Baltimore Nonprofit Analysis

Table of Contents

library(readxl)  
library(here)  
library(tidyverse)  
library(stringr)  
library(naniar)

# Data Import

df\_simplified<-read\_excel(here::here("Nonprofit\_Baltimore\_Analysis.xlsx"), sheet = 3)

## Warning: Expecting numeric in W4084 / R4084C23: got 'of art'

Information about the data: Here’s an [article](https://nccs.urban.org/nccs/datasets/bmf/) describing some of the datasets from the IRS 990s. We’re using the Business Master File (BMF). There’s a section titled “Minimum Filing Threshold” that explains a data limitation (and why we’re seeing so many 0’s). See [here](https://www.irs.gov/charities-non-profits/exempt-organizations-annual-reporting-requirements-overview-annual-return-filing-exceptions) for more info.

It seems that if there is a value less than 50,000 other than zero, it must mean that the organization decided to submit to the IRS, because otherwise they would be listed as a zero. It is not possible to distinguish a true zero from a zero due to not meeting the threshold of 50,000 and just not submitting. See [this guide](https://github.com/jhudsl/Baltimore_nonprofits_assets/blob/8094191b88bcef6503c132c7cc73408c12ddc916/Guide_to_Using_NCCS_Data_202.pdf), page 5 in the “minimum filing threshold” section.

It therefore makes sense to remove zero values and to report this caveat that the data is incomplete because many nonprofits that had assets less than 50,000 are not included.

However for the high vs nonhigh asset we could keep these - because zero values would still be less than the threshold regardless.

Adding to this NA values can be considered less than 50000, as organizations are not required to report an amount if they have less than 50000.

# Tidying data and Exploratory Analysis

## Asset amount

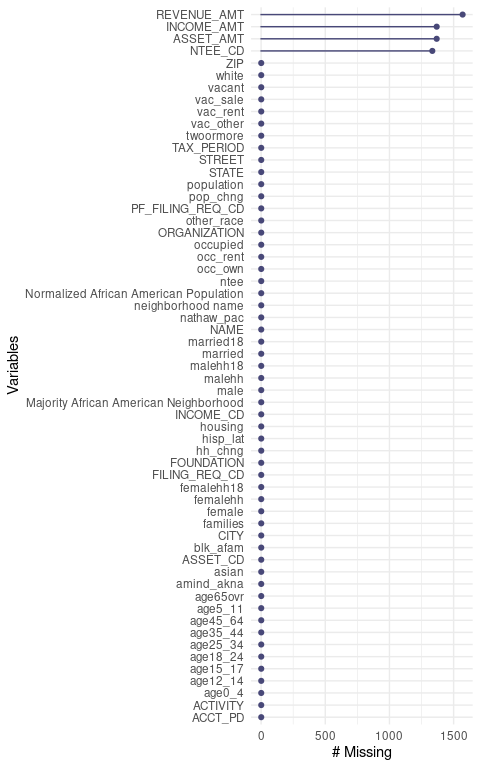
First let’s check how many zero values there are for asset amounts.

df\_simplified %>% filter(ASSET\_AMT==0)%>% nrow()

## [1] 1218

Now we will check if there are NA values for asset amounts.

gg\_miss\_var(df\_simplified)



Yes, indeed there are…

NA and zero values likely mean the nonprofit did not need to submit to the IRS. It is impossible to know however, if a zero is actually a true zero. NA values could mean something else.

Thus, we will recode asset amount based on a threshold of greater than or equal to 500,000 as high asset and less than 500,000 (including zero) as not high asset. Note we keep our NA values with this recoding.

df\_simplified<-df\_simplified %>%  
 # modify Asset amount variable to be numeric  
 mutate(ASSET\_AMT = as.numeric(ASSET\_AMT)) %>%  
 #create a variable about high asset amount (threshold being $500,000)  
 mutate(ASSET\_High = case\_when(ASSET\_AMT >= 500000 ~ TRUE,  
 ASSET\_AMT < 500000 ~ FALSE))

Here we can see the NA values:

table(df\_simplified$ASSET\_High, useNA = "always")

##   
## FALSE TRUE <NA>   
## 1954 761 1368

Now we will replace NA values with False as well:

df\_simplified <- df\_simplified %>%   
 mutate(ASSET\_High = replace\_na(ASSET\_High, FALSE)) #NA values will be coded as not high asset (if no cases match the other two ASSET\_AMT statements then code as FALSE) based on the above logic

Check that this worked and that there are no NA values now:

table(df\_simplified$ASSET\_High, useNA = "always")

##   
## FALSE TRUE <NA>   
## 3322 761 0

Now we will convert these to text as another variable and also create a log version of the asset amount to normalize it, as plots and analysis may be easier to interpret if there are many values that are high or low.

df\_simplified<-df\_simplified %>%  
 mutate(ASSET\_High\_text = case\_when(ASSET\_High == TRUE ~ "High Asset",  
 ASSET\_High == FALSE ~ "Low Asset")) %>%  
 # we will also create new log of asset amount variable  
 mutate(ASSET\_AMT\_log = log(ASSET\_AMT))

## Neighborhood category

Next we will modify the data to include a variable about the percentage of African American/Black people.

