Zoning out: an analysis of zoning and property values in Washington, DC

Brian Bontempo, Derrick Lee, and Aaron R. Williams

Introduction

1. Need Statement

Washington, DC 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 the high cost of 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, DC's restrictive zoning, height limitation, and excessive historic 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, DC and in its four quadrants. As subsequent sections will demonstrate, there are rich sources of information about property values and zoning rules in Washington, DC, but it is difficult to extract information about population parameters from those sources. A probability sample of residences was taken to estimate the value of residential properties and the proportion of properties with zoning that does not restrict multi-family housing.

2. Target Population

Our target population is the set of all official residential housing units in the city of Washington, DC, or at least all residential housing units included in the Master Address Repository. This population explicitly excludes any government or commercial units such as offices, museums, or public spaces. The target population also includes both occupied and unoccupied residential units, and individual units contained in multi-family residential buildings.

3. Sampling Frame

- I. Address Residential Units
  - https://opendata.dc.gov/datasets/c3c0ae91dca54c5d9ce56962fa0dd645_68
- II. Address Points
  - http://opendata.dc.gov/datasets/address-points

To obtain a list of all residential housing units in Washington, DC, 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 DC. 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 to avoid double counting. We were able to generate a final list of all residential housing units in Washington, DC, 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 proved valuable for stratifcation.

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

set_urbn_defaults(style = "print")
```

```r
# 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")
}
```

```r
# Adress 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))
```

```r
# ARU contains about 7,000 residential units with status set to "RETIRED".
# We drop these cases.
aru <- aru %>%
  filter(status != "RETIRE")
```

```r
# 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)
```

```r
# AP 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.
ap <- ap %>%
  filter(res_type != "NON RESIDENTIAL")
```

```r
# Address points contains residential units with status set to "RETIRED".
# We drop these cases as well.
ap <- ap %>%
  filter(status != "RETIRE")
```

```r
# 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% of observations in Address Residential Units and 0.07% of
# observations in Address Points.
ap <- ap %>%
  filter(!is.na(fulladdress))

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

```r
# Address Points has interesting variables not present in ARU.
# 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.
aru_expanded <- aru %>%
  select(-status) %>%
  left_join(ap, by = c("fulladdress", "address_id")) %>%
  select(quadrant = quadrant.x, everything(), -quadrant.y)

rm(aru)
```

```r
### Combination

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

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

```r
# Finally, we can combine the two datasets to create a sampling frame that
# contains approximately every residential address in Washington, DC

sampling_frame <- bind_rows(ap, aru_expanded)

rm(ap, aru_expanded)

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

4. Primary Parameters

Our primary parameters of interest were the average appraised property value of the DC residential units in our sampling frame and the proportion of units in areas that don't restrict multi-family housing like apartments and condos. Our goal was not only to produce reliable estimates for the entire target population (for which we used a stratified random sample), but also to obtain reliable estimates for each of our strata – i.e., for the NW, NE, SW, and SE quadrants of DC.

- Mean appraised property value in Washington, DC
- Mean appraised property value in each quadrant of Washington, DC
- Proportion of homes in Washington, DC in zones that allow for multi-family housing
- Proportion of homes in each quadrant of Washington, DC in zones that allow for multi-family housing

5. Questionnaire/Instrument

Although we visited several of the sample units in person, not much relevant information could have been gained from visiting the addresses. In particular, we did not have sufficient

knowledge or information to accurately appraise a unit's property value, nor could we have identified a unit's zoning type. Nonetheless, even though we had to use online sources to obtain our data, we had to manually enter each address into a given website; in other words, the data for our parameters of interest were not already neatly organized, and it would not have been feasible for us to obtain measurements for all the units in our sampling frame (thus necessitating a sample survey).

