# myCodeWeek2

*Boyang Tang*

```
##Download data from the website
dest = "https://www.fhwa.dot.gov/bridge/nbi/2016/delimited/WI16.txt"
dat = fread(dest)
```

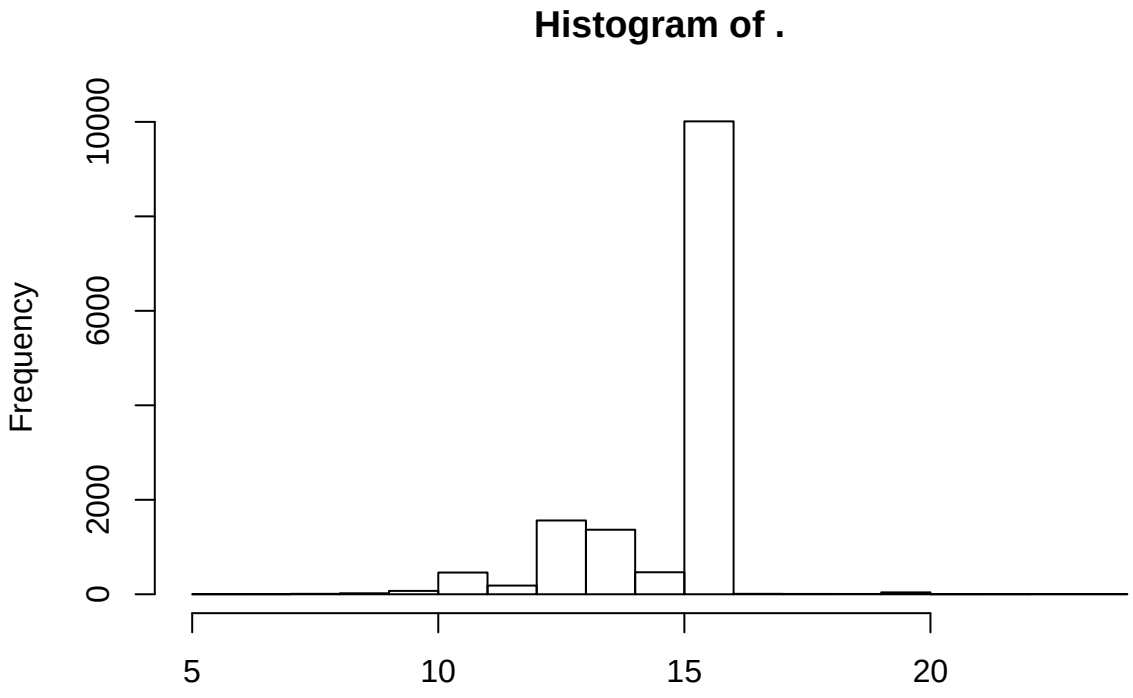
```
## Warning in fread(dest): Bumped column 125 to type character on data row
## 11570, field contains '00PE093'. Coercing previously read values in this
## column from logical, integer or numeric back to character which may not
## be lossless; e.g., if '00' and '000' occurred before they will now be just
## '0', and there may be inconsistencies with treatment of ',,' and ',NA,' too
## (if they occurred in this column before the bump). If this matters please
## rerun and set 'colClasses' to 'character' for this column. Please note that
## column type detection uses a sample of 1,000 rows (100 rows at 10 points)
## so hopefully this message should be very rare. If reporting to datatable-
## help, please rerun and include the output from verbose=TRUE.
```

```
dat = as.tbl(dat)
classes = sapply(dat, class) #get the variables of this data

##View the data
dat
```

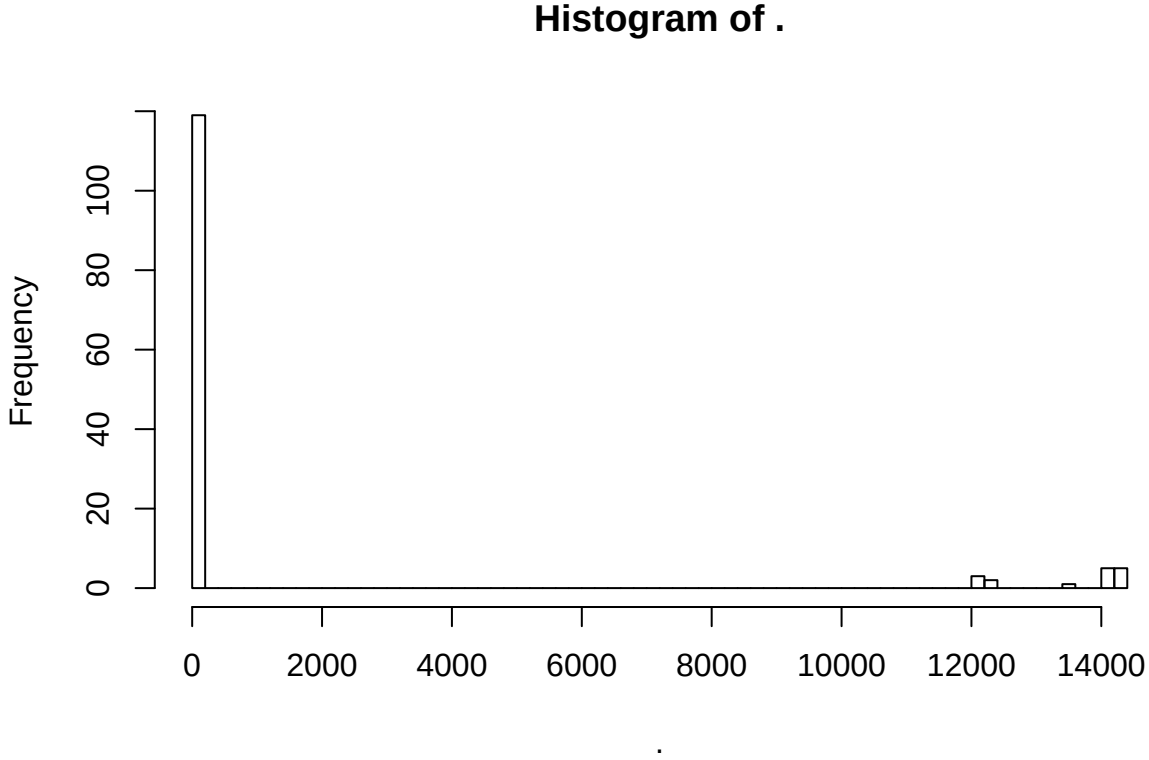
```
## # A tibble: 14,230 <U+00D7> 135
##    STATE_CODE_001 STRUCTURE_NUMBER_008 RECORD_TYPE_005A ROUTE_PREFIX_005B
##             <int>                <chr>            <int>             <int>
## 1              55      00000000000F303                1                 6
## 2              55      00000000000F310                1                 6
## 3              55      00000000000F311                1                 6
## 4              55      