# create new Percent\_AA variable by converting  
#`Normalized African American Population` variable   
#into a percentage and rounding  
 df\_simplified<-df\_simplified %>%  
 mutate(Percent\_AA =   
 round(`Normalized African American Population`\*100, digits = 1)) %>%  
  
 # create new Majority\_AA variable that indicates if Percent\_AA is greater than or equal to 50% or not  
 mutate(Majority\_AA = case\_when(  
 Percent\_AA >= 50 ~ "Yes",   
 Percent\_AA < 50 ~ "No")) %>%  
 # create a new variable about this in text  
 mutate(Neighborhood = case\_when(  
 Percent\_AA >= 50 ~ "Majority\nBlack",   
 Percent\_AA < 50 ~ "Majority\nNon-Black")) %>%   
 # make this a factor and order by level appearance in the data  
 mutate(Neighborhood = as\_factor(Neighborhood),  
 Neighborhood = forcats::fct\_inorder(Neighborhood))

## Quantile data

To group the data by quantiles, we first remove organizations with zero assets because we don’t know if zero values are real. If assets are under 50,000, organizations can report as zero also.

Similarly, NA values could be anything between 0 and less than 50,000. Thus we aren’t sure what those asset amounts are.

# make a new dataframe without zeros and NA asset amounts  
df\_simplified\_no\_zero<-df\_simplified %>%  
 drop\_na(ASSET\_AMT) %>% #redundant but shows we are dropping NA values  
 filter(ASSET\_AMT>0) %>% # ASSET\_AMT must be greater than zero  
 # get quartiles  
 mutate(ASSET\_quartile = ntile(ASSET\_AMT, 4)) %>%  
 # create new quartile variable that specifies quartiles by text  
 mutate(ASSET\_quartile\_text = case\_when(ASSET\_quartile == 1 ~ "1st\_Quartile",  
 ASSET\_quartile == 2 ~ "2nd\_Quartile",  
 ASSET\_quartile == 3 ~ "3rd\_Quartile",  
 ASSET\_quartile == 4 ~ "4th\_Quartile"))  
  
#Now check:  
table(df\_simplified\_no\_zero$ASSET\_quartile, useNA = "always")

##   
## 1 2 3 4 <NA>   
## 375 374 374 374 0

df\_simplified\_no\_zero%>% group\_by(ASSET\_quartile\_text) %>% summarise(range = range(ASSET\_AMT))

## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in  
## dplyr 1.1.0.  
## ℹ Please use `reframe()` instead.  
## ℹ When switching from `summarise()` to `reframe()`, remember that `reframe()`  
## always returns an ungrouped data frame and adjust accordingly.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## `summarise()` has grouped output by 'ASSET\_quartile\_text'. You can override  
## using the `.groups` argument.

## # A tibble: 8 × 2  
## # Groups: ASSET\_quartile\_text [4]  
## ASSET\_quartile\_text range  
## <chr> <dbl>  
## 1 1st\_Quartile 1  
## 2 1st\_Quartile 77638  
## 3 2nd\_Quartile 77996  
## 4 2nd\_Quartile 528175  
## 5 3rd\_Quartile 529732  
## 6 3rd\_Quartile 3783266  
## 7 4th\_Quartile 3804811  
## 8 4th\_Quartile 3267270835

Without removal (except NAs)- since these asset amounts might influence quartiles.

df\_simplified\_no\_removal<-df\_simplified %>%  
 drop\_na(ASSET\_AMT) %>% #dropping NA values  
 # get quartiles  
 mutate(ASSET\_quartile = ntile(ASSET\_AMT, 4)) %>%  
 # create new quartile variable that specifies quartiles by text  
 mutate(ASSET\_quartile\_text = case\_when(ASSET\_quartile == 1 ~ "1st\_Quartile",  
 ASSET\_quartile == 2 ~ "2nd\_Quartile",  
 ASSET\_quartile == 3 ~ "3rd\_Quartile",  
 ASSET\_quartile == 4 ~ "4th\_Quartile"))  
  
#Now check:  
table(df\_simplified\_no\_removal$ASSET\_quartile, useNA = "always")

##   
## 1 2 3 4 <NA>   
## 679 679 679 678 0

df\_simplified\_no\_removal %>% group\_by(ASSET\_quartile\_text) %>% summarise(range = range(ASSET\_AMT))

## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in  
## dplyr 1.1.0.  
## ℹ Please use `reframe()` instead.  
## ℹ When switching from `summarise()` to `reframe()`, remember that `reframe()`  
## always returns an ungrouped data frame and adjust accordingly.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## `summarise()` has grouped output by 'ASSET\_quartile\_text'. You can override  
## using the `.groups` argument.

## # A tibble: 8 × 2  
## # Groups: ASSET\_quartile\_text [4]  
## ASSET\_quartile\_text range  
## <chr> <dbl>  
## 1 1st\_Quartile 0  
## 2 1st\_Quartile 0  
## 3 2nd\_Quartile 0  
## 4 2nd\_Quartile 8141  
## 5 3rd\_Quartile 8251  
## 6 3rd\_Quartile 735297  
## 7 4th\_Quartile 738933  
## 8 4th\_Quartile 3267270835

Turns out this doesn’t really work because the 1st and 2nd quartiles can’t be distinguished as we don’t know if the values of 0 are actually some number below 50,000 and both quartlies have a max of less than 50,000. So we will stick with our removal.

