Feb' **3**='Mar' **4**='Apr' **5**='May' **6**='Jun'

**7**='Jul' **8**='Aug' **9**='Sep' **10**='Oct' **11**='Nov' **12**='Dec';

**run**;

**proc** **freq** data=freq ;

tables Day Month Year / nocum nopercent;

format Day days. Month months.;

**run**;

1. **R:** Repeat question 1, compute the frequencies for the days of the week, months of the ear, and year, corresponding to the admission dates. And print the frequencies.

**PROC** **EXPORT** DATA= hw5.hosp

OUTFILE= "P:\STAT-6494-Data Management in SAS and R\data\hosp.csv"

DBMS=CSV REPLACE;

PUTNAMES=YES;

**RUN**;

hosp<-read.csv("P:/STAT-6494-Data Management in SAS and R/data/hosp.csv",

header=T, stringsAsFactors=FALSE)

AdmitDate<-as.Date(hosp$AdmitDate,"%m/%d/%Y")

day<-weekdays(AdmitDate)

month<-months(AdmitDate)

year<-format(AdmitDate,'%Y')

table(day)

table(month)

table(year)

1. **SAS:** Using the Hosp data set, compute the number of months from the admission date and December 31, 2007 (call it MonthsDec). Also, compute the number of months from the admission date to today’s date (call it MonthsToday). Use a date interval function to solve this problem. List the first 20 observations for your solution.

**data** freq;

set freq;

MonthsDec=intck('month',AdmitDate,**'31Dec2007'd**);

MonthsToday=intck('month',AdmitDate,today());

**run**;

**proc** **print** data=freq (obs=**20**);

var MonthsDec MonthsToday;

**run**;

1. **R:** Repeat question 3, compute the number of months from the admission date and December 31, 2007 (call it MonthsDec). Also, compute the number of months from the admission date to today’s date (call it MonthsToday). List the first 20 observations for your solution.

enddate<-as.Date("2007-12-31")

library(lubridate)

MonthsDec<-interval(ymd(AdmitDate),ymd(enddate)) %/% months(1)

MonthsDec[1:20]

MonthsToday<-interval(ymd(AdmitDate),ymd(Sys.Date())) %/% months(1)

MonthsToday[1:20]

1. **SAS:** Using the Medical data set, compute frequencies for the days of the week for the date of the visit (VisitDate). Supply a format for the days of the week and months of the year.

**data** freq;

set hw5.medical(keep=VisitDate);

Day=weekday(VisitDate);

Month=month(VisitDate);

**run**;

**proc** **format** ;

value days **1**='Sun' **2**='Mon' **3**='Tue' **4**='Wed' **5**='Thu' **6**='Fri' **7**='Sat';

value months **1**='Jan' **2**='Feb' **3**='Mar' **4**='Apr' **5**='May' **6**='Jun'

**7**='Jul' **8**='Aug' **9**='Sep' **10**='Oct' **11**='Nov' **12**='Dec';

**run**;

**proc** **freq** data=freq ;

tables Day / nocum nopercent;

format Day days. Month months.;

**run**;

1. **SAS:** The data set Medical contains a variable called VisitDate. Create a temporary SAS data set (Interval) with the variables in Medical plus a new variable (Quarter), representing the number of quarters from January 1, 2006.

**data** Interval;

set hw5.medical(keep=VisitDate);

Quarter=intck('qtr',**'01Jan2006'd**,VisitDate);

**run**;

**proc** **print** data=interval;

**run**;

1. **SAS:** You want to see each patient in the Medical data set on the same day of the week 5 weeks after they visited the clinic (the variable name is VisDate). Provide a listing of the patient number (Patno), the visit date, and the date for the return visit. And also print out today’s date.

**data** followup;

set hw5.medical;

return=intnx('week',VisitDate,**5**,'sameday');

today=today();

format return mmddyy10. today mmddyy10.;

**run**;

**proc** **print** data=followup;

var patno visitdate return today;

**run**;

1. **R:** Repeat question 7 in R.

**PROC** **EXPORT** DATA= hw5.medical

OUTFILE= "P:\STAT-6494-Data Management in SAS and R\data\medical.csv"

DBMS=CSV REPLACE;

PUTNAMES=YES;

**RUN**;

medical<-read.csv("P:/STAT-6494-Data Management in SAS and R/data/medical.csv",

header=T, stringsAsFactors=FALSE)

medical$VisitDate<-as.Date(medical$VisitDate,"%m/%d/%Y")

medical$Return<-medical$VisitDate+35

medical$Today<-Sys.Date()

medical[c("Patno","VisitDate","Return","Today")]

1. **R:** Friday the 13th is considered by some as an unlucky day. Write a program in R which counts how many times the 13th of a month coincides with a Friday. Do it for 2010-2018. Which year is the “unluckiest”?

start<-as.Date("01/01/2010","%m/%d/%Y")

end<-as.Date("12/31/2018","%m/%d/%Y")

startn<-as.numeric(start)

endn<-as.numeric(end)

n=0

for(i in startn:endn){

j<-as.Date(i,origin="1970-01-01")

if (weekdays(j)=="Friday" & format(j,'%d')=="13"){

n=n+1

cat(format(j,"%Y-%m-%d"),'\n')

}

}

n

year<-rep("NA",n)

k=1

for(i in startn:endn){

j<-as.Date(i,origin="1970-01-01")

if (weekdays(j)=="Friday" & format(j,'%d')=="13"){

year[k]<-format(j,'%Y')

k=k+1

}

}

table(year)

Answer:16 times. Year 2012 and 2015 are the “unluckiest”.

