Analyses

Trini et al

2023-07-06

Contents

This file will analyze street tree data at neighborhood level

setup + loading packages

```r
# load packages

library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(sf) #simple features - vector GIS support
```

```
## Linking to GEOS 3.11.2, GDAL 3.6.2, PROJ 9.2.0; sf_use_s2() is TRUE
```

```r
library(mapview) #quick web maps
```

```
## The legacy packages maptools, rgdal, and rgeos, underpinning this package
## will retire shortly. Please refer to R-spatial evolution reports on
## https://r-spatial.org/r/2023/05/15/evolution4.html for details.
## This package is now running under evolution status 0
```

```r
library(tidylog) #makes dplyr and tidyr verbose
```

```
##
## Attaching package: 'tidylog'
##
## The following objects are masked from 'package:dplyr':
##
##     add_count, add_tally, anti_join, count, distinct, distinct_all,
##     distinct_at, distinct_if, filter, filter_all, filter_at, filter_if,
##     full_join, group_by, group_by_all, group_by_at, group_by_if,
##     inner_join, left_join, mutate, mutate_all, mutate_at, mutate_if,
##     relocate, rename, rename_all, rename_at, rename_if, rename_with,
##     right_join, sample_frac, sample_n, select, select_all, select_at,
##     select_if, semi_join, slice, slice_head, slice_max, slice_min,
##     slice_sample, slice_tail, summarise, summarise_all, summarise_at,
##     summarise_if, summarize, summarize_all, summarize_at, summarize_if,
##     tally, top_frac, top_n, transmute, transmute_all, transmute_at,
##     transmute_if, ungroup
##
## The following objects are masked from 'package:tidyr':
##
##     drop_na, fill, gather, pivot_longer, pivot_wider, replace_na,
##     spread, uncount
##
## The following object is masked from 'package:stats':
##
##     filter
```

```r
library(dplyr)
library(vegan)
```

```
## Loading required package: permute
## Loading required package: lattice
## This is vegan 2.6-4
```

```r
library(codyn)
library(ggplot2)

#set wd
# run baltimore neighborhoods to get clean file
getwd()
```

```
## [1] "C:/Users/trini/Documents/Street Trees/BaltimoreTrees/BaltimoreStreetTreeProject/scripts"
```

```r
list.files()
```

```
## [1] "Baltimore neighborhoods.Rmd"
## [2] "First-Analyses.pdf"
## [3] "First-Analyses.Rmd"
## [4] "First Analyses.Rmd"
## [5] "OLD Check_street_tree_data_versions.Rmd"
## [6] "temporary_land_cover_reconciler.R"
```

```r
list.files('../../BaltimoreStreetTreeProject_Large_Data/') # means "up one level"
```

```
## [1] "bc_forestry_trees_20190319"
## [2] "bc_forestry_trees_20190319.xlsx"
## [3] "Neighborhood"
## [4] "neighs.csv"
## [5] "street_trees_with_neigh_attributes_2023-06-13.csv"
## [6] "street_trees_with_neigh_attributes_2023-06-15.csv"
## [7] "street_trees_with_neigh_attributes_2023-06-21.csv"
## [8] "street_trees_with_neigh_attributes_2023-07-06.csv"
```

```r
list.files('../../BaltimoreStreetTreeProject_Large_Data/', recursive = TRUE) # means "up one level"
```

```
##  [1] "bc_forestry_trees_20190319.xlsx"
##  [2] "bc_forestry_trees_20190319/bc_forestry_trees_20190319.cpg"
##  [3] "bc_forestry_trees_20190319/bc_forestry_trees_20190319.dbf"
##  [4] "bc_forestry_trees_20190319/bc_forestry_trees_20190319.prj"
##  [5] "bc_forestry_trees_20190319/bc_forestry_trees_20190319.sbn"
##  [6] "bc_forestry_trees_20190319/bc_forestry_trees_20190319.sbx"
##  [7] "bc_forestry_trees_20190319/bc_forestry_trees_20190319.shp"
##  [8] "bc_forestry_trees_20190319/bc_forestry_trees_20190319.shp.xml"
##  [9] "bc_forestry_trees_20190319/bc_forestry_trees_20190319.shx"
## [10] "Neighborhood/Neighborhood.cpg"
## [11] "Neighborhood/Neighborhood.dbf"
## [12] "Neighborhood/Neighborhood.geojson"
## [13] "Neighborhood/Neighborhood.prj"
## [14] "Neighborhood/Neighborhood.shp"
## [15] "Neighborhood/Neighborhood.shx"
## [16] "Neighborhood/Neighborhood.xml"
## [17] "neighs.csv"
## [18] "street_trees_with_neigh_attributes_2023-06-13.csv"
## [19] "street_trees_with_neigh_attributes_2023-06-15.csv"
## [20] "street_trees_with_neigh_attributes_2023-06-21.csv"
## [21] "street_trees_with_neigh_attributes_2023-07-06.csv"
```

#make clean

```r
list.files('../../BaltimoreStreetTreeProject_Large_Data')
```

```
## [1] "bc_forestry_trees_20190319"
## [2] "bc_forestry_trees_20190319.xlsx"
## [3] "Neighborhood"
```

```
## [4] "neighs.csv"
## [5] "street_trees_with_neigh_attributes_2023-06-13.csv"
## [6] "street_trees_with_neigh_attributes_2023-06-15.csv"
## [7] "street_trees_with_neigh_attributes_2023-06-21.csv"
## [8] "street_trees_with_neigh_attributes_2023-07-06.csv"
```

```
clean <-
  read_csv('../../BaltimoreStreetTreeProject_Large_Data/street_trees_with_neigh_attributes_2023-06-13.
  #
select(-ID,-OnStr, -OBSERV2, -OBSERV3, -LOC_TYPE, -HARD_SCAPE, -PARCELID, -COLLECTOR, -Inv_Time, -INSPEC
  rename(Neighborhood = Name)
```

```
## Rows: 163574 Columns: 51
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (25): UniqueID, Street, OnStr, Side, SPP, MULTI_STEM, CONDITION, MT, OB...
## dbl  (22): OBJECTID.x, ID, Address, DBH, TREE_HT, X_COORD, Y_COORD, OBJECTID...
## date  (2): Inv_Date, INSPECT_DT
## time  (2): Inv_Time, Inspect_TM
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## select: dropped 13 variables (ID, OnStr, OBSERV2, OBSERV3, LOC_TYPE, ...)
##
## rename: renamed one variable (Neighborhood)
```

#filter tree data for only present tree species, create NB demographics data frame

```
data<- clean %>%
  mutate(Species=ifelse(SPP=='Acer spp.'|SPP=='Carya spp.'|SPP=='Cornus spp.'|SPP=='Cornus x'|SPP=='Fic
  filter(Species!='Vacant Site'&Species!='Vacant Potential'&Species!='Stump'&Species!='Vacant Site Not S
  filter(CONDITION!="Dead"&CONDITION!="Stump"&CONDITION!="Sprout")%>%
  mutate(present=1) %>% #for each tree make a present column with 1
  filter(Species!= "unknown tree")
```

```
## mutate: new variable 'Species' (character) with 259 unique values and <1% NA

## filter: removed 69,898 rows (43%), 93,676 rows remaining

## filter: removed 2,127 rows (2%), 91,549 rows remaining

## mutate: new variable 'present' (double) with one unique value and 0% NA

## filter: removed 110 rows (<1%), 91,439 rows remaining
```

```
#neighborhood demographics data frame
NBdem<- data%>%
  mutate(PercentWhite = White/Population*100) %>% #converting race data into percentages
  mutate(PercentBlk = Blk_AfAm/Population*100) %>%
  mutate(PercentNtv = AmInd_AkNa/Population*100) %>%
  mutate(PercentAsian = Asian/Population*100) %>%
```

```
  mutate(PercentNatHaw_Pac = NatHaw_Pac/Population*100) %>%
  mutate(PercentLat_Hisp = Hisp_Lat/Population*100) %>%
  mutate(PercentOther_Race = Other_Race/Population*100) %>%
  select(Neighborhood, Population, PercentWhite, PercentBlk, PercentNtv, PercentAsian, PercentNatHaw_Pac
  group_by(Neighborhood) %>%
  unique()
```

```
## mutate: new variable 'PercentWhite' (double) with 270 unique values and <1% NA

## mutate: new variable 'PercentBlk' (double) with 270 unique values and <1% NA

## mutate: new variable 'PercentNtv' (double) with 217 unique values and <1% NA

## mutate: new variable 'PercentAsian' (double) with 246 unique values and <1% NA

## mutate: new variable 'PercentNatHaw_Pac' (double) with 93 unique values and <1% NA

## mutate: new variable 'PercentLat_Hisp' (double) with 269 unique values and <1% NA

## mutate: new variable 'PercentOther_Race' (double) with 263 unique values and <1% NA

## select: dropped 38 variables (OBJECTID.x, UniqueID, Address, Street, Side, ...)

## group_by: one grouping variable (Neighborhood)
```

```
NBdem |> head()
```

```
## # A tibble: 6 x 9
## # Groups:   Neighborhood [6]
##   Neighborhood     Population PercentWhite PercentBlk PercentNtv PercentAsian
##   <chr>                 <dbl>        <dbl>      <dbl>      <dbl>        <dbl>
## 1 Cross Country          5305         72.3       19.3     0.0754        0.697
## 2 Fallstaff              4896         26.6       47.4     0.429         1.04
## 3 <NA>                     NA         NA         NA      NA            NA
## 4 Glen                   7766         37.1       53.5     0.180         1.16
## 5 Cheswolde              7613         68.4       21.2     0.105         3.40
## 6 Mount Washington       4090         68.4       18.0     0.220         3.23
## # i 3 more variables: PercentNatHaw_Pac <dbl>, PercentLat_Hisp <dbl>,
## #   PercentOther_Race <dbl>
```

#For each NB sum the number of trees by species (abundance)

```
NBabund <-data %>%
  group_by(Neighborhood, Species) %>%
  summarize(abund=sum(present))
```

```
## group_by: 2 grouping variables (Neighborhood, Species)

## summarize: now 9,767 rows and 3 columns, one group variable remaining (Neighborhood)
```

#calculate richness and evenness

```r
richeven <-community_structure(NBabund, abundance.var="abund", time.var= NULL, replicate.var = "Neighbo
 left_join(NBdem)
```

```
## Warning in community_structure(NBabund, abundance.var = "abund", time.var =
## NULL, : Evenness values contain NAs because there are plots with only one
## species

## Joining with `by = join_by(Neighborhood)`
## left_join: added 8 columns (Population, PercentWhite, PercentBlk, PercentNtv,
## PercentAsian, ...)
## > rows only in x 0
## > rows only in y ( 1)
## > matched rows 277
## > =====
## > rows total 277
```

```r
richeven |> head()
```

```
##        Neighborhood richness      Evar Population PercentWhite PercentBlk
## 1             Abell       60 0.4258009        975    64.820513   17.33333
## 2         Allendale       51 0.5536432       3354     1.282051   95.67680
## 3           Arcadia       41 0.4112469       1077    49.582173   40.48282
## 4         Arlington       32 0.4889207       2163     2.404068   92.04808
## 5 Armistead Gardens       19 0.6048929       3779    46.652554    4.18100
## 6         Ashburton       60 0.3281453       2243     2.407490   91.08337
##   PercentNtv PercentAsian PercentNatHaw_Pac PercentLat_Hisp PercentOther_Race
## 1  0.0000000    7.4871795        0.00000000        5.333333         1.1282051
## 2  0.2981515    0.1490757        0.02981515        1.162791         0.8944544
## 3  0.4642526    1.2070566        0.00000000        6.035283         1.7641597
## 4  0.2773925    1.2020342        0.09246417        1.941748         0.6010171
## 5  0.8467849    1.0849431        0.05292405       50.568934        33.9243186
## 6  0.1337494    0.5795809        0.13374944        2.674989         1.3374944
```

#vacant tree sites

```r
#filter for vacant sites by neighborhood

Vacant<-clean %>%
  filter(SPP=='Vacant Site'|SPP=='Vacant Potential'|SPP=='Stump'|SPP=='NA'|SPP=='unknown shrub'|SPP=='Z
  mutate(present = 1) %>%
  select(Neighborhood, SPP, present) %>%
  group_by(Neighborhood) %>%
  unique()
```

```
## filter: removed 97,528 rows (60%), 66,046 rows remaining

## mutate: new variable 'present' (double) with one unique value and 0% NA

## select: dropped 36 variables (OBJECTID.x, UniqueID, Address, Street, Side, ...)

## group_by: one grouping variable (Neighborhood)
```

```r
#number of vacant sites per neighborhood
NBvacant <-Vacant %>%
  group_by(Neighborhood) %>%
  summarize(vacant=sum(present))
```

```
## group_by: one grouping variable (Neighborhood)
```

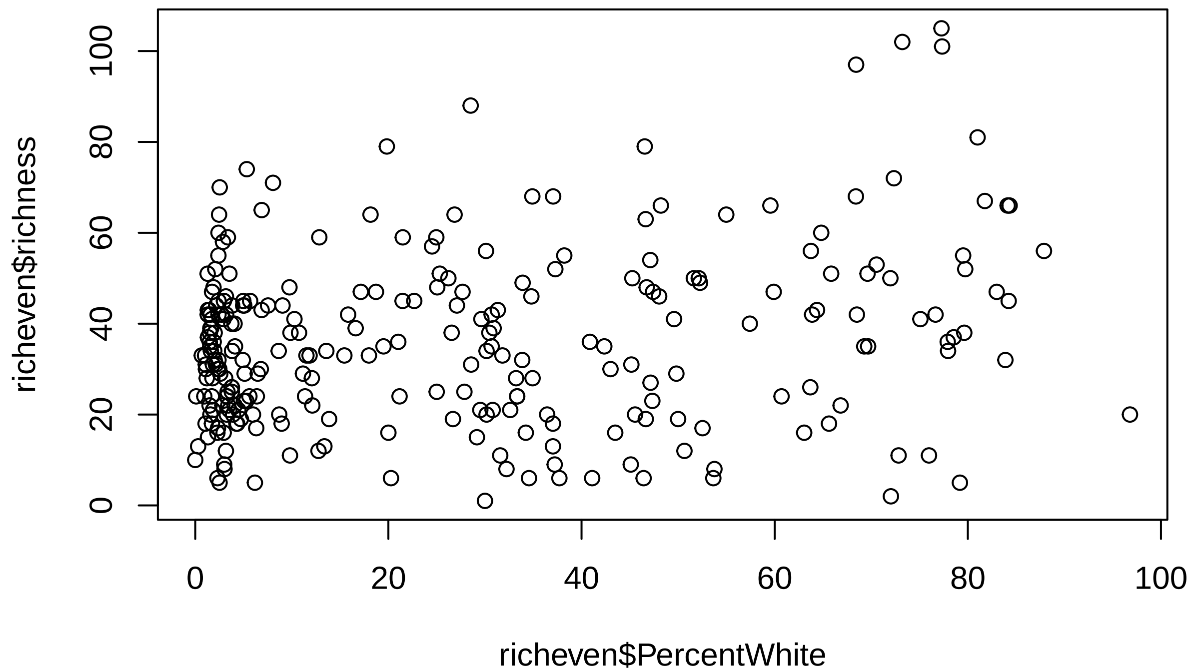
```
## summarize: now 275 rows and 2 columns, ungrouped
```

some quick correlation analyses

```r
#percent white vs richness
cor.test(richeven$PercentWhite, richeven$richness)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richeven$PercentWhite and richeven$richness
## t = 4.0248, df = 268, p-value = 7.426e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1228690 0.3481954
## sample estimates:
##       cor
## 0.2387431
```

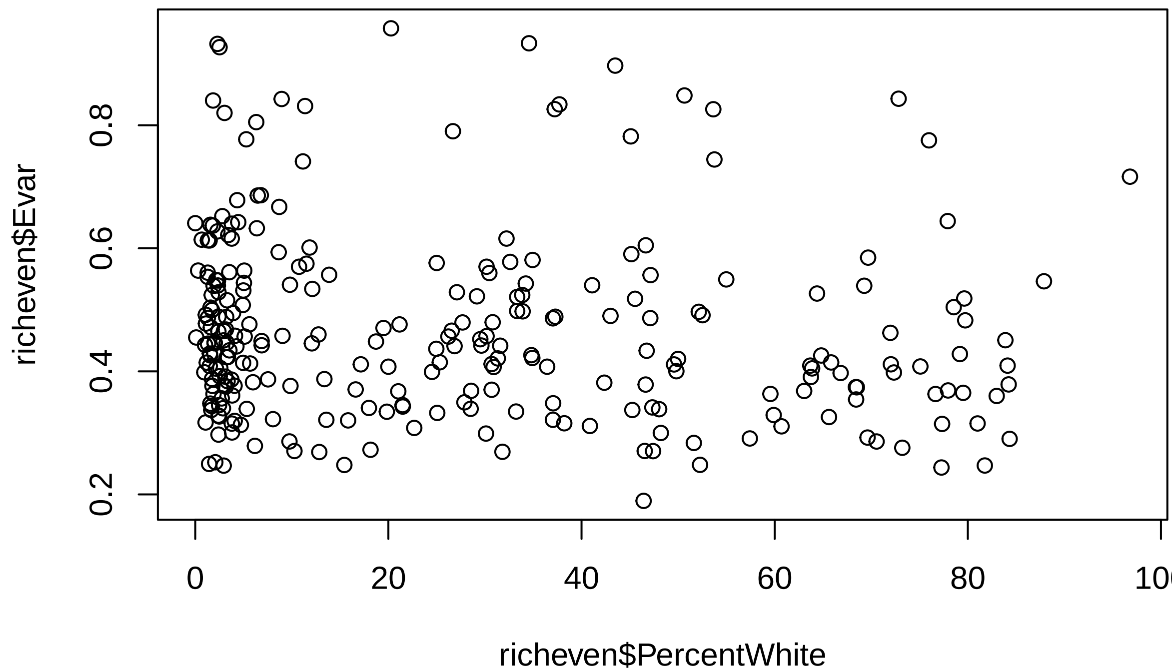
```r
plot(richeven$PercentWhite, richeven$richness)
```




```r
#percent white vs evenness
cor.test(richeven$PercentWhite, richeven$Evar)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richeven$PercentWhite and richeven$Evar
## t = -1.7618, df = 267, p-value = 0.07924
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.22392713  0.01255879
## sample estimates:
##        cor
## -0.1072002
```