## National Taxonomy of Exempt Entities (NTEE) recoding

To provide more information to readers about what the nonprofits do, we will convert the National Taxonomy of Exempt Entities (NTEE) codes based on [this](https://urbaninstitute.github.io/nccs-legacy/ntee/ntee.html) and [this (see page 15)](https://www.irs.gov/pub/irs-tege/p4838.pdf).

df\_simplified <-df\_simplified %>%  
 mutate(NTEE\_text = case\_when(  
 str\_starts(NTEE\_CD, pattern = "A")~ "Arts", # if NTEE\_CD starts with A make new variable value "Arts" etc.  
 str\_starts(NTEE\_CD, pattern = "B") ~ "Education",  
 str\_starts(NTEE\_CD, pattern = "C|D") ~ "Environment/Animals", # if NTEE\_CD value starts with C or D make new variable value "Environment/Animals"  
 str\_starts(NTEE\_CD, pattern = "E|F|G|H") ~ "Health",  
 str\_starts(NTEE\_CD, pattern = "I|J|K|L|M|N|O|P") ~ "Human Services",  
 str\_starts(NTEE\_CD, pattern = "Q") ~ "International Affairs",  
 str\_starts(NTEE\_CD, pattern = "R|S|T|U|V|W") ~ "Societal Benefit",  
 str\_starts(NTEE\_CD, pattern = "X") ~ "Religious",  
 TRUE ~ "NA")) # this line is redundant as this would happen automatically - but everything else will be NA

Also for the quantile data:

df\_simplified\_no\_zero <-df\_simplified\_no\_zero %>%  
 mutate(NTEE\_text = case\_when(  
 str\_starts(NTEE\_CD, pattern = "A")~ "Arts", # if NTEE\_CD starts with A make new variable value "Arts" etc.  
 str\_starts(NTEE\_CD, pattern = "B") ~ "Education",  
 str\_starts(NTEE\_CD, pattern = "C|D") ~ "Environment/Animals", # if NTEE\_CD value starts with C or D make new variable value "Environment/Animals"  
 str\_starts(NTEE\_CD, pattern = "E|F|G|H") ~ "Health",  
 str\_starts(NTEE\_CD, pattern = "I|J|K|L|M|N|O|P") ~ "Human Services",  
 str\_starts(NTEE\_CD, pattern = "Q") ~ "International Affairs",  
 str\_starts(NTEE\_CD, pattern = "R|S|T|U|V|W") ~ "Societal Benefit",  
 str\_starts(NTEE\_CD, pattern = "X") ~ "Religous",  
 TRUE ~ "NA")) # this line is redundant as this would happen automatically - but everything else will be NA

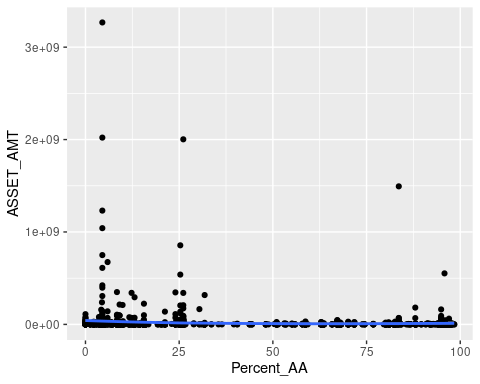
# Visualizations and Analysis

## Deeper possible visualizations

First without log normalization figure:

df\_simplified\_no\_zero %>%   
ggplot(aes(y = ASSET\_AMT, x = Percent\_AA)) +  
 geom\_point() + geom\_smooth(method = "loess")

## `geom\_smooth()` using formula = 'y ~ x'

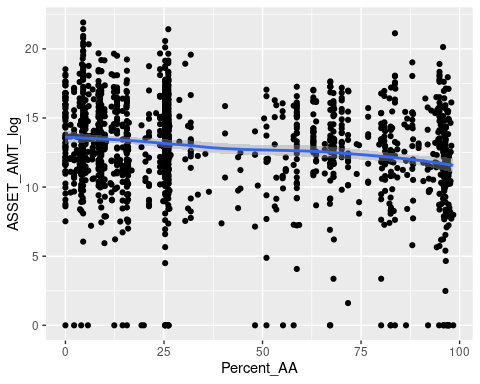
 We can see that some of the dat points are much higher and this makes it challenging to see the lower data values.

Now let’s look at normalized version.

Overall log Asset amount figure:

df\_simplified\_no\_zero %>%   
ggplot(aes(y = ASSET\_AMT\_log, x = Percent\_AA)) +  
 geom\_point() + geom\_smooth(method = "loess")

## `geom\_smooth()` using formula = 'y ~ x'

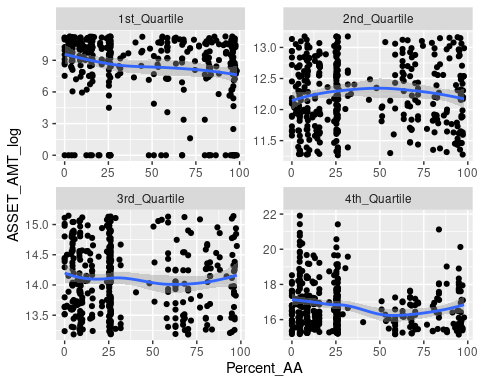


## Quartile plots

Quartiles with log asset data:

df\_simplified\_no\_zero %>%   
ggplot(aes(y = ASSET\_AMT\_log, x = Percent\_AA)) +  
 geom\_point() + facet\_wrap(~ ASSET\_quartile\_text, scales = "free") +geom\_smooth()

