City Open Data Portals and Library Data

```
# Load relevant libraries
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#library(Hmisc)
#library(summarytools)
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.5.2
```

```
library(splitstackshape)
```

```
## Warning: package 'splitstackshape' was built under R version 3.5.2
```

```
library(httr)
```

```
## Warning: package 'httr' was built under R version 3.5.2
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
# retrieve token needed to access data in private github repository
token <- read.delim("~/Google Drive File Stream/My Drive/Keys/g_access.txt", stringsAsFactor = FALSE, h
```

```
## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
## incomplete final line found by readTableHeader on '~/Google Drive File Stream/My
## Drive/Keys/g_access.txt'
```

```
# Read in portals data stored in private github repository
url=paste0("https://raw.githubusercontent.com/OpenDataLiteracy/JCDL-Extended/master/Data/KAS_JCDL_Preli
x=GET(url, add_headers(Authorization = paste("token", token, sep = " ")))
portals <- content(x, type="text/csv", encoding = "UTF-8")
```

```
## Parsed with column specification:
## cols(
##   City = col_character(),
##   State = col_character(),
##   Portal_URL = col_character(),
##   DatePortalAccessed = col_date(format = ""),
##   Software = col_character(),
##   TotalDataSetsAvailable = col_number(),
##   NotesDataSetsAvailable = col_character(),
##   CountVettedPublicLibData = col_double(),
##   TypePLDataAvailable = col_character(),
```

```
##   LibraryDataCategories = col_character(),
##   DateLibDataLastUpdated = col_date(format = ""),
##   Notes = col_character(),
##   OpenDataCensusScore2017 = col_character()
## )
```

```
# Read in IMLS data stored in private github repository
url=paste0("https://raw.githubusercontent.com/OpenDataLiteracy/JCDL-Extended/master/Data/IMLS_Data_PLS_
x=GET(url, add_headers(Authorization = paste("token", token, sep = " ")))
imls <- content(x, type="text/csv", encoding = "UTF-8")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   STABR = col_character(),
##   FSCSKEY = col_character(),
##   LIBID = col_character(),
##   LIBNAME = col_character(),
##   ADDRESS = col_character(),
##   CITY = col_character(),
##   ZIP4 = col_character(),
##   ADDRES_M = col_character(),
##   CITY_M = col_character(),
##   ZIP4_M = col_character(),
##   CNTY = col_character(),
##   C_RELATN = col_character(),
##   C_LEGBAS = col_character(),
##   C_ADMIN = col_character(),
##   C_FSCS = col_character(),
##   GEOCODE = col_character(),
##   LSABOUND = col_character(),
##   STARTDAT = col_character(),
##   ENDDATE = col_character(),
##   F_POPLSA = col_character()
##   # ... with 54 more columns
## )
```

```
## See spec(...) for full column specifications.
```

```
# Read in OKFN Open Data Scores data stored in private github repository
#url=paste0("https://raw.githubusercontent.com/OpenDataLiteracy/JCDL-Extended/master/Data/2020-01-16_Do
#x=GET(url, add_headers(Authorization = paste("token", token, sep = " ")))
#census_scores <- content(x, type="text/csv", encoding = "UTF-8")
```

```
# Read in IMLS data stored in private github repository
#url=paste0("https://raw.githubusercontent.com/OpenDataLiteracy/JCDL-Extended/master/Data/PLS_FY17_Outl
#x=GET(url, add_headers(Authorization = paste("token", token, sep = " ")))
#imls_outlet <- content(x, type="text/csv", encoding = "UTF-8")
```

```
# Fix date formats
portals$DateLibDataLastUpdated <- as.Date(portals$DateLibDataLastUpdated, format = "%Y-%m-%d")
portals$DatePortalAccessed <- as.Date(portals$DatePortalAccessed, format = "%Y-%m-%d")
# Variables to datatype character
portals$Portal_URL <- as.character(portals$Portal_URL)
# Replace N/A with na
portals$OpenDataCensusScore2017 <- ifelse(portals$OpenDataCensusScore2017 != "N/A", portals$OpenDataCens
```

```r
# Variable to numeric
portals$OpenDataCensusScore2017 <- as.numeric(portals$OpenDataCensusScore2017)
```