00000000000F315                1                 6
## 5              55      00000000000F317                1                 6
## 6              55      00000000000F318                1                 6
## 7              55      00000000000F321                1                 6
## 8              55      00000000000F323                1                 6
## 9              55      00000000000F324                1                 6
## 10             55      00000000000F325                1                 6
## # ... with 14,220 more rows, and 131 more variables:
## #   SERVICE_LEVEL_005C <int>, ROUTE_NUMBER_005D <chr>,
## #   DIRECTION_005E <int>, HIGHWAY_DISTRICT_002 <chr>,
## #   COUNTY_CODE_003 <int>, PLACE_CODE_004 <int>, FEATURES_DESC_006A <chr>,
## #   CRITICAL_FACILITY_006B <lgl>, FACILITY_CARRIED_007 <chr>,
## #   LOCATION_009 <chr>, MIN_VERT_CLR_010 <dbl>, KILOPOINT_011 <dbl>,
## #   BASE_HWY_NETWORK_012 <int>, LRS_INV_ROUTE_013A <chr>,
## #   SUBROUTE_NO_013B <int>, LAT_016 <int>, LONG_017 <int>,
## #   DETOUR_KILOS_019 <int>, TOLL_020 <int>, MAINTENANCE_021 <int>,
## #   OWNER_022 <int>, FUNCTIONAL_CLASS_026 <int>, YEAR_BUILT_027 <int>,
## #   TRAFFIC_LANES_ON_028A <int>, TRAFFIC_LANES_UND_028B <int>,
## #   ADT_029 <int>, YEAR_ADT_030 <int>, DESIGN_LOAD_031 <chr>,
## #   APPR_WIDTH_MT_032 <dbl>, MEDIAN_CODE_033 <int>,
## #   DEGREES_SKEW_034 <int>, STRUCTURE_FLARED_035 <int>,
## #   RAILINGS_036A <chr>, TRANSITIONS_036B <chr>, APPR_RAIL_036C <chr>,
## #   APPR_RAIL_END_036D <chr>, HISTORY_037 <int>, NAVIGATION_038 <chr>,
```