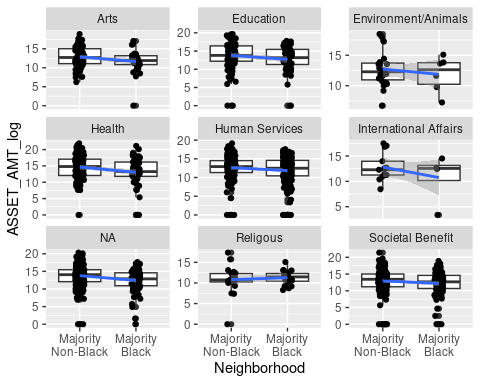
## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



Look at log asset data for each NTEE type- **remember the caveat that there are many organizations that are not included because of NA or zero value ASSET\_AMT**. However, still we can see that there is a trend towards lower amount of assets for most categories even with this limited data.

df\_simplified\_no\_zero %>%  
 ggplot(aes(y = ASSET\_AMT\_log, x = Neighborhood)) +  
 geom\_boxplot()+ geom\_jitter(width = .08) +   
 facet\_wrap(~ NTEE\_text, scales = "free\_y") +   
 geom\_smooth(method = "lm", se=TRUE, aes(group=1))

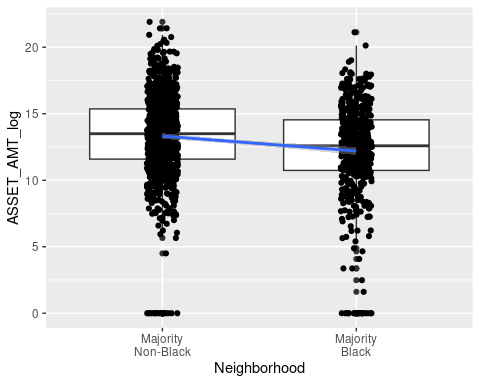
## `geom\_smooth()` using formula = 'y ~ x'



Compare all organizations by neighborhood AA status for log asset data. *remember the caveat that there are many organizations that are not included because of NA or zero value ASSET\_AMT*

df\_simplified\_no\_zero %>%  
 ggplot(aes(y = ASSET\_AMT\_log, x = Neighborhood)) +  
 geom\_boxplot()+ geom\_jitter(width = .08) + geom\_smooth(method = "lm", se=TRUE, aes(group=1))

## `geom\_smooth()` using formula = 'y ~ x'

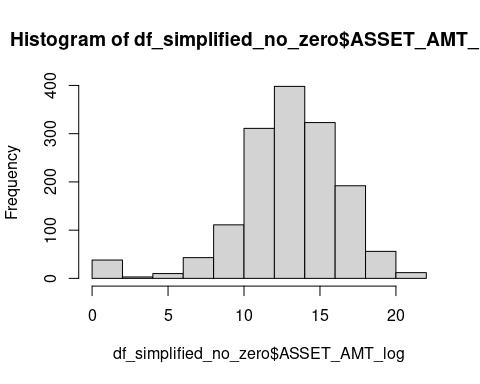


## Association Tests

summary(glm(data = df\_simplified\_no\_zero, ASSET\_AMT ~Percent\_AA)) # for every increase in percent AA of the neighborhood there was a 266,249$ decrease in asset amount of the nonprofits in the neighborhood

##   
## Call:  
## glm(formula = ASSET\_AMT ~ Percent\_AA, data = df\_simplified\_no\_zero)  
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 29664870 5349712 5.545 3.47e-08 \*\*\*  
## Percent\_AA -266249 107553 -2.476 0.0134 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 1.84704e+16)  
##   
## Null deviance: 2.7726e+19 on 1496 degrees of freedom  
## Residual deviance: 2.7613e+19 on 1495 degrees of freedom  
## AIC: 60322  
##   
## Number of Fisher Scoring iterations: 2

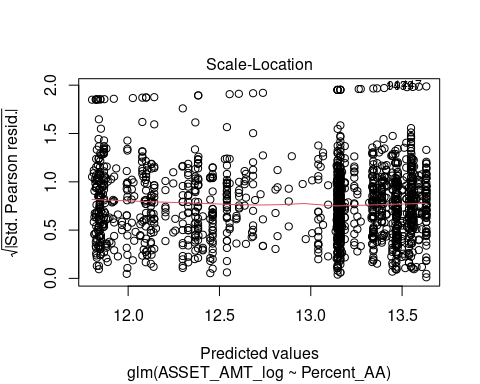
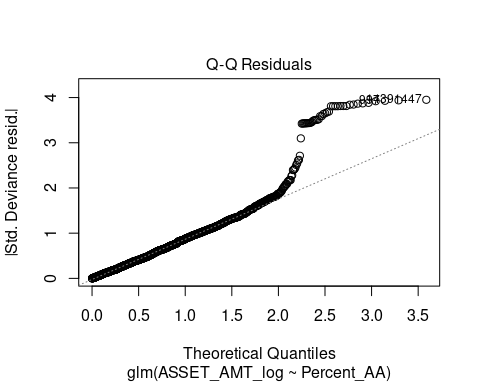
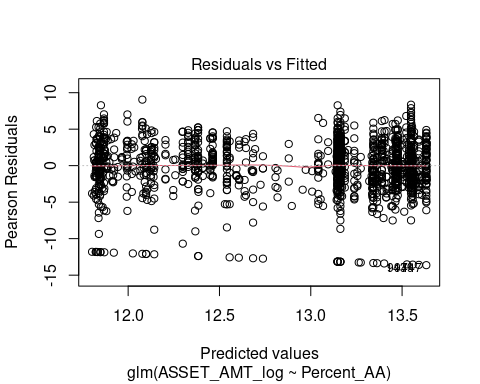
# there is a less than 5% risk of concluding that an association exists between asset amount a percent AA of neighborhood when there is no actual association.  
  