```r
# Split categories into individual categories based on comma dileaneation, then create dummy variables.
portals <- cSplit_e(portals, split.col = "LibraryDataCategories", sep = ",", type = "character",
         mode = "binary", fixed = TRUE, fill = 0)
```

```r
# Calculate (and add column) the percentage of the city portal data that is public library related
portals$ProportionPublicLibData <- portals$CountVettedPublicLibData / portals$TotalDataSetsAvailable
```

```r
#write.csv(portals, "~/Documents/Github/JCDL-Extended/Data/CityPortals_CategoryDummies")
```

```r
# Fix different capitalizations
portals <- portals %>%
       mutate(City = tolower(City),
              State = tolower(State))

imls <- imls %>%
       mutate(CITY = tolower(CITY),
              STABR = tolower(STABR))

# Merge datasets on city, state
merged <- merge(portals, imls, by.x = c("City", "State"), by.y = c("CITY", "STABR"), all.x = TRUE)
```

```r
#merged <- merge(temp, imls_outlet, by = "LIBNAME", all.x = TRUE)
```

```r
# Variables to datatype factor
# jcdl$Locale <- as.factor(jcdl$Locale)
# jcdl$ReportingStatus <- as.factor(jcdl$ReportingStatus)
# jcdl$MailingZip <- as.factor(jcdl$MailingZip)
# jcdl$Year <- as.factor(jcdl$Year)
```

```r
# Check unique values
unique(merged$Software)
```

```
## [1] "Other"        "Socrata"      "Arcgis"       "Junar"        NA
## [6] "CKAN"         "Opendatasoft" "DKAN"         "JKAN"
```

```r
# See what values we have for Locale
unique(merged$GEOCODE)
```

```
##  [1] "CO1" "OTH" "MA1" "SD1" "CI1" "SD2" "CI2" NA    "MA2" "MC2" "CO2" "MC1"
```

```r
# See what values we have for Locale
unique(merged$LOCALE_ADD)
```

```
## [1] 11 21 12 13 NA 33 41
```

```r
# See what values we have for Locale
unique(merged$LOCALE_MOD)
```

```
## [1] 11 21 12 13 NA 42 33 22 23
```

Definitions of LOCALE features see PLS Data File Documentation (https://www.imls.gov/sites/default/files/fy2017_pls_data_file_documentation.pdf) page 23

```r
# Create a Description column for both LOCALE_ADD and LOCALE_MOD and fill based on condition
merged <- merged %>%
    mutate(LOCALE_ADD_DESCR = case_when(
```

```
                LOCALE_ADD == 11 ~ "City Large",
                LOCALE_ADD == 12 ~ "City Midsize",
                LOCALE_ADD == 13 ~ "City Small",
                LOCALE_ADD == 21 ~ "Suburban Large",
                LOCALE_ADD == 22 ~ "Suburban Midsize",
                LOCALE_ADD == 23 ~ "Suburban Small",
                LOCALE_ADD == 31 ~ "Town Fringe",
                LOCALE_ADD == 32 ~ "Town Distant",
                LOCALE_ADD == 33 ~ "Town Remote",
                LOCALE_ADD == 41 ~ "Rural Fringe",
                LOCALE_ADD == 42 ~ "Rural Distant",
                LOCALE_ADD == 43 ~ "Rural Remote"),
            LOCALE_MOD_DESCR = case_when(
                LOCALE_MOD == 11 ~ "City Large",
                LOCALE_MOD == 12 ~ "City Midsize",
                LOCALE_MOD == 13 ~ "City Small",
                LOCALE_MOD == 21 ~ "Suburban Large",
                LOCALE_MOD == 22 ~ "Suburban Midsize",
                LOCALE_MOD == 23 ~ "Suburban Small",
