dot_census_explore

Hunter York

9/19/2020

Contents

Introduction

This is an exploratory document detailing the relationship between census and dot occupation codes.

Task 1

Number of DOT occupations nested within each census occupation category

Code setup

The following code cleans and merges the DOT and Census crosswalk file to the complete DOT title set by the first three digits of the DOT code. Codes that have two digits were inferred to have a leading 0, and a quick check seems to validate this.

```
# read in map. This is DOT1939 to 1940 Census map
dot_cen_map <- read_xlsx("../ref/DOT1939_to_CEN1940.xlsx") %>%
  data.table()

# read in occ to occ1940 conversion since the data uses the latter
occ_ooc_conv <- read_xls("../ref/occ_occ1940.xls") %>%
  data.table()

# merge
dot_cen_map <- merge(dot_cen_map, occ_ooc_conv[!is.na(OCC1940),],
                     by.x = "Census1940", by.y = "OCC1940",
                     all.x = T)
dot_cen_map[,Census1940 := OCC]

# clean up messy vars, for 2 digit codes, insert a 0
dot_cen_map[, clean_dot_code := gsub("[^0-9]", "", DOT1939)]
dot_cen_map[nchar(clean_dot_code) == 2,
            clean_dot_code := paste0("0",clean_dot_code)]
dot_cen_map[, Census1940 := gsub("\n", "", Census1940)]
dot_cen_map[, `Titles of 1940 Census` := gsub("\n",
                                              "",
                                              `Titles of 1940 Census` )]

# load complete list of dot codes
dot_complete <- read_xlsx("../ref/DOT_1ed_1939.xlsx",col_names = F) %>%
  data.table() %>% setnames(., c("dot_code", "dot_description"))

# create nested categories using 5 digit dot codes
# and merge onto main file
dot_complete_5 <- dot_complete[nchar(dot_code) == 5]
setnames(dot_complete_5, c("dot_description"),
         c("dot_parent_desc"))
dot_complete[,first_five_dot := substr(dot_code,1,5)]
dot_complete <- merge(dot_complete, dot_complete_5,
                      by.x = c("first_five_dot"),
                      by.y = c("dot_code"))

# create new var of prefix of dot codes to match map
dot_complete[,clean_dot_code := substr(gsub("[^0-9]", "", dot_code),1,3)]

# do a cartesian merge to allow all combos of of dot codes
# to match potential census codes
mg_cen_dot_map <- merge(dot_complete, dot_cen_map, all.y = T, allow.cartesian = T, by = "clean_dot_code

# also  only include most granular dot codes
```

```
mg_cen_dot_map <- mg_cen_dot_map[nchar(dot_code) != 5]

# drop nas from output. These are header rows in the dot_cen_map, so
# nothing is really dropped
mg_cen_dot_map <- mg_cen_dot_map[!is.na(clean_dot_code) &
                                   clean_dot_code != "<NA>" &
                                   clean_dot_code != ""]

# only preserve important columns
mg_cen_dot_map <- mg_cen_dot_map[,.(clean_dot_code, dot_code, dot_description, `Titles of 1940 Census`,
```

Drawbacks

A quick face validity seems to indicate this is an appropriate merge for some Census codes and an inappropriate merge for others. For example, I will show the Census category "Artists and art teachers" which is Census1940 == "001" in the crosswalk file. As you can see, this merge does a good job capturing all kinds of artists and art teachers, (there's 95 matches).

```
head(mg_cen_dot_map[Census1940 == "001"])
```

```
##    clean_dot_code dot_code                      dot_description
## 1:            004  0-04.01         Painter IV (profess. & kin.)
## 2:            004  0-04.01     Cover Designer (profess. & kin.)
## 3:            004  0-04.01    Painter, Crayon (profess. & kin.)
## 4:            004  0-04.01    Painter, Fresco (profess. & kin.)
## 5:            004  0-04.01     Painter, Genre (profess. & kin.)
## 6:            004  0-04.01 Painter, Landscape (professs. & kin.)
##       Titles of 1940 Census Census1940                          dot_parent_desc
## 1: Artists and art teachers        001 ARTISTS, SCULPTORS, AND TEACHERS OF ART
## 2: Artists and art teachers        001 ARTISTS, SCULPTORS, AND TEACHERS OF ART
## 3: Artists and art teachers        001 ARTISTS, SCULPTORS, AND TEACHERS OF ART
## 4: Artists and art teachers        001 ARTISTS, SCULPTORS, AND TEACHERS OF ART
## 5: Artists and art teachers        001 ARTISTS, SCULPTORS, AND TEACHERS OF ART
## 6: Artists and art teachers        001 ARTISTS, SCULPTORS, AND TEACHERS OF ART
##                  Occupation
## 1: Artists and art teachers
## 2: Artists and art teachers
## 3: Artists and art teachers
## 4: Artists and art teachers
## 5: Artists and art teachers
## 6: Artists and art teachers
```

Compare this to Census1940 == "134" which is the Census category "Postmasters". This gets matched to a number of government service professions. Perhaps of which the only relevant one might be "Censor II." (There are a total of 17 matches). I'm afraid carrying out the analysis without a step to weed out these bad matches would be a mistake at least if we're to interpret the number of occupation codes nested within each census crosswalk as jobs. Potential solutions might be to explore the extent to which this phenomenon is limited to trades and the extent to which this represents true hierarchical structures? I'd imagine trades will have a more proper nesting structure such that a census code will capture a number of DOT codes that are encapsulated within it, whereas hierarchical agency census codes might map to DOT codes that capture a number of irrelevant professions. This might also be a good place to incorporate industry codings.

```
tail(mg_cen_dot_map[Census1940 == "134"])
```

```
##    clean_dot_code dot_code         dot_description Titles of 1940 Census
```

```
## 1:            094  0-94.94 game and fish protector          Postmasters
## 2:            094  0-94.94    game and fish warden          Postmasters
## 3:            094  0-94.94 Fish Warden (gov. ser.)          Postmasters
## 4:            094  0-94.94 Game Warden (gov. ser.)          Postmasters
## 5:            094  0-94.94          game protector          Postmasters
## 6:            094  0-94.95   Censor II (gov. ser.)          Postmasters
##    Census1940            dot_parent_desc  Occupation
## 1:        134 PUBLIC OFFICIALS, N. E. C. Postmasters
## 2:        134 PUBLIC OFFICIALS, N. E. C. Postmasters
## 3:        134 PUBLIC OFFICIALS, N. E. C. Postmasters
## 4:        134 PUBLIC OFFICIALS, N. E. C. Postmasters
## 5:        134 PUBLIC OFFICIALS, N. E. C. Postmasters
## 6:        134 PUBLIC OFFICIALS, N. E. C. Postmasters
```

Visualizing it anyway

Regardless of the above problem, I'm moving forward with the analysis for now. Below is a graph of the number of DOT codes that match each Census code using the crosswalk as it currently stands.

All merges are done using the Census variables, some of which can be mapped to the several Census Occupation title. I've dropped "988", "Not specified".

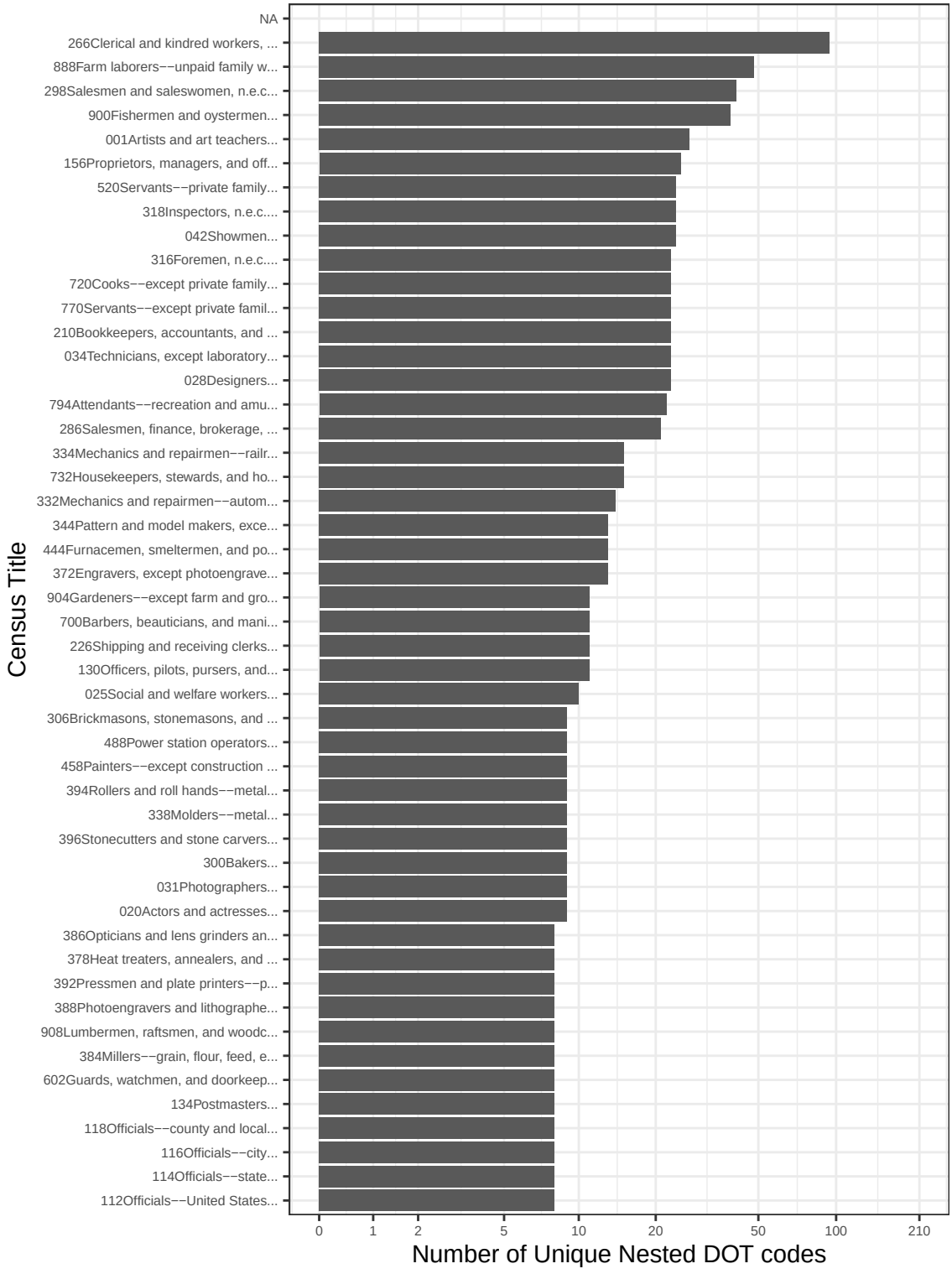
"occ" and "occ1950" do not match 1:1.

All graphs use the parent dot category concatenated with the Census code variable on the y axis.