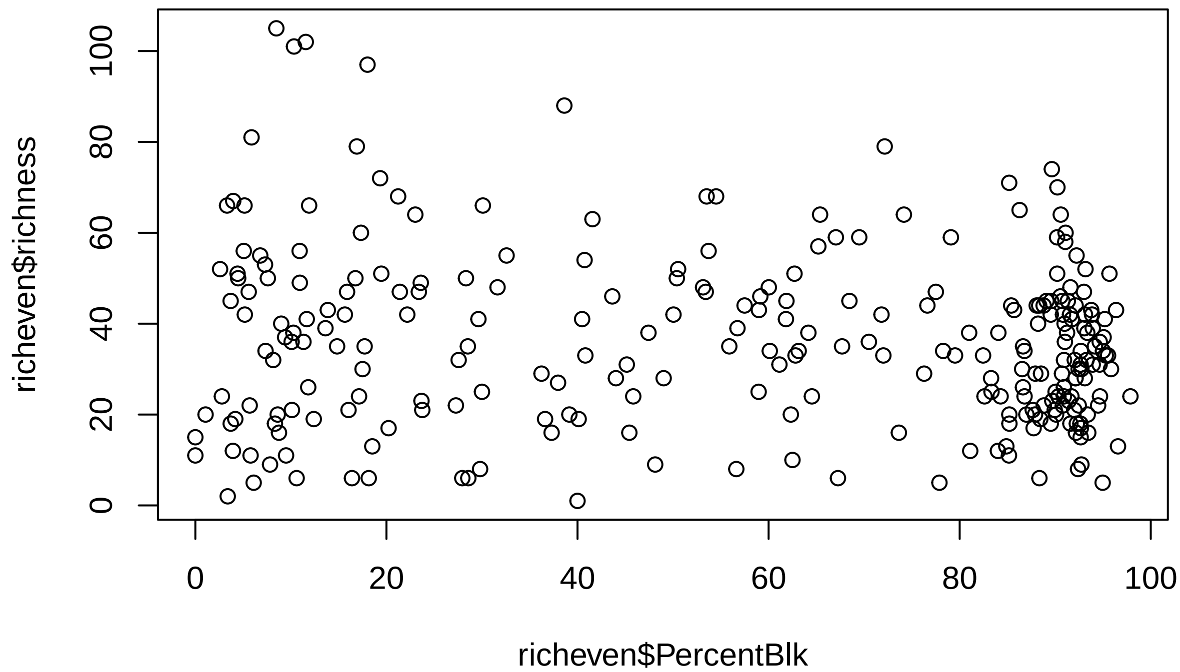
```r
plot(richeven$PercentWhite, richeven$Evar)
```




```
#percent black richness
cor.test(richeven$PercentBlk, richeven$richness)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richeven$PercentBlk and richeven$richness
## t = -2.0912, df = 268, p-value = 0.03745
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.242419476 -0.007447224
## sample estimates:
##        cor
## -0.1267105
```

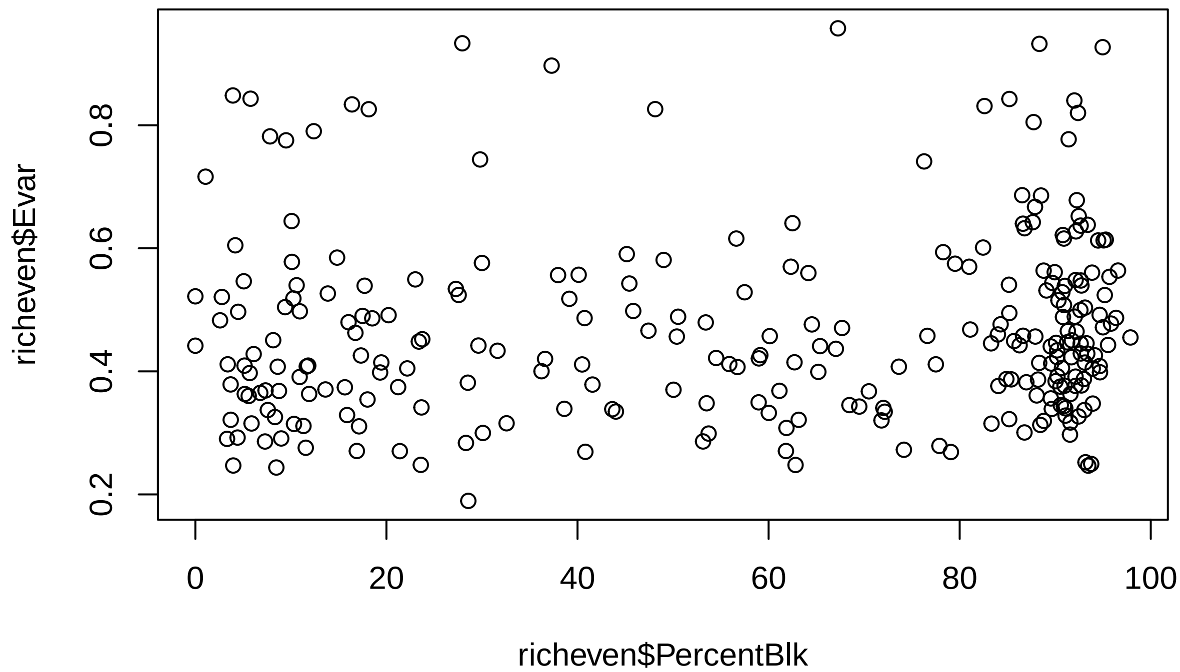
```
plot(richeven$PercentBlk, richeven$richness)
```




```
#percent black evenness
cor.test(richeven$PercentBlk, richeven$Evar)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richeven$PercentBlk and richeven$Evar
## t = 1.2785, df = 267, p-value = 0.2022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.04198287  0.19577816
## sample estimates:
##        cor
## 0.07800673
```

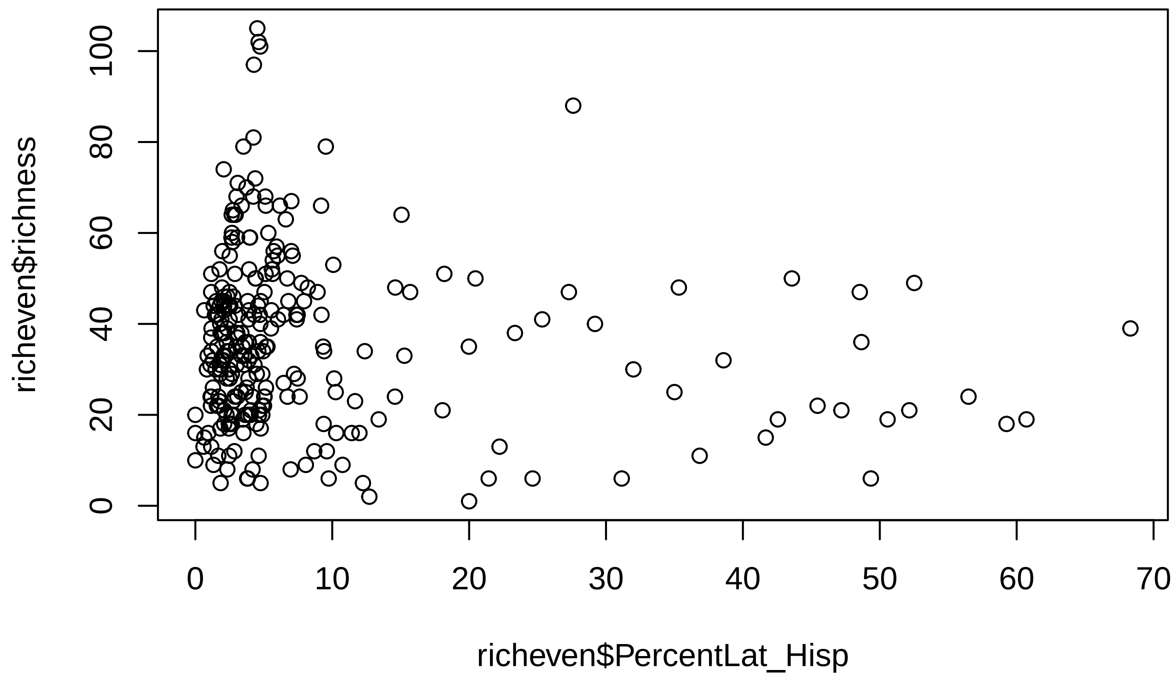
```
plot(richeven$PercentBlk, richeven$Evar)
```




```r
# #percent native american and alaska native
# cor.test(richeven$PercentNtv, richeven$richness)
# plot(richeven$PercentNtv, richeven$richness)

#percent Latino and Hispanic richness
cor.test(richeven$PercentLat_Hisp, richeven$richness)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richeven$PercentLat_Hisp and richeven$richness
## t = -2.1024, df = 268, p-value = 0.03645
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.243059028 -0.008126777
## sample estimates:
##        cor
## -0.1273791
```

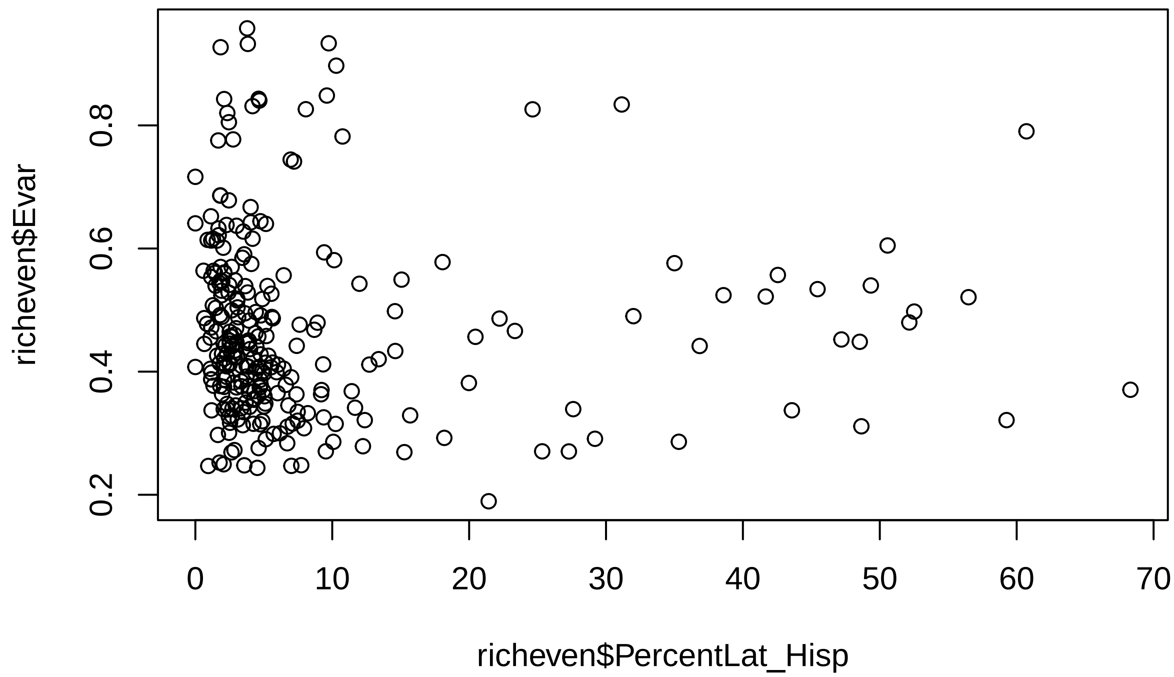
```r
plot(richeven$PercentLat_Hisp, richeven$richness)
```




```r
#percent Latino and Hispanic evenness
cor.test(richeven$PercentLat_Hisp, richeven$Evar)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richeven$PercentLat_Hisp and richeven$Evar
## t = -0.15254, df = 267, p-value = 0.8789
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1287891  0.1103862
## sample estimates:
##          cor
## -0.009334981
```

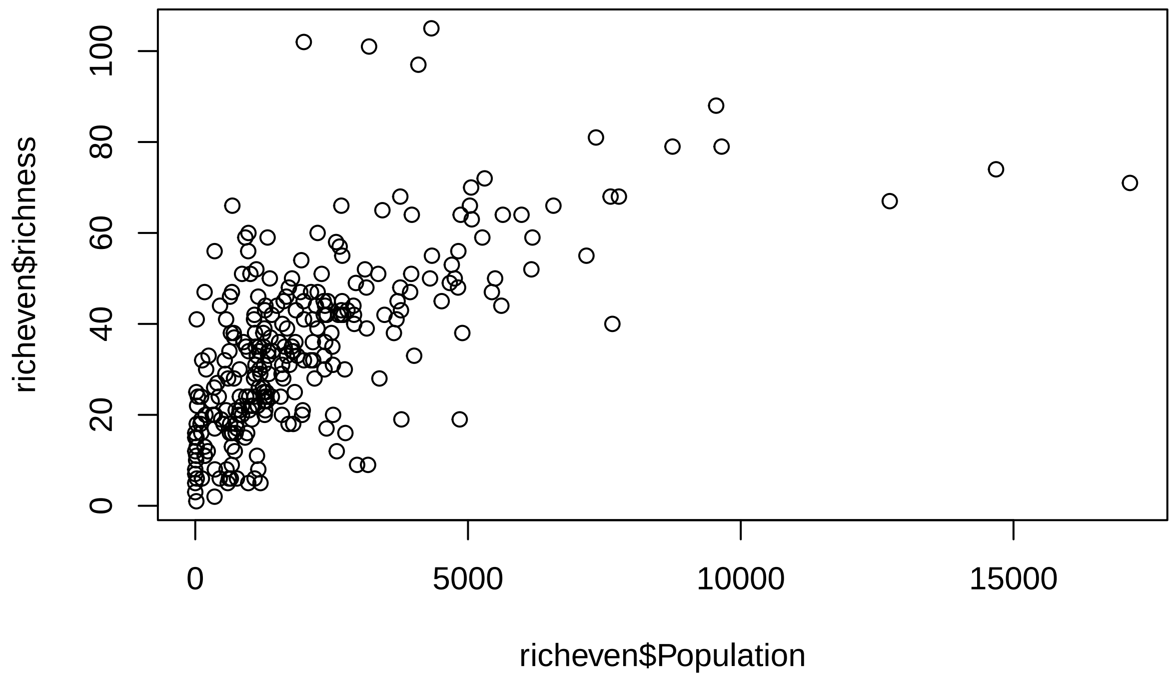
```r
plot(richeven$PercentLat_Hisp, richeven$Evar)
```




```r
#population vs richness
cor.test(richeven$Population, richeven$richness)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richeven$Population and richeven$richness
## t = 13.539, df = 275, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5560038 0.6982316
## sample estimates:
##       cor
## 0.6324181
```

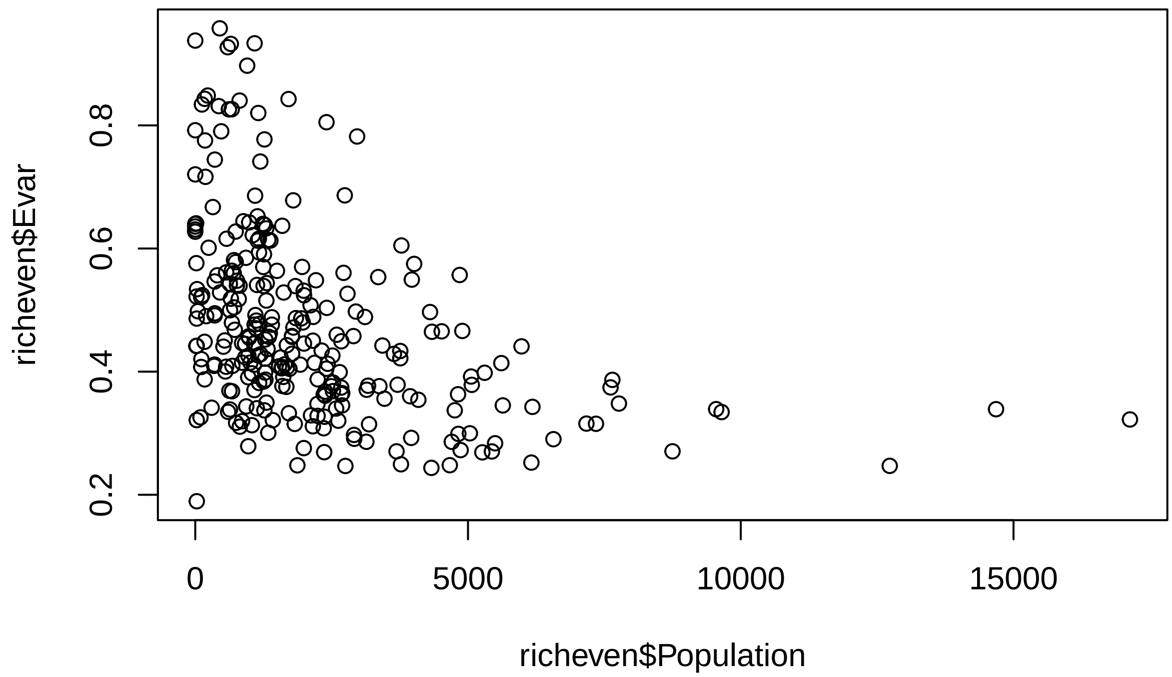
```r
plot(richeven$Population, richeven$richness)
```




```
#population vs evenness
cor.test(richeven$Population, richeven$Evar)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richeven$Population and richeven$Evar
## t = -7.3553, df = 274, p-value = 2.215e-12
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.500155 -0.302499
## sample estimates:
##        cor
## -0.4060652
```

```
plot(richeven$Population, richeven$Evar)
```



