**MSDS 6370-404 TERM PROJECT:**

**SRS, Stratified Proportional and Neyman Sampling of 1994 US Census Data**

**Submitted by:** James Hosker & Vinh Le

**Due Date:** Nov 22, 2017

**Course #:** MSDS 6370-404 (Wed, 8:30PM Class)

1. **Introduction**

In this report, we provide a summary of our predictive analysis of a mean variable from 1994 Adult US Census data set, using a simple random sample (SRS), stratified proportional sample (STRprop) and stratified Neyman (STRNeyman) sample. We will compare and determine which of these three methods is better in estimating the mean using the standard error of the mean. We will then randomly generate five samples for two of the three design samples and determine if the actual mean and median from the entire population is within a 95% confidence interval of the estimated mean from our two sample designs.

**1.1 Data Description and Source**

We used the **Census Income Data Set** from the UCI Machine Learning Repository (see links <https://archive.ics.uci.edu/ml/datasets/Census+Income>. The dataset was created by Barry Becker from the 1994 Adult US Census database. This set of reasonably clean records was extracted from the 1994 Census Data using the following conditions: ((Adult Age> 16) && (Adjusted Gross Income > 100) && (AFNL Weight > 1) && (Hours Worked per Week > 0)).

The training dataset has 32,561 rows of responses from individuals with 15 attributes (14 explanatory and 1 response variable). The explanatory variables can be broken down into 6 continuous (quantitative) variables and 9 categorical variables. The quantitative variables are age, final weight (US Census Factor), education number, capital gain, capital loss, and hours worked/week. The categorical variables are working class, education, marital status, occupation, relationship, race, sex, native country, and income. Income is the response variable where income is classified as categorically as >$50K or <=$50K.

For our analysis we will focus on the relationship between two explanatory variables:

* 1. Education Number: numerical variable for number of years of education
  2. Occupation: categorical variable for job occupation such as Executive Managerial, Sales, Craft-repair, etc.

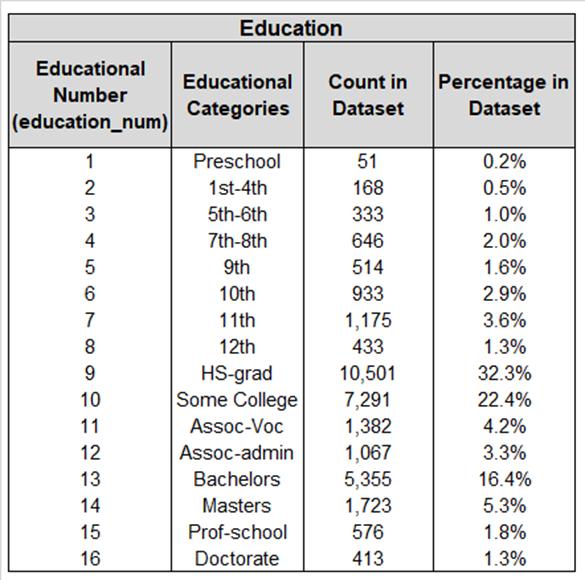
***Before we begin we highlight that the SAS code for tables in Introduction, Task 01 and Task 02 can be found in*** **Appendix I**.

**1.2 Data Cleaning**

We cleaned the data using Python in Jupyter (see **Appendix II** for details on the data cleaning and our Python Code). First, we adjusted for missing data. Occupations that were blank but had working class of *‘None’* were set to *‘None’* because if you did not work you did not have an occupation. In addition, occupations that were blank were assigned to category ‘Undisclosed’. There was no missing data for the number of years of education; however, the number of years of education was initially read in as a character that was converted to an integer. Second, we deleted some data. We deleted any census response where most (10 or more) of the explanatory variables were blank (about 100 samples from the population set). Third, we had redundant data because education category and number of years of education represented the same information; therefore, we created a table in the next section to equate them as choose to use the number of years of education. We created a final dataset (called occed) that contained only the variables education\_num (for educational number of years) and occnum (for occupational number).