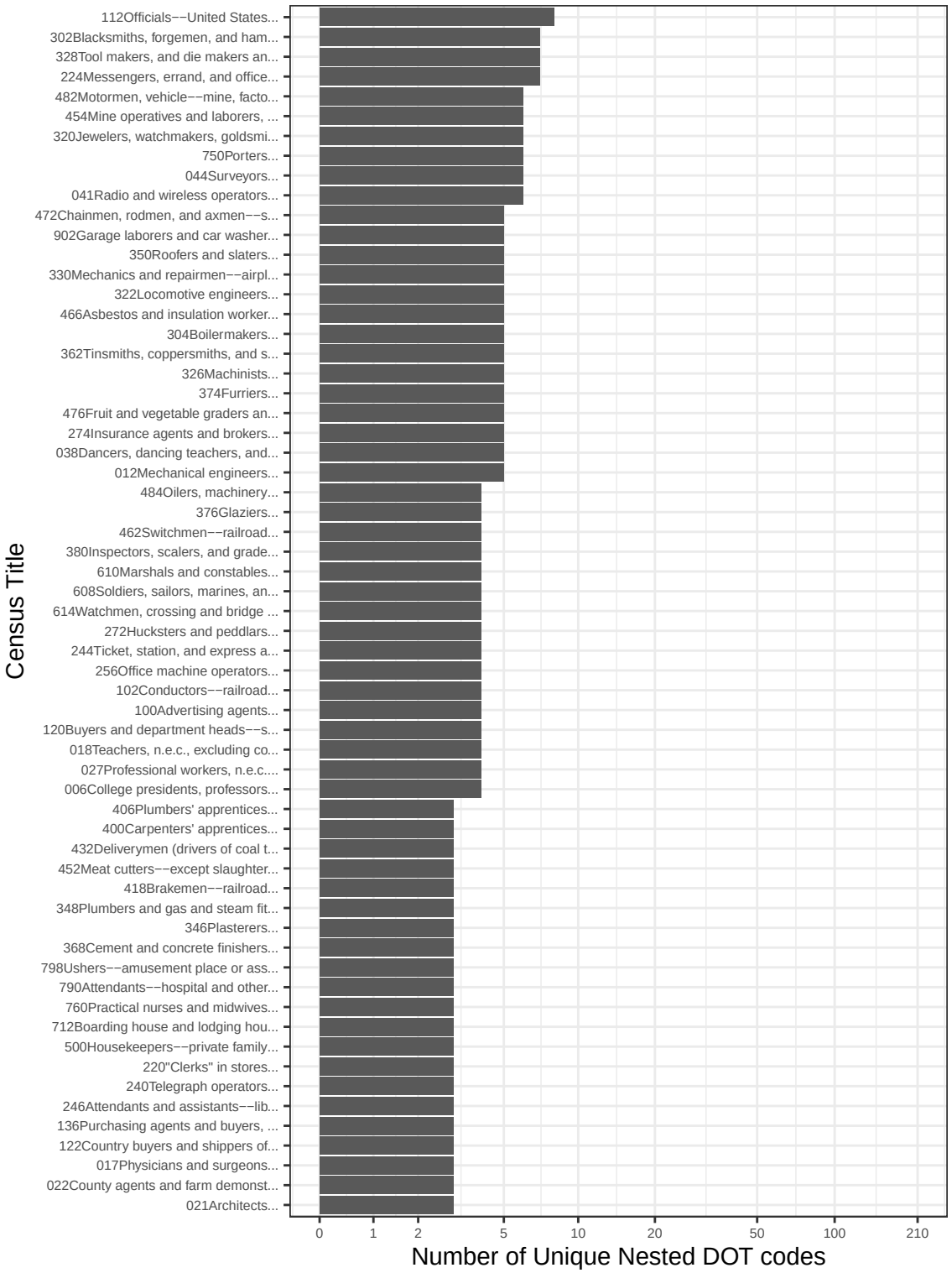
Some dot_codes map to several dot descriptions. "1-01.02" is "Bookkeeper II (clearical) and"general bookkeeper" amongst others. When I tabulate the number of dot codes, I'm tabulating each dot code, not the different dot_descriptions. When a census category matches multiple 3-digit dot codes, I include all nested dot codes.

```
### THIS CAN BE TOGGLED WITH LENGTH(UNIQUE(DOT DESCRIPTIONS))
# count unique codes by census category
cen_dot_code_counts<- mg_cen_dot_map[,.(num_dot_codes = length(unique(dot_code)),
                                        num_dot_titles = length(unique(dot_description))),
                                     by = .(Census1940, Occupation)]

# create variable for graphing
cen_dot_code_counts[, var_title := paste0(Census1940,
                                          substr(Occupation,
                                                 1,30), "...")]

# Remove unmapped
cen_dot_code_counts <- cen_dot_code_counts[Census1940 != "988"]

# sort by freq
cen_dot_code_counts[,var_title := factor(var_title,

                                         levels = cen_dot_code_counts[order(num_dot_codes),
                                                                      var_title])]

cen_dot_code_counts <- cen_dot_code_counts[order(num_dot_codes)]
```