#DBH - classify trees by size

```
DBHclass<-data%>%
  mutate(size_class=ifelse(DBH>=0&DBH<=5, "small", ifelse(DBH>5&DBH<20, "medium", ifelse(DBH>=20, "larg
  group_by(Neighborhood, size_class) %>%
  summarize(n=length(DBH)) %>%
  pivot_wider(names_from = size_class, values_from = n, values_fill = 0)
```

```
## mutate: new variable 'size_class' (character) with 3 unique values and 0% NA

## group_by: 2 grouping variables (Neighborhood, size_class)

## summarize: now 815 rows and 3 columns, one group variable remaining (Neighborhood)

## pivot_wider: reorganized (size_class, n) into (large, medium, small) [was 815x3, now 278x4]
```

```
#joining with neighborhood demographics
DBHdem <- DBHclass %>%
  left_join(NBdem)
```

```
## Joining with 'by = join_by(Neighborhood)'

## left_join: added 8 columns (Population, PercentWhite, PercentBlk, PercentNtv, PercentAsian, ...)
```

```
##              > rows only in x     0
##              > rows only in y  (   0)
##              > matched rows     278
##              >                 =====
##              > rows total       278
```

#average DBH by neighborhood

```r
NBDBHAvg <- data %>%
  select(Neighborhood, DBH) %>%
  group_by(Neighborhood) %>%
  summarise(AvgDBH = mean(DBH), .groups = 'drop')
```

```
## select: dropped 38 variables (OBJECTID.x, UniqueID, Address, Street, Side, ...)
## group_by: one grouping variable (Neighborhood)
## summarise: now 278 rows and 2 columns, ungrouped
```

```r
AvgDBHdem <- NBDBHAvg %>%
  left_join(NBdem)
```

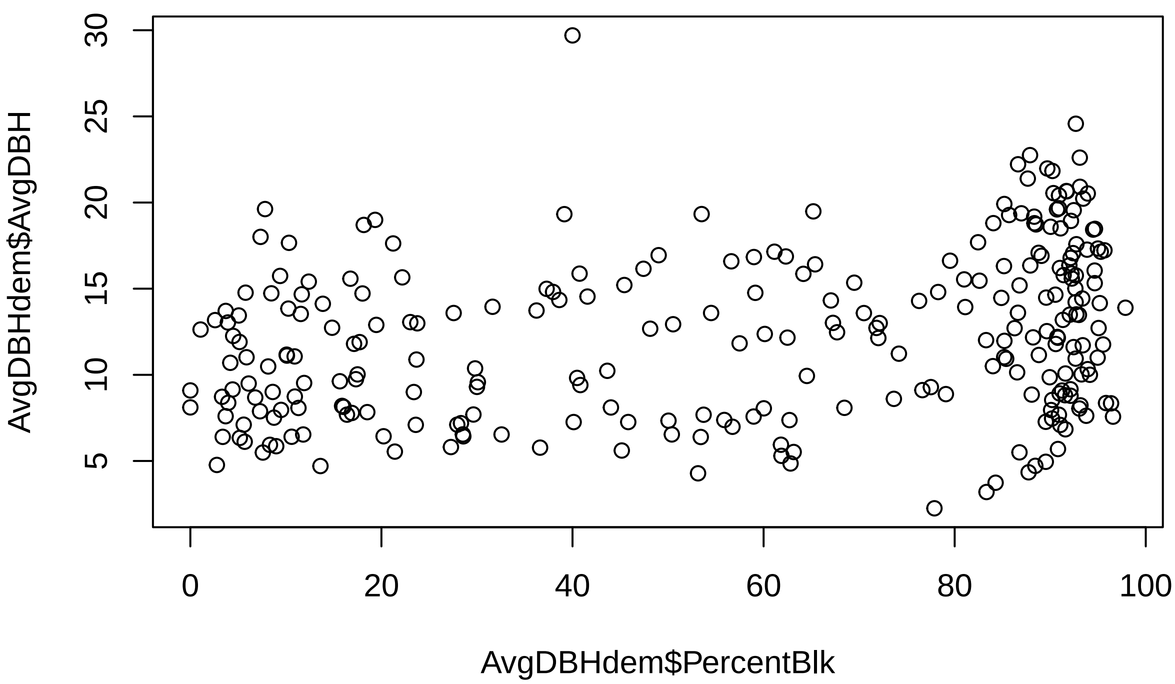
```
## Joining with `by = join_by(Neighborhood)`
## left_join: added 8 columns (Population, PercentWhite, PercentBlk, PercentNtv, PercentAsian, ...)
##              > rows only in x     0
##              > rows only in y  (   0)
##              > matched rows     278
##              >                 =====
##              > rows total       278
```

#correlations for DBH

```r
#percent black
cor.test(AvgDBHdem$PercentBlk,AvgDBHdem$AvgDBH)
```

```
##
##  Pearson's product-moment correlation
##
## data:  AvgDBHdem$PercentBlk and AvgDBHdem$AvgDBH
## t = 5.0437, df = 268, p-value = 8.425e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1814388 0.3997621
## sample estimates:
##       cor
## 0.2944373
```

```r
plot(AvgDBHdem$PercentBlk,AvgDBHdem$AvgDBH)
```

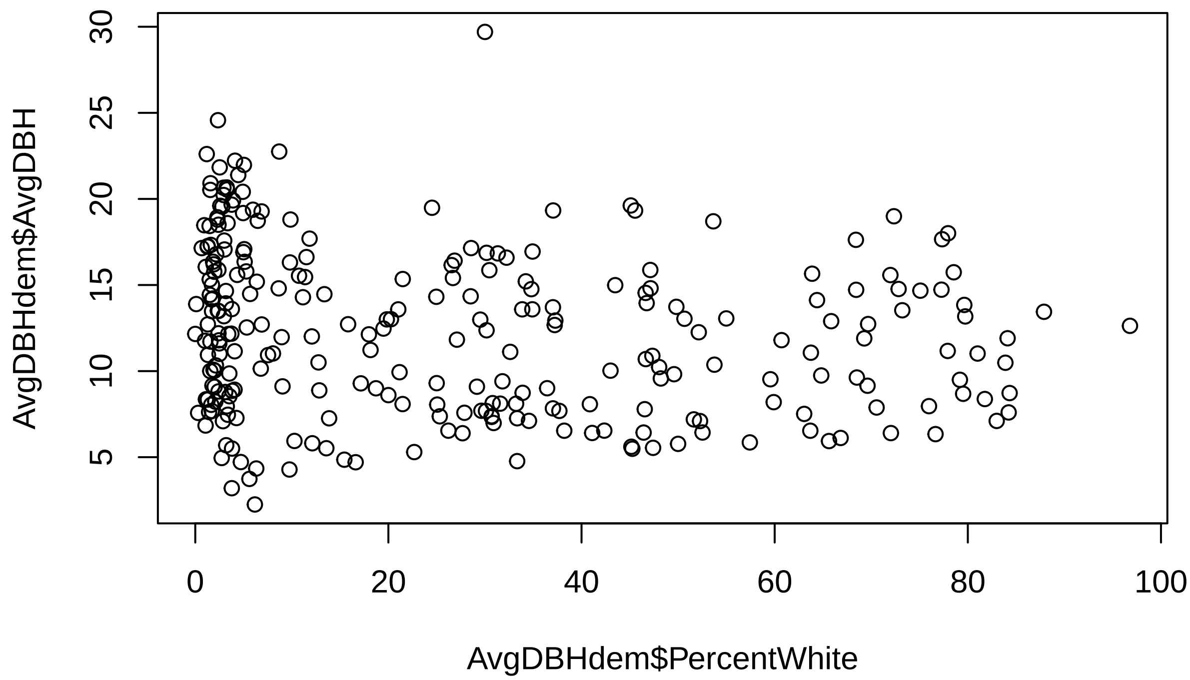



```r
cor.test(AvgDBHdem$PercentWhite,AvgDBHdem$AvgDBH)
```

```
##
##  Pearson's product-moment correlation
##
## data:  AvgDBHdem$PercentWhite and AvgDBHdem$AvgDBH
## t = -3.1756, df = 268, p-value = 0.00167
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.30291930 -0.07270651
## sample estimates:
```

```
##         cor
## -0.1904296
```

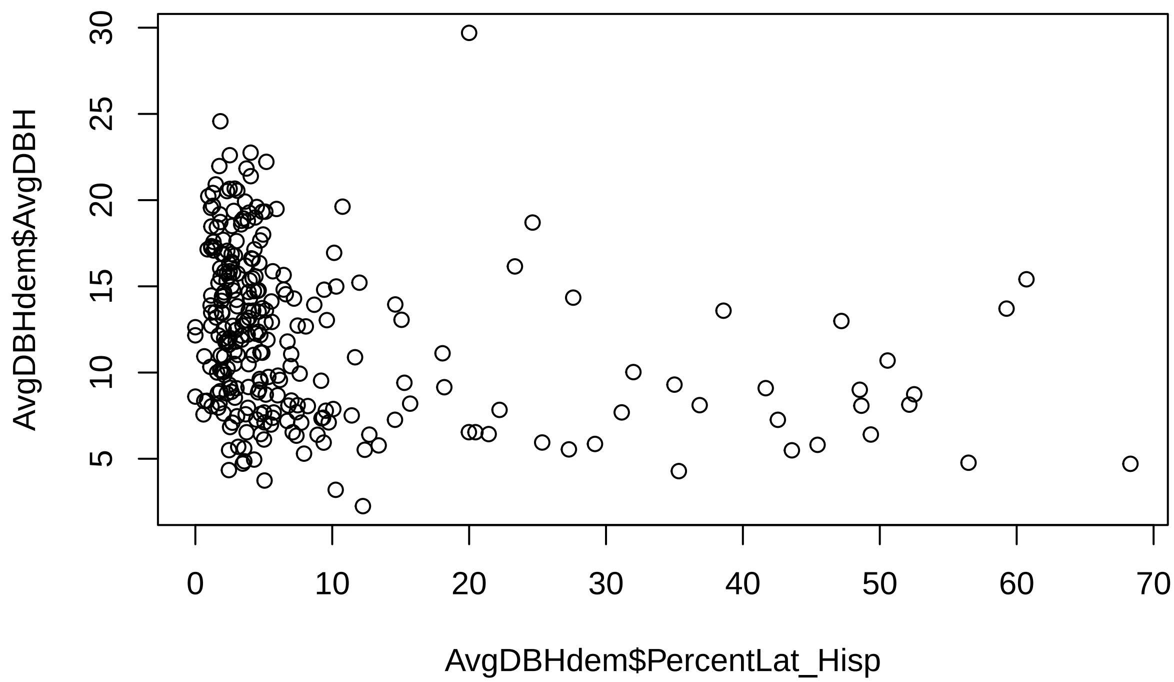
```
plot(AvgDBHdem$PercentWhite,AvgDBHdem$AvgDBH)
```




```
cor.test(AvgDBHdem$PercentLat_Hisp,AvgDBHdem$AvgDBH)
```

```
##
##  Pearson's product-moment correlation
##
## data:  AvgDBHdem$PercentLat_Hisp and AvgDBHdem$AvgDBH
## t = -4.822, df = 268, p-value = 2.385e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3888110 -0.1688695
## sample estimates:
##        cor
## -0.2825495
```

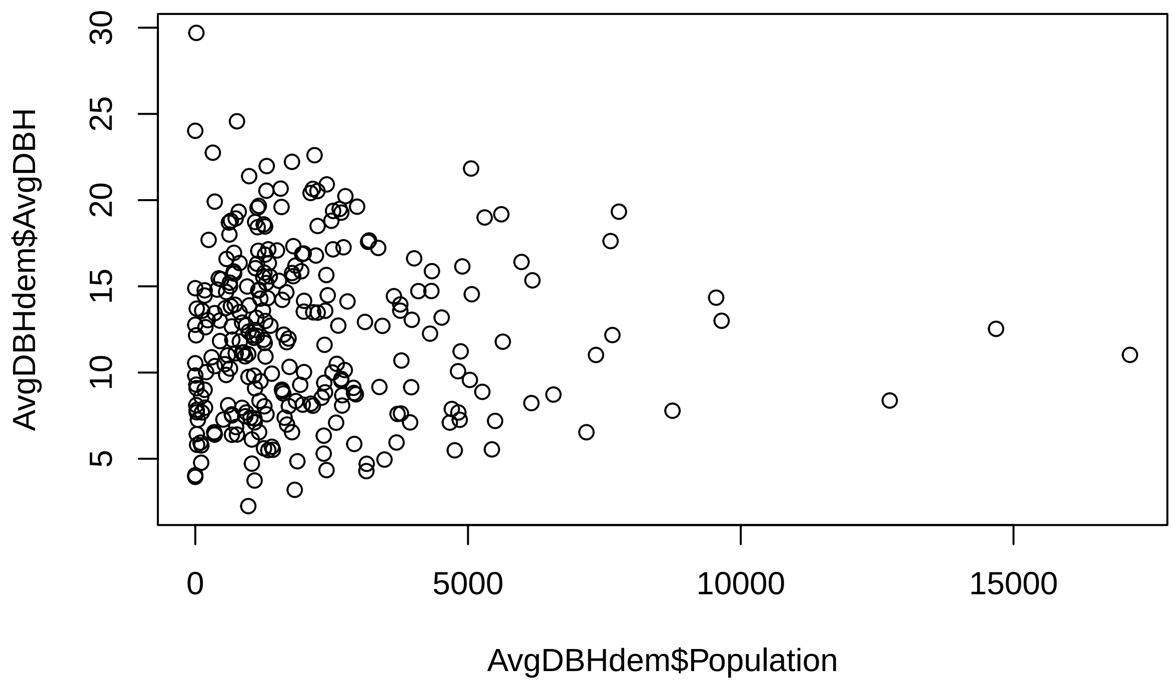
```
plot(AvgDBHdem$PercentLat_Hisp,AvgDBHdem$AvgDBH)
```




```
cor.test(AvgDBHdem$Population,AvgDBHdem$AvgDBH)
```

```
##
##  Pearson's product-moment correlation
##
## data:  AvgDBHdem$Population and AvgDBHdem$AvgDBH
## t = -0.23991, df = 275, p-value = 0.8106
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1320958  0.1035666
## sample estimates:
##         cor
## -0.01446548
```

```
plot(AvgDBHdem$Population,AvgDBHdem$AvgDBH)
```



