STAT-506: SAS Project

Description: The data worked with consists of my daily and workout health history captured from my wearable.

**Topics: 6,8,3,12,15,21,18,20,25,27,14,9**

/\* 6. Read Excel spreadsheets into SAS using:

a) LIBNAME statement to read multiple pages of a excel file

b) Proc IMPORT (generate the code)

c) Proc EXPORT (generate the code)

d) LIBNAME statement with CLEAR option

\*/

libname health excel "W:\My Documents\My SAS Files\Microsoft\_Health\_20150418\_20150831.xls"; /\* 6a \*/

PROC IMPORT OUT= WORK.Activity /\* 6b \*/

DATAFILE= "W:\My Documents\My SAS Files\Microsoft\_Health\_201

50418\_20150831.xls"

DBMS=EXCEL REPLACE;

RANGE="'Activity summary$'";

GETNAMES=YES;

MIXED=NO;

SCANTEXT=YES;

USEDATE=YES;

SCANTIME=YES;

RUN;

PROC EXPORT DATA= WORK.ACTIVITY /\* 6c \*/

OUTFILE= "W:\My Documents\SAS Project\exported.xls"

DBMS=EXCEL REPLACE;

SHEET="activity";

RUN;

libname health clear; /\* 6d \*/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

/\* 8. Use the following to read in a data set that has FORMATTED data:

a) INFILE

b) INPUT using at least 4 INFORMATs

c) FORMAT statement to re-format at least 2 variables

d) Proc PRINT

\*/

libname health "W:\My Documents\SAS Project";

data health.activity; /\* 8a \*/

length Event\_Type $10. Cardio\_Benefit $12. Sleep\_Restoration $8. Sleep\_Auto\_Detect $6. ;

infile "W:\My Documents\SAS Project\Activity.csv" dlm=',' dsd missover;

input Input\_Date :mmddyy9. Start\_Time :time. Event\_Type Duration\_Seconds Seconds\_Paused Calories\_Burned

Calories\_Burned\_Carbs Calories\_Burned\_Fats HR\_Lowest HR\_Peak HR\_Average

UV\_Exposure\_Minutes Total\_Kilometers\_Moved Cardio\_Benefit Minutes\_Under\_50\_HR

Minutes\_In\_HRZ\_Very\_Light\_50\_60 Minutes\_In\_HRZ\_Light\_60\_70

Minutes\_In\_HRZ\_Moderate\_70\_80 Minutes\_In\_HRZ\_Hard\_80\_90 Minutes\_In\_HRZ\_Very\_Hard\_90\_Plus

HR\_Finish HR\_Recovery\_Rate\_1\_Min HR\_Recovery\_Rate\_2\_Min Recovery\_Time\_Seconds Bike\_Average\_KPH

Bike\_Max\_KPH Elevation\_Highest\_Meters Elevation\_Lowest\_Meters Elevation\_Gain\_Meters

Elevation\_Loss\_Meters Wake\_Up\_Time :MDYAMPM20.2 Seconds\_Awake Seconds\_Asleep\_Total Seconds\_Asleep\_Restful

Seconds\_Asleep\_Light Wake\_Ups Seconds\_to\_Fall\_Asleep Sleep\_Efficiency Sleep\_Restoration

Sleep\_HR\_Resting Sleep\_Auto\_Detect GW\_Plan\_Name GW\_Reps\_Performed GW\_Rounds\_Performed

Golf\_Course\_Name Golf\_Course\_Par Golf\_Total\_Score Golf\_Par\_or\_Better Golf\_Pace\_of\_Play\_Minutes

Golf\_Longest\_Drive\_Yards;

run;

data health.activity; /\* 8b \*/

length Event\_Type $10. Cardio\_Benefit $12. Sleep\_Restoration $8. Sleep\_Auto\_Detect $6. ;

infile "W:\My Documents\SAS Project\Activity.csv" dlm=',' dsd missover;

input Input\_Date :mmddyy9. Start\_Time :time. Event\_Type :$10. Duration\_Seconds :7. Seconds\_Paused Calories\_Burned

Calories\_Burned\_Carbs Calories\_Burned\_Fats HR\_Lowest HR\_Peak HR\_Average

UV\_Exposure\_Minutes Total\_Kilometers\_Moved Cardio\_Benefit Minutes\_Under\_50\_HR