1. **SAS**: Using the SAS data set Hosp, create a temporary SAS data set called Monday2002, consisting of observations from Hosp where the admission data (AdmitDate) falls on a Monday and the year is 2002. Include in this new data set a variable called Age, computed as the person’s age as of the admission date, rounded to the nearest year.

**data** Monday2002;

set hw5.hosp;

where (weekday(AdmitDate)=**2** and year(AdmitDate)=**2002**);

Age=year(AdmitDate)-year(DOB);

**run**;

1. **SAS**: Using the SAS data set Bicycles, create two temporary SAS data sets as follows: Mountain\_USA consists of all observations from Bicycles where Country is USA and Model is Mountain Bike. Road\_France consists of all observations from Bicycles where Country is France and Model is Road Bike. Print these two data sets.

**data** Mountain\_USA;

set hw5.bicycles;

where (Country='USA' and Model='Mountain Bike');

**run**;

**data** Road\_France;

set hw5.bicycles;

where (Country='France' and Model='Road Bike');

**run**;

1. **SAS**: Run the program here to create a SAS data set called Markup:

data markup;

input manuf : $10. Markup;

datalines;

Cannondale 1.05

Trek 1.07

;

run;

Combine this data set with the Bicycles data set so that each observation in the Bicycles data set now has a markup value of 1.05 or 1.07, depending on whether the bicycle is made by Cannondale or Trek. In this new data set (call it Markup\_Prices), create a new variable (NewTotal) computed as TotalSales times Markup.

**data** bicycle;

set hw5.bicycles;

**run**;

**proc** **sort** data=bicycle;

by manuf;

**run**;

**data** Markup\_Prices;

merge bicycle markup;

by manuf;

**run**;

1. **SAS**: Using the Purchase and Inventory data sets, provide a list of all Models (and the Price) that were not purchased.

**proc** **sort** data=hw5.Inventory;

by model;

**run**;

**proc** **sort** data=hw5.Purchase;

by model;

**run**;

**data** notpur;

merge hw5.Inventory

hw5.Purchase(in=inPurchase);

by Model;

if inPurchase=**0** then output notpur;

keep model price;

**run**;

**proc** **print** data=notpur;

**run**;

1. **SAS**: Merge the two SAS data sets, Demographic and Survey1, based on an identifier. In Demographic, this identifier is called ID; in Survey1, the identifier is called Subj. Both are character variables.

**proc** **sort** data=hw5.Demographic ;

by ID;

**run**;

**proc** **sort** data=hw5.Survey1;

by Subj;

**run**;

**data** new;

merge hw5.Demographic

hw5.Survey1(rename=(Subj=ID));

by ID;

**run**;

1. **R**: Repeat question 14 in R.

**PROC** **EXPORT** DATA= HW5.Demographic

OUTFILE= "P:\STAT-6494-Data Management in SAS and R\data\Demographic.csv"

DBMS=CSV REPLACE;

PUTNAMES=YES;

**RUN**;

**PROC** **EXPORT** DATA= HW5.Survey1

OUTFILE= "P:\STAT-6494-Data Management in SAS and R\data\Survey1.csv"

DBMS=CSV REPLACE;

PUTNAMES=YES;

**RUN**;

demographic<-read.csv("P:/STAT-6494-Data Management in SAS and R/data/demographic.csv",

header=T, stringsAsFactors=FALSE)

survey1<-read.csv("P:/STAT-6494-Data Management in SAS and R/data/survey1.csv",

header=T, stringsAsFactors=FALSE)

new<-merge(demographic,survey1,by.x="ID",by.y="Subj",all=T)

new

1. **SAS**: Data set Inventory contains two variables: Model (an 8-byte character variable) and Price (a numeric value). The price of Model M567 has changed to 25.95 and the price of Model X999 has changed to 35.99. Create a temporary SAS data set (call it NewPrices) by update the prices in the Inventory data set.

**proc** **sort** data=hw5.Inventory;

by model;

**run**;

**data** new;

input Model $8. Price;

datalines;

M567 25.95

X999 35.99

;

**run**;

**proc** **sort** data=new;

by model;

**run**;

**data** NewPrices;

update hw5.Inventory new;

by model;

**run**;