#coefficient of variation

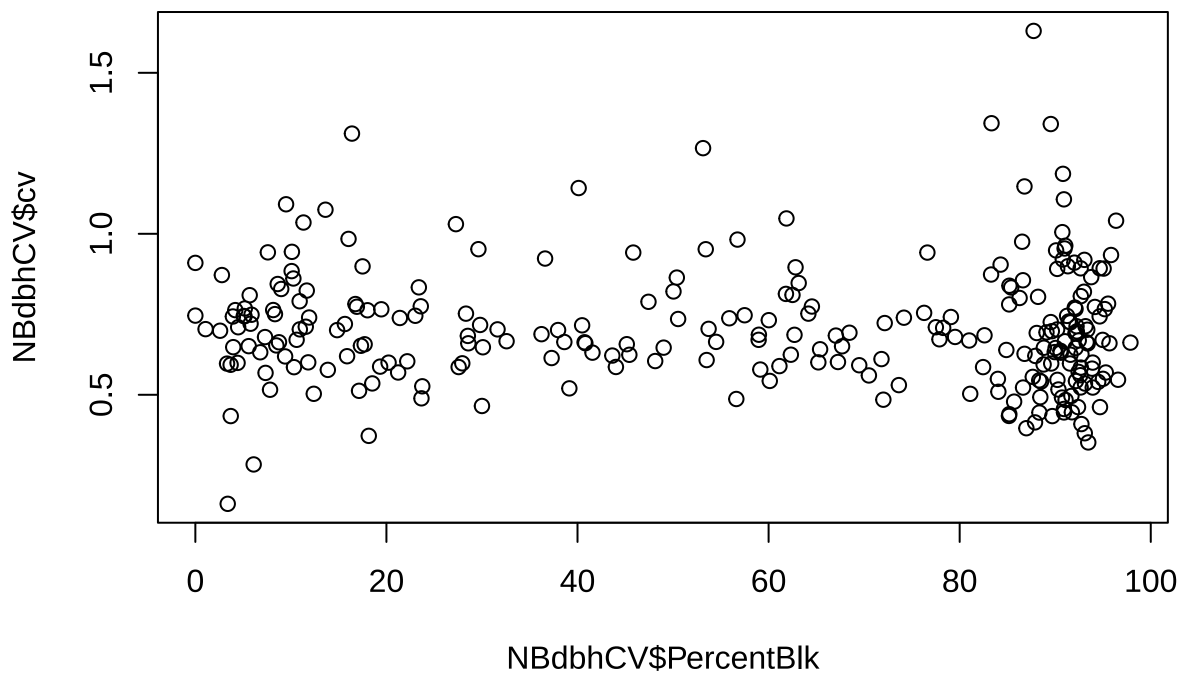
```
#coefficient of variation for all DBH measurements
NBdbhCV<- data %>%
  group_by(Neighborhood) %>%
  summarise(mean = mean(DBH),
            sd = sd(DBH)) %>%
  mutate(cv = sd/mean) %>%
  left_join(NBdem)
```

```
## group_by: one grouping variable (Neighborhood)

## summarise: now 278 rows and 3 columns, ungrouped

## mutate: new variable 'cv' (double) with 278 unique values and <1% NA

## Joining with `by = join_by(Neighborhood)`
## left_join: added 8 columns (Population, PercentWhite, PercentBlk, PercentNtv,
## PercentAsian, ...)
## > rows only in x 0
## > rows only in y ( 0)
## > matched rows 278
## > =====
## > rows total 278
```

```r
cor.test(NBdbhCV$PercentBlk,NBdbhCV$cv)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBdbhCV$PercentBlk and NBdbhCV$cv
## t = -0.67164, df = 267, p-value = 0.5024
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1598815  0.0789167
## sample estimates:
##         cor
## -0.04106885
```

```r
plot(NBdbhCV$PercentBlk,NBdbhCV$cv)
```

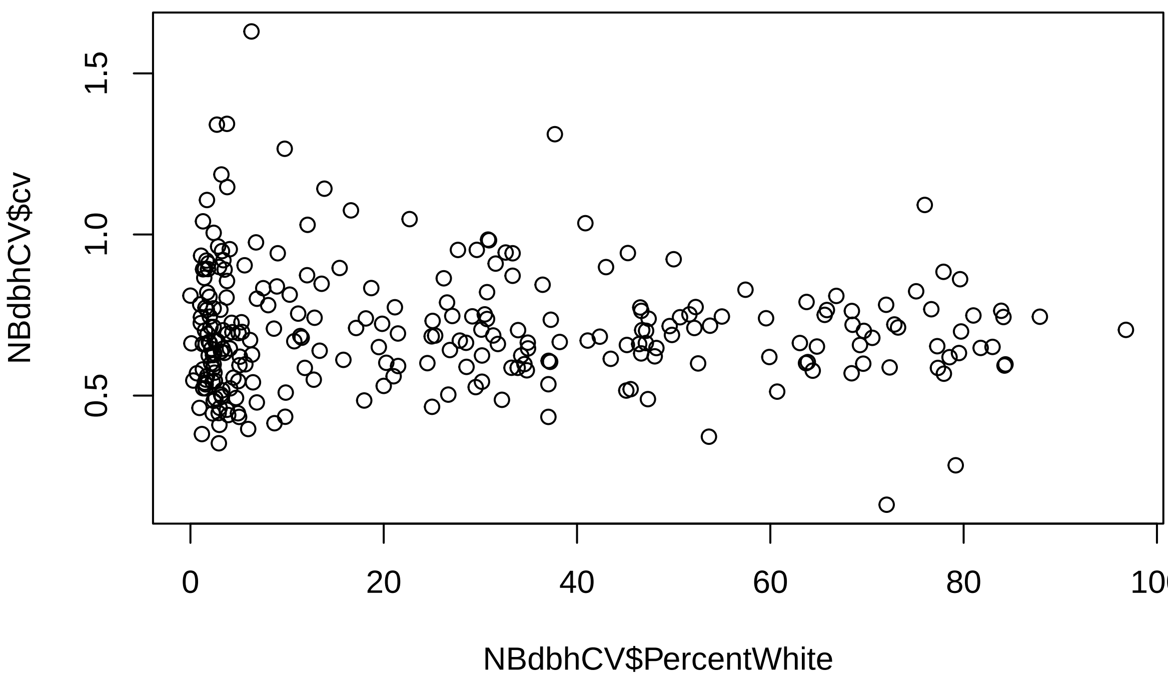



```r
cor.test(NBdbhCV$PercentWhite,NBdbhCV$cv)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBdbhCV$PercentWhite and NBdbhCV$cv
## t = -0.83853, df = 267, p-value = 0.4025
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  -0.16980688  0.06876968
## sample estimates:
##         cor
## -0.05124976
```

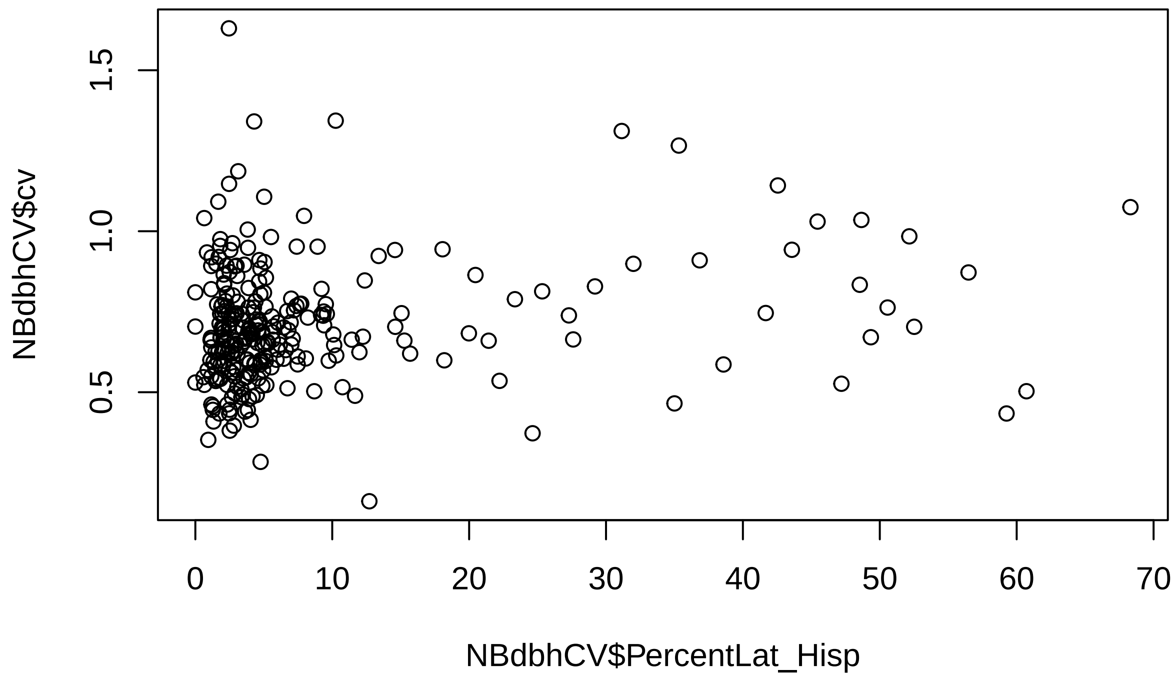
```r
plot(NBdbhCV$PercentWhite,NBdbhCV$cv)
```




```r
cor.test(NBdbhCV$PercentLat_Hisp,NBdbhCV$cv)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBdbhCV$PercentLat_Hisp and NBdbhCV$cv
## t = 3.0707, df = 267, p-value = 0.002356
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.06656223 0.29771181
## sample estimates:
##       cor
## 0.1846899
```

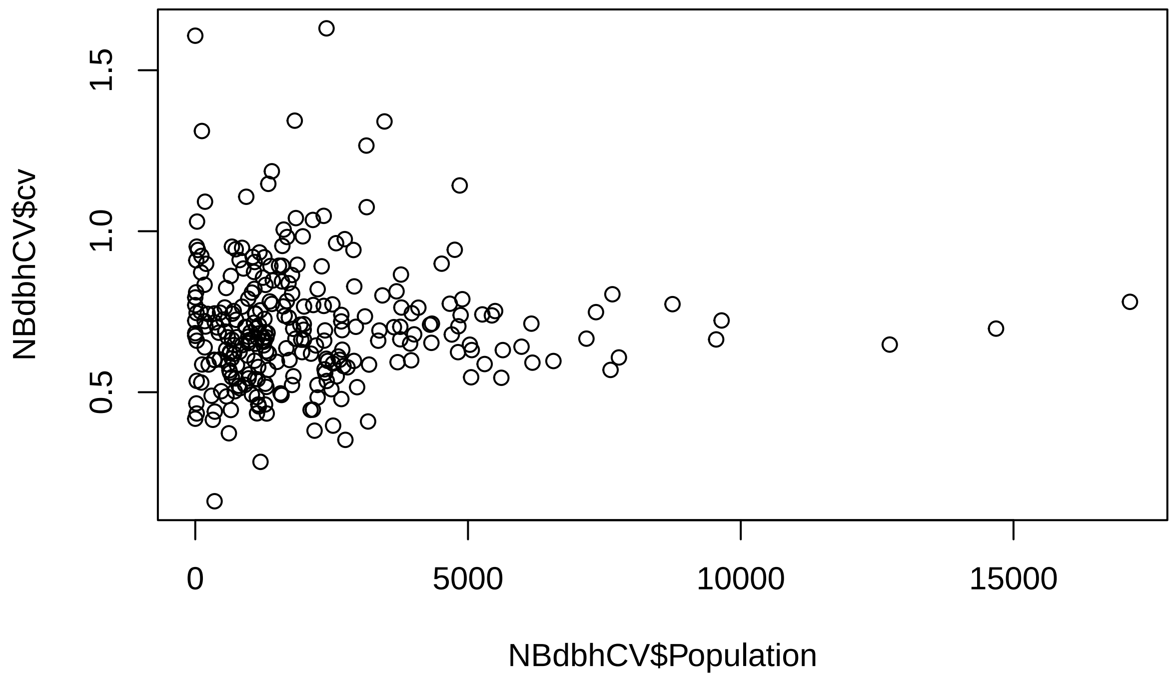
```r
plot(NBdbhCV$PercentLat_Hisp,NBdbhCV$cv)
```




```r
cor.test(NBdbhCV$Population,NBdbhCV$cv)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBdbhCV$Population and NBdbhCV$cv
## t = 0.15187, df = 274, p-value = 0.8794
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1090129  0.1271057
## sample estimates:
##         cor
## 0.009174282
```

```r
plot(NBdbhCV$Population,NBdbhCV$cv)
```



#Abundance, Richness, evenness for small trees

```
NBabundSM <-data %>%
  filter(DBH>=0&DBH<=5) %>%
  group_by(Neighborhood, Species) %>%
  summarize(abund=sum(present))
```

```
## filter: removed 61,718 rows (67%), 29,721 rows remaining
```

```
## group_by: 2 grouping variables (Neighborhood, Species)
```

```
## summarize: now 5,317 rows and 3 columns, one group variable remaining (Neighborhood)
```

```
#calculate richness and eveness for small trees
richevenSM<-community_structure(NBabundSM, abundance.var="abund", time.var= NULL, replicate.var = "Neigh
  left_join(NBdem) %>%
  unique()
```

```
## Warning in community_structure(NBabundSM, abundance.var = "abund", time.var =
## NULL, : Evenness values contain NAs because there are plots with only one
## species
```

```
## Joining with `by = join_by(Neighborhood)`
```

```
## left_join: added 8 columns (Population, PercentWhite, PercentBlk, PercentNtv, PercentAsian, ...)
```

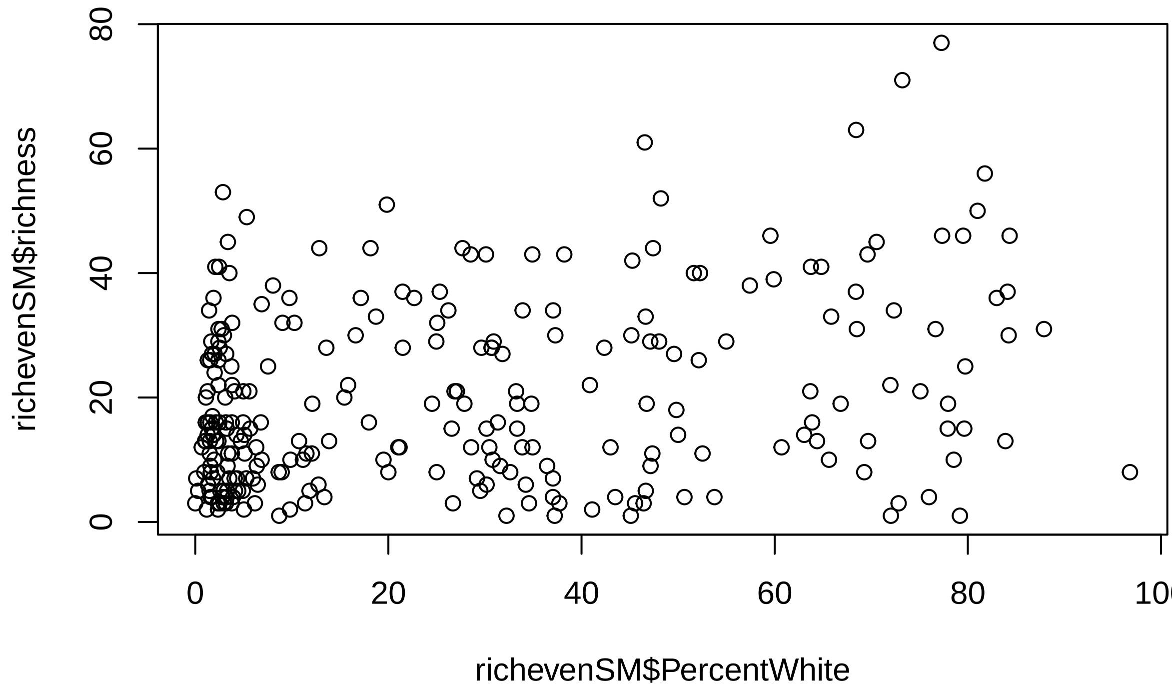
```
##            > rows only in x     0

##            > rows only in y  (  8)

##            > matched rows     270

##            >                 =====

##            > rows total       270
```

#Correlations for small trees

```r
#percent white vs richness
cor.test(richevenSM$PercentWhite, richevenSM$richness)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richevenSM$PercentWhite and richevenSM$richness
## t = 5.1106, df = 262, p-value = 6.199e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1871569 0.4070136
## sample estimates:
##      cor
## 0.301081
```

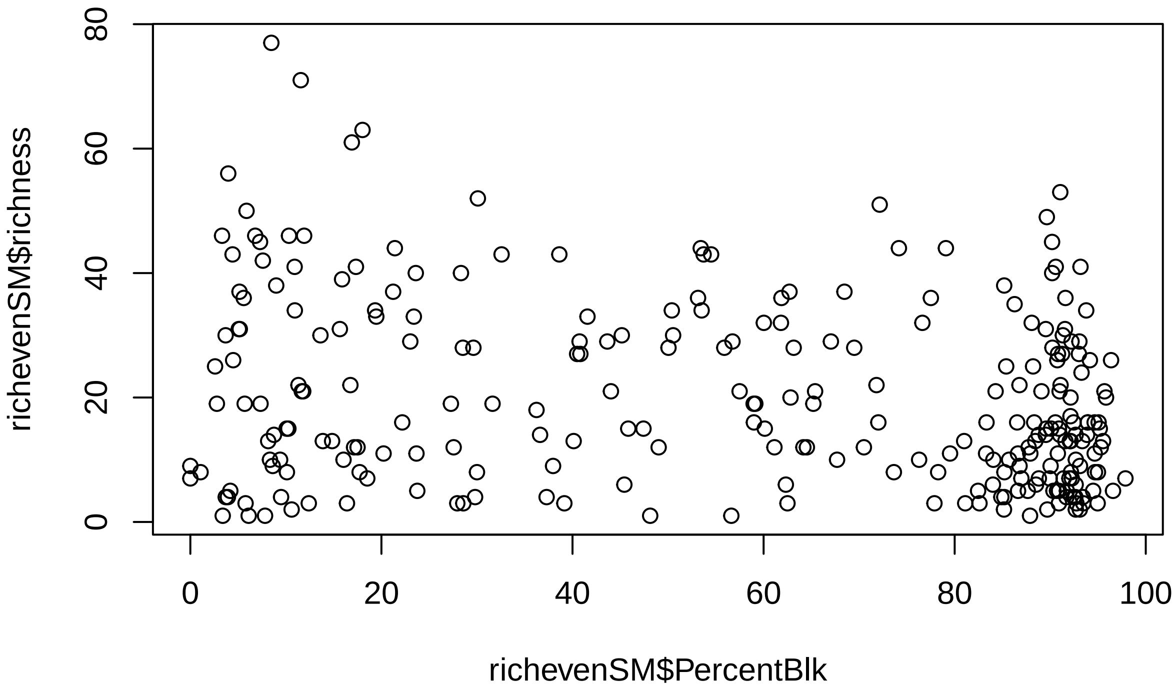
```r
plot(richevenSM$PercentWhite, richevenSM$richness)
```




```
#percent black
cor.test(richevenSM$PercentBlk, richevenSM$richness)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richevenSM$PercentBlk and richevenSM$richness
## t = -4.0208, df = 262, p-value = 7.589e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3515751 -0.1239616
## sample estimates:
##        cor
## -0.2410807
```