Our primary instruments for measuring and capturing data were the government housing database web portals and Redfin:

- I. DC Zone Map
  - http://maps.dcoz.dc.gov/zr16/
- II. Redfin
  - https://www.redfin.com
- III. DC Tax Service Web Portal
  - https://www.taxpayerservicecenter.com/RP_Search.jsp?search_type=Assessment

Zone district types were captured with the official DC Zoning Map (I). Once the street number, street name, and quadrant are entered, the website generates the address's various zoning data, including its Zone district type. Once the type was identified, we recorded the following data points for each residential housing unit in our sample: (1) Street Number, (2) Street Name, (3) Quadrant, (4) Unit Number, (5) Unit Type, (6) Zone Type. Example zone district types are RA-4, which permits medium to high-desnity apartments, and RF-1, which permits development of attached rowhouses on small lots.

Additionally, we captured the appraised values (Land and Improvement values) for each unit with Redfin (II) and the DC Tax Service web portal (III). First, with Redfin, we entered the street number, street name, quadrant, and city of each unit to return various property details, from which we obtained the Land and Addition (Improvement) dollar amounts of the unit. If we were not able to acquire the data points through Redfin, we then utilized the DC Tax Service web portal with the same data entry to obtain the Land and Improvement values of the properties. For condominiums and single-family units, we obtained unit-specific appraised values, while for multi-family apartments, we obtained their total building appraised values and adjusted for the number of residents in the building. These two additional data points were then appended to the above six data points for the selected sample units: (7) Land Value, (8) Improvement Value.

6. Pilot Survey

In order to determine the sample allocation for our stratified random sample, we first needed to obtain an estimate for the variance of the appraised property values in our target population. By Theorem 9.1 (Wright 165), the sample variance s^2 is an unbiased estimator of the population variance S^2 under simple random sampling. Therefore, for our pilot survey, we took a simple random sample of size 25 from each of the four DC quadrants and combined these samples to obtain a stratified random sample of size 100 (though computers technically use deterministic methods, we used R to approximate the selection of this stratified random sample).

By Problem 9.A.3 (Wright 169), when a variable of interest can take on only the values 0 and 1, the sample variance can be rewritten as $s^2 = \frac{n}{n-1}\hat{p}(1-\hat{p})$. Since n is arbitrary, then this result also holds for N. In other words, when we are trying to estimate a proportion, the population variance can be written as $S^2 = \frac{N}{N-1}p(1-p)$. Since $p(1-p)$ attains its maximum value at p = 0.5, we decided to be conservative by simply "assuming" that the proportion of housing units in our population in zones that don't restrict multi-family equals 0.5. Thus, we did not need to calculate the pilot sample variance for our zoning type parameter.

For the appraised property value, on the other hand, we attempted to enter each of the addresses into the DC Tax Service web portal and Redfin. Unfortunately, there were missing values for several of the housing units. In an attempt to rectify this problem, we recognized that since we took a simple random sample from each quadrant, then, within each quadrant, Theorem 9.2 (Wright 169) implies that the sample proportion of housing units with missing values is an unbiased estimator of the true proportion of housing units with missing values. Since it would not have been feasible to determine exactly how many housing units in our population have missing values, we therefore calculated the sample proportion of missing values for each quadrant and computed "adjusted" N_h values for each of our four strata. Consequently, we ended up with an "adjusted" N value, which was our estimated number of housing units that did not have missing values. These were the N_h and N values that we used in our sample selection and allocation algorithms.

```
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",
        col.names = c("$\\bar{y}$", "$s^2$", "Proportion missing"))
```

