analysis_vi

Hunter York

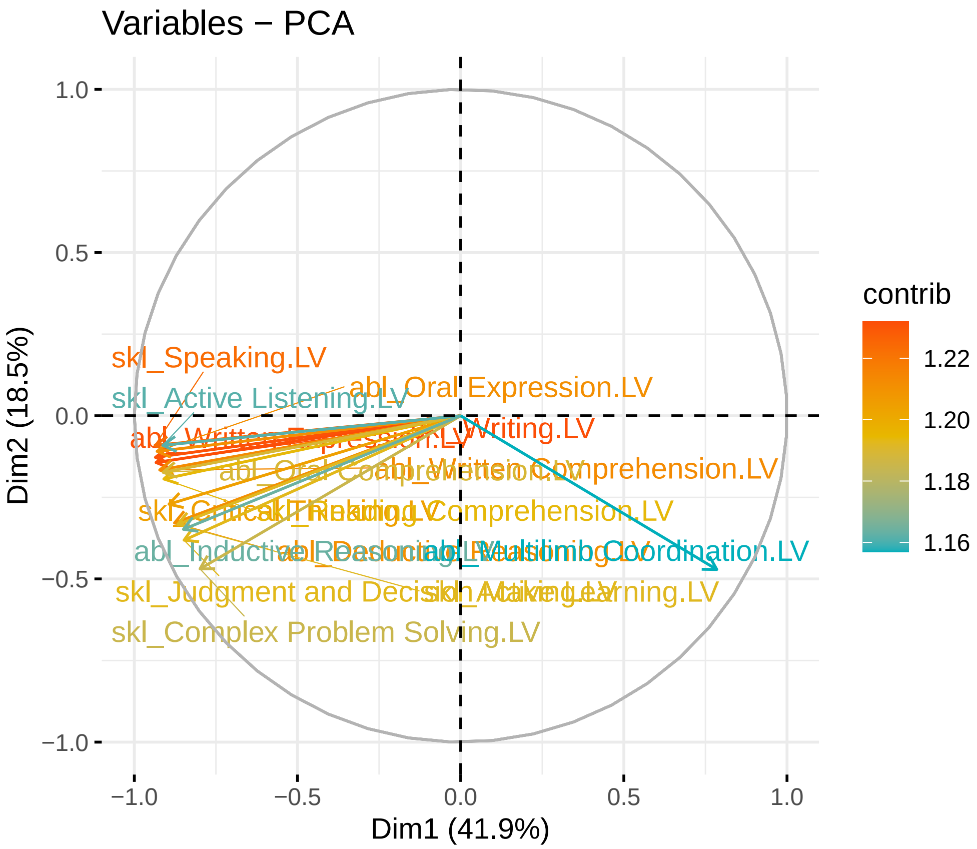
11/22/2020

Intro - Skills/Abilities/Knowledge

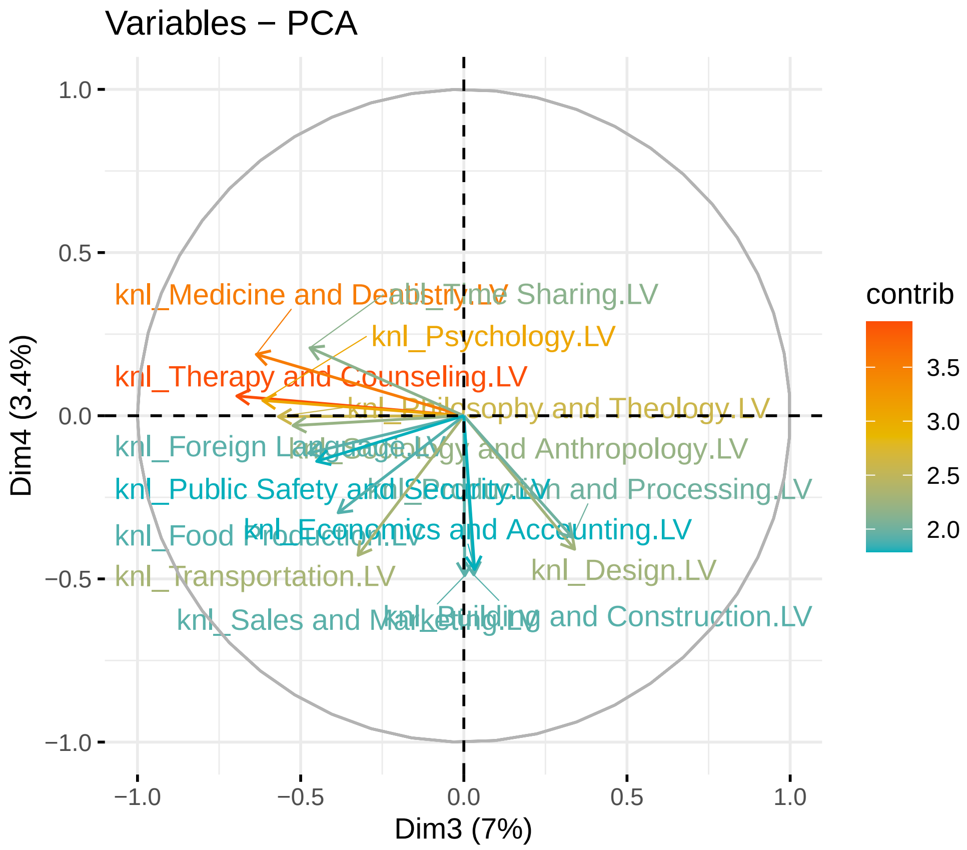
Similar to a few weeks ago, I've added data from ONET on abilities and knowledge, which are enumerated for each occupation on a similar scale to skills (0-1). Abilities in particular capture some physical aspects of occupations that are specific to less elite occupations, which will hopefully boost the predictive power of the following analyses in beingable to model the labor system.

PCA analysis to visualize how the new variables are clustered

```r
fviz_pca_var(res.pca,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE ,
             select.var = list(contrib = 15)    # Avoid text overlapping
)
```



```
fviz_pca_var(res.pca,
             axes = c(3,4),
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE,# Avoid text overlapping,
             select.var = list(contrib = 15)
)
```

