



Project no. *FP6-2004-506590*

Project acronym *WOLIWEB*

Project title *The socio-economic determinants of citizens' work life attitudes, preferences and perceptions, using data from the continuous web-based European Wage Indicator Survey*

Instrument: STREP

Thematic Priority PRIORITY 7, Research Area 3.2.2.

Calculating the coefficients for the Salary Checker

Deliverable: D22c Explaining the Salary Checker

Part of deliverable D22: Dataset-year 1, year 2, year 3, including the weights and the coefficients for the Salary Checks

Report Version: Final

Date of preparation: 31.03.2007

Authors of this report: Kea Tijdens, University of Amsterdam
Joyce Jacobs, University of Amsterdam

TABLE OF CONTENTS

1. INTRODUCTION	2
2. THE SALARY CHECK SCREENS.....	3
3. VARIABLES USED IN THE SALARY CHECK.....	6
4. DETERMINING THE OCCUPATIONS INVOLVED	7
5. CALCULATE HOURLY WAGES	8
6. ANNUAL INCREASE OF WAGES	21
7. PREPARE THE VARIABLES FOR THE REGRESSION.....	22
8. TEST VARIABLES PER OCCUPATION	24
9. REGRESSION PER OCCUPATION.....	26
10. PROFILES.....	29
11. FINALLY	31

1. INTRODUCTION

WOLIWEB addresses the impact of the socio-economic framework on attitudes, preferences, and perceptions. Attitudes, preferences, and perceptions are inherently subjective in nature. They are potentially influenced by a host of factors related to one's socio-economic framework, referring to concepts such as occupation; labour market status; earnings; working, household and leisure time; marital status and family phase; socio-economic status; gender and ethnical background. WOLIWEB aims for quantitative analyses, and the data needed are gathered through the international, continuous web-based *WageIndicator*, consisting of

- an attractive website with labour market related information for a large public;
- a crowd-pulling Salary Check providing with very detailed salary information related to a set of variables such as education, firm size, supervisory position;
- a *WageIndicator* questionnaire with 67 – 85 questions; the dataset provides insight in issues related to work and wages;
- nation wide promotion, publicity, and answering visitors' email.

This paper is WOLIWEB's Deliverable D22c, which is part of Work Package 5 'Synchronizing questionnaire and managing data collection' (p. 27 Annex 1). According to the List of deliverables, WOLIWEB should deliver the dataset-year 1, year 2, year 3, including the weights and coefficients for the Salary Checks. The CD-Rom includes the coefficients for the Salary Checks, made in the past years for the WOLIWEB countries. This report (D22c) details the calculation of the coefficients.

The *Salary Check* in the *WageIndicator* websites are critical for the number of webvisitors. Per country, the *Salary Check* provides reliable information about wages for many occupations. The *Salary Check* can be made when a country has approximately 8.000-10.000 completed questionnaires. This paper explains the operation of *Salary Check* in the *WageIndicator* website. It builds on a previous paper that explained the operation of *Salary Check* in the Dutch *WageIndicator* website.

2. THE SALARY CHECK SCREENS

The *Salary Check* in the *WageIndicator* websites consists of three-five subsequent screens. The number of screens varies per country. The maximum number of input screens is four: three screens with occupational information and one with personal questions. The *Salary Check* always starts with an input screen where the webvisitor can tick an occupation or, when the number of occupations is too large to put on one screen, an occupational group (see Table 1). In this case, the second screen is the input screen where the webvisitor can tick an occupation (see Table 2). In the third input screen, a number of questions are asked to identify the visitor's profile (see Table 3). The maximum number is ten. The tables show the screens of the Dutch Loonwijzer, because there is no English version available yet.

Table 1. *First input screen to identify the visitor's occupational group*

Kies je beroepsgroep
☐ (elektro) technicus, - monteur, elektricien
☐ (para) medisch, analist, verzorgend
☐ administratief beroep
☐ adviseur, consulent, consultant, voorlichter
☐ agrarisch beroep
☐ beveiliging, politie, defensie
☐ bouw
☐ catering, horeca, huishouding, schoonmaak
☐ docent, trainer, onderzoeker
☐ finance, bank, verzekering, belasting, accountant
☐ grafisch, media, pr, marketing, artistiek
☐ ict-beroep
☐ inkoper, verkoper, sales
☐ logistiek, transport, planner
☐ maatschappelijk werk, welzijn
☐ personeelswerk
☐ produktiemedewerker industrie
☐ secretaresse, secretaris
☐ staf, management, juridisch
☐ winkel

Table 2. Second input screen to identify the visitor's occupation

Selecteer een van de volgende beroepen
☐ webprogrammeur
☐ bouw, reparatie computer rand apparatuur
☐ applicatieprogrammeur
☐ computer-, systeemoperator
☐ onderhouds-, storingsmonteur, service-engineer software
☐ data-, mainframe-, server-, netwerkbeheerder
☐ databeheerder, -manager
☐ netwerk-, serverbeheerder
☐ helpdeskmedewerker
☐ systeem-, probleem-, informatie-analist
☐ systeemontwerper, -programmeur (techn)
☐ programmeur (techn)
☐ automatisering, informatica - monteur, technicus
☐ automatisering - hoofd afdeling, projectleider
☐ ict: systeemanalist, -ontwerper
☐ hoofd automatisering, rekencentrum
☐ programmeur niet technisch, geen applicatieprogrammeur
☐ ict: applicatie-, systeem-, netwerk-, serverbeheerder

Table 3. Third input screen to identify the visitor's profile

Selecteer je opleiding:
☐ Basisschool
☐ VBO/Leerlingwezen
☐ VMBO/MAVO/MULO
☐ HAVO/VWO/MBO
☐ HBO
☐ Universiteit

Hoeveel jaren werkervaring heb je in totaal?

0	
1	
2	

Heb je een leidinggevende positie?

☐ nee
☐ ja

Zijn je meeste collega's mannen?

☐ nee
☐ ja

The fourth screen is the output screen providing information about the gross wage per hour, per week, per 4 weeks, per month and per year for the chosen occupation (see Table 4). For the Dutch *Salary Check*, these calculations are based on a working week of 38 hours, but the screen allows clicking standard working hours from 36 to 40 and instantly the gross wage is recalculated.

Table 4. Output screen presenting gross wages for the chosen occupation.

Jouw collega's verdienen bruto gemiddeld	
Per uur	€ 9,91
Per week	€ 376,49
Per 4 weken	€ 1506
Per maand	€ 1630
Per jaar	€ 19578
Uren per week	
38	▼

3. VARIABLES USED IN THE SALARY CHECK

There are seven variables used in the Dutch *Salary Check* notably education, total years of service, supervisory position, predominantly male colleagues, re-entrant women, firm size, and promoted in current firm. *Education* and *years of service* are the typical human capital variables in the wage equations. Education is measured in 6 levels, and experience is measured in years in a range of 1 to 30. *Supervisory position* and *promoted in current firm* are variables reflecting wages according to the pay and grading systems used in most firms. The variable *predominantly male colleagues* is a proxy for inter-industry and inter-occupational wage differentials. Being a *re-entrant woman* reflects an effect of re-entry on wages beyond the effect of less years of service. *Firm size* reflects the commonly found wage differentials across firm sizes, which may reflect that a larger firm has more hierarchical levels and is therefore more likely to have employees in higher wage groups. These variables are all dichotomous. These variables can vary across countries, due to the fact that in some countries other variables have also an effect on wages. For example, in Germany the variable *region* is added because working in East or West Germany influence wages. The characteristics ticked in the third screen are inputs for the calculation rule (see Table 5). For any chosen occupation the *Salary Check* calculates instantly a gross wage per hour, week and month.

Table 5. The salary-rule for a clerical worker in manufacturing and building.

EXAMPLE: <salary-rule>

```
1.96570 + __education__ * 0.04153 + __experience__ * 0.03141
+ __experience__ * __experience__ * -0.00047 + __supervisoryjob__ * 0.06296
+ __coworkersAreMale__ * 0.01313 + __femaleReentrant__ * -0.04532
+ __hasBeenPromoted__ * 0.07521 + __lessThan100Empl__ * -0.03669
+ __moreThan500Empl__ * 0.08038
```

4. DETERMINING THE OCCUPATIONS INVOLVED

Any occupation with at least 50 observations will be used for the calculation of the wages. Based on a frequency table of ISCO9NUM (or OCC9ISCO) one can determine the occupations with 50 respondents or more. Comparable occupations with less than 50 respondents can be joined to a new occupation, if desired. Split occupations with large numbers of respondents. For example, the Dutch dataset counts more than 4,000 secretaries. This group is split into three: secretaries in manufacturing, secretaries in trade and services, and secretaries in the public sector. For other occupations, very detailed occupational titles can be grouped into one occupation, for example several categories of nurses can be grouped into one occupation, called 'nurse'.

