**Answers to Exercise 1**

**1) View the system-defined time use variables.** Click on "Time Use Variables" from the home page under "DATA."

Do any system-defined time use variables capture participation in sports, exercise, and recreation in the morning?

No, system-defined time use variables only specify activities. To analyze participation in sports, exercise, and recreation in the morning, a user would need to create a time use variable in the ATUS-X system. One approach would be to modify ACT\_SPORTS or BLS\_LEIS\_SPORT and apply a time filter to restrict the time use variable to mornings; another would be to build a new time use variable from scratch, selecting appropriate activities and applying a time filter.

**2) Learn more about time use variables.** Click on "About ATUS-X" from the home page. Jump down to "Creating and Selecting Time Use Variables." This section describes how to create time use variables in the ATUS-X system. For more information about what a time use variable is, go to <https://www.atusdata.org/atus/time_use_documentation.shtml>.

**a) Activities**

What major category includes participation in sports, exercise, and recreation? Hint: view the activity coding tree.

Sports, Exercise, and Recreation

**b) Filters (other activity-level characteristics)**

What filters are available from the system?

Time of day, secondary activity (childcare, eating, drinking), location, with whom

What time use variable filters do you need to use to capture participation in sports, exercise, and recreation in the morning?

Time of day

Can you create a single time use variable that captures the time spent in participation in sports, exercise, and recreation from midnight to 6 a.m.?

No, the ATUS diary day covers the period from 4 a.m. of the previous day to 4 a.m. of the reporting day. Create one time use variable that covers the midnight to 4 a.m. period and a second that covers the 4 a.m. to 6 a.m. period and then add these variables in a statistics package to create a measure of participation in sports, exercise, and recreation from midnight to 6 a.m.

**Answers to Exercise 2**

In this exercise, you will become familiar with household- and person-level documentation on the ATUS-X site. You will learn which weights to use and when to use them and you will learn the difference between variables with and without the \_CPS8 suffix.

**1) View the "weight" variables available from the "person" drop-down menu on the ATUS-X site.**

Which weight variable is appropriate for creating estimates of time use from 2003 to 2012?

WT06

Which weight variable is appropriate for creating estimates of time use for Eating and Health Module respondents?

EHWT

Which replicate weights should be used with WT06?

RWT06

**2) View** **the "work status" variables from the "person" drop-down menu on the ATUS-X site**.

**EMPSTAT vs. EMPSTAT\_CPS8**

When is EMPSTAT collected?

At the time of the ATUS interview.

What is the universe for EMPSTAT?

ATUS respondents.

When is EMPSTAT\_CPS8 collected?

At the time of the final CPS interview, 2-5 months before the ATUS interview.

What is the universe for EMPSTAT\_CPS8?

Civilian CPS household members age 15+.

**Answers to Exercise 3**

**1) Distinguishing between data samples (years)**

How many individuals are in the 2003 sample? 20720

How many individuals are in the 2008 sample? 12723

**3) Weekdays vs. Weekends and the Importance of Weights**

**3a. Get frequencies of the newly created WEEKDAY variable by sample.**

How many survey respondents completed the survey on a weekday during each sample year?

2003 10223 2008 6202

What percentage of individuals completed the survey on a weekend during each sample year?

2003 50.66 2008 51.25

**3b. Redo the analysis with the weight variable (WT06) to get accurate estimates.**

Using weights, what percentage of individuals completed the survey on a weekend during each sample year?

2003 28.49 2008 28.41

**Exercise 3: Stata Syntax**

**1) Distinguishing between data samples (years)**

tab year

**2) Create WEEKDAY to distinguish between weekdays and weekends.**

tab day

gen weekday=1

replace weekday=0 if day==1 | day==7

**3) Weekdays vs. Weekends and the Importance of Weights**

**3a. Get frequencies of the newly created WEEKDAY variable by sample.**

tab weekday year

tab weekday year, col

**3b. Redo the analysis with the weight variable (WT06) to get accurate estimates.**

svyset [weight=wt06]

svy: tab weekday year, col

**Exercise 3: SAS Syntax**

**1) Distinguishing between data samples (years)**

PROC FREQ;

TABLES year;

RUN;

**2) Create WEEKDAY to distinguish between weekdays and weekends.**

PROC FREQ;

TABLES day;

RUN;

DATA atusdat.*filename*;

SET atusdat.*filename*;

IF day = 1 THEN weekday = 0;

ELSE IF day =7 THEN weekday = 0;

ELSE weekday = 1;

RUN;

**3) Weekdays vs. Weekends and the Importance of Weights**

**3a. Get frequencies of the newly created WEEKDAY variable by sample.**

PROC FREQ;

TABLES weekday\*year;

RUN;

**3b. Redo the analysis with the weight variable (WT06) to get accurate estimates.**

PROC FREQ;

TABLES weekday\*year;

WEIGHT WT06;

RUN;

**Exercise 3: SPSS Syntax**

**1) Distinguishing between data samples (years)**

freq year.

**2) Create WEEKDAY to distinguish between weekdays and weekends.**

freq day.

recode day (1=0) (2 thru 6=1) (7=0) into weekday.

variable labels weekday 'Weekday Binary'.

execute.

**3) Weekdays vs. Weekends and the Importance of Weights**

**3a. Get frequencies of the newly created WEEKDAY variable by sample.**

crosstabs

/tables=weekday by year

/cells=count column.

**3b. Redo the analysis with the weight variable (WT06) to get accurate estimates.**

weight by wt06.

crosstabs

/tables=weekday by year

/cells=count column.

**Answers to Exercise 4**

**4b. Get the mean number of minutes respondents spent working by WEEKDAY.**

Using weights, what was the mean number of minutes worked on weekdays vs. weekends?

weekday 256.80 weekend 77.15

**4c. Redo the analysis, excluding those who did not report doing any work on their diary day.**

Using weights, what was the mean number of minutes worked on weekdays vs. weekends?

weekday 463.62 weekend 327.44

**4d. See how time spent working varies by day of the week and sample.**

Using weights, what was the mean number of minutes worked by day of the week and sample among those who reported *any* work on their diary day?

2003: weekday 462.28 weekend 329.14

2008: weekday 464.88 weekend 325.90

**Exercise 4: Stata Syntax**

**4b. Get the mean number of minutes respondents spent working by WEEKDAY.**

svyset [weight=wt06]

svy: mean act\_work, over(weekday)

**4c. Redo the analysis, excluding those who did not report doing any work on their diary day.**

gen nz\_act\_work=act\_work if act\_work>0

svy: mean nz\_act\_work, over(weekday)

**4d. See how time spent working varies by day of the week and sample.**

svy: mean nz\_act\_work, over(weekday year)

**Exercise 4: SAS Syntax**

**4b. Get the mean number of minutes respondents spent working by WEEKDAY.**

PROC MEANS mean std;

CLASS weekday;

VAR act\_work;

WEIGHT WT06;

RUN;

**4c. Redo the analysis, excluding those who did not report doing any work on their diary day.**

PROC MEANS mean std;

CLASS weekday;

VAR act\_work;

WHERE act\_work NE 0;

WEIGHT WT06;

RUN;

**4d. See how time spent working varies by day of the week and sample.**

PROC MEANS mean std;

CLASS weekday year;

VAR act\_work;

WHERE act\_work NE 0;

WEIGHT WT06;

RUN;

**Exercise 4: SPSS Syntax**

**4b. Get the mean number of minutes respondents spent working by WEEKDAY.**

means tables=act\_work by weekday

/cells mean count stddev.

**4c. Redo the analysis, excluding those who did not report doing any work on their diary day.**

recode act\_work (0=sysmis) (else=copy) into nz\_act\_work.

variable labels nz\_act\_work 'Minutes for any work'.

execute.

means tables=nz\_act\_work by weekday

/cells mean count stddev.

**4d. See how time spent working varies by day of the week and sample.**

means tables=nz\_act\_work by weekday by year

/cells mean count stddev.

