

## A. Aggregating Time Diary Data

The examples presented below provide a basic framework for aggregating data, and require some familiarity with SAS or SPSS programming syntax.

### SAS Example of aggregating activity data for an individual

\*Create SAS data file from inline data records;

```
DATA SMPL;
INFILE CARDS ;
INPUT
CASE_ID $1-13 DAY 14-14 COLA 15-17 COLB 18-22 COLC 23-27 COLD 28-28
COLF 29-30 COLG_A 31-31 COLG_B 32-32 COLG_C 33-33 COLG_D 34-34 COLG_E 35-35
COLG_F 36-36 COLG_G 37-37 COLG_H 38-38 COLG_I 39-39 COLG_J 40-40 COLG_K
41-41 COLH_A 42-42 COLH_B 43-43 COLH_C 44-44 COLH_D 45-45 COLH_E 46-46
COLH_F 47-47 COLH_G 48-48 COLH_H 49-49 COLH_I 50-50 COLH_J 51-51 COLH_K
52-52 COLJ 53-55 WDAYWEND 56-56 DURATION 57-61;
IF COLD IN(9,0) THEN COLD=.;
IF COLF IN(99,0) THEN COLF=.;
IF COLG_A IN(8,9) THEN COLG_A=.;
IF COLG_B IN(8,9) THEN COLG_B=.;
IF COLG_C IN(8,9) THEN COLG_C=.;
IF COLG_D IN(8,9) THEN COLG_D=.;
IF COLG_E IN(8,9) THEN COLG_E=.;
IF COLG_F IN(8,9) THEN COLG_F=.;
IF COLG_G IN(8,9) THEN COLG_G=.;
IF COLG_H IN(8,9) THEN COLG_H=.;
IF COLG_I IN(8,9) THEN COLG_I=.;
IF COLG_J IN(8,9) THEN COLG_J=.;
IF COLG_K IN(8,9) THEN COLG_K=.;
IF COLH_A IN(8,9) THEN COLH_A=.;
IF COLH_B IN(8,9) THEN COLH_B=.;
IF COLH_C IN(8,9) THEN COLH_C=.;
IF COLH_D IN(8,9) THEN COLH_D=.;
IF COLH_E IN(8,9) THEN COLH_E=.;
IF COLH_F IN(8,9) THEN COLH_F=.;
IF COLH_G IN(8,9) THEN COLH_G=.;
IF COLH_H IN(8,9) THEN COLH_H=.;
IF COLH_I IN(8,9) THEN COLH_I=.;
IF COLH_J IN(8,9) THEN COLH_J=.;
IF COLH_K IN(8,9) THEN COLH_K=.;
FORMAT
COLA COLD COLF COLJ DAY WDAYWEND
COLG_A COLG_B COLG_C COLG_D COLG_E COLG_F COLG_G COLG_H COLG_I COLG_J COLG_K
COLH_A COLH_B COLH_C COLH_D COLH_E COLH_F COLH_G COLH_H COLH_I COLH_J COLH_K F3.
COLB COLC DURATION TIME5. ;
CARDS;
0004-001-04-12459 025200010 125200
0004-001-04-124072520025500010 1 300
0004-001-04-124092550025800010 1 300
0004-001-04-124092580026100010 1 300
0004-001-04-12439261002700001010000000000000000000000100 01 900
0004-001-04-124092700027300010 1 300
0004-001-04-1259727300276000200000000010010000000000 01 300
0004-001-04-12509276004320008000000000000100000000001 0115600
0004-001-04-12509432005490008000000000000100000000001 0111700
0004-001-04-1259954900567000200000000000100000000009621 1800
0004-001-04-1248756700594000401000000000000000000100 01 2700
```

```

0004-001-04-1249959400600000201000000000000000000100 01 600
0004-001-04-1288960000612000101000000000000000000100 01 1200