5. CALCULATE HOURLY WAGES

The following SPSS syntax is for your information on how hourly wages are calculated for release 6. These variables are already in the dataset you receive quarterly.

***** ASSIGN CURRENCIES .

```
compute WAGECUR=wagecurr.
var lab WAGECUR 'Currency'.
form WAGECUR (f2).
add val lab WAGECUR 1 'PL Zlotty' 2 'EURO' 3 'GB Pound' 4 'US Dollar' 5 'DK Krone' 6 'BR
Cruzeiro' 7 'IN Rupee' 8 'ZA Rand'
9 'AR Peso' 10 'HU Forint' 11 'KR Won' 12 'MX Peso' 13 'TR Lira' 14 'NO Kroon' 15 'JP Yen' 16
'CN Renminbi'.
if (country = 616 and wagecurr=1) WAGECUR=1.
if (country = 616 and wagecurr=2) WAGECUR=2.
if (country = 616 and wagecurr=3) WAGECUR=3.
if (country = 616 and wagecurr=4) WAGECUR=4.
if (country = 616 and (sysmis(wagecurr) or missing(wagecurr))) WAGECUR=1.
if (country=56 or country=246 or country=276 or country=380 or country=528 or country=724 or
country=250) WAGECUR=2.
if (country=826 ) WAGECUR=3.
if (country=840 ) WAGECUR=4.
if (country=208 ) WAGECUR=5.
if (country=76 ) WAGECUR=6.
if (country=356 ) WAGECUR=7.
if (country=710 ) WAGECUR=8.
if (country=32 ) WAGECUR=9.
if (country=348 ) WAGECUR=10.
if (country=410 ) WAGECUR=11.
if (country=484 ) WAGECUR=12.
if (country=792 ) WAGECUR=13.
if (country=578 ) WAGECUR=14.
if (country=392 ) WAGECUR=15.
if (country=156 ) WAGECUR=16.
```

***** COMPUTE WAGED HOURS .

```
compute HRSWAG1= SYSMIS.
var lab HRSWAG1 'Working hours a week used for calculating hourly wages'.
format HRSWAG1 (f4.2).
mis val HRSWAG1 (-9 thru -1).

if (missing(HRSCON) and missing(HRSFIR) and missing(HRSREA) and missing(HRSWAG))
HRSWAG1=-9.
execute.
if (sysmis(HRSWAG1) and missing(HRSCON) and missing(HRSFIR) and missing(HRSREA) and
HRSWAG>0) HRSWAG1=HRSWAG.
execute.
if (sysmis(HRSWAG1) and missing(HRSCON) and missing(HRSFIR) and HRSREA>0 and
missing(HRSWAG)) HRSWAG1=HRSREA.
execute.
```

if (sysmis(HRSWAG1) and missing(HRSCON) and HRSFIR>0 and missing(HRSREA) and missing(HRSWAG)) HRSWAG1=HRSFIR.
execute.

if (sysmis(HRSWAG1) and HRSCON>0 and missing(HRSFIR) and missing(HRSREA) and missing(HRSWAG)) HRSWAG1=HRSCON.
execute.

if (sysmis(HRSWAG1) and HRSCON>0 and HRSCON=HRSWAG) HRSWAG1=HRSCON.
execute.

if (sysmis(HRSWAG1) and HRSCON>0 and HRSCON=HRSREA and missing(HRSWAG)) HRSWAG1=HRSCON.
execute.

if (sysmis(HRSWAG1) and HRSCON>0 and HRSCON=HRSFIR and missing(HRSWAG)) HRSWAG1=HRSCON.
execute.

if (sysmis(HRSWAG1) and HRSCON>0 and missing(HRSFIR) and HRSREA>0 and HRSREA=HRSWAG) HRSWAG1=HRSWAG.
execute.

if (sysmis(HRSWAG1) and HRSCON>0 and HRSFIR=HRSREA and missing(HRSWAG)) HRSWAG1=HRSREA.
execute.

if (sysmis(HRSWAG1) and missing(HRSCON) and HRSFIR>0 and HRSFIR=HRSREA and missing(HRSWAG)) HRSWAG1=HRSFIR.
execute.

if (sysmis(HRSWAG1) and missing(HRSCON) and HRSFIR>0 and HRSFIR=HRSWAG) HRSWAG1=HRSWAG.
execute.

if (sysmis(HRSWAG1) and missing(HRSCON) and missing(HRSFIR) and HRSREA=HRSWAG and HRSWAG>0) HRSWAG1=HRSWAG.
execute.

if (sysmis(HRSWAG1) and missing(HRSCON) and missing(HRSFIR) and HRSREA>HRSWAG and (hrshisto=1 or hrscont4=1) and HRSWAG<10) HRSWAG1=HRSREA.
execute.

if (sysmis(HRSWAG1) and missing(HRSCON) and missing(HRSFIR) and HRSREA>HRSWAG and (hrshisto=0 or hrscont4=2)) HRSWAG1=HRSWAG.
execute.

if (sysmis(HRSWAG1) and missing(HRSCON) and missing(HRSFIR) and HRSREA<HRSWAG) HRSWAG1=HRSWAG.
execute.

if (sysmis(HRSWAG1) and missing(HRSCON) and missing(HRSREA) and HRSFIR>HRSWAG) HRSWAG1=HRSWAG.
execute.

if (sysmis(HRSWAG1) and missing(HRSCON) and missing(HRSREA) and HRSFIR<HRSWAG) HRSWAG1=HRSWAG.
execute.

if (sysmis(HRSWAG1) and HRSCON=1 and HRSFIR=1 and HRSREA=1 and HRSWAG>1) HRSWAG1=HRSWAG.
execute.

if (sysmis(HRSWAG1) and HRSCON=1 and HRSFIR=1 and HRSREA=HRSWAG) HRSWAG1=HRSWAG.
execute.

***** CONTROLLING FOR PAID OVERTIME HOURS.**

if ((wageotim=1 or wageotim=2) and hrsreal0=3 and HRSCON=HRSWAG and HRSREA<(1.1*HRSCON)) HRSWAG1=HRSREA.
execute.

***** CONTROLLING FOR DAY HOURS INSTEAD OF WEEK HOURS FOR FULLTIMERS.**

if (sysmis(HRSWAG1) and (hrshisto=1 or hrscont4=1) and HRSCON<10 and HRSFIR<10 and (wageotim=1 or wageotim=2) and hrsreal0=3 and HRSREA>0) HRSWAG1=HRSREA.
execute.

if (sysmis(HRSWAG1) and (hrshisto=1 or hrscont4=1) and HRSCON<10 and HRSFIR<10 and HRSWAG>0) HRSWAG1=HRSWAG.
execute.

if (sysmis(HRSWAG1) and (hrshisto=1 or hrscont4=1) and HRSCON<10 and HRSFIR<10 and HRSWAG>0) HRSWAG1=HRSWAG.
execute.

if (sysmis(HRSWAG1) and (hrshisto=1 or hrscont4=1) and HRSCON=HRSFIR and HRSFIR=HRSREA and HRSWAG<10) HRSWAG1=HRSFIR.
execute.

if ((hrshisto=1 or hrscont4=1) and missing(HRSCON) and missing(HRSREA) and HRSFIR>10 and HRSWAG<10) HRSWAG1=HRSFIR.
execute.

***** CONTROLLING FOR HOURS FOR PARTTIMERS.**

if (sysmis(HRSWAG1) and (hrshisto=0 or hrscont4=2) and HRSFIR=HRSREA and (wageotim=1 or wageotim=2) and hrsreal0=3 and HRSREA>0) HRSWAG1=HRSREA.
execute.

if (sysmis(HRSWAG1) and (hrshisto=0 or hrscont4=2) and HRSFIR=HRSREA and HRSWAG>0) HRSWAG1=HRSWAG.
execute.

***** CONTROLLING FOR MISTAKES IN HRSWAG.**

if (sysmis(HRSWAG1) and HRSCON=HRSFIR and HRSFIR=HRSREA and HRSWAG>HRSREA) HRSWAG1=HRSREA.
execute.

if (sysmis(HRSWAG1) and HRSCON=HRSFIR and HRSFIR=HRSREA and HRSWAG<HRSREA) HRSWAG1=HRSREA.
execute.

if (sysmis(HRSWAG1) and HRSCON=HRSFIR and HRSREA=HRSWAG and HRSWAG>HRSCON) HRSWAG1=HRSREA.
execute.

if (sysmis(HRSWAG1) and HRSCON=HRSFIR and HRSREA=HRSCON and HRSWAGe>0 and HRSCON<(1.1*HRSWAG)) HRSWAG1=HRSWAG.
execute.