                LOCALE_MOD == 31 ~ "Town Fringe",
                LOCALE_MOD == 32 ~ "Town Distant",
                LOCALE_MOD == 33 ~ "Town Remote",
                LOCALE_MOD == 41 ~ "Rural Fringe",
                LOCALE_MOD == 42 ~ "Rural Distant",
                LOCALE_MOD == 43 ~ "Rural Remote")
            )
```

```
# Change datatype of LOCALE DESCRs
merged$LOCALE_ADD_DESCR <- as.factor(merged$LOCALE_ADD_DESCR)
merged$LOCALE_MOD_DESCR <- as.factor(merged$LOCALE_MOD_DESCR)
```

```
merged %>%
  count(City, State, sort = T) %>%
  filter(n > 1)
```

```
## # A tibble: 17 x 3
##    City        State     n
##    <chr>       <chr> <int>
##  1 pittsburgh  pa       16
##  2 austin      tx        4
##  3 birmingham  al        3
##  4 dallas      tx        3
##  5 portland    or        3
##  6 syracuse    ny        3
##  7 albuquerque nm        2
##  8 boise       id        2
##  9 ferndale    mi        2
## 10 houston     tx        2
## 11 new orleans la        2
## 12 phoenix     az        2
## 13 providence  ri        2
## 14 san antonio tx        2
## 15 san diego   ca        2
## 16 scottsdale  az        2
```

```
## 17 st. louis   mo        2
```

```
# Count how many library systems are included in the individual cities add colum to dataframe
merged <- merged %>%
  add_count(City, State, name = "CountLibSysinCity")
```

```
multi_libs <- merged %>%
  filter(CountLibSysinCity > 1)

write.csv(multi_libs, "~/Documents/Github/JCDL-Extended/Data/Cities_with_multiple_library_systems")
```

```
single_libs <- merged %>%
  filter(CountLibSysinCity == 1)
```

```
# Uncomment this to view a nice table of descriptive stats in the Viewer
#summarytools::view(summarytools::dfSummary(merged))
```

```
# Uncomment for descriptive stats
#Hmisc::describe(merged)
```

```
# Simple linear regression model
single_libs_PropLR <- lm(ProportionPublicLibData ~ OpenDataCensusScore2017, data=single_libs)
summary(single_libs_PropLR)
```

```
##
## Call:
## lm(formula = ProportionPublicLibData ~ OpenDataCensusScore2017,
##     data = single_libs)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.015373 -0.012181 -0.010284 -0.001986  0.123804
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              9.863e-03  6.378e-03   1.546    0.127
## OpenDataCensusScore2017 3.011e-06  6.674e-06   0.451    0.653
##
## Residual standard error: 0.02769 on 63 degrees of freedom
##   (49 observations deleted due to missingness)
## Multiple R-squared:  0.003221,   Adjusted R-squared:  -0.0126
## F-statistic: 0.2036 on 1 and 63 DF,  p-value: 0.6534
```

```
# Simple linear regression model
single_libs_VettLR <- lm(CountVettedPublicLibData ~ OpenDataCensusScore2017, data=single_libs)
summary(single_libs_VettLR)
```

```
##
## Call:
## lm(formula = CountVettedPublicLibData ~ OpenDataCensusScore2017,
##     data = single_libs)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.381 -2.054 -0.728  0.171 53.180
##
## Coefficients:
```

```
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -0.491990   1.748490  -0.281   0.7793