Table 1: Pilot survey summary statistics

$\bar{y}$	s^2	Proportion missing
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",
        col.names = c("Quadrant", "$\\bar{y}$", "$s^2$", "Proportion missing"))
```

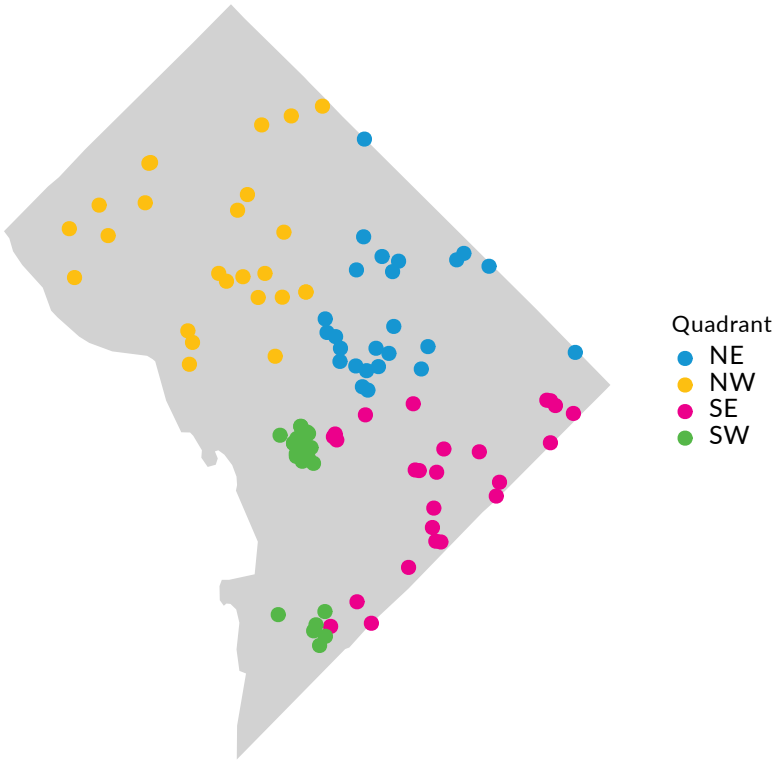
Table 2: Pilot survey summary statistics by quadrant

Quadrant	$\bar{y}$	s^2	Proportion missing
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 were interested in estimating the sample mean of property values for all DC residences using stratification, the sample means of property values for all residences in each quadrant of DC, the sample proportion of residences in zones that don't restrict multi-family housing for all DC residences using stratifcation, and the sample proportions of residences in zones that don't restrcit multi-family housing. All four of these estimation processes required allocating a certain number of sample units to each stratum and each process has a different optimal allocation.

For the stratified estimates we specified a maximum variance, V_0, and used Exact Optimal

Allocation for $\bar{y}_{str}$ and $\hat{p_{str}}$. For the estimates within each quadrant, we specified an error, e, and calculated an appropriate sample size for simple random sampling and a 90 percent confidence interval. Finally, we aligned all four optimal allocations and took the maximum n_h for each quadrant. This didn't necessarily result in an optimal allocation for any one of our four estimates but it ensured that our allocation was adequate for each of our four estimates.

Condition 1: Sample mean

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

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

$V(\bar{y}_h) = \frac{N_h - n_h}{N_h} \frac{S_h^2}{n_h}$ (By Wright Theorem 9.1, since we are taking a simple random sample from each stratum h)

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

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

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

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

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

$V(\bar{y}_{str}) = \sum_{h=1}^{H} V(\bar{y}_{str})$ (by the independence of $\bar{y}_1, ..., \bar{y}$)

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

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

...

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

...

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

For a desired bound V_0 on the sampling variance $V(\bar{y}_{str})$, one can 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, so 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 $V(\bar{y}_{str}|n_{13}, ..., N_H) \leq V_0$, then stop with $n_1 = n_{13}, ..., n_H = n_{H3}$. Otherwise, continue until step j, where $V(\bar{y}_{str}|n_{1j}, ..., n_{Hj}) \leq V_0$. The final allocation is $n_{1j}, ..., n_{Hj})$ and $n = n_{1j} + \cdots + n_{Hj}$.

First we estimate s^2 and the missing proportion for each stratum.

```
# 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,
      col.names = c("Stratum", "$s^2$", "Missing Prop.", "$N_h$", "N"))
```

Stratum	s^2	Missing Prop.	N_h	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

Next we estimated the variance with an initial allocation of (1, 1, 1, 1) in each stratum.

