Zoning out: an analysis of zoning and property values in Washington, D.C.

Brian Bontempo, Derrick Lee, and Aaron R. Williams

Introduction

1. Need Statement

Washington D.C. is consistently ranked as one of the most-expensive places to live in America (Goetz (2019), Forhlich (2019), Kiersz (2019)). A main driver of affordability challenges in the District is housing. Rents and home purchase prices have grown dramatically during the last two decades and affordability challenges have surged across the community's 68 square miles. At the same, Washington D.C.'s restrictive zoning, height limitation, and excessive historical preservation have constrained housing supply during a period of robust economic and population growth.

This study aims to understand the cost of housing and the nature of zoning in Washington, D.C. and in its four quadrants. As subsequent sections will demonstrate, there are rich sources of information about property values and zoning rules in Washington, D.C., but it is difficult to extract information about population parameters from those sources. A probability sample of residences is to be taken to estimate the value of residential properties and the proportion of properties with zoning that allows for multi-family housing.

2. Target Population

- Clearly describe the *finite* population: All residential units in the District of Columbia.

3. Sampling Frame

1. Address Residential Units (https://opendata.dc.gov/datasets/c3c0ae91dca54c5d9ce56962fa0dd645_68)
2. Address Points (http://opendata.dc.gov/datasets/address-points)

To obtain a list of all residential housing units in Washington, D.C., we sourced two datasets prepared by The District of Columbia Geographical Information System (DC GIS) on June 25th, 2019. The first dataset, Address Residential Units(I), contains all Multifamily residential units and attributes specifically pertaining to units within condominiums and apartments. With a total of 251,180 records, the dataset is extensive as it lists individual units of these Multifamily complexes in DC. The set contains multiple housing units for each street address with units being differentiated by their unit numbers (ie. Apt. 26).

The second dataset, Address Point(II), is a comprehensive list of all primary addresses within D.C. It includes 147,650 records of Single-Family, Multifamily, and non-residential addresses. Unlike the first dataset, the Address Point dataset has more variables to describe each address, such as ward ID, census block ID, and active residential occupancy counts. However, the dataset does not list the individual units within the Multifamily complexes - but rather just lists street addresses for the complexes. An apartment building that shows up as many addresses in the Address Residential Units dataset only shows up once in this data set.

To form our Sampling Frame, we merged the two datasets - extracting key data points from each dataset. This required dropping street addresses from the Address Points dataset that appear in Address Residential Units. This avoided double counting. We were able to generate a final list of all residential housing units in Washington, D.C., comprised of 348,094 Single-family and Multi-family units. The R code in the appendix includes all transformations used to create the final dataset. Both source datasets and the final dataset include quadrant for every obsevration, which will prove valuable for stratifcation.

```
# load necessary packages
library(tidyverse)
library(knitr)
library(urbnmapr)
library(urbnthemes)
library(survey)
```

```
# Download the data
# file path to csv with addresses
aru_file_path <-
  "https://opendata.arcgis.com/datasets/c3c0ae91dca54c5d9ce56962fa0dd645_68.csv"

ap_file_path <-
  "https://opendata.arcgis.com/datasets/aa514416aaf74fdc94748f1e56e7cc8a_0.csv"

# create a directory for downloading the data
if (!dir.exists("data/")) {
  dir.create("data")
}

# if the data doesn't already exist, download the data
if (!file.exists("data/aru.csv")) {
  download.file(aru_file_path, "data/aru.csv")
}

if (!file.exists("data/ap.csv")) {
  download.file(ap_file_path, "data/ap.csv")
}
```

Address Residential Units

The first dataset is Address Residential Units

The dataset does not contain a variable for quadrant, so we extract quadrant from the full address.

```
aru <- read_csv("data/aru.csv") %>%
  rename_all(tolower) %>%
  select(unit_id, address_id, fulladdress, status, unitnum, unittype)

# extract quadrant
aru <- aru %>%
  mutate(quadrant = str_sub(fulladdress, start = -2, end = -1))
```

Address Residential Units contains residential units with status set to "RETIRED". We drop these cases as well.

```
count(aru, status) %>%
  kable()
```

status	n
ACTIVE	244046
ASSIGNED	47
RETIRE	7087

```
aru <- aru %>%
  filter(status != "RETIRE")
```

Address Points

```
# load the data and convert the variable names to lower case
ap <- read_csv("data/ap.csv", guess_max = 10000) %>%
  rename_all(tolower) %>%
  select(address_id, status, type_, entrancetype, quadrant, fulladdress,
         objectid_1, assessment_nbhd, cfsa_name, census_tract, vote_prcnct,
         ward, zipcode, anc, census_block, census_blockgroup, latitude,
         longitude, active_res_unit_count, res_type, active_res_occupancy_count)
```

Address Points contains residential units, non-residential units, and mixed-use units. Residential units and mixed-use units contain residences that belong to our sampling frame. We drop non-residential units.

```
count(ap, res_type) %>%
  kable()
```

res_type	n
MIXED USE	473
NON RESIDENTIAL	15807
RESIDENTIAL	131370

```
ap <- ap %>%
  filter(res_type != "NON RESIDENTIAL")
```

Address points contains residential units with status set to "RETIRED". We drop these cases as well.

```r
count(ap, status) %>%
  kable()
```

status	n
ACTIVE	128490
ASSIGNED	668
RETIRE	2675
TEMPORARY	10

```r
ap <- ap %>%
  filter(status != "RETIRE")
```

After the above filtering, there are 98 observations from Address Points and 3,706 observations in Address Residential Units that have missing addresses. We investigated joining the two datasets on `address_id` to fill in the address but all records missing an address in one dataset were missing an address in the other dataset.