***** FILLING ODD VALUES.**

if (sysmis(HRSWAG1) and missing(HRSCON) and missing(HRSFIR) and HRSREA>0 and HRSWAG>0 and HRSREA>HRSWAG and HRSWAG>30) HRSWAG1=HRSWAG.
execute.

if (sysmis(HRSWAG1) and hrshisto=1 and HRSWAG<11 and HRSCON>0) HRSWAG1=HRSCON.
execute.

if (sysmis(HRSWAG1) and hrshisto=1 and HRSWAG<11 and HRSFIR>0) HRSWAG1=HRSFIR.
execute.

```
if (sysmis(HRSWAG1) and hrshisto=1 and HRSWAG<11 and HRSCON>0)
HRSWAG1=HRSCON.
execute.
if (sysmis(HRSWAG1) and hrshisto=1 and HRSWAG<11 and HRSREA>0)
HRSWAG1=HRSREA.
execute.
if (sysmis(HRSWAG1) and hrshisto=0 and HRSWAG>0 ) HRSWAG1=HRSWAG.
execute.
if (sysmis(HRSWAG1) and hrshisto=0 and HRSFIR>0) HRSWAG1=HRSFIR.
execute.
if (sysmis(HRSWAG1) and hrshisto=0 and HRSCON>0) HRSWAG1=HRSCON.
execute.
if (sysmis(HRSWAG1) and hrshisto=0 and HRSREA>0) HRSWAG1=HRSREA.
execute.
if (sysmis(HRSWAG1) and HRSFIR>0 and HRSREA>0 and HRSWAG>0)
HRSWAG1=HRSWAG.
execute.
if (sysmis(HRSWAG1) and missing(HRSFIR) and HRSREA>0 and HRSWAG>0)
HRSWAG1=HRSWAG.
execute.
if (sysmis(HRSWAG1) and HRSCON>0 and missing(HRSFIR) and missing(HRSREA) and
HRSWAG<HRSCON+5 and HRSWAG>HRSCON-5 ) HRSWAG1=HRSWAG.
execute.
if (sysmis(HRSWAG1) and HRSCON<10 and missing(HRSFIR) and missing(HRSREA) and
HRSWAG>30 and hrshisto=1) HRSWAG1=HRSWAG.
execute.
if (sysmis(HRSWAG1) and HRSCON>0 and missing(HRSFIR) and missing(HRSREA) and
HRSCON<HRSWAG and hrshisto=1) HRSWAG1=HRSCON.
execute.
if (sysmis(HRSWAG1) and HRSCON>0 and missing(HRSFIR) and missing(HRSREA) and
HRSCON>HRSWAG and hrshisto=1) HRSWAG1=HRSWAG.
execute.
if (sysmis(HRSWAG1) and missing(HRSCON) and HRSFIR>0 and missing(HRSREA) and
HRSWAG<HRSFIR+6 and HRSWAG>HRSFIR-6 ) HRSWAG1=HRSWAG.
execute.
if (sysmis(HRSWAG1) and HRSCON>0 and HRSWAG<HRSCON+5 and HRSWAG>HRSCON-
5 ) HRSWAG1=HRSWAG.
execute.
if (sysmis(HRSWAG1) and HRSCON>0 and HRSREA<HRSCON+5 and HRSREA>HRSCON-5
and missing(HRSWAG)) HRSWAG1=HRSCON.
execute.
if (sysmis(HRSWAG1) and missing(HRSCON) and HRSREA<HRSFIR+5 and
HRSREA>HRSFIR-5 and missing(HRSWAG)) HRSWAG1=HRSFIR.
execute.
if (sysmis(HRSWAG1) and missing(HRSCON) and HRSFIR>40 and HRSWAG>0 and
missing(HRSREA)) HRSWAG1=HRSWAG.
execute.
if (sysmis(HRSWAG1) and missing(HRSCON) and HRSFIR<=40 and HRSWAG>0 and
missing(HRSREA)) HRSWAG1=HRSWAG.
execute.
if (sysmis(HRSWAG1) and HRSCON<10 and hrshisto=1 and HRSWAG>30)
HRSWAG1=HRSWAG.
execute.
if (sysmis(HRSWAG1) and HRSCON>30 and missing(HRSFIR) and HRSREA>0 and
missing(HRSWAG )) HRSWAG1=HRSCON.
execute.
if (sysmis(HRSWAG1) and HRSFIR=HRSWAG) HRSWAG1=HRSWAG.
execute.
```

*** FILLING THE MISSING VALUES.

if ((sysmis(HRSWAG1) or missing(HRSWAG1)) and hrshisto=1 and HRSFIR>1)
HRSWAG1=HRSFIR.

execute.

if ((sysmis(HRSWAG1) or missing(HRSWAG1)) and hrshisto=0 and HRSREA>0)
HRSWAG1=HRSREA.

execute.

if ((sysmis(HRSWAG1) or missing(HRSWAG1)) and hrshisto=1) HRSWAG1=40.
execute.

if (HRSWAG1>100) HRSWAG1=SYSMIS.
des HRSWAG1.

***** COMPUTE GROSS and NET WAGE, USED FOR CALCULATING HOURLY WAGE.

comp WAGEGR1= WAGEGR.

comp WAGENE1=WAGENE .

var lab WAGEGR1 'Last gross wages in nat currency NOT CONTROLLED FOR PAYPERIOD'.

var lab WAGENE1 'Last nett wages in nat currency NOT CONTROLLED FOR PAYPERIOD'.

format WAGEGR1 WAGENE1 (f10.2).

missing val WAGEGR1 WAGENE1 (-1, -9).

*** CHECK FOR WAGEGR1 < WAGENE1.

comp SYSMISA=SYSMIS.

comp SYSMISB=SYSMIS.

execute.

if (WAGEGR1 < WAGENE1) SYSMISA = WAGEGR1.

if (WAGEGR1 < WAGENE1) SYSMISB = WAGENE1.

execute.

if (SYSMISA > 0) WAGENE1 = SYSMISA.

if (SYSMISB > 0) WAGEGR1 = SYSMISB.

execute.

*** ASSIGN TEMPORARY GROSS WAGE = NETT WAGE WHEN GROSS WAGE IS MISSING

if (missing(WAGEGR) and WAGENE > 0) WAGEGR1 = WAGENE.

execute.

*** TO CHECK WAGEPERIOD ASSIGN TEMPORARY GROSS WAGE = EURO LEVEL FOR
NON-EURO COUNTRIES BASED ON EXCHANGE RATE

*** val lab WAGECUR 1 'PL Zlotty' 2 'EURO' 3 'GB Pound' 4 'US Dollar' 5 'DK Krone' 6 'BR
Cruzeiro' 7 'IN Rupee' 8 'ZA Rand' 9 'AR Peso' 10 'HU Forint' 11 'KR Won' 12 'MX Peso' 13 'TR
Lira' 14 'NO Kroon' 15 'JP Yen' 16 'CN Renminbi'.

if (WAGECUR =1) WAGEGR1= 0.25896 * WAGEGR1.

if (WAGECUR =2) WAGEGR1= 1 * WAGEGR1.

if (WAGECUR =3) WAGEGR1= 1.45275 * WAGEGR1.

if (WAGECUR =4) WAGEGR1= 0.68835 * WAGEGR1.

if (WAGECUR =5) WAGEGR1= 0.13403 * WAGEGR1.

if (WAGECUR =6) WAGEGR1= 0.36306 * WAGEGR1.

if (WAGECUR =7) WAGEGR1= 0.01885 * WAGEGR1.

if (WAGECUR =8) WAGEGR1= 0.13397 * WAGEGR1.

*** ASSIGN TEMPORARY NETT WAGE = EURO LEVEL FOR NON-EURO COUNTRIES.

if (WAGECUR =1) WAGENE1= 0.25896 * WAGENE1.

if (WAGECUR =2) WAGENE1= 1 * WAGENE1.

if (WAGECUR =3) WAGENE1= 1.45275 * WAGENE1.

if (WAGECUR =4) WAGENE1= 0.68835 * WAGENE1.

if (WAGECUR =5) WAGENE1= 0.13403 * WAGENE1.

