W6 Pulling it All Together

6.1 What about using statistical software to select samples?

Frame

e.g. Census Blocks

seqno	rent	own
1	196	118
2	327	0
3	331	1
4	266	81
5	334	23
6	354	34
7	395	3
8	397	7
9	145	278
10	383	91
11	584	75
12	1	0
13	0	1
14	1	0
15	0	1
16	1	0
17	1	0
18	0	1
19	0	1
20	1	0
21	0	1
22	0	1
23	0	1
24	0	1
25	1	0
26	1	0
27	0	1
28	0	1
29	0	1
30	0	1

$N = 975$

$n = 20$

$f = \frac{1}{48.675}$

- SRS (WOR)
- SRS WR
- Systematic
- PPS
- (Stratified)

Using R

• Change Directory

setwd("M:\\Coursera sampling methods")

• Open data file

frame <- read.table(file = "frame.txt", header=TRUE, sep = "\t")

• View data in a spreadsheet

edit(frame)

```
## library(sampling)
## simple random sample ##
  ## without replacement ##
sam.srswor <- srswor(n=20, N=975)
sample.srswor <- frame[which(x = (sam.srswor == 1)), ]
```

```
sam.srswor
 [1] 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
[38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
> sample.srswor
     seqno     rent    owner
7        7      395        3
21      21        0        1
93      93        5        0
185    185        0        9
222    222        0       11
228    228       10        1
246    246        0       12
301    301        0       14
306    306       14        0
347    347        0       17
467    467        6       16
472    472        8       14
495    495       12       12
506    506        6       18
538    538        0       26
705    705        6       33
778    778       51        0
804    804        1       57
809    809       10       49
879    879       64       20
```

Simple Random sampling with Replacement

```
sam.srswr <- srswr(n=20, N=975)
sample.srswr <- frame[which(x=(sam.srswr >= 1)), ]
sam.srswr
```

```
 [1] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 0 0 0 0 0 0
[38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
[75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
…
```

```
> sample.srswr
    seqno    rent   owner     dup
24     24       0       1       1
157   157       0       8       1
213   213       1      10       2
240   240       2      10       1
278   278       2      11       1
281   281       4      10       1
296   296       0      14       1
352   352       9       8       1
464   464       5      17       1
485   485       2      21       1
547   547       6      21       1
614   614       2      29       1
675   675      35       1       1
739   739      28      15       1
760   760      46       2       1
820   820      53       8       1
881   881      10      74       1
886   886      86       0       1
941   941     142      15       1
956   956     171      15       1
```

Systematic Samples

Random start – equal size

```
prob.sys <- rep(x=20/975, times=975)
sam.sys <- UPsystematic(pik=prob.sys)
sample.sys <- frame[which(x = (sam.sys == 1)), ]
```

```
> sam.sys
  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
 [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [75] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
…
```

- Interval = 975/20 = 48.75
- Gaps of 48 and 49

```
> sample.sys
    seqno    rent   owner
33     33       1       0
82     82       0       4
131   131       7       0
180   180       0       9
228   228      10       1
277   277       8       5
326   326      16       0
375   375       5      12
423   423       6      14
472   472       8      14
521   521       9      16
570   570       3      25
618   618      14      18
667   667      10      25
716   716      40       0
765   765      37      12
813   813       2      58
862   862      75       2
911   911     114       0
960   960     184      14
```

```
> ## PPS samples ##
>       ## random selections ##
> prob.pps <- inclusionprobabilities(a = frame$owner_hu, n = 20)
Warning message:
In inclusionprobabilities(a = frame$owner_hu, n = 20) :
  there are zero values in the initial vector a

> sam.pps <- UPbrewer(pik = prob.pps)
> sample.pps <- frame[which(x = (sam.pps == 1)),]
> sam.pps
  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
…
```

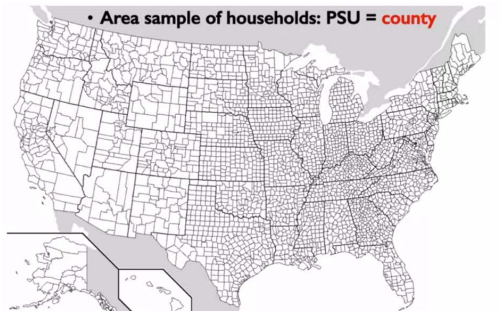
```
> sample.pps
     seqno     rent    owner
149    149        1        6
236    236        0       12
252    252        2       10
403    403        6       13
410    410        0       19
447    447        0       21
491    491        5       19
569    569       12       16
576    576        0       28
620    620        0       32
639    639       10       23
687    687        1       36
693    693        8       29
738    738        3       40
762    762       12       36
920    920       52       71
923    923        5      121
944    944       18      140
962    962       87      120
975    975       41      257
```

```
> sample.ppssys
     seqno     rent    owner
9        9      145      278
181    181        2        7
276    276        0       13
357    357        6       11
417    417        5       14
470    470        7       15
518    518       14       11
563    563        3       24
606    606       27        3
647    647        2       32
687    687        1       36
726    726        1       40
768    768        1       48
804    804        1       57
837    837        2       64
869    869       46       33
919    919      119        2
944    944       18      140
962    962       87      120
973    973       41      234
```

6.2 Stratified Multistage Sampling

• First, identify sampling units:
- PSU (Primary Sampling Units)
- Secondary units within PSU, etc.

e.g. Area Sample of households: PSU = county



• Then stratify at each stage
- stratification as a general purpose tool …
- assure representation
- potentially provide gains in precision

• For example, PSU stratification
• Follow principles similar to element stratification
- use cluster characteristics to **create** homogeneous, mutually exclusive, exhaustive **groups**
- stratifying variables, boundaries etc. follow element sampling stratification principles
- **allocate** sample clusters across strata: proportionate, paired, equal, other (Neyman or minimum variance unusual)
- select samples from each group
- later, after data collection, computer estimates separately for each group – statistic and sampling variance
- **combine** results across groups

• Purpose
- control the distribution of the sample
- decrease sampling variance

e.g. Area sample of households: Stratum = MSA(Metropolitan Statistical Areas)
4 categories: very large – medium sized – smaller sized – non metropolitan

e.g. Area sample of households: Stratification = implicit

• Second stage units
Area sample of households: Second stage units

- within selected PSU
- Census tracts across entire selected county?