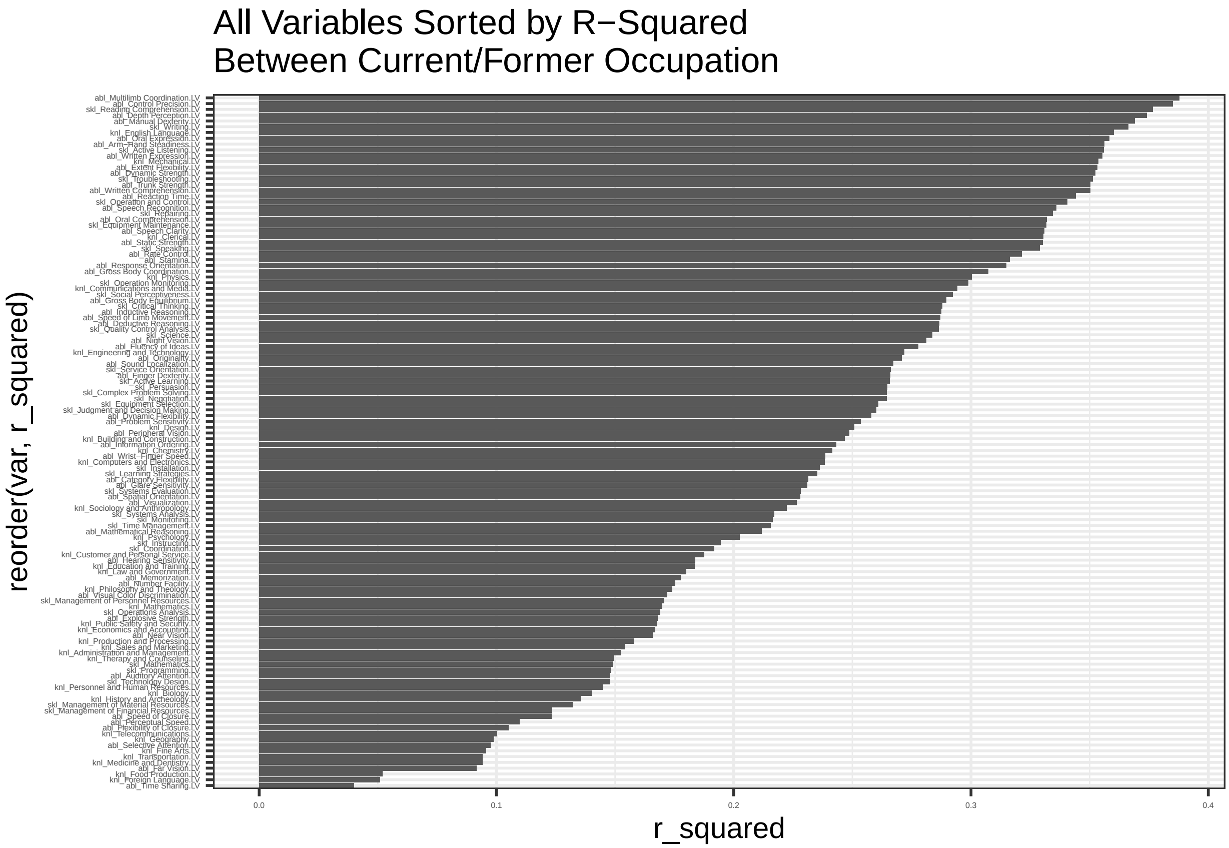


Amongst occupational movers, which skills/knowledges/abilities are most correlated pre-/post-move

This section regresses each skill/ability/knowledge from last year against the same skill/ability/knowledge from the current year amongst people who have changed occupations. Reported values are the r-squared from this regression.

Some are highly correlated, while others are less. These weights contain information that could be used later to construct occupational groupings based on the salience of job qualities in determining the regidity of job transitions, but for now all skills/abilities/knowledges are considered equally in the construction of job categories.

```
moved <- acs[OCC1990 != OCC90LY]

lmr <- function(data, var){
  lm_out <- lm(as.formula(paste0("`",var, "_current` ~ `", var, "_ly`")), data = data)
  return(data.table(slope = summary(lm_out)$coefficients[2, 1],
                    se = summary(lm_out)$coefficients[2, 2],
                    r_squared =summary(lm_out)$r.squared,
                    var = var))
}

lapply(vars[vars %like% "skl|abl|knl"], lmr, data = moved) %>% rbindlist() -> lmr_out_full

ggplot(lmr_out_full) +
  geom_bar(aes(x = r_squared, y = reorder(var, r_squared)), stat = "identity") +
  theme(axis.text = element_text(size = 3)) +
  ggtitle("All Variables Sorted by R-Squared\nBetween Current/Former Occupation")
```

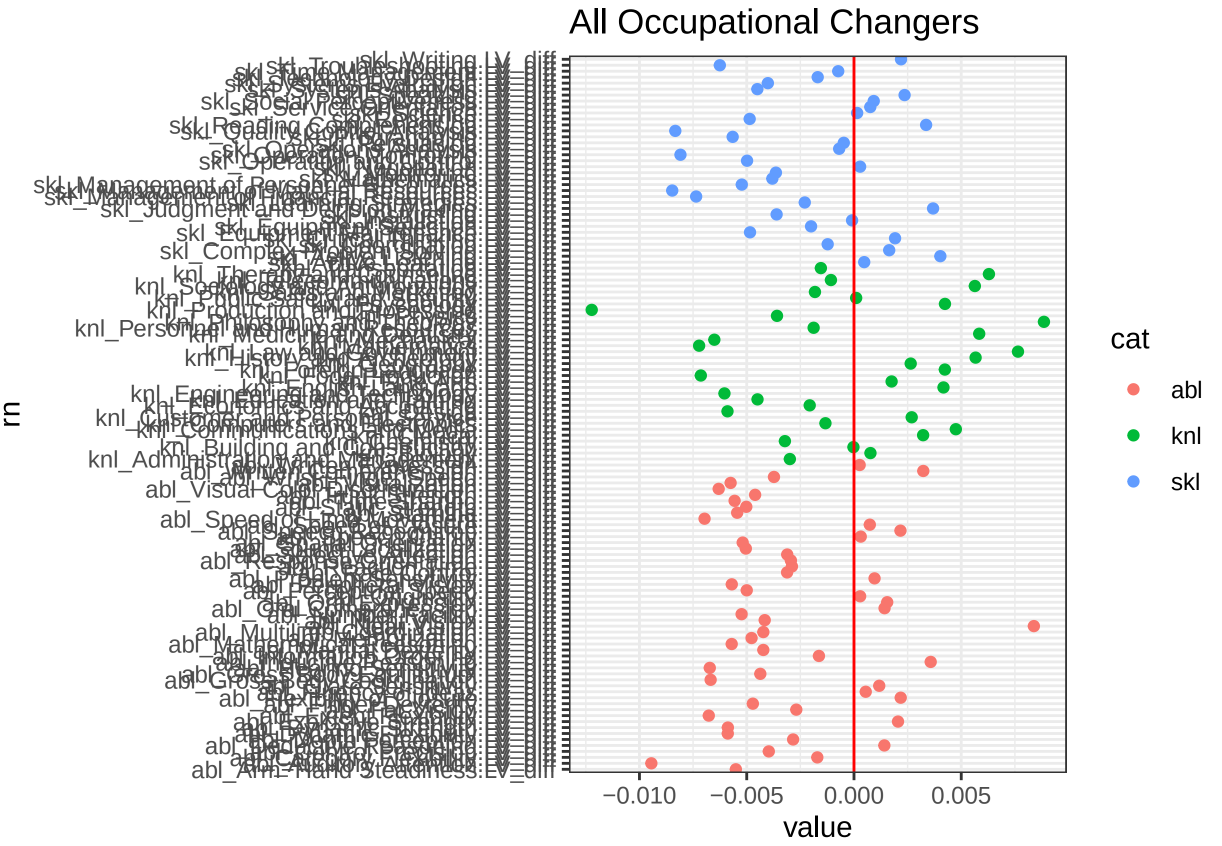


See how skills/knowledges/abilities change pre/post job transition based on the context of the transition