Minutes\_In\_HRZ\_Very\_Light\_50\_60 Minutes\_In\_HRZ\_Light\_60\_70

Minutes\_In\_HRZ\_Moderate\_70\_80 Minutes\_In\_HRZ\_Hard\_80\_90 Minutes\_In\_HRZ\_Very\_Hard\_90\_Plus

HR\_Finish HR\_Recovery\_Rate\_1\_Min HR\_Recovery\_Rate\_2\_Min Recovery\_Time\_Seconds Bike\_Average\_KPH

Bike\_Max\_KPH Elevation\_Highest\_Meters Elevation\_Lowest\_Meters Elevation\_Gain\_Meters

Elevation\_Loss\_Meters Wake\_Up\_Time :MDYAMPM20.2 Seconds\_Awake Seconds\_Asleep\_Total Seconds\_Asleep\_Restful

Seconds\_Asleep\_Light Wake\_Ups Seconds\_to\_Fall\_Asleep Sleep\_Efficiency Sleep\_Restoration

Sleep\_HR\_Resting Sleep\_Auto\_Detect GW\_Plan\_Name GW\_Reps\_Performed GW\_Rounds\_Performed

Golf\_Course\_Name Golf\_Course\_Par Golf\_Total\_Score Golf\_Par\_or\_Better Golf\_Pace\_of\_Play\_Minutes

Golf\_Longest\_Drive\_Yards;

run;

data health.activity; /\* 8c \*/

length Event\_Type $10. Cardio\_Benefit $12. Sleep\_Restoration $8. Sleep\_Auto\_Detect $6. ;

infile "W:\My Documents\SAS Project\Activity.csv" dlm=',' dsd missover;

input Input\_Date :mmddyy9. Start\_Time :time. Event\_Type :$10. Duration\_Seconds :7. Seconds\_Paused Calories\_Burned

Calories\_Burned\_Carbs Calories\_Burned\_Fats HR\_Lowest HR\_Peak HR\_Average

UV\_Exposure\_Minutes Total\_Kilometers\_Moved Cardio\_Benefit Minutes\_Under\_50\_HR

Minutes\_In\_HRZ\_Very\_Light\_50\_60 Minutes\_In\_HRZ\_Light\_60\_70

Minutes\_In\_HRZ\_Moderate\_70\_80 Minutes\_In\_HRZ\_Hard\_80\_90 Minutes\_In\_HRZ\_Very\_Hard\_90\_Plus

HR\_Finish HR\_Recovery\_Rate\_1\_Min HR\_Recovery\_Rate\_2\_Min Recovery\_Time\_Seconds Bike\_Average\_KPH

Bike\_Max\_KPH Elevation\_Highest\_Meters Elevation\_Lowest\_Meters Elevation\_Gain\_Meters

Elevation\_Loss\_Meters Wake\_Up\_Time :MDYAMPM20.2 Seconds\_Awake Seconds\_Asleep\_Total Seconds\_Asleep\_Restful

Seconds\_Asleep\_Light Wake\_Ups Seconds\_to\_Fall\_Asleep Sleep\_Efficiency Sleep\_Restoration

Sleep\_HR\_Resting Sleep\_Auto\_Detect GW\_Plan\_Name GW\_Reps\_Performed GW\_Rounds\_Performed

Golf\_Course\_Name Golf\_Course\_Par Golf\_Total\_Score Golf\_Par\_or\_Better Golf\_Pace\_of\_Play\_Minutes

Golf\_Longest\_Drive\_Yards;

Sleep\_Efficiency = Sleep\_Efficiency/100;

format Input\_Date date9. Sleep\_Efficiency percent8.2 ;

run;

proc contents data=health.activity;

run;

proc print data=health.activity; /\* 8d \*/

format Start\_Time hhmm8.2 Wake\_Up\_Time MDYAMPM20.2;

run;

/\* 3. Print SAS data set(s) using:

a) Proc PRINT

b) SPLIT= option

c) TITLE

d) FOOTNOTE

\*/

PROC IMPORT OUT= HEALTH.Daily

DATAFILE= "W:\My Documents\SAS Project\Microsoft\_Health\_2015

0418\_20150831.xls"

DBMS=EXCEL REPLACE;

RANGE="'Daily summary$'";

GETNAMES=YES;

MIXED=NO;

SCANTEXT=YES;

USEDATE=YES;

SCANTIME=YES;