0004-001-04-12801612006300001000000000100100000000009631 1800
0004-001-04-1248863000648000100000000010000000000009631 1800
0004-001-04-1279964800654000200000000010010000000000 01 600
0004-001-04-1276965400682200600000000011100000000014391 2820
0004-001-04-1276968220711000600000000011100000000018771 2880
0004-001-04-1276971100738000600000000011100000000019621 2700
0004-001-04-1279973800744000200000000010010000000000 01 600
0004-001-04-124087440075000010 1 600
0004-001-04-124597500086400010 111400
0004-001-04-17459 027000045 027000
0004-001-04-17877270003240004501000100000000100000009620 5400
0004-001-04-174483240033300045100000000001010100000 00 900
0004-001-04-1791933300405001450101010000000010000000 00 7200
0004-001-04-17876405004320004500010000000011000000008050 2700
0004-001-04-1748143200441000 0000000000000000000000 00 900
0004-001-04-174484410045000045011000000000010000000 00 900
0004-001-04-1749945000462000200100000000010000000009620 1200
0004-001-04-179894620048600010100000000000000000100 00 2400
0004-001-04-1781748600522000200100000000010000000009620 3600
0004-001-04-1786652200567000400000000100001000000001 00 4500
0004-001-04-1789956700573000200100000000010000000009620 600
0004-001-04-1787757300604200101000000000010000001009620 3120
0004-001-04-1793960420635400101000000000010000001009620 3120
0004-001-04-179626354066600010100000000001000000100 00 3060
0004-001-04-1787666600675000100000000011000000000100 00 900
0004-001-04-174396750068400010100000000000000000100 00 900
0004-001-04-1791968400720002100000000010010000000000 00 3600
0004-001-04-174087200073800010 0 1800
0004-001-04-178777380075600010100000000000000000100 00 1800
0004-001-04-174597560086400010 010800
;
run ;

/*
SAS solution-1: PROC SORT, PROC SUMMARY, and DATA STEP
1. Sort data set by key variables,
2. Use PROC SUMMARY to write separate files for weekday and weekend
time by case_id and activity code (cola),
3. Merge data file from step(2).

*/
PROC SORT DATA=SMPL NODUP;
BY CASE_ID WDAYWEND COLA;
run;

proc summary data = smpl(where=(wdaywend= 1)); by case_id
cola ; var duration ; idday; output out=weekday(drop= __type__freq_rename=(day=weekday))
sum(duration)=wdaytime ;
run ;

proc summary data = smpl(where=(wdaywend= 0)); by case_id
cola ; var duration ; idday; output out=weekend(drop= __type__freq_rename=(day=weekend))
sum(duration)=wendtime ;
run ;

data act_time ;
merge weekday weekend ;