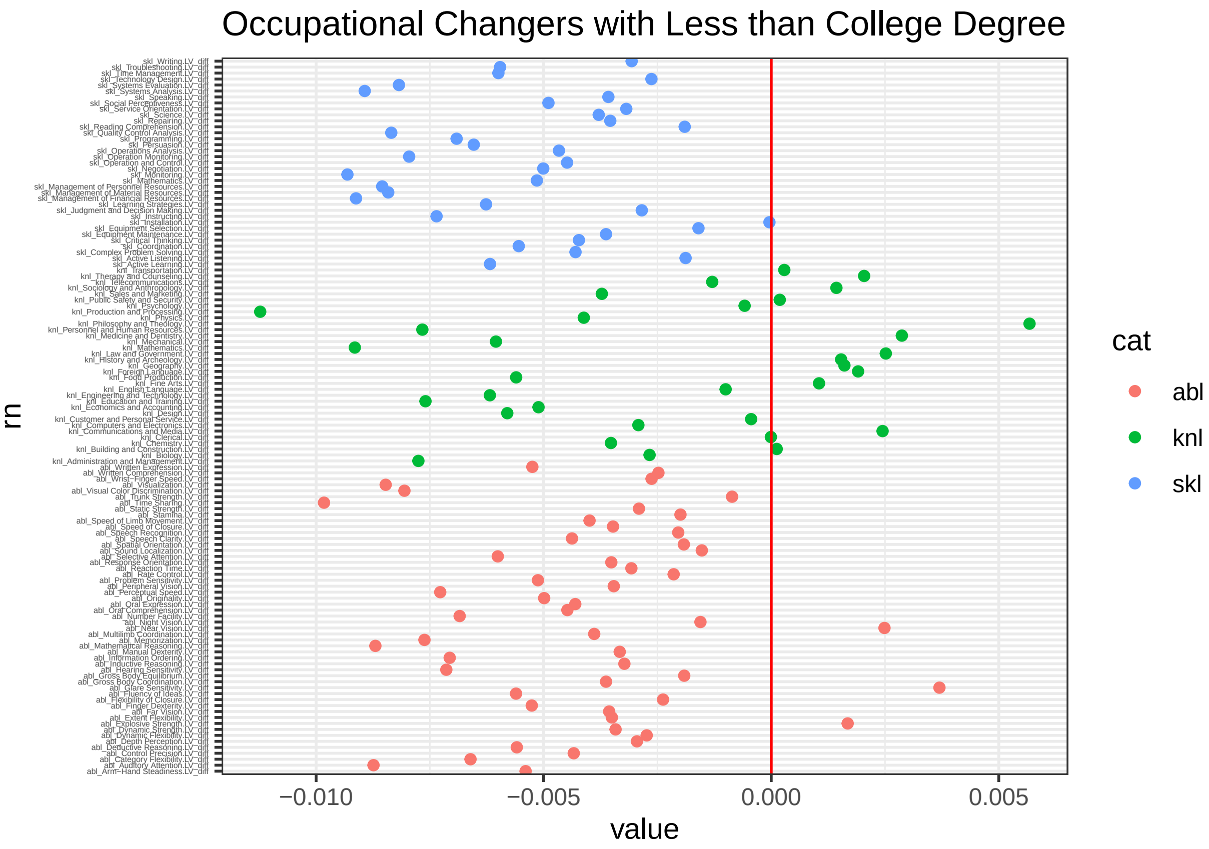
The following graphs show the net change in average skill amongst certain classes of job movers: based on educational attainment, whether or not the job transition is in the same industry or a different industry, if the job transition was coupled with a physical relocation and the context of that relocation (moved due to lost job/moved due to promotion for example), if the worker was part time last year due to having been out of work, etc.

```
for(c.skill in vars){
  #acs[, paste0(c.skill, "_distance") := abs(get(paste0(c.skill, '_current')) - get(paste0(c.skill, '_l
  acs[, paste0(c.skill, "_diff") := (get(paste0(c.skill, '_current')) - get(paste0(c.skill, '_ly')))]

}




acs[, ed_num := as.numeric(as.character(factor(educ,levels = levels(acs$educ), labels = c(0,0,0,2.5,1,2
                                                                                          5,6,7.5, 7,8,
                                                                                          13,14,14,15,1


acs[ed_num <= 14, ed_categ := "Less than HS"]
acs[ed_num > 14, ed_categ := "College Plus"]
```

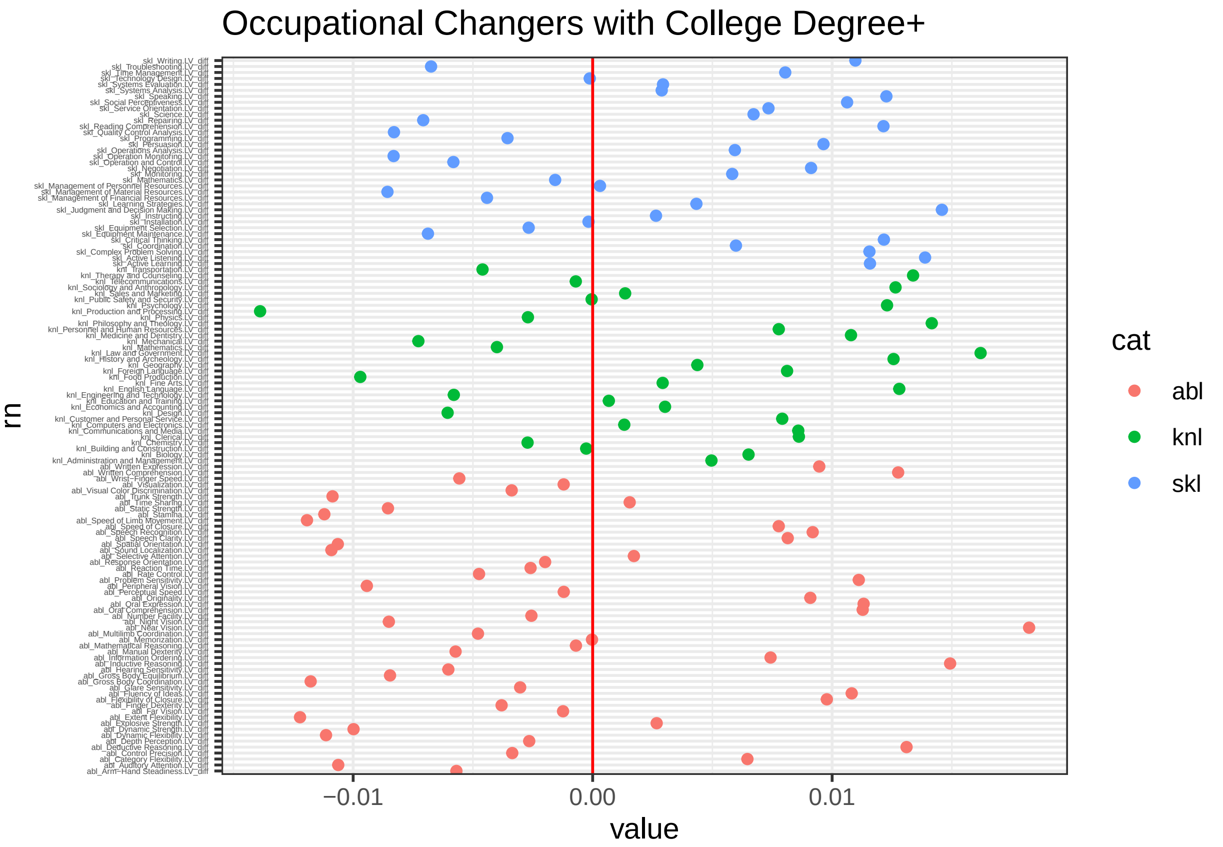
```
# subset to flows
skills_flows <- acs[OCC10LY != OCC2010]

# # graph distribution
# ggplot(skills_flows[!(OCC2010 == 4760 & OCC10LY == 4850),.SD, .SDcols = paste0(vars[!vars %like% "ski
#   geom_histogram(aes(x = value)) +
#   facet_wrap(~variable) +
#   xlim(0,1) +
#   geom_abline(intercept = 2831.867, slope = -2831.867)

skills_flows[,colMeans(.SD, na.rm = T), .SDcols = names(skills_flows)[names(skills_flows) %like% "skl|al
                                                                        names(skills_flows) %like% "dif
  melt(.) %>%
  data.table(., keep.rownames = T) %>%
  .[, cat := substr(rn, 1,3)] %>%
  ggplot(.) +
  geom_point(aes(y = rn, x = value, color = cat)) +
  geom_vline(xintercept = 0, color = "red") +
  ggtitle("All Occupational Changers")
```