RUN;

proc print data=health.Daily ; /\* 3a \*/

run;

proc print data=health.Daily Split="\_"; /\* 3b \*/

run;

title "Sneha's Daily Activity"; /\* 3c \*/

proc print data=health.Daily Split="\_";

run;

title ;

footnote "Data from Sneha's Health Band"; /\* 3d \*/

proc print data=health.Daily Split="\_";

run;

/\* 12. Merge two or more data sets—at least two different merges must be done

a) One must have non-matches

(1) Use IN= options with a IF conditional statement

(2) Direct output to 2 different datasets (matches and non-matches)

b) One must use the options below

(1) RENAME=

(2) DROP=/KEEP= options

\*/

/\* both datasets are already sorted by Input\_Date \*/

data matches nonmatches(drop=Calories\_Burned HR\_Peak) ; /\* 12a \*/

merge health.activity (in=a) health.daily (rename= (Date=Input\_Date)in=d) ;

by Input\_Date;

if (a=1 and d=1 ) then output matches;

else output nonmatches;

keep Input\_Date Calories\_Burned HR\_Peak;

run;

data merged (keep= Input\_Date Duration\_Seconds Calories\_Burned Total\_Kilometers\_Moved ); /\* 12b \*/

merge health.activity (rename=(Seconds\_Paused=Pause\_Time\_Sec) in=a) health.daily (rename= (Date=Input\_Date)in=d) ;

by Input\_Date;

run;

/\* 15. Demonstrate the following

a) Use of RETAIN

b) Sum Statement (variable + expression;)

c) Use of FIRST. and LAST.

d) Proc SORT

\*/

data total; /\* 15a \*/

set health.activity;

retain Total\_Calories\_Burned 0;

Total\_Calories\_Burned = Total\_Calories\_Burned + Calories\_Burned;

keep Input\_Date Calories\_Burned Total\_Calories\_Burned;

run;

data total\_to\_date; /\* 15b \*/

set health.activity;

Total\_Calories\_Burned + Calories\_Burned;

keep Input\_Date Calories\_Burned Total\_Calories\_Burned;

run;

proc sort data=health.activity /\* 15d \*/

out=activity;

by Event\_Type Cardio\_Benefit;

run;

data aggregate; /\* 15c \*/

set activity;

by Event\_Type Cardio\_Benefit;

if first.Event\_Type then Calorie\_Count = 0;

else Calorie\_Count + Calories\_Burned;

if last.Cardio\_Benefit then output;

run;

/\* 21. Converting Variable Type

a) Character to Numeric

b) Numeric to Character

c) Demonstrate automatic conversion (need log here)

d) Proc CONTENTS showing the original variable type

\*/

data health.conversion; /\* 21b \*/

set health.daily (rename=(Steps=numSteps Total\_Kilometers\_Moved=numTLM));

Steps = put(numSteps,5.);

Total\_Kilometers\_Moved = put(numTLM, 10.2);

keep Date Steps Total\_Kilometers\_Moved;

run;

data health.toNum (drop=charSteps charTLM); /\* 21a \*/

set health.conversion (rename=(Steps=charSteps Total\_Kilometers\_Moved=charTLM));

Steps=input(charSteps,5.);

Total\_Kilometers\_Moved=input(charTLM,10.2);

run;

data health.autoconversion; /\* 21c \*/

set health.conversion;

Meters\_Moved = Total\_Kilometers\_Moved \* 1000;

run;

/\* Log \*/

/\*

14

15 data health.autoconversion; /\* 21c \*/

16 set health.conversion;

17 Meters\_Moved = Total\_Kilometers\_Moved \* 1000;

18 run;

NOTE: Character values have been converted to numeric values at the places given by:

(Line):(Column).

17:20

NOTE: There were 136 observations read from the data set HEALTH.CONVERSION.

NOTE: The data set HEALTH.AUTOCONVERSION has 136 observations and 4 variables.