We dropped the missing values which represented about 1.5 percent of observations in Address Residential Units and 0.07 percent of observations in Address Points.

```r
ap <- ap %>%
  filter(!is.na(fulladdress))

aru <- aru %>%
  filter(!is.na(fulladdress))
```

Merge variables

Address Points has interesting variables not present in Address Residential Units. So we merge the Address Points dataset with the Address Residential Units dataset. The join works for all but 572 cases, most of which are in a new building at the Wharf.

```r
aru_expanded <- aru %>%
  select(-status) %>%
  left_join(ap, by = c("fulladdress", "address_id")) %>%
  select(quadrant = quadrant.x, everything(), -quadrant.y)

anti_join(aru, ap, by = c("fulladdress", "address_id"))
```

```
## # A tibble: 572 x 7
##    unit_id address_id fulladdress         status unitnum unittype quadrant
##      <dbl>      <dbl> <chr>               <chr>  <chr>   <chr>    <chr>
##  1  223379     276680 600 WATER STREET SW ACTIVE 6-12    RENTAL   SW
##  2  223380     276680 600 WATER STREET SW ACTIVE 6-13    RENTAL   SW
##  3  223381     276680 600 WATER STREET SW ACTIVE 6-14    RENTAL   SW
##  4  223384     276680 600 WATER STREET SW ACTIVE 1-1     RENTAL   SW
```

```
##  5  223389     276680 600 WATER STREET SW ACTIVE 1-6     RENTAL   SW
##  6  223392     276680 600 WATER STREET SW ACTIVE 1-9     RENTAL   SW
##  7  223494     276680 600 WATER STREET SW ACTIVE 8-16    RENTAL   SW
##  8  223497     276680 600 WATER STREET SW ACTIVE 9-3     RENTAL   SW
##  9  223503     276680 600 WATER STREET SW ACTIVE 9-9     RENTAL   SW
## 10  223508     276680 600 WATER STREET SW ACTIVE 9-14    RENTAL   SW
## # ... with 562 more rows
```

```
rm(aru)
```

Combination

Next, we need to drop addresses in the Address Points dataset that exist in the Address Residential Units dataset so we don't over count addresses in multi-dwelling units.

```
ap <- ap %>%
  filter(!address_id %in% unique(aru_expanded$address_id))
```

Finally, we can combine the two datasets to create a sampling frame that contains approximately every residential address in Washington D.C.

```
sampling_frame <- bind_rows(ap, aru_expanded)

rm(ap, aru_expanded)

#summarize_all(addresses, list(~sum(is.na(.))))

write_csv(sampling_frame, "sampling_frame.csv")
```

4. Primary Parameters

- D.C. sample mean of property values
- Quadrant sample mean of property values
- D.C. sample proportion of multi-family zoning
- Quadrant sample proportion of multi-family zoning

5. Questionnaire/Instrument

6. Pilot Survey

```
set.seed(20190714)

pilot_sample <- sampling_frame %>%
  group_by(quadrant) %>%
```

```
  sample_n(25)

write_csv(pilot_sample, "data/pilot_sample.csv")

rm(pilot_sample)
```

```
# load the completed pilot survey and clean the values
pilot_sample <- read_csv("data/pilot_sample_completed.csv") %>%
  mutate(land_value = ifelse(!is.na(rf_land_value),
                             rf_land_value,
                             land_value),
         improvement_value = ifelse(!is.na(rf_improvement_value),
                                    rf_improvement_value,
                                    improvement_value)) %>%
  mutate(property_value = land_value + improvement_value) %>%
  mutate(property_value = ifelse(unittype == "RENTAL" &
                                   active_res_occupancy_count > 4 &
                                   property_value > 500000,
                                 property_value / active_res_occupancy_count,
                                 property_value
                                 ))
```

```
pilot_sample %>%
  summarize(mean = mean(property_value, na.rm = TRUE),
            s_squared_h = var(property_value, na.rm = TRUE),
            missing_prop = mean(is.na(property_value))) %>%
  kable(caption = "Pilot survey summary statistics")
```

Table 4: Pilot survey summary statistics

mean	s_squared_h	missing_prop
454852.5	259886899569	0.17

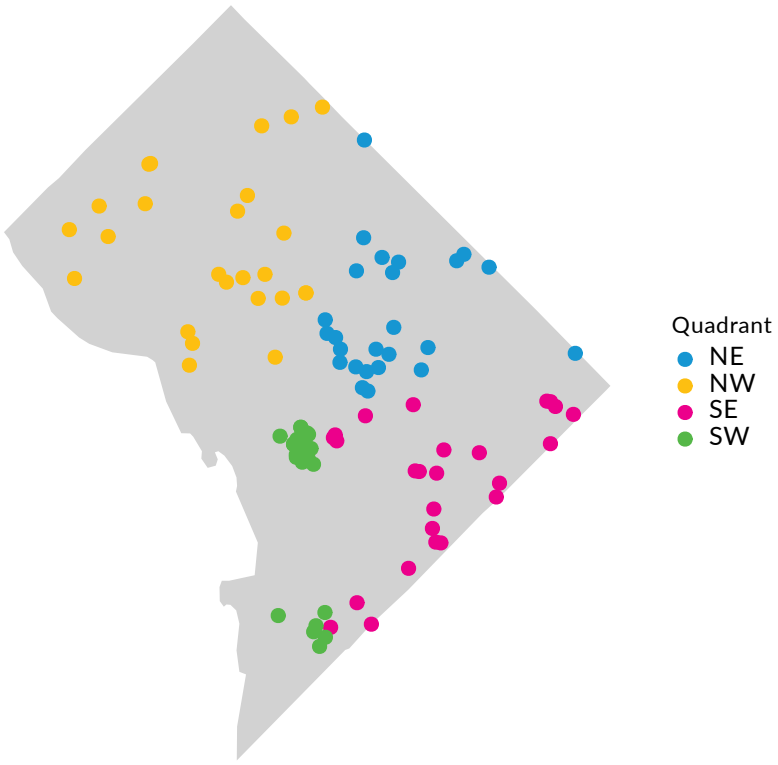
```
pilot_sample %>%
  group_by(quadrant) %>%
  summarize(mean = mean(property_value, na.rm = TRUE),
            s_squared_h = var(property_value, na.rm = TRUE),
            missing_prop = mean(is.na(property_value))) %>%
  kable(caption = "Pilot survey summary statistics by quadrant")
```

Table 5: Pilot survey summary statistics by quadrant

quadrant	mean	s_squared_h	missing_prop
NE	408489.5	55231295979	0.08
NW	781327.7	715270634804	0.12
SE	305901.6	71519718277	0.28
SW	283103.1	25025018879	0.20

Map of the pilot survey sample units

```
states %>%
  filter(state_name == "District of Columbia") %>%
  ggplot() +
  geom_polygon(aes(x = long, y = lat, group = group),
               fill = "#d2d2d2",
               size = 0.3) +
  geom_point(data = pilot_sample,
             aes(x = longitude, y = latitude, color = quadrant),
             size = 2) +
  scale_color_manual(values = palette_urbn$categorical[[6]][c(1, 2, 5, 6)]) +
  coord_map() +
  labs(color = "Quadrant",
       x = NULL,
       y = NULL) +
  theme_urbn_map()
```



7. Determination of Sample and Strata Sizes

We are interested in estimating the sample mean of property values for all D.C. residences using stratification, the sample means of property values for all residences in each quadrant of

D.C., the sample proportion of residences with different zoning types for all D.C. residences using stratifcation, and the sample proportions of residences with different zoning types in each quadrant of D.C.. All four of these four estimation processes requires allocating a certain number of sample units to each strata and each process has a different optimal allocation.