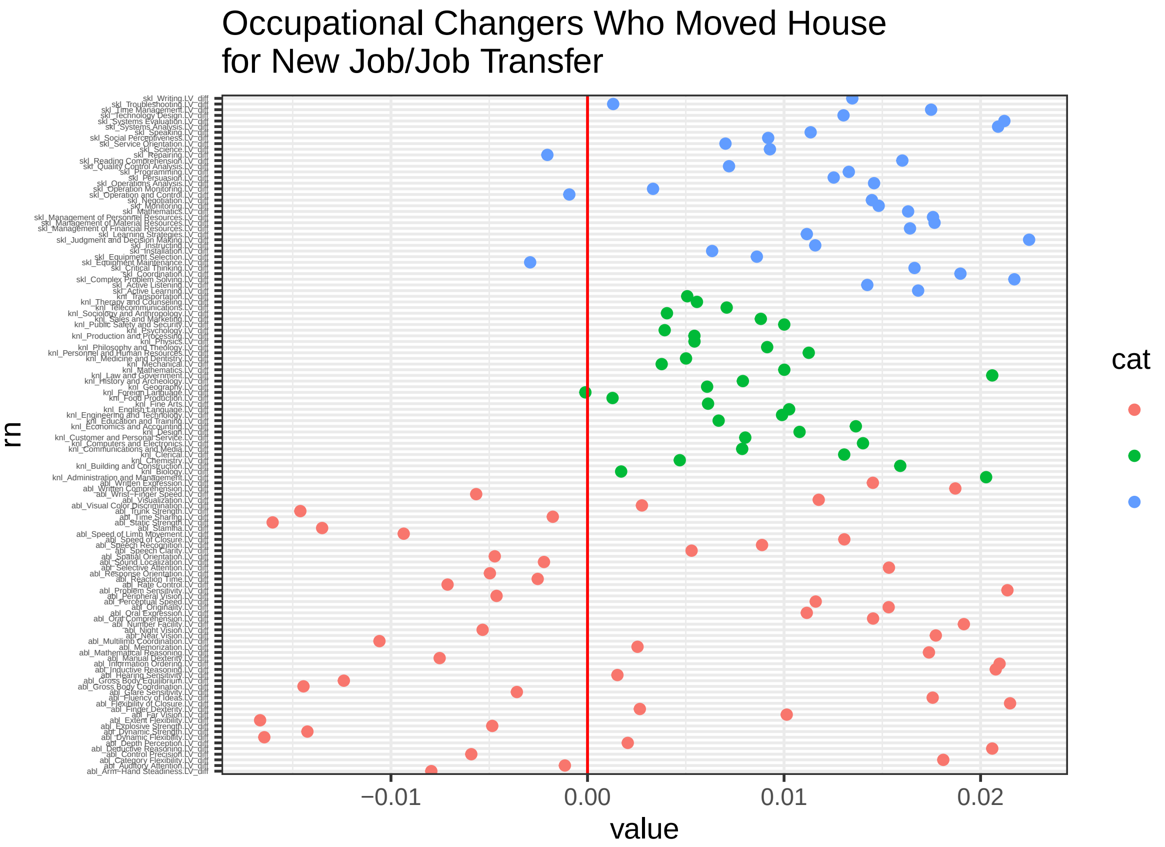
```
skills_flows[ed_categ %like% "Less", colMeans(.SD, na.rm = T), .SDcols = names(skills_flows)[names(skil
                                                                                               names(ski
  melt(.) %>%
  data.table(., keep.rownames = T) %>%
  .[, cat := substr(rn, 1,3)] %>%
  ggplot(.) +
  theme(axis.text.y = element_text(size = 3)) +
  geom_point(aes(y = rn, x = value, color = cat)) +
  geom_vline(xintercept = 0, color = "red")+
  ggtitle("Occupational Changers with Less than College Degree")
```