```
## #   NAV_VERT_CLR_MT_039 <dbl>, NAV_HORR_CLR_MT_040 <dbl>,
## #   OPEN_CLOSED_POSTED_041 <chr>, SERVICE_ON_042A <int>,
## #   SERVICE_UND_042B <int>, STRUCTURE_KIND_043A <int>,
## #   STRUCTURE_TYPE_043B <int>, APPR_KIND_044A <int>, APPR_TYPE_044B <int>,
## #   MAIN_UNIT_SPANS_045 <int>, APPR_SPANS_046 <int>,
## #   HORR_CLR_MT_047 <dbl>, MAX_SPAN_LEN_MT_048 <dbl>,
## #   STRUCTURE_LEN_MT_049 <dbl>, LEFT_CURB_MT_050A <dbl>,
## #   RIGHT_CURB_MT_050B <dbl>, ROADWAY_WIDTH_MT_051 <dbl>,
## #   DECK_WIDTH_MT_052 <dbl>, VERT_CLR_OVER_MT_053 <dbl>,
## #   VERT_CLR_UND_REF_054A <chr>, VERT_CLR_UND_054B <dbl>,
## #   LAT_UND_REF_055A <chr>, LAT_UND_MT_055B <dbl>,
## #   LEFT_LAT_UND_MT_056 <dbl>, DECK_COND_058 <chr>,
## #   SUPERSTRUCTURE_COND_059 <chr>, SUBSTRUCTURE_COND_060 <chr>,
## #   CHANNEL_COND_061 <chr>, CULVERT_COND_062 <chr>,
## #   OPR_RATING_METH_063 <int>, OPERATING_RATING_064 <dbl>,
## #   INV_RATING_METH_065 <int>, INVENTORY_RATING_066 <dbl>,
## #   STRUCTURAL_EVAL_067 <chr>, DECK_GEOMETRY_EVAL_068 <chr>,
## #   UNDCLRENCE_EVAL_069 <chr>, POSTING_EVAL_070 <int>,
## #   WATERWAY_EVAL_071 <chr>, APPR_ROAD_EVAL_072 <int>,
## #   WORK_PROPOSED_075A <int>, WORK_DONE_BY_075B <int>,
## #   IMP_LEN_MT_076 <dbl>, DATE_OF_INSPECT_090 <int>,
## #   INSPECT_FREQ_MONTHS_091 <int>, FRACTURE_092A <chr>,
## #   UNDWATER_LOOK_SEE_092B <chr>, SPEC_INSPECT_092C <chr>,
## #   FRACTURE_LAST_DATE_093A <int>, UNDWATER_LAST_DATE_093B <int>,
## #   SPEC_LAST_DATE_093C <int>, BRIDGE_IMP_COST_094 <int>,
## #   ROADWAY_IMP_COST_095 <int>, TOTAL_IMP_COST_096 <int>,
## #   YEAR_OF_IMP_097 <int>, OTHER_STATE_CODE_098A <int>,
## #   OTHER_STATE_PCNT_098B <int>, OTHR_STATE_STRUC_NO_099 <chr>,
## #   STRAHNET_HIGHWAY_100 <int>, PARALLEL_STRUCTURE_101 <chr>,
## #   TRAFFIC_DIRECTION_102 <int>, TEMP_STRUCTURE_103 <lgl>,
## #   HIGHWAY_SYSTEM_104 <int>, ...
```

```
is.na(dat) %>% rowSums %>% hist #check num of missing values of each column
```




```
is.na(dat) %>% colSums %>% hist(breaks = 100) #check missing entries
```




```
fun = function(x){ return(which(x>20)) }#which is more than 20
(bad =  is.na(dat) %>% colSums %>% fun)
```

```
##  CRITICAL_FACILITY_006B        SUBROUTE_NO_013B    OPERATING_RATING_064
```

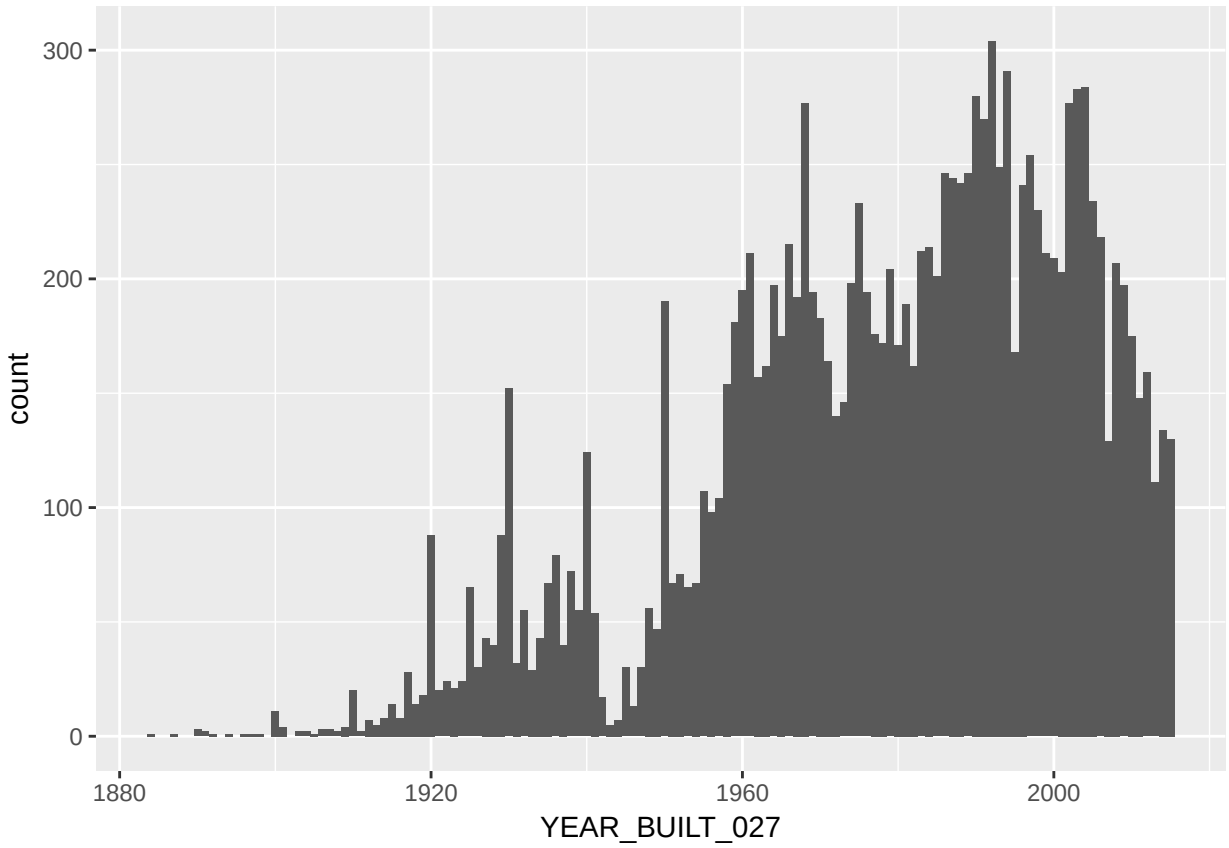
```
##                      12                      19                      73
##    INVENTORY_RATING_066      WORK_PROPOSED_075A       WORK_DONE_BY_075B
##                      75                      82                      83
## FRACTURE_LAST_DATE_093A UNDWATER_LAST_DATE_093B     SPEC_LAST_DATE_093C
##                      90                      91                      92
##     BRIDGE_IMP_COST_094    ROADWAY_IMP_COST_095      TOTAL_IMP_COST_096
##                      93                      94                      95
##         YEAR_OF_IMP_097   OTHER_STATE_CODE_098A   OTHER_STATE_PCNT_098B
##                      96                      97                      98
##      TEMP_STRUCTURE_103   PERCENT_ADT_TRUCK_109     PIER_PROTECTION_111
##                     103                     111                     113
##      MIN_NAV_CLR_MT_116                 REMARKS             PROJ_SUFFIX
##                     118                     123                     126
##         NBI_TYPE_OF_IMP         DTL_TYPE_OF_IMP            SPECIAL_CODE
##                     127                     128                     129
##               STEP_CODE
##                     130
```

```
dat = dat[,-bad] #get rid of the columns with too many missing values.