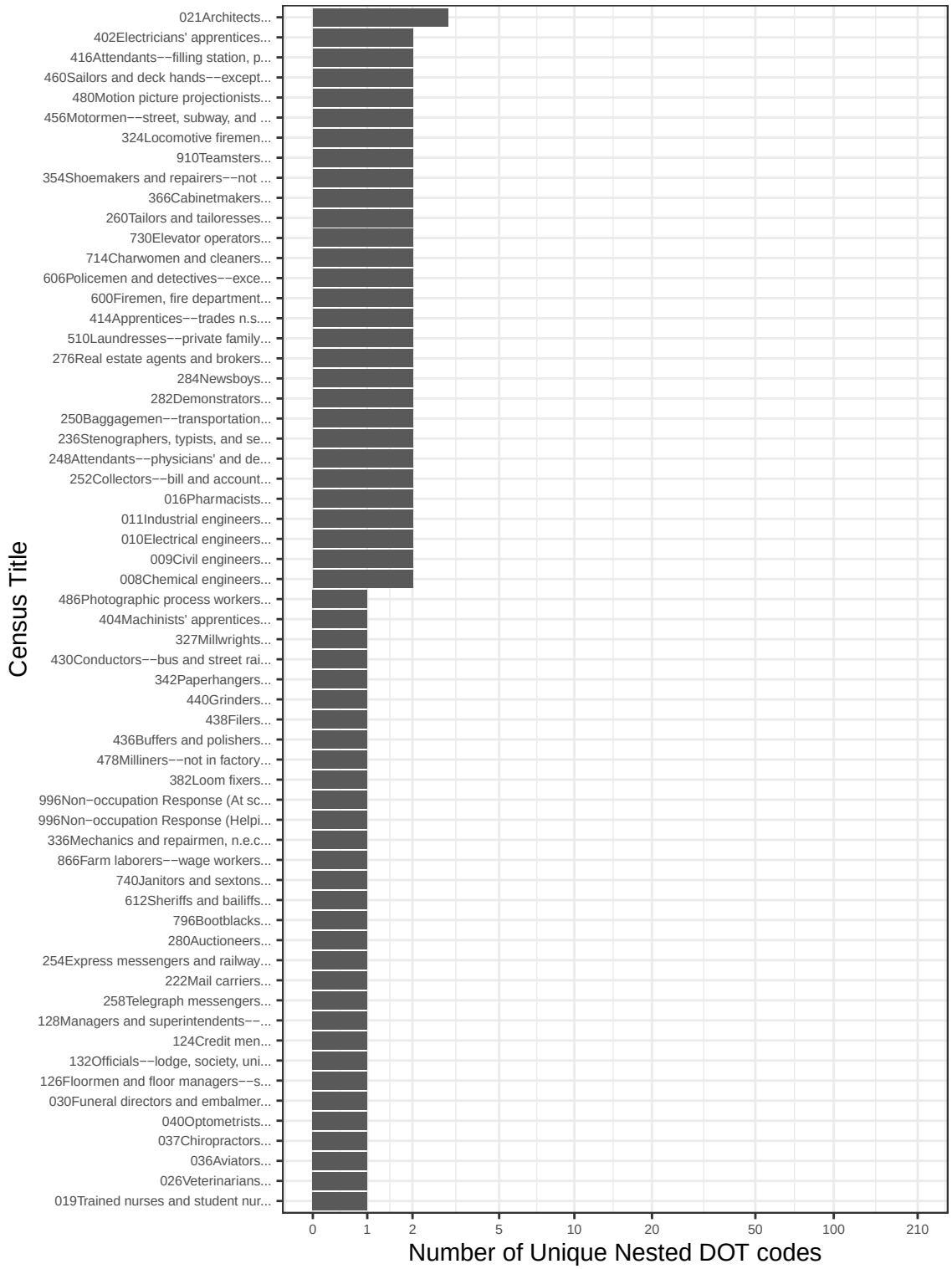
Results: Dot Codes per census code



Rows 120 to 180

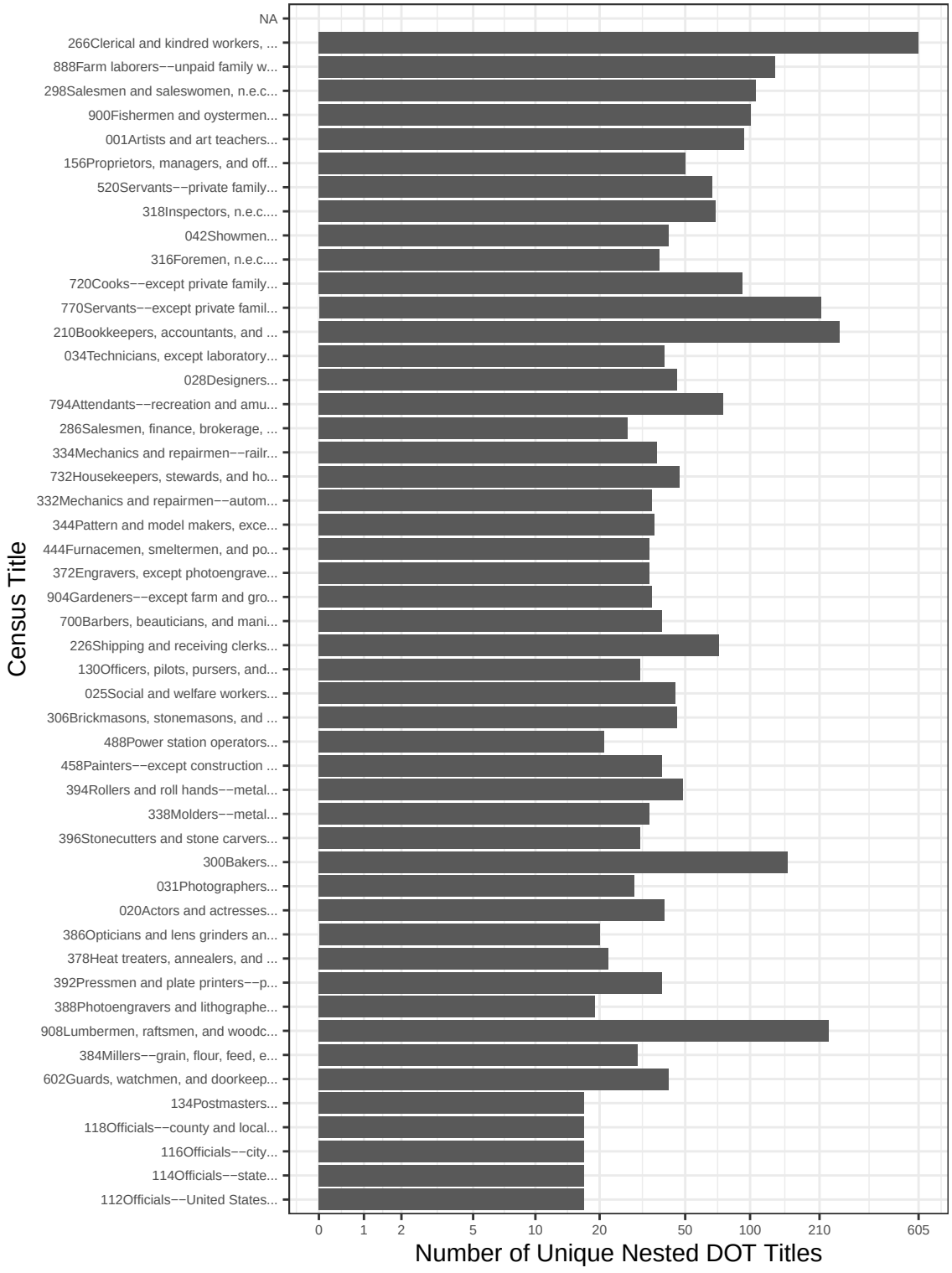


Rows 60 to 120

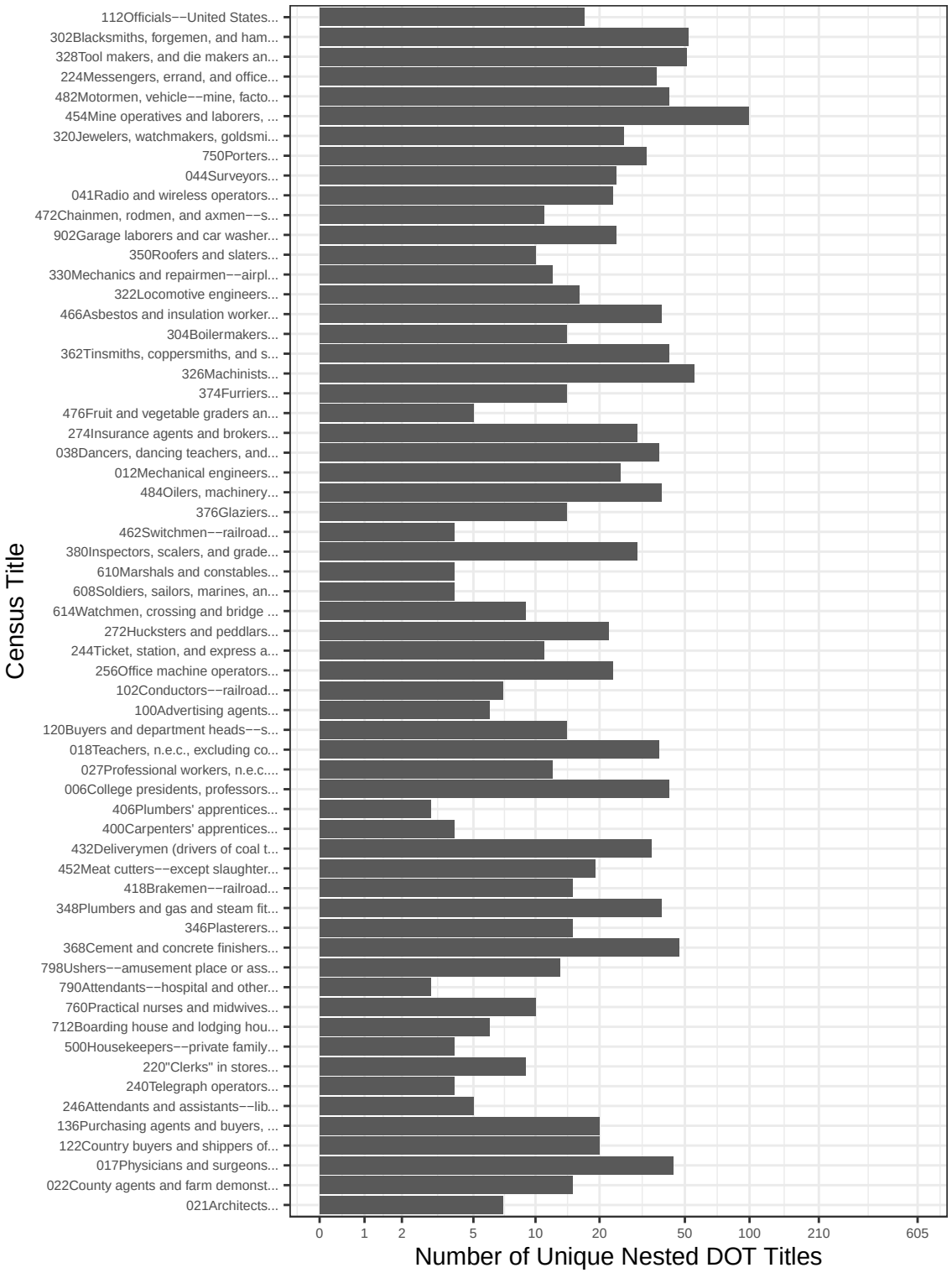

Census Title
021Architects...
402Electricians' apprentices...
416Attendants−−filling station, p...
460Sailors and deck hands−−except...
480Motion picture projectionists...
456Motormen−−street, subway, and ...
324Locomotive firemen...
910Teamsters...
354Shoemakers and repairers−−not ...
366Cabinetmakers...
260Tailors and tailoresses...
730Elevator operators...
714Charwomen and cleaners...
606Policemen and detectives−−exce...
600Firemen, fire department...
414Apprentices−−trades n.s....
510Laundresses−−private family...
276Real estate agents and brokers...
284Newsboys...
282Demonstrators...
250Baggagemen−−transportation...
236Stenographers, typists, and se...
248Attendants−−physicians' and de...
252Collectors−−bill and account...
016Pharmacists...
011Industrial engineers...
010Electrical engineers...
009Civil engineers...
008Chemical engineers...
486Photographic process workers...
404Machinists' apprentices...
327Millwrights...
430Conductors−−bus and street rai...
342Paperhangers...
440Grinders...
438Filers...
436Buffers and polishers...
478Milliners−−not in factory...
382Loom fixers...
996Non−occupation Response (At sc...
996Non−occupation Response (Helpi...
336Mechanics and repairmen, n.e.c...
866Farm laborers−−wage workers...
740Janitors and sextons...
612Sheriffs and bailiffs...
796Bootblacks...
280Auctioneers...
254Express messengers and railway...
222Mail carriers...
258Telegraph messengers...
128Managers and superintendents−−...
124Credit men...
132Officials−−lodge, society, uni...
126Floormen and floor managers−−s...
030Funeral directors and embalmer...
040Optometrists...
037Chiropractors...
036Aviators...
026Veterinarians...
019Trained nurses and student nur...
0
1
2
5
10
20
50
100
210
Number of Unique Nested DOT codes

Rows 0 to 60

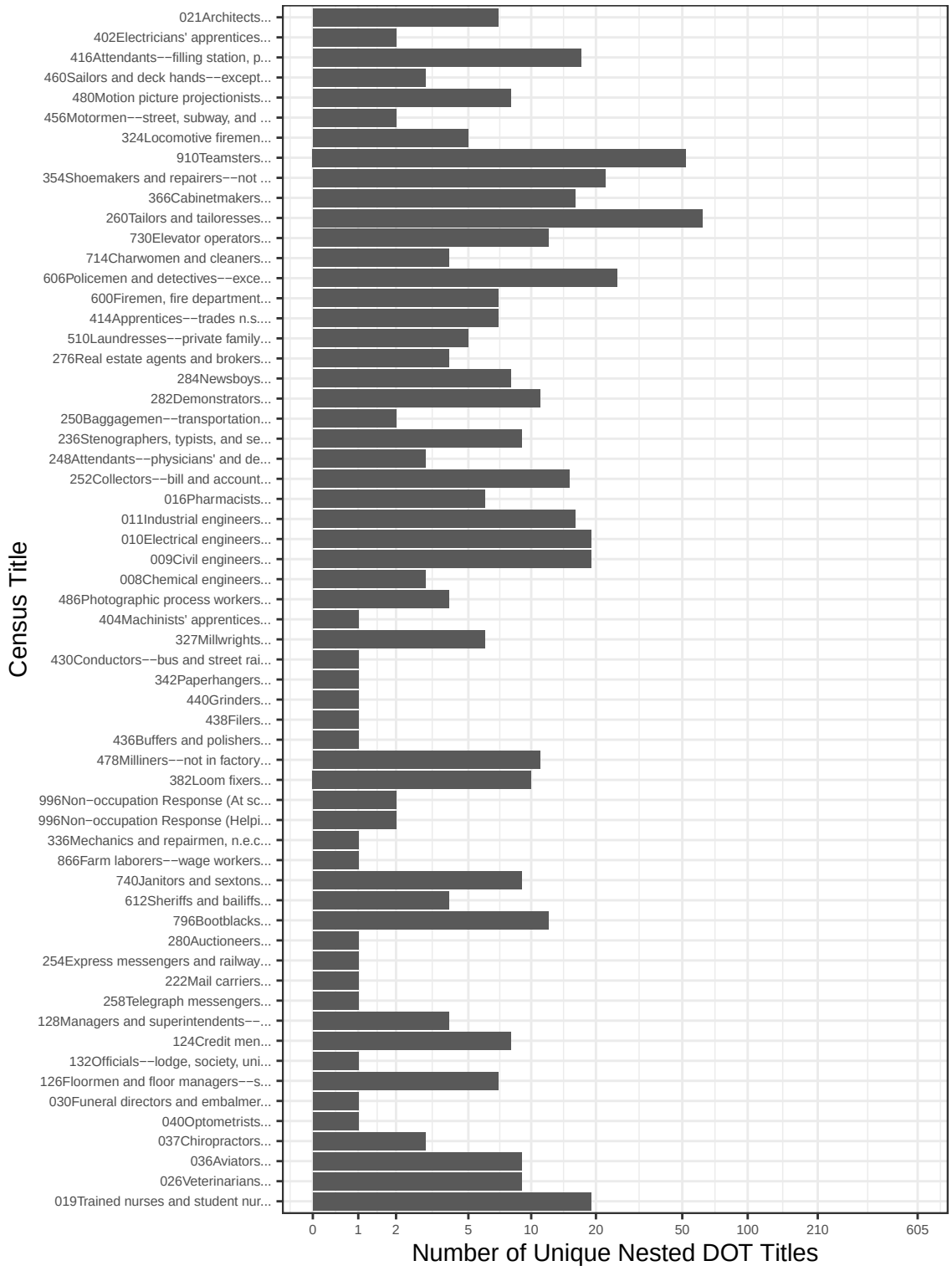
Results: Dot Titles per census code



Rows 120 to 180



Rows 60 to 120


Census Title
021Architects...
402Electricians' apprentices...
416Attendants−−filling station, p...
460Sailors and deck hands−−except...
480Motion picture projectionists...
456Motormen−−street, subway, and ...
324Locomotive firemen...
910Teamsters...
354Shoemakers and repairers−−not ...
366Cabinetmakers...
260Tailors and tailoresses...
730Elevator operators...
714Charwomen and cleaners...
606Policemen and detectives−−exce...
600Firemen, fire department...
414Apprentices−−trades n.s....
510Laundresses−−private family...
276Real estate agents and brokers...
284Newsboys...
282Demonstrators...
250Baggagemen−−transportation...
236Stenographers, typists, and se...
248Attendants−−physicians' and de...
252Collectors−−bill and account...
016Pharmacists...
011Industrial engineers...
010Electrical engineers...
009Civil engineers...
008Chemical engineers...
486Photographic process workers...
404Machinists' apprentices...
327Millwrights...
430Conductors−−bus and street rai...
342Paperhangers...
440Grinders...
438Filers...
436Buffers and polishers...
478Milliners−−not in factory...
382Loom fixers...
996Non−occupation Response (At sc...
996Non−occupation Response (Helpi...
336Mechanics and repairmen, n.e.c...
866Farm laborers−−wage workers...
740Janitors and sextons...
612Sheriffs and bailiffs...
796Bootblacks...
280Auctioneers...
254Express messengers and railway...
222Mail carriers...
258Telegraph messengers...
128Managers and superintendents−−...
124Credit men...
132Officials−−lodge, society, uni...
126Floormen and floor managers−−s...
030Funeral directors and embalmer...
040Optometrists...
037Chiropractors...
036Aviators...
026Veterinarians...
019Trained nurses and student nur...
0
1
2
5
10
20
50
100
210
605
Number of Unique Nested DOT Titles

Rows 0 to 60

Task II - How does this correlate with within-occupation heterogeneity in earnings?

Setup

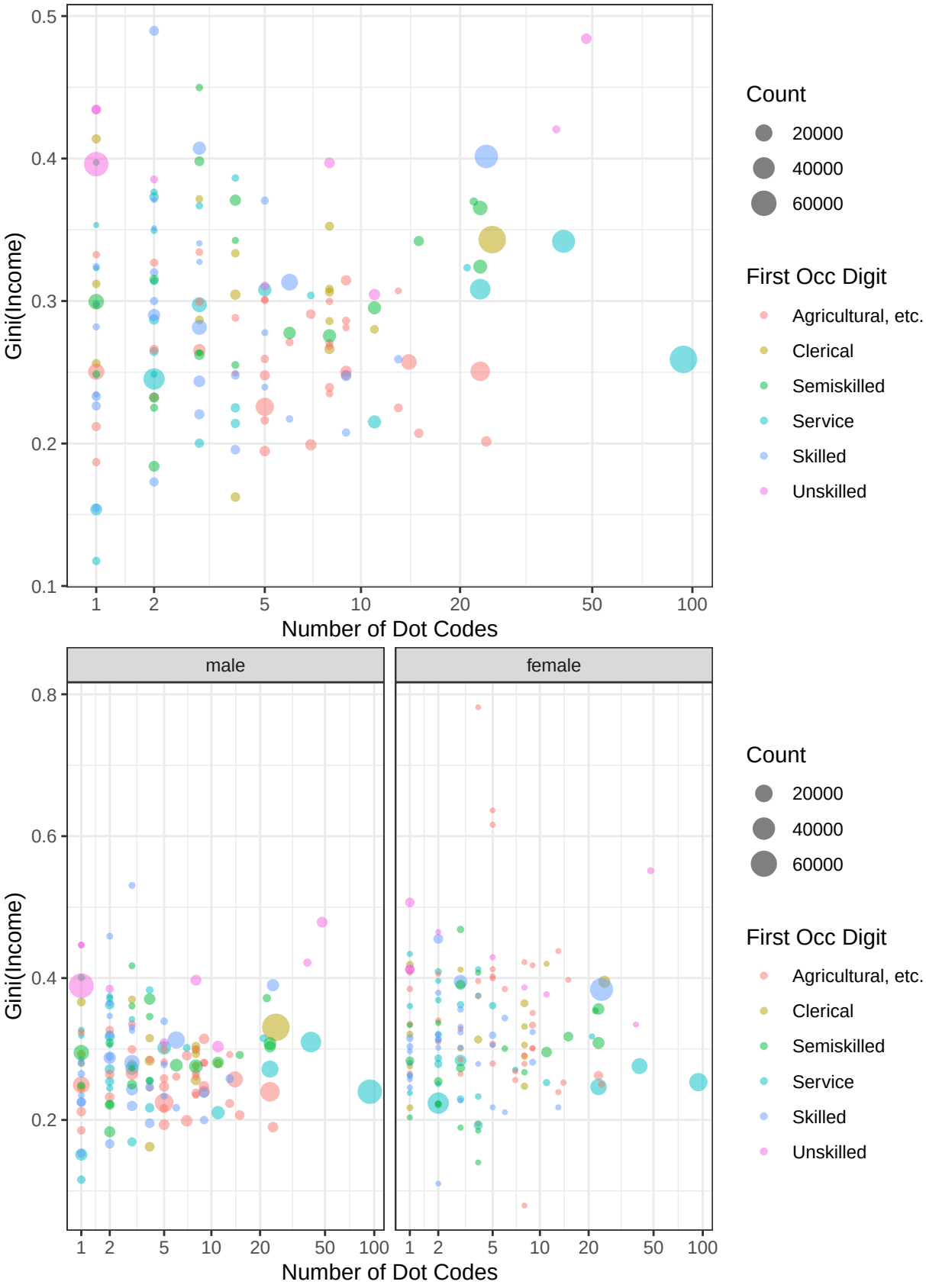
The following data use a 5% sample of the 1940 census.