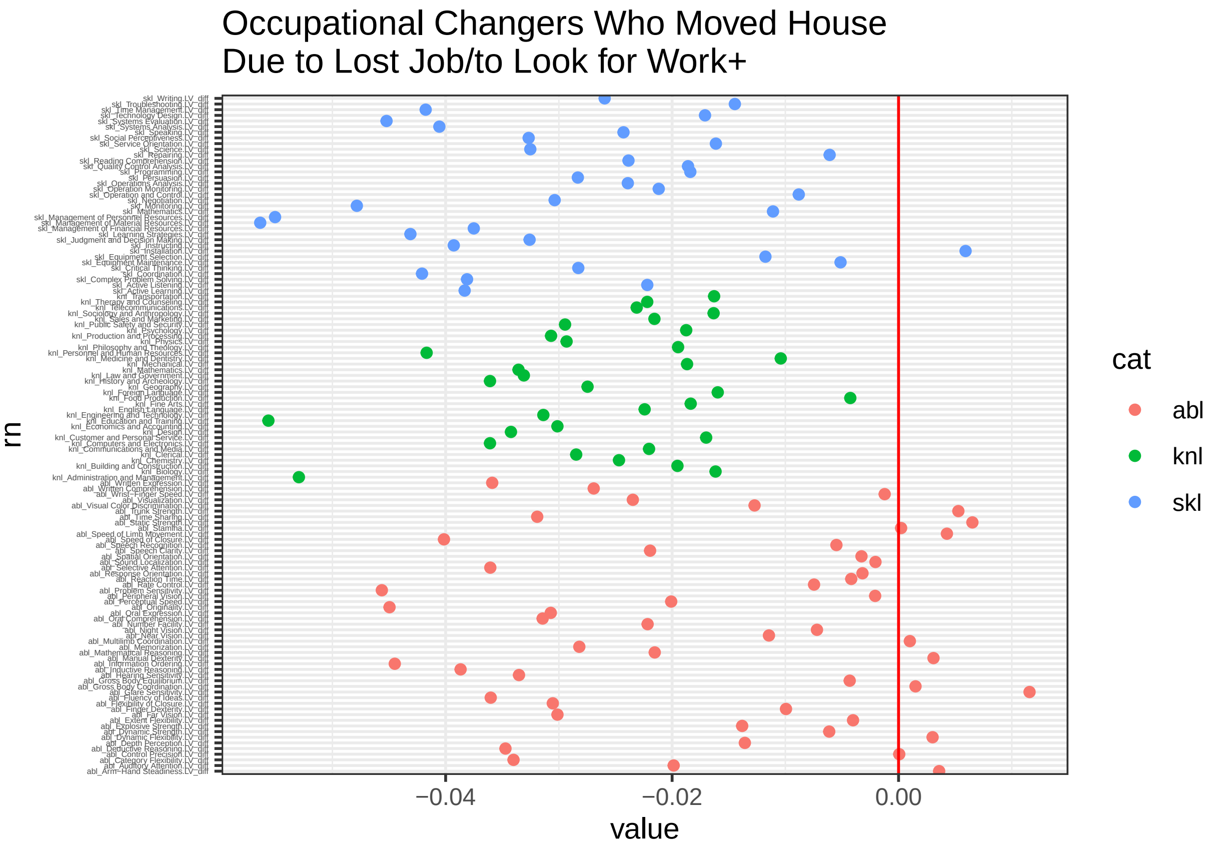
```
skills_flows[!ed_categ %like% "Less", colMeans(.SD, na.rm = T), .SDcols = names(skills_flows)[names(skil
                                                                                                 names(sl
  melt(.) %>%
  data.table(., keep.rownames = T) %>%
  .[, cat := substr(rn, 1,3)] %>%
  ggplot(.) +
  geom_point(aes(y = rn, x = value, color = cat)) +
  theme(axis.text.y = element_text(size = 3)) +
  geom_vline(xintercept = 0, color = "red")+
  ggtitle("Occupational Changers with College Degree+")
```



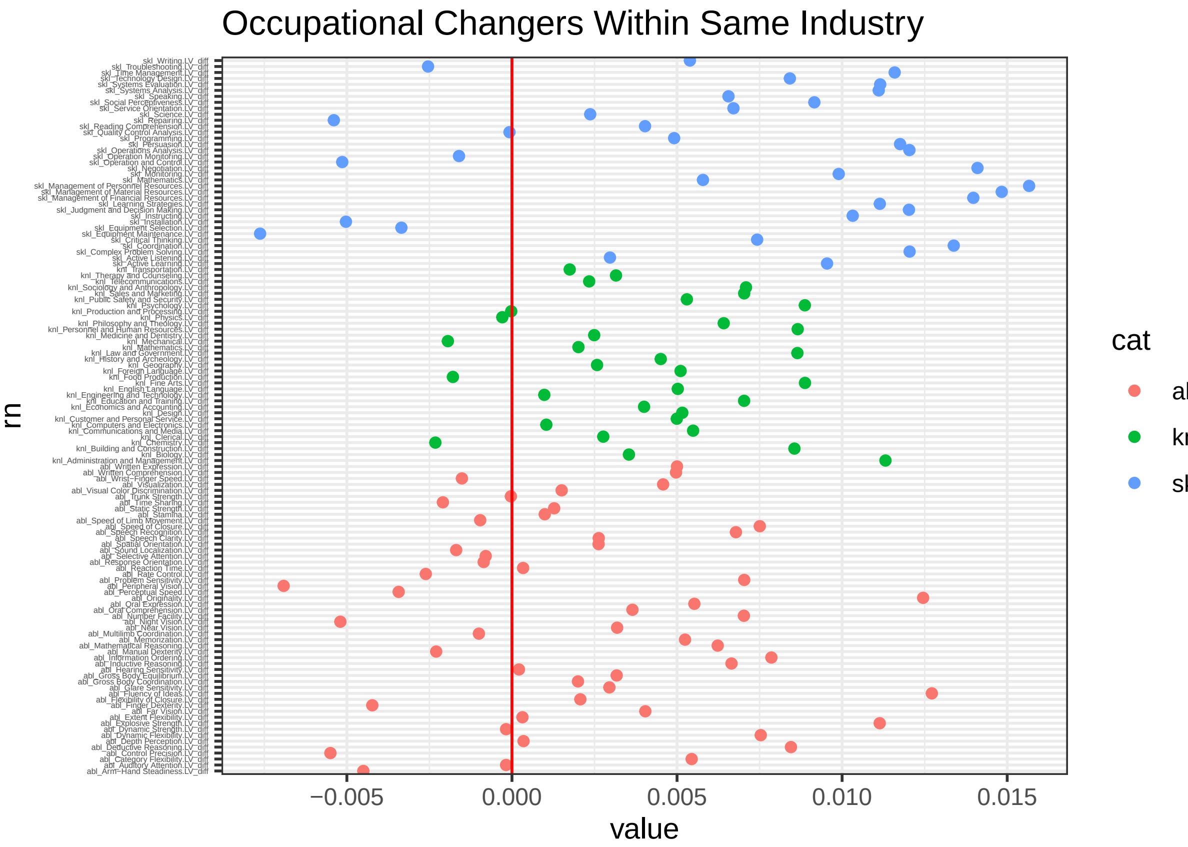
```
skills_flows[as.numeric(whymove) %in% c(5), colMeans(.SD, na.rm = T),
             .SDcols = names(skills_flows)[names(skills_flows) %like% "skl|abl|knl" &
                                           names(skills_flows) %like% "diff"]] %>%
  melt(.) %>%
  data.table(., keep.rownames = T) %>%
  .[, cat := substr(rn, 1,3)] %>%
  ggplot(.) +
  geom_point(aes(y = rn, x = value, color = cat)) +
  theme(axis.text.y = element_text(size = 3)) +
  geom_vline(xintercept = 0, color = "red")+
  ggtitle("Occupational Changers Who Moved House\nfor New Job/Job Transfer")
```