## OpenDataCensusScore2017  0.003209   0.001830   1.754   0.0843 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.591 on 63 degrees of freedom
##   (49 observations deleted due to missingness)
## Multiple R-squared:  0.04656,    Adjusted R-squared:  0.03143
## F-statistic: 3.077 on 1 and 63 DF,  p-value: 0.08429
```

```
# Simple linear regression model
single_libs_TotLR <- lm(TotalDataSetsAvailable ~ OpenDataCensusScore2017, data=single_libs)
summary(single_libs_TotLR)
```

```
##
## Call:
## lm(formula = TotalDataSetsAvailable ~ OpenDataCensusScore2017,
##     data = single_libs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -362.13 -132.93  -34.65   34.89 1592.45
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -10.13129   72.31391  -0.140  0.88903
## OpenDataCensusScore2017   0.25786    0.07567   3.408  0.00115 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 313.9 on 63 degrees of freedom
##   (49 observations deleted due to missingness)
## Multiple R-squared:  0.1556, Adjusted R-squared:  0.1422
## F-statistic: 11.61 on 1 and 63 DF,  p-value: 0.001147
```

```
# Simple linear regression model
single_libs_PropSizeLR <- lm(ProportionPublicLibData ~ LOCALE_ADD_DESCR, data=single_libs)
summary(single_libs_PropSizeLR)
```

```
##
## Call:
## lm(formula = ProportionPublicLibData ~ LOCALE_ADD_DESCR, data = single_libs)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.020408 -0.019721 -0.012534 -0.000056  0.123059
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     0.012534   0.005340   2.347   0.0218 *
## LOCALE_ADD_DESCRCity Midsize    0.007187   0.009102   0.790   0.4325
## LOCALE_ADD_DESCRCity Small      0.007874   0.013838   0.569   0.5712
## LOCALE_ADD_DESCRRural Fringe   -0.012534   0.034195  -0.367   0.7151
## LOCALE_ADD_DESCRSuburban Large  0.007466   0.016021   0.466   0.6427
## LOCALE_ADD_DESCRTown Remote    -0.012534   0.034195  -0.367   0.7151
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03378 on 69 degrees of freedom
##   (39 observations deleted due to missingness)
## Multiple R-squared:  0.01841,    Adjusted R-squared:  -0.05272
## F-statistic: 0.2588 on 5 and 69 DF,  p-value: 0.9339
```

```
# Simple linear regression model
single_libs_VettSizeLR <- lm(CountVettedPublicLibData ~ LOCALE_ADD_DESCR, data=single_libs)
summary(single_libs_VettSizeLR)
```

```
##
## Call:
## lm(formula = CountVettedPublicLibData ~ LOCALE_ADD_DESCR, data = single_libs)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.850 -1.850 -1.286 -0.248 55.150
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       2.850      1.177   2.421   0.0181 *
## LOCALE_ADD_DESCRCity Midsize     -1.755      2.007  -0.875   0.3849
## LOCALE_ADD_DESCRCity Small       -1.564      3.051  -0.513   0.6097
## LOCALE_ADD_DESCRRural Fringe     -2.850      7.538  -0.378   0.7065
## LOCALE_ADD_DESCRSuburban Large   -2.450      3.532  -0.694   0.4902
## LOCALE_ADD_DESCRTown Remote      -2.850      7.538  -0.378   0.7065
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.446 on 69 degrees of freedom