For the stratified estimates we will specify a maximum variance, V_0, and use Exact Optimal Allocation for $\bar{y}_{str}$ and $\hat{p_{str}}$. For the estimates within each quadrant, we will specify an error, e, and calculate an appropriate sample size for simple random sampling and a 90 percent confidence interval. Finally, we will align all four optimal allocations and take the maximum n_h for each quadrant. This won't necessarily result in an optimal allocation for any one of our four estimates but it will ensure that our allocation is adequate for each of our four estimates.

Condition 1: Sample mean

We begin with a derivation of Exact Optimal Sample Allocation for $\bar{y}$.

Decomposition of $V(\bar{y}_h)$:

By Wright (12.4), $V(\bar{y}_{str}) = \sum_{h=1}^{H}(\frac{N_h}{N})^2 V(\bar{y}_h) = \sum_{h=1}^{H}(\frac{N_h}{N})^2 \frac{N_h - n_h}{N_h}\frac{S_h^2}{n_h}$

$V(\bar{y}_h) = (\frac{N_h}{N})^2 \frac{N_h - n_h}{N_h}\frac{S_h^2}{n_h}$

$V(\bar{y}_h) = (\frac{N_h^2}{N^2})(1 - \frac{n_h}{N_h})\frac{S_h^2}{n_h}$

$V(\bar{y}_h) = (\frac{N_h^2 S_h^2}{N^2})(\frac{1}{n_h}) - \frac{N_h^2 n_h S_h^2}{N^2 N_h n_h}$

$V(\bar{y}_h) = (\frac{N_h^2 S_h^2}{N^2})(\frac{1}{n_h}) - \frac{N_h S_h^2}{N^2}$

$V(\bar{y}_h) = (\frac{N_h^2 S_h^2}{N^2})(1 - \frac{1}{1\cdot 2} - \frac{1}{2\cdot 3} - ... - \frac{1}{n_h(n_h-1)}) - \frac{N_h S_h^2}{N^2}$

$V(\bar{y}_h) = \frac{N_h(N_h-1)S_h^2}{N^2} - \frac{N_h^2 S_h^2}{N^2\cdot 1\cdot 2} - \frac{N_h^2 S_h^2}{N^2\cdot 2\cdot 3} - ... - -\frac{N_h^2 S_h^2}{N^2 n_h(n_h-1)}$

Decomposition of $V(\bar{y}_{str})$

$V(\bar{y}_{str}) = \sum_{h=1}^{H} \frac{N_h(N_h-1)S_h^2}{N^2}$

$-\frac{N_1^2 S_1^2}{N^2\cdot 1\cdot 2} - \frac{N_1^2 S_1^2}{N^2\cdot 2\cdot 3} - ... - -\frac{N_1^2 S_1^2}{N^2 n_1(n_1-1)}$

...

$-\frac{N_h^2 S_h^2}{N^2\cdot 1\cdot 2} - \frac{N_h^2 S_h^2}{N^2\cdot 2\cdot 3} - ... - -\frac{N_h^2 S_h^2}{N^2 n_h(n_h-1)}$

...

$-\frac{N_H^2 S_H^2}{N^2\cdot 1\cdot 2} - \frac{N_H^2 S_H^2}{N^2\cdot 2\cdot 3} - ... - -\frac{N_H^2 S_H^2}{N^2 n_H(n_H-1)}$

For a desired bound V_0 on the sampling variance $V(\bar{y}_{str})$, we may find an optimal allocation using the following algorithm:

1) Assign, for each stratum, 1 unit to be selected for the sample.

2) Fill in the following table and number these values starting from 1, in decreasing order.

$\frac{N_1^2 S_1^2}{N^2 \cdot 1 \cdot 2}$	$\frac{N_1^2 S_1^2}{n^2 \cdot 2 \cdot 3}$	$\frac{N_1^2 S_1^2}{N^2 \cdot 3 \cdot 4}$	$\cdots$
$\frac{N_2^2 S_2^2}{N^2 \cdot 1 \cdot 2}$	$\frac{N_2^2 S_2^2}{N^2 \cdot 2 \cdot 3}$	$\frac{N_2^2 S_2^2}{N^2 \cdot 3 \cdot 4}$	$\cdots$
.	.	.	$\cdots$
.	.	.	$\cdots$
.	.	.	$\cdots$
$\frac{N_H^2 S_H^2}{N^2 \cdot 1 \cdot 2}$	$\frac{N_H^2 S_H^2}{N^2 \cdot 2 \cdot 3}$	$\frac{N_H^2 S_H^2}{N^2 \cdot 3 \cdot 4}$	$\cdots$

3) Since the initial allocation is $(n_{11}, n_{21}, ..., n_{H1}) = (1, 1, ..., 1)$, compute $V(\bar{y}_{str}|n_{11} = 1, n_{21} = 1, ..., n_{H1} = 1) = \sum_{h=1}^{H} \frac{N_h(N_h - 1)S_h^2}{N^2}$

4) Pick value (1) from the table and increase the associated stratum's sample size by 1, o that the updated allocation is $(n_{12}, n_{22}, ..., n_{H2})$, where exactly one of the n_{h2}'s is equal to 2 and the rest are equal to 1. Then, compute $V(\bar{y}_{str}|n_{12}, ..., n_{H2} = V(\bar{y}_{str}|n_{11}, ..., n_{H1}) - \frac{1}{N^2}$ where "(1)" represents the largest value from the table. If $V(\bar{y}_{str}|N_{12}, ..., n_{H2} \leq V_0$, then stop with $n_1 = n_{12}, ..., N_H = N_{H2}$. Otherwise, go to step 5.