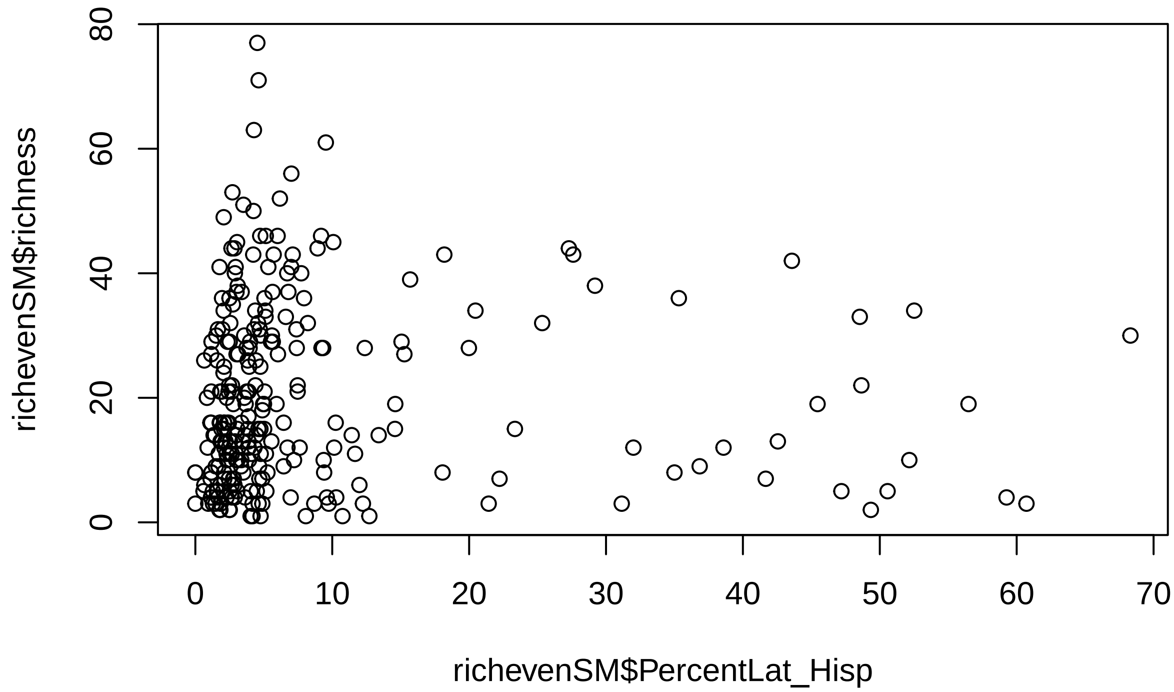
```
plot(richevenSM$PercentBlk, richevenSM$richness)
```




```r
#percent latino richness
cor.test(richevenSM$PercentLat_Hisp, richevenSM$richness)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richevenSM$PercentLat_Hisp and richevenSM$richness
## t = -0.14828, df = 262, p-value = 0.8822
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1297437  0.1116903
## sample estimates:
##          cor
## -0.009160183
```

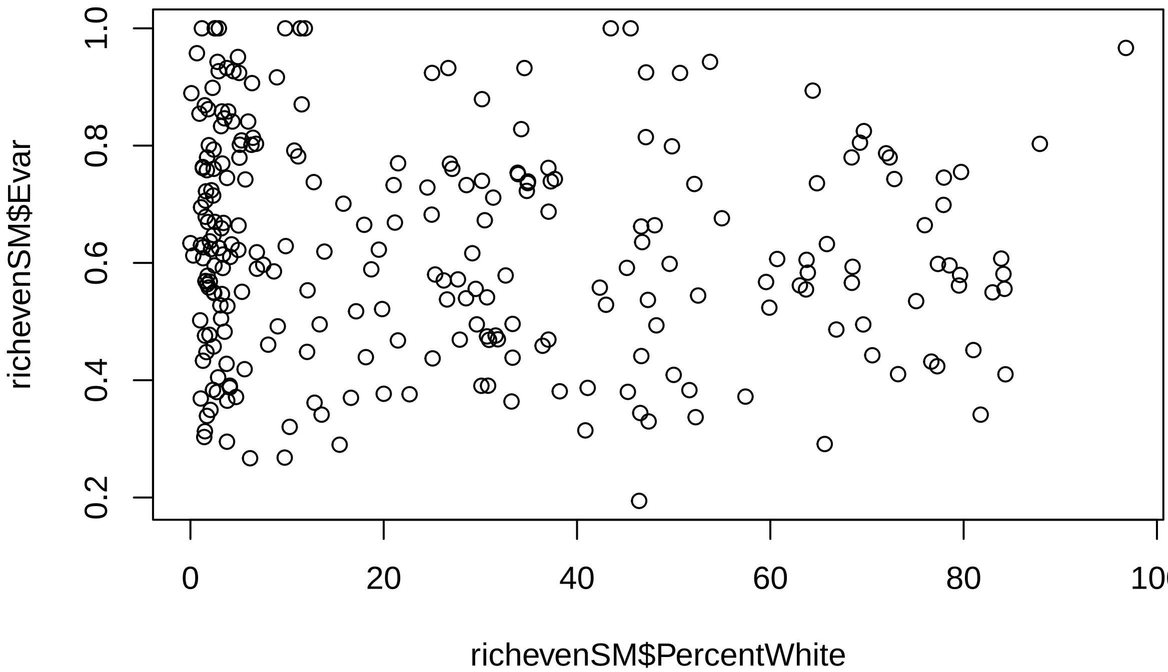
```r
plot(richevenSM$PercentLat_Hisp, richevenSM$richness)
```




```r
#percent white vs evenness
cor.test(richevenSM$Evar,richevenSM$PercentWhite)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richevenSM$Evar and richevenSM$PercentWhite
## t = -1.5793, df = 256, p-value = 0.1155
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.21774182  0.02418639
## sample estimates:
##         cor
## -0.09822883
```

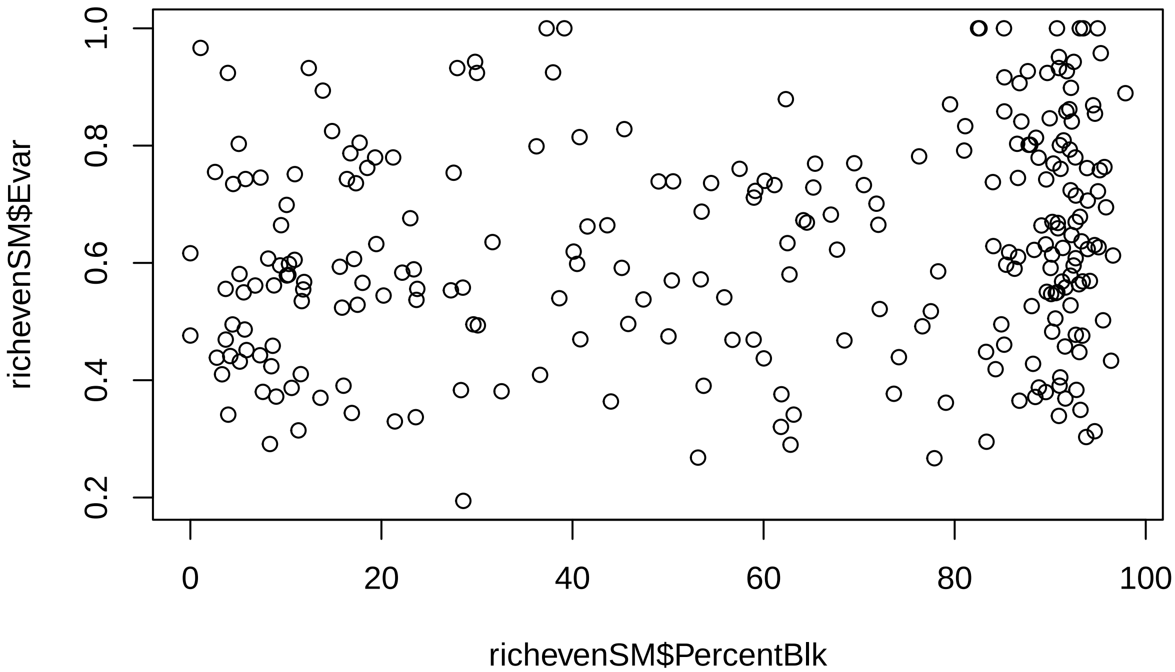
```r
plot(richevenSM$PercentWhite, richevenSM$Evar)
```




```r
#percent black
cor.test(richevenSM$PercentBlk, richevenSM$Evar)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richevenSM$PercentBlk and richevenSM$Evar
## t = 2.8842, df = 256, p-value = 0.004258
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.05650354 0.29317765
## sample estimates:
##       cor
## 0.1774044
```

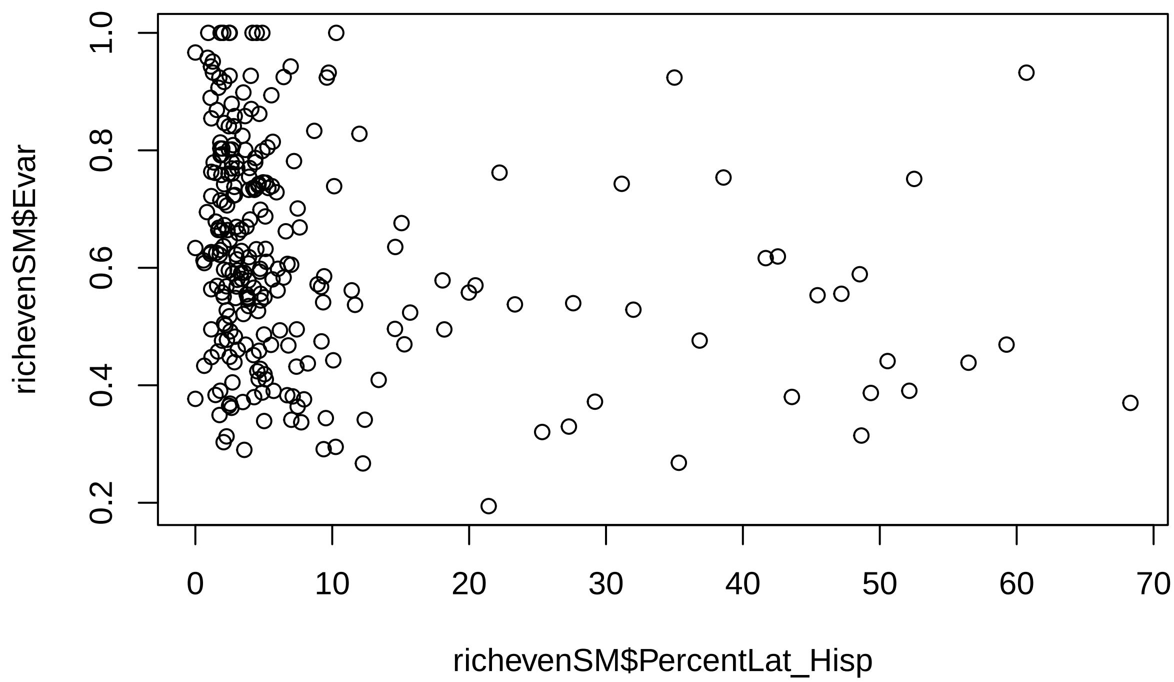
```r
plot(richevenSM$PercentBlk, richevenSM$Evar)
```




```r
#percent hispanic
cor.test(richevenSM$PercentLat_Hisp, richevenSM$Evar)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richevenSM$PercentLat_Hisp and richevenSM$Evar
## t = -3.733, df = 256, p-value = 0.0002332
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3399012 -0.1080816
## sample estimates:
##        cor
## -0.2272076
```

```r
plot(richevenSM$PercentLat_Hisp, richevenSM$Evar)
```



#Abundance, Richness, evenness for medium trees

```
NBabundM <-data %>%
  filter(DBH>5&DBH<20) %>%
  group_by(Neighborhood, Species) %>%
  summarize(abund=sum(present))
```

```
## filter: removed 45,007 rows (49%), 46,432 rows remaining
```

```
## group_by: 2 grouping variables (Neighborhood, Species)
```

```
## summarize: now 6,508 rows and 3 columns, one group variable remaining (Neighborhood)
```

```
#calculate richness and eveness for medium trees
richevenM<-community_structure(NBabundM, abundance.var="abund", time.var= NULL, replicate.var = "Neighbo
  left_join(NBdem) %>%
  unique()
```

```
## Joining with `by = join_by(Neighborhood)`
```

```
## left_join: added 8 columns (Population, PercentWhite, PercentBlk, PercentNtv, PercentAsian, ...)
```

```
##            > rows only in x     0
```

```
##            > rows only in y  (  2)
```

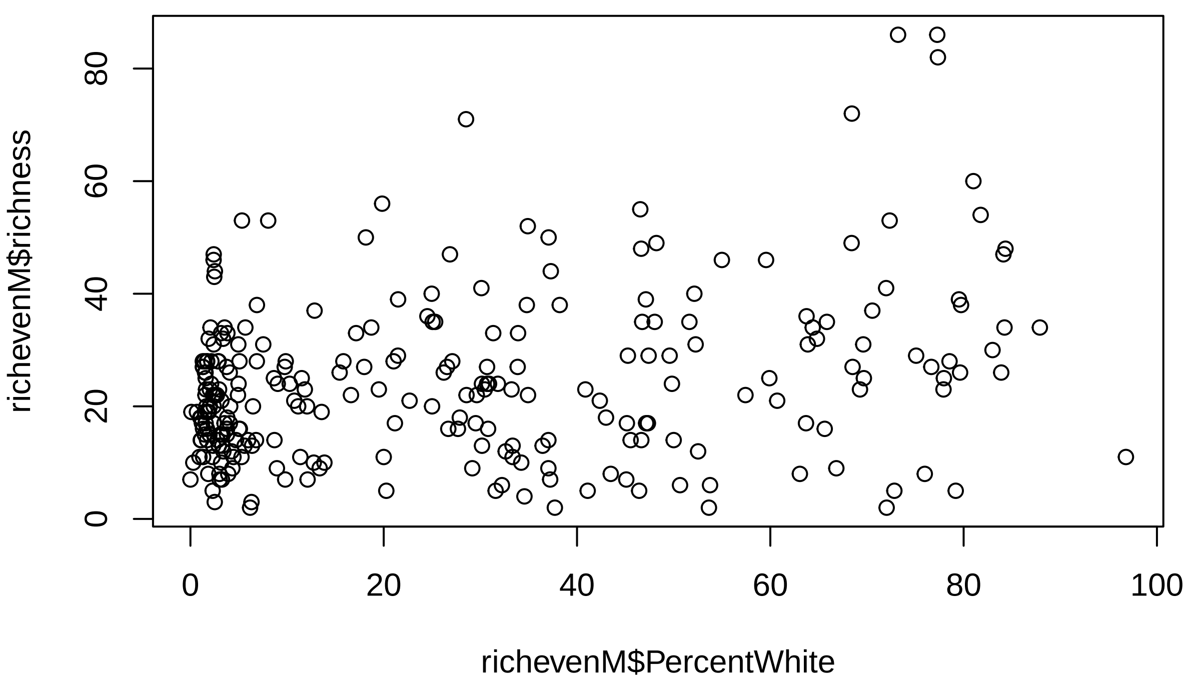
```
##              > matched rows     276

##              >                 =====

##              > rows total       276
```

#correlations for medium trees

```r
#percent white vs richness
cor.test(richevenM$PercentWhite, richevenM$richness)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richevenM$PercentWhite and richevenM$richness
## t = 5.1967, df = 267, p-value = 4.034e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1903780 0.4078884
## sample estimates:
##       cor
## 0.3030752
```