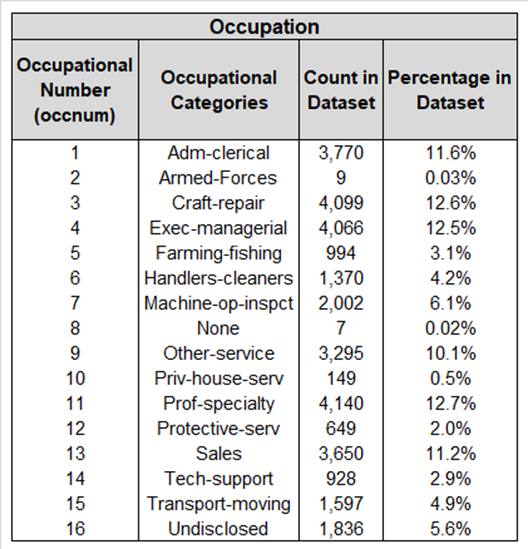
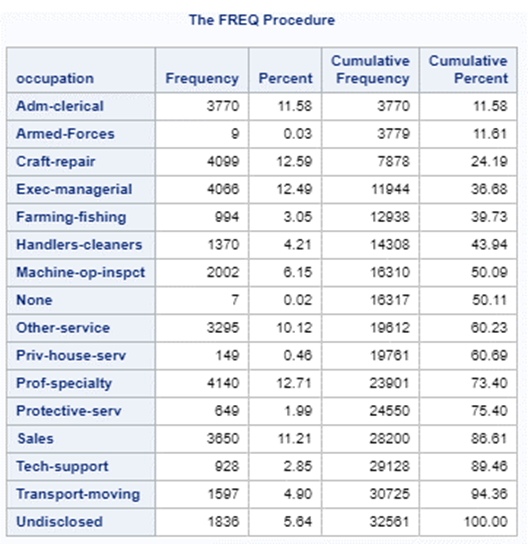
**1.3 Data Exploration (Education Number & Occupation)**

First for our data exploration, **Table 1** equates the education number to educational category. We used the PROC FREQ in SAS and then translated that into percentages for each education number of years. From our population sample, the larger categories of educational numbers are 9 for High School Graduate, 10 for Some College; and 13 for Bachelors Degree. We note that education number 8 is equivalent to 12th grade so there are 4 years assumed.



**Table1: PROC FREQ SAS Output and Education Number (Yrs.) vs. Educational Category**

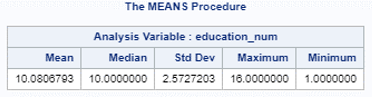
Second, **Table 2** equates the occupational number to occupation. We transform the category of occupation that will be used as our stratum to a number to make is easier to analyze using our SAS code. We used the PROC FREQ in SAS and then translated that into percentages for each occupational category. From our population sample, the larger categories of occupational numbers are Adminstrative-Clerical, Craft-Repair, Executive Managerial, Other-Services; Professional-Speciality, and Sales.



**Table2: PROC FREQ SAS Output and Occupation Category vs. Occupation Number**

**1.4 The Appropriate Sample Size**

In this section, we calculate the appropriate sample size for SRS, STRProportional and STRNeyman. **Table 3** shows the output of PROC MEAN in SAS, which shows an actual mean of 10.0807, median of 10 and standard deviation of 2.5727. **Equation 1** shows our calculation and inputs for the initial number of random samples. We assume a 95% confidence interval for our number of samples.



**Table 3: PROC MEAN Output of Total Population (Training Data Set)**

* where S=standard deviation=2.5727; α=0.05 (I95% or moe=margin of error); =1.96 from lookup table (see <https://people.wku.edu/david.neal/109/Unit4/ConfInt.pdf>)

In this case, we apply the finite population correction (fpc) in equation 2 because > 10% of the population (~31%).

= ~= 7,750 (Eq. 2)

* where N = Total Population = 32,561 from out training data set and = 10,171.

For our analysis of the three model we will use the sample size . We note here that this is still are large sample set; and therefore, in comparing each sample design, we may see smaller amounts of differentiation. On way we could reduce the sample set is to increase the margin of error (moe) from 0.05. We tried this and re-ran the analysis to follow and found to see more differentiation we would have to increase the moe to 0.10 or higher. Therefore, we kept the moe at 0.05.