if (WAGECUR =6) WAGENE1= 0.36306 * WAGENE1.

```
if (WAGECUR =7) WAGENE1= 0.01885 * WAGENE1.  
if (WAGECUR =8) WAGENE1= 0.13397 * WAGENE1.
```

```
***** COMPUTE WAGEPER3, USED FOR CALCULATING HOURLY WAGE.
```

```
comp WAGEPER3=wageperi.  
var lab WAGEPER3 'Wage period CHECKED'.  
for WAGEPER3 (F2).  
mis val WAGEPER3 (-1,-9).  
val lab WAGEPER3  
1 '1 calendar month'  
2 '4 weeks'  
3 '2 weeks'  
4 '1 week'  
5 '1 day'  
6 '1 hour'  
7 '1 year'  
10 '2 calendar months'  
11 '3 calendar months'  
-1 ' Not (const > 9)'  
-9 'User missing'.
```

```
***** CHECK FOR RELIABLE GROSS EARNINGS CONSIDERING WAGEPERIOD.
```

```
*** EARNINGS EURO > 150000 ARE CONSIDERED ANNUAL EARNINGS.  
if ( WAGEGR1 > 150000 ) WAGEPER3=7.  
execute.
```

```
*** EARNINGS EURO < 10 ARE CONSIDERED HOURLY EARNINGS.  
if ( WAGEGR1 <=10 ) WAGEPER3=6.  
execute.
```

```
*** THESE EARNINGS ARE CONSIDERED MONTHLY IF WAGEPER3 IS MISSING.  
if (missing (WAGEPER3) and HRSWAG1>30 and HRSWAG1<=45 and WAGEGR1>1000 and  
WAGEGR1<=5000) WAGEPER3=1.  
execute.
```

```
*** THESE EARNINGS ARE CONSIDERED 3-MONTHLY IF WAGEPER3 IS MISSING OR <7.  
if ((missing (WAGEPER3) or WAGEPER3<7) and HRSWAG1>30 and HRSWAG1<=45 and  
WAGEGR1>15000 and WAGEGR1<=45000) WAGEPER3=11.  
execute.
```

```
*** THESE EARNINGS ARE CONSIDERED ANNUAL IF WAGEPER3 IS MISSING OR <7.  
if ((missing (WAGEPER3) or WAGEPER3<7) and HRSWAG1>30 and HRSWAG1<=45 and  
WAGEGR1>45000) WAGEPER3=7.  
execute.
```

```
*** EARNINGS EURO 50 >< 600 ARE CONSIDERED WEEKLY WHEN REPORTED HOURLY.  
if (WAGEGR1>50 and WAGEGR1<=600 and WAGEPER3 = 6) WAGEPER3=4.  
execute.
```

```
*** EARNINGS EURO 600 ><1200 ARE CONSIDERED 2 WEEKs WHEN REPORTED  
<=WEEKLY.  
if ( WAGEGR1>600 and WAGEGR1<=1200 and (WAGEPER3 = 5 or WAGEPER3 = 6))  
WAGEPER3=3.  
execute.
```

```
*** EARNINGS EURO > 1200 ARE CONSIDERED 4 WEEKs WHEN REPORTED <=2 WEEKs.  
if (WAGEGR1>1200 and (WAGEPER3 = 4 or WAGEPER3 = 5 or WAGEPER3 = 6))  
WAGEPER3=2.
```

execute.

*** WHEN PAY PERIOD STILL MISSING, ASSIGN MONTHLY PAY PERIOD.

if (missing(WAGEPER3) and WAGEGR1>0) WAGEPER3=1.

*** UNDO TEMPORARY GROSS WAGE = NETT WAGE WHEN GROSS WAGE IS MISSING.

if (missing(WAGEGR) and WAGENE > 0) WAGEGR1 = SYSMIS.

execute.

***** COMPUTE AGAIN GROSS and NET WAGE, USED FOR CALCULATING HOURLY WAGE

comp WAGEGR1= WAGEGR.

comp WAGENE1=WAGENE .

var lab WAGEGR1 'Last gross wages in nat currency NOT CONTROLLED FOR PAYPERIOD'.

var lab WAGENE1 'Last nett wages in nat currency NOT CONTROLLED FOR PAYPERIOD'.

format WAGEGR1 WAGENE1 (f10.2).

missing val WAGEGR1 WAGENE1 (-1, -9).

*** CHECK FOR WAGEGR1 < WAGENE1.

comp SYSMISA=SYSMIS.

comp SYSMISB=SYSMIS.

execute.

if (WAGEGR1 < WAGENE1) SYSMISA = WAGEGR1.

if (WAGEGR1 < WAGENE1) SYSMISB = WAGENE1.

execute.

if (SYSMISA > 0) WAGENE1 = SYSMISA.

if (SYSMISB > 0) WAGEGR1 = SYSMISB.

execute.

des WAGENE1 WAGEGR1.

*** CHECK FOR EXTREME VALUES.

if (WAGEGR1>1000000 and (wagecur=2 or wagecur=3 or wagecur=4)) WAGENE1=SYSMIS.

if (WAGEGR1>1000000 and (wagecur=2 or wagecur=3 or wagecur=4)) WAGEGR1=SYSMIS.

des WAGEGR1 WAGENE1 WAGEPER3.

***** CONTROL FOR WAGESUM - BENEFITS & ALLOWANCES, BASED ON WASUM.

comp WASUM1= WASUM.

var lab WASUM1 'Sum benefits, used for calculating hourly wages'.

form WASUM1 (f12.2).

if (WASUM1>0 and WASUM1>0.66*WAGEGR1) WASUM1=SYSMIS .

execute.

do if (WASUM1>0).

comp WAGEGR1=WAGEGR1-WASUM1.

comp WAGENE1=WAGENE1-WASUM1.

if (WAGENE1<0) WAGENE1=0.

end if.

execute.

des WASUM1.

***** COMPUTE HOURLY WAGES 1ST TIME.

compute WAGEGRHR=SYSMIS .

compute WAGENEHR= SYSMIS .

format WAGEGRHR WAGENEHR (f8.2).

var lab WAGEGRHR 'Hourly gross wage in national currency'.

var lab WAGENEHR 'Hourly nett wage in national currency'.

mis val WAGEGRHR WAGENEHR (-1,-9).

```

do if (HRSWAG1 > 0 and WAGEGR1 > 0 ).
if (WAGEPER3 = 1 ) WAGEGRHR=WAGEGR1/(4.33*HRSWAG1).
if (WAGEPER3 = 2 ) WAGEGRHR=WAGEGR1/(4*HRSWAG1).
if (WAGEPER3 = 3 ) WAGEGRHR=WAGEGR1/(2*HRSWAG1).
if (WAGEPER3 = 4 ) WAGEGRHR=WAGEGR1/(HRSWAG1).
if (WAGEPER3 = 6 ) WAGEGRHR=WAGEGR1.
if (WAGEPER3 = 7 ) WAGEGRHR=WAGEGR1/(4.33*HRSWAG1*12).
if (WAGEPER3 = 10) WAGEGRHR=WAGEGR1/(4.33*HRSWAG1*2).
if (WAGEPER3 = 11) WAGEGRHR=WAGEGR1/(4.33*HRSWAG1*3).
end if.
execute.

```

```

do if (HRSWAG1 > 0 and WAGENE1 > 0 ).
if (WAGEPER3 = 1 ) WAGENEHR=WAGENE1/(4.33*HRSWAG1).
if (WAGEPER3 = 2 ) WAGENEHR=WAGENE1/(4*HRSWAG1).
if (WAGEPER3 = 3 ) WAGENEHR=WAGENE1/(2*HRSWAG1).
if (WAGEPER3 = 4 ) WAGENEHR=WAGENE1/(HRSWAG1).
if (WAGEPER3 = 6 ) WAGENEHR=WAGENE1.
if (WAGEPER3 = 7 ) WAGENEHR=WAGENE1/(4.33*HRSWAG1*12).
if (WAGEPER3 = 10) WAGENEHR=WAGENE1/(4.33*HRSWAG1*2).
if (WAGEPER3 = 11) WAGENEHR=WAGENE1/(4.33*HRSWAG1*3).
end if.
execute.

```

```

do if (HRSWAG1 > 0 and WAGEGR1 > 0 and hrsdayp2 > 0).
if (WAGEPER3 = 5 ) WAGEGRHR=WAGEGR1/(HRSWAG1/hrsdayp2).
if (WAGEPER3 = 5 ) WAGENEHR=WAGENE1/(HRSWAG1/hrsdayp2).
end if.
execute.

```

***** CHECK FOR ODD RATIOS GROSS AND NETT WAGES.

*** CHECK GROSS WAGE > 3 * NETT WAGE.

```

if (WAGEGR1 >= 3*WAGENE1 and WAGENE1 > 0 and WAGENEHR>10 and
(WAGEGR1/10)>=WAGENE1 and (WAGEGR1/100)<WAGENE1) WAGEGR1=WAGEGR1/10.
execute.

```

```

if (WAGEGR1 >= 3*WAGENE1 and WAGENE1 > 0 and WAGENEHR>10 and
(WAGEGR1/100)>=WAGENE1 ) WAGEGR1=WAGEGR1/100.
execute.

```

```

if (WAGEGR1 >= 3*WAGENE1 and WAGENE1 > 0 and WAGENEHR<=10 and
(WAGEGR1/10)>=WAGENE1 and (WAGEGR1/100)<WAGENE1) WAGENE1=WAGENE1*10.
execute.

```

```

if (WAGEGR1 >= 3*WAGENE1 and WAGENE1 > 0 and WAGENEHR<=10 and
(WAGEGR1/100)>=WAGENE1) WAGENE1=WAGENE1*100.
execute.