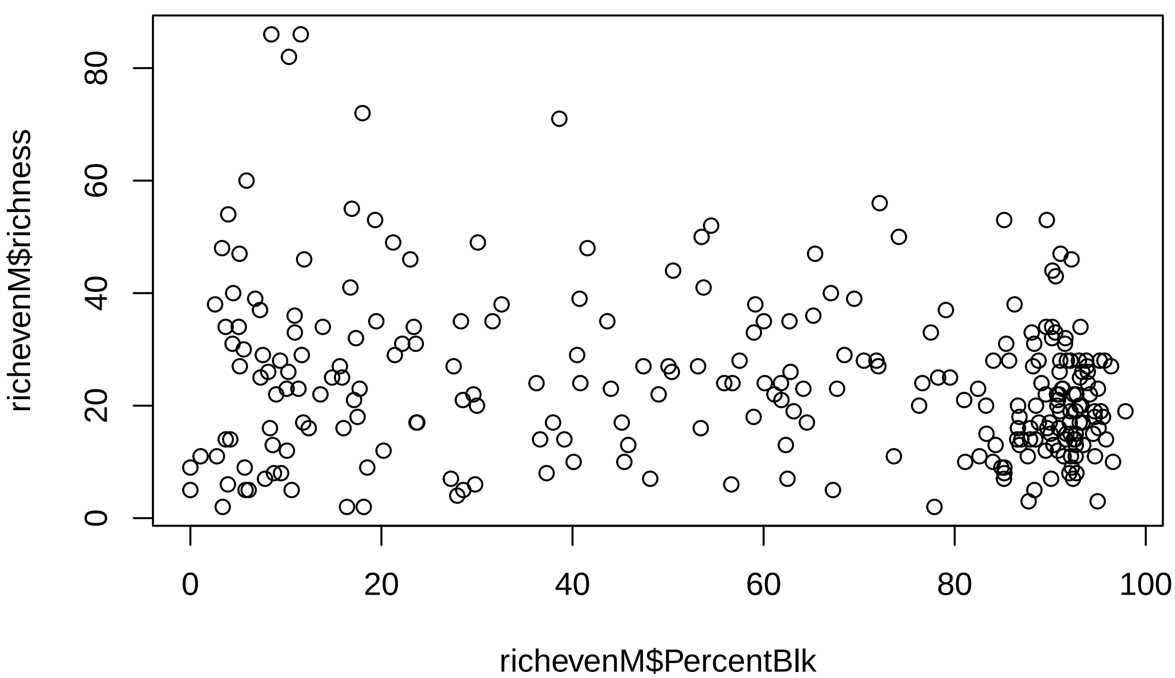
```r
plot(richevenM$PercentWhite, richevenM$richness)
```



```r
#percent black richness
cor.test(richevenM$PercentBlk, richevenM$richness)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richevenM$PercentBlk and richevenM$richness
## t = -3.097, df = 267, p-value = 0.002163
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.29915221 -0.06813628
## sample estimates:
##        cor
## -0.1862167
```

```r
plot(richevenM$PercentBlk, richevenM$richness)
```

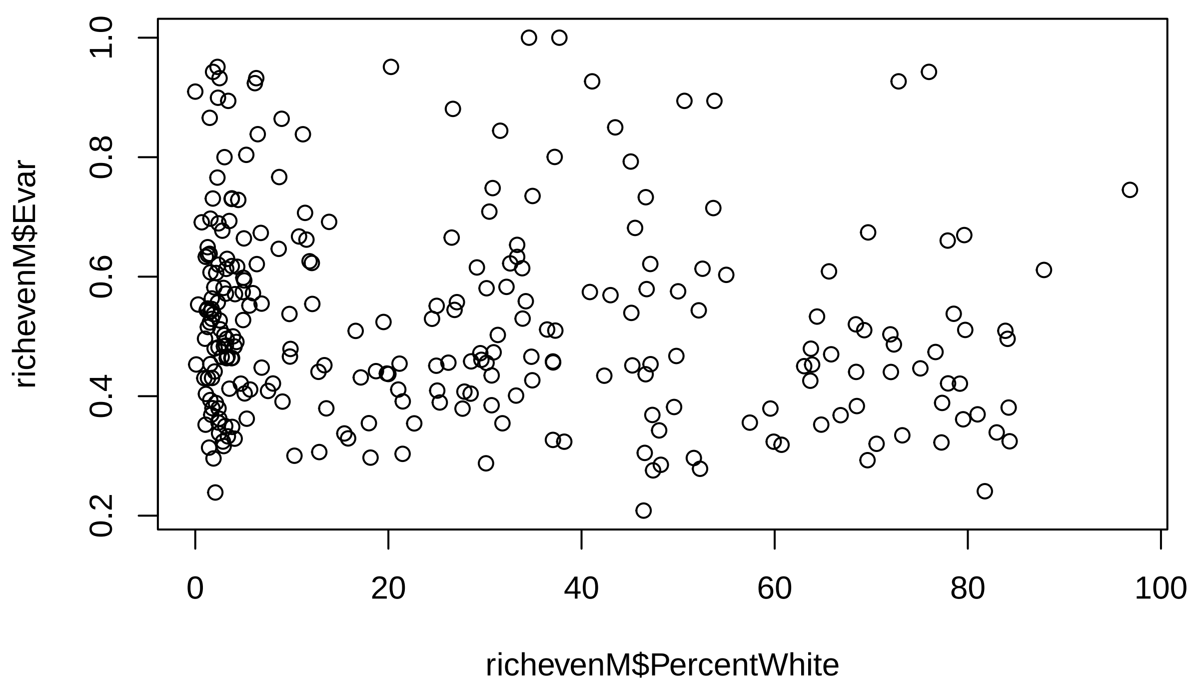



```r
#percent white vs evenness
cor.test(richevenM$Evar,richevenM$PercentWhite)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richevenM$Evar and richevenM$PercentWhite
```

```
## t = -2.2709, df = 267, p-value = 0.02395
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.25308322 -0.01835603
## sample estimates:
##        cor
## -0.1376518
```

```
plot(richevenM$PercentWhite, richevenM$Evar)
```



#Abundance, Richness, evenness for large trees

```
NBabundL <-data %>%
  filter(DBH<=20) %>%
  group_by(Neighborhood, Species) %>%
  summarize(abund=sum(present))
```

```
## filter: removed 14,904 rows (16%), 76,535 rows remaining

## group_by: 2 grouping variables (Neighborhood, Species)

## summarize: now 8,874 rows and 3 columns, one group variable remaining (Neighborhood)
```

```r
#calculate richness and eveness for medium trees
richevenL<-community_structure(NBabundL, abundance.var="abund", time.var= NULL, replicate.var = "Neighbo
  left_join(NBdem) %>%
  unique()
```

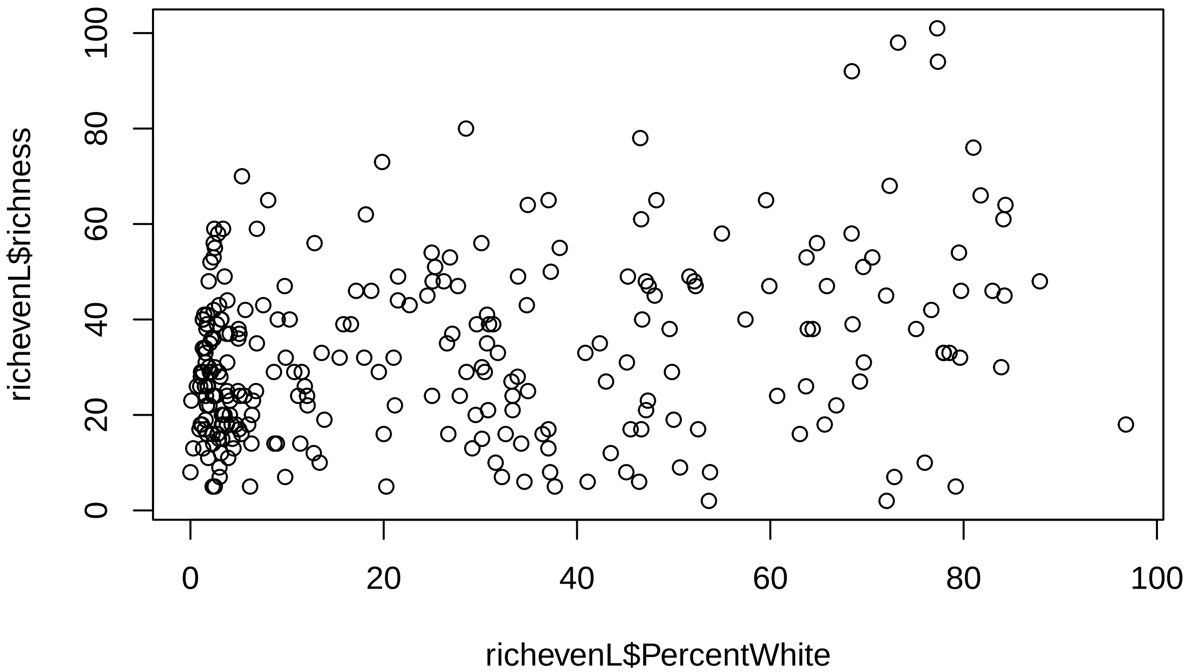
```
## Joining with `by = join_by(Neighborhood)`

## left_join: added 8 columns (Population, PercentWhite, PercentBlk, PercentNtv, PercentAsian, ...)

##            > rows only in x     0

##            > rows only in y  (   2)

##            > matched rows     276

##            >                 =====

##            > rows total       276
```

#correlations for large trees

```r
#percent white vs richness
cor.test(richevenL$PercentWhite, richevenL$richness)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richevenL$PercentWhite and richevenL$richness
## t = 4.7954, df = 267, p-value = 2.701e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1676451 0.3881235
## sample estimates:
##      cor
## 0.281597
```

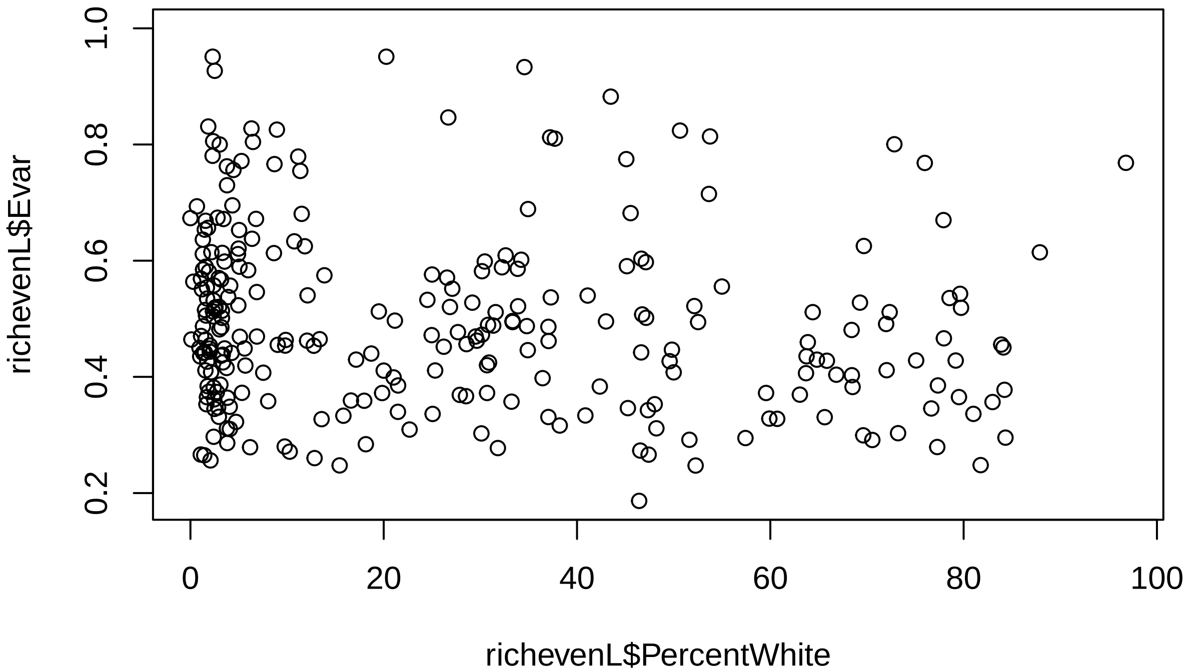
```r
plot(richevenL$PercentWhite, richevenL$richness)
```




```r
#percent white vs evenness
cor.test(richevenL$Evar,richevenL$PercentWhite)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richevenL$Evar and richevenL$PercentWhite
## t = -2.4821, df = 267, p-value = 0.01368
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.26501892 -0.03114252
## sample estimates:
##        cor
## -0.1501811
```

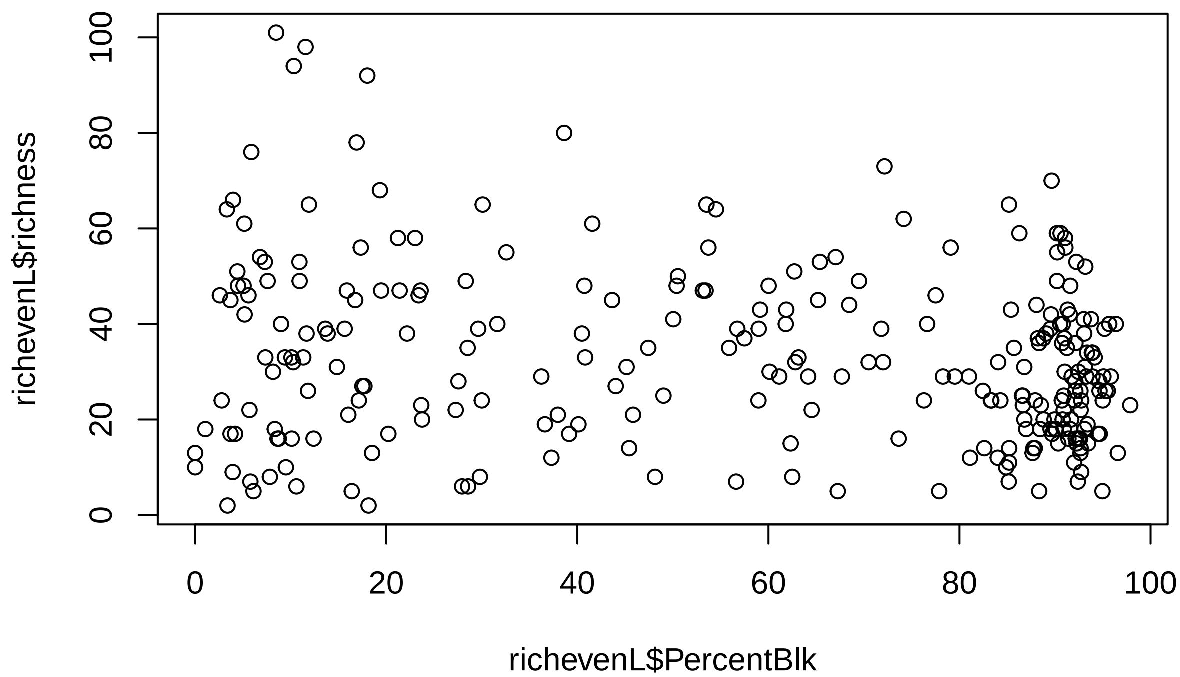
```r
plot(richevenL$PercentWhite, richevenL$Evar)
```




```r
#percent black
cor.test(richevenL$PercentBlk, richevenL$richness)
```

```
##
##  Pearson's product-moment correlation
##
## data:  richevenL$PercentBlk and richevenL$richness
## t = -3.0767, df = 267, p-value = 0.002311
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.29804152 -0.06692243
## sample estimates:
##        cor
## -0.1850393
```

```r
plot(richevenL$PercentBlk, richevenL$richness)
```



neighborhood land coverage csv

```
neighs <- read.csv('../../BaltimoreStreetTreeProject_Large_Data\\neighs.csv') %>%
  rename(Number = X357459) %>%
  rename (Neighborhood = Beechfield)
```

```
## rename: renamed one variable (Number)
```

```
## rename: renamed one variable (Neighborhood)
```

```
#pivot column and join with NB demographics
NBland <- neighs %>%
  pivot_wider(names_from = Tree.Canopy, values_from = Number) %>%
  left_join(NBdem)
```

```
## pivot_wider: reorganized (Number, Tree.Canopy) into (Herbaceous, Barren, Structures, Other Impervious
```

```
## Joining with ‘by = join_by(Neighborhood)‘
## left_join: added 8 columns (Population, PercentWhite, PercentBlk, PercentNtv,
## PercentAsian, ...)
## > rows only in x 2
```

```
## > rows only in y ( 1)
## > matched rows 277
## > =====
## > rows total 279
```

```r
head(NBland)
```

```
## # A tibble: 6 x 21
##   Neighborhood            Herbaceous Barren Structures `Other Impervious`  Roads
##   <chr>                        <int>  <int>      <int>              <int>  <int>
## 1 Beechfield                  240759    782     110292             132252 102820
## 2 Belair-Parkside              17044     NA      15802              11261  11745
## 3 Curtis Bay Industrial ~     132683  63428     137024             628274  50949
## 4 Eastwood                     14292     NA      18408              21759  16836
## 5 Forest Park Golf Course     142923    475      15442              18976  22741
## 6 Harwood                      12612     NA      61684              31956  36965
## # i 15 more variables: `Tree Canopy over Structures` <int>,
## #   `Tree Canopy over Other Impervious` <int>, `Tree Canopy over Roads` <int>,
## #   `Tree Canopy` <int>, Water <int>, `Emergent Wetlands` <int>,
## #   `Scrub\\Shrub` <int>, Population <dbl>, PercentWhite <dbl>,
## #   PercentBlk <dbl>, PercentNtv <dbl>, PercentAsian <dbl>,
## #   PercentNatHaw_Pac <dbl>, PercentLat_Hisp <dbl>, PercentOther_Race <dbl>
```