colnames(dat)
```

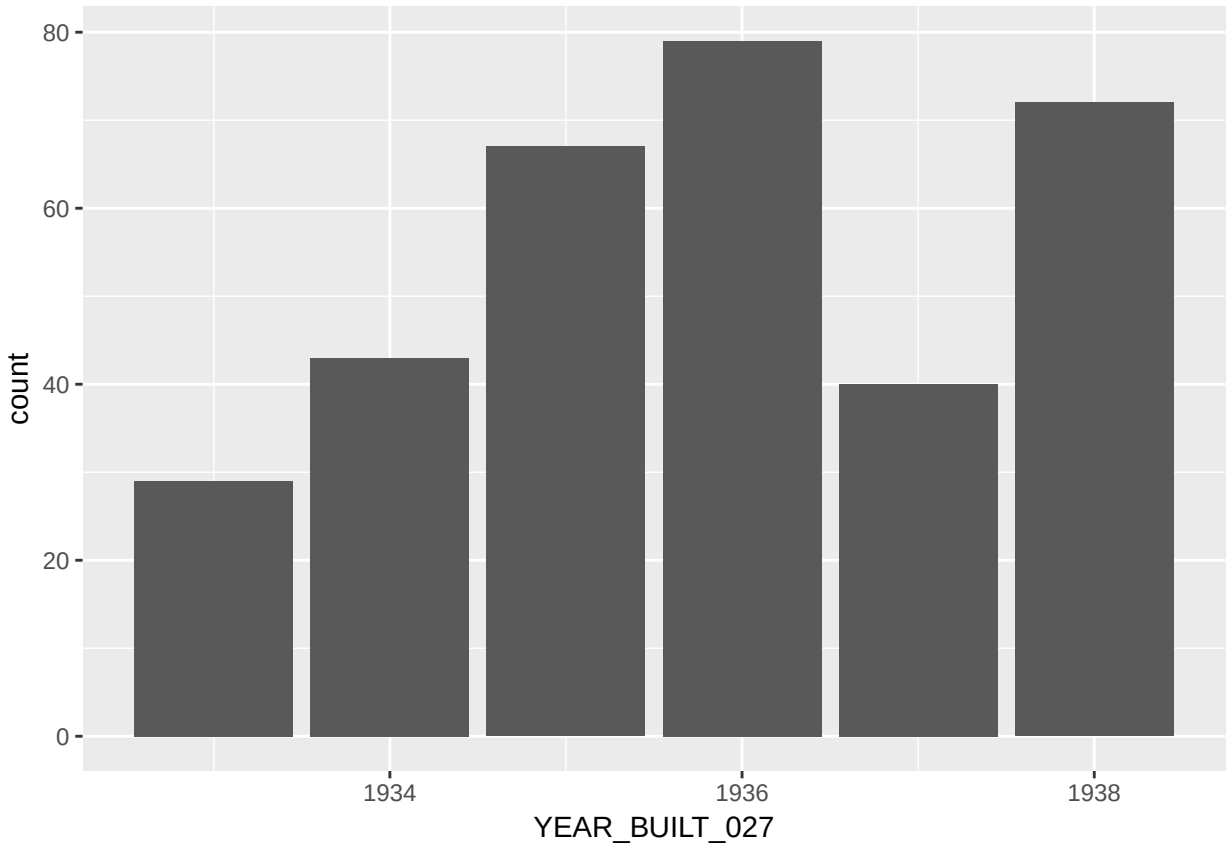
```
##   [1] "STATE_CODE_001"          "STRUCTURE_NUMBER_008"
##   [3] "RECORD_TYPE_005A"        "ROUTE_PREFIX_005B"
##   [5] "SERVICE_LEVEL_005C"      "ROUTE_NUMBER_005D"
##   [7] "DIRECTION_005E"          "HIGHWAY_DISTRICT_002"
##   [9] "COUNTY_CODE_003"         "PLACE_CODE_004"
##  [11] "FEATURES_DESC_006A"      "FACILITY_CARRIED_007"
##  [13] "LOCATION_009"            "MIN_VERT_CLR_010"
##  [15] "KILOPOINT_011"           "BASE_HWY_NETWORK_012"
##  [17] "LRS_INV_ROUTE_013A"      "LAT_016"
##  [19] "LONG_017"                "DETOUR_KILOS_019"
##  [21] "TOLL_020"                "MAINTENANCE_021"
##  [23] "OWNER_022"               "FUNCTIONAL_CLASS_026"
##  [25] "YEAR_BUILT_027"          "TRAFFIC_LANES_ON_028A"
##  [27] "TRAFFIC_LANES_UND_028B"  "ADT_029"
##  [29] "YEAR_ADT_030"            "DESIGN_LOAD_031"
##  [31] "APPR_WIDTH_MT_032"       "MEDIAN_CODE_033"
##  [33] "DEGREES_SKEW_034"        "STRUCTURE_FLARED_035"
##  [35] "RAILINGS_036A"           "TRANSITIONS_036B"
##  [37] "APPR_RAIL_036C"          "APPR_RAIL_END_036D"
##  [39] "HISTORY_037"             "NAVIGATION_038"
##  [41] "NAV_VERT_CLR_MT_039"     "NAV_HORR_CLR_MT_040"
##  [43] "OPEN_CLOSED_POSTED_041"  "SERVICE_ON_042A"
##  [45] "SERVICE_UND_042B"        "STRUCTURE_KIND_043A"
##  [47] "STRUCTURE_TYPE_043B"     "APPR_KIND_044A"
##  [49] "APPR_TYPE_044B"          "MAIN_UNIT_SPANS_045"
##  [51] "APPR_SPANS_046"          "HORR_CLR_MT_047"
##  [53] "MAX_SPAN_LEN_MT_048"     "STRUCTURE_LEN_MT_049"
##  [55] "LEFT_CURB_MT_050A"       "RIGHT_CURB_MT_050B"
##  [57] "ROADWAY_WIDTH_MT_051"    "DECK_WIDTH_MT_052"
##  [59] "VERT_CLR_OVER_MT_053"    "VERT_CLR_UND_REF_054A"
##  [61] "VERT_CLR_UND_054B"       "LAT_UND_REF_055A"
##  [63] "LAT_UND_MT_055B"         "LEFT_LAT_UND_MT_056"
##  [65] "DECK_COND_058"           "SUPERSTRUCTURE_COND_059"
```

```