```

```

if (WAGEGR1 >= 3*WAGENE1 ) WAGEGR1=sysmis.

```

***** COMPUTE HOURLY WAGES FINAL.

```

compute WAGEGRHR=SYSMIS .

```



```

compute WAGENEHR= SYSMIS .
format WAGEGRHR WAGENEHR (f8.2).
var lab WAGEGRHR 'Hourly gross wage in national currency'.
var lab WAGENEHR 'Hourly nett wage in national currency'.
mis val WAGEGRHR WAGENEHR (-1,-9).

do if (HRSWAG1 > 0 and WAGEGR1 >0 ).
if (WAGEPER3 = 1 ) WAGEGRHR=WAGEGR1/(4.33*HRSWAG1).
if (WAGEPER3 = 2 ) WAGEGRHR=WAGEGR1/(4*HRSWAG1).
if (WAGEPER3 = 3 ) WAGEGRHR=WAGEGR1/(2*HRSWAG1).
if (WAGEPER3 = 4 ) WAGEGRHR=WAGEGR1/(HRSWAG1).
if (WAGEPER3 = 6 ) WAGEGRHR=WAGEGR1.
if (WAGEPER3 = 7 ) WAGEGRHR=WAGEGR1/(4.33*HRSWAG1*12).
if (WAGEPER3 = 10) WAGEGRHR=WAGEGR1/(4.33*HRSWAG1*2).
if (WAGEPER3 = 11) WAGEGRHR=WAGEGR1/(4.33*HRSWAG1*3).
end if.
execute.

do if (HRSWAG1 > 0 and WAGENE1 > 0 ).
if (WAGEPER3 = 1 ) WAGENEHR=WAGENE1/(4.33*HRSWAG1).
if (WAGEPER3 = 2 ) WAGENEHR=WAGENE1/(4*HRSWAG1).
if (WAGEPER3 = 3 ) WAGENEHR=WAGENE1/(2*HRSWAG1).
if (WAGEPER3 = 4 ) WAGENEHR=WAGENE1/(HRSWAG1).
if (WAGEPER3 = 6 ) WAGENEHR=WAGENE1.
if (WAGEPER3 = 7 ) WAGENEHR=WAGENE1/(4.33*HRSWAG1*12).
if (WAGEPER3 = 10) WAGENEHR=WAGENE1/(4.33*HRSWAG1*2).
if (WAGEPER3 = 11) WAGENEHR=WAGENE1/(4.33*HRSWAG1*3).
end if.
execute.

do if (HRSWAG1 > 0 and WAGEGR1 >0 and hrsdayp2 >0).
if (WAGEPER3 = 5 ) WAGEGRHR=WAGEGR1/(HRSWAG1/hrsdayp2).
if (WAGEPER3 = 5 ) WAGENEHR=WAGENE1/(HRSWAG1/hrsdayp2).
end if.
execute.

***** COMPUTE GROSS WAGES FOR OBS WITH NETT WAGES ONLY (PER COUNTRY).

comp WAGEDIFF= SYSMIS .
var lab WAGEDIFF 'WAGEGRHR / WAGENEHR '.
form WAGEDIFF (f4.2).
if (WAGEGRHR>0 and WAGENEHR>0) WAGEDIFF=WAGEGRHR / WAGENEHR .

means WAGEDIFF by country.

** assign the mean wage difference per country, derived from means WAGEDIFF by country.

do if (release<6).
if (SYSMIS (WAGEGRHR) and WAGENEHR>0 and country= 56) WAGEGRHR=WAGENEHR *
1.600026053 .
if (SYSMIS (WAGEGRHR) and WAGENEHR>0 and country= 208 )WAGEGRHR=WAGENEHR *
1.452306427 .
if (SYSMIS (WAGEGRHR) and WAGENEHR>0 and country= 246 )WAGEGRHR=WAGENEHR *
1.621979862 .
if (SYSMIS (WAGEGRHR) and WAGENEHR>0 and country= 276 )WAGEGRHR=WAGENEHR *
1.457264569 .
if (SYSMIS (WAGEGRHR) and WAGENEHR>0 and country= 528 )WAGEGRHR=WAGENEHR
1.543768068 .

```

```
if (SYSMIS (WAGEGRHR) and WAGENEHR>0 and country= 724 )WAGEGRHR=WAGENEHR *  
1.294129797  
if (SYSMIS (WAGEGRHR) and WAGENEHR>0 and country= 826 )WAGEGRHR=WAGENEHR *  
1.390037731  
end if.
```

```
des WAGEDIFF.
```

```
*** DELETE EXTREME HOURLY WAGES 0.5% and 99.5% per country.
```

```
*** BELGIUM / BELGIE.
```

```
USE ALL.  
COMPUTE filter_$=(country=56).  
VARIABLE LABEL filter_$ 'country=56 (FILTER)'.  
VALUE LABELS filter_$ 0 'Not Selected' 1 'Selected'.  
FORMAT filter_$ (f1.0).  
FILTER BY filter_$.  
EXECUTE .
```

```
FREQUENCIES VARIABLES=WAGEGRHR /FORMAT=NOTABLE /PERCENTILES= 0.5 99.5  
/ORDER= ANALYSIS .  
if ( WAGEGRHR <1.00 or WAGEGRHR>121.259122) WAGENEHR=SYSMIS.  
if ( WAGEGRHR <1.00 or WAGEGRHR>121.259122) WAGEGRHR=SYSMIS.  
EXECUTE.
```

```
*** DENMARK / DENEMARKEN..
```

```
USE ALL.  
COMPUTE filter_$=(country=208).  
VARIABLE LABEL filter_$ 'country=208 (FILTER)'.  
VALUE LABELS filter_$ 0 'Not Selected' 1 'Selected'.  
FORMAT filter_$ (f1.0).  
FILTER BY filter_$.  
EXECUTE .
```

```
*** FREQUENCIES VARIABLES=WAGEGRHR /FORMAT=NOTABLE /PERCENTILES= 0.5 99  
/ORDER= ANALYSIS .  
*** if ( WAGEGRHR <1.07991759 or WAGEGRHR>223.244994) WAGENEHR=SYSMIS.  
***EXECUTE.  
***if ( WAGEGRHR <1.07991759 or WAGEGRHR>223.244994) WAGEGRHR=SYSMIS.  
***EXECUTE.
```

```
*** FINLAND / FINLAND.
```

```
USE ALL.  
COMPUTE filter_$=(country=246).  
VARIABLE LABEL filter_$ 'country=246 (FILTER)'.  
VALUE LABELS filter_$ 0 'Not Selected' 1 'Selected'.  
FORMAT filter_$ (f1.0).  
FILTER BY filter_$.  
EXECUTE .
```

```
FREQUENCIES VARIABLES=WAGEGRHR /FORMAT=NOTABLE /PERCENTILES= 0.5 99.5  
/ORDER= ANALYSIS .  
if ( WAGEGRHR <2.95000406or WAGEGRHR>85.27306208597) WAGENEHR=SYSMIS.  
if ( WAGEGRHR < 2.95000406 or WAGEGRHR>85.27306208597) WAGEGRHR=SYSMIS.
```

EXECUTE.

*** GERMANY / DUTSLAND.

USE ALL.
COMPUTE filter_\$=(country=276).
VARIABLE LABEL filter_\$ 'country=276 (FILTER)'.
VALUE LABELS filter_\$ 0 'Not Selected' 1 'Selected'.
FORMAT filter_\$ (f1.0).
FILTER BY filter_\$.
EXECUTE .

FREQUENCIES VARIABLES=WAGEGRHR /FORMAT=NOTABLE /PERCENTILES= 0.5 99.5
/ORDER= ANALYSIS .
if (WAGEGRHR <3.459251184247 or WAGEGRHR>72.1709006) WAGENEHR=SYSMIS.
if (WAGEGRHR <3.459251184247 or WAGEGRHR>72.1709006) WAGEGRHR=SYSMIS.
EXECUTE.

*** ITALY / ITALIE.

USE ALL.
COMPUTE filter_\$=(country=380).
VARIABLE LABEL filter_\$ 'country=380 (FILTER)'.
VALUE LABELS filter_\$ 0 'Not Selected' 1 'Selected'.
FORMAT filter_\$ (f1.0).
FILTER BY filter_\$.
EXECUTE .

*** FREQUENCIES VARIABLES=WAGEGRHR /FORMAT=NOTABLE /PERCENTILES= 0.5
99.5 /ORDER= ANALYSIS .
*** if (WAGEGRHR <1.07991759 or WAGEGRHR>223.244994) WAGENEHR=SYSMIS.
*** if (WAGEGRHR <1.07991759 or WAGEGRHR>223.244994) WAGEGRHR=SYSMIS.
*** EXECUTE.