5) Pick value (2) from the table and increase the associated stratum's sample size by 1, so that the updated allocation is $(n_{13}, ..., n_{H3})$. Then compute $V(\bar{y}_{str}|n_{13}, ..., n_{H3}) = V(\bar{y}_{str}|n_{12}, ..., n_{H2} - \frac{(2)}{N^2}$, where "(2)" represents the second value from the table. If $V(\bar{y}_{str}|n_{13}, ..., N_H = n_{H3}$. Otherwise, continue until step j, where $V(\bar{y}_s tr|n_{1j}, ..., n_{Hj}) \leq V_0$. The final allocation is $n_{1j}, ..., n_{Hj}$ and $n = n_{1j} + \cdots + n_{Hj}$.

```
# find Nh and s2 for each strata
# (1) and (2)
s_squared_h <- pilot_sample %>%
  group_by(stratum = quadrant) %>%
  summarize(s_squared_h = var(property_value, na.rm = TRUE),
            missing_prop = mean(is.na(property_value)))

Nh <- sampling_frame %>%
  count(stratum = quadrant) %>%
  rename(Nh = n)

strata <- left_join(s_squared_h, Nh, by = "stratum") %>%
  # adjust N because of missingness
```

```
  mutate(Nh = Nh * (1 - missing_prop)) %>%
  mutate(N = sum(Nh))

rm(s_squared_h, Nh)

kable(strata)
```

stratum	s_squared_h	missing_prop	Nh	N
NE	55231295979	0.08	68953.08	297153.2
NW	715270634804	0.12	166931.60	297153.2
SE	71519718277	0.28	49423.68	297153.2
SW	25025018879	0.20	11844.80	297153.2

Step 3: $\hat{V}(\bar{y}|1,1,1,1) = \sum_{h=1}^{H}(\frac{N_h}{N})^2 \frac{N_h - n_h}{N_H} \frac{s_h^2}{n_h} = \sum_{h=1}^{H}(\frac{N_h}{N})^2 \frac{N_h - 1}{N_H} \frac{s_h^2}{1}$

(Wright 12.5)

```
# Let the initial allocation be (n_11, n_21, n_31, n_41) = (1, 1, 1, 1)
# (3) and (6)
starting_variance <- strata %>%
  mutate(strata_variance = Nh * (Nh - 1) * s_squared_h / N^2)

kable(starting_variance)
```

stratum	s_squared_h	missing_prop	Nh	N	strata_variance
NE	55231295979	0.08	68953.08	297153.2	2973894581
NW	715270634804	0.12	166931.60	297153.2	225727358777
SE	71519718277	0.28	49423.68	297153.2	1978456290
SW	25025018879	0.20	11844.80	297153.2	39758730

```
starting_variance <- starting_variance %>%
  summarize(V = sum(strata_variance)) %>%
  pull()

starting_variance
```

```
## [1] 230719468378
```

Step 3:

Priority value $= \frac{N_1^2 \cdot s_1^2}{N_1^2 \cdot n_h(n_h - 1)}$

```
# create a table of priority values
# (4) and (5)
n_strata <-
  tibble(stratum = rep(strata$stratum, strata$Nh)) %>%
```

```
  group_by(stratum) %>%
  mutate(n = row_number()) %>%
  ungroup() %>%
  left_join(strata, by = "stratum")

# step 2
priority_values <- n_strata %>%
  group_by(stratum) %>%
  # rewritten to avoid integer overflow
  # mutate(priority_value = (Nh ^ 2 * s_squared_h) / (n * lag(n) * N ^ 2)) %>%
  mutate(priority_value = (Nh ^ 2 / n) * (s_squared_h / lag(n)) * (1 / N ^ 2)) %>%
  ungroup() %>%
  arrange(desc(priority_value))

kable(head(select(priority_values, -missing_prop), n = 10))
```

stratum	n	s_squared_h	Nh	N	priority_value
NW	2	715270634804	166931.6	297153.2	112864355500
NW	3	715270634804	166931.6	297153.2	37621451833
NW	4	715270634804	166931.6	297153.2	18810725917
NW	5	715270634804	166931.6	297153.2	11286435550
NW	6	715270634804	166931.6	297153.2	7524290367
NW	7	715270634804	166931.6	297153.2	5374493119
NW	8	715270634804	166931.6	297153.2	4030869839
NW	9	715270634804	166931.6	297153.2	3135120986
NW	10	715270634804	166931.6	297153.2	2508096789
NW	11	715270634804	166931.6	297153.2	2052079191

Step 4:

```
# (7)
priority_values <- priority_values %>%
  mutate(agg_priority_value = cumsum(priority_value)) %>%
  mutate(marginal_variance = starting_variance - agg_priority_value) %>%
  mutate(marginal_sd = sqrt(marginal_variance))

kable(head(select(priority_values, -missing_prop, -N), n = 10), digits = 0)
```

stratum	n	s_squared_h	Nh	priority_value	agg_priority_value	marginal_variance	marginal_sd
NW	2	715270634804	166932	112864355500	112864355500	117855112878	343300
NW	3	715270634804	166932	37621451833	150485807333	80233661045	283255
NW	4	715270634804	166932	18810725917	169296533250	61422935128	247837
NW	5	715270634804	166932	11286435550	180582968800	50136499578	223912
NW	6	715270634804	166932	7524290367	188107259166	42612209212	206427
NW	7	715270634804	166932	5374493119	193481752285	37237716093	192971
NW	8	715270634804	166932	4030869839	197512622125	33206846253	182227
NW	9	715270634804	166932	3135120986	200647743111	30071725267	173412
NW	10	715270634804	166932	2508096789	203155839900	27563628478	166023
NW	11	715270634804	166932	2052079191	205207919091	25511549287	159723

```
rm(n_strata)
```

```
condition1 <- priority_values %>%
  mutate(stratum = factor(stratum)) %>%
  filter(marginal_variance >= ((0.1 * (mean(pilot_sample$property_value, na.rm = TRUE))) ^ 2))

condition1 <- condition1 %>%
  count(stratum, .drop = FALSE)
```

Condition 2: Sample means within strata

We are interested in comparing $\bar{y}_h$ from the four different quadrants.

$n = \frac{N\sigma^2}{(N-1)\frac{e^2}{z^2_{\frac{\alpha}{2}}}+\sigma^2}$

We can use s^2 from our pilot survey as an unbiased estimate for σ^2.