  
hist(df\_simplified\_no\_zero$ASSET\_AMT\_log)



summary(glm(data = df\_simplified\_no\_zero, ASSET\_AMT\_log~Percent\_AA)) # for every increase in percent AA of the neighborhood there was a 266,249$ decrease in asset amount of the nonprofits in the neighborhood

##   
## Call:  
## glm(formula = ASSET\_AMT\_log ~ Percent\_AA, data = df\_simplified\_no\_zero)  
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 13.632645 0.135946 100.280 < 2e-16 \*\*\*  
## Percent\_AA -0.018598 0.002733 -6.805 1.46e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 11.92755)  
##   
## Null deviance: 18384 on 1496 degrees of freedom  
## Residual deviance: 17832 on 1495 degrees of freedom  
## AIC: 7963.1  
##   
## Number of Fisher Scoring iterations: 2

glm(data = df\_simplified\_no\_zero, ASSET\_AMT\_log ~ Percent\_AA) %>% plot(which = 1:3)



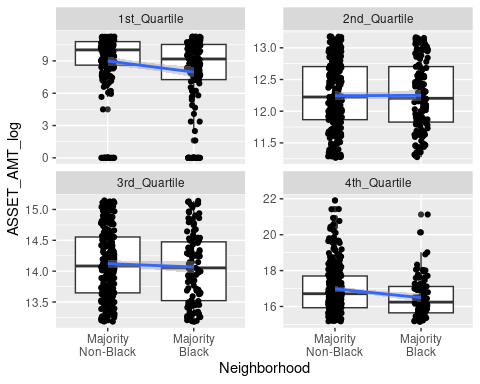
#nonparametric test - because the residiuals looked skewed in the above qqplot  
cor.test(df\_simplified\_no\_zero$ASSET\_AMT, df\_simplified\_no\_zero$Percent\_AA, method = "spearman", exact = FALSE)

##   
## Spearman's rank correlation rho  
##   
## data: df\_simplified\_no\_zero$ASSET\_AMT and df\_simplified\_no\_zero$Percent\_AA  
## S = 641433720, p-value = 1.055e-08  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## -0.1471965

Look at quartiles with log asset data: *remember the caveat that there are many organizations that are not included because of NA or zero value ASSET\_AMT*

df\_simplified\_no\_zero %>%  
 ggplot(aes(y = ASSET\_AMT\_log, x = Neighborhood)) +  
 geom\_boxplot()+ geom\_jitter(width = .08) + geom\_smooth(method = "lm", se=TRUE, aes(group=1)) + facet\_wrap(~ASSET\_quartile\_text, scales = "free\_y")

## `geom\_smooth()` using formula = 'y ~ x'



First create data to make visualization easier *caveat for the data: that there are many organizations that are not included because of NA or zero value ASSET\_AMT*

quartile\_data <-df\_simplified\_no\_zero %>%   
 group\_by(ASSET\_quartile\_text, Neighborhood) %>%   
 count()  
quartile\_data

## # A tibble: 8 × 3  
## # Groups: ASSET\_quartile\_text, Neighborhood [8]  
## ASSET\_quartile\_text Neighborhood n  
## <chr> <fct> <int>  
## 1 1st\_Quartile "Majority\nNon-Black" 217  
## 2 1st\_Quartile "Majority\nBlack" 158  
## 3 2nd\_Quartile "Majority\nNon-Black" 235  
## 4 2nd\_Quartile "Majority\nBlack" 139  
## 5 3rd\_Quartile "Majority\nNon-Black" 263  
## 6 3rd\_Quartile "Majority\nBlack" 111  
## 7 4th\_Quartile "Majority\nNon-Black" 268  
## 8 4th\_Quartile "Majority\nBlack" 106

Create percentage variable for each quantile

quartile\_data <- quartile\_data %>%  
 group\_by(ASSET\_quartile\_text) %>%   
 mutate(Percent = round(n/sum(n)\*100))   
quartile\_data

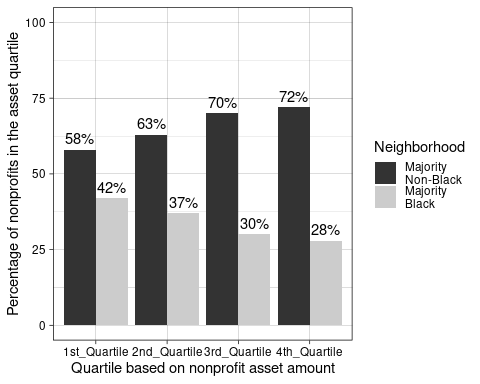
## # A tibble: 8 × 4  
## # Groups: ASSET\_quartile\_text [4]  
## ASSET\_quartile\_text Neighborhood n Percent  
## <chr> <fct> <int> <dbl>  
## 1 1st\_Quartile "Majority\nNon-Black" 217 58  
## 2 1st\_Quartile "Majority\nBlack" 158 42  
## 3 2nd\_Quartile "Majority\nNon-Black" 235 63  
## 4 2nd\_Quartile "Majority\nBlack" 139 37  
## 5 3rd\_Quartile "Majority\nNon-Black" 263 70  
## 6 3rd\_Quartile "Majority\nBlack" 111 30  
## 7 4th\_Quartile "Majority\nNon-Black" 268 72  
## 8 4th\_Quartile "Majority\nBlack" 106 28