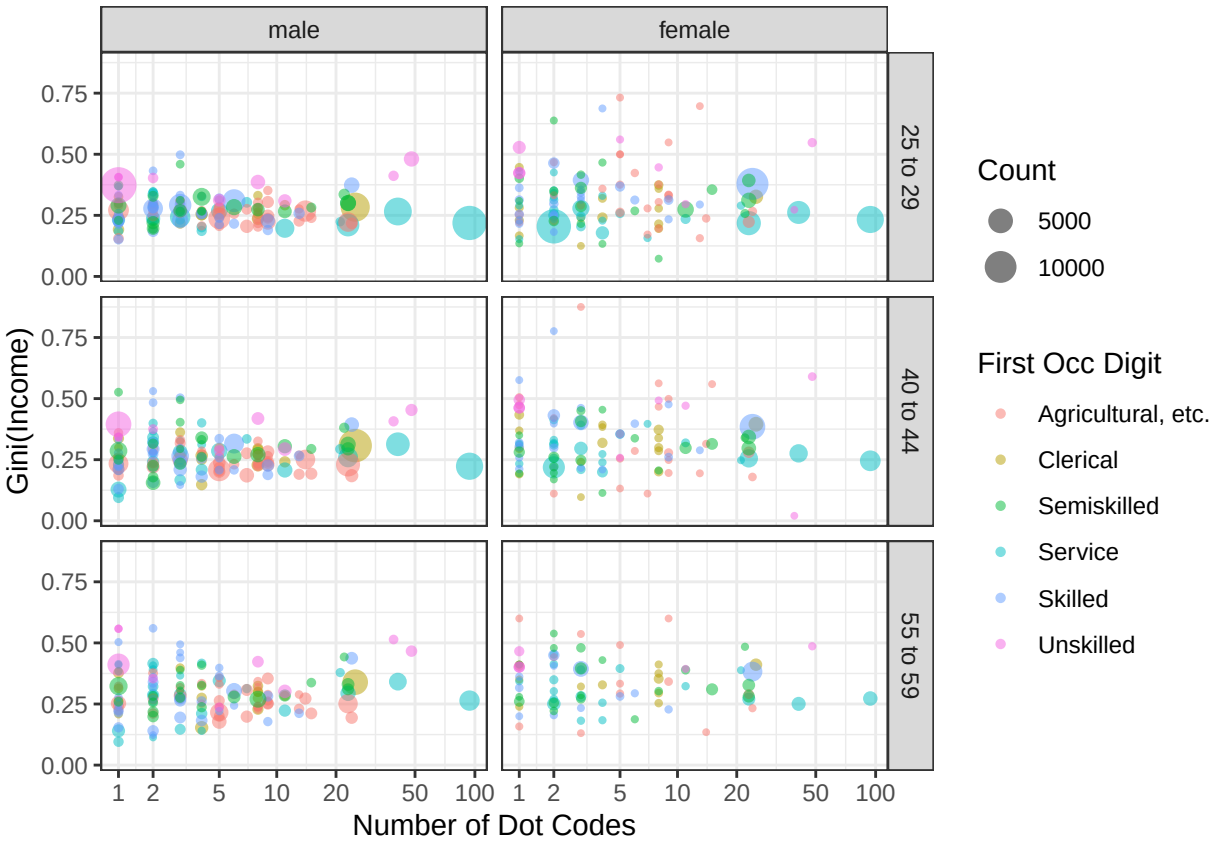
For simplicity's sake, I'm subsetting to current workers who aren't in school in the working age population 25-64 and who work at least 30 hour weeks on average. I'm also dropping the many many rows that have people reported as working but as earning nothing (20% of the sample).

```
# load data and save as r object for faster loading
#census_1940 <- data.table(read.dta13("inputs/usa_00003.dta"))
#saveRDS(census_1940, "inputs/usa_00003.rds")

# load data from saved rds
census_1940 <- readRDS("../inputs/usa_00003.rds")

# only keep relevant vars for now
census_1940 <- census_1940[,.(serial, hhwt, region, statefip, urban,
                              metro, farm,
                              multgen, pernum, perwt,sex,age,agemonth, race,hispan, bpl,
                              school, higraded, empstat, empstatd, labforce, occ,
                              occ1950, ind, ind1950, classwkr, classwkrd, wkswork1,
                              wkswork2, hrswork1, hrswork2, durunemp, uocc, uocc95,
                              uind, uclasswk, incwage, incnonwg, occscore, sei, presgl,
                              erscor50, edscor50, npboss50, migrate5, migrate5d)]

# subset
census_1940 <- census_1940[empstat == "employed" &
                             school == "no, not in school" &
                             age %in% 25:64 &
                             hrswork1 >= 30 &
                             incwage != 999998 &
                             incwage > 0]

# derive a income/hr worked per week var
census_1940[, income_per_hr := incwage/hrswork1]

# derive a 5-year age var
census_1940[, age_cat := paste0(floor(as.numeric(as.character(age))/5)*5,
                                " to ",
                                floor(as.numeric(as.character(age))/5)*5 + 4)]

# make occ var char
census_1940[, occ := as.character(occ)]

# make first dig var
census_1940[, first_dig := substr(occ, 1,1)]

# make map for first dig categories
map <- data.table(first_dig = as.character(0:9),
                  occ_categ = c("Professional", "Clerical", "Service",
                                "Agricultural, etc.", "Skilled", "Skilled",
                                "Semiskilled", "Semiskilled", "Unskilled",
```

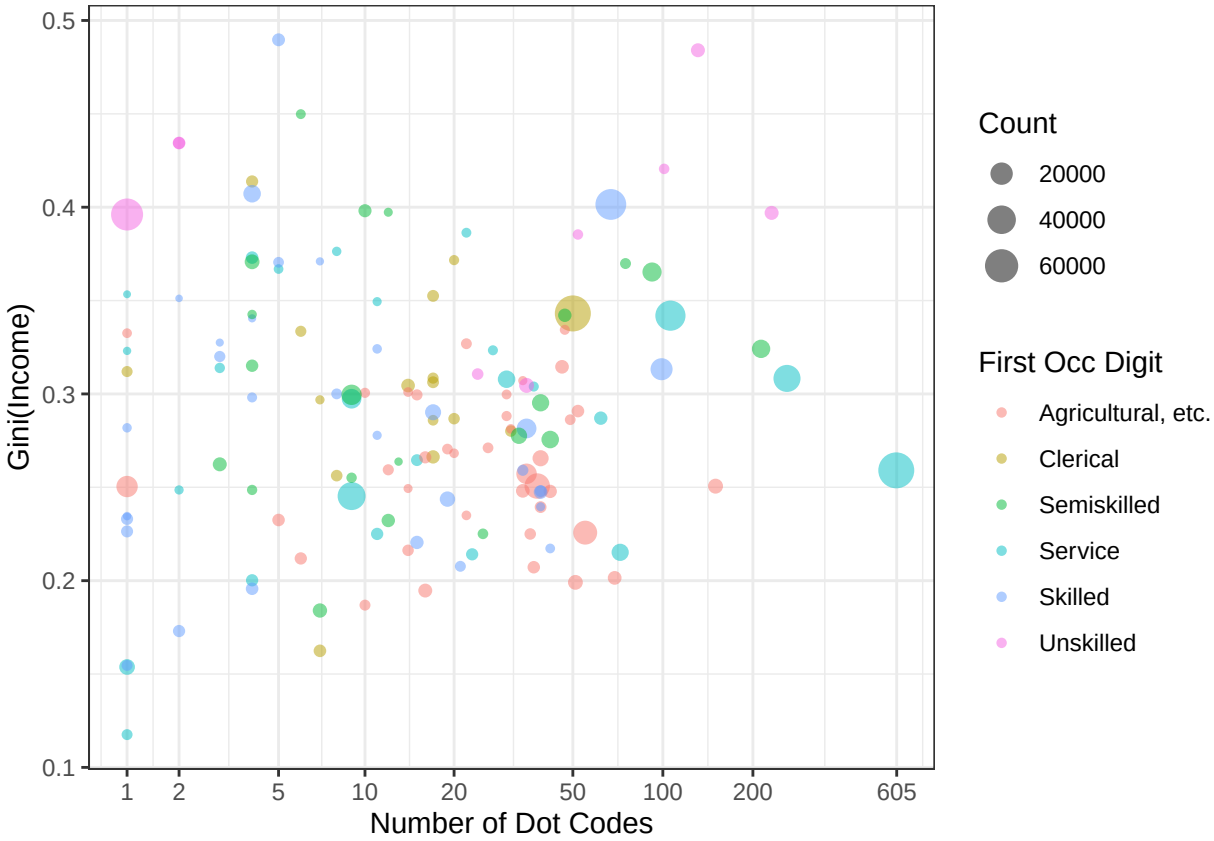
```
                                    "Unskilled"))
census_1940 <- merge(census_1940, map, by = "first_dig")

# calculate wage income gini coeff by occ (and other)
gini_occ <- census_1940[,.(inc_gini = Gini(incwage),
                           sd_inc = sd(incwage),                                avg_inc_per_hr = mean(income_

                           inc_per_hr_gini = Gini(income_per_hr),
                           number = .N),
                         by = .(occ,occ_categ)]
gini_occ_sex <- census_1940[,.(inc_gini = Gini(incwage),
                           inc_per_hr_gini = Gini(income_per_hr),                          avg_inc_per_

                             sd_inc = sd(incwage),
                           number = .N),
                           by = .(occ, occ_categ,sex)]
gini_occ_sex_age <- census_1940[,.(inc_gini = Gini(incwage),
                           inc_per_hr_gini = Gini(income_per_hr),                          avg_inc_per_

                             sd_inc = sd(incwage),
                           number = .N),
                           by = .(occ,occ_categ, sex,age_cat)]

gini_occ <- merge(gini_occ, cen_dot_code_counts, by.x = "occ", by.y = "Census1940")
gini_occ_sex <- merge(gini_occ_sex, cen_dot_code_counts, by.x = "occ", by.y = "Census1940")
gini_occ_sex_age <- merge(gini_occ_sex_age, cen_dot_code_counts, by.x = "occ", by.y = "Census1940")
```

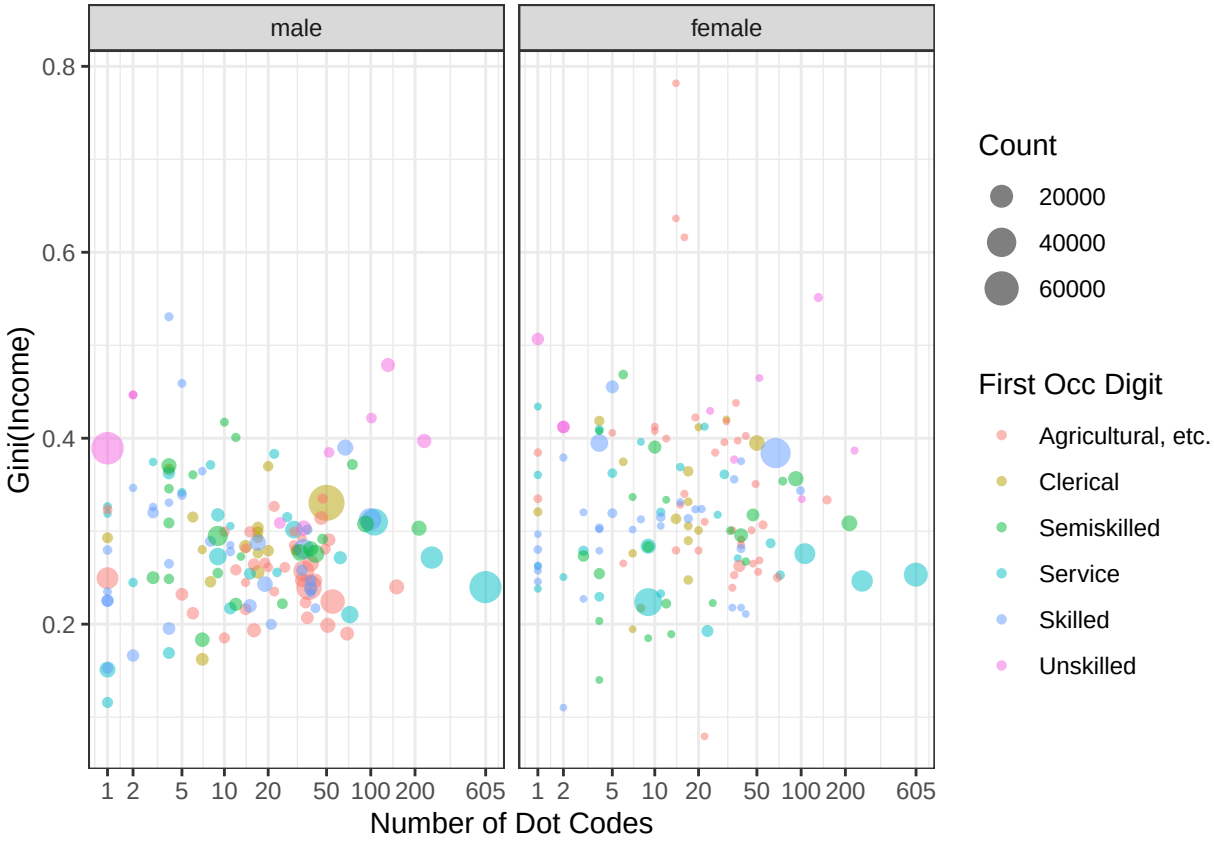
Graph Income Inequality against number of DOT/Census Occ Category