$n = \frac{Ns^2}{(N-1)\frac{e^2}{z^2_{\frac{\alpha}{2}}}+s^2}$

We want \$120,000 precision at a 90% confidence level for the mean of property value in each strata.

```
condition2 <- strata %>%
  mutate(n = (Nh * s_squared_h) / ((Nh - 1) * (120000 ^ 2 / qnorm(0.95) ^ 2) + s_squared_h))

condition2 %>%
  kable()
```

stratum	s_squared_h	missing_prop	Nh	N	n
NE	55231295979	0.08	68953.08	297153.2	10.375719
NW	715270634804	0.12	166931.60	297153.2	134.281297
SE	71519718277	0.28	49423.68	297153.2	13.434099
SW	25025018879	0.20	11844.80	297153.2	4.700356

Condition 3: Sample proportion

We begin with a derivation of Exact Optimal Sample Allocation for $\hat{p}$.

Decomposition of $V(\hat{p}_{str})$

By Wright (12.14), $V(\hat{p}_{str}) = \sum_{h=1}^{H}(\frac{N_h}{N})^2 V(p_h) = \sum_{h=1}^{H}(\frac{N_h}{N})^2 \frac{N_h-n_h}{N_h-1}\frac{p(1-p)}{n_h}$

$V(\hat{p}_h) = (\frac{N_h}{N})^2 \frac{N_h-n_h}{N_h-1}\frac{p(1-p)}{n_h}$

$V(\hat{p}_h) = \frac{N_h^2}{N^2}\frac{N_h}{N_h-1}\frac{p(1-p)}{n_h} - \frac{N_h^2}{N^2}\frac{n_h}{N_h-1}\frac{p(1-p)}{n_h}$

$$V(\hat{p}_h) = \frac{N_h^2}{N^2}\frac{N_h}{N_h-1}\frac{p(1-p)}{n_h} - \frac{N_h^2}{N^2}\frac{n_h}{N_h-1}\frac{p(1-p)}{n_h}$$

$$V(\hat{p}_h) = \frac{N_h^3 p(1-p)}{N^2(N_h-1)}\frac{1}{n_h} - \frac{N_h^2 p(1-p)}{N^2(N_h-1)}$$

$$V(\hat{p}_h) = \frac{N_h^3 p(1-p)}{N^2(N_h-1)}(1 - \frac{1}{1\cdot 2} - \frac{1}{2\cdot 3} - \cdots - \frac{1}{n_h(n_h-1)}) - \frac{N_h^2 p(1-p)}{N^2(N_h-1)}$$

$$V(\hat{p}_h) = \frac{N_h^3 p(1-p)}{N^2(N_h-1)} - \frac{N_h^3 p(1-p)}{N^2(N_h-1)\cdot 1\cdot 2} - \frac{N_h^3 p(1-p)}{N^2(N_h-1)\cdot 2\cdot 3} - \cdots - \frac{N_h^3 p(1-p)}{N^2(N_h-1)\cdot n_h(n_h-1)} - \frac{N_h^2 p(1-p)}{N^2(N_h-1)}$$

$$V(\hat{p}_h) = \frac{(N_h^3-N_h^2) p(1-p)}{N^2(N_h-1)} - \frac{N_h^3 p(1-p)}{N^2(N_h-1)\cdot 1\cdot 2} - \frac{N_h^3 p(1-p)}{N^2(N_h-1)\cdot 2\cdot 3} - \cdots - \frac{N_h^3 p(1-p)}{N^2(N_h-1)\cdot n_h(n_h-1)}$$

$$V(\hat{p}_h) = \frac{N_h^2(N_h-1) p(1-p)}{N^2(N_h-1)} - \frac{N_h^3 p(1-p)}{N^2(N_h-1)\cdot 1\cdot 2} - \frac{N_h^3 p(1-p)}{N^2(N_h-1)\cdot 2\cdot 3} - \cdots - \frac{N_h^3 p(1-p)}{N^2(N_h-1)\cdot n_h(n_h-1)}$$

Decomposition of $V(\hat{p}_{str})$

$$V(\hat{p}_{str}) = \sum_{h=1}^{H} \frac{N_h^2(N_h-1)p(1-p)}{N^2(N_h-1)}$$

$$-\frac{N_1^3 p(1-p)}{N^2(N_1-1)\cdot 1\cdot 2} - \frac{N_1^3 p(1-p)}{N^2(N_1-1)\cdot 2\cdot 3} - \ldots - \frac{N_1^3 p(1-p)}{N^2(N_1-1) n_h(n_h-1)}$$

$$\ldots$$

$$-\frac{N_h^3 p(1-p)}{N^2(N_h-1)\cdot 1\cdot 2} - \frac{N_h^3 p(1-p)}{N^2(N_h-1)\cdot 2\cdot 3} - \ldots - \frac{N_h^3 p(1-p)}{N^2(N_h-1) n_h(n_h-1)}$$

$$\ldots$$

$$-\frac{N_H^3 p(1-p)}{N^2(N_H-1)\cdot 1\cdot 2} - \frac{N_H^3 p(1-p)}{N^2(N_H-1)\cdot 2\cdot 3} - \ldots - \frac{N_H^3 p(1-p)}{N^2(N_H-1) n_h(n_h-1)}$$

For a desired bound on V_0 on the sampling variance $V(\hat{p}_{str})$, we may find an optimal allocation using the following algorithm:

1) Assign, for each stratum, 1 unit to be selected for the sample.

2) Fill in the following table and number these values starting from 1, in decreasing order. We assume $p_h = 0.5$ because that is where the variance reaches its global maximum.