##  [67] "SUBSTRUCTURE_COND_060"   "CHANNEL_COND_061"
##  [69] "CULVERT_COND_062"        "OPR_RATING_METH_063"
##  [71] "INV_RATING_METH_065"     "STRUCTURAL_EVAL_067"
##  [73] "DECK_GEOMETRY_EVAL_068"  "UNDCLRENCE_EVAL_069"
##  [75] "POSTING_EVAL_070"        "WATERWAY_EVAL_071"
##  [77] "APPR_ROAD_EVAL_072"      "IMP_LEN_MT_076"
##  [79] "DATE_OF_INSPECT_090"     "INSPECT_FREQ_MONTHS_091"
##  [81] "FRACTURE_092A"           "UNDWATER_LOOK_SEE_092B"
##  [83] "SPEC_INSPECT_092C"       "OTHR_STATE_STRUC_NO_099"
##  [85] "STRAHNET_HIGHWAY_100"    "PARALLEL_STRUCTURE_101"
##  [87] "TRAFFIC_DIRECTION_102"   "HIGHWAY_SYSTEM_104"
##  [89] "FEDERAL_LANDS_105"       "YEAR_RECONSTRUCTED_106"
##  [91] "DECK_STRUCTURE_TYPE_107" "SURFACE_TYPE_108A"
##  [93] "MEMBRANE_TYPE_108B"      "DECK_PROTECTION_108C"
##  [95] "NATIONAL_NETWORK_110"    "BRIDGE_LEN_IND_112"
##  [97] "SCOUR_CRITICAL_113"      "FUTURE_ADT_114"
##  [99] "YEAR_OF_FUTURE_ADT_115"  "FED_AGENCY"
## [101] "DATE_LAST_UPDATE"        "TYPE_LAST_UPDATE"
## [103] "DEDUCT_CODE"             "PROGRAM_CODE"
## [105] "PROJ_NO"                 "STATUS_WITH_10YR_RULE"
## [107] "SUFFICIENCY_ASTERC"      "SUFFICIENCY_RATING"
## [109] "STATUS_NO_10YR_RULE"     "CAT10"
```