```
# 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,
      col.names = c("Stratum", "$s^2$", "Missing Prop.", "$N_h$", "N", "Initial variance"))
```

Stratum	s^2	Missing Prop.	N_h	N	Initial 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()
```

This resulted in a total starting variance of 230719468377.896. Next we calculated priority values for each potential sample unit up to the entire sampling frame and ordered those priority values from largest to smallest. We then sequentially subtracted each additional prioirty value from the starting variance to get the variance of the estimator with the given allocation.

```
# 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))
```

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

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

stratum	n	s_squared_h	Nh	priority_value	agg_priority_value	variance
NW	2	715270634804	166932	112864355500	112864355500	117855112878
NW	3	715270634804	166932	37621451833	150485807333	80233661045
NW	4	715270634804	166932	18810725917	169296533250	61422935128
NW	5	715270634804	166932	11286435550	180582968800	50136499578
NW	6	715270634804	166932	7524290367	188107259166	42612209212
NW	7	715270634804	166932	5374493119	193481752285	37237716093
NW	8	715270634804	166932	4030869839	197512622125	33206846253
NW	9	715270634804	166932	3135120986	200647743111	30071725267
NW	10	715270634804	166932	2508096789	203155839900	27563628478
NW	11	715270634804	166932	2052079191	205207919091	25511549287

```
rm(n_strata)
```

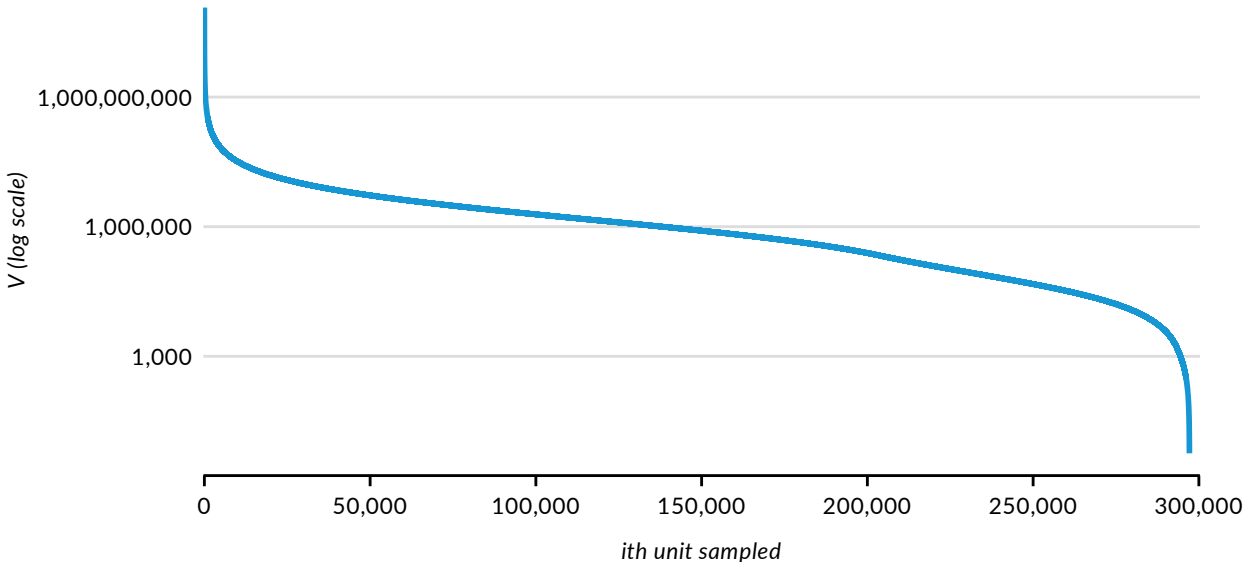
We chose a V_0 of 0.1 times the mean from the pilot survey squared and then selected the largest priority values that achieved V_0 the quickest.

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

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

Variance of $\bar{y}_{str}$ with Exact Optimal Allocation at the ith unit

```
priority_values %>%
  mutate(x = row_number()) %>%
  ggplot(aes(x = x, y = variance)) +
  geom_line() +
  scale_x_continuous(expand = expand_scale(mult = c(0.001, 0.001)),
                     limits = c(0, 300000),
                     labels = scales::comma,
                     breaks = seq(0, 300000, 50000)) +
  scale_y_log10(labels = scales::comma) +
  labs(x = "ith unit sampled",
```

```
      y = "V (log scale)") +
  theme(plot.margin = margin(r = 20))
```



Condition 2: Sample means within strata

We were 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 used 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 wanted \$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 began with a derivation of Exact Optimal Sample Allocation for $\hat{p}$.