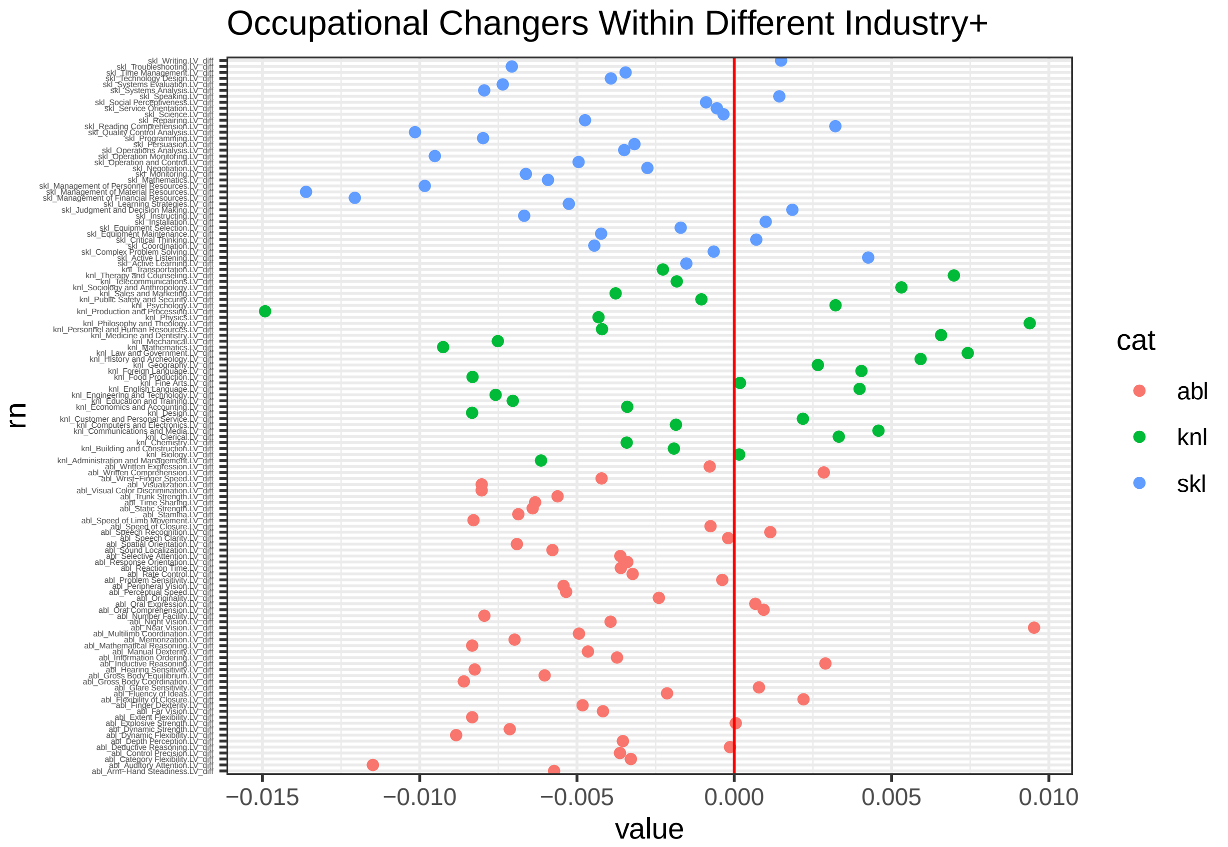
```
skills_flows[as.numeric(whymove) %in% c(6), colMeans(.SD, na.rm = T),
             .SDcols = names(skills_flows)[names(skills_flows) %like% "skl|abl|knl" &
                                            names(skills_flows) %like% "diff"]] %>%
  melt(.) %>%
  data.table(., keep.rownames = T) %>%
  .[, cat := substr(rn, 1,3)] %>%
  ggplot(.) +
  geom_point(aes(y = rn, x = value, color = cat)) +
  theme(axis.text.y = element_text(size = 3)) +
  geom_vline(xintercept = 0, color = "red")+
  ggtitle("Occupational Changers Who Moved House\nDue to Lost Job/to Look for Work+")
```



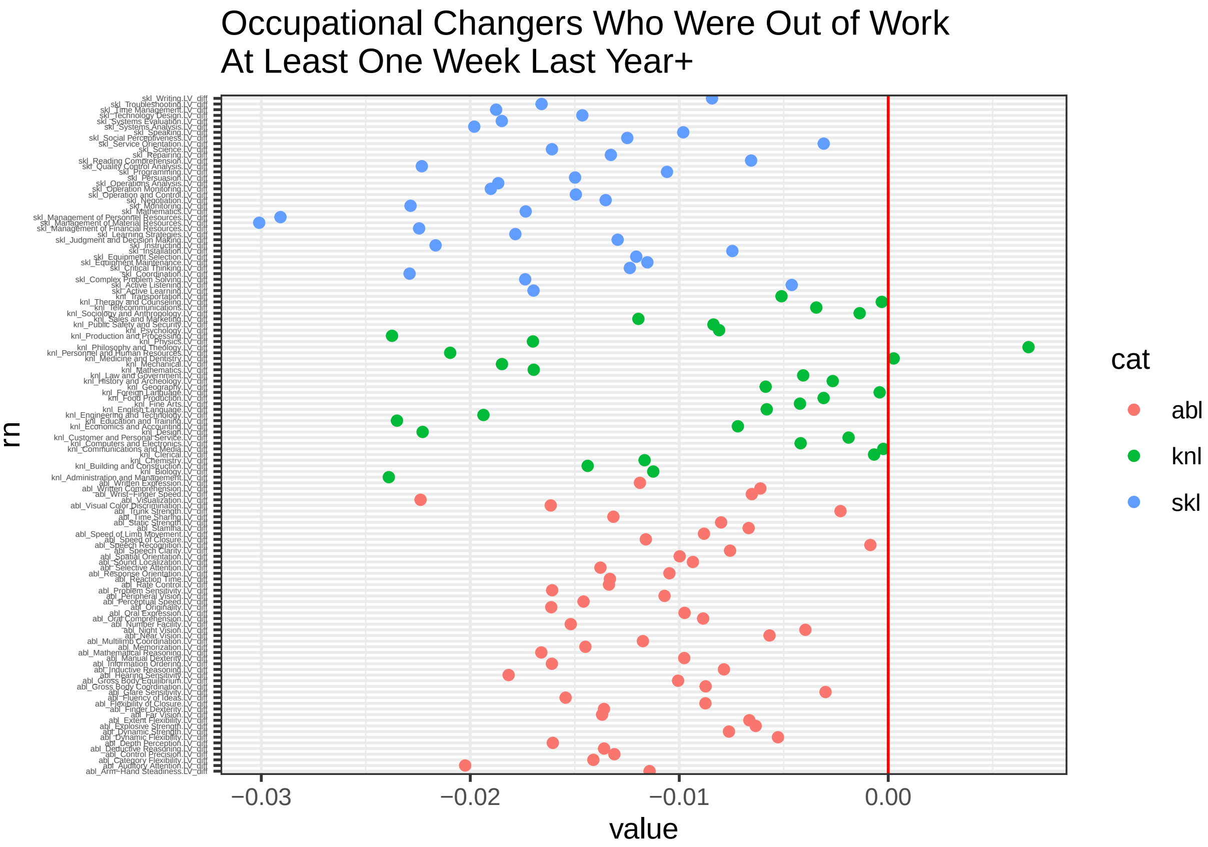
```
skills_flows[ind == indly, colMeans(.SD, na.rm = T),
             .SDcols = names(skills_flows)[names(skills_flows) %like% "skl|abl|knl" &
                                            names(skills_flows) %like% "diff"]] %>%
  melt(.) %>%
  data.table(., keep.rownames = T) %>%
  .[, cat := substr(rn, 1,3)] %>%
  ggplot(.) +
  geom_point(aes(y = rn, x = value, color = cat)) +
  geom_vline(xintercept = 0, color = "red")+
  theme(axis.text.y = element_text(size = 3)) +
  ggtitle("Occupational Changers Within Same Industry")
```