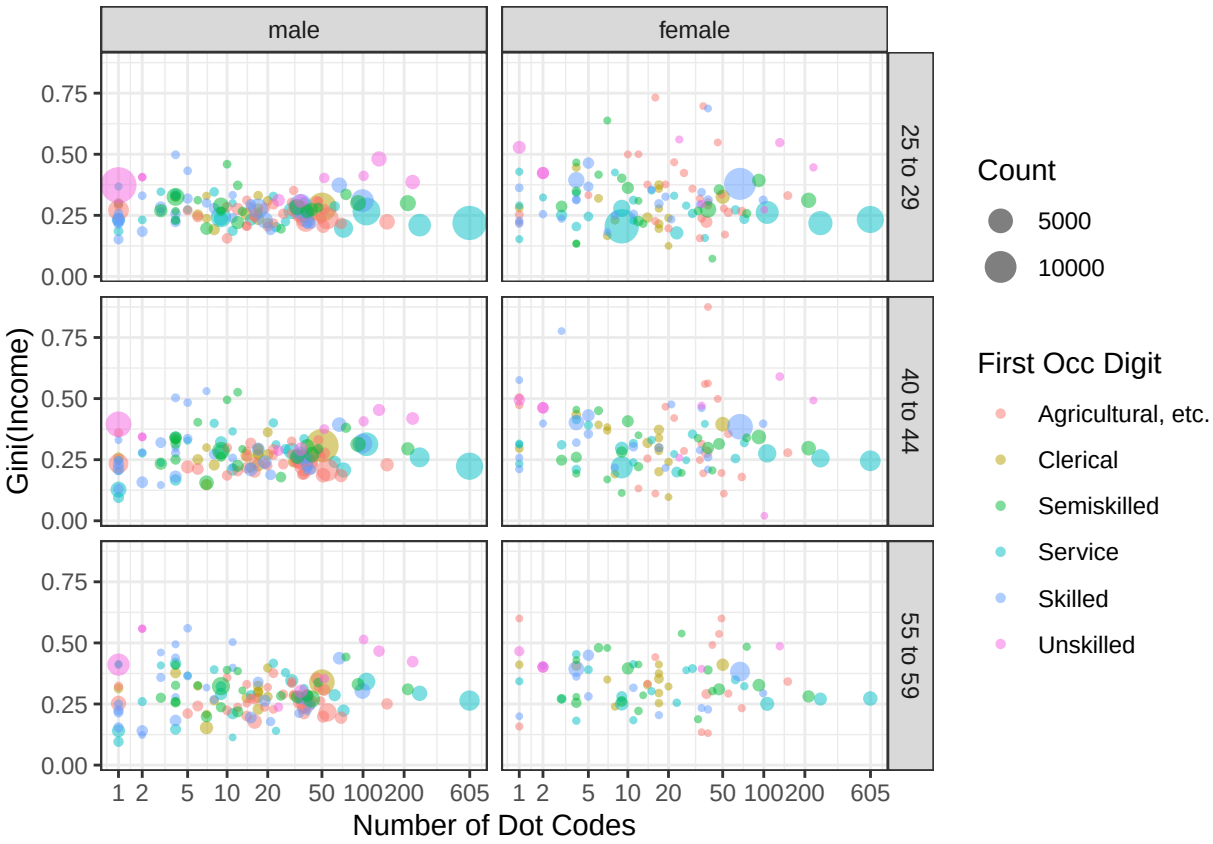



male
female
25 to 29
40 to 44
55 to 59
Gini(Income)
0.00
0.25
0.50
0.75
1
2
5
10
20
50
100
Number of Dot Codes
Count
5000
10000
First Occ Digit
Agricultural, etc.
Clerical
Semiskilled
Service
Skilled
Unskilled

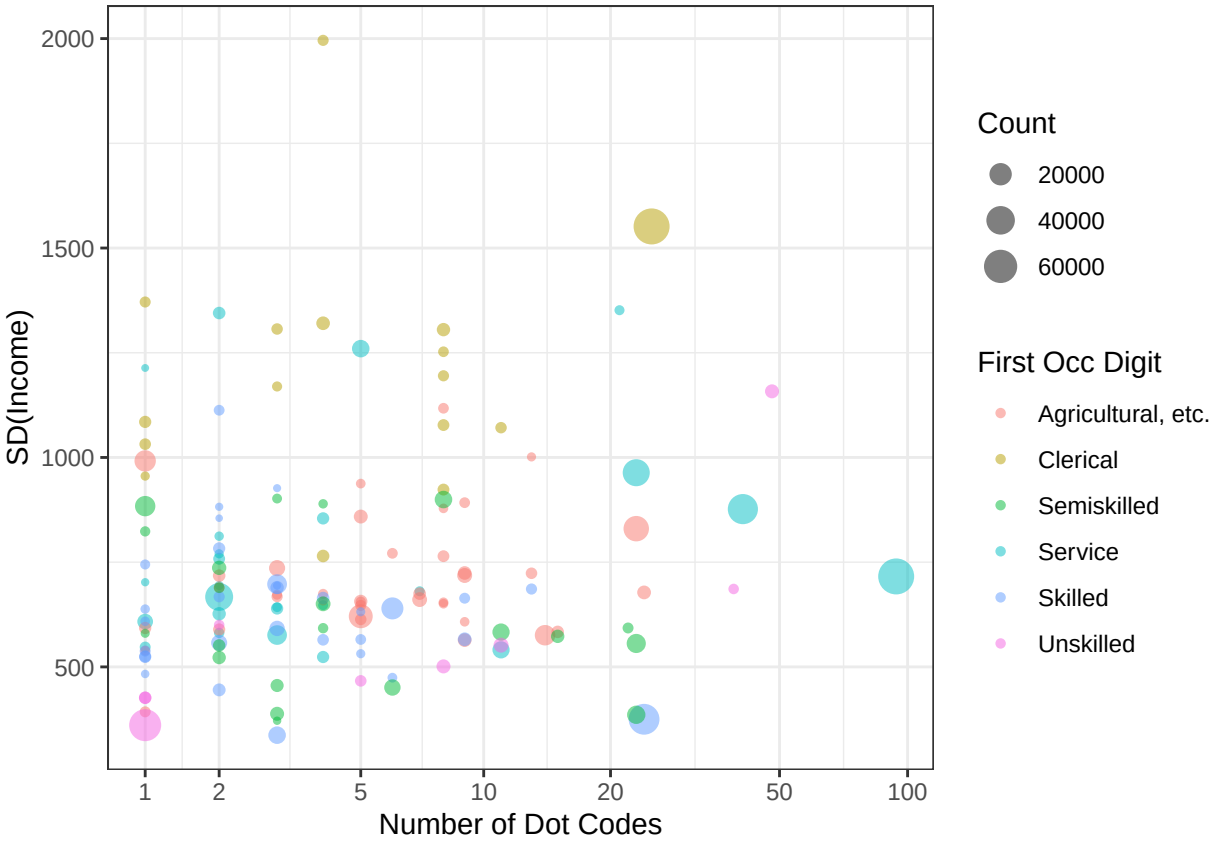
Graph Income Inequality against number of DOT Titles instead of codes/Census Occ Category

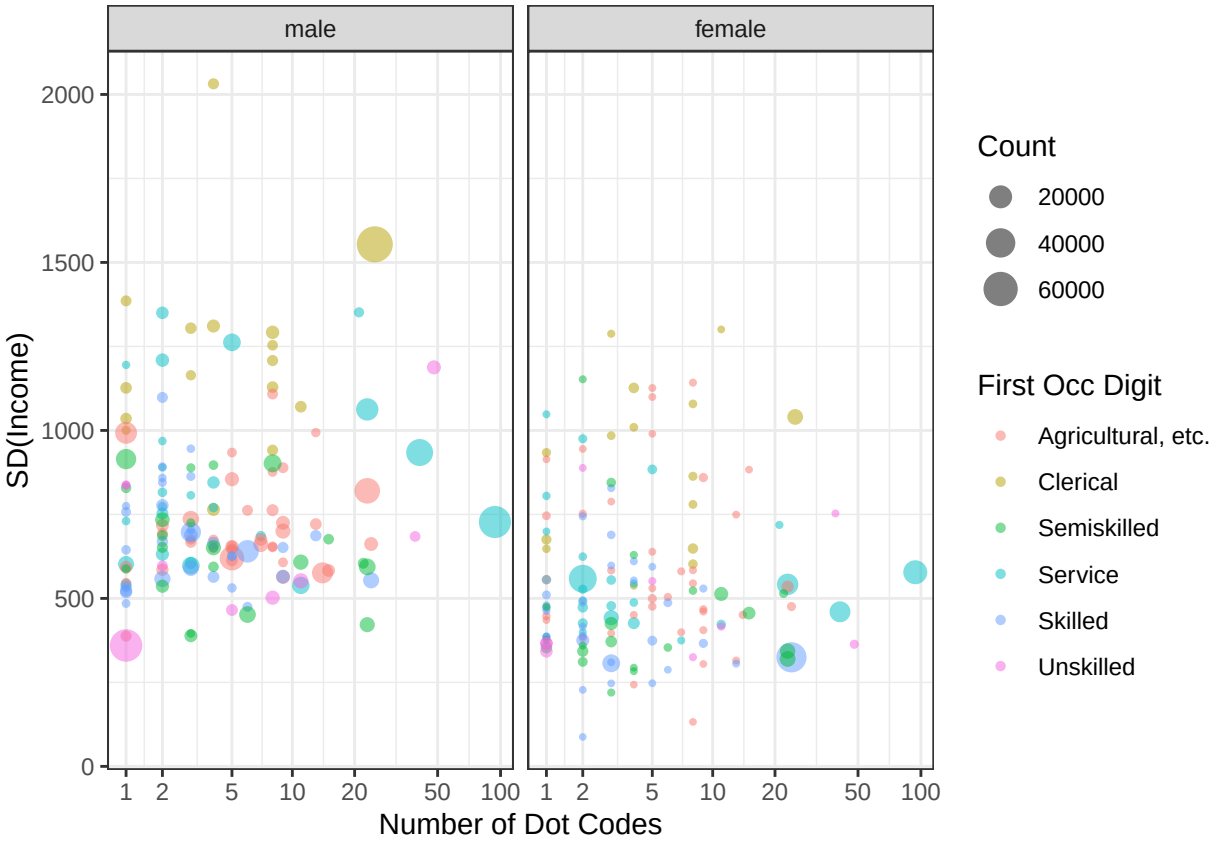


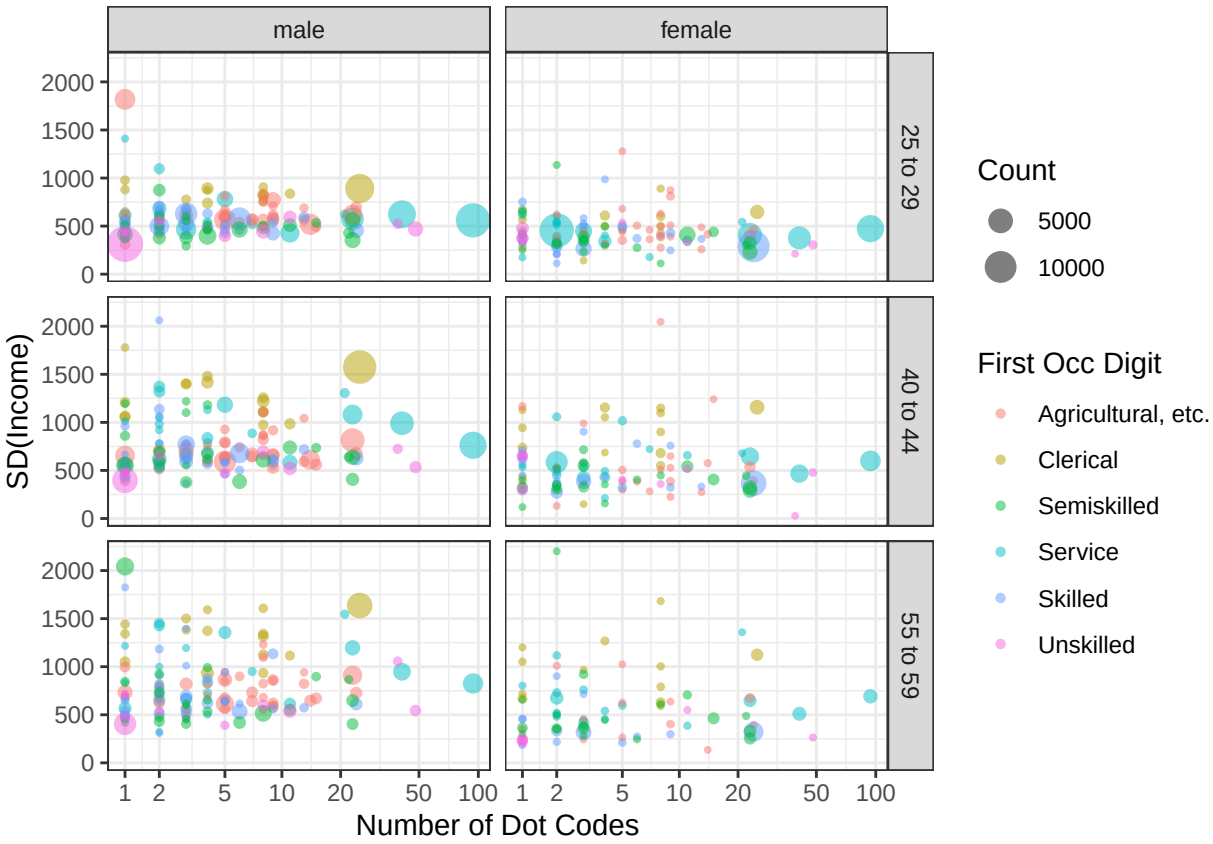



male
female
25 to 29
40 to 44
55 to 59
Gini(Income)
0.00
0.25
0.50
0.75
1
2
5
10
20
50
100
200
605
Number of Dot Codes
Count
5000
10000
First Occ Digit
Agricultural, etc.
Clerical
Semiskilled
Service
Skilled
Unskilled