Visuals…of the above data:

quart\_plot <- quartile\_data %>%   
 ggplot(aes(x= ASSET\_quartile\_text, y = Percent, fill = Neighborhood)) +   
 geom\_col(position = position\_dodge(width = .9))+  
 scale\_y\_continuous(labels = function(x) paste0(x, "%")) +  
 ylim(0,100) +  
 scale\_fill\_grey() +  
 theme\_linedraw() +  
 geom\_text(aes(label = paste0(Percent, "%")), position = position\_dodge(width = .9), vjust = -.5)

## Scale for y is already present.  
## Adding another scale for y, which will replace the existing scale.

quart\_plot + labs(x = "Quartile based on nonprofit asset amount", y = "Percentage of nonprofits in the asset quartile")



**this does NOT include all 4,082 organizations**

## Overall Percentage Plot

First let’s get a count of each - **NOTE we are keeping zero values and NA as low asset**! The NA neighborhood means there is only one neighborhood that did not fit the categories or have information. We can drop this neighborhood.

df\_simplified %>%  
 count(ASSET\_High\_text, Neighborhood)

## # A tibble: 5 × 3  
## ASSET\_High\_text Neighborhood n  
## <chr> <fct> <int>  
## 1 High Asset "Majority\nNon-Black" 539  
## 2 High Asset "Majority\nBlack" 222  
## 3 Low Asset "Majority\nNon-Black" 1589  
## 4 Low Asset "Majority\nBlack" 1732  
## 5 Low Asset <NA> 1

df\_simplified <-df\_simplified %>%  
 drop\_na(Neighborhood)  
  
df\_simplified %>%  
 count(ASSET\_High\_text, Neighborhood)

## # A tibble: 4 × 3  
## ASSET\_High\_text Neighborhood n  
## <chr> <fct> <int>  
## 1 High Asset "Majority\nNon-Black" 539  
## 2 High Asset "Majority\nBlack" 222  
## 3 Low Asset "Majority\nNon-Black" 1589  
## 4 Low Asset "Majority\nBlack" 1732

High\_asset\_data <-df\_simplified %>%   
 group\_by(ASSET\_High\_text, Neighborhood) %>%   
 count()   
High\_asset\_data

## # A tibble: 4 × 3  
## # Groups: ASSET\_High\_text, Neighborhood [4]  
## ASSET\_High\_text Neighborhood n  
## <chr> <fct> <int>  
## 1 High Asset "Majority\nNon-Black" 539  
## 2 High Asset "Majority\nBlack" 222  
## 3 Low Asset "Majority\nNon-Black" 1589  
## 4 Low Asset "Majority\nBlack" 1732

Create percentage variable for each category:

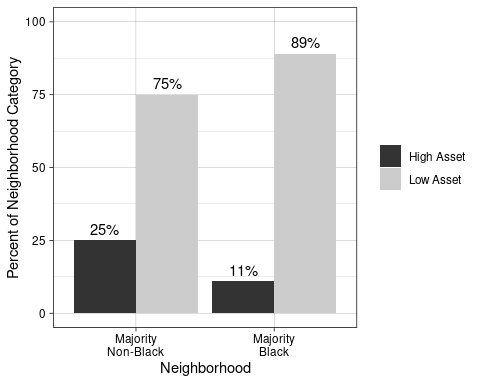
High\_asset\_data <- High\_asset\_data %>%  
 group\_by(Neighborhood) %>%   
 mutate(Percent\_AA\_cat = round(n/sum(n)\*100))   
High\_asset\_data

## # A tibble: 4 × 4  
## # Groups: Neighborhood [2]  
## ASSET\_High\_text Neighborhood n Percent\_AA\_cat  
## <chr> <fct> <int> <dbl>  
## 1 High Asset "Majority\nNon-Black" 539 25  
## 2 High Asset "Majority\nBlack" 222 11  
## 3 Low Asset "Majority\nNon-Black" 1589 75  
## 4 Low Asset "Majority\nBlack" 1732 89

Visuals…of the above data:

High\_asset\_data %>%   
 ggplot(aes(x= Neighborhood, y = Percent\_AA\_cat, fill = ASSET\_High\_text)) +   
 geom\_col(position = position\_dodge(width = .9))+  
 scale\_y\_continuous(labels = function(x) paste0(x, "%")) +  
 ylim(0,100) +  
 geom\_text(aes(label = paste0(Percent\_AA\_cat, "%")), position = position\_dodge(width = .9), vjust = -.5) +  
 ylab("Percent of Neighborhood Category") +  
 theme\_linedraw() +  
 scale\_fill\_grey() +  
 theme(legend.title = element\_blank())

## Scale for y is already present.  
## Adding another scale for y, which will replace the existing scale.

 **this includes all 4,082 organizations**

## High vs non asset by category

First create data to make visualization easier

High\_asset\_data <-df\_simplified %>%  
 group\_by(ASSET\_High\_text, Neighborhood, NTEE\_text) %>%   
 count()  
High\_asset\_data