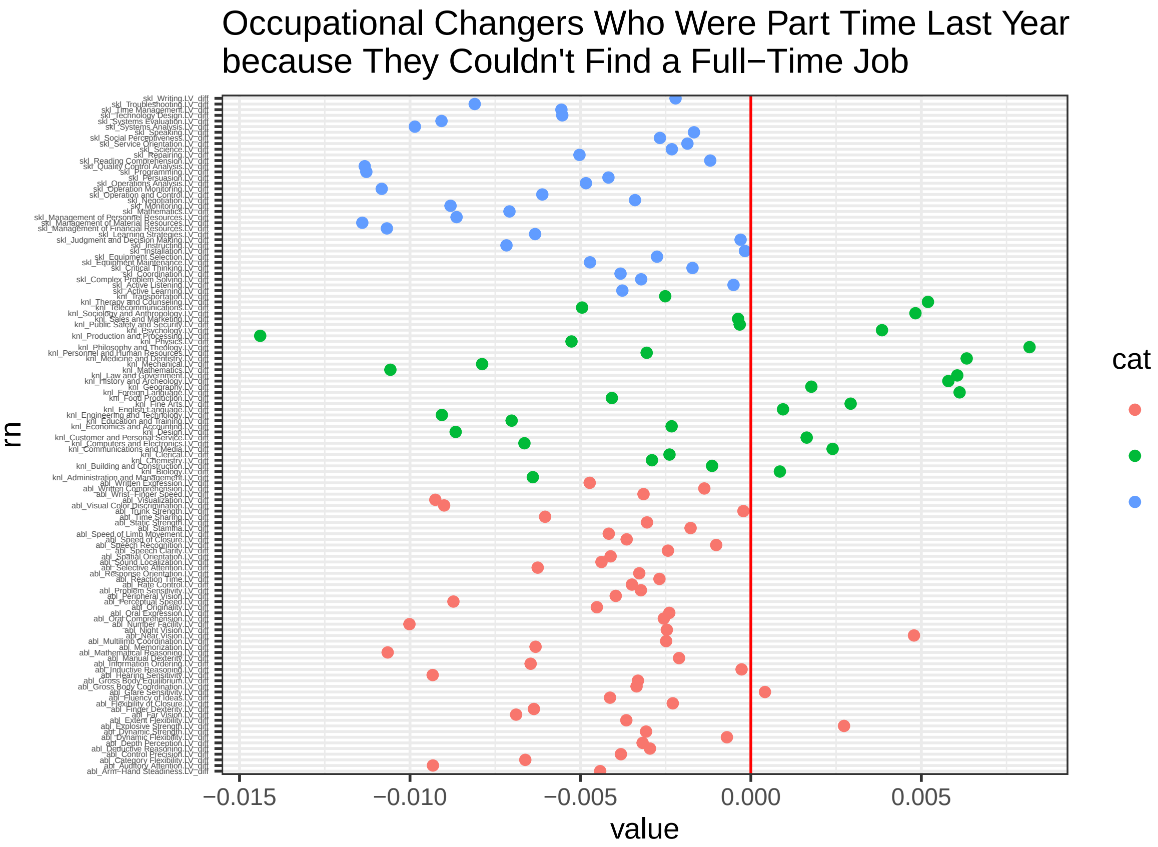
```
skills_flows[ind != indly, colMeans(.SD, na.rm = T),
             .SDcols = names(skills_flows)[names(skills_flows) %like% "skl|abl|knl" &
                                            names(skills_flows) %like% "diff"]] %>%
  melt(.) %>%
  data.table(., keep.rownames = T) %>%
  .[, cat := substr(rn, 1,3)] %>%
  ggplot(.) +
  geom_point(aes(y = rn, x = value, color = cat)) +
  geom_vline(xintercept = 0, color = "red")+
  theme(axis.text.y = element_text(size = 3)) +
  ggtitle("Occupational Changers Within Different Industry+")
```



```
skills_flows[as.numeric(strechlk) %in% 2:5, colMeans(.SD, na.rm = T),
             .SDcols = names(skills_flows)[names(skills_flows) %like% "skl|abl|knl" &
                                          names(skills_flows) %like% "diff"]] %>%
  melt(.) %>%
  data.table(., keep.rownames = T) %>%
  .[, cat := substr(rn, 1,3)] %>%
  ggplot(.) +
  geom_point(aes(y = rn, x = value, color = cat)) +
  geom_vline(xintercept = 0, color = "red")+
  theme(axis.text.y = element_text(size = 3)) +
  ggtitle("Occupational Changers Who Were Out of Work\nAt Least One Week Last Year+")
```



```
skills_flows[as.numeric(whyptly) %in% 1, colMeans(.SD, na.rm = T),
             .SDcols = names(skills_flows)[names(skills_flows) %like% "skl|abl|knl" &
                                            names(skills_flows) %like% "diff"]] %>%
  melt(.) %>%
  data.table(., keep.rownames = T) %>%
  .[, cat := substr(rn, 1,3)] %>%
  ggplot(.) +
  geom_point(aes(y = rn, x = value, color = cat)) +
  theme(axis.text.y = element_text(size = 3)) +
  geom_vline(xintercept = 0, color = "red") +
  ggtitle("Occupational Changers Who Were Part Time Last Year\nbecause They Couldn't Find a Full-Time Jo
```