*** NETHERLANDS / NEDERLAND.

USE ALL.
COMPUTE filter_\$=(country=528).
VARIABLE LABEL filter_\$ 'country=528 (FILTER)'.
VALUE LABELS filter_\$ 0 'Not Selected' 1 'Selected'.
FORMAT filter_\$ (f1.0).
FILTER BY filter_\$.
EXECUTE .

FREQUENCIES VARIABLES=WAGEGRHR /FORMAT=NOTABLE /PERCENTILES= 0.5 99.5
/ORDER= ANALYSIS .
if (WAGEGRHR <1 or WAGEGRHR>125) WAGENEHR=SYSMIS.
if (WAGEGRHR <1 or WAGEGRHR>125) WAGEGRHR=SYSMIS.
EXECUTE.

*** POLAND / POLEN.

```
USE ALL.
COMPUTE filter_$=(country=616).
VARIABLE LABEL filter_$ 'country=616 (FILTER)'.
VALUE LABELS filter_$ 0 'Not Selected' 1 'Selected'.
FORMAT filter_$ (f1.0).
FILTER BY filter_$.
EXECUTE .
```

```
FREQUENCIES VARIABLES=WAGEGRHR /FORMAT=NOTABLE /PERCENTILES= 0.5 99.5
/ORDER= ANALYSIS .
if ( WAGEGRHR <1.0396 or WAGEGRHR>323.7194) WAGENEHR=SYSMIS.
if ( WAGEGRHR <1.0396 or WAGEGRHR>323.7194) WAGEGRHR=SYSMIS.
EXECUTE.
```

*** SPAIN / SPANJE.

```
USE ALL.
COMPUTE filter_$=(country=724).
VARIABLE LABEL filter_$ 'country=724 (FILTER)'.
VALUE LABELS filter_$ 0 'Not Selected' 1 'Selected'.
FORMAT filter_$ (f1.0).
FILTER BY filter_$.
EXECUTE .
```

```
FREQUENCIES VARIABLES=WAGEGRHR /FORMAT=NOTABLE /PERCENTILES= 0.5 99
/ORDER= ANALYSIS .
if ( WAGEGRHR <1.3410 or WAGEGRHR>80.8691) WAGENEHR=SYSMIS.
if ( WAGEGRHR <1.3410 or WAGEGRHR>80.8691) WAGEGRHR=SYSMIS.
EXECUTE.
```

*** UNITED KINGDOM / ENGELAND.

```
USE ALL.
COMPUTE filter_$=(country=826).
VARIABLE LABEL filter_$ 'country=826 (FILTER)'.
VALUE LABELS filter_$ 0 'Not Selected' 1 'Selected'.
FORMAT filter_$ (f1.0).
FILTER BY filter_$.
EXECUTE .
```

```
FREQUENCIES VARIABLES=WAGEGRHR /FORMAT=NOTABLE /PERCENTILES= 0.5 99
/ORDER= ANALYSIS .
if ( WAGEGRHR <1.8647 or WAGEGRHR>63.0495) WAGENEHR=SYSMIS.
if ( WAGEGRHR <1.8647 or WAGEGRHR>63.0495) WAGEGRHR=SYSMIS.
EXECUTE.
```

*** ADD NEW COUNTRIES IN RELEASE 6.

```
FILTER OFF.
USE ALL.
EXECUTE .
```

```
*** COMPUTE WEEKLY WAGES.
compute WAGEGRWE=(WAGEGRHR*HRSWAG1).
format WAGEGRWE (f8.2).
var lab WAGEGRWE 'Weekly gross wage in national cur (NOT standardised for hrs)'.
execute.
```

des WAGEGRWE.

```
*** COMPUTE MONTHLY WAGES.
compute WAGEGRMO=(WAGEGRHR*HRSWAG1*4.33).
format WAGEGRMO (f8.2).
var lab WAGEGRMO 'Monthly gross wage in national currency (NOT standardised for hrs)'.
execute.
```

des WAGEGRMO.

```
*** COMPUTE ANNUAL WAGES.
compute WAGEGRAN=(WAGEGRWE*52).
format WAGEGRAN (f8.2).
var lab WAGEGRAN 'Annual gross wage in national cur (NOT standardised for hrs)'.
execute.
```

des WAGEGRAN.

```
*** LOGARITM OF GROSS HOURLY WAGE *****.
```

```
compute WAGEGRHL = LN(WAGEGRHR) .
var lab WAGEGRHL 'Log hourly gross wage in national currency'.
execute.
```

des WAGEGRHL.

```
*** COMPUTE HOURLY WAGES FOR NETHERLANDS INCLUDING 8% HOLLIDAY
ALLOWANCE FOR RESPONDENTS WHO INDICATED RECEIVING THIS.
```

```
USE ALL.
COMPUTE filter_$=(COUNTRY=528).
VARIABLE LABEL filter_$ 'COUNTRY=528 (FILTER)'.
VALUE LABELS filter_$ 0 'Not Selected' 1 'Selected'.
FORMAT filter_$ (f1.0).
FILTER BY filter_$.
EXECUTE .
```

```
comp WAGEGRHH=WAGEGRHR.
var lab WAGEGRHH 'Hourly gross wage in Netherlands including 8% holiday allowance if
indicated'.
if (FRANHOLI=1 or wahollid>0) WAGEGRHH=WAGEGRHR*1.08.
if (frhollid=1) WAGEGRHH=WAGEGRHR*1.08.
EXECUTE .
```

```
MEANS TABLES=WAGEGRHH by surveyy.
```

```
FILTER OFF.
USE ALL.
EXECUTE .
```

6. ANNUAL INCREASE OF WAGES

The cumulative dataset will be used for the *Salary Check*, meaning wages of different years are in the same dataset. Therefore it is necessary to calculate the annual increase of the wages for the different years for every country. Table 6 gives an example of how this has been done for the Netherlands and for Germany for the *Salary Check* of 2005. These annual increase must be delivered by the countries to Kea Tijdens, in order to calculate the different wages to one level.

Table 6. Calculation of the annual increases for the Netherlands and Germany

```
***** NETHERLANDS ANNUAL INCREASE 2004 - 2005 0.5%.

comp sysmis=0.
recode sysmis(1=1)(else=sysmis).
execute.

comp WAGE05NL=sysmis.
var lab WAGE05NL 'Hourly gross wage NL at the level of 2005 including 8% holiday allowance'.
if (surveyy = 2000 and country=528) WAGE05NL= WAGEGRHH* 1.124 .
if (surveyy = 2001 and country=528) WAGE05NL= WAGEGRHH* 1.082851638 .
if (surveyy = 2002 and country=528) WAGE05NL= WAGEGRHH* 1.047530289 .
if (surveyy = 2003 and country=528) WAGE05NL= WAGEGRHH* 1.0208901 .
if (surveyy = 2004 and country=528) WAGE05NL= WAGEGRHH* 1.008071749 .
if (surveyy = 2005 and country=528) WAGE05NL= WAGEGRHH* 1.
execute.

MEANS TABLES=WAGE05NL by surveyy.

if (country=528) WAGEL5NL = LN(WAGE05NL) .
var lab WAGEL5NL 'Log hourly gross wage NL at the level of 2005 including 8% holiday allowance'.
execute.

des WAGE05NL WAGEGHHL.

***** GERMANY ANNUAL INCREASE 2004 - 2005 0.5%.

comp WAGE05DE=sysmis.
var lab WAGE05DE 'Hourly gross wage DE at the level of 2005'.
if (surveyy = 2004 and country=276) WAGE05DE= WAGEGRHH* 1.005.
if (surveyy = 2005 and country=276) WAGE05DE= WAGEGRHH* 1.
execute.

if (country=276) WAGEL5DE = LN(WAGE05DE) .
var lab WAGEL5DE 'Log hourly gross wage DE at the level of 2005'.
execute.
```

7. PREPARE THE VARIABLES FOR THE REGRESSION

The Dutch variables have been used as an example in the next syntax.

***** EDUCATION.

COMP EDUCAsc=EDUCAT.

var lab EDUCAsc 'ISCED education in 6 categories'.

FORMAT EDUCAsc (F2).

VAL LAB EDUCAsc

1 'Basic education'

2 'Lower sec or 2nd stage of basic education'

3 'Upper secondary education'

4 'Post-secondary non-tertiary education'

5 'First stage of tertiary education'

6 'Second stage of tertiary education'.

RECODE EDUCAsc (0 thru 1=1)(2=2)(3=3)(4=4)(5=5)(6=6).

fre EDUCAsc.