1. **Task 01 – Analysis & Comparison of Three Sampling Techniques**

Here, we examine the output from our three sampling techniques or designs. First, we show how we produced the output form our simple random sample (SRS). We note again that the SAS code can be found in **Appendix I**. We used PROC SURVEYSELECT in SAS with a sample size of 7750 and set the seed to 9118 rather than a randomly selected number. This way both my partner and I could reproduce the same results. SRS assumes and equal probability of selection for the 7,750 sample from the total population of 32,561.

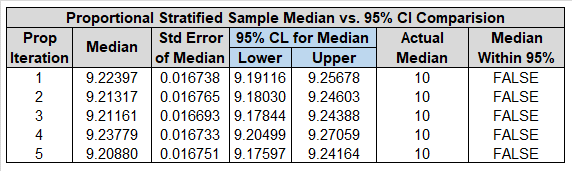
The standard deviation is a representation of the spread of each of the data points. The standard deviation is used to help determine validity of the data based the number of data points displayed within each level of standard deviation. Standard error functions more as a way to determine the accuracy of the sample or the accuracy of multiple samples by analyzing deviation within the means.

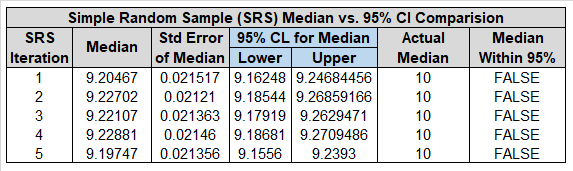
Read more: Standard Error https://www.investopedia.com/terms/s/standard-error.asp#ixzz4yMMaRrRa

Follow us: Investopedia on Facebook

1. **Task 02 – Analysis & Comparison of Three Sampling Techniques**

Here we examine 5 different independent samples from a stratified method using proportional allocation and 5 different independent simple random samples for a total of 10 samples. To do that, we change the seeding in PROCSURVEYSELECT to create the 10 different samples for our comparison. The SAS code is provided in the appendix. We then calculate the 95% confidence interval from each independent sample and compare that to the population mean and median. The result shows that the actual mean falls within the 95% confidence interval for Proportional and SRS allocation 100% of the time. The actual median however does not fall within the 95% confidence interval for neither Proportional nor Simple Random Samples from the 10 samples. Occupations stratums can account for some of this and a close split between education numbers 9 and 10. Bootstrapping could be an alternative sampling method to get closer to the actual median.





**APPENDIX I: SAS Code for Our Analysis**

**TASK 1**:

**TASK 2**:

data AdultTrain;

infile '/folders/myfolders/AdultTrain.csv' dlm=',' firstobs=2 ;

input item age workclass$ final\_wgt education$ education\_num marital\_status$ occupation$ relationship$ race$ sex$ captital\_gain capital\_loss hours\_per\_week native\_cntry$ income$ ;

if occupation = 'Adm-cler' then occnum = 1;

if occupation = 'Armed-Fo' then occnum = 2;

if occupation = 'Craft-re' then occnum = 3;

if occupation = 'Exec-man' then occnum = 4;

if occupation = 'Farming-' then occnum = 5;

if occupation = 'Handlers' then occnum = 6;

if occupation = 'Machine-' then occnum = 7;

if occupation = 'None' then occnum = 8;

if occupation = 'Other-se' then occnum = 9;

if occupation = 'Priv-hou' then occnum = 10;

if occupation = 'Prof-spe' then occnum = 11;

if occupation = 'Protecti' then occnum = 12;

if occupation = 'Sales' then occnum = 13;

if occupation = 'Tech-sup' then occnum = 14;

if occupation = 'Transpor' then occnum = 15;

if occupation = 'Undisclo' then occnum = 16;

run;

/\*\*\*\* SRS \*\*\*\*/

proc surveyselect data=AdultTrain method = srs

sampsize=7750 seed=91113 out = srssample1;

title "SRS allocation";

run;

proc surveyselect data=AdultTrain method = srs

sampsize=7750 seed=91114 out = srssample2;

title "SRS allocation";

run;

proc surveyselect data=AdultTrain method = srs

sampsize=7750 seed=91115 out = srssample3;

title "SRS allocation";

run;

proc surveyselect data=AdultTrain method = srs

sampsize=7750 seed=91116 out = srssample4;

title "SRS allocation";

run;

proc surveyselect data=AdultTrain method = srs

sampsize=7750 seed=91117 out = srssample5;

title "SRS allocation";

run;

proc surveymeans data = srssample1 mean median CLM sum CLSUM;

title "SRS Stats";

var education\_num;