##   (39 observations deleted due to missingness)
## Multiple R-squared:  0.01807,    Adjusted R-squared:  -0.05308
## F-statistic: 0.254 on 5 and 69 DF,  p-value: 0.9364
```

```
# Simple linear regression model
single_libs_TotSizeLR <- lm(TotalDataSetsAvailable ~ LOCALE_ADD_DESCR, data=single_libs)
summary(single_libs_TotSizeLR)
```

```
##
## Call:
## lm(formula = TotalDataSetsAvailable ~ LOCALE_ADD_DESCR, data = single_libs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -184.75  -94.59  -36.75   24.25 1407.25
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      194.75      34.28   5.681 2.93e-07 ***
## LOCALE_ADD_DESCRCity Midsize    -100.23      58.42  -1.716   0.0907 .
## LOCALE_ADD_DESCRCity Small       -71.32      88.82  -0.803   0.4247
## LOCALE_ADD_DESCRRural Fringe    -155.75     219.49  -0.710   0.4803
## LOCALE_ADD_DESCRSuburban Large  -144.75     102.83  -1.408   0.1637
## LOCALE_ADD_DESCRTown Remote     -154.75     219.49  -0.705   0.4831
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 216.8 on 69 degrees of freedom
##   (39 observations deleted due to missingness)
## Multiple R-squared:  0.06435,    Adjusted R-squared:  -0.003451
## F-statistic: 0.9491 on 5 and 69 DF,  p-value: 0.4552
```

```
# Simple linear regression model total datasets avail and total operating revenue
single_libs_TotRevCityLR <- lm(TotalDataSetsAvailable ~ TOTINCM, data=single_libs)
summary(single_libs_TotRevCityLR)
```

```
##
## Call:
## lm(formula = TotalDataSetsAvailable ~ TOTINCM, data = single_libs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -197.49  -91.21  -59.77   24.59 1456.41
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 9.968e+01  3.310e+01   3.012  0.00357 **
## TOTINCM     1.695e-06  8.121e-07   2.086  0.04043 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 211.7 on 73 degrees of freedom
##   (39 observations deleted due to missingness)
## Multiple R-squared:  0.05628,    Adjusted R-squared:  0.04335
## F-statistic: 4.353 on 1 and 73 DF,  p-value: 0.04043
```

```
# Simple linear regression model
single_libs_TotRevLR <- lm(CountVettedPublicLibData ~ TOTINCM, data=single_libs)
summary(single_libs_TotRevLR)
```

```
##
## Call:
## lm(formula = CountVettedPublicLibData ~ TOTINCM, data = single_libs)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.333 -1.639 -0.837 -0.379 50.928
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.283e-01  1.101e+00   0.207   0.8363
## TOTINCM     6.350e-08  2.702e-08   2.350   0.0215 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.044 on 73 degrees of freedom
##   (39 observations deleted due to missingness)
## Multiple R-squared:  0.07032,    Adjusted R-squared:  0.05758
## F-statistic: 5.522 on 1 and 73 DF,  p-value: 0.02149
```

```r
# Simple linear regression model
single_libs_TotStaffLR <- lm(CountVettedPublicLibData ~ TOTSTAFF, data=single_libs)
summary(single_libs_TotStaffLR)
```

```
##
## Call:
## lm(formula = CountVettedPublicLibData ~ TOTSTAFF, data = single_libs)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.509 -1.972 -0.650 -0.046 50.358
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.331742   1.242988  -0.267   0.7903