**Answers to Exercise 5 – see Populated Tables Excel file**

**Exercise 5: Stata Syntax**

svyset [weight=wt06];

/\*Recode day of the week\*/

gen newday=.;

replace newday=1 if day>=2 & day<=6;

replace newday=2 if day==7;

replace newday=3 if day==1;

/\*keep only respondents 25 to 64\*/

keep if age>=25 & age<=64;

/\*generate new time use variables\*/

gen ex\_new=ex64+ex46;

gen exal\_new=exal64+exal46;

gen exoth\_new=exoth64+exoth46;

/\*create dummy variable for female\*/

gen female=0;

replace female=1 if sex==2;

/\*generate means for exercising by gender, by gender & day\*/

svy: mean ex612 ex126 ex\_new, over(female);

svy: mean ex612 ex126 ex\_new, over(female newday);

/\*generate means for exercising alone by gender, by gender & day\*/

svy: mean exal612 exal126 exal\_new, over(female);

svy: mean exal612 exal126 exal\_new, over(female newday);

/\*generate means for exercising with others by gender, by gender & day\*/

svy: mean exoth612 exoth126 exoth\_new, over(female);

svy: mean exoth612 exoth126 exoth\_new, over(female newday);

**Exercise 5: SAS Syntax**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Recode day of week.

Add time from 4a.m.-6a.m. and 6p.m.-4a.m.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**data** ex5;

set atusdat.*filename*;

ex\_new = ex64 + ex46;

exal\_new = exal64 + exal46;

exoth\_new = exoth64 + exoth46;

newday = **0**;

if day ge **2** and day le **6** then newday = **1**;

if day = **1** then newday = **2**;

if day = **7** then newday = **3**;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sort by sex and calculate men's and women's weighted

means for the exercise variables not specific to day

of the week.

Use AGE to restrict the universe.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**proc** **sort** data = ex5;

by sex;

**run**;

**proc** **means** data = ex5;

where age ge **25** and age le **64**;

class sex;

var

ex612

ex126

ex\_new

exal612

exal126

exal\_new

exoth612

exoth126

exoth\_new;

weight wt06;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sort by newday. Calculate weighted means

for men and women specific to day of the week.

Use AGE to restrict universe.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**proc** **sort** data = ex5;

by newday;

**run**;

**proc** **means** data = ex5;

where sex = **1** and age ge **25** and age le **64**;

class newday;

var

ex612

ex126

ex\_new

exal61

exal126

exal\_new

exoth612

exoth126

exoth\_new

;

weight wt06;

**run**;

**proc** **means** data = ex5;

where sex = **2** and age ge **25** and age le **64**;

class newday;

var

ex612

ex126

ex\_new

exal612

exal126

exal\_new

exoth612

exoth126

exoth\_new;

weight wt06;

**run**;

**Exercise 5: SPSS Syntax**

\*\*turn on the weight variable and filter by age.

WEIGHT by wt06.

COMPUTE filter\_$=(age ge 25 & age le 64).

VARIABLE LABEL filter\_$ 'age filter'.

FORMAT filter\_$ (f1.0).

FILTER BY filter\_$.

\*\*recode day of the week.

compute newday=1.

if (day=7) newday=2.

if (day=1) newday=3.

execute.

\*\*recode female.

compute female=0.

if (sex=2) female=1.

execute.

\*\*add together 6:00 pm to 4:00 am & 4:00 am to 6:00 am time use variables to create 6:00 pm to 6:00 am variables.

COMPUTE ex\_new= ex64 + ex46.

COMPUTE exal\_new= exal64 + exal46.

COMPUTE exoth\_new= exoth64 + exoth46.

\*\*generate means by gender and day of the week.

MEANS TABLES ex612 ex126 ex\_new by female by newday.

MEANS TABLES exal612 exal126 exal\_new by female by newday.

MEANS TABLES exoth612 exoth126 exoth\_new by female by newday.

**Answers to Exercise 6 – see Populated Tables Excel file**

**Exercise 6: Stata Syntax**

do *filename*.do;

keep if kidund18==1;

svyset [weight=wt06];

/\*generate a gender-marital status variable\*/

gen genmarr=.;

replace genmarr=1 if sex==2 & spousepres!=1;

replace genmarr=2 if sex==2 & spousepres==1;

replace genmarr=3 if sex==1 & spousepres==1;

/\*recode education\*/

gen newed=.;

replace newed=1 if educ>=10 & educ<30;

replace newed=2 if educ>=30 & educ<40;

replace newed=3 if educ>=40 & educ<=43;

/\*generate time use statistics for "total" variable for each genmarr-newed combination\*/

/\*the syntax immediately following is the same as what is below:

svy, subpop(if newed==1 & genmarr==1): mean okpc okrpt oktv okoth;

svy, subpop(if newed==1 & genmarr==2): mean okpc okrpt oktv okoth;

svy, subpop(if newed==1 & genmarr==3): mean okpc okrpt oktv okoth;

svy, subpop(if newed==2 & genmarr==1): mean okpc okrpt oktv okoth;

svy, subpop(if newed==2 & genmarr==2): mean okpc okrpt oktv okoth;

svy, subpop(if newed==2 & genmarr==3): mean okpc okrpt oktv okoth;

svy, subpop(if newed==3 & genmarr==1): mean okpc okrpt oktv okoth;

svy, subpop(if newed==3 & genmarr==2): mean okpc okrpt oktv okoth;

svy, subpop(if newed==3 & genmarr==3): mean okpc okrpt oktv okoth;

\*/

svy: mean okpc okrpt oktv okoth, over(newed genmarr);

\*generate time use statistics for no spouse/with spouse columns for married men and women\*;

foreach x in nosp wsp {;

foreach num2 of numlist 2 3 {;

foreach num of numlist 1 2 3 {;

svy, subpop(if newed==`num' & genmarr==`num2'): mean okpc`x' okrpt`x' oktv`x' okoth`x';

};

};

};

**Exercise 6: SAS Syntax**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

First, recode the education variable to create the

three groups you want to analyze.

Then, Generate a gender-marital status-variable.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**data** ex6;

set atusdat.*filename*;

newed = **0**;

if educ ge **10** and educ le **21** then newed = **1**;

if educ ge **30** and educ le **32** then newed = **2**;

if educ ge **40** and educ le **43** then newed = **3**;

**genmarr=0;**

ifsex= **2** andspousepres ne **1** thengenmarr = **1;**

ifsex= **2** andspousepres **= 1** thengenmarr **= 2;**

ifsex= **3** andspousepres **= 1** thengenmarr **= 3;**

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sort by the new education variable.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**proc** **sort** data = ex6;

by newed;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Calculate means for each group.

Use EDUC to group by education level.

Use KIDUND18 to identify parents with own household

children under 18.

Use SEX to distinguish men and women.

Use SPOUSEPRES to distinguish married and non-married

respondents.

Use WT06 to weight.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**proc** **means** data = ex6;

where kidund18 = **1** and genmarr = **1**;

class newed;

var

okpc

okrpt

oktv

okoth

;

weight wt06;

**run**;

**proc** **means** data = ex6;

where kidund18 = **1** and genmarr = **2**;

class newed;

var

okpc

okpcnosp

okpcwsp

okrpt

okrptnosp

okrptwsp

oktv

tvkidnosp

oktvwsp

okoth

okothnosp

okothwsp

;

weight wt06;

**run**;

**proc** **means** data = ex6;

where kidund18 = **1** and genmarr = 3;

class newed;

var

okpc

okpcnosp

okpcwsp

okrpt

okrptnosp

okrptwsp

oktv

oktvnosp

oktvwsp

okoth

okothnosp

okothwsp

;

weight wt06;

**run**;

**Exercise 6: SPSS Syntax**

\*\*turn on weights.

WEIGHT by wt06.

COMPUTE filter\_$=(kidund18=1).

VARIABLE LABEL filter\_$ 'kid filter'.

FORMAT filter\_$ (f1.0).

FILTER BY filter\_$.

VALUE LABELS

Sex

1 'male'

2 'female'.

\*\*create gender/marriage categories for analysis.

COMPUTE genmarr=0.

IF (sex=1 and spousepres~=1) genmarr=1.

IF (sex=1 and spousepres=1) genmarr=2.

IF (sex=0 and spousepres=1) genmarr=3.

Execute.

\*\*recode education.

COMPUTE newed=0.

IF (educ ge 10 and educ lt 30) newed=1.

IF (educ ge 30 and educ lt 40) newed=2.

IF (educ ge 40 and educ le 43) newed=3.

Execute.

\*generate time use statistics for "total" variable for each genmarr-newed combination\*.

MEANS TABLES=okpc okrpt oktv okoth BY newed BY genmarr.

\*generate time use statistics for no spouse column for married men and women\*.

MEANS TABLES=okpcnosp okrptnosp oktvnosp okothnosp BY newed BY genmarr.

\*generate time use statistics for with spouse column for married men and women\*

MEANS TABLES=okpcwsp okrptwsp oktvwsp okothwsp BY newed BY genmarr.