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

By noting that $\hat{p}_h$ is a special case of $\bar{y}_h$, and that when $y_i \in \{0,1\}$ for all i, we can rewrite S_h^2 as $S_h^2 = \frac{N_h}{N_h-1}p_h(1-p_h)$, then it follows from our decomposition of $V(\bar{y}_{str})$ that

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

$V(\hat{p}_{str}) = \sum_{h=1}^{H}\frac{N_h^2}{N^2}(\frac{(N_h h-1)S_h^2}{N_h} - \frac{S_h^2}{1\cdot 2} - \frac{S_h^2}{2\cdot 3} - ... - \frac{S_h^2}{n_h(n_h-1)})$

$V(\hat{p}_{str}) = \sum_{h=1}^{H}\frac{N_h^2}{N^2}(\frac{(N_h-1)}{N_h}\frac{N_h p_h(1-p_h)}{(N_h-1)} - \frac{N_h p_h(1-p_h)}{(N_h-1)(1\cdot 2)} - \frac{N_h p_h(1-p_h)}{(N_h-1)(2\cdot 3)} - ... - \frac{N_h p_h(1-p_h)}{(N_h-1)(n_h)(n_h-1)})$

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

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

$-\frac{N_1^3 p_1(1-p_1)}{N^2(N_1-1)\cdot 1\cdot 2} - \frac{N_1^3 p_1(1-p_1)}{N^2(N_1-1)\cdot 2\cdot 3} - ... - \frac{N_1^3 p_1(1-p_1)}{N^2(N_1-1)n_h(n_h-1)}$

...

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

...

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

For a desired bound on V_0 on the sampling variance $V(\hat{p}_{str})$, one 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.

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

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, so 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 $V(\hat{p}_{str}|n_{13}, ..., N_H \leq V_0$, then stop with $n_1 = n_{13}, ..., n_H = n_{H3}$. 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,
      col.names = c("Stratum", "$N_h$", "N", "$s^2_h"))
```

Stratum	N_h	N	$s^2_h
NE	74949	348094	0.25
NW	189695	348094	0.25
SE	68644	348094	0.25
SW	14806	348094	0.25

Next we estimated the variance with an initial allocation of (1, 1, 1, 1) in each stratum.

```
# 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,
      col.names = c("Stratum", "$N_h$", "N", "$s^2_h", "Initial variance"))
```

Stratum	N_h	N	$s^2_h	Initial 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()
```

This resulted in a total starting variance of 0.0960076. Next we calculated priority values for each potential sample unit up to the entire sampling frame and ordered those priority values from largest to smallest. We then sequentially subtracted each additional prioirty value from the starting variance to get the variance of the estimator conditional on the given allocation.

```
# 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))
```