```

```

by case_id cola ;
rename cola=act_code ;
run ;

```

```

* SAS-SQL Solution (more elegant);
proc sql ;
create table weekday as
select case_id, cola, day as weekday, sum(duration) as wdaytime format=time5.
from smpl
where day between 1 and 5
group by case_id, day, cola
;
create table weekend as
select case_id, cola, day as weekend, sum(duration) as wendtime format=time5.
from smpl
where day between 6 and 7
group by case_id, day, cola
;
create table act_time as
select coalesce(a.case_id, b.case_id) as case_id,
coalesce(a.cola, b.cola) as act_code,
a.wdaytime, a.weekday, b.wendtime, b.weekend
from weekday a full join weekend b
on a.case_id = b.case_id and a.cola = b.cola
;
quit ;

```

## B. SPSS Example of aggregating activity data for an individual

\* Create SPSS system data file from inline data .

```

DATA LIST /
CASE_ID 1-13(A) DAY 14-14 COLA 15-17 COLB 18-22 COLC 23-27 COLD 28-28
COLF 29-30 COLG_A 31-31 COLG_B 32-32 COLG_C 33-33 COLG_D 34-34 COLG_E 35-35
COLG_F 36-36 COLG_G 37-37 COLG_H 38-38 COLG_I 39-39 COLG_J 40-40 COLG_K
41-41 COLH_A 42-42 COLH_B 43-43 COLH_C 44-44 COLH_D 45-45 COLH_E 46-46
COLH_F 47-47 COLH_G 48-48 COLH_H 49-49 COLH_I 50-50 COLH_J 51-51 COLH_K
52-52 COLJ 53-55 WDAYWEND 56-56 DURATION 57-61 .
BEGIN DATA
0004-001-04-12459 025200010 125200
0004-001-04-124072520025500010 1 300
0004-001-04-124092550025800010 1 300
0004-001-04-124092580026100010 1 300
0004-001-04-124392610027000010100000000000000000000000000100 01 900
0004-001-04-124092700027300010 1 300
0004-001-04-125972730027600020000000000100100000000000 01 300
0004-001-04-12509276004320008000000000000100000000001 0115600
0004-001-04-1250943200549000800000000000100000000001 0111700
0004-001-04-12599549005670002000000000001000000000009621 1800
0004-001-04-124875670059400040100000000000000000000100 01 2700
0004-001-04-124995940060000020100000000000000000000100 01 600
0004-001-04-128896000061200010100000000000000000000100 01 1200
0004-001-04-1280161200630000100000000001001000000000009631 1800
0004-001-04-12488630006480001000000000010000000000009631 1800
0004-001-04-127996480065400020000000000100100000000000 01 600
0004-001-04-127696540068220060000000000111000000000014391 2820
0004-001-04-127696822071100060000000000111000000000018771 2880
0004-001-04-127697110073800060000000000111000000000019621 2700

```

```

0004-001-04-1279973800744000200000000010010000000000 01 600
0004-001-04-124087440075000010 1 600
0004-001-04-124597500086400010 111400
0004-001-04-17459 027000045 027000
0004-001-04-17877270003240004501000100000000100000009620 5400
0004-001-04-1744832400333000451000000000001010100000 00 900
0004-001-04-1791933300405001450101010000000010000000 00 7200
0004-001-04-17876405004320004500010000000011000000008050 2700
0004-001-04-1748143200441000 0000000000000000000000 00 900
0004-001-04-1744844100450000450110000000000010000000 00 900
0004-001-04-1749945000462000200100000000010000000009620 1200
0004-001-04-179894620048600010100000000000000000100 00 2400
0004-001-04-1781748600522000200100000000010000000009620 3600
0004-001-04-1786652200567000400000000100001000000001 00 4500
0004-001-04-1789956700573000200100000000010000000009620 600
0004-001-04-17877573006042001010000000000010000001009620 3120
0004-001-04-17939604206354001010000000000010000001009620 3120
0004-001-04-1796263540666000101000000000001000000100 00 3060
0004-001-04-1787666600675000100000000011000000000100 00 900
0004-001-04-174396750068400010100000000000000000100 00 900
0004-001-04-1791968400720002100000000010010000000000 00 3600
0004-001-04-174087200073800010 0 1800
0004-001-04-1787773800756000101000000000000000000100 00 1800
0004-001-04-174597560086400010 010800
END DATA .

```

```

RECODE COLD (9,0=SYSMIS) /
COLF (0,99=SYSMIS) /
COLG_A TO COLH_K (8,9=SYSMIS) .

```

```

FORMATS COLB COLC DURATION (TIME) .

```

```

SORT CASES BY CASE_ID DAY COLA .
COMPUTE WEEKDAY = (WDAYWEND=1) .
FILTER BY WEEKDAY . * select kids with weekday time only .
EXECUTE .

```

```

* Write activity aggregate data file .
AGGREGATE
/OUTFILE='[path]\WDAYTIME.sav'
/BREAK=case_id cola
/WEEKDAY= FIRST(day) /WDAYTIME= SUM(duration).

```

```

USE ALL. * reset cases to full file .
COMPUTE WEEKEND = (WDAYWEND=0) .
FILTER BY WEEKEND . * select kids with weekend time only .
AGGREGATE
/OUTFILE='[path]\WENDTIME.sav'
/BREAK=case_id cola
/WEEKEND= FIRST(day) /WENDTIME= SUM(duration).

```

```

* merge weekday and weekend data files .
MATCH FILES /FILE='[path]\WDAYTIME.sav'
/FILE='[path]\WENDTIME.sav'
/BY case_id cola.
FORMATS WDAYTIME WENDTIME (TIME) .
EXECUTE.

```