**Answers to Exercise 7 – see Populated Tables Excel file**

**Exercise 7: Stata Syntax**

keep if rectype==2;

save \_person.dta, replace;

/\*mark persons who are 25+ and are parent, sibling, other relative\*/

gen \_hhadult=0;

replace \_hhadult=1 if age>=25 & age<=85 & (relate==24 | relate==25 | relate==26);

/\*create a new variable that indicates whether anyone in the hh is 25+ and relate==24-26

then because we want it to be a 0/1 variable, replace any values >1 with 1\*/

egen hhadult=sum(\_hhadult),by(caseid);

replace hhadult=1 if hhadult>=1;

/\*keep these variables only for ATUS respondents\*/

keep if lineno==1;

/\*create categorical indicator of age of youngest child\*/

gen ryoungest=.;

replace ryoungest=1 if ageychild>=0 & ageychild<=2;

replace ryoungest=2 if ageychild>=3 & ageychild<=5;

replace ryoungest=3 if ageychild>=6 & ageychild<=12;

replace ryoungest=4 if ageychild>=13 & ageychild<=17;

/\*create indicator of martial status\*/

gen married=.;

replace married=1 if spousepres==1;

replace married=0 if spousepres==2 | spousepres==3;

keep if sex==2;

keep if kidund18==1;

save, replace;

svyset [weight=wt06];

svy: mean childcare if ryoungest!=., over(married);

svy: mean childcare, over(married ryoungest);

svy: mean childcare, over(hhadult married ryoungest);

svy: mean childcare if ryoungest!=., over(married hhadult);

**Exercise 7: SAS Syntax**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Use only the person records from the hierarchical file

to create household characteristic variables.

hhadult = flag indicating presence of a parent, sibling

or other relative over the age of 25

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**data** rost;

set atusdat.ex7;

if rectype ne **2** then delete;

**run**;

**data** hhchar;

set rost;

by caseid;

retain hhadult;

if first.caseid then do;

hhadult = **0**;

huh = age;

end;

if (relate = **24** or relate = **25** or relate = **26**) and age ge **25** and age le **85**

then \_hhadult = **1**;

if last.caseid then output hhchar;

keep caseid hhadult;

**run**;

**data** hhchar;

set ex7;

if ageychild ge **18** then ageychild= **.**;

lineno = **1**;

**run**;

ryoungest= **.**;

if ageychild ge **0** and ageychild le **2** then ryoungest = **1**;

if ageychild ge **3** and ageychild le **5** then ryoungest = **2**;

if ageychild ge **6** and ageychild le **12** then ryoungest = **3**;

if ageychild ge **13** and ageychild le **17** then ryoungest = **4**;

married = **.**;

if spousepres = **1** then married = **1**;

if spousepres = **2** or spousepres = **3** then married = **0**;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sort by ryoungest and married.

Calculate weighted means for married and nonmarried mothers

by ryoungest and all together.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**proc** **sort** data = ex7;

by married ryoungest;

**run**;

**proc** **means** data = ex7;

where sex = **2** and kidund18 = **1**;

class married ryoungest;

var childcare;

weight wt06;

**run**;

**proc** **means** data = ex7;

where sex = **2** and kidund18 = **1**;

class married;

var childcare;

weight wt06;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Repeat means for households with at least one adult

relative (other than a spouse).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**proc** **means** data = ex7;

where sex = **2** and kidund18 = **1** and hhadult = **1**;

class married ryoungest;

var childcare;

weight wt06;

**run**;

**proc** **means** data = ex7;

where sex = **2** and kidund18 = **1** and hhadult = **1**;

class married;

var childcare;

weight wt06;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Repeat means for households with no additional

adults.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**proc** **means** data = ex7;

where sex = **2** and kidund18 = **1** and hhadult = **0**;

class married ryoungest;

var childcare;

weight wt06;

**run**;

**proc** **means** data = ex7;

where sex = **2** and kidund18 = **1** and hhadult = **0**;

class married;

var childcare;

weight wt06;

**run**;

**Exercise 7: SPSS Syntax**

\*\*turn on the weight variable.

WEIGHT by wt06.

COMPUTE filter\_$=(age ge 25 & age le 64).

VARIABLE LABEL filter\_$ 'age filter'.

FORMAT filter\_$ (f1.0).

FILTER BY filter\_$.

\*\*keep only person records.

SELECT IF (relate=2).

\*\*\*AGGREGATE ACROSS PERSON RECORDS.

\*\*mark individuals as adults.

RECODE age (25 thru 85=1) (ELSE=0) INTO xadult.

EXECUTE.

\*\*mark individuals as related adults.

RECODE relate (24 thru 26=1) (ELSE=0) INTO xrelated.

EXECUTE.

\*\*flag people who are aduls & related.

COMPUTE xhhadult=0.

if (xadult=1) & (xrelated=1) xhhadult=1.

execute.

\*\*attach indicator of household adult to all persons in each household.

AGGREGATE

/OUTFILE=\* MODE=ADDVARIABLES

/BREAK=caseid

/hhadult=MAX(xhhadult).

\*\*keep only ATUS respondents.

SELECT IF (lineno=1).

\*\*\*RECODES FOR ANALYSIS.

\*\*recode the age of the youngest kid.

COMPUTE rageychild=$sysmis.

IF (ageychild>=0) & (ageychild<=2) ryoungest=1.

IF (ageychild>=3) & (ageychild<=5) ryoungest=2.

IF (ageychild>=6) & (ageychild<=12) ryoungest=3.

IF (ageychild>=13) & (ageychild<=17) ryoungest=4.

\*\*create an indicator of whether the ATUS respondent is married or not.

COMPUTE married=$sysmis.

IF (spousepres=1) married=1.

IF (spousepres>1) married=0.

\*\*keep only women and those with a child under 18 in the household.

SELECT IF (sex=2).

SELECT IF (ryoungest~=$sysmis).

\*\*generate means.

MEANS TABLES=childcare BY married.

MEAN TABLES=childcare BY married ryoungest.

MEANS TABLES=childcare BY hhadult married rageychild.

MEANS TABLES=childcare BY married hhadult.

**Answers to Exercise 8 – see Populated Tables Excel file**

**Exercise 8: Stata Syntax**

do *filename*.do;

/\*keep only the person record\*/

keep if rectype==2;

/\*keep only the variables we need from the person record\*/

keep caseid lineno wt06;

/\*keep these variables only for ATUS respondents\*/

keep if lineno==1;

sort caseid lineno;

/\*save to merge with summary activity file later\*/

save hourly\_activities\_person.dta, replace;

clear;

do *filename*.do;

/\*keep only activity records\*/

keep if rectype==3;

/\*recode six digit activity into a 4 category variable\*/

gen newact=4;

replace newact=1 if activity>=010100 & activity<010200;

replace newact=2 if activity>=050000 & activity<060000;

replace newact=3 if activity>=120000 & activity<140000;

lab var newact "four category activity variable";

lab def newactl 1 "sleep" 2 "work" 3 "leisure & sports" 4 "other";

lab val newact newactl;

/\*generate dummies based only newly created variable\*/

tab newact, gen(act);

/\*flag the last activity record for each person for use below\*/

gen last\_act=1 if caseid!=caseid[\_n+1];

/\*get start/stop HH:MM from HH:MM:SS formatted time variable\*/

gen \_start=substr(start,1,5);

gen \_stop=substr(stop,1,5);

/\*convert start/stop times to minutes\*/

/\*the stata egenmore package that the next commands use need to be installed

. ssc inst egenmore

or replaced by

. ssc inst egenmore, replace

\*/

egen \_minstart=minutes(\_start);

egen \_minstop=minutes(\_stop);

/\*fix start/stop minutes so that minutes after midnight keep getting bigger instead or starting over at zero\*/

gen minstart=\_minstart;

replace minstart=\_minstart+1440 if \_minstart>=0 & \_minstart<240;

gen minstop=\_minstop;

replace minstop=\_minstop+1440 if \_minstop>=0 & \_minstop<240;

replace minstop=1680 if last\_act==1; //the stop time for the last activity may extend beyond 4:00 a.m. (see DURATION\_EXT)

/\*create start/stop times for each hour of the day\*/

foreach num in 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 {; //start/stop hours

gen hour`num'start=0;

replace hour`num'start=241 + ((`num'-4)\*60);

gen hour`num'stop=0;

replace hour`num'stop=241 + ((`num'-4)\*60) + 59;

};

/\*make sure calculations worked correctly\*/

assert hour04stop==hour04start+59;

assert hour04stop==300;

assert hour23stop==hour23start+59;

assert hour27stop==1680;

/\*create dummy indicators of each of four activities occuring during each hour of the day\*/