Graph Standard deviation against number of DOT/Census Occ Category

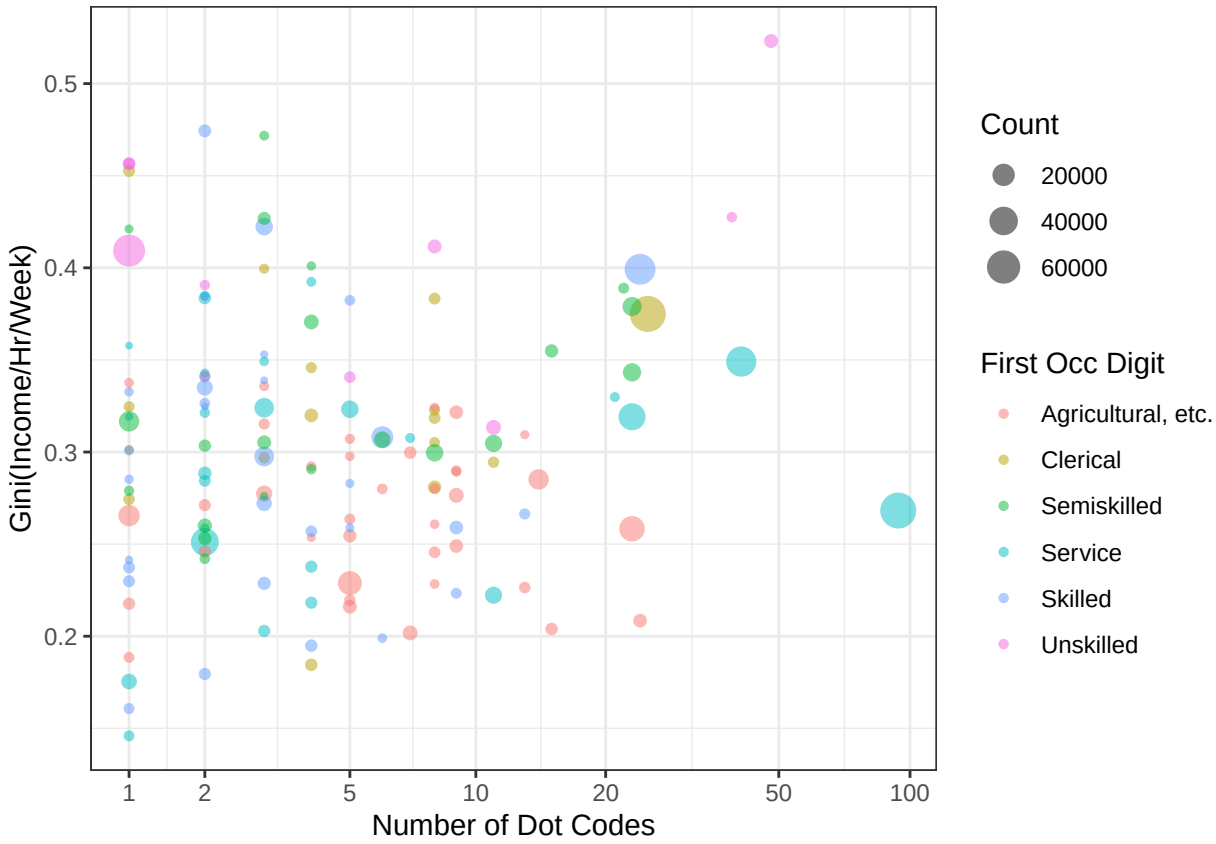




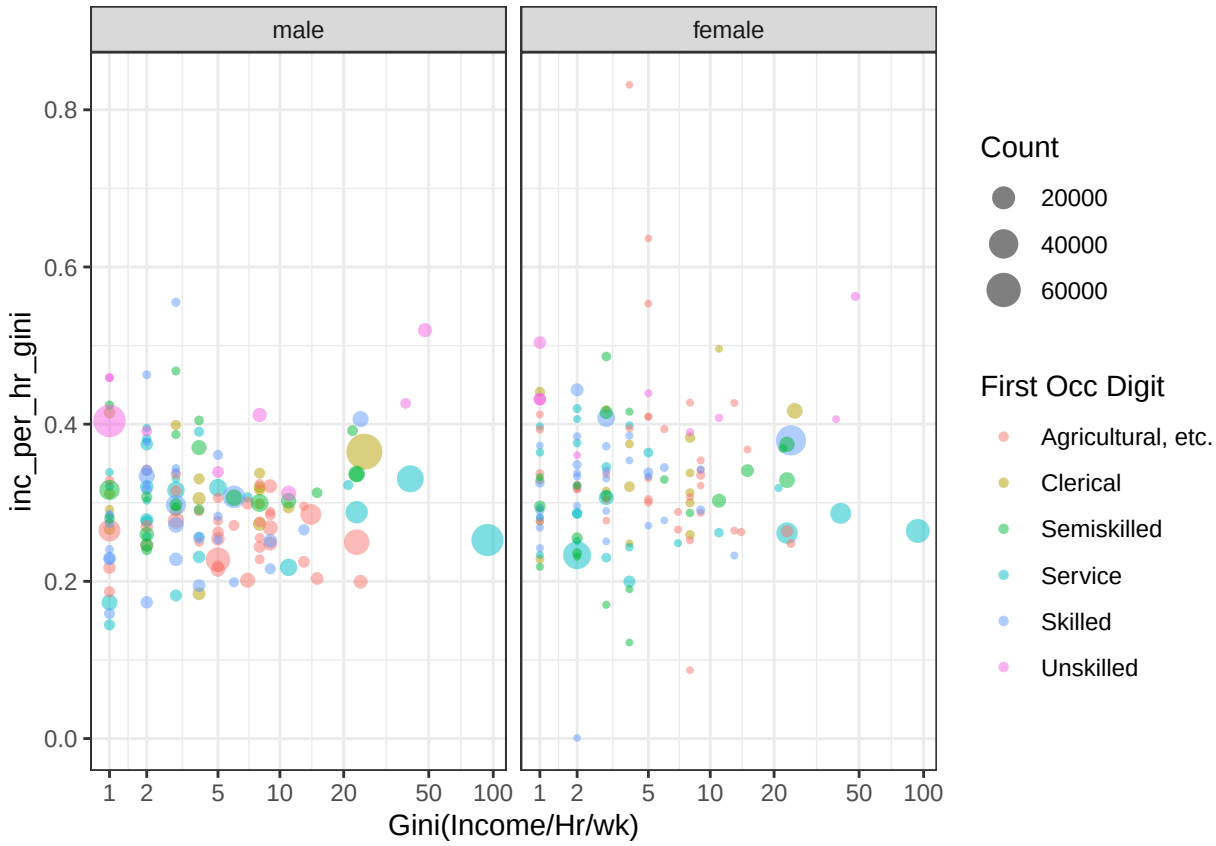


Graph Income/hr Inequality against number of DOT/Census Occ Category

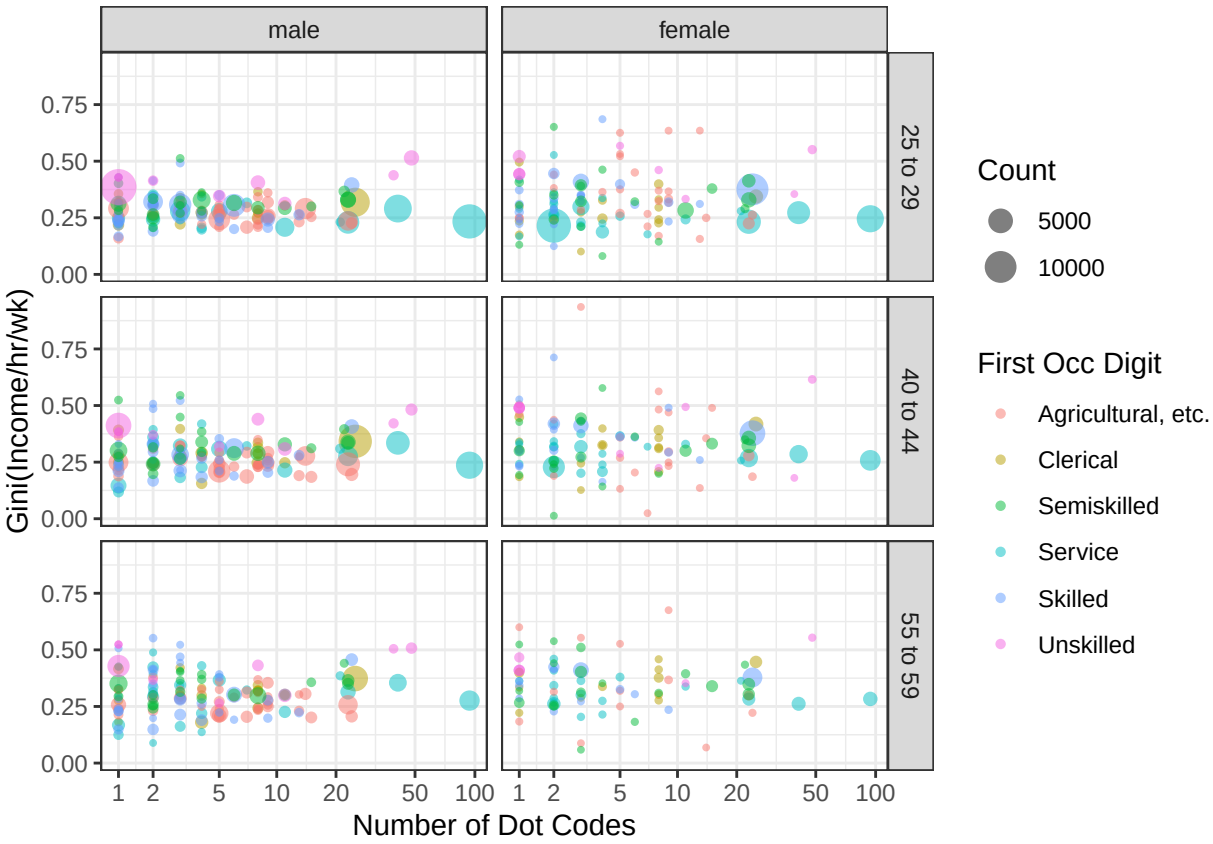
```
ggplot(gini_occ) +
  geom_point(aes(x = num_dot_codes,
                 y = inc_per_hr_gini,
                 size = number,
              color = occ_categ),
             alpha = .5,
             shape = 16) +
  scale_x_continuous(trans = "pseudo_log",
                     breaks = c(0,1,2,5,10,20,50,100,200))+
  xlab("Number of Dot Codes") +
  ylab("Gini(Income/Hr/Week)") +
  labs(size = "Count", color = "First Occ Digit")
```




```
ggplot(gini_occ_sex) +
  geom_point(aes(x = num_dot_codes,
                 y = inc_per_hr_gini,
                 size = number,
              color = occ_categ),
             alpha = .5,
             shape = 16) +
    scale_x_continuous(trans = "pseudo_log",
                       breaks = c(0,1,2,5,10,20,50,100,200))+
  xlab("Number of Dot Codes") +
  facet_wrap(~sex)+
  xlab("Gini(Income/Hr/wk)") +
  labs(size = "Count", color = "First Occ Digit")
```




```
ggplot(gini_occ_sex_age[age_cat %like% "25|40|55"]) +
  geom_point(aes(x = num_dot_codes,
                 y = inc_per_hr_gini,
                 size = number,
              color = occ_categ),
             alpha = .5,
             shape = 16) +
    scale_x_continuous(trans = "pseudo_log",
                       breaks = c(0,1,2,5,10,20,50,100,200))+
  xlab("Number of Dot Codes") +
  facet_grid(age_cat~sex)+
  ylab("Gini(Income/hr/wk)") +
  labs(size = "Count", color = "First Occ Digit")
```