## # A tibble: 36 × 4  
## # Groups: ASSET\_High\_text, Neighborhood, NTEE\_text [36]  
## ASSET\_High\_text Neighborhood NTEE\_text n  
## <chr> <fct> <chr> <int>  
## 1 High Asset "Majority\nNon-Black" Arts 34  
## 2 High Asset "Majority\nNon-Black" Education 54  
## 3 High Asset "Majority\nNon-Black" Environment/Animals 9  
## 4 High Asset "Majority\nNon-Black" Health 65  
## 5 High Asset "Majority\nNon-Black" Human Services 75  
## 6 High Asset "Majority\nNon-Black" International Affairs 4  
## 7 High Asset "Majority\nNon-Black" NA 174  
## 8 High Asset "Majority\nNon-Black" Religious 2  
## 9 High Asset "Majority\nNon-Black" Societal Benefit 122  
## 10 High Asset "Majority\nBlack" Arts 8  
## # ℹ 26 more rows

#Create percentage variable for each category  
High\_asset\_data <- High\_asset\_data %>%  
 group\_by(NTEE\_text) %>%   
 mutate(Percent\_ntee\_cat = round(n/sum(n)\*100))   
High\_asset\_data

## # A tibble: 36 × 5  
## # Groups: NTEE\_text [9]  
## ASSET\_High\_text Neighborhood NTEE\_text n Percent\_ntee\_cat  
## <chr> <fct> <chr> <int> <dbl>  
## 1 High Asset "Majority\nNon-Black" Arts 34 13  
## 2 High Asset "Majority\nNon-Black" Education 54 18  
## 3 High Asset "Majority\nNon-Black" Environment/Ani… 9 15  
## 4 High Asset "Majority\nNon-Black" Health 65 25  
## 5 High Asset "Majority\nNon-Black" Human Services 75 9  
## 6 High Asset "Majority\nNon-Black" International A… 4 9  
## 7 High Asset "Majority\nNon-Black" NA 174 13  
## 8 High Asset "Majority\nNon-Black" Religious 2 0  
## 9 High Asset "Majority\nNon-Black" Societal Benefit 122 23  
## 10 High Asset "Majority\nBlack" Arts 8 3  
## # ℹ 26 more rows

Visuals…of the above data:

High\_asset\_data %>%   
 ggplot(aes(x= Neighborhood, y = Percent\_ntee\_cat, fill = ASSET\_High\_text)) +   
 geom\_col(position = position\_dodge(width = .9))+  
 scale\_y\_continuous(labels = function(x) paste0(x, "%")) +  
 ylim(0, 100) +  
 geom\_text(aes(label = paste0(Percent\_ntee\_cat, "%")), position = position\_dodge(width = .9), vjust = -.5) + facet\_wrap(~NTEE\_text) +  
 theme\_linedraw() +  
 scale\_fill\_grey() +  
 theme(legend.title = element\_blank()) +  
 ylab("Percentage for each category")

**this includes all 4,082 organizations**

## Count plots/Tables

### Different kinds of orgs

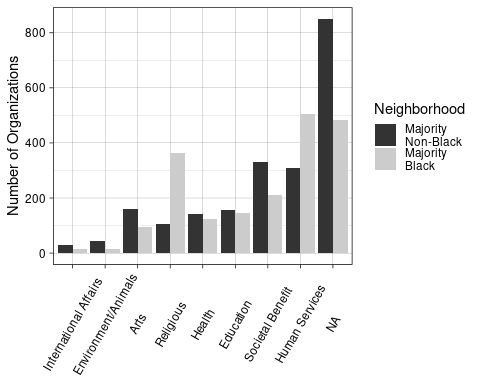
library(forcats)  
df\_simplified %>% group\_by(NTEE\_text) %>%summarize(count = n()) %>%   
 mutate(NTEE\_text = str\_replace(string = NTEE\_text, pattern = "NA", replacement = "Unclassified")) %>%  
 mutate(Percentage = round(count/sum(count)\*100, digits = 2)) %>%  
 arrange(NTEE\_text)

## # A tibble: 9 × 3  
## NTEE\_text count Percentage  
## <chr> <int> <dbl>  
## 1 Arts 255 6.25  
## 2 Education 305 7.47  
## 3 Environment/Animals 59 1.45  
## 4 Health 264 6.47  
## 5 Human Services 814 19.9   
## 6 International Affairs 43 1.05  
## 7 Religious 469 11.5   
## 8 Societal Benefit 540 13.2   
## 9 Unclassified 1333 32.7

Total\_NTEE <-df\_simplified %>% group\_by(NTEE\_text) %>%summarize(count = n()) %>%   
 mutate(NTEE\_text = str\_replace(string = NTEE\_text, pattern = "NA", replacement = "Unclassified")) %>%  
 arrange(NTEE\_text)

df\_simplified %>%   
 group\_by(NTEE\_text, Neighborhood) %>%  
 summarize(count = n()) %>%   
 mutate(NTEE\_text = as\_factor(NTEE\_text)) %>%  
 ggplot(aes(x = fct\_reorder(NTEE\_text, count, min), y = count , fill = Neighborhood)) +   
 scale\_fill\_viridis\_d() +  
 geom\_col(position =position\_dodge(width = .9)) +   
 ylab ("Number of Organizations") +  
 theme\_linedraw() +  
 theme(axis.text.x = element\_text(angle = 60, vjust = .5),  
 axis.title.x = element\_blank()) +  
 scale\_fill\_grey()

## `summarise()` has grouped output by 'NTEE\_text'. You can override using the  
## `.groups` argument.  
## Scale for fill is already present. Adding another scale for fill, which will  
## replace the existing scale.