/\*I used a loop here because it is more efficient. You could do this for each combination of act1-act4 and hour04-hour27:

gen \_act1\_04=0;

replace \_act1\_04=1 if minstart>=hour04start & minstart<hour04stop & act1==1;

replace \_act1\_04=1 if minstop>=hour04start & minstop<hour04stop & act1==1;

replace \_act1\_04=1 if minstop>=hour04stop & minstart<=hour04start & act1==1;

If you type that syntax out a few times, you'll start to see a pattern, which indicates that you can use a loop instead of

re-typing it 96 times in this case for each activity-hour combination.\*/

foreach var of varlist act1 act2 act3 act4 {; //four activities defined above

foreach num in 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 {; //hours in the day

gen \_`var'\_`num'=0;

\*if act start time occurs within hour time range\*;

replace \_`var'\_`num'=1 if minstart>=hour`num'start & minstart<hour`num'stop & `var'==1;

\*if act stop time occurs within hour time range\*;

replace \_`var'\_`num'=1 if minstop>=hour`num'start & minstop<hour`num'stop & `var'==1;

\*if act start/stop time are before/after hour time range\*;

replace \_`var'\_`num'=1 if minstop>=hour`num'stop & minstart<=hour`num'start & `var'==1;

}; //end activities

}; //end hours

/\*create indicators of whether an activity ever occurs in a one hour interval\*/

foreach var of varlist act1 act2 act3 act4 {; //four activities defined above

foreach num in 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 {; //hours in the day

egen `var'\_`num'=max(\_`var'\_`num'), by(caseid);

}; //end activities

}; //end hours

save hourly\_activities\_activity.dta, replace;

/\*keep only the first activity for each person to merge back with the person record\*/

keep if actline==1;

/\*keep only the variables we want to merge with the person record\*/

keep caseid actline act1\_04-act1\_27 act2\_04-act2\_27 act3\_04-act3\_27 act4\_04-act4\_27;

sort caseid;

merge 1:1 caseid using hourly\_activities\_person.dta; //person and activity files to which we are merging have only ATUS respondents

tab \_merge; //make sure everything worked properly

drop \_merge;

sort caseid;

save hourly\_activities\_merge.dta, replace;

svyset [weight=wt06];

svy: mean act1\_04-act1\_27 act2\_04-act2\_27 act3\_04-act3\_27 act4\_04-act4\_27;

**Exercise 8: SAS Syntax**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

First, call in the activity records of the hierarchical

file and drop extraneous variables.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**data** act;

set atusdat.*filename*;

if rectype ne **3** then delete;

drop caseid lineno pernum wt06;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

This step turns start and stop time into numeric variables

from 240-1680.

START1440 = start time in minutes

STOP1440 = stop time in minutes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**data** acttime;

set act;

starth = scan(start,**1**,':');

startm = scan(start,**2**,':');

start1440 = starth\***60**+startm;

if start1440 <**240** then start1440 = start1440+ **1440**;

stop1440 = start1440 + duration;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

The next section of code initializes arrays that contain indicator

variables for whether a person slept (act1), worked (act2), played (act3) or engaged in other activities (act4) in a given hour of the diary day. The suffixes attached to these variables refer to the end of the hour (e.g., act1\_5 refers to the indicator for whether sleep occurred between 4 a.m. and 5 a.m.). Hours after midnight at the end of the diary day have suffixes referring to the end of the hour plus 24. Arrays also are created for the hour, the start time of the hour in minutes and the stop time of the hour in minutes. For hours after midnight, 1440 is added to the number of minutes (e.g., the start time in minutes for the hour ending at 3 a.m. is 1560). These arrays are used together with the activity code to assign values to the activity indicator variables.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**data** acttime2;

set acttime;

by caseid;

array act1(**24**) act1\_5-act1\_28;

array act2(**24**) act2\_5-act2\_28;

array act3(**24**) act3\_5-act3\_28;

array act4(**24**) act4\_5-act4\_28;

array hr(**24**) hr5-hr28;

array starttime(**24**) start5-start28;

array endtime(**24**) end5-end28;

retain act1\_5-act1\_28 act2\_5-act2\_28 act3\_5-act3\_28 act4\_5-act4\_28;

if first.caseid then do i=**1** to **24**;

act1\_(i)=**0**;

act2\_(i)=**0**;

act3\_(i)=**0**;

act4\_(i)=**0**;

end;

do i=**1** to **24**;

starttime(i)=**180** + i\***60**;

endtime(i)=**240** + i\***60**;

end;

do i=**1** to **24**;

hr(i)=**0**;

if (start1440 gt starttime(i) and start1440 le endtime(i))

or (stop1440 gt starttime(i) and stop1440 le endtime(i))

or (start1440 le starttime(i) and stop1440 gt endtime(i))

then hr(i)=**1**;

if activity=**010101** and hr(i)=**1** then act1\_(i)=**1**;

if activity ge **050101** and activity le **059999** and hr(i)=**1**

then act2\_(i)=**1**;

if activity ge **120000** and activity le **139999** and hr(i)=**1**

then act3\_(i)=**1**;

if activity ne **010101** and (activity lt **050101** or activity gt **059999**)

and (activity lt **120000** or activity gt **139999**) and hr(i) = **1**

then act4\_(i)=**1**;

end;

if last.caseid then output acttime2;

keep caseid act1\_5-act1\_28 act2\_5-act2\_28 act3\_5-act3\_28 act4\_5-act4\_28;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Create a file with just caseid and WT06, which

is on the person level records. Sort and merge

with acttime2.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**data** resp;

set atusdat.*filename*;

if rectype ne **2** then delete;

keep caseid wt06;

**run**;

**proc** **sort** data = resp;

by caseid;

**run**;

**proc** **sort** data = acttime2;

by caseid;

**run**;

**data** resptime;

merge resp acttime2;

by caseid;

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Calculated weighted means for each of the dummy

variables to get the proportion of people who are

engaged in that activity at any given time.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

filename props "c:\atus\props.xls";

ods html body = props;

**proc** **means** data = resptime;

var act1\_5-act1\_28 act2\_5-act2\_28 act3\_5-act3\_28 act4\_5-act4\_28;

weight wt06;

**run**;

ods html close;

**Exercise 8: SPSS Syntax**

\*\*split the file into person and activity records after reading it in.

SORT CASES BY rectype(A) caseid(A) pernum(A).

SAVE OUTFILE= 'hier\_full.sav'

/KEEP ALL.

select if (RECTYPE=2).

SORT CASES BY rectype(A) caseid(A) pernum(A).

SAVE OUTFILE= 'hier\_person.sav'

/KEEP caseid lineno wt06.

GET FILE= 'hier\_full.sav'.

select if (RECTYPE=3).

SORT CASES BY rectype(A) caseid(A) pernum(A).

SAVE OUTFILE= 'hier\_activity.sav'

/KEEP ALL.

\*\*recode activities.

\*sleep.

compute act1=0.

if (activity>=010100 & activity<010200) act1=1.

\*work.

compute act2=0.

if (activity>=050000 & activity<060000) act2=1.

\*leisure & sports.

compute act3=0.

if (activity>=120000 & activity<140000) act3=1.

\*other activities.

compute act4=0.

if (act1=0 & act2=0 & act3=0) act4=1.

execute.

\*\*flag the last activity record for each person for use below.

AGGREGATE

/OUTFILE=\* MODE=ADDVARIABLES

/BREAK=caseid

/xlast\_case=MAX(actline).

compute last\_case=0.

if (xlast\_case=actline) last\_case=1.

execute.

\*\*convert start/stop times to minutes.

\*\*make sure START & STOP are in HH:MM:SS format.

COMPUTE xminstart=CTIME.MINUTES(start).

COMPUTE xminstop=CTIME.MINUTES(stop).

EXECUTE.

\*fix start/stop minutes so that minutes after midnight keep getting bigger instead or starting over at zero.

compute minstart=xminstart.

IF (xminstart>=0 & xminstart<240) minstart=xminstart+1440 .

compute minstop=xminstop.

IF (xminstop>=0 & xminstop<240) minstop=xminstop+1440 .

IF (last\_case=1) minstop=1680.

EXECUTE.

\*\*create start/stop times for each hour of the day.

\*create start minutes.

do repeat

num = 04 to 27 /

varstart = hourstart04 to hourstart27.

compute varstart=241+((num-4)\*60).

end repeat.

execute.

\*create stop minutes.

do repeat

num = 04 to 27 /

varstop = hourstop04 to hourstop27.

compute varstop=241+((num-4)\*60)+59.

end repeat.

execute.

\*\*create dummy indicators of each of four activities occuring during each hour of the day.