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

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

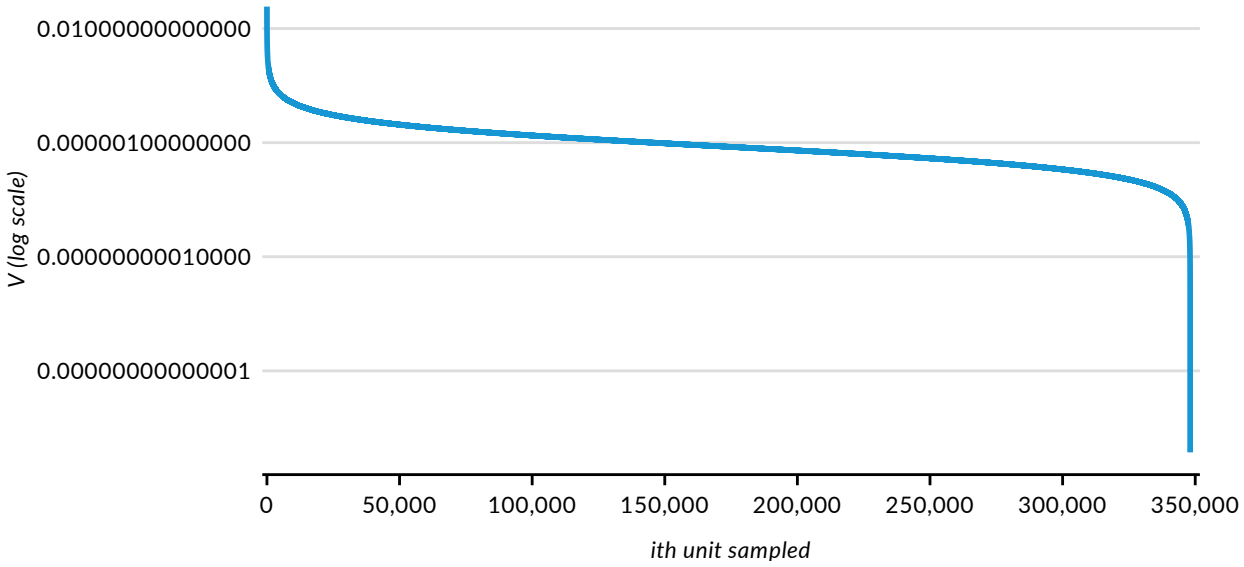
stratum	n	Nh	s_squared_h	priority_value	agg_priority_value	variance
NW	2	189695	0.25	0.0371220	0.0371220	0.0588857
NW	3	189695	0.25	0.0123740	0.0494960	0.0465117
NW	4	189695	0.25	0.0061870	0.0556830	0.0403247
NE	2	74949	0.25	0.0057950	0.0614780	0.0345297
SE	2	68644	0.25	0.0048610	0.0663390	0.0296686
NW	5	189695	0.25	0.0037122	0.0700512	0.0259564
NW	6	189695	0.25	0.0024748	0.0725260	0.0234816
NE	3	74949	0.25	0.0019317	0.0744577	0.0215500
NW	7	189695	0.25	0.0017677	0.0762254	0.0197823
SE	3	68644	0.25	0.0016203	0.0778457	0.0181619

```
rm(n_strata)
```

We chose a V_0 of 0.1 times the mean from the pilot survey squared and then selected the largest priority values that achieved V_0 the quickest.

Variance of $\hat{p}_{str}$ with Exact Optimal Allocation at the ith unit

```
priority_values %>%
  mutate(x = row_number()) %>%
  ggplot(aes(x = x, y = variance)) +
  geom_line() +
  scale_x_continuous(expand = expand_scale(mult = c(0.005, 0.005)),
                     limits = c(0, 350000),
                     labels = scales::comma,
                     breaks = seq(0, 350000, 50000)) +
  scale_y_log10() +
  labs(x = "ith unit sampled",
       y = "V (log scale)") +
  theme(plot.margin = margin(r = 20))
```




```
condition3 <- priority_values %>%
  filter(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",
        digits = 2)
```