$\frac{\frac{1}{4}N_1^3}{N^2(N_1-1)\cdot 1\cdot 2}$	$\frac{\frac{1}{4}N_1^3}{N^2(N_1-1)\cdot 2\cdot 3}$	$\frac{\frac{1}{4}N_1^3}{N^2(N_1-1)\cdot 3\cdot 4}$	$\cdots$
$\frac{\frac{1}{4}N_2^3}{N^2(N_2-1)\cdot 1\cdot 2}$	$\frac{\frac{1}{4}N_2^3}{N^2(N_2-1)\cdot 2\cdot 3}$	$\frac{\frac{1}{4}N_2^3}{N^2(N_2-1)\cdot 3\cdot 4}$	$\cdots$
$\cdot$	$\cdot$	$\cdot$	$\cdots$
$\cdot$	$\cdot$	$\cdot$	$\cdots$
$\cdot$	$\cdot$	$\cdot$	$\cdots$
$\frac{\frac{1}{4}N_H^3}{N^2(N_H-1)\cdot 1\cdot 2}$	$\frac{\frac{1}{4}N_H^3}{N^2(N_H-1)\cdot 2\cdot 3}$	$\frac{\frac{1}{4}N_H^3}{N^2(N_H-1)\cdot 3\cdot 4}$	$\cdots$

3) Since the initial allocation is $(n_{11}, n_{21}, ..., n_{H1}) = (1, 1, ..., 1)$, compute $V(\hat{p}_{str}|n_{11} = 1, n_{21} = 1, ..., n_{H1} = 1) = \frac{1}{N^2}\sum_{h=1}^{H}((N_h^2 - N_h)S_h^2)$

4) Pick value (1) from the table and increase the associated stratum's sample

size by 1, o that the updated allocation is $(n_{12}, n_{22}, ..., n_{H2})$, where exactly one of the n_{h2}'s is equal to 2 and the rest are equal to 1. Then, compute $V(\hat{p}_{str}|n_{12}, ..., n_{H2} = V(\hat{p}_{str}|n_{11}, ..., n_{H1}) - \frac{1}{N^2}$ where "(1)" represents the largest value from the table. If $V(\hat{p}_{str}|N_{12}, ..., n_{H2} \leq V_0$, then stop with $n_1 = n_{12}, ..., N_H = N_{H2}$. Otherwise, go to step 5.

5) Pick value (2) from the table and increase the associated stratum's sample size by 1, so that the updated allocation is $(n_{13}, ..., n_{H3})$. Then compute $V(\hat{p}_{str}|n_{13}, ..., n_{H3}) = V(\hat{p}_{str}|n_{12}, ..., n_{H2} - \frac{(2)}{N^2}$, where "(2)" represents the second value from the table. If $V(\hat{p}_{str}|n_{13}, ..., N_H = n_{H3}$. Otherwise, continue until step j, where $V(\hat{p}_{str}|n_{1j}, ..., n_{Hj}) \leq V_0$. The final allocation is $n_{1j}, ..., n_{Hj})$ and $n = n_{1j} + \cdots + n_{Hj}$.

```
#
strata <- sampling_frame %>%
  count(stratum = quadrant) %>%
  rename(Nh = n) %>%
  mutate(N =sum(Nh),
         s_squared_h = 0.5 * (1 - 0.5))

kable(strata)
```

stratum	Nh	N	s_squared_h
NE	74949	348094	0.25
NW	189695	348094	0.25
SE	68644	348094	0.25
SW	14806	348094	0.25

```
# Let the initial allocation be (n_11, n_21, n_31, n_41) = (1, 1, 1, 1)
# (3) and (6)
starting_variance <- strata %>%
  mutate(strata_variance = (Nh ^ 2 * (Nh - 1) * 0.25) / (N ^ 2 * (Nh - 1)))

kable(starting_variance)
```

stratum	Nh	N	s_squared_h	strata_variance
NE	74949	348094	0.25	0.0115899
NW	189695	348094	0.25	0.0742435
SE	68644	348094	0.25	0.0097219
SW	14806	348094	0.25	0.0004523

```
starting_variance <- starting_variance %>%
  summarize(V = sum(strata_variance)) %>%
  pull()

starting_variance
```

```
## [1] 0.09600763
```

```
# create a table of priority values
# (4) and (5)

n_strata <-
  sampling_frame %>%
  count(quadrant)

n_strata <- tibble(stratum = rep(n_strata$quadrant, n_strata$n)) %>%
  group_by(stratum) %>%
  mutate(n = row_number()) %>%
  left_join(strata, by = "stratum")

# step 2
priority_values <- n_strata %>%
  group_by(stratum) %>%
  mutate(priority_value = (0.25 * Nh ^ 3) / (N ^ 2 * (Nh - 1) * n * lag(n))) %>%
  ungroup() %>%
  arrange(desc(priority_value))

kable(head(priority_values, n = 10))
```

stratum	n	Nh	N	s_squared_h	priority_value
NW	2	189695	348094	0.25	0.0371220
NW	3	189695	348094	0.25	0.0123740
NW	4	189695	348094	0.25	0.0061870
NE	2	74949	348094	0.25	0.0057950
SE	2	68644	348094	0.25	0.0048610
NW	5	189695	348094	0.25	0.0037122
NW	6	189695	348094	0.25	0.0024748
NE	3	74949	348094	0.25	0.0019317
NW	7	189695	348094	0.25	0.0017677
SE	3	68644	348094	0.25	0.0016203

```
# (7)
priority_values <- priority_values %>%
  mutate(agg_priority_value = cumsum(priority_value)) %>%
  mutate(marginal_variance = starting_variance - agg_priority_value) %>%
  mutate(marginal_sd = sqrt(marginal_variance))

kable(head(select(priority_values, -N), n = 10), align = "l")
```

stratum	n	Nh	s_squared_h	priority_value	agg_priority_value	marginal_variance	marginal_sd
NW	2	189695	0.25	0.0371220	0.0371220	0.0588857	0.2426637
NW	3	189695	0.25	0.0123740	0.0494960	0.0465117	0.2156657
NW	4	189695	0.25	0.0061870	0.0556830	0.0403247	0.2008101
NE	2	74949	0.25	0.0057950	0.0614780	0.0345297	0.1858216
SE	2	68644	0.25	0.0048610	0.0663390	0.0296686	0.1722459

stratum	n	Nh	s_squared_h	priority_value	agg_priority_value	marginal_variance	marginal_sd
NW	5	189695	0.25	0.0037122	0.0700512	0.0259564	0.1611100
NW	6	189695	0.25	0.0024748	0.0725260	0.0234816	0.1532372
NE	3	74949	0.25	0.0019317	0.0744577	0.0215500	0.1467991
NW	7	189695	0.25	0.0017677	0.0762254	0.0197823	0.1406494
SE	3	68644	0.25	0.0016203	0.0778457	0.0181619	0.1347661

```
rm(n_strata)
```

```
condition3 <- priority_values %>%
  filter(marginal_variance >= ((0.1 * 0.5) ^ 2))

condition3 <- count(condition3, stratum)
```

Condition 4: Sample proportion within strata

We are interested in comparing $\hat{p}_h$ from the four different quadrants.