land coverage correlations

```r
#Percent white vs land structures
cor.test(NBland$PercentWhite, NBland$Herbaceous)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentWhite and NBland$Herbaceous
## t = -0.62471, df = 268, p-value = 0.5327
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.15679450  0.08161506
## sample estimates:
##         cor
## -0.03813235
```

```r
cor.test(NBland$PercentWhite, NBland$Barren)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentWhite and NBland$Barren
## t = 0.66243, df = 133, p-value = 0.5088
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1127041  0.2241306
```

```
## sample estimates:
##        cor
## 0.05734505
```

```r
cor.test(NBland$PercentWhite, NBland$Structures)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentWhite and NBland$Structures
## t = 1.1883, df = 268, p-value = 0.2358
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.04738946  0.19012872
## sample estimates:
##        cor
## 0.07239598
```

```r
cor.test(NBland$PercentWhite, NBland$`Other Impervious`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentWhite and NBland$‘Other Impervious‘
## t = 0.55131, df = 268, p-value = 0.5819
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.08606425  0.15242091
## sample estimates:
##        cor
## 0.03365743
```

```r
cor.test(NBland$PercentWhite, NBland$Roads)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentWhite and NBland$Roads
## t = -0.038598, df = 268, p-value = 0.9692
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1216994  0.1170511
## sample estimates:
##          cor
## -0.002357719
```

```r
cor.test(NBland$PercentWhite, NBland$`Tree Canopy over Structures`)
```

```
##
##  Pearson's product-moment correlation
##
```

```
## data:  NBland$PercentWhite and NBland$`Tree Canopy over Structures`
## t = 2.4914, df = 267, p-value = 0.01333
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.03170578 0.26554305
## sample estimates:
##       cor
## 0.1507321
```

```r
cor.test(NBland$PercentWhite, NBland$`Tree Canopy over Other Impervious`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentWhite and NBland$`Tree Canopy over Other Impervious`
## t = 2.1825, df = 268, p-value = 0.02994
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.01297703 0.24761785
## sample estimates:
##       cor
## 0.1321482
```

```r
cor.test(NBland$PercentWhite, NBland$`Tree Canopy over Roads`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentWhite and NBland$`Tree Canopy over Roads`
## t = 2.7496, df = 268, p-value = 0.006372
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.04719807 0.27949008
## sample estimates:
##       cor
## 0.1656407
```

```r
cor.test(NBland$PercentWhite, NBland$`Tree Canopy`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentWhite and NBland$`Tree Canopy`
## t = 2.1076, df = 267, p-value = 0.036
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.008454002 0.243790697
## sample estimates:
##       cor
## 0.1279226
```

```r
cor.test(NBland$PercentWhite, NBland$`Water`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentWhite and NBland$Water
## t = -0.37793, df = 68, p-value = 0.7067
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2777686  0.1912487
## sample estimates:
##         cor
## -0.04578272
```

```r
cor.test(NBland$PercentWhite, NBland$`Tree Canopy`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentWhite and NBland$`Tree Canopy`
## t = 2.1076, df = 267, p-value = 0.036
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.008454002 0.243790697
## sample estimates:
##       cor
## 0.1279226
```

```r
cor.test(NBland$PercentWhite, NBland$`Emergent Wetlands`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentWhite and NBland$`Emergent Wetlands`
## t = -0.10637, df = 6, p-value = 0.9188
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.7258664  0.6821427
## sample estimates:
##         cor
## -0.04338471
```

```r
#Percent Black vs land structures
cor.test(NBland$PercentBlk, NBland$Herbaceous)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentBlk and NBland$Herbaceous
## t = -0.7282, df = 268, p-value = 0.4671
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  -0.16294929  0.07533772
## sample estimates:
##         cor
## -0.04443782
```

```r
cor.test(NBland$PercentBlk, NBland$Barren)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentBlk and NBland$Barren
## t = -2.8352, df = 133, p-value = 0.005296
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.39188383 -0.07270925
## sample estimates:
##        cor
## -0.2387336
```

```r
cor.test(NBland$PercentBlk, NBland$Structures)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentBlk and NBland$Structures
## t = -2.4051, df = 268, p-value = 0.01685
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.26021406 -0.02643615
## sample estimates:
##        cor
## -0.1453533
```

```r
cor.test(NBland$PercentBlk, NBland$`Other Impervious`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentBlk and NBland$`Other Impervious`
## t = -2.9194, df = 268, p-value = 0.003805
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.28888416 -0.05738957
## sample estimates:
##        cor
## -0.1755627
```

```r
cor.test(NBland$PercentBlk, NBland$Roads)
```

```
##
```

```
##  Pearson's product-moment correlation
##
## data:  NBland$PercentBlk and NBland$Roads
## t = -1.4864, df = 268, p-value = 0.1384
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.20755803  0.02926926
## sample estimates:
##         cor
## -0.09042257
```

```
cor.test(NBland$PercentBlk, NBland$`Tree Canopy over Structures`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentBlk and NBland$`Tree Canopy over Structures`
## t = -0.6812, df = 267, p-value = 0.4963
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.16045147  0.07833532
## sample estimates:
##         cor
## -0.04165284
```

```
cor.test(NBland$PercentBlk, NBland$`Tree Canopy over Other Impervious`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentBlk and NBland$`Tree Canopy over Other Impervious`
## t = -1.6647, df = 268, p-value = 0.09714
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2179112  0.0184315
## sample estimates:
##       cor
## -0.101167
```

```
cor.test(NBland$PercentBlk, NBland$`Tree Canopy over Roads`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentBlk and NBland$`Tree Canopy over Roads`
## t = -1.2448, df = 268, p-value = 0.2143
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.19344384  0.04395493
## sample estimates:
##         cor
## -0.07581879
```

```r
cor.test(NBland$PercentBlk, NBland$`Tree Canopy`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentBlk and NBland$`Tree Canopy`
## t = -2.0975, df = 267, p-value = 0.03689
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.243212914 -0.007839842
## sample estimates:
##        cor
## -0.1273184
```

```r
cor.test(NBland$PercentBlk, NBland$`Water`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentBlk and NBland$Water
## t = -0.5338, df = 68, p-value = 0.5952
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2950925  0.1730023
## sample estimates:
##         cor
## -0.06459765
```

```r
cor.test(NBland$PercentBlk, NBland$`Tree Canopy`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentBlk and NBland$`Tree Canopy`
## t = -2.0975, df = 267, p-value = 0.03689
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.243212914 -0.007839842
## sample estimates:
##        cor
## -0.1273184
```

```r
cor.test(NBland$PercentBlk, NBland$`Emergent Wetlands`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentBlk and NBland$`Emergent Wetlands`
## t = -0.47538, df = 6, p-value = 0.6513
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.7892339  0.5938860
## sample estimates:
##        cor
## -0.1905177
```

```r
#Percent Population vs land structures
cor.test(NBland$Population, NBland$Herbaceous)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$Population and NBland$Herbaceous
## t = 6.6556, df = 275, p-value = 1.524e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2663022 0.4697044
## sample estimates:
##       cor
## 0.3724677
```

```r
cor.test(NBland$Population, NBland$Barren)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$Population and NBland$Barren
## t = -1.564, df = 140, p-value = 0.1201
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.28951989  0.03442819
## sample estimates:
##        cor
## -0.1310423
```

```r
cor.test(NBland$Population, NBland$Structures)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$Population and NBland$Structures
## t = 12.531, df = 273, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5234140 0.6743677
## sample estimates:
##       cor
## 0.6042857
```

```r
cor.test(NBland$Population, NBland$`Other Impervious`)
```

```
##
```

```
##  Pearson's product-moment correlation
##
## data:  NBland$Population and NBland$`Other Impervious`
## t = 1.2412, df = 275, p-value = 0.2156
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.04359821  0.19081771
## sample estimates:
##        cor
## 0.07464081
```

```r
cor.test(NBland$Population, NBland$Roads)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$Population and NBland$Roads
## t = 12.075, df = 275, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5058893 0.6606697
## sample estimates:
##       cor
## 0.5886485
```

```r
cor.test(NBland$Population, NBland$`Tree Canopy over Structures`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$Population and NBland$`Tree Canopy over Structures`
## t = 7.4533, df = 273, p-value = 1.207e-12
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3078871 0.5049217
## sample estimates:
##       cor
## 0.4111961
```

```r
cor.test(NBland$Population, NBland$`Tree Canopy over Other Impervious`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$Population and NBland$`Tree Canopy over Other Impervious`
## t = 12.212, df = 275, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5108262 0.6644048
## sample estimates:
##      cor
## 0.592982
```

```r
cor.test(NBland$Population, NBland$`Tree Canopy over Roads`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$Population and NBland$`Tree Canopy over Roads`
## t = 9.0529, df = 275, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3829283 0.5651025
## sample estimates:
##       cor
## 0.4791593
```

```r
cor.test(NBland$Population, NBland$`Tree Canopy`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$Population and NBland$`Tree Canopy`
## t = 4.6503, df = 274, p-value = 5.16e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1574251 0.3765127
## sample estimates:
##      cor
## 0.270467
```

```r
cor.test(NBland$Population, NBland$`Water`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$Population and NBland$Water
## t = -1.5206, df = 74, p-value = 0.1326
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.38444009  0.05348239
## sample estimates:
##        cor
## -0.1740721
```

```r
cor.test(NBland$Population, NBland$`Tree Canopy`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$Population and NBland$`Tree Canopy`
## t = 4.6503, df = 274, p-value = 5.16e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

```
##  0.1574251 0.3765127
## sample estimates:
##       cor
## 0.270467
```

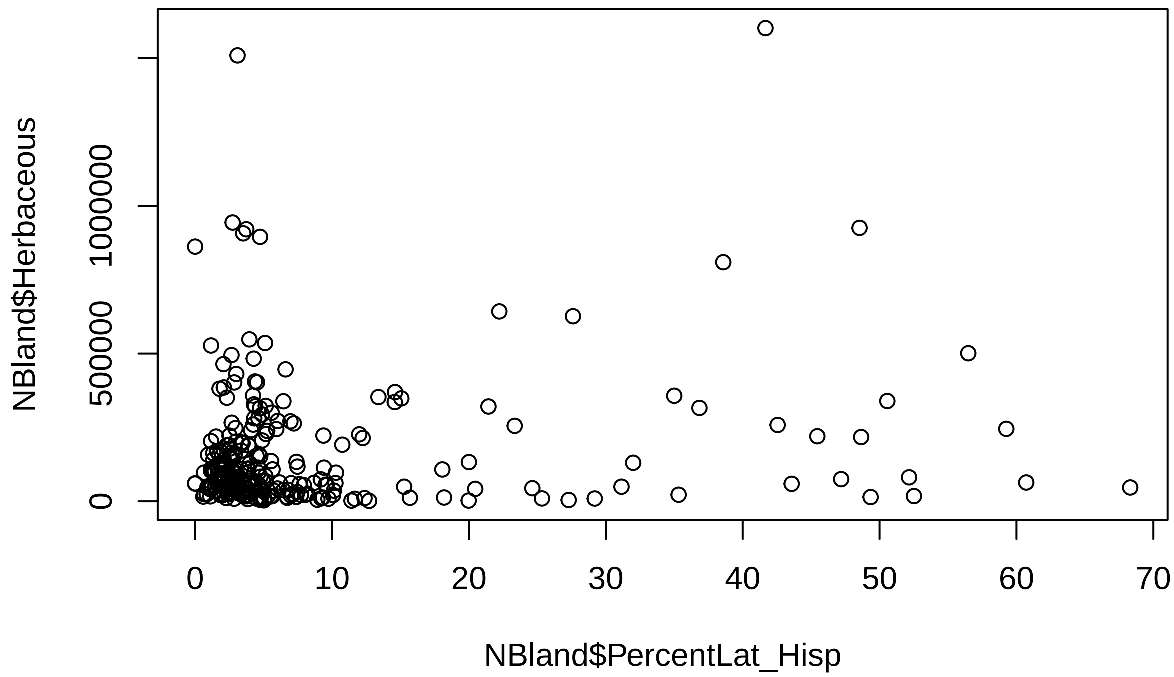
```r
cor.test(NBland$Population, NBland$`Emergent Wetlands`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$Population and NBland$`Emergent Wetlands`
## t = -1.0874, df = 6, p-value = 0.3186
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.8635189  0.4186278
## sample estimates:
##        cor
## -0.4057353
```

```r
#Percent Latino vs land structures
cor.test(NBland$PercentLat_Hisp, NBland$Herbaceous)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentLat_Hisp and NBland$Herbaceous
## t = 2.6839, df = 268, p-value = 0.007729
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.04324744 0.27583656
## sample estimates:
##       cor
## 0.1617881
```

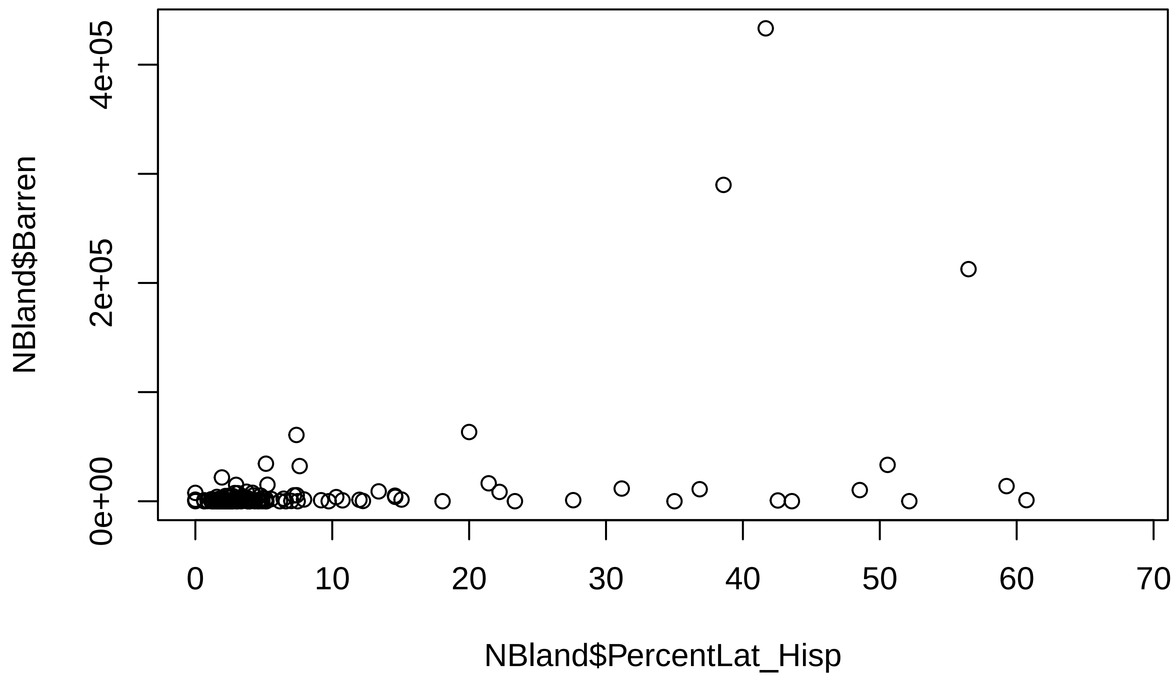
```r
plot(NBland$PercentLat_Hisp, NBland$Herbaceous)
```




```
cor.test(NBland$PercentLat_Hisp, NBland$Barren) # r = 0.41
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentLat_Hisp and NBland$Barren
## t = 5.2501, df = 133, p-value = 5.864e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2638387 0.5451229
## sample estimates:
##       cor
## 0.4143263
```

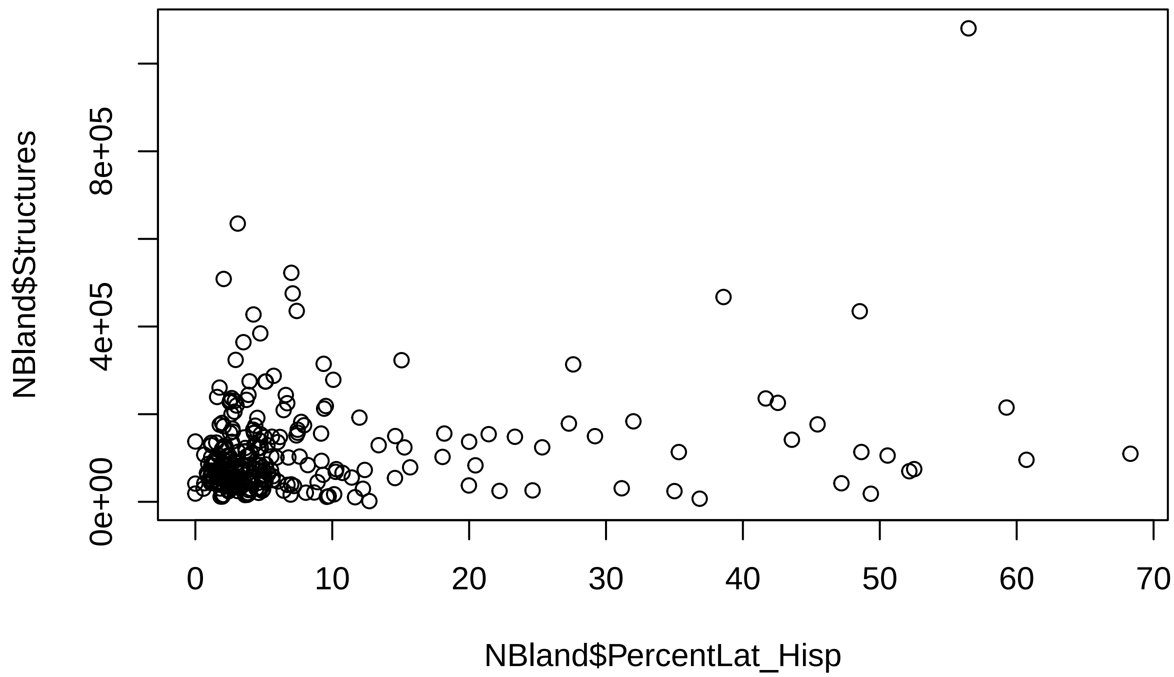
```
plot(NBland$PercentLat_Hisp, NBland$Barren)
```