## TOTSTAFF     0.009171   0.003748   2.447   0.0168 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.023 on 73 degrees of freedom
##   (39 observations deleted due to missingness)
## Multiple R-squared:  0.07581,    Adjusted R-squared:  0.06315
## F-statistic: 5.988 on 1 and 73 DF,  p-value: 0.01681
```

```r
# Create temporary matrix to use in Pearson's correlation
x <- single_libs %>% select(TOTINCM, TOTSTAFF)
x$TOTINCM <- as.numeric(x$TOTINCM)
```

```r
# Calculate Pearson's correlation
cor(x, use = "complete.obs")
```

```
##            TOTINCM  TOTSTAFF
## TOTINCM  1.0000000 0.9337921
## TOTSTAFF 0.9337921 1.0000000
```

```r
# Run (simple) LM dependent variable ProportionPublicLibData against all
# possible numeric columns

# Adapted from https://stackoverflow.com/questions/30583917/regression-loop-in-r-for-data-frames

for(i in names(single_libs))
{
    if(is.numeric(single_libs[,i]))  ##if column is numeric run regression
    {
       fit <- lm(ProportionPublicLibData ~ single_libs[,i], data=single_libs)
       coeff <- summary(fit)$coefficients[,4][2] #output only the p-values
       writeLines(paste(coeff,i,"\n"))
    }
}
```

```r
# Run (simple) LM dependent variable CountVettedPublicLibData against all
# possible numeric columns

# Adapted from https://stackoverflow.com/questions/30583917/regression-loop-in-r-for-data-frames

for(i in names(single_libs))
```

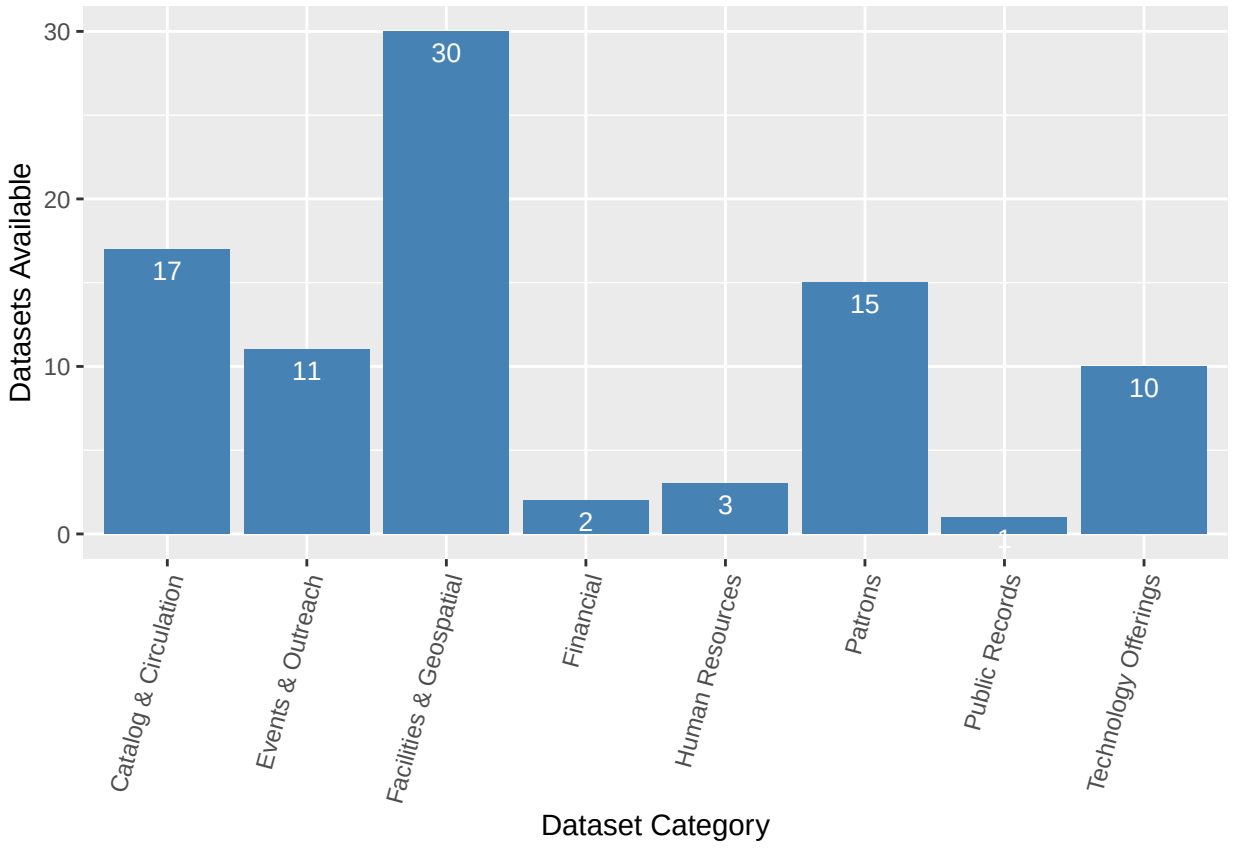
```
{
    if(is.numeric(single_libs[,i]))  ##if column is numeric run regression
    {
       fit <- lm(CountVettedPublicLibData ~ single_libs[,i], data=single_libs)
       coeff <- summary(fit)$coefficients[,4][2] #output only the p-values
       writeLines(paste(coeff,i,"\n"))
    }
}
```

```
dfm <- reshape2::melt(single_libs[,c('LibraryDataCategories_CatalogAndCirculation','LibraryDataCategori
                      'LibraryDataCategories_FacilitiesAndGeospatial', 'LibraryDataCategories_Financial',
                      'LibraryDataCategories_HumanResources', 'LibraryDataCategories_Patrons',
                      'LibraryDataCategories_PublicRecords', 'LibraryDataCategories_TechnologyOfferings')]
```

```
## No id variables; using all as measure variables
```

```
# total datasets by category
dfm <- dfm %>% group_by(variable) %>% summarise("total" = sum(value))
```

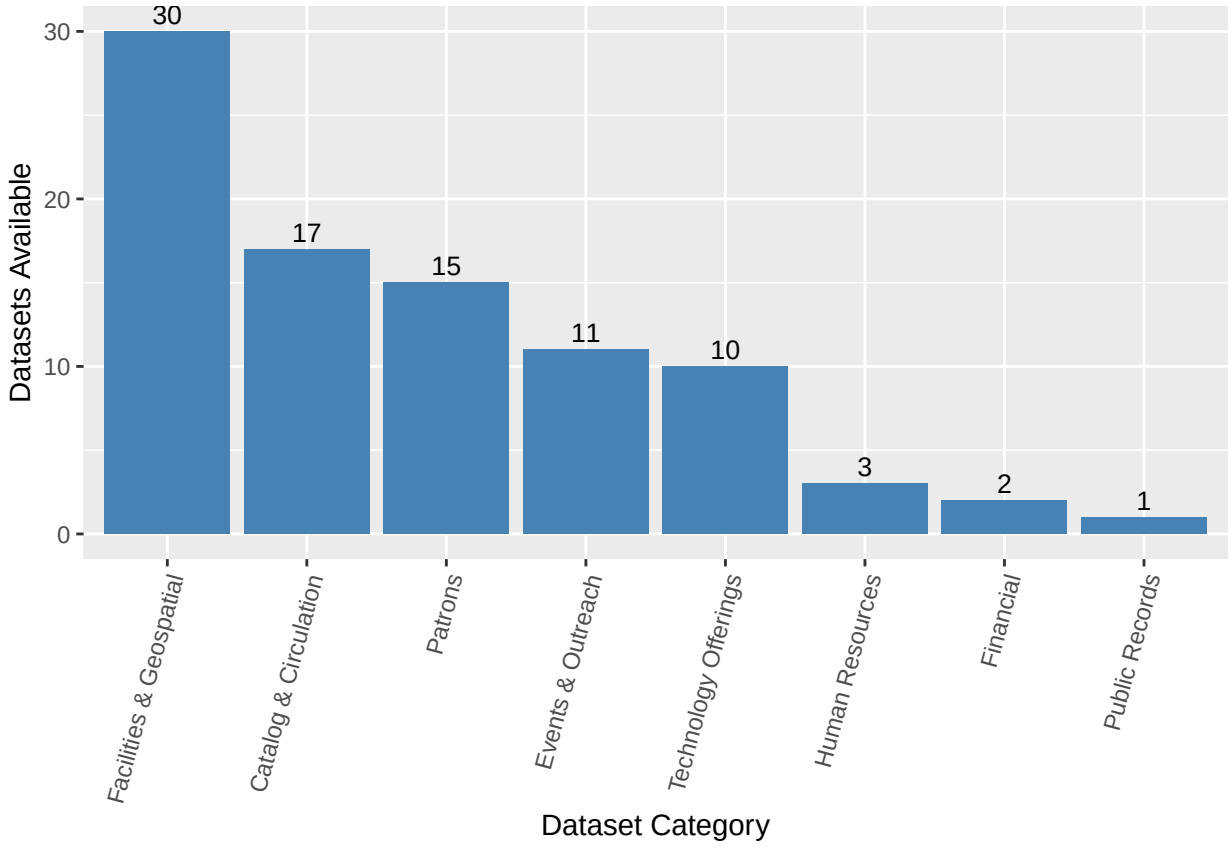
```
# barchart of datset totals by category
p <- ggplot(dfm, aes(variable, total)) +
  geom_bar(stat="identity", fill="steelblue") +
  scale_x_discrete(labels=c("LibraryDataCategories_CatalogAndCirculation" = "Catalog & Circulation",
                            "LibraryDataCategories_EventsAndOutreach" = "Events & Outreach",
                            "LibraryDataCategories_FacilitiesAndGeospatial" = "Facilities & Geospatial"
                            "LibraryDataCategories_Financial" = "Financial",
                            "LibraryDataCategories_HumanResources" = "Human Resources",
                            "LibraryDataCategories_Patrons" = "Patrons",
                            "LibraryDataCategories_PublicRecords" = "Public Records",