$n = \frac{Np(1-p)}{(N-1)\frac{e^2}{z^2_{\frac{\alpha}{2}}}+p(1-p)}$

We can assume that $p = 0.5$.

$n = \frac{\frac{1}{4}N}{(N-1)\frac{e^2}{z^2_{\frac{\alpha}{2}}}+\frac{1}{4}}$

We want 0.1 precision at a 90% confidence level for the mean of proportion with multi-family zoning in each strata.

```
condition4 <- strata %>%
  mutate(n = (Nh * 0.25) / ((Nh - 1) * (0.1 ^ 2 / qnorm(0.95) ^ 2) + 0.25))

condition4 %>%
  kable()
```

stratum	Nh	N	s_squared_h	n
NE	74949	348094	0.25	67.57850
NW	189695	348094	0.25	67.61483
SE	68644	348094	0.25	67.57299
SW	14806	348094	0.25	67.33552

Combining the above conditions

We want to sample at a rate that meets the four different requirements from above

1. $V_0 > V(\bar{y}_{str})$ for the sample mean

2. \$50,000 precision at a 90% confidence level for $\bar{y}_h$ in each strata
3. $V_0 > V(\hat{p}_h)$ for the sample proportion
4. 0.1 precision at a 90% confidence level for $\hat{p}$ in each strata

```
tibble(quadrant = condition1$stratum,
       `1.` = condition1$n,
       `2.` = condition2$n,
       `3.` = condition3$n,
       `4.` = condition4$n) %>%
  kable(caption = "Recommended strata sizes across the four conditions")
```

Table 18: Recommended strata sizes across the four conditions

quadrant	1.	2.	3.	4.
NE	14	10.375719	21	67.57850
NW	132	134.281297	53	67.61483
SE	11	13.434099	19	67.57299
SW	1	4.700356	3	67.33552

```
nh <- tibble(quadrant = condition1$stratum,
       `1.` = condition1$n,
       `2.` = condition2$n,
       `3.` = condition3$n,
       `4.` = condition4$n) %>%
  gather(key = "key", value = "nh", -quadrant) %>%
  group_by(quadrant) %>%
  summarize(nh = ceiling(max(nh)))

nh %>%
  kable(caption = "Maximum recommended strata sizes across the four conditions")
```

Table 19: Maximum recommended strata sizes across the four conditions

quadrant	nh
NE	68
NW	135
SE	68
SW	68

8. Sampling Plan

```
survey <- group_split(sampling_frame, quadrant) %>%
  map2_df(nh$nh, sample_n)

survey %>%
```

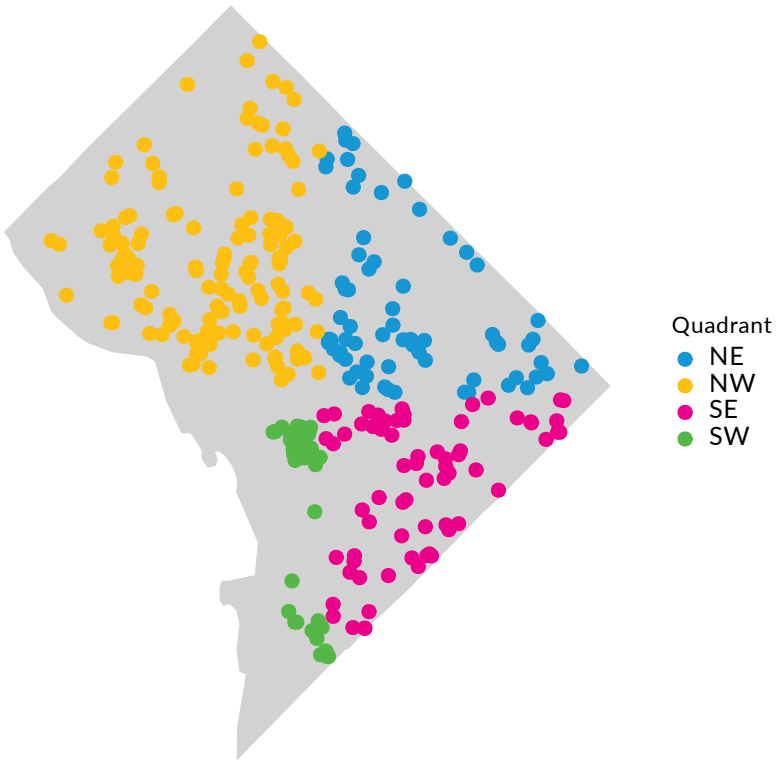
```
  sample_n(10) %>%
  select(status, fulladdress, res_type) %>%
  kable()
```

status	fulladdress	res_type
ACTIVE	57 N STREET NW	RESIDENTIAL
ACTIVE	222 M STREET SW	MIXED USE
ACTIVE	1019 LAMONT STREET NW	RESIDENTIAL
ACTIVE	3629 S STREET NW	RESIDENTIAL
ACTIVE	300 M STREET SW	RESIDENTIAL
ACTIVE	1305 CONGRESS STREET SE	RESIDENTIAL
ACTIVE	307 17TH STREET SE	RESIDENTIAL
ACTIVE	800 4TH STREET SW	RESIDENTIAL
ACTIVE	711 E STREET SE	RESIDENTIAL
ACTIVE	3130 WISCONSIN AVENUE NW	RESIDENTIAL

```
write_csv(survey, "data/survey.csv")
```

Map of the survey sample units

```
states %>%
  filter(state_name == "District of Columbia") %>%
  ggplot() +
  geom_polygon(aes(x = long, y = lat, group = group),
               fill = "#d2d2d2",
               size = 0.3) +
  geom_point(data = survey,
             aes(x = longitude, y = latitude, color = quadrant),
             size = 2) +
  scale_color_manual(values = palette_urbn$categorical[[6]][c(1, 2, 5, 6)]) +
  coord_map() +
  labs(color = "Quadrant",
       x = NULL,
       y = NULL) +
  theme_urbn_map()
```



9. Estimation

```
Nh <- count(sampling_frame, quadrant)

represented_zones <- read_csv("represented-zones.csv")

final_survey <- read_csv("data/final-survey.csv") %>%
  mutate(property_value = land_value + improvement_value) %>%
  mutate(property_value = ifelse(unittype == "RENTAL" &
                                   active_res_occupancy_count > 4 &
                                   property_value > 500000,
                                 property_value / active_res_occupancy_count,
                                 property_value
                                 ))

final_survey <- left_join(final_survey, Nh, by = "quadrant") %>%
  rename(fpc = n)

final_survey <- left_join(final_survey, represented_zones, by = "zoning")


strat_design <- svydesign(id = ~1, data = final_survey, strata = ~quadrant, fpc = ~fpc)
```

Sample statistics

$\bar{y}_{str}$

The first parameter we estimated was the stratifed sample mean of the appraised property value in Washington DC. We also estimated a 95% confidence interval for the parameter.