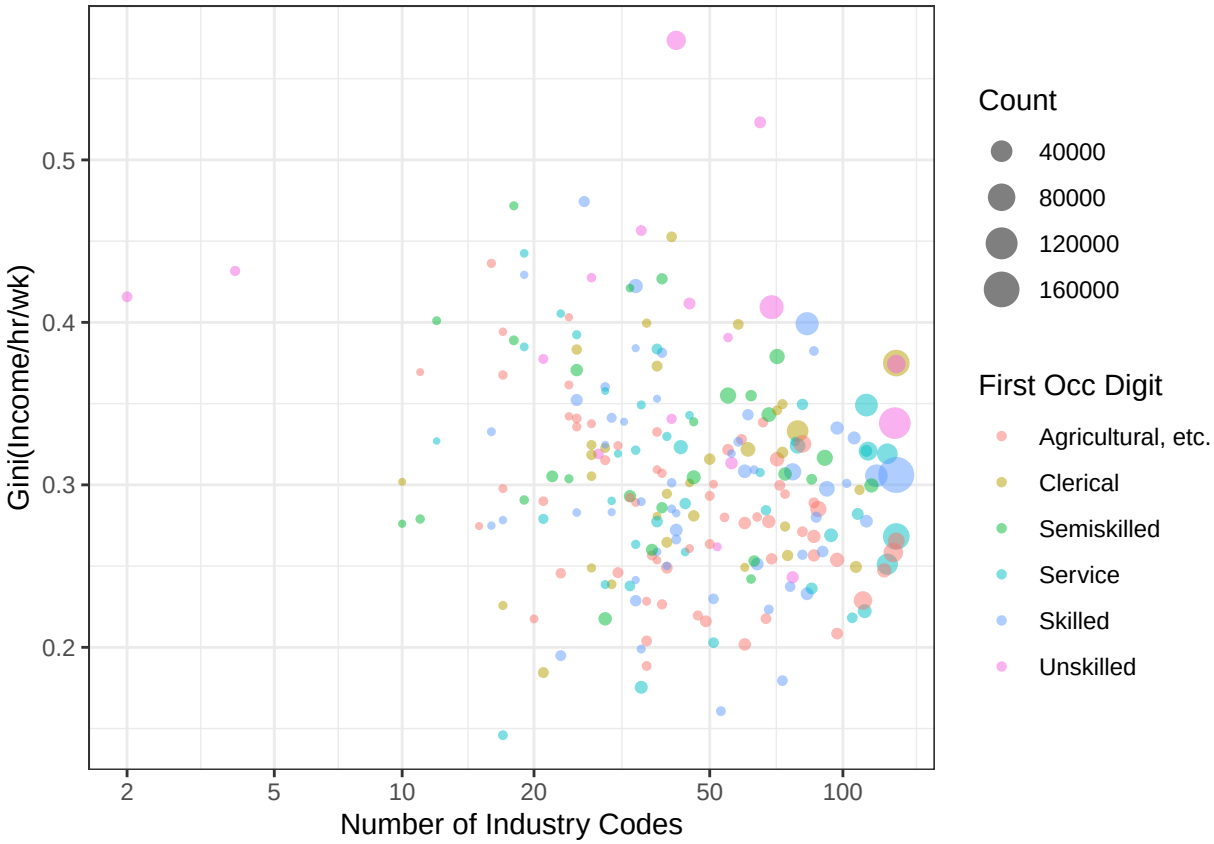
Task III - Does Census-Coded Industry give us any additional insight?

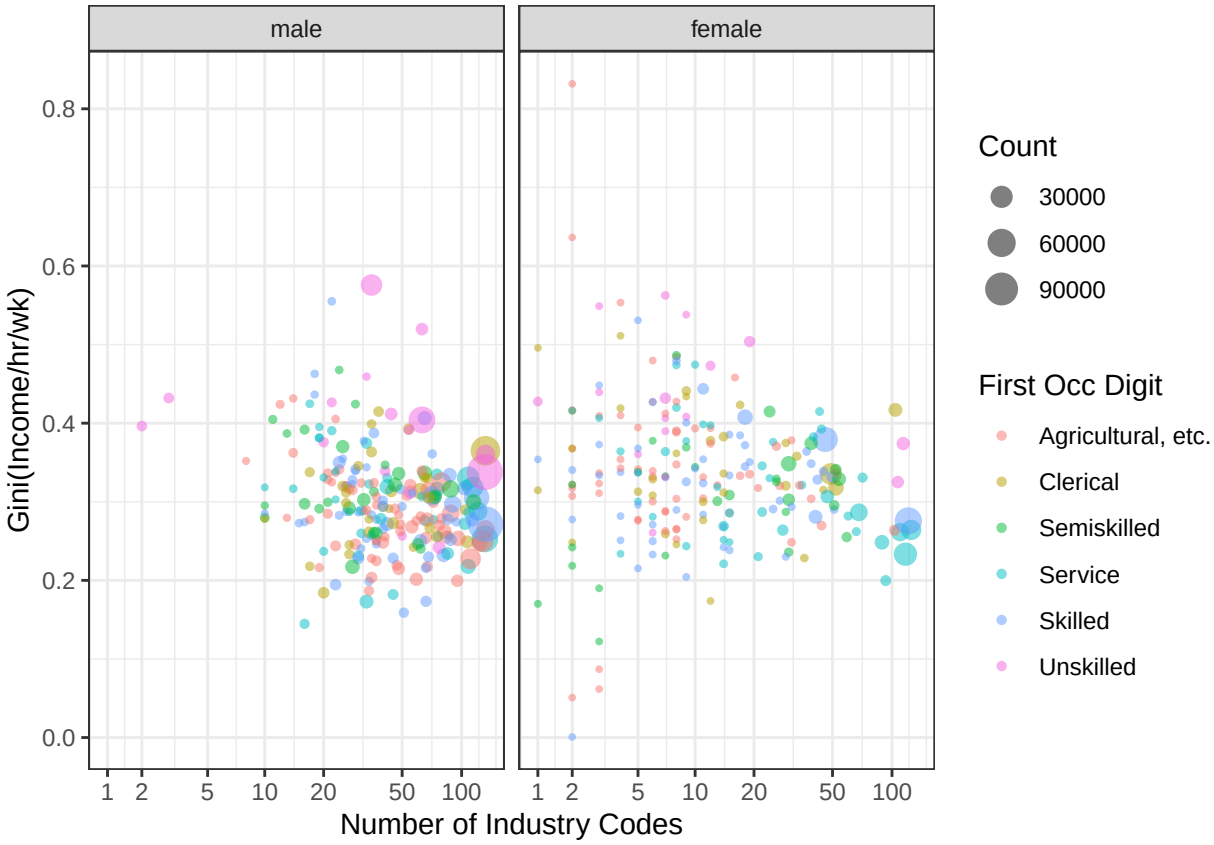
In this section, I repeat the above logic, but instead of the number of dot codes nested in each census category, I'm using the number of census industry categories nested into each census occupation category. Ultimately, I think a second crosswalk could be established (or maybe a natural language processing thing) to more accurately nest dot codes within each census occupation and industry combination.

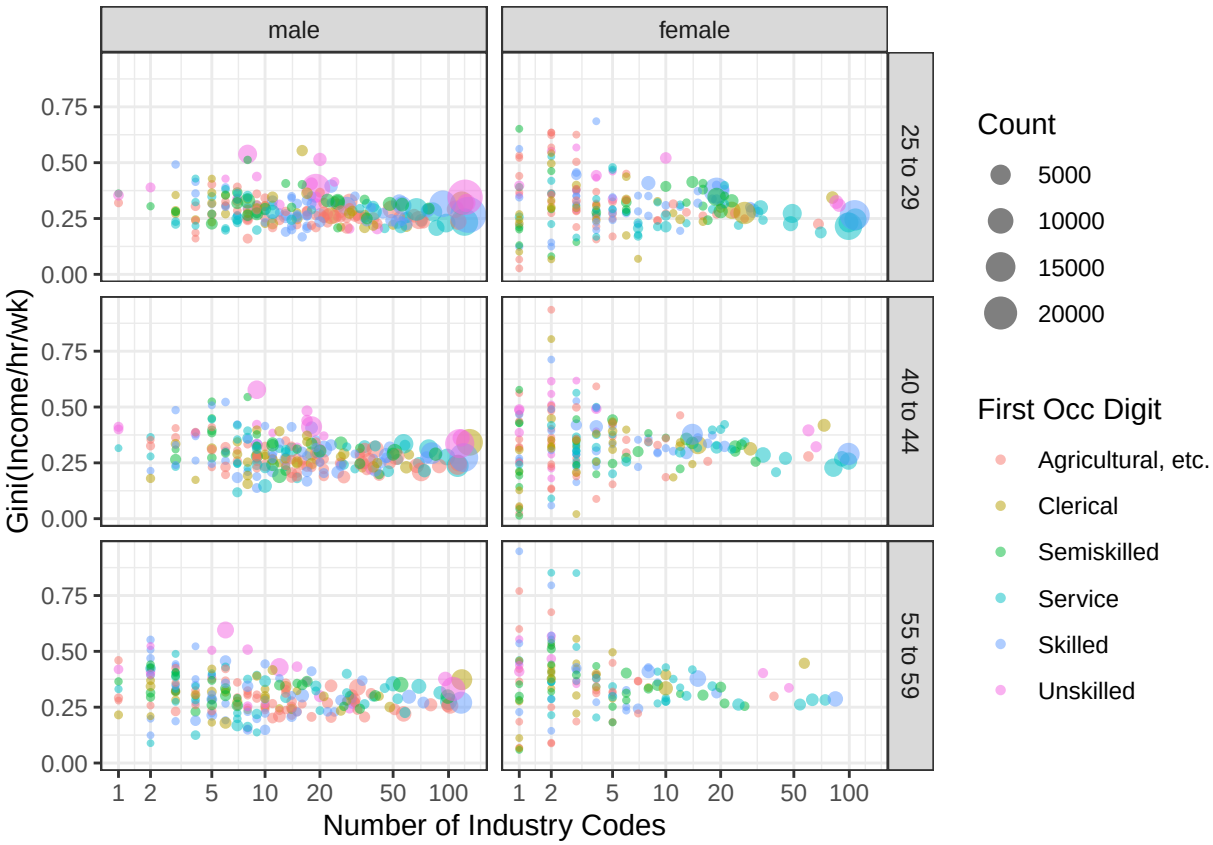
```
# merge census 1940 file on ind codes
ind_codes <- read_xlsx("../ref/IND1940.xlsx") %>% data.table()
ind_codes[, IND := as.numeric(IND)]
census_1940 <- merge(census_1940,
                     ind_codes[!is.na(IND) & IND != 996],
                     by.x = "ind", by.y = "IND", all.x = T)

gini_ind_occ <- census_1940[,.(inc_gini = Gini(incwage),
                               sd_inc = sd(incwage),
                               inc_per_hr_gini = Gini(income_per_hr),
                               avg_inc_per_hr = mean(income_per_hr/52),
                               num_ind_codes = length(unique(Industry)),
                               number = .N),
                            by = .(occ,occ_categ)]
gini_ind_occ_sex <- census_1940[,.(inc_gini = Gini(incwage),
                               inc_per_hr_gini = Gini(income_per_hr),
                                 sd_inc = sd(incwage),
                               avg_inc_per_hr = mean(income_per_hr/52),
```

```
                            num_ind_codes = length(unique(Industry)),
                            number = .N),
                            by = .(occ, occ_categ,sex)]
gini_ind_occ_sex_age <- census_1940[,.(inc_gini = Gini(incwage),
                            inc_per_hr_gini = Gini(income_per_hr),
                              sd_inc = sd(incwage),
                            avg_inc_per_hr = mean(income_per_hr/52),
                            num_ind_codes = length(unique(Industry)),
                            number = .N),
                            by = .(occ,occ_categ, sex,age_cat)]

# # calculate wage income gini coeff by occ (and other)
# gini_ind_occ <- census_1940[,.(inc_gini = Gini(incwage),
#                              sd_inc = sd(incwage),
#                              inc_per_hr_gini = Gini(income_per_hr),
#                              number = .N),
#                            by = .(occ,ind,Industry,occ_categ)]
# gini_ind_occ_sex <- census_1940[,.(inc_gini = Gini(incwage),
#                              inc_per_hr_gini = Gini(income_per_hr),
#                                sd_inc = sd(incwage),
#                              number = .N),
#                              by = .(occ,ind,Industry, occ_categ,sex)]
# gini_ind_occ_sex_age <- census_1940[,.(inc_gini = Gini(incwage),
#                              inc_per_hr_gini = Gini(income_per_hr),
#                                sd_inc = sd(incwage),
#                              number = .N),
#                              by = .(occ,ind,Industry,occ_categ, sex,age_cat)]
#
# gini_ind_occ <- merge(gini_ind_occ, cen_dot_code_counts, by.x = "occ", by.y = "Census1940")
# gini_ind_occ_sex <- merge(gini_ind_occ_sex, cen_dot_code_counts, by.x = "occ", by.y = "Census1940")
# gini_ind_occ_sex_age <- merge(gini_ind_occ_sex_age, cen_dot_code_counts, by.x = "occ", by.y = "Census
```