                            "LibraryDataCategories_TechnologyOfferings" = "Technology Offerings")) +
  theme(axis.text.x = element_text(angle = 75, hjust = 1)) +
  geom_text(aes(label=total), vjust=1.6, color="white", size=3.5) +
  xlab("Dataset Category") +
  ylab("Datasets Available")
p
```




```
ggsave("~/Documents/Github/JCDL-Extended/Images/categories_barchart.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
# same barchart as above but arranged in descending order of dataset count
p <- ggplot(dfm, aes(x = reorder(variable, -total), total)) +
  geom_bar(stat="identity", fill="steelblue") +
  scale_x_discrete(labels=c("LibraryDataCategories_CatalogAndCirculation" = "Catalog & Circulation",
                            "LibraryDataCategories_EventsAndOutreach" = "Events & Outreach",
                            "LibraryDataCategories_FacilitiesAndGeospatial" = "Facilities & Geospatial",
                            "LibraryDataCategories_Financial" = "Financial",
                            "LibraryDataCategories_HumanResources" = "Human Resources",
                            "LibraryDataCategories_Patrons" = "Patrons",
                            "LibraryDataCategories_PublicRecords" = "Public Records",
                            "LibraryDataCategories_TechnologyOfferings" = "Technology Offerings")) +
  theme(axis.text.x = element_text(angle = 75, hjust = 1)) +
  geom_text(aes(label=total), vjust=-0.4, color="black", size=3.5) +
  xlab("Dataset Category") +
  ylab("Datasets Available")
p
```




```
ggsave("~/Documents/Github/JCDL-Extended/Images/categories_barchart_sorted.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
nrow(single_libs[single_libs$CountVettedPublicLibData > 0,])
```

```
## [1] 59
```

```
length(single_libs$CountVettedPublicLibData[single_libs$CountVettedPublicLibData > 0])
```

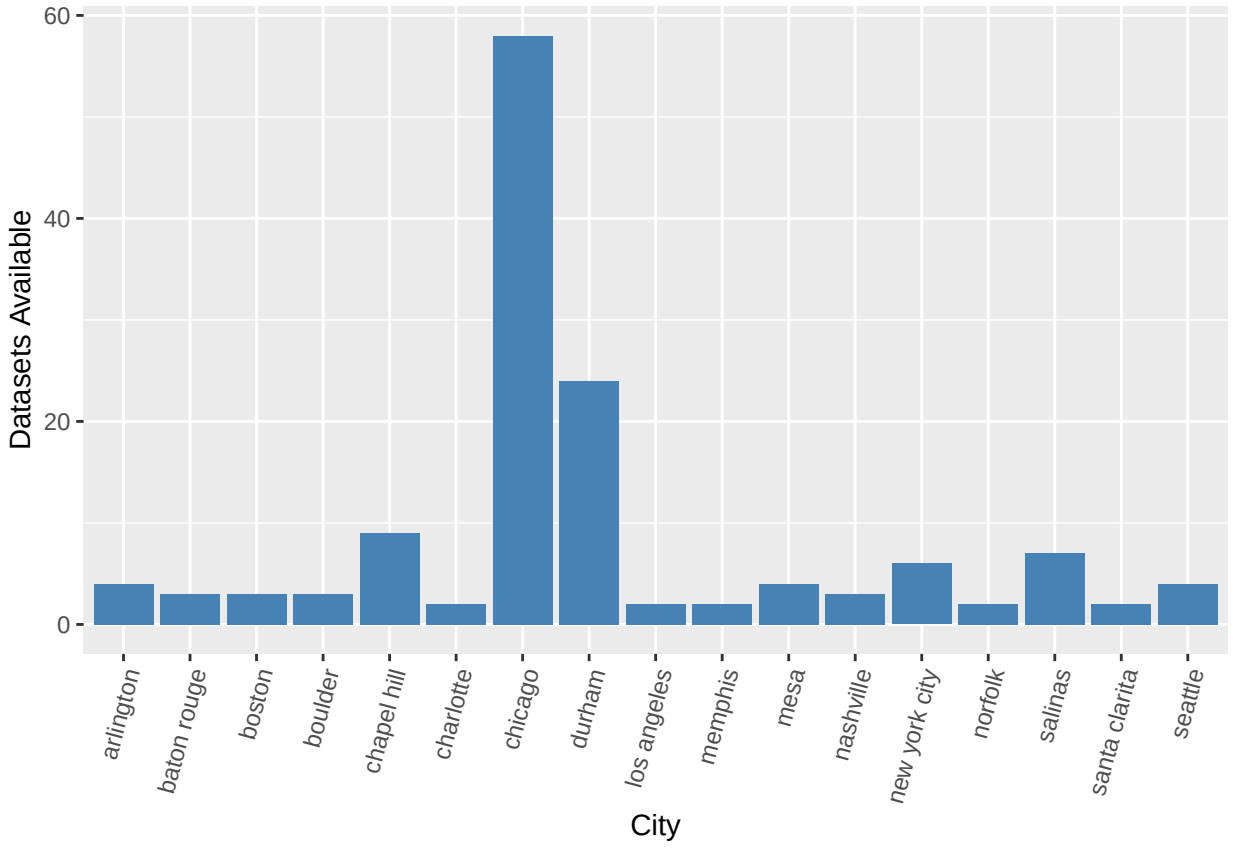
```
## [1] 59
```

```
sum(which(single_libs$CountVettedPublicLibData > 0))
```

```
## [1] 2082
```

```
# Create df with just necessary columns and rows where CountVettedPublicLibData is greater than 0
no_ds <- single_libs %>%
  select(City, State, Software, TotalDataSetsAvailable, CountVettedPublicLibData, ProportionPublicLibDa
  filter(CountVettedPublicLibData > 0)
```

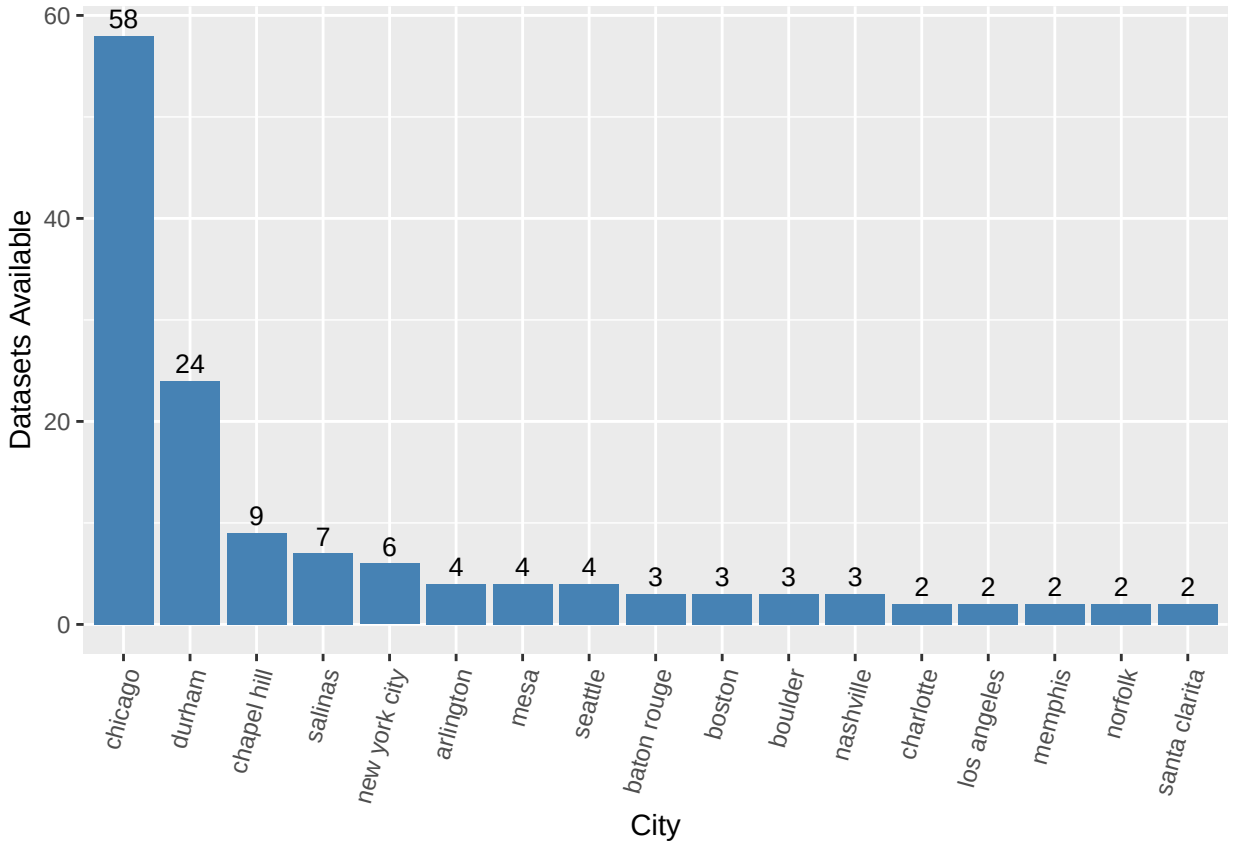
```
# barchart of cities with more than 1 library dataset
no_ds2 <- filter(no_ds, CountVettedPublicLibData > 1)
p1 <- ggplot(no_ds2, aes(City, CountVettedPublicLibData)) +
  geom_bar(stat="identity", fill="steelblue") +
  theme(axis.text.x = element_text(angle = 75, hjust = 1)) +
  xlab("City") +
  ylab("Datasets Available")
p1
```




```
ggsave("~/Documents/Github/JCDL-Extended/Images/cities_barchart.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
# sorted barchart of cities with more than 1 library dataset
p1 <- ggplot(no_ds2, aes(x = reorder(City, -CountVettedPublicLibData), CountVettedPublicLibData)) +
  geom_bar(stat="identity", fill="steelblue") +
  theme(axis.text.x = element_text(angle = 75, hjust = 1)) +