$\bar{y}_{str} = \sum_{h=1}^{H} \frac{N_h}{N} \bar{y}_h$

$\hat{V}(\bar{y}_h) = \sum_{h=1}^{H} (\frac{N_h}{N})^2 \frac{N_h - n_h}{N_h} \frac{s_h^2}{n_h}$

$(\bar{y}_{str} - Z_{\frac{\alpha}{2}} \sqrt{\hat{V}(\bar{y}_h)}, \bar{y}_{str} + Z_{\frac{\alpha}{2}} \sqrt{\hat{V}(\bar{y}_h)})$

```
tibble(
  Mean = svymean(~property_value, strat_design, na.rm = TRUE)[1],
  Lower = confint(svymean(~property_value, strat_design, na.rm = TRUE))[1],
  Upper = confint(svymean(~property_value, strat_design, na.rm = TRUE))[2]
) %>%
  kable(caption = "Mean appraised property value and 95% confidence interval",
        digits = 0)
```

Table 21: Mean appraised property value and 95% confidence interval

Mean	Lower	Upper
478338	421675	535001

$\bar{y}_h$

We were interested in comparing mean appraised property values across the District with each other. To this end, we treated each strata as its own simple random sample and calculated estimates and 95% confidence intervals for the parameters in each strata.

$\bar{y}_h = \sum_{i=1}^{n_h}$

$s^2 = \frac{\sum_{i=1}^{n_h} (y_i - \bar{y}_h)^2}{n_h - 1}$ where i is the ith observation in the strata.

$\hat{V}(\bar{y}_h) = (\frac{N_h - n_h}{N_h})^2 \frac{s^2}{n}$

```
svyby(~property_value,            # variable to estimate
      ~quadrant,                  # subgroup variable
```

```
      design = strat_design,
      FUN = svymean,              # function to use on each subgroup
      keep.names = FALSE,         # does not include row.names
      na.rm = TRUE,
      vartype = "ci"
      ) %>%
  kable(digits = 0)
```

quadrant	property_value	ci_l	ci_u
NE	404074	346970	461179
NW	544274	454045	634504
SE	372299	275534	469064
SW	345243	289347	401139

$\hat{p}_{str}$

The third main parameter estimated was the stratifed proportion of addresses with multi-family zoning in Washington DC. We also estimated a 95% confidence interval for the parameter.

$\hat{p}_{str} = \sum_{h=1}^{H} \frac{N_h}{N} \hat{p}_h$

$\hat{V}(\hat{p}_{str}) = \sum_{h=1}^{H} (\frac{N_h}{N})^2 \frac{N_h - n_h}{N_h} \frac{\hat{p}_h(1-\hat{p}_h}{n_h - 1}$

$(\hat{p}_{str} - Z_{\frac{\alpha}{2}} \sqrt{\hat{V}(\hat{p}_{str})}, \hat{p}_{str} + Z_{\frac{\alpha}{2}} \sqrt{\hat{V}(\hat{p}_{str})})$

```
tibble(
  Mean = svymean(~multifamily, strat_design)[1],
  Lower = confint(svymean(~multifamily, strat_design))[1],
  Upper = confint(svymean(~multifamily, strat_design))[2]
) %>%
  kable(caption = "",
        digits = 3)
```

Mean	Lower	Upper
0.417	0.364	0.471

$\hat{p}_h$

Finally, we were interested in comparing the proportion of addresses with multi-family zoning in quadrants across the District with each other. To this end, we treated each strata as its own simple random sample and calculated estimates and 95% confidence intervals for the

parameters in each strata.

$\hat{p}_h = \frac{\sum_{i=1}^{n_h} y_i}{n_h}$ where y_i is the indicator variable

$$y_i = \begin{cases} 1 & \text{if the ith unit has the attribute} \\ 0 & \text{if the ith unit does not have the attribute} \end{cases}$$

$\hat{V}(\hat{p}_h) = (\frac{N_h - n_h}{N_h})\frac{\hat{p}_h(1-\hat{p}_h)}{n_h - 1}$

$(\hat{p}_h - Z_{\frac{\alpha}{2}}\sqrt{\hat{V}(\hat{p}_h)}, \hat{p}_h + Z_{\frac{\alpha}{2}}\sqrt{\hat{V}(\hat{p}_h)})$

```
svyby(~multifamily,              # variable to estimate
      ~quadrant,                    # subgroup variable
      design = strat_design,
      FUN = svymean,                # function to use on each subgroup
      keep.names = FALSE,           # does not include row.names
      na.rm = TRUE,
      vartype = "ci"
      ) %>%
  kable(digits = 3)
```

quadrant	multifamily	ci_l	ci_u
NE	0.294	0.185	0.403
NW	0.563	0.479	0.647
SE	0.103	0.030	0.176
SW	0.632	0.517	0.748

10. Discussion

data quality mattered a lot!

Appendix A

Bibliography

Forhlich, Thomas C. 2019. "What It Actually Costs to Live in America's Most Expensive Cities." USA TODAY. https://www.usatoday.com/story/money/2019/04/04/what-it-actually-costs-to-live-in-americas-most-expensive-cities/37748097/.

Goetz, Lisa. 2019. "Top 10 Most Expensive Cities in the U.s." Investopedia. https://www.investopedia.com/articles/personal-finance/080916/top-10-most-expensive-cities-us.asp.

Kiersz, Andy. 2019. "The 15 Most Expensive Cities in America." Business Insiders. https://www.businessinsider.com/most-expensive-cities-in-america-2019-5#10-washington-arlington-alexandria-dc-va-md-wv-had-a-price-level-184-higher-than-the-national-average-