NOTE: DATA statement used (Total process time):

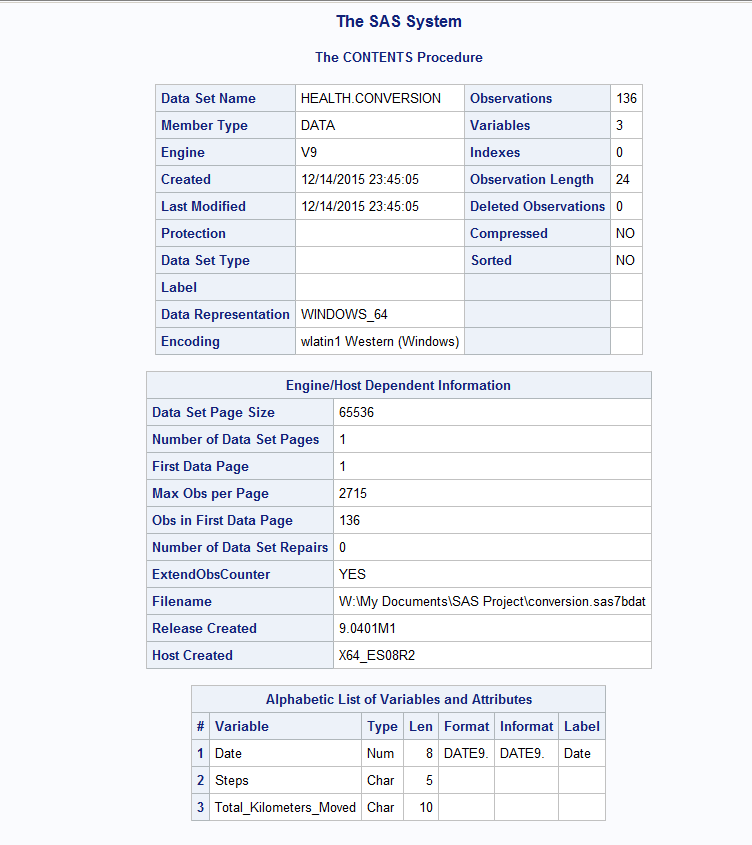
real time 0.22 seconds

cpu time 0.00 seconds

\*/

proc contents data=health.conversion; /\* 21d \*/

run;



/\* 18. Use the following fuctions

a) SUBSTR

b) LENGTH

c) SCAN

d) PROPCASE

e) One of (RIGHT, LEFT, UPCASE, LOWCASE, CHAR)

f) CATX (or CAT, CATS, CATT)

\*/

data health.functions;

set health.activity;

length Status $40.;

if (Cardio\_Benefit = ' ') then Status = "Lazy Sneha did not workout today!";

else do;

select (substr(Cardio\_Benefit,1,1)); /\* 18a \*/

when ('M') Status = "Sneha worked out pretty well";

when ('L') Status = "Sneha can push herself to do better";

otherwise Status = "Error";

end;

end;

keep Input\_Date Cardio\_Benefit Status;

run;

data health.functions;

set health.activity;

length Emoticon $11.;

if (length(Event\_Type)>5 or length(Event\_Type)<4) then Emoticon = "Thumbs Up"; /\* 18b \*/

else Emoticon = "Thumbs Down";

keep Event\_Type Emoticon;

run;

data health.functions2;

set health.functions;

UporDown = scan(Emoticon,2); /\* 18c \*/

Keep Event\_Type UporDown;

run;

data health.functions3;

set health.functions;

low = lowcase(Event\_Type); /\* 18d \*/

proper = propcase(low); /\* 18e \*/

Keep Event\_Type low proper;

run;

data health.functions4;

set health.activity;

Description = catx("-",Event\_Type,Cardio\_Benefit); /\* 18f \*/

keep Input\_Date Description;

run;

/\* 20. Use the following functions

a) ROUND

b) One of (CEIL, FLOOR, INT)

c) One of (SUM, MEAN)

d) One of (MIN, MAX, N)

e) One of (NMISS, CMISS)

\*/

data health.math;

set health.daily;

Kilometers\_Moved = round(Total\_Kilometers\_Moved,0.1); /\* 20a \*/

keep Date Kilometers\_Moved;

run;

data health.math2;

set health.daily;

Max\_Kilometers\_Moved = ceil(Total\_Kilometers\_Moved); /\* 20b \*/

keep Date Max\_Kilometers\_Moved;

run;

data health.math3;

set health.activity;

total\_calories\_burned = sum(Calories\_Burned\_Carbs,Calories\_Burned\_Fats); /\* 20c \*/

average\_calories\_burned = mean(Calories\_Burned\_Carbs,Calories\_Burned\_Fats);

keep Input\_Date total\_calories\_burned average\_calories\_burned;

run;

data health.math4;

set health.activity;

min\_heart\_rate= min(HR\_Lowest,HR\_Average,HR\_Finish); /\* 20d \*/

keep Input\_Date min\_heart\_rate;

run;

data health.math5;

set health.activity;

number\_of\_missing\_hr = nmiss (HR\_Lowest,HR\_peak,HR\_Average,HR\_Finish); /\* 20e \*/

keep Input\_Date number\_of\_missing\_hr;

run;