Table 13: Recommended strata sizes across the four conditions

quadrant	1.	2.	3.	4.
NE	14	10.38	21	67.58
NW	132	134.28	53	67.61
SE	11	13.43	19	67.57
SW	1	4.70	3	67.34

```r
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 14: Maximum recommended strata sizes across the four conditions

quadrant	nh
NE	68
NW	135
SE	68
SW	68

8. Sampling Plan

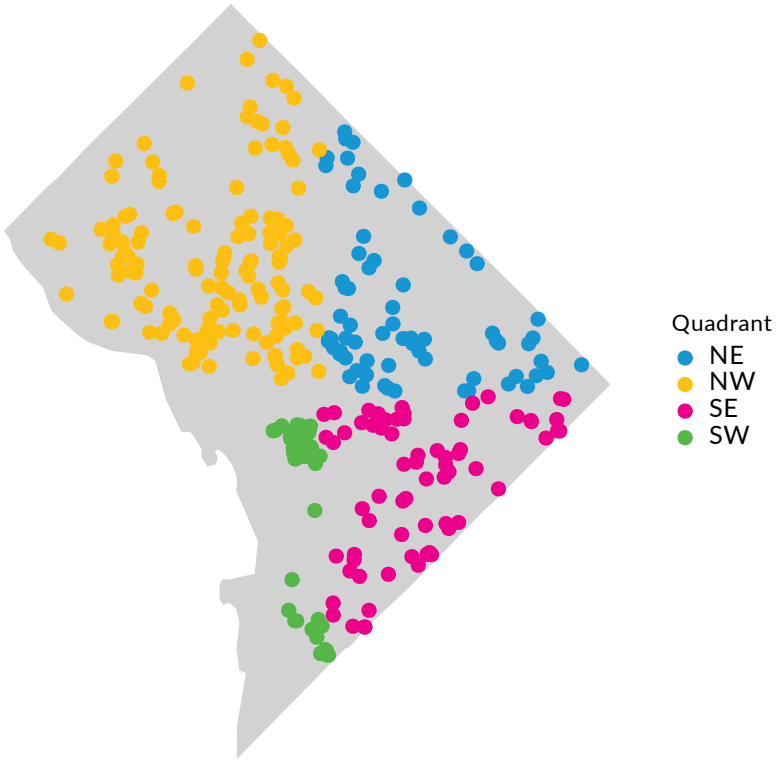
```r
survey <- group_split(sampling_frame, quadrant) %>%
  map2_df(nh$nh, sample_n)

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("data/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 16: 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

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

10. Discussion

Survey conclusions

We estimated that homes are very valuable in Washington, DC. We estimated a home price of $478,338 with a 95% confidence interval of ($421,675, $535,001). That said, there is meaningful heterogeneity across quadrants with homes in Northwest DC being much more valuable than homes in the other three quadrants.

We estimated fewer than half of homes in Washington, DC are in areas with zoning that does not restrict multi-family housing. We estimate that 0.417 of homes are in areas with zoning that does not restrict multi-family housing with a 95% confidence interval of (0.364, 0.471). This estimate is particularly dramatic because our sampling frame is based on addresses and not square area. In other words, the probability of selecting a home in an area with dense housing is greater than selecting a home in an area with sparse housing. It is likely that less than 0.417 of the residential land area in the District does not restrict multi-family housing.

Like appraised property values, there is significant heterogeneity in the proportion of homes in different DC quadrants that are in zones that don't restrict multi-family housing. The pattern, with NW being less restrictive than SE and NE, runs counter to the popular story that housing is most restricted in Northwest DC. This suggests that other limitations could be at play such as historical preservation, obstinate neighborhood associations, or restrictive permitting. The high share of homes in areas that allow for multi-family housing in SW reflects recent developments in Waterfront and The Wharf and also the high share of the quadrant that doesn't have housing at all because of L'Enfant, Interstate 395, and the military base.

Process conclusions

> "You go to data analysis with the data you have, not the data you might want or wish to have at a later time." ~ Anonymous

Data quality created challenges for estimations of mean appraised property values. Some units in the sampling frame, pilot survey, and final survey were missing values from both the DC Tax Service Web Portal and Redfin. We did our best to accommodate these missing values, but the uncertainty of our estimates almost certainly exceeded our calculated standard errors. In particular, many of the missing property values appeared to

come from public housing of different sorts. This means the true parameter for mean housing value in Washington, DC is likely slightly lower than our estimate.

We did not face the same challenges with zoning type, where measurement error was low to non-existent. We were able to fill in zoning type for all sampled units and classification into zones that allow for multi-family housing and zones that restrict multi-family housing was straightforward.

Two key takeaways from this project: 1. Applications of probability sampling with low measurement error are more enjoyable than application with high measurement error or survey nonresponse. 2. Measurement error and survey nonresponse are pressing issues in applications of probability sampling and warrant further exploration by the authors of this report.

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-