***** EXPERIENCE.

compute tenuexpe= SURVEYY -yyfstjob.

var lab tenuexpe 'CALC years of exp with all empl'.

format tenuexpe (f4).

if (BREAK0DE GE 1 and yyrenter ge 1900 and yybreak ge 1900) tenuexpe= SURVEYY -yyfstjob - (yyrenter - yybreak).

if (BREAK01 GE 1 and yyrenter ge 1900 and yybreak ge 1900) tenuexpe= SURVEYY -yyfstjob - (yyrenter - yybreak).

if (BREAK0 GE 1 and yyrenter ge 1900 and yybreak ge 1900) tenuexpe= SURVEYY -yyfstjob - (yyrenter - yybreak).

IF (tenuexpe LT 0) tenuexpe=prut.

fre tenuexpe.

compute tenuexsq=tenuexpe*tenuexpe.

var lab tenuexsq 'CALC years of exp squared'.

***** SUPERVISORY JOB.

FRE supv1 .

COMPUTE supvdich=0.

VAR LAB supvdich 'SUPERVISORY POSITION'.

VAL LAB supvdich 0 'NO' 1 'YES'.

IF (supv1 GE 1) supvdich=1.

IF (SYSMIS(supv1)) supvdich=PRUT.

FRE supvdich.

***** MALE COWORKERS.

FRE depmale.

compute depmale2=depmale.

var lab depmale2 'CALC most colleagues in similar positions are men 8=0 for Salarycheck'.

format depmale2 (f2).

recode depmale2 (1=1)(else=0).

val lab depmale2 jobprom3 1 'yes' 0 'no incl na'.

fre depmale depmale2.

***** RENTRANT.

COMPUTE reentrant=PRUT.

VAR LAB reentrant 're-entrant woman'.

```
val lab reentrant 0 'no' 1 'yes'.
form reentrant (f2).
if (sex eq 2) reentrant =0.
if (sex eq 1) reentrant =0.
if (sex eq 1 and BREAK0DE ge 1) reentrant =1.
if (sex eq 1 and BREAK01 ge 1) reentrant =1.
if (sex eq 1 and BREAK0 ge 1) reentrant =1.
fre reentrant.

*** JOBPROMOTION.
fre jobpromo.
compute jobprosc=jobpromo.
var lab jobprosc 'CALC have been promoted 8=0 for Salarycheck'.
form jobprosc (f2).
if (sysmis(jobpromo)) jobprosc=prut.
recode jobprosc (1=1)(else=0).
fre jobprosc.

*** FIRMSIZE.
fre firmsize.
compute firmsiz3=firmsize.
recode firmsiz3 (1 thru 2=1)(3=2)(4 thru hi=3)(else=sysmis).
compute firmsiz4=firmsiz3.
compute firmsiz5=firmsiz3.
var lab firmsiz3 'CALC firm size 3 groups <100 .. > 500 for Salarycheck'.
var lab firmsiz4 'CALC firm < 100 empl for Salarycheck'.
var lab firmsiz5 'CALC firm > 500 empl for Salarycheck'.
val lab firmsiz3 1 '< 100' 2 '100-500' 3 '>500'.
val lab firmsiz4 firmsiz5 1 'yes' 0 'no'.
form firmsiz3 firmsiz4 firmsiz5 (f2).
recode firmsiz4 (1=1)(else=0).
recode firmsiz5 (3=1)(else=0).
fre firmsize firmsiz3 firmsiz4 firmsiz5 .
```


8. TEST VARIABLES PER OCCUPATION

To check whether all categories within the variables used in the *Salary Check* have more than 10 observations, it is needed to run a frequency table per occupation.

**** SALARY CHECK COEFFICIENTS FOR OCCUPATION 'Bürokauffrau/-mann'.

USE ALL.

COMPUTE filter_\$=(occupade eq 'Bürokauffrau/-mann').

FILTER BY filter_\$.

EXECUTE .

fre EDUCAsc supvdich depmale2 reentrant jobprosc firmsiz4 firmsiz5.

Test missings:

		EDUCAsc ISCED education in 6 categories	supvdich SUPERVIS ORY POSITION	depmale2 CALC most colleagues in similar positions are men 8=0 for Salaryche ck	reentrant re-entrant woman	jobprosc CALC have been promoted 8=0 for Salaryche ck	firmsiz4 CALC firm < 100 empl for Salaryche ck	firmsiz5 CALC firm > 500 empl for Salaryche ck
N	Valid Missin g	423 2	420 5	425 0	424 1	425 0	425 0	425 0

Test the number of observations in every variable within this occupation. If a cell has less than 10 observations, recode the variable into less categories. If this is not possible, for example the variable supervisory position has only two categories, or because after recoding the variable has still cells with less than 10 observations, the variable is deleted from the regression model.

TEST NR OF OBSERVATIONS IN EDUCAsc ISCED education in 6 categories

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1 Basic education	49	11.5	11.6	11.6
	2 Lower sec or 2nd stage of basic education	213	50.1	50.4	61.9
	3 Upper secondary education	74	17.4	17.5	79.4
	4 Post-secondary non-tertiary education	70	16.5	16.5	96.0
	5 First stage of tertiary education	17	4.0	4.0	100.0
	Total	423	99.5	100.0	
Missing	System	2	.5		
Total		425	100.0		

supvdich SUPERVISORY POSITION

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0 no	251	59.1	59.8	59.8
	1 yes	169	39.8	40.2	100.0
	Total	420	98.8	100.0	
Missing	System	5	1.2		
Total		425	100.0		

Depmale2 CALC most colleagues in similar positions are men 8=0 for Salarycheck

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0 no incl na	197	46.4	46.4	46.4
	1 yes	228	53.6	53.6	100.0
	Total	425	100.0	100.0	

reentrant re-entrant woman

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0 no	369	86.8	87.0	87.0
	1 yes	55	12.9	13.0	100.0
	Total	424	99.8	100.0	
Missing	System	1	.2		
Total		425	100.0		

jobprosc CALC have been promoted 8=0 for Salarycheck

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0	349	82.1	82.1	82.1
	1	76	17.9	17.9	100.0
	Total	425	100.0	100.0	

firmsiz4 CALC firm < 100 empl for Salarycheck

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0 no	341	80.2	80.2	80.2
	1 yes	84	19.8	19.8	100.0
	Total	425	100.0	100.0	

firmsiz5 CALC firm > 500 empl for Salarycheck

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	0 no	162	38.1	38.1	38.1
	1 yes	263	61.9	61.9	100.0
	Total	425	100.0	100.0	

9. REGRESSION PER OCCUPATION

The coefficients and the constant in the calculation rules are derived from regression analyses. A multiple regression analysis shows the effect of the particular independent variable on a wage, taking into account the effects of the remaining independent variables in the analysis. For these analyses, the Dutch WageIndicator data have been used. The *Salary Check* dataset is used to run wage regressions per occupation, predicting the logarithm of the hourly wages.

REGRESSION

```
/MISSING LISTWISE
/STATISTICS COEFF OUTS R ANOVA
/CRITERIA=PIN(.05) POUT(.10)
/NOORIGIN
/DEPENDENT WAGEGRHL
/METHOD=ENTER EDUCAsc TENUEXPE TENUXSQ SUPVDICH DEPMAL2 RENTRANT
JOBPROSC FIRMSIZ4 FIRMSIZ5 .
```

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.588(a)	.346	.346	.36551

a Predictors: (Constant), FIRMSIZ5 firm > 500 empl for Salarycheck, SUPVDICH Supervisory position Y/N, RENTRAN2 Re-entrant male=0, EDUCAsc ISCED education in 6 categories, DEPMAL2 Most colleagues in similar positions are men -9=0 for Salarycheck, JOBPROSC Has been promoted -9=0 for Salarycheck, TENUXSQ Tenure squared, FIRMSIZ4 firm < 100 empl for Salarycheck, TENUEXPE Tenure (survey year-year first job)

ANOVA(b)

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	8928.884	9	992.098	7426.070	.000(a)
	Residual	16891.965	126440	.134		
	Total	25820.849	126449			

a Predictors: (Constant), FIRMSIZ5 firm > 500 empl for Salarycheck, SUPVDICH Supervisory position Y/N, RENTRAN2 Re-entrant male=0, EDUCAsc ISCED education in 6 categories, DEPMAL2 Most colleagues in similar positions are men -9=0 for Salarycheck, JOBPROSC Has been promoted -9=0 for Salarycheck, TENUXSQ Tenure squared, FIRMSIZ4 firm < 100 empl for Salarycheck, TENUEXPE Tenure (survey year-year first job)

b Dependent Variable: WAGEGRHL

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
(Constant)	1.797	.005		387.342	.000
EDUCAsc ISCED education in 6 categories	.072	.000	.372	157.501	.000
TENUEXPE Tenure (survey year-year first job)	.041	.000	.950	119.995	.000
TENUXSQ Tenure squared	-.001	.000	-.602	-77.199	.000
SUPVDICH Supervisory position Y/N	.088	.002	.095	38.930	.000
DEPMALE Most colleagues in similar positions are men -9=0 for Salarycheck	.063	.002	.070	29.685	.000
RENTANT Re-entrant male=0	-.121	.004	-.078	-32.054	.000
JOBPROSC Has been promoted -9=0 for Salarycheck	.069	.002	.074	30.709	.000
FIRMSIZ4 firm < 100 empl for Salarycheck	-.093	.003	-.100	-36.656	.000
FIRMSIZ5 firm > 500 empl for Salarycheck	.038	.003	.030	11.097	.000

a Dependent Variable: WAGEGRHL

Sometimes the coefficients of EDUCAsc, tenuexpe, supvdich and jobprosc have negative values and the coefficients of reentrant and depmale2 have positive values. The regression has to be run again without these variables, or it can be decided not to include the occupation at all in the *Salary Check*. The latter is particularly the case when the coefficient is significant. The main reason for not including these counter-predictable variables is that they cause a load of emails, because they are so much against general expectations. A web-visitor can not understand that for example higher education or supervisory position is associated with fewer instead of higher wages. There are however occupations where this is the case. For the truck driver, having a supervisory position negatively influences the wage. Promotion may have a negative effect too, for example when promotion is a mean to enter an occupation, for example becoming a secretary from the position of clerical worker. In that case, the respondent is promoted, but starts at the lowest level within the occupation.