/\* 25. Demonstrate the following:

a) OPTIONS SYMBOLGEN

b) %put

c) %let: Create and use at least three user macro variables

d) Use at least three automatic macro Variables

e) %SYMDEL

\*/

options symbolgen; /\* 25a \*/

%put Health Data Captured from MS Band ; /\* 25b \*/

%let a = Run; /\* 25c \*/

%let b = 20000;

%let c = Good;

title "&a Activity";

proc print data=health.activity;

where Event\_Type contains "&a";

run;

title;

proc print data=health.activity;

where Duration\_Seconds > &b ;

run;

proc print data=health.activity;

where Sleep\_Restoration = "&c" ;

run;

/\* 25d \*/

%put Date is &sysdate ;

%put Last dataset &SYSLAST;

%put Who is working on SAS? &sysuserid;

%symdel a b c; /\* 25e \*/

/\* 27. Write and call at least two macro programs (different from those in another topic) using:

a) %MACRO

b) %MEND

c) OPTIONS MPRINT;

d) OPTIONS MCOMPILENOTE=ALL

\*/

options mcompilenote = all; /\* 27d \*/

%macro daily(day); /\* 27a \*/

proc print data=health.daily;

where date = &day;

run;

%mend; /\* 27b \*/

options mprint; /\* 27c \*/

%daily('29APR2015'd);

%macro daily(day); /\* 27a \*/

proc print data=health.daily;

where date = &day;

run;

%mend; /\* 27b \*/

%macro sleep\_efficiency(wake\_ups); /\* 27a \*/

proc print data=health.activity;

where Wake\_Ups > &wake\_ups ;

var Input\_Date Wake\_Ups Seconds\_Awake Seconds\_Asleep\_Total Seconds\_Asleep\_Restful Seconds\_Asleep\_Light Sleep\_Efficiency;

run;

%mend; /\* 27b \*/

%sleep\_efficiency(3);

/\* 14. Demonstrate the following

a) SELECT group

b) Explicit output to multiple datasets

c) DROP=/KEEP= options in data statement

d) FIRSTOBS= and OBS=

\*/

data Sleep (keep= Wake\_Up\_Time Seconds\_Awake Seconds\_Asleep\_Total

Seconds\_Asleep\_Restful Seconds\_Asleep\_Light

Wake\_Ups Seconds\_to\_Fall\_Asleep Sleep\_Efficiency

Sleep\_Restoration Sleep\_HR\_Resting Sleep\_Auto\_Detect) /\* 14c \*/

Run (keep= Duration\_Seconds Calories\_Burned Total\_Kilometers\_Moved )

Exercise (keep= Start\_Time Event\_Type Duration\_Seconds

Seconds\_Paused Calories\_Burned Calories\_Burned\_Carbs Calories\_Burned\_Fats

HR\_Lowest HR\_Peak HR\_Average UV\_Exposure\_Minutes Cardio\_Benefit)

Errored;

set health.activity;

select(Event\_Type); /\* 14 a \*/

when ('Exercise') output Exercise; /\* 14b \*/

when ('Sleep') output Sleep;

when ('Run') output Run;

otherwise output Errored;

end;

run;

title;

proc print data=work.Exercise (firstobs=5 obs=20); /\* 14d \*/

run;

/\*9. Demonstrate the following

a) Creating new variables

b) Usage of two of (YEAR, QTR, MONTH, DAY, WEEKDAY)

c) TODAY()

d) MDY()

\*/

data health.dateuse;

set health.activity end=final;

Birthday = mdy(03,30,1991); /\* 9d \*/

call symputx('bdy',put(Birthday,mmddyy10.));

Year = year(Input\_date); /\* 9a \*/

Month = month(Input\_date); /\* 9b \*/

how\_long\_ago = today() - input\_date; /\* 9c \*/

total\_calories\_burned + calories\_burned;

if final then call symputx('tcb',total\_calories\_burned);

keep Input\_Date Year Month how\_long\_ago;

run;

footnote "Total Caloried Burned while working out : &tcb";

footnote2 "Sneha has her birthday on : &bdy";

proc print data=health.dateuse;

run;