Graph Income/hr Inequality against number of DOT/Census Occ Category

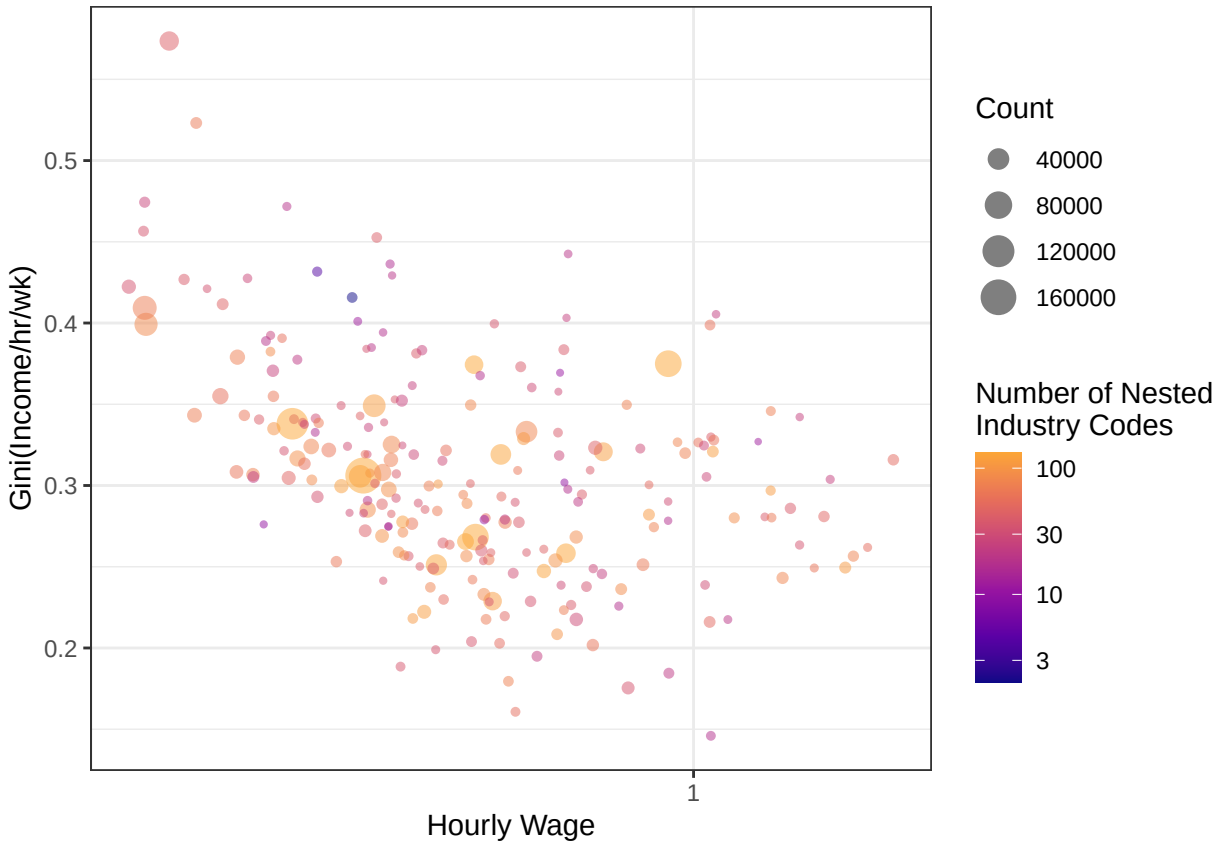




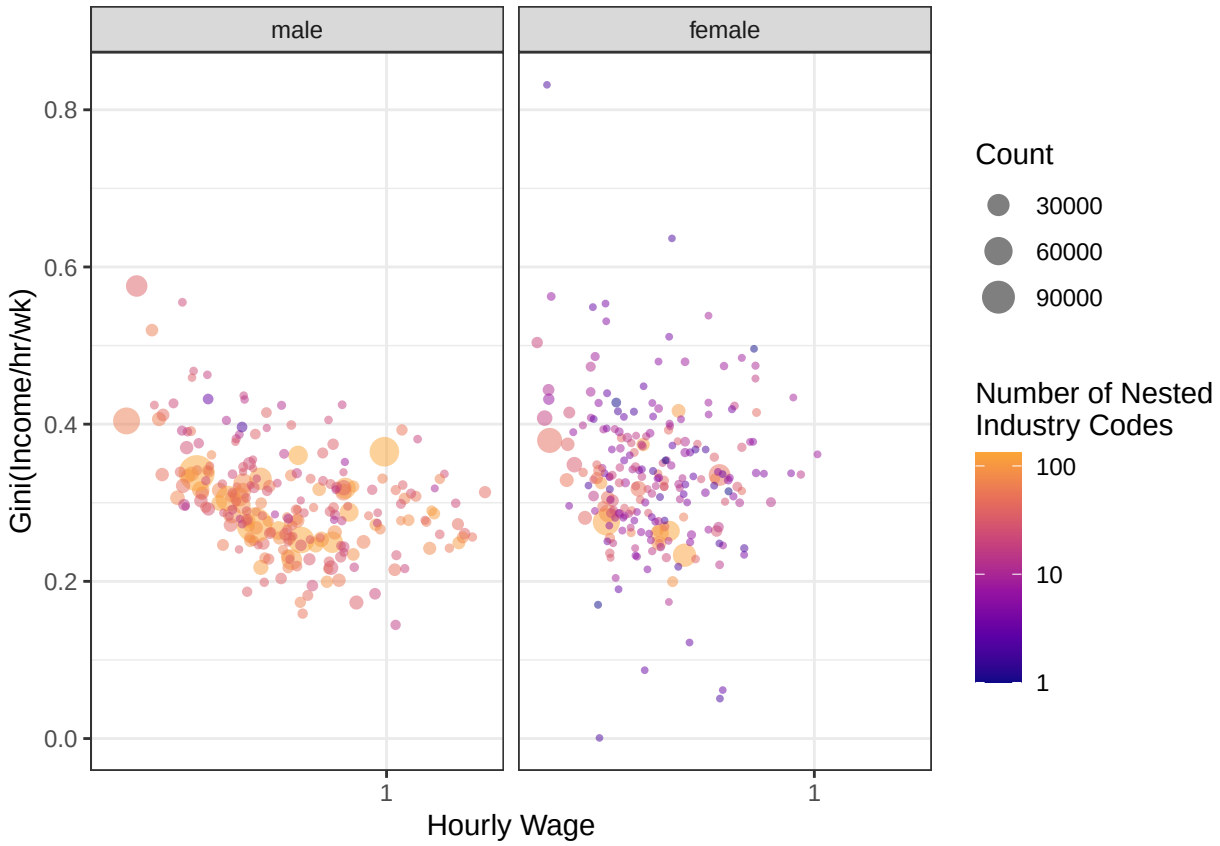


Graph Income/hr Inequality against Average earnings, with number of industry codes controlling size of points

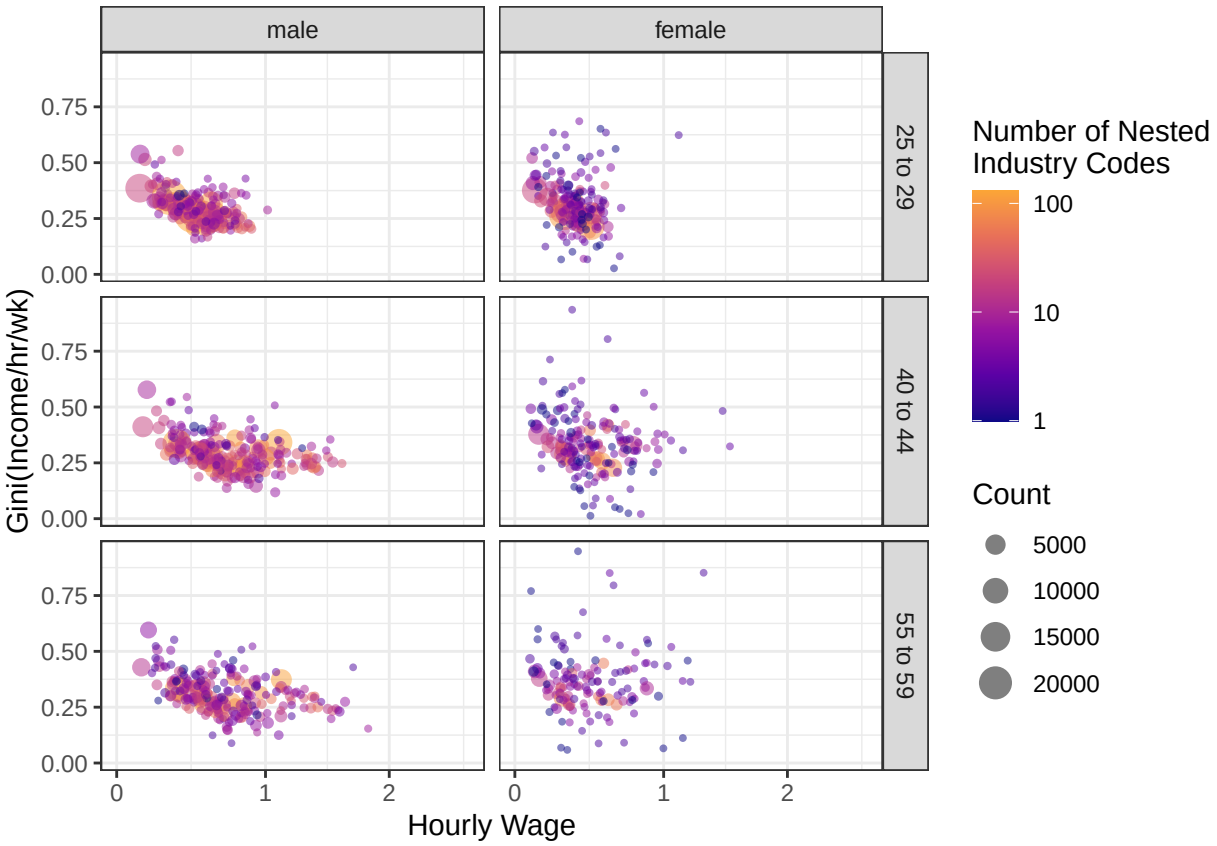
```
ggplot(gini_ind_occ) +
  geom_point(aes(x = avg_inc_per_hr,
                 y = inc_per_hr_gini,
                 size = number,
              color = num_ind_codes),
             alpha = .5,
             shape = 16) +
  scale_x_continuous(trans = "pseudo_log",
                     breaks = c(0,1,2,5,10,20,50,100,200))+
  scale_color_viridis_c(option = "C",begin = 0, end = .8, trans = "log10")+
  xlab("Hourly Wage") +
  ylab("Gini(Income/hr/wk)") +
  labs(size = "Count", color = "Number of Nested\nIndustry Codes")
```




```
ggplot(gini_ind_occ_sex) +
  geom_point(aes(x = avg_inc_per_hr,
                 y = inc_per_hr_gini,
                 size = number,
              color = num_ind_codes),
             alpha = .5,
             shape = 16) +
  scale_x_continuous(trans = "pseudo_log",
                     breaks = c(0,1,2,5,10,20,50,100,200))+
  facet_wrap(~sex)+
  scale_color_viridis_c(option = "C",begin = 0, end = .8, trans = "log10")+
  xlab("Hourly Wage") +
  ylab("Gini(Income/hr/wk)") +
  labs(size = "Count", color = "Number of Nested\nIndustry Codes")
```




```
ggplot(gini_ind_occ_sex_age[age_cat %like% c("25|40|55")]) +
  geom_point(aes(x = avg_inc_per_hr,
                 y = inc_per_hr_gini,
                 size = number,
              color = num_ind_codes),
             alpha = .5,
             shape = 16) +
  scale_x_continuous(trans = "pseudo_log",
                     breaks = c(0,1,2,5,10,20,50,100,200))+
  facet_grid(age_cat~sex)+
  scale_color_viridis_c(option = "C",begin = 0, end = .8, trans = "log10")+
  xlab("Hourly Wage") +
  ylab("Gini(Income/hr/wk)") +
  labs(size = "Count", color = "Number of Nested\nIndustry Codes")
```



Do we think we can do a better crosswalk using industry AND DOT?

Let's take for example the census occupation category with the greatest number of industries (Census_1940 == 156) which maps to the DOT parent occupation categories of HOTEL AND RESTAURANT MANAGERS, RETAIL MANAGERS, WHOLESALE MANAGERS, HATCHERY MEN which in turn contains 26 unique dot codes and 484 unique job titles. This census occupation coding also corresponds to 132 unique industries, amongst which are "Hotels and lodging places", "Federal government n.e.c.", etc. so a crosswalk is indeed possible.