\*I used a repeat statment here for each activity to avoid repeating the same syntax 96 times.

do repeat

varstart = hourstart04 to hourstart27 /

varstop = hourstop04 to hourstop27 /

var = xact1\_04 to xact1\_27.

compute var=0.

if ((minstart>=varstart & minstart<varstop) or (minstop>=varstart & minstop<varstop) or (minstop>=varstop & minstart<=varstart) & act1=1) var=1.

end repeat.

execute.

do repeat

varstart = hourstart04 to hourstart27 /

varstop = hourstop04 to hourstop27 /

var = xact2\_04 to xact2\_27.

compute var=0.

if ((minstart>=varstart & minstart<varstop) or (minstop>=varstart & minstop<varstop) or (minstop>=varstop & minstart<=varstart) & act2=1) var=1.

end repeat.

execute.

do repeat

varstart = hourstart04 to hourstart27 /

varstop = hourstop04 to hourstop27 /

var = xact3\_04 to xact3\_27.

compute var=0.

if ((minstart>=varstart & minstart<varstop) or (minstop>=varstart & minstop<varstop) or (minstop>=varstop & minstart<=varstart) & act3=1) var=1.

end repeat.

execute.

do repeat

varstart = hourstart04 to hourstart27 /

varstop = hourstop04 to hourstop27 /

var = xact4\_04 to xact4\_27.

compute var=0.

if ((minstart>=varstart & minstart<varstop) or (minstop>=varstart & minstop<varstop) or (minstop>=varstop & minstart<=varstart) & act4=1) var=1.

end repeat.

execute.

\*\*aggregate all of the xact\* variables for each hour.

\*I've done this for the 4:00 a.m. - 5:00 a.m. hour for each activity; this needs to be done for each one-hour interval.

AGGREGATE

/OUTFILE=\* MODE=ADDVARIABLES

/BREAK=caseid

/act1\_04=MAX(xact1\_04).

AGGREGATE

/OUTFILE=\* MODE=ADDVARIABLES

/BREAK=caseid

/act2\_04=MAX(xact2\_04).

AGGREGATE

/OUTFILE=\* MODE=ADDVARIABLES

/BREAK=caseid

/act3\_04=MAX(xact3\_04).

AGGREGATE

/OUTFILE=\* MODE=ADDVARIABLES

/BREAK=caseid

/act4\_04=MAX(xact4\_04).

\*\*Last steps are to save & merge with person-level file, generate estimates.

\*SAVE OUTFILE= '\_activity.sav'

\* /KEEP caseid actline act1\_04 act1\_05...act4\_26 act4\_27.

**Exercise 9: Understanding the Well-being Module**

**1) Examine frequencies of the six subjective well-being items.**

**1a. Understanding subjective well-being coding: HAPPINESS and STRESS.**

In how many *activities* do respondents report being 'very happy'? 11,957

In how many *activities* do respondents report being 'very stressed'? 1,397

**1b. Why are so many records coded as 'NIU (Not in universe)'?**

The subjective well-being reports are collected from *only (up to) three* randomly selected activities for each respondent. Thus, most of the activities that the respondent reports during the diary day are not selected for the well-being module and are therefore coded as NIU. For activities to be eligible for selection, they must be at least five minutes in duration and be activities *other than*  sleeping, grooming, personal activities, refusal, or don't know.

**1c. How many activity records have subjective well-being data available?**

38,160

**3) See populated tables in Excel.**

**4) Interpret the mean for happiness.**

The average happiness score of all types of activities across all of the well-being module respondents is 4.25 on a scale of 0 to 6.

**5) See populated tables in Excel.**

**6) See populated tables in Excel.**

**Exercise 9: Stata Syntax**

**\*1a. Understanding subjective well-being coding: HAPPINESS and STRESS.**

\*Save data file and keep activity records only for now.

save data, replace

keep if rectype==3

\*Method 1: basic tabulation

tabulate scpain

tabulate schappy

tabulate scsad

tabulate sctired

tabulate scstress

tabulate meaning

\*Method 2: write a loop

foreach var of varlist scpain schappy scsad sctired scstress meaning {

tab `var'

}

**\*1c. How many activity records have subjective well-being data available?**

\*Method 1: drop NIU if not needed

foreach var of varlist scpain schappy scsad sctired scstress meaning {

drop if `var'==99

}

\*Method 2: do not drop NIU and only tabulate variables that have non 99 values

foreach var of varlist scpain schappy scsad sctired scstress meaning {

tab `var' if `var'!=99

}

**\*2) Recode NIU as missing.**

\*Method 1: repeat for all measures

\*pain\*

generate newpain=.

replace newpain=scpain if scpain==0 | scpain<=6

tab newpain

\*Method 2: use a loop

foreach var of varlist scpain schappy scsad sctired scstress meaning {

generate new`var'=.

replace new`var'=`var' if `var'==0 | `var'<=6

}

**\*3) Generate weighted means for the six subjective well-being items using AWBWT.**

svyset [weight=awbwt]

svy: tab newpain

svy: tab newhappy

svy: tab newsad

svy: tab newfatigue

svy: tab newstress

svy: tab newmeaning

\***5) Generate weighted means and unweighted N's for each subjective well-being measure during sports, exercise, and recreation and during work and work-related activities. Using the extract you used to answer the previous questions in this exercise, write a program to create the estimates needed to populate Table 9.  Solutions to the exercise, including sample programs, are available in the answers packet so that you can check your work.**

generate newactivity=0

replace newactivity=1 if activity>=130101 & activity<=139999

replace newactivity=2 if activity>=050101 & activity<=059999

tab newactivity

by newactivity, sort: tabstat newsad newhappy newpain newfatigue newstress newmeaning [aw=awbwt]

save data, replace

**\*6) Generate weighted person-level estimates using WBWT of average subjective well-being (for each of the six well-being measures) for men and women aged 25 to 64 during paid work and work-related activities.**

\*keep only activity records with non-missing well-being data

keep if newactivity==2 & awbwt>0 & awbwt!=.

\*generate a new activity line number to use later

bysort caseid: gen newactline=\_n

tab newactline

\*create average well-being for only work activities

egen newpain\_mean=mean(newpain), by(caseid)

egen newhappy\_mean=mean(newhappy) , by(caseid)

egen newsad\_mean=mean(newsad) , by(caseid)

egen newfatigue\_mean=mean(newfatigue) , by(caseid)

egen newstress\_mean=mean(newstress) , by(caseid)

egen newmeaning\_mean= mean(newmeaning) , by(caseid)

sort caseid newactline

keep if newactline==1

keep caseid newpain\_mean newhappy\_mean newsad\_mean newfatigue\_mean newstress\_mean newmeaning\_mean

save well\_being.dta, replace

\*now back to the person records—prune the sample

use data, clear

keep if rectype==2

keep if age>=25 & age<=64

sort caseid

save person\_level.dta, replace

merge 1:1 caseid using well\_being.dta

keep if \_merge==3

save gender\_wellbeingdata.dta, replace

svyset [weight=wbwt]

svy: mean newpain\_mean newhappy\_mean newsad\_mean newfatigue\_mean newstress\_mean newmeaning\_mean, over(sex)

table sex, c(n newpain\_mean n newhappy\_mean n newsad\_mean)

table sex, c(n newfatigue\_mean n newstress\_mean n newmeaning\_mean)

**Exercise 9: SAS Syntax**

**\*1a. Understanding subjective well-being coding: HAPPINESS and STRESS.**

PROC FREQ;

TABLES schappy meaning scsad scstress sctired scpain;

RUN;

**\*2) Recode NIU as missing.**

DATA atusdat.*filename*;

SET atusdat.*filename*;

IF schappy>6 THEN schappy1=.;

ELSE schappy1=schappy;

IF meaning>6 THEN meaning1=.;

ELSE meaning1=meaning;

IF scsad>6 THEN scsad1=.;

ELSE scsad1=scsad;

IF scstress>6 THEN scstress1=.;

ELSE scstress1=scstress;

IF sctired>6 THEN sctired1=.;

ELSE sctired1=sctired;

IF scpain>6 THEN scpain1=.;

ELSE scpain1=scpain;

RUN;

**\*3) Generate weighted means for the six subjective well-being items using AWBWT.**

PROC MEANS;

VAR schappy1 meaning1 scsad1 scstress1 sctired1 scpain1;

WEIGHT awbwt;

RUN;

\***5) Generate weighted means and unweighted N's for each subjective well-being measure during sports, exercise, and recreation and during work and work-related activities. Using the extract you used to answer the previous questions in this exercise, write a program to create the estimates needed to populate Table 9.  Solutions to the exercise, including sample programs, are available in the answers packet so that you can check your work.**