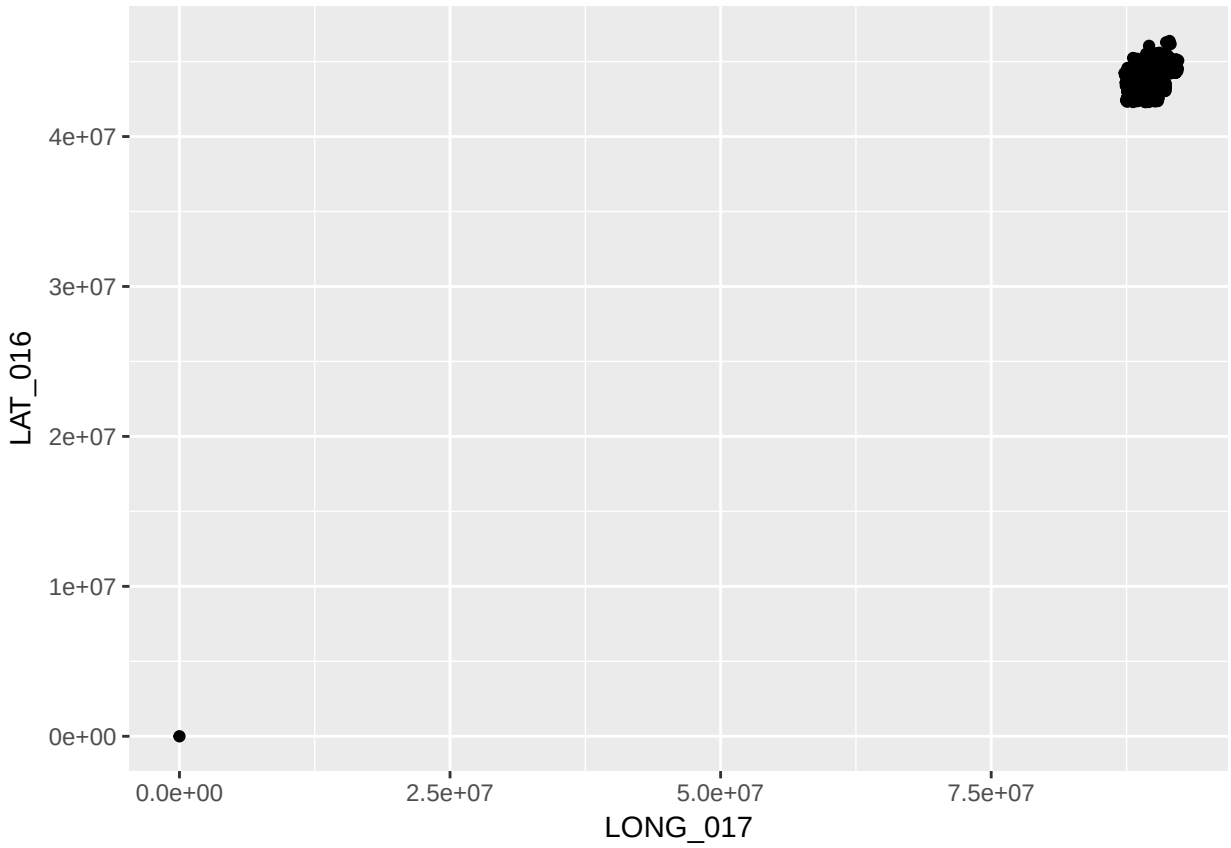
```r
##Keep part of the variables
keep = c("COUNTY_CODE_003", "LAT_016" ,"LONG_017" ,"OWNER_022", "YEAR_BUILT_027",
         "TRAFFIC_LANES_ON_028A", "TRAFFIC_LANES_UND_028B")
x = dat[,match(keep, colnames(dat))]

##Lets see when the bridges are built
ggplot(data = x) +
  geom_bar(mapping = aes(x = YEAR_BUILT_027 ))
```



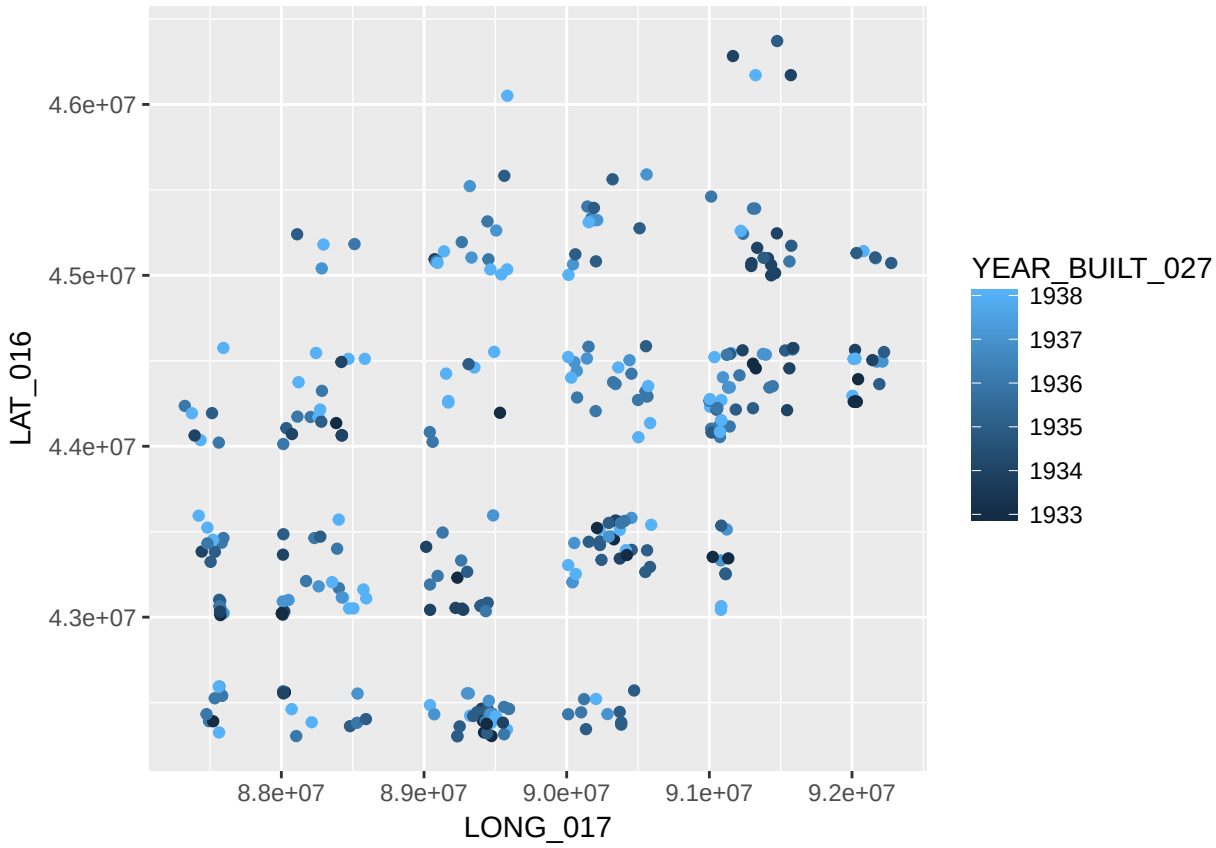

```
##I would like to focus on the bridges built during the Roosevelt's New Deal,
##that is between 1933 to 1938
nd = x %>% filter(YEAR_BUILT_027 > 1932 & YEAR_BUILT_027 < 1939)
ggplot(data = nd) +
  geom_bar(mapping = aes(x = YEAR_BUILT_027))
```



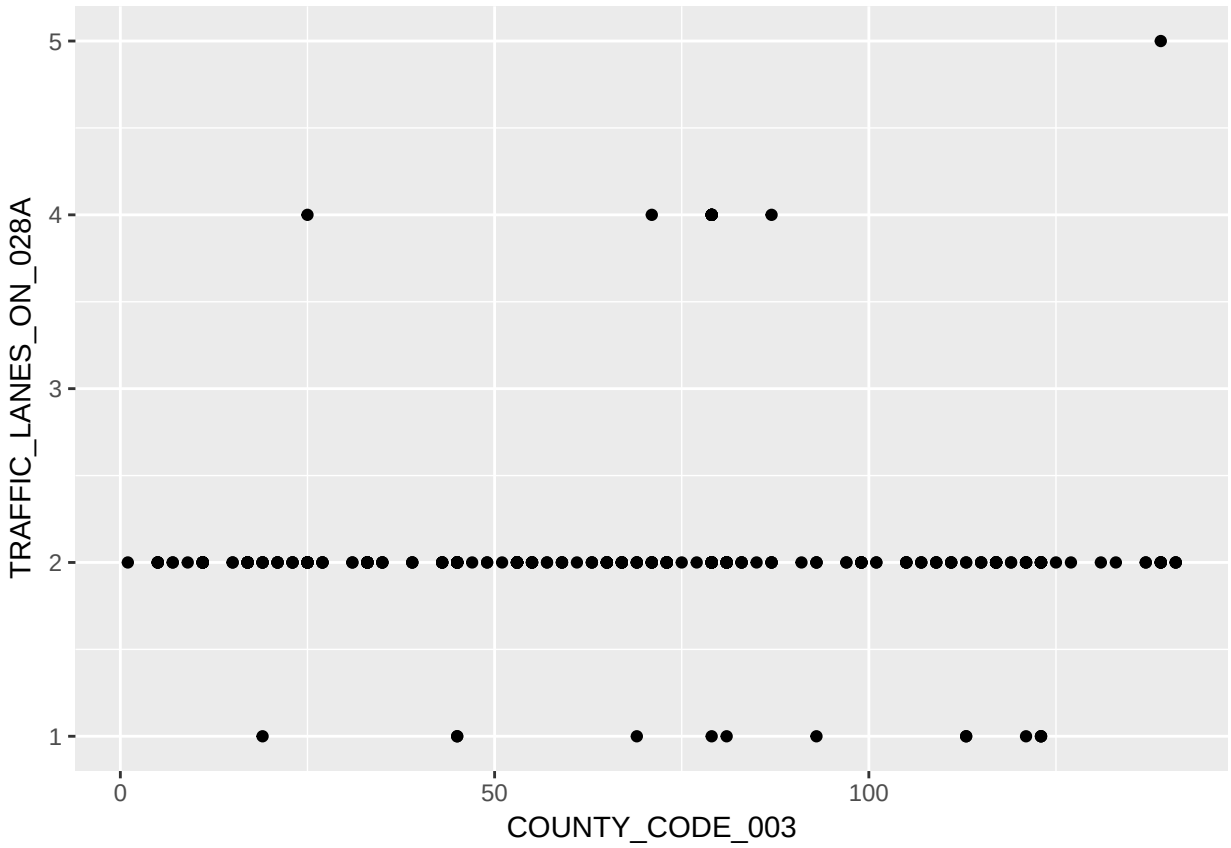

```
##Then I would like to see where these bridges are built
ggplot(data = nd) +
  geom_point(mapping = aes(y = LAT_016, x = LONG_017))
```



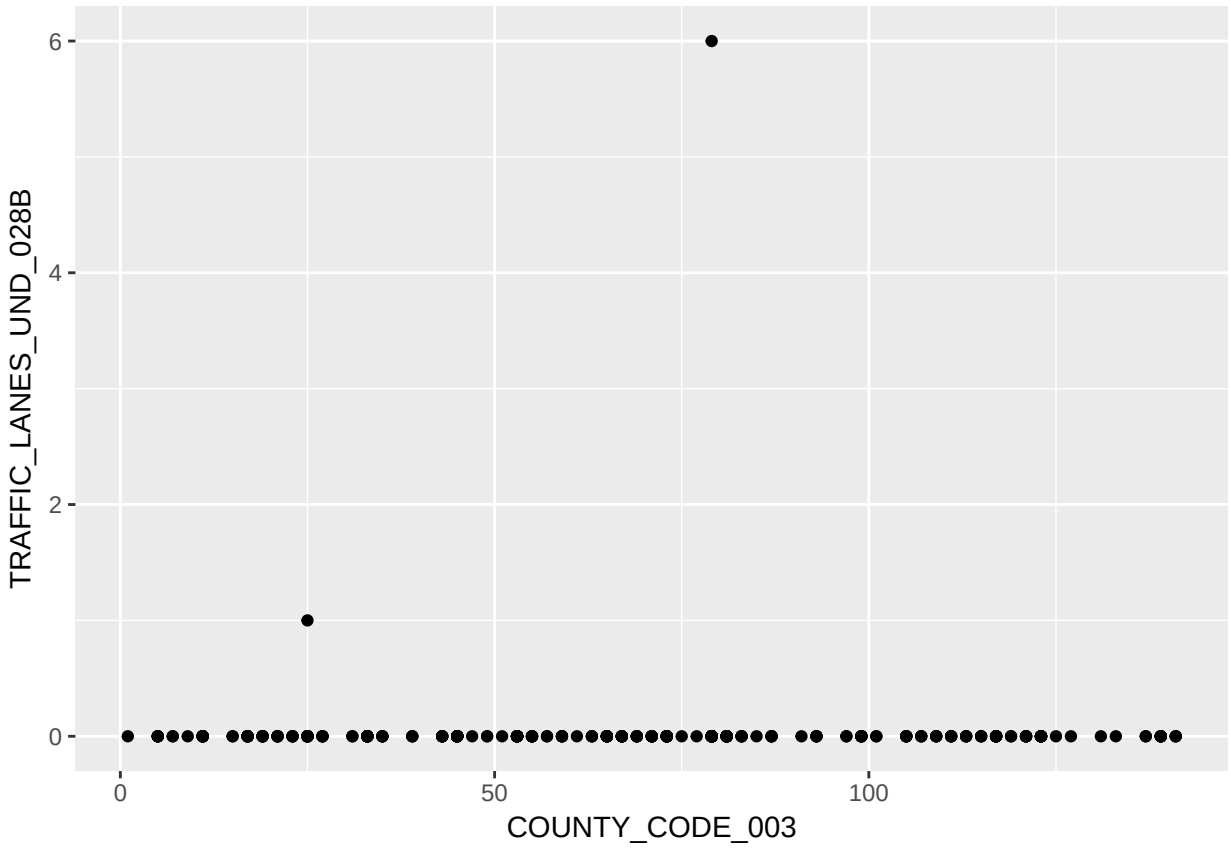

```
#It seems there is an error point at 0 latitude.