```
cor.test(NBland$PercentLat_Hisp, NBland$Structures)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentLat_Hisp and NBland$Structures
## t = 3.778, df = 268, p-value = 0.0001949
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1083990 0.3352419
## sample estimates:
##       cor
## 0.2248651
```

```
plot(NBland$PercentLat_Hisp, NBland$Structures)
```




```r
cor.test(NBland$PercentLat_Hisp, NBland$`Other Impervious`) # r = 0.4
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentLat_Hisp and NBland$`Other Impervious`
## t = 7.1451, df = 268, p-value = 8.509e-12
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2947130 0.4957198
## sample estimates:
##       cor
## 0.4000157
```

```r
cor.test(NBland$PercentLat_Hisp, NBland$Roads)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentLat_Hisp and NBland$Roads
## t = 4.2179, df = 268, p-value = 3.376e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1341211 0.3582091
```

```
## sample estimates:
##       cor
## 0.2495023
```

```r
cor.test(NBland$PercentLat_Hisp, NBland$`Tree Canopy over Structures`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentLat_Hisp and NBland$`Tree Canopy over Structures`
## t = -2.6175, df = 267, p-value = 0.009362
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.27261453 -0.03931961
## sample estimates:
##        cor
## -0.1581737
```

```r
cor.test(NBland$PercentLat_Hisp, NBland$`Tree Canopy over Other Impervious`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentLat_Hisp and NBland$`Tree Canopy over Other Impervious`
## t = -0.0062959, df = 268, p-value = 0.995
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1197550  0.1189968
## sample estimates:
##           cor
## -0.0003845849
```

```r
cor.test(NBland$PercentLat_Hisp, NBland$`Tree Canopy over Roads`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBland$PercentLat_Hisp and NBland$`Tree Canopy over Roads`
## t = -2.3122, df = 268, p-value = 0.02152
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.25497398 -0.02082677
## sample estimates:
##       cor
## -0.139855
```

```r
cor.test(NBland$PercentLat_Hisp, NBland$`Tree Canopy`)
```

```
##
##  Pearson's product-moment correlation
##
```

```
## data:  NBland$PercentLat_Hisp and NBland$‘Tree Canopy‘
## t = 0.52701, df = 267, p-value = 0.5986
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.08770001  0.15125083
## sample estimates:
##        cor
## 0.03223603
```

```
cor.test(NBland$PercentLat_Hisp, NBland$`Water`)
```

```
##
##  Pearson’s product-moment correlation
##
## data:  NBland$PercentLat_Hisp and NBland$Water
## t = 1.3467, df = 68, p-value = 0.1825
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.07669854  0.38169820
## sample estimates:
##       cor
## 0.1611803
```

```
cor.test(NBland$PercentLat_Hisp, NBland$`Tree Canopy`)
```

```
##
##  Pearson’s product-moment correlation
##
## data:  NBland$PercentLat_Hisp and NBland$‘Tree Canopy‘
## t = 0.52701, df = 267, p-value = 0.5986
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.08770001  0.15125083
## sample estimates:
##        cor
## 0.03223603
```

```
cor.test(NBland$PercentLat_Hisp, NBland$`Emergent Wetlands`)
```

```
##
##  Pearson’s product-moment correlation
##
## data:  NBland$PercentLat_Hisp and NBland$‘Emergent Wetlands‘
## t = 0.73268, df = 6, p-value = 0.4914
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5238984  0.8247035
## sample estimates:
##       cor
## 0.2865693
```

#data frame with total area in pixels

```r
(test <-
    read_csv('C:\\Users\\trini\\Documents\\Street Trees\\BaltimoreTrees\\BaltimoreStreetTreeProject_Larg
             , col_names = c('id', 'n', 'lc_class')) %>%
    tidylog::filter(lc_class != 0) %>%
    tidylog::filter(!is.na(lc_class)) %>%
    tidylog::pivot_wider(names_from = lc_class, values_from = n, values_fill = 0) %>% # need old pipe
    tidylog::mutate(total_area_in_pixels = rowSums(.[,-1]))                          # for dot notatio
  )
```

```
## Rows: 2482 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): id, lc_class
## dbl (1): n
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## filter: no rows removed
##
## filter: no rows removed
##
## pivot_wider: reorganized (n, lc_class) into (Tree Canopy, Herbaceous, Barren, Structures, Other Impe
##
## mutate: new variable 'total_area_in_pixels' (double) with 279 unique values and 0% NA
```

```
## # A tibble: 279 x 14
##    id       `Tree Canopy` Herbaceous Barren Structures `Other Impervious`  Roads
##    <chr>            <dbl>      <dbl>  <dbl>      <dbl>              <dbl>  <dbl>
##  1 Beechfi~        357459     240759    782     110292             132252 102820
##  2 Belair--~        24678      17044      0      15802              11261  11745
##  3 Curtis ~         51425     132683  63428     137024             628274  50949
##  4 Eastwood           647      14292      0      18408              21759  16836
##  5 Forest ~         50564     142923    475      15442              18976  22741
##  6 Harwood          23091      12612      0      61684              31956  36965
##  7 Hawkins~        977950    1601502 433301     235913            1189806 445056
##  8 Highlan~          6795       9401      0     149803              97044  92424
##  9 Hoes He~         44564      35939      0      37089              40984  26368
## 10 Bayview           5373      17684      0      74641              74966  64690
## # i 269 more rows
## # i 7 more variables: `Tree Canopy over Structures` <dbl>,
## #   `Tree Canopy over Other Impervious` <dbl>, `Tree Canopy over Roads` <dbl>,
## #   Water <dbl>, `Emergent Wetlands` <dbl>, `Scrub\\Shrub` <dbl>,
## #   total_area_in_pixels <dbl>
```

```r
NBarea <- test %>%
  rename(Area = total_area_in_pixels) %>%
  rename(Neighborhood = id) %>%
  left_join(NBdem) %>%
  left_join(richeven) %>%
  mutate(PopDensity = Population/Area)
```

```
## rename: renamed one variable (Area)
```

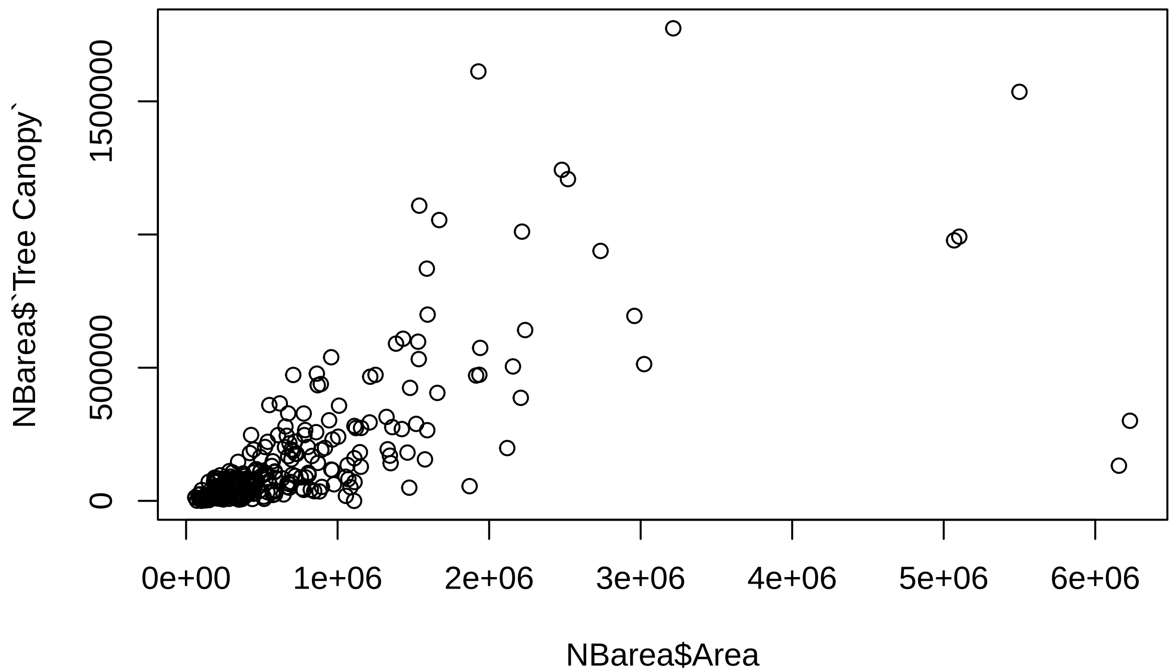
```
## rename: renamed one variable (Neighborhood)
## Joining with `by = join_by(Neighborhood)`left_join: added 8 columns (Population, PercentWhite, Percen
##            > rows only in x     2
##            > rows only in y  (  1)
##            > matched rows     277
##            >                 =====
##            > rows total       279
## Joining with `by = join_by(Neighborhood, Population, PercentWhite, PercentBlk, PercentNtv, PercentAs
##            > rows only in x     2
##            > rows only in y  (  0)
##            > matched rows     277
##            >                 =====
##            > rows total       279
## mutate: new variable 'PopDensity' (double) with 272 unique values and 1% NA
```

area correlations

```r
#area and land cover types
cor.test(NBarea$Area, NBarea$`Tree Canopy`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Area and NBarea$`Tree Canopy`
## t = 15.769, df = 277, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6204570 0.7450341
## sample estimates:
##       cor
## 0.6877764
```

```r
plot(NBarea$Area, NBarea$`Tree Canopy`)
```




```
cor.test(NBarea$Area, NBarea$Herbaceous)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Area and NBarea$Herbaceous
## t = 27.063, df = 277, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8160036 0.8811058
## sample estimates:
##       cor
## 0.8518108
```

```
cor.test(NBarea$Area, NBarea$Barren)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Area and NBarea$Barren
## t = 11.952, df = 277, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5001211 0.6558064
## sample estimates:
```

```
##       cor
## 0.5832958
```

```r
cor.test(NBarea$Area, NBarea$Structures)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Area and NBarea$Structures
## t = 18.996, df = 277, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6962175 0.7990120
## sample estimates:
##       cor
## 0.7521545
```

```r
cor.test(NBarea$Area, NBarea$`Other Impervious`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Area and NBarea$`Other Impervious`
## t = 21.178, df = 277, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7368523 0.8272967
## sample estimates:
##       cor
## 0.7862499
```

```r
cor.test(NBarea$Area, NBarea$Roads)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Area and NBarea$Roads
## t = 29.007, df = 277, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8349874 0.8937656
## sample estimates:
##     cor
## 0.86737
```

```r
cor.test(NBarea$Area, NBarea$`Tree Canopy over Structures`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Area and NBarea$`Tree Canopy over Structures`
```

```
## t = 7.8748, df = 277, p-value = 7.782e-14
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3266667 0.5190541
## sample estimates:
##       cor
## 0.4276917
```

```r
cor.test(NBarea$Area, NBarea$`Tree Canopy over Other Impervious`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Area and NBarea$`Tree Canopy over Other Impervious`
## t = 19.045, df = 277, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6972132 0.7997106
## sample estimates:
##       cor
## 0.7529936
```

```r
cor.test(NBarea$Area, NBarea$`Tree Canopy over Roads`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Area and NBarea$`Tree Canopy over Roads`
## t = 9.2954, df = 277, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3926617 0.5722722
## sample estimates:
##       cor
## 0.4876093
```

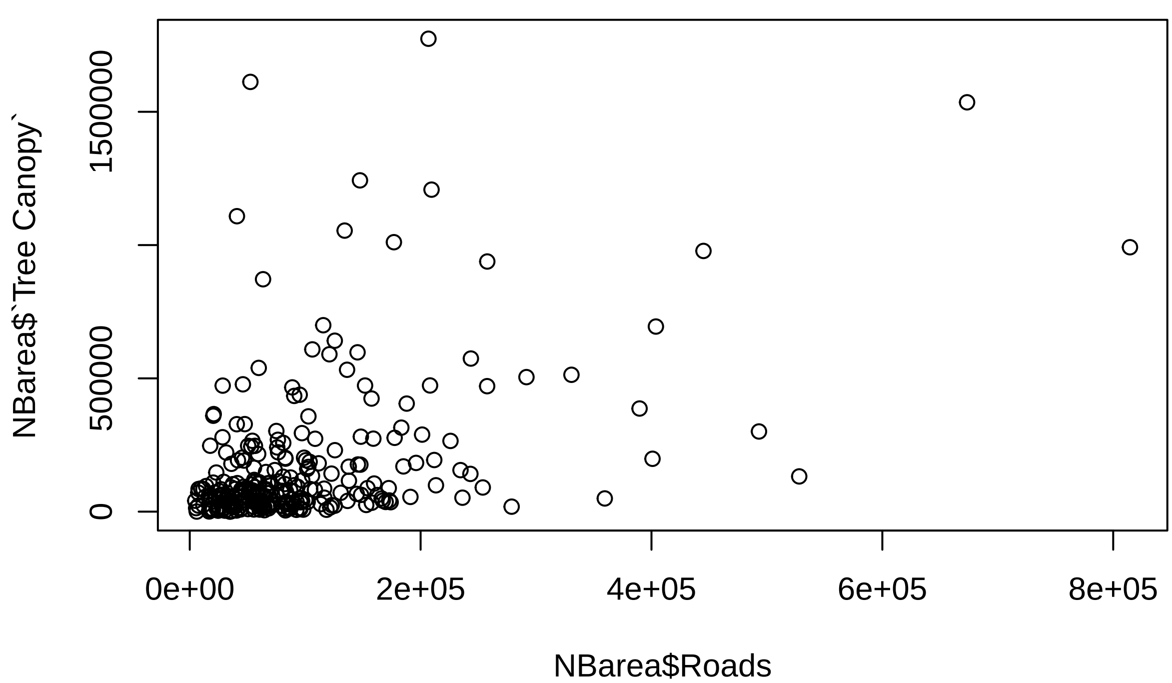
```r
cor.test(NBarea$Area, NBarea$Water)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Area and NBarea$Water
## t = 5.2743, df = 277, p-value = 2.689e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1914569 0.4051550
## sample estimates:
##       cor
## 0.3020965
```

```r
#roads and land cover
cor.test(NBarea$Roads, NBarea$`Tree Canopy`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Roads and NBarea$`Tree Canopy`
## t = 8.6082, df = 277, p-value = 5.669e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3614759 0.5473112
## sample estimates:
##       cor
## 0.4594064
```

```r
plot(NBarea$Roads, NBarea$`Tree Canopy`)
```




```r
cor.test(NBarea$Roads, NBarea$Herbaceous)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Roads and NBarea$Herbaceous
## t = 17.265, df = 277, p-value < 2.2e-16
```

```
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6581707 0.7721099
## sample estimates:
##       cor
## 0.7199568
```

```r
cor.test(NBarea$Roads, NBarea$Barren)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Roads and NBarea$Barren
## t = 7.2165, df = 277, p-value = 5.133e-12
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2941179 0.4922459
## sample estimates:
##       cor
## 0.3978097
```

```r
cor.test(NBarea$Roads, NBarea$Structures)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Roads and NBarea$Structures
## t = 27.565, df = 277, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8211782 0.8845659
## sample estimates:
##       cor
## 0.8560582
```

```r
cor.test(NBarea$Roads, NBarea$`Other Impervious`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Roads and NBarea$`Other Impervious`
## t = 15.451, df = 277, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6118286 0.7387813
## sample estimates:
##       cor
## 0.6803766
```

```r
cor.test(NBarea$Roads, NBarea$`Tree Canopy over Structures`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Roads and NBarea$`Tree Canopy over Structures`
## t = 5.3307, df = 277, p-value = 2.031e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.194564 0.407849
## sample estimates:
##       cor
## 0.3050264
```

```
cor.test(NBarea$Roads, NBarea$`Tree Canopy over Other Impervious`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Roads and NBarea$`Tree Canopy over Other Impervious`
## t = 16.558, df = 277, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6409324 0.7597851
## sample estimates:
##       cor
## 0.7052805
```

```
cor.test(NBarea$Roads, NBarea$`Tree Canopy over Roads`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Roads and NBarea$`Tree Canopy over Roads`
## t = 6.6957, df = 277, p-value = 1.191e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2675253 0.4700608
## sample estimates:
##       cor
## 0.3732316
```

```
#roads and richness/evenness
cor.test(NBarea$Roads, NBarea$richness)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Roads and NBarea$richness
## t = 7.6303, df = 275, p-value = 3.855e-13
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3156958 0.5106959
## sample estimates:
```

```
##      cor
## 0.417999
```

```r
cor.test(NBarea$Roads, NBarea$Evar)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$Roads and NBarea$Evar
## t = -5.484, df = 274, p-value = 9.447e-08
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4170744 -0.2039976
## sample estimates:
##       cor
## -0.314492
```

```r
#density vs richness/evenness
cor.test(NBarea$PopDensity, NBarea$richness)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$PopDensity and NBarea$richness
## t = 2.3608, df = 275, p-value = 0.01893
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.02347403 0.25456662
## sample estimates:
##       cor
## 0.1409396
```

```r
cor.test(NBarea$PopDensity, NBarea$Evar)
```

```
##
##  Pearson's product-moment correlation
##
## data:  NBarea$PopDensity and NBarea$Evar
## t = -3.2303, df = 274, p-value = 0.001387
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.30275956 -0.07516797
## sample estimates:
##        cor
## -0.1915372
```

#calculate land cover percentages