DATA atusdat.*filename*;

SET atusdat.*filename*;

IF 130101<=activity<=139999 THEN group=1;

ELSE IF 050101<=activity<=059999 THEN group=2;

ELSE group=.;

RUN;

\* to get means of the subjective well-being by group (sports=1, working=2) \*;

PROC MEANS;

CLASS group;

VAR schappy1 meaning1 scsad1 scstress1 sctired1 scpain1;

WEIGHT awbwt;

RUN;

**\*6) Generate weighted person-level estimates using WBWT of average subjective well-being (for each of the six well-being measures) for men and women aged 25 to 64 during paid work and work-related activities.**

DATA work;

SET atusdat.*filename*;

WHERE group=2 & awbwt>0;

KEEP caseid schappy1 meaning1 scsad1 scstress1 sctired1 scpain1;

RUN;

PROC SORT DATA=work;

BY caseid;

RUN;

DATA work;

SET work;

count+1;

BY caseid;

IF first.caseid THEN count=1;

RUN;

\* to transpose the activity-level records to the person-level records \*;

DATA work1;

SET work;

BY caseid;

RETAIN happy1\_1-happy1\_3 meaning1\_1-meaning1\_3 sad1\_1-sad1\_3 stress1\_1-stress1\_3 tired1\_1-tired1\_3 pain1\_1-pain1\_3;

ARRAY happy1(3) happy1\_1-happy1\_3;

ARRAY meaningful1(3) meaning1\_1-meaning1\_3;

ARRAY sad1(3) sad1\_1-sad1\_3;

ARRAY stress1(3) stress1\_1-stress1\_3;

ARRAY tired1(3) tired1\_1-tired1\_3;

ARRAY pain1(3) pain1\_1-pain1\_3;

IF first.caseid THEN DO;

DO i=1 TO 3;

happy1(i)=.;

meaningful1(i)=.;

sad1(i)=.;

stress1(i)=.;

tired1(i)=.;

pain1(i)=.;

END;

END;

happy1(count) = schappy1;

meaningful1(count) = meaning1;

sad1(count) = scsad1;

stress1(count) = scstress1;

tired1(count) = sctired1;

pain1(count) = scpain1;

IF last.caseid THEN OUTPUT;

DROP schappy1 meaning1 scsad1 scstress1 sctired1 scpain1;

RUN;

DATA work1;

SET work1;

tothappy1=sum(of happy1\_1 – happy1\_3);

totmeaning1=sum(of meaning1\_1 – meaning1\_3);

totsad1=sum(of sad1\_1 – sad1\_3);

totstress1=sum(of stress1\_1 – stress1\_3);

tottired1=sum(of tired1\_1 – tired1\_3);

totpain1=sum(of pain1\_1 – pain1\_3);

avghappy1=tothappy1/count;

avgmeaning1=totmeaning1/count;

avgsad1=totsad1/count;

avgstress1=totstress1/count;

avgtired1=tottired1/count;

avgpain1=totpain1/count;

RUN;

DATA menwomen;

SET atusdat.*filename*;

KEEP caseid age sex wbwt;

WHERE 25<=age<=64;

RUN;

\* to merge two data sets \*;

PROC SORT DATA = work1;

BY caseid;

RUN;

PROC SORT DATA = menwomen;

BY caseid;

RUN;

DATA work1menwmn;

MERGE work1(in=a) menwomen(in=b);

BY caseid;

IF a=1 & b=1;

RUN;

PROC MEANS;

CLASS sex;

VAR avghappy1 avgmeaning1 avgsad1 avgstress1 avgtired1 avgpain1;

WEIGHT wbwt;

RUN;

**Exercise 9: SPSS Syntax**

**\*1a. Understanding subjective well-being coding: HAPPINESS and STRESS.**

FREQUENCIES VARIABLES=SCPAIN SCHAPPY SCSAD SCSTRESS MEANING SCTIRED

/ORDER=ANALYSIS.

MISSING VALUES SCPAIN (96,97,99).

**\*2) Recode NIU as missing.**

COMPUTE SCPAIN1=SCPAIN.

RECODE SCPAIN1 (7 thru Highest=SYSMIS).

COMPUTE SCHAPPY1=SCHAPPY.

RECODE SCHAPPY1 (7 thru Highest=SYSMIS).

COMPUTE SCSAD1=SCSAD.

RECODE SCSAD1 (7 thru Highest=SYSMIS).

COMPUTE SCSTRESS1=SCSTRESS.

RECODE SCSTRESS1 (7 thru Highest=SYSMIS).

COMPUTE MEANING1=MEANING.

RECODE MEANING1 (7 thru Highest=SYSMIS).

COMPUTE SCTIRED1=SCTIRED.

RECODE SCTIRED1 (7 thru Highest=SYSMIS).

**\*3) Generate weighted means for the six subjective well-being items using AWBWT.**

weight by AWBWT.

MEANS TABLES=SCPAIN1 SCHAPPY1 SCSAD1 SCSTRESS1 MEANING1 SCTIRED1

/CELLS=MEAN.

\***5) Generate weighted means and unweighted N's for each subjective well-being measure during sports, exercise, and recreation and during work and work-related activities. Using the extract you used to answer the previous questions in this exercise, write a program to create the estimates needed to populate Table 9.  Solutions to the exercise, including sample programs, are available in the answers packet so that you can check your work.**

IF ACTIVITY GE 130101 AND ACTIVITY LE 139999 GROUP=1.

IF ACTIVITY GE 50101 AND ACTIVITY LE 59999 GROUP=2.

VAL LAB GROUP 1 'SPORTS, EXERCISE & RECREATION'

2 'WORKING & WORKING RELATED ACTIVITIES'.

MEANS TABLES=SCPAIN1 SCHAPPY1 SCSAD1 SCSTRESS1 MEANING1 SCTIRED1 BY GROUP

/CELLS=MEAN COUNT.

**\*6) Generate weighted person-level estimates using WBWT of average subjective well-being (for each of the six well-being measures) for men and women aged 25 to 64 during paid work and work-related activities.**

SAVE OUTFILE="WELL\_BEING.SAV".

SELECT IF GROUP=2.

WEIGHT OFF.

AGGREGATE

/OUTFILE='Z:\timeuse\atus\staff\joan\workshop2013\PERSONS\_WELLBEING.sav'

/BREAK=CASEID

/SCPAIN1\_mean=MEAN(SCPAIN1)

/SCHAPPY1\_mean=MEAN(SCHAPPY1)

/SCSAD1\_mean=MEAN(SCSAD1)

/SCSTRESS1\_mean=MEAN(SCSTRESS1)

/MEANING1\_mean=MEAN(MEANING1)

/SCTIRED1\_mean=MEAN(SCTIRED1)

/WBWT=MEAN(AWBWT ).

GET FILE="PERSONS.SAV".

SELECT IF AGE GE 25 AND AGE LE 64.

STAR JOIN

/SELECT t0.CASEID, t0.LINENO, t0.WT06, t0.PERNUM, t0.SEX, t0.AGE, t1.SCPAIN1\_mean,

t1.SCHAPPY1\_mean, t1.SCSAD1\_mean, t1.SCSTRESS1\_mean, t1.MEANING1\_mean, t1.SCTIRED1\_mean, t1.WBWT

/FROM \* AS t0

/JOIN 'Z:\timeuse\atus\staff\joan\workshop2013\PERSONS\_WELLBEING.sav' AS t1

ON t0.CASEID=t1.CASEID

/OUTFILE FILE=\*.

WEIGHT BY WBWT.

MEANS TABLES=SCPAIN1\_mean SCHAPPY1\_mean SCSAD1\_mean SCSTRESS1\_mean MEANING1\_mean SCTIRED1\_mean BY SEX

/CELLS=MEAN.

WEIGHT OFF.

MEANS TABLES=SCPAIN1\_mean SCHAPPY1\_mean SCSAD1\_mean SCSTRESS1\_mean MEANING1\_mean SCTIRED1\_mean BY SEX

/CELLS=COUNT.

**Exercise 10: Spouse characteristics in ATUS-X**

The aims of this exercise are to become familiar with 1) thinking about how to leverage the rich information available in the ATUS about respondents and household members, 2) creating couple-level variables, and 3) analyzing couple-level data. While the ATUS only collects time use information for one member of each surveyed household, sociodemographic information is obtained for all the members of the household. ATUS-X provides the following characteristics about the ATUS respondent's spouse: age, sex, race, Hispanic origin, educational attainment, spouse employed, employment status, usual work hours and weekly earnings. *Note that when analyzing couple-level characteristics from the ATUS, you may need to keep in mind whether the respondent is male or female.*

Create a rectangular extract including all the members in respondent’s household for the 2012 sample along with the following variables: time spent in household activities (ACT\_HHACT), individual characteristics (AGE, SEX, EDUC, EMPSTAT), and spouse characteristics (SPOUSEPRES, SPAGE, SPSEX, SPSPEDUC, SPEMPSTAT).