/\*weight SamplingWeight;\*/

title "SRS allocation";

run;

proc surveymeans data = srssample2 mean median CLM sum CLSUM;

title "SRS Stats";

var education\_num;

/\*weight SamplingWeight;\*/

title "SRS allocation";

run;

proc surveymeans data = srssample3 mean median CLM sum CLSUM;

title "SRS Stats";

var education\_num;

/\*weight SamplingWeight;\*/

title "SRS allocation";

run;

proc surveymeans data = srssample4 mean median CLM sum CLSUM;

title "SRS Stats";

var education\_num;

/\*weight SamplingWeight;\*/

title "SRS allocation";

run;

proc surveymeans data = srssample5 mean median CLM sum CLSUM;

title "SRS Stats";

var education\_num;

/\*weight SamplingWeight;\*/

title "SRS allocation";

run;

/\*\*\* Proportional Stratification \*\*\*/

proc sort data=AdultTrain;

by occnum;

run;

proc surveyselect data=AdultTrain out=prop\_samp sampsize=7750;

strata occnum / alloc=prop nosample;

run;

proc print data=prop\_samp;

run;

proc sort data=AdultTrain;

by occnum;

run;

proc surveyselect data=AdultTrain method = srs out = propsample1

sampsize = (897,2,976,968,237,326,477,2,784,35,985,154,869,221,380,437) seed=91113;

strata occnum;

title "Proportional Allocation Sampling";

run;

proc surveyselect data=AdultTrain method = srs out = propsample2

sampsize = (897,2,976,968,237,326,477,2,784,35,985,154,869,221,380,437) seed=91114;

strata occnum;

title "Proportional Allocation Sampling";

run;

proc surveyselect data=AdultTrain method = srs out = propsample3

sampsize = (897,2,976,968,237,326,477,2,784,35,985,154,869,221,380,437) seed=91115;

strata occnum;

title "Proportional Allocation Sampling";

run;

proc surveyselect data=AdultTrain method = srs out = propsample4

sampsize = (897,2,976,968,237,326,477,2,784,35,985,154,869,221,380,437) seed=91116;

strata occnum;

title "Proportional Allocation Sampling";

run;

proc surveyselect data=AdultTrain method = srs out = propsample5

sampsize = (897,2,976,968,237,326,477,2,784,35,985,154,869,221,380,437) seed=91117;

strata occnum;

title "Proportional Allocation Sampling";

run;

data strsizes;

input occnum \_total\_;

datalines;

1 3770

2 9

3 4099

4 4066

5 994

6 1370

7 2002

8 7

9 3295

10 149

11 4140

12 649

13 3650

14 928

15 1597

16 1836

;

run;

/\*\*\*\*\*\*5 Stratified Sampling\*\*\*\*\*/

proc surveymeans data = propsample1 total = strsizes

mean median CLM sum CLSUM;

var education\_num;

weight SamplingWeight;

strata occnum;

title "Proportional allocation";

run;

proc surveymeans data = propsample2 total = strsizes

mean median CLM sum CLSUM;

var education\_num;

weight SamplingWeight;

strata occnum;

title "Proportional allocation";

run;

proc surveymeans data = propsample3 total = strsizes

mean median CLM sum CLSUM;

var education\_num;

weight SamplingWeight;

strata occnum;

title "Proportional allocation";

run;

proc surveymeans data = propsample4 total = strsizes

mean median CLM sum CLSUM;

var education\_num;

weight SamplingWeight;

strata occnum;

title "Proportional allocation";

run;

proc surveymeans data = propsample5 total = strsizes

mean median CLM sum CLSUM;

var education\_num;

weight SamplingWeight;

strata occnum;

title "Proportional allocation";

run;

**APPENDIX II: Python Code in Jupyter for Data Cleaning**