# Using Google Consumer Surveys

*Lars Vilhuber*

## Contents

This document reports a simple experiment using Google Consumer Surveys.

```
# define basic parameters
price_quoted <- 3
price_minimum <- 0.1
budget <- 500
coupon <- 75
result_file <- c("Employment and search.xls")
```

## Survey instrument

We based our survey instrument on the primary labor force question on the CPS, somewhat modified:

Please complete the following survey to access this premium content.

Question 1 of 2 or fewer:

**Last week, did you do ANY work for pay?**

○ No, I am retired

○ No, I am disabled

○ No, I was unable to work

○ Yes

○ No

OR

↻ Show me a different question

↱ Skip survey

Google    INFO    PRIVACY

We followed up for all non-employed (all those not responding with "Yes"") with a question on search intensity that was not based on the CPS:

## Results

Google first ran a test (which resulted in being 4 responses) to assess how much to charge us. Google claims that prices are between $0.1 and $3. In our case, the quoted price was $3 per response, and our budget was $500 (although we were offered a one-time coupon valued at $75. We thus requested 191 cases. Once accepted, Google posted the survey, targetting the "general population in the United States on the Google Consumer Survey publisher network".

```
# get overview data
xls_overview <- read.xls(result_file,sheet=1)
xls_topline <- read.xls(result_file,sheet=2)
xls_data <- read.xls(result_file,sheet=3)
```

To our initial surprise, we had received 461 cases. As it turns out, since our labor force question served as a "screener", we were only charged for complete questions, i.e., responses that had completed both Q1 and Q2 of which we had obtained 192. Thus, the final response was

| Question.text | Response.c |
| --- | --- |
| Last week, did you do ANY work for pay? | |
| If you are not working, how many jobs did you apply for (by email, website, letter, in person, etc.) last week? | |

From this, we can compute a few statistics of interest.

```r
results <- data.frame(c(0,0,0),row.names = c("Employment-Population Rate","Unemployment rate","OLF rate
results[,1:2] <- 0
names(results) <- c("Google","BLS")
# define OLF, pop, unemployed, employed
pop <- xls_overview[1,c("Response.count")]
nonemp <- xls_overview[2,c("Response.count")]
employed <- pop - nonemp
olfind <- xls_data$Question..2.Answer=="0 (I am not looking for work)"
olf <- nrow(xls_data[olfind,])
google.dates <- c(
  min(as.POSIXct(strptime(xls_data[,"Time..UTC."],"%Y-%m-%d %H:%M:%S"),tz="UTC")),
  max(as.POSIXct(strptime(xls_data[,"Time..UTC."],"%Y-%m-%d %H:%M:%S"),tz="UTC"))
  )
# now compute stats
results["Employment-Population Rate",1] <-
  round(employed*100/pop,2)
results["Unemployment rate",1] <-
  round((nonemp-olf)*100/employed,2)
results["OLF rate",1] <-
  round(olf*100/pop,2)
```

which we measured between 2016-01-08 11:10:46 and 2016-01-08 20:20:05.

For comparison purposes, we obtained the BLS unemployment rate and employment-population ratio from FRED:

```r
# define the series of interest
fred_unemployment <- c("UNRATENSA")
fred_unemployment20 <- c("LNU04000024")
fred_emppop_all   <- c("LNU02300000")
fred_emppop_men   <- c("LNU02300001")
fred_emppop_women <- c("LNU02300002")
fred_olf_count    <- c("LNU05000000")
fred_pop_count    <- c("CNP16OV")

#bls.urate <- get(getSymbols(fred_unemployment,src='FRED'))
bls.urate20 <- get(getSymbols(fred_unemployment20,src='FRED'))
bls.emppop_all <- get(getSymbols(fred_emppop_all,src='FRED'))
#bls.emppop_men <- get(getSymbols(fred_emppop_men,src='FRED'))
#bls.emppop_women <- get(getSymbols(fred_emppop_women,src='FRED'))
#
# Compute the OLF rate
cnt.olf <- get(getSymbols(fred_olf_count,src='FRED'))
cnt.pop <- get(getSymbols(fred_pop_count,src='FRED'))
bls.olf_rate <- cnt.olf*100/cnt.pop
#
# attach the closest value - this might change over time, but that's fine.
```

```
# found at http://stackoverflow.com/questions/8186960/finding-the-most-recent-observation-earlier-than-
# For emppop ratio, use all - we will break it out by gender later
results["Employment-Population Rate",2] <-
  round(bls.emppop_all[findInterval(google.dates[1],index(bls.emppop_all)),]
,2)
results["Unemployment rate",2] <-
  round(bls.urate20[findInterval(google.dates[1],index(bls.urate20)),],2)
results["OLF rate",2] <-
  round(bls.olf_rate[findInterval(google.dates[1],index(bls.olf_rate)),],2)
```

The final (unweighted) result is then as follows (note: the BLS numbers won't add up quite to 100, since they stem from different series):

|                              | Google | BLS   |
| ---------------------------- | ------ | ----- |
| Employment-Population Rate   | 58.35  | 59.40 |
| Unemployment rate            | 16.36  | 4.50  |
| OLF rate                     | 32.10  | 37.59 |

## Appendix

```
# libraries needed
library(quantmod) # read FRED
library(gdata) # read Excel
options("getSymbols.warning4.0"=FALSE)
Sys.info()
```

```
##                                                    sysname
##                                                    "Linux"
##                                                    release
##                                         "3.16.7-29-desktop"
##                                                    version
## "#1 SMP PREEMPT Fri Oct 23 00:46:04 UTC 2015 (6be6a97)"
##                                                    nodename
##                                                   "zotique2"
##                                                    machine
##                                                    "x86_64"
##                                                      login
##                                                  "vilhuber"
##                                                       user
##                                                  "vilhuber"
##                                              effective_user
##                                                  "vilhuber"
```