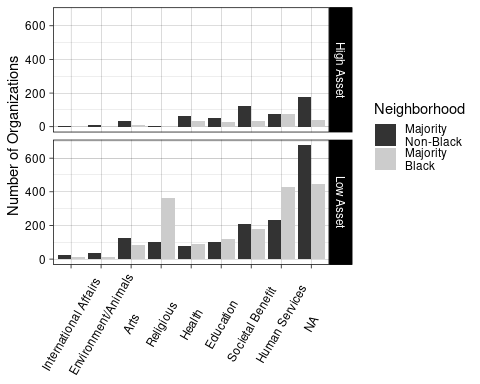


**This includes all 4,082 organizations** There was no removal of organizations based on asset amount, just to get a sense of what oganizations are in Baltimore.

plot2 <- df\_simplified %>%   
 mutate(NTEE\_text = as\_factor(NTEE\_text),  
 NTEE\_text = forcats::fct\_relevel(NTEE\_text, "International Affairs", "Environment/Animals", "Arts", "Religious", "Health","Education", "Societal Benefit", "Human Services", "NA" )) %>%  
 group\_by(NTEE\_text, Neighborhood, ASSET\_High\_text) %>%  
 summarize(count = n()) %>%   
 ggplot(aes(x = NTEE\_text, y = count , fill = Neighborhood)) +   
 geom\_col(position =position\_dodge(width = .9)) +   
 facet\_grid(rows = vars(ASSET\_High\_text)) +  
 ylab ("Number of Organizations") +  
 theme\_linedraw() +  
 theme(axis.text.x = element\_text(angle = 60, vjust = .5),  
 axis.title.x = element\_blank()) +  
 scale\_fill\_grey()

## `summarise()` has grouped output by 'NTEE\_text', 'Neighborhood'. You can  
## override using the `.groups` argument.

plot2



**This includes all 4,082 organizations** There was no removal of organizations based on asset amount, just to get a sense of what oganizations are in Baltimore.

### High Asset Orgs

High\_counts <- df\_simplified %>%   
 mutate(NTEE\_text = as\_factor(NTEE\_text),  
 NTEE\_text = forcats::fct\_relevel(NTEE\_text, "International Affairs", "Environment/Animals", "Arts", "Religious", "Health","Education", "Societal Benefit", "Human Services", "NA" )) %>%  
 group\_by(NTEE\_text, ASSET\_High\_text) %>%  
 summarize(count = n()) %>% filter(ASSET\_High\_text == "High Asset") %>%  
 mutate(NTEE\_text = str\_replace(string = NTEE\_text, pattern = "NA", replacement = "Unclassified"))

## `summarise()` has grouped output by 'NTEE\_text'. You can override using the  
## `.groups` argument.

full\_join(Total\_NTEE, High\_counts, by = "NTEE\_text") %>%  
 mutate("Percentage\_of\_each\_code" = round(count.y/count.x \*100, digits = 2)) %>%  
 arrange(NTEE\_text)

## # A tibble: 9 × 5  
## NTEE\_text count.x ASSET\_High\_text count.y Percentage\_of\_each\_code  
## <chr> <int> <chr> <int> <dbl>  
## 1 Arts 255 High Asset 42 16.5   
## 2 Education 305 High Asset 84 27.5   
## 3 Environment/Animals 59 High Asset 12 20.3   
## 4 Health 264 High Asset 97 36.7   
## 5 Human Services 814 High Asset 152 18.7   
## 6 International Affairs 43 High Asset 5 11.6   
## 7 Religious 469 High Asset 3 0.64  
## 8 Societal Benefit 540 High Asset 155 28.7   
## 9 Unclassified 1333 High Asset 211 15.8

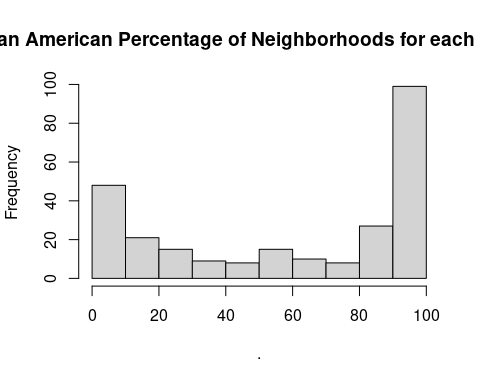
## Distribution of percent AA

Now to take a look at if 50% African American makes sense. What do the neighborhoods look like?

# get the neighborhood values if no removing rows for nonprofits with NA or zero assets  
neighborhood\_AAperc <- df\_simplified %>%   
 distinct(`neighborhood name`, Percent\_AA)  
  
# get the neighborhood values after removing rows for nonprofits with NA or zero assets  
neighborhood\_AAperc\_nozero <- df\_simplified\_no\_zero %>%   
 distinct(`neighborhood name`, Percent\_AA)

We can see that there are many neighborhoods that have a more extreme percentage.

# get the neighborhood values if no removing rows for nonprofits with NA or zero assets  
  
neighborhood\_AAperc%>% pull(Percent\_AA) %>% hist(main = "African American Percentage of Neighborhoods for each nonprofit")



# get the neighborhood values after removing rows for nonprofits with NA or zero assets  
neighborhood\_AAperc\_nozero %>% pull(Percent\_AA) %>% hist(main = "African American Percentage of Neighborhoods \n (removed neighborhoods with only zero or NA assets)")