1. Get the characteristics of the sample.

* How many couples are in the sample? 6378
* How many respondents are male? 3101 Female? 3277
* What proportion of respondents are employed? 61%
* What proportion of spouses are employed? 64%
* What proportion of couples both members have the same educational attainment (1: less than HS diploma, 2: HS diploma, no college, 3: Some college, 4: college degree or higher)? 56.9%

1. Only for heterosexual couples, create typologies of couples according to the age, employment status and educational attainment of both members of the couple.

* How many dual earner couples are there in the sample? 3124 Male breadwinner? 1536
* In how many couples is the male older than the female? 1140
* How many hypergamous couples are there in the sample? 1198 Hypogamous? 1396

1. Compare time spent in household activities according to the characteristics of the couples.

* What is the mean number of minutes spent in household activities for:
  + men in dual-earner couple arrangements? 77.69
  + women in dual-earner couple arrangements? 122.56
* What is the difference between men's and women's time spent in household activities when:
  + men are older than their wives? 88.94-154.44=-65.5
  + wives are older than their husbands? 72.25-157.55=-85.3
  + husbands and wives are within five years of one another? 90.96-157.76=-66.8
  + men are more educated than their wives? 81.12-163.04=-81.92
  + women are more educated than their husbands? 94.34-145.40=-51.06
  + husbands and wives have the same level of education? 90.85-169.46=-68.61

**Exercise 10: Stata Syntax**

**1) Get the characteristics of the sample.**

\*couple in the sample and sex\*

generate couple=0

replace couple=1 if spousepres<3

tabulate couple

tabulate couple spsex

**2) Only for heterosexual couples, create typologies of couples according to the age, employment status and educational attainment of both members of the couple.**

\*employed respondents including both employed at work(empstat=1) and employed but absent(empstat=2)\*

tab empstat

gen employed=0

replace employed=1 if empstat==1|empstat==2

replace employed=. if employed>=96 | couple==0

gen sp\_employed=0

replace sp\_employed=1 if spempstat==1|spempstat==2

replace sp\_employed=. if spempstat>=96| couple==0

tab employed sp\_employed if couple==1 [aw=wt06], row column cell

\*education\*

gen educatt=0

replace educatt=. if educ==999 | educ==.

replace educatt=1 if educ>=10 & educ<=17

replace educatt=2 if educ>=20 & educ<=21

replace educatt=3 if educ>=30 & educ<=32

replace educatt=4 if educ>=40 & educ<=43

gen speducatt=0

replace speducatt=. if speduc==999 | speduc==.

replace speducatt=1 if speduc>=10 & speduc<=17

replace speducatt=2 if speduc>=20 & speduc<=21

replace speducatt=3 if speduc>=30 & speduc<=32

replace speducatt=4 if speduc>=40 & speduc<=43

gen sameeduc=0

replace sameeduc=. if couple==0

replace sameeduc=1 if educatt==speducatt

tab sameeduc [aw=wt06]

\*heterosexual couples, typologies of couples according to the age, employement and educational attainment\*

gen diffsex=.

replace diffsex=1 if sex!=spsex

replace diffsex=0 if sex==spsex | sex==.

replace diffsex=. if couple==0

keep if diffsex==1

tab diffsex

gen age\_diff=age-spage

gen age\_type=.

replace age\_type=1 if age\_diff>5 & spsex==1

replace age\_type=2 if age\_diff<=5 & age\_diff>=-5 & spsex==1

replace age\_type=3 if age\_diff<-5 & spsex==1

replace age\_type=3 if age\_diff>5 & spsex==2

replace age\_type=2 if age\_diff<=5 & age\_diff>=-5 & spsex==2

replace age\_type=1 if age\_diff<-5 & spsex==2

tab age\_type

gen empstat\_type=.

replace empstat\_type=. if couple==0 | employed==. | sp\_employed==.

replace empstat\_type=1 if employed==1 & sp\_employed==1 & spsex==1

replace empstat\_type=2 if employed==0 & sp\_employed==1 & spsex==1

replace empstat\_type=3 if employed==1 & sp\_employed==0 & spsex==1

replace empstat\_type=4 if employed==0 & sp\_employed==0 & spsex==1

replace empstat\_type=1 if employed==1 & sp\_employed==1 & spsex==2

replace empstat\_type=2 if employed==1 & sp\_employed==0 & spsex==2

replace empstat\_type=3 if employed==0 & sp\_employed==1 & spsex==2

replace empstat\_type=4 if employed==0 & sp\_employed==0 & spsex==2

tab empstat\_type

gen educatt\_type=.

replace educatt\_type=. if couple==0 | educatt==. | speducatt==.

replace educatt\_type=1 if educatt==speducatt & spsex==1

replace educatt\_type=2 if educatt<speducatt & spsex==1

replace educatt\_type=3 if educatt>speducatt & spsex==1

replace educatt\_type=1 if educatt==speducatt & spsex==2

replace educatt\_type=2 if educatt>speducatt & spsex==2

replace educatt\_type=3 if educatt<speducatt & spsex==2

tab educatt\_type

**3) Compare time spent in household activities according to the characteristics of the couples**

by empstat\_type spsex, sort: tabstat act\_hhact [aw=wt06]

by age\_type spsex, sort: tabstat act\_hhact [aw=wt06]

by educatt\_type spsex, sort:tabstat act\_hhact [aw=wt06]

**Exercise 10: SAS Syntax**

**1) Get the characteristics of the sample.**

DATA atusdat.*filename*;

SET atusdat.*filename;*

couple=0;

IF spousepres<3 THEN couples=1;

RUN;

PROC FREQ;

TABLES couples;

RUN;

PROC FREQ;

TABLES couples \* sex;

RUN;

DATA atusdat.*filename*;

SET atuadat.*filename*;

IF (empstat=1 or empstat=2) THEN employed=1;

ELSE employed=0;

IF spempstat>=96 THEN sp\_employed=.;

ELSE IF (spempstat=1 or spempstat=2) THEN sp\_employed=1;

ELSE sp\_employed=0;

RUN;

PROC MEANS;

VAR employed sp\_employed;

WEIGHT wt06;

RUN;

DATA atusdat.*filename*;

SET atusdat.*filename*;

IF educ=999 THEN edattan=.;

ELSE IF 10<=educ<=17 THEN edattan=1;

ELSE IF 20<=educ<=21 THEN edattan=2;

ELSE IF 30<=educ<=32 THEN edattan=3;

ELSE IF 40<=educ<=43 THEN edattan=4;

ELSE edattan=0;

IF speduc>=998 THEN sp\_edattan=.;

ELSE IF 10<=speduc<=17 THEN sp\_edattan=1;

ELSE IF 20<=speduc<=21 THEN sp\_edattan=2;

ELSE IF 30<=speduc<=32 THEN sp\_edattan=3;

ELSE IF 40<=speduc<=43 THEN sp\_edattan=4;

ELSE sp\_edattan=0;

IF couples=0 THEN same\_educ=.;

ELSE IF edattan=sp\_edattan THEN same\_educ=1;

ELSE same\_educ=0;

RUN;

PROC MEANS;

VAR same\_educ;

WEIGHT wt06;

RUN;

**2) Only for heterosexual couples, create typologies of couples according to the age, employment status and educational attainment of both members of the couple.**

DATA atusdat.*filename*;

SET atusdat.*filename*;

WHERE spsex~=. & sex~=spsex;

RUN;

DATA atusdat.*filename*;

SET atusdat.*filename*;

dif\_age = age-spage;

IF sex=1 THEN DO;

IF couples=0 or dif\_age=. THEN type\_age=.;

ELSE IF dif\_age>5 THEN type\_age=1;

ELSE IF -5<=dif\_age<=5 THEN type\_age=2;

ELSE IF dif\_age<-5 THEN type\_age=3;

IF couples=0 or employed=. or sp\_employed=. THEN type\_emp=.;

ELSE IF employed=1 & sp\_employed=1 THEN type\_emp=1;

ELSE IF employed=1 & sp\_employed=0 THEN type\_emp=2;

ELSE IF employed=0 & sp\_employed=1 THEN type\_emp=3;