nd = nd %>% filter(LAT_016 > 0)
ggplot(data = nd) +
  geom_point(mapping = aes(y = LAT_016, x = LONG_017, color = YEAR_BUILT_027))
```



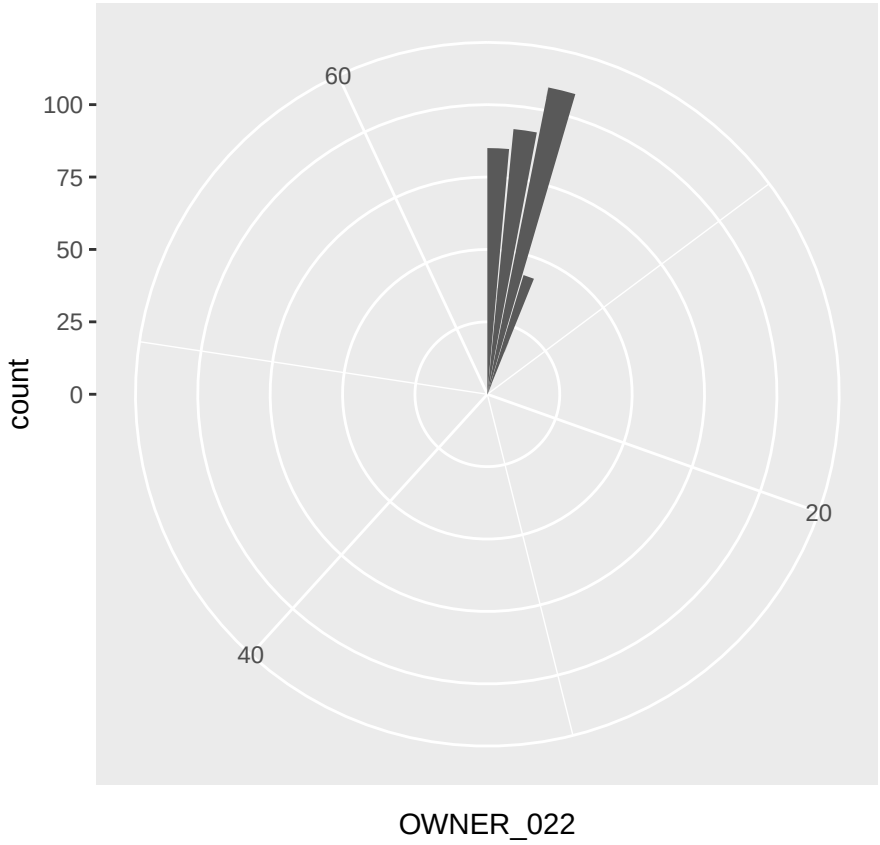

```
#Throught our the period, bridges are built all over Wisconsin.

##Next I would like to see bridges in some particular country have more lanes
ggplot(data = nd) +
  geom_point(mapping = aes(x = COUNTY_CODE_003, y = TRAFFIC_LANES_ON_028A))
```



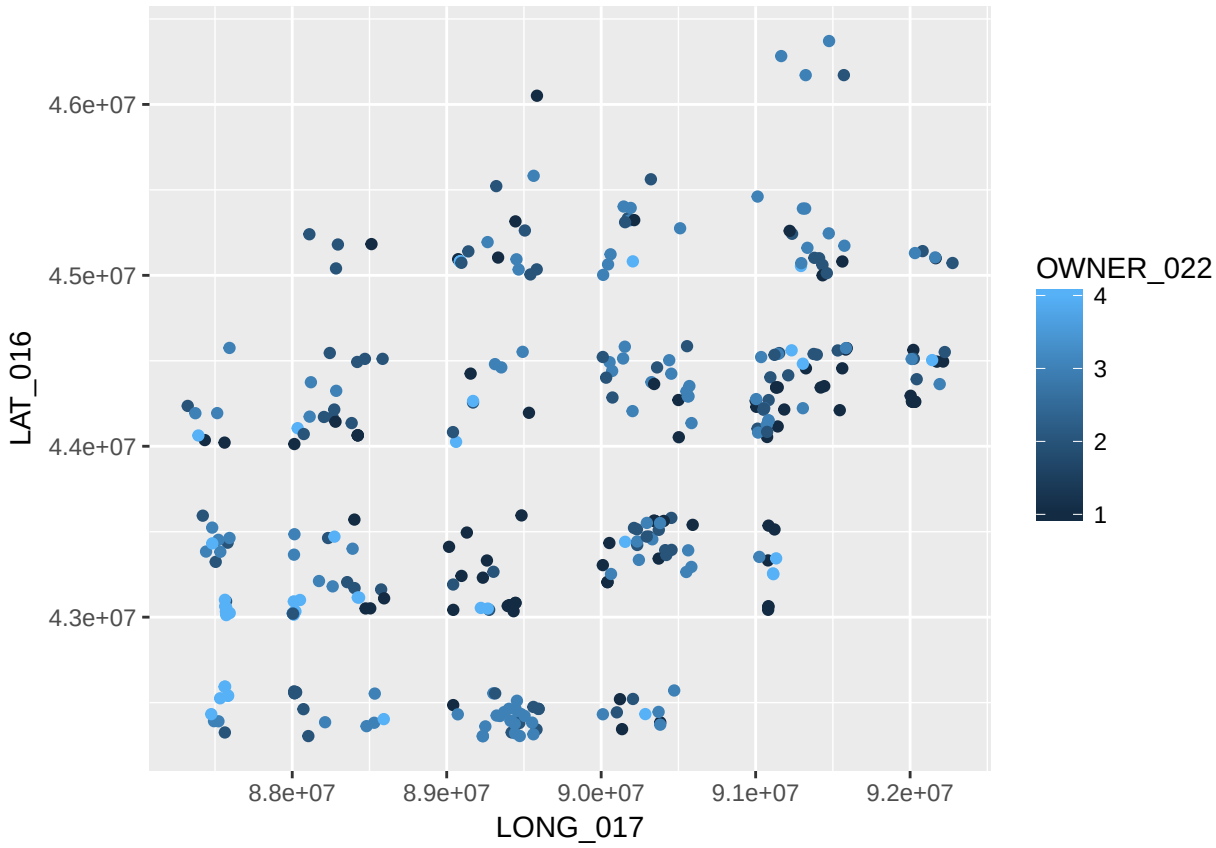

```
ggplot(data = nd) +
  geom_point(mapping = aes(x = COUNTY_CODE_003, y = TRAFFIC_LANES_UND_028B))
```




```
##Last let's take a look who owns these bridges
ggplot(data = nd) +
  geom_bar(mapping = aes(x = OWNER_022)) +
  coord_polar()
```




```
#Most bridges are owned by only four owners.

nd2 = nd %>% filter(OWNER_022 < 10)
ggplot(data = nd2) +
  geom_point(mapping = aes(y = LAT_016, x = LONG_017, color = OWNER_022))
```




```
#There is no specific geographic patterns of the ownership of the bridges.
```