The Rsquare in the Model summary of the regression must be at least .30. In case it is lower, it is preferably not to include the occupation in the *Salary Check*, because

obviously the wages for this occupation are determined by other variables than the ones included in the regression.

The unstandardized coefficients can then be inserted in an excel sheet, using PASTE from SPSS. The occupation code, occupation label and the unweighted N of the occupation must also be inserted (see Table 7).

Table 7. *Example of the excel sheet with the coefficients*

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	code	occupation	constant	EDUCAsc	tenuexpe	tenuexsq	supvdich	depmale2	rentrant	jobprosc	firmsiz4	firmsiz5	N
2	2640400	constructiebankwerker	2.8000	0.0000	0.0226	-0.0005	0.0000	-0.4500	0.0292	0.2312	-0.1283	-0.3287	83
3	2640900	metaalbewerker	2.1078	0.0448	0.0667	-0.0014	0.2009	-0.2704	0.0000	0.0000	0.0528	0.1421	78
4	2640901	plaatwerker carosseriebouw	1.3535	0.0446	0.0654	-0.0012	0.0430	0.2297	-0.0152	0.0000	0.1581	0.3863	107
5	2641400	monteur, samensteller meta	1.8472	0.0000	0.0449	-0.0007	0.0261	0.0692	-0.1227	0.1058	0.0205	0.1104	144
6	2641500	keurder, keurmeester metaa	2.5054	0.0184	0.0000	0.0000	0.2689	0.0101	0.0000	0.1637	-0.0325	-0.2308	66
7	2650500	maker-hersteller autobanden	2.6288	0.0038	0.0000	0.0000	0.2346	-0.1969	0.0000	0.1589	-0.3434	0.0000	51

10. PROFILES

A profile is made per occupation consisting of the variables used in the wage regression. For these profiles (Table 8), the Dutch WageIndicator data and the Dutch variables have been used.

Table 8. Example of a profile on the output screen



For generating a table with the profiles, it is necessary to work with the occupations without the labels.

Use all.

Compute ISCO9NU1 = ISCO9NUM.

Var lab ISCO9NU1 'ISCO9NUM for profiles'.

Execute.

***EXPERIENCE: recode into groups

compute tenuexp5 = tenuexpe.

recode tenuexp5 (0 thru 4 = 1) (5 thru 9 = 2) (10 thru 19 = 3) (20 thru 29 = 4) (30 thru hi = 5).

var lab tenuexp5 'work experience in 5 groups'.

val lab tenuexp5 1 '< 5 years' 2 '5 – 9 years' 3 '10 – 19 years' 4 '20 – 29 years' 5 '30 + years'.

fre tenuexp5.

***FIRMSIZE

Use the variable firmsiz3 for the profiles to get the percentage of the three firm size groups in the profile table (less than 100, 100 – 500, more than 500).

*** Select the occupations that have been used in the *Salary Check*

USE ALL.

COMPUTE filter_p =

(ISCO9NU1 = 131000000 | ISCO9NU1 = 213101000 | ISCO9NU1 = 213104000 | ISCO9NU1 = 213200010 | ISCO9NU1 = 213900007 | ISCO9NU1 = 214400000 | ISCO9NU1 = 241100000 | ISCO9NU1 = 244600000 | ISCO9NU1 = 247000000 | ISCO9NU1 = 312100001) .

VARIABLE LABEL filter_p '(occupations in salary checker) (FILTER)'.

VALUE LABELS filter_p 0 'Not selected' 1 'Selected'.

FORMAT filter_p (f1.0).

FILTER BY filter_p.
EXECUTE .

Create a filter for every variable used in the Salary Checker so no empty cell will taken into account.

*** filter for the variable EXPERIENCE

```
compute filter_k = ((gender >= 0) and (tenuexp5 >= 0) and (supvdich >= 0) and (ostwest >= 0) and (firmsiz3 >= 0)).
var lab filter_k 'cases without empty cells.'
val lab filter_k 0 'not selected' 1 'selected'.
format filter_k (f1.0).
filter by filter_k.
execute.
```

***combine the filter for the occupations and the variables

```
compute filter_l = ((filter_k = 1) and (filter_p = 1)).
var lab filter_l 'filter for profiles'.
val lab filter_l 0 'not selected' 1 'selected'.
format filter_l (f1.0).
filter by filter_l.
execute.
```

***profile table

```
TABLES
/FORMAT BLANK MISSING(' ')
/GBASE=CASES
/TABLE=isconlB1 BY supvdiBE > (STATISTICS) + educatx > (STATISTICS) + firmsBE3 >
(STATISTICS) +
firmfeBE > (STATISTICS) + firmowBE > (STATISTICS) + tenuex5 > (STATISTICS)
/STATISTICS
cpct (isconlB1( PCT5.1 ) 'Row %':isconlB1 ).
```

Insert this table in a second worksheet in the excel sheet, using PASTE from SPSS (see Table 9). The labels in the first row of this excel sheet are literally used in the output screen. The code corresponding to the occupation label must also be inserted.

Table 9. Example of the excel sheet with the profiles

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	code	opleiding: Basisschool	opleiding: VBO/Leerling wezen	opleiding: VMBO/M AVO/MU LO	opleiding: HAVO/V WO/MBO	opleiding: HBO	opleiding: Universiteit	werknemer: minder dan 100	werknemer: 100 - 500	werknemer: meer dan 500	ervaring: < 5 jaar	ervaring: 5 - 9 jaar	ervaring: 10 - 19 jaar	ervaring: 20 - 29 jaar	ervaring: 30+ jaar
2	2640400	2.34349133	47.32610158	27.15285	22.55502	0.62253	0	85.404951	11.018135	3.5769137	9.2212	19.9431	22.9101	23.3848	24.5408
3	2640900	10.0726064	36.33641914	11.94028	35.35619	6.2945	0	84.069991	10.876283	5.0537261	13.096	15.0928	36.8893	20.3368	14.5849
4	2640901	6.74525442	25.08254455	27.94818	34.09272	4.74543	1.3858668	83.519447	6.3403273	10.140225	6.4442	31.6753	21.9043	16.7973	23.1789
5	2641400	2.6633532	46.30857971	14.26376	36.23703	0.52728	0	73.857807	22.184316	3.9578766	8.014	12.2075	20.8729	25.9113	32.9944
6	2641500	5.60614635	15.4794704	23.10362	32.91517	20.4629	2.5326965	46.494947	43.222428	10.282625	5.9025	7.51936	31.2502	20.0894	35.2385
7	2650500	13.192975	16.28805742	47.77125	22.74772	0	0	90.787483	9.2125175	0	15.001	20.4533	23.2666	18.2038	23.075

The codes must be exactly the same and in exactly the same order as they were in the coefficient excel sheet.

11. FINALLY

A third excel sheet must be inserted with the variables, variable labels and the values used (see Table 10).

Table 10. Example of the third excel sheet with the information of the variables

	A	B	C	D
1	Variable	Label used in Salary Check	Category used in Salary Check	value
2	EDUCAsc	Select your education level		
3	EDUCAsc		Basic education	1
4	EDUCAsc		Lower sec or 2nd stage of basic education	2
5	EDUCAsc		Upper secondary education	3
6	EDUCAsc		Post-secondary non-tertiary education	4
7	EDUCAsc		First stage of tertiary education	5
8	EDUCAsc		Second stage of tertiary education	6
9	supvdich	Do you have a supervisory position?		
10	supvdich		Yes	1
11	supvdich		No	0

When all three excel sheets are made, the *Salary Check* can be uploaded into the Questionnaire Management System (QMS), and will be ready for testing within two weeks.