  geom_text(aes(label=CountVettedPublicLibData), vjust=-0.4, color="black", size=3.5) +
  xlab("City") +
  ylab("Datasets Available")
p1
```




```
ggsave("~/Documents/Github/JCDL-Extended/Images/cities_barchart_sorted_morethanone.png")
```

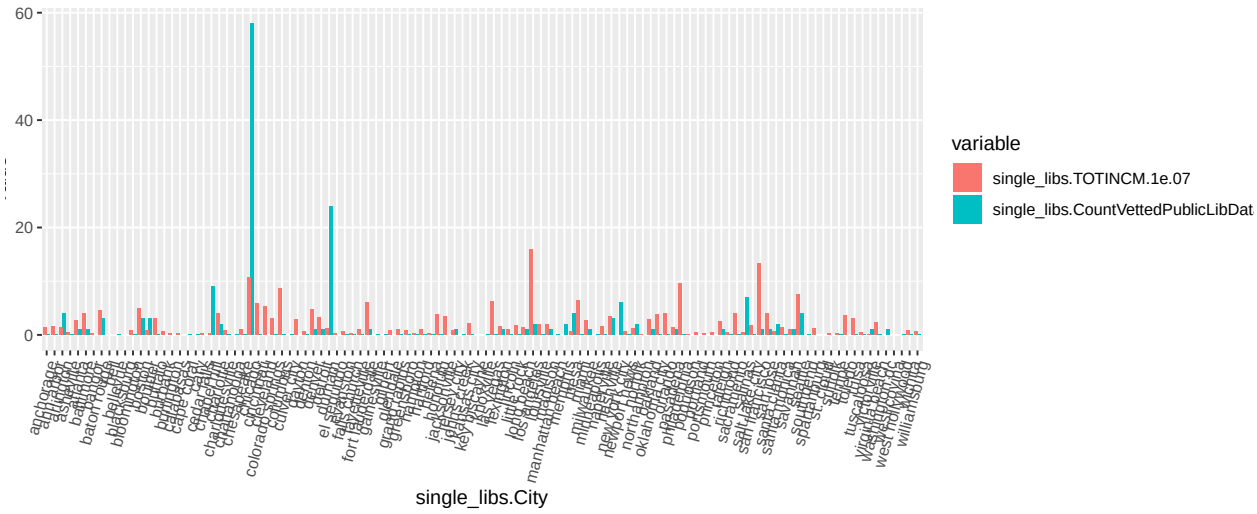
```
## Saving 6.5 x 4.5 in image
```

```
df1 <- data.frame(single_libs$TOTINCM/10000000, single_libs$CountVettedPublicLibData, single_libs$City)
df2 <- reshape2::melt(df1, id.vars='single_libs.City')
head(df2)
```

```
##   single_libs.City                  variable     value
## 1        anchorage single_libs.TOTINCM.1e.07 1.3233568
## 2        ann arbor single_libs.TOTINCM.1e.07 1.6098184
## 3        arlington single_libs.TOTINCM.1e.07 0.8461039
## 4        arlington single_libs.TOTINCM.1e.07 1.3648924
## 5        asheville single_libs.TOTINCM.1e.07 0.5171832
## 6          atlanta single_libs.TOTINCM.1e.07 2.7096418
```

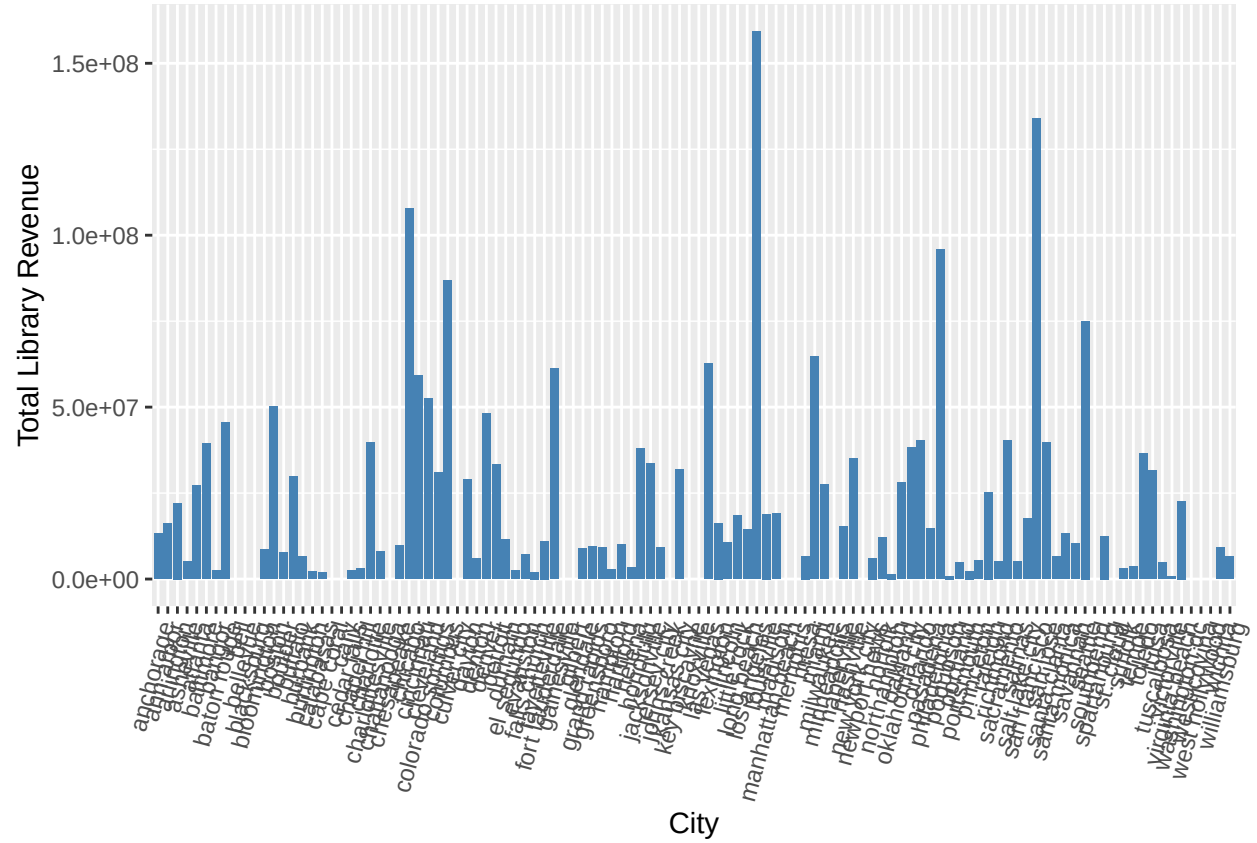
```
ggplot(df2, aes(x=single_libs.City, y=value, fill=variable)) +
    geom_bar(stat='identity', position='dodge')+
  theme(axis.text.x = element_text(angle = 75, hjust = 1))
```

```
## Warning: Removed 44 rows containing missing values (geom_bar).
```




```
p1 <- ggplot(single_libs, aes(City, TOTINCM)) +
  geom_bar(stat="identity", fill="steelblue") +
  theme(axis.text.x = element_text(angle = 75, hjust = 1)) +
  xlab("City") +
  ylab("Total Library Revenue")
p1
```

```
## Warning: Removed 21 rows containing missing values (position_stack).
```




```
ggsave("~/Documents/Github/JCDL-Extended/Images/cities_revenue_barchart.png")
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 21 rows containing missing values (position_stack).
```

```
single_libs %>%
  select(City, CountVettedPublicLibData) %>%
  arrange(desc(CountVettedPublicLibData))
```

```
## # A tibble: 114 x 2
##    City          CountVettedPublicLibData
##    <chr>                            <dbl>
##  1 chicago                             58
##  2 durham                              24
##  3 chapel hill                          9
##  4 salinas                              7
##  5 new york city                        6
##  6 arlington                            4
##  7 mesa                                 4
##  8 seattle                              4
##  9 baton rouge                          3
## 10 boston                               3
## # ... with 104 more rows
```

```
single_libs %>%
  select(City, OpenDataCensusScore2017) %>%
  arrange(desc(OpenDataCensusScore2017))
```

```
## # A tibble: 114 x 2
##    City          OpenDataCensusScore2017
##    <chr>                           <dbl>
##  1 san francisco                    1845
##  2 las vegas                        1830
##  3 new york city                    1740
##  4 los angeles                      1710
##  5 chicago                          1655
##  6 philadelphia                     1595
##  7 santa monica                     1560
##  8 anchorage                        1430
##  9 baton rouge                      1425
## 10 seattle                          1410
## # ... with 104 more rows
```