ELSE IF employed=0 & sp\_employed=0 THEN type\_emp=4;

IF couples=0 or sp\_edattan=. THEN type\_educ=.;

ELSE IF edattan=sp\_edattan THEN type\_educ=1;

ELSE IF edattan>sp\_edattan THEN type\_educ=2;

ELSE IF edattan<sp\_edattan THEN type\_educ=3;

END;

IF sex=2 THEN DO;

IF couples=0 or dif\_age=. THEN type\_age=.;

ELSE IF dif\_age>5 THEN type\_age=3;

ELSE IF -5<=dif\_age<=5 THEN type\_age=2;

ELSE IF dif\_age<-5 THEN type\_age=1;

IF couples=0 or employed=. or sp\_employed=. THEN type\_emp=.;

ELSE IF employed=1 & sp\_employed=1 THEN type\_emp=1;

ELSE IF employed=0 & sp\_employed=1 THEN type\_emp=2;

ELSE IF employed=1 & sp\_employed=0 THEN type\_emp=3;

ELSE IF employed=0 & sp\_employed=0 THEN type\_emp=4;

IF couples=0 or sp\_edattan=. THEN type\_educ=.;

ELSE IF edattan=sp\_edattan THEN type\_educ=1;

ELSE IF edattan<sp\_edattan THEN type\_educ=2;

ELSE IF edattan>sp\_edattan THEN type\_educ=3;

END;

RUN;

PROC FREQ;

TABLES type\_age type\_emp type\_educ;

RUN;

**3) Compare time spent in household activities according to the characteristics of the couples**

PROC MEANS;

VAR act\_hhact;

CLASS sex type\_emp;

WEIGHT wt06;

RUN;

PROC MEANS;

VAR act\_hhact;

CLASS sex type\_age;

WEIGHT wt06;

RUN;

PROC MEANS;

VAR act\_hhact;

CLASS sex type\_educ;

WEIGHT wt06;

RUN;

**Exercise 10: SPSS Syntax**

**1) Get the characteristics of the sample.**

\*\*\*\*\*\*Create variable couples.

COMPUTE couples=0.

IF spousepres<3 couples=1.

VAR LAB couples 'Living with spouse or unmarried partner'.

VAL LAB couples 0 'No' 1 'Yes'.

FREQUENCIES COUPLES.

CROSSTAB COUPLES BY SEX.

WEIGHT BY WT06.

COMPUTE EMPLOYED=0.

IF EMPSTAT=1 OR EMPSTAT=2 EMPLOYED=1.

VAR LAB EMPLOYED "Respondent employed".

FREQUENCIES EMPLOYED.

COMPUTE SP\_EMPLOYED=0.

IF SPEMPSTAT=1 OR SPEMPSTAT=2 SP\_EMPLOYED=1.

IF SPEMPSTAT GE 96 SP\_EMPLOYED =9.

MISSING VALUES SP\_EMPLOYED (9).

VAR LAB SP\_EMPLOYED "Spouse employed".

VAL LAB EMPLOYED SP\_EMPLOYED 0 'No' 1 'Yes'.

FREQUENCIES SP\_EMPLOYED.

COMPUTE EDATTAN=0.

IF EDUC GE 10 AND EDUC LE 17 EDATTAN=1.

IF EDUC GE 20 AND EDUC LE 21 EDATTAN=2.

IF EDUC GE 30 AND EDUC LE 32 EDATTAN=3.

IF EDUC GE 40 AND EDUC LE 43 EDATTAN=4.

IF EDUC=999 EDATTAN=9.

VAR LAB EDATTAN "Educational attainment of the respondent".

MISSING VALUES EDATTAN (9).

COMPUTE SP\_EDATTAN=0.

IF SPEDUC GE 10 AND SPEDUC LE 17 SP\_EDATTAN=1.

IF SPEDUC GE 20 AND SPEDUC LE 21 SP\_EDATTAN=2.

IF SPEDUC GE 30 AND SPEDUC LE 32 SP\_EDATTAN=3.

IF SPEDUC GE 40 AND SPEDUC LE 43 SP\_EDATTAN=4.

IF SPEDUC GE 998 SP\_EDATTAN=9.

VAR LAB SP\_EDATTAN "Educational attainment of the spouse".

MISSING VALUES SP\_EDATTAN (9).

VAL LAB EDATTAN SP\_EDATTAN 1 'Less than HS diploma'

2 ' HS diploma, no college'

3 'Some college'

4 'College degree+'.

COMPUTE SAME\_EDUC=0.

IF EDATTAN=SP\_EDATTAN SAME\_EDUC=1.

IF COUPLES=0 SAME\_EDUC=9.

MISSING VALUES SAME\_EDUC (9).

FREQUENCIES SAME\_EDUC.

**2) Create typologies of couples according to the age, employment status and educational attainment of both members of the couple.**

\*\*\*Select Heterosexual couples.

compute same=SEX NE SPSEX.

COMPUTE DIF\_AGE=AGE-SPAGE.

EXECUTE.

DO IF SEX=1.

COMPUTE TYPE\_AGE=0.

IF DIF\_AGE GT 5 TYPE\_AGE=1.

IF DIF\_AGE GE -5 AND DIF\_AGE LE 5 TYPE\_AGE=2.

IF DIF\_AGE LT -5 TYPE\_AGE=3.

IF COUPLES=0 TYPE\_AGE=9.

VAR LAB TYPE\_AGE 'Type of couple according to the age'.

VAL LAB TYPE\_AGE 1 'Man more than 5 years older than woman'

2 'Man and woman same age'

3 'Woman more than 5 years old than man'.

COMPUTE TYPE\_EMP=0.

IF EMPLOYED=1 AND SP\_EMPLOYED=1 TYPE\_EMP=1.

IF EMPLOYED=1 AND SP\_EMPLOYED=0 TYPE\_EMP=2.

IF EMPLOYED=0 AND SP\_EMPLOYED=1 TYPE\_EMP=3.

IF EMPLOYED=0 AND SP\_EMPLOYED=0 TYPE\_EMP=4.

IF COUPLES=0 TYPE\_EMP=9.

VAR LAB TYPE\_EMP 'Type of couple according to the employment status'.

VAL LAB TYPE\_EMP 1 'Dual-earner couple'

2 'Male breadwinner couple'

3 'Female breadwinner couple'

4 'None employed'.

COMPUTE TYPE\_EDUC=0.

IF EDATTAN=SP\_EDATTAN TYPE\_EDUC=1.

IF EDATTAN>SP\_EDATTAN TYPE\_EDUC=2.

IF EDATTAN<SP\_EDATTAN TYPE\_EDUC=3.

IF COUPLES=0 TYPE\_EDUC=9.

VAR LAB TYPE\_EDUC 'Type of couple according to the educational attainment'.

VAL LAB TYPE\_EDUC 1 'Homogamy' 2 'Hypergamy' 3 'Hypogamy'.

MISSING VALUES TYPE\_AGE TYPE\_EMP TYPE\_EDUC (0, 9).

END IF.

DO IF SEX=2.

COMPUTE TYPE\_AGE=0.

IF DIF\_AGE GT 5 TYPE\_AGE=3.

IF DIF\_AGE GE -5 AND DIF\_AGE LE 5 TYPE\_AGE=2.

IF DIF\_AGE LT -5 TYPE\_AGE=1.

IF COUPLES=0 TYPE\_AGE=9.

COMPUTE TYPE\_EMP=0.

IF EMPLOYED=1 AND SP\_EMPLOYED=1 TYPE\_EMP=1.

IF EMPLOYED=0 AND SP\_EMPLOYED=1 TYPE\_EMP=2.

IF EMPLOYED=1 AND SP\_EMPLOYED=0 TYPE\_EMP=3.

IF EMPLOYED=0 AND SP\_EMPLOYED=0 TYPE\_EMP=4.

IF COUPLES=0 TYPE\_EMP=9.

COMPUTE TYPE\_EDUC=0.

IF EDATTAN=SP\_EDATTAN TYPE\_EDUC=1.

IF EDATTAN<SP\_EDATTAN TYPE\_EDUC=2.

IF EDATTAN>SP\_EDATTAN TYPE\_EDUC=3.

IF COUPLES=0 TYPE\_EDUC=9.

END IF.

WEIGHT OFF.

FREQUENCIES TYPE\_AGE TYPE\_EMP TYPE\_EDUC.

**3) Compare time spent in household activities according to the characteristics of the couples.**

WEIGHT BY WT06.

MEANS TABLES=ACT\_HHACT BY SEX BY TYPE\_AGE TYPE\_EMP TYPE\_EDUC

/CELLS=MEAN.