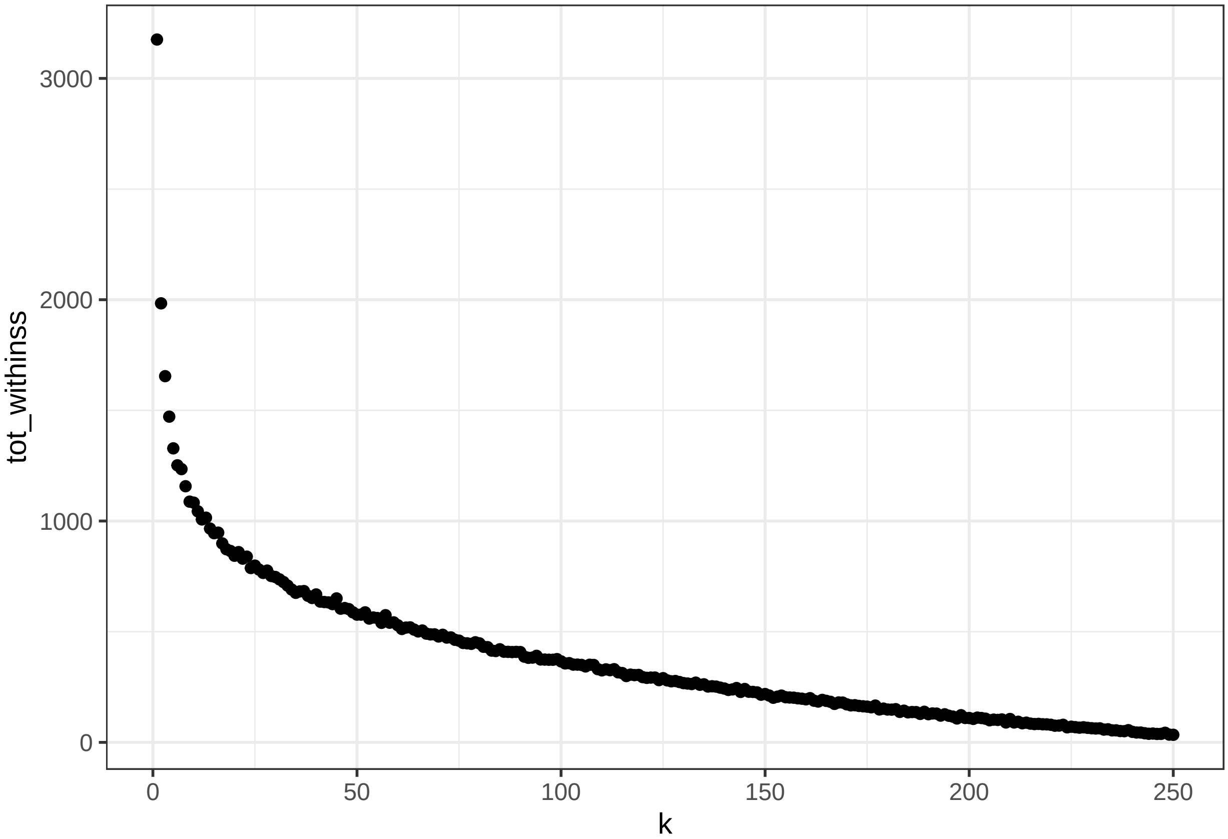


Occupational Changers Who Were Part Time Last Year because They Couldn't Find a Full−Time Job

K-Means clustering as a means of creating occupational groupings

The following graph shows how, as k increases, the within-cluster variance decreases. K-Means clustering is performed on the skills data only, and so it is NOT weighted by the prevalence of each occupation in the data. Starting at around 30 clusters, the marginal utility of each additional cluster begins to level off, perhaps indicating that this is about the most information that can be extracted from these data.

```
df <- acs[!duplicated(acs$OCC1990), .SD, .SDcols = names(acs)[names(acs) %like% "skl|knl|abl|OCC1990" &
df_occ <- df[,OCC1990]
df[, OCC1990 := NULL]

acs[, c(paste0(vars, "_current"), paste0(vars, "_ly")) := NULL]

optr <- function(k){
  stats::kmeans(df, centers = k) -> temp2
  return(data.table(k =k, tot_withinss = temp2$tot.withinss))
}

lapply(1:250, optr) %>% rbindlist -> optr_out
ggplot(optr_out) +
  geom_point(aes(x = k, y = tot_withinss))
```



See how different k affects mobility

Set number of k to correspond with meso, micro, macro class schemas and some other random numbers.

k = {4,9,67, 20,30,40,50,80,90,100,120}

The following code creates k clusters based on the skills/abilities/knowledge data.

```
c(4,9,67, 20,30,40,50,80,90,100,120) -> k_cand

for(k in k_cand){
  stats::kmeans(df, centers = k) -> temp2
  df_temp2 <- data.table(OCC1990 = df_occ, kmeans_cluster = temp2$cluster)
  setnames(df_temp2, "kmeans_cluster", paste0("kmeans_cluster_", k, "_current"))
  df_temp2[, paste0("kmeans_cluster_", k, "_current") := paste0(",", get(paste0("kmeans_cluster_", k, "
                                                                ",")]
  acs <- merge(acs, df_temp2, by = "OCC1990")

  df_temp3 <- data.table(OCC90LY = df_occ, kmeans_cluster = temp2$cluster)
  setnames(df_temp3, "kmeans_cluster", paste0("kmeans_cluster_", k, "_ly"))
  df_temp3[, paste0("kmeans_cluster_", k, "_ly") := paste0(",", get(paste0("kmeans_cluster_", k, "_ly")
                                                           ",")]

  acs <- merge(acs, df_temp3, by = "OCC90LY")


  centers <- temp2$centers
```

```
melt(centers) %>% data.table() %>% .[,.(value = sum(value)), by = Var1] %>% .[order(value, decreasing

centers_long <- melt(centers)
centers_long <- data.table(centers_long)
centers_long[, Var1 := paste0(",", Var1, ",")]
centers_long <- merge(centers_long, centers_long, by = "Var2", allow.cartesian = T)
centers_dist <- data.table(centers_long)[,.(geo_mean_dist = mean(abs(value.x - value.y))),
                                         by = .(Var1.x, Var1.y)]
setnames(centers_dist, c(paste0("kmeans_cluster_", k, "_current"),
                         paste0("kmeans_cluster_", k, "_ly"), paste0("geo_means_", k, "_dist")))

# create vectors of nearest neighbors
for(k_temp in paste0(",", 1:k, ",")){
  centers_dist[get(paste0("kmeans_cluster_", k, "_ly")) == k_temp] %>%
    .[order(get(paste0("geo_means_", k, "_dist")))] %>%
    .[,get(paste0("kmeans_cluster_", k, "_current"))] %>%
    .[1:3] %>%
    paste0(., collapse = "|") -> tempp
  centers_dist[get(paste0("kmeans_cluster_", k, "_ly")) == k_temp,
               paste0("kmeans_cluster_rank_3_", k, "_conc"):=
                 ifelse(get(paste0("kmeans_cluster_", k, "_current")) %like% tempp, 1,0) ]
}



#
acs <- merge(acs, centers_dist, by = c(paste0("kmeans_cluster_", k, "_current"),
                                       paste0("kmeans_cluster_", k, "_ly")))
centers_dist[,   paste0("kmeans_cluster_", k, "_ly") :=
               as.numeric(gsub(",", "", get(  paste0("kmeans_cluster_", k, "_ly"))))]
centers_dist[,   paste0("kmeans_cluster_", k, "_current") :=
               as.numeric(gsub(",", "", get(  paste0("kmeans_cluster_", k, "_current"))))]
acs[,   paste0("kmeans_cluster_", k, "_ly") :=
      as.numeric(gsub(",", "", get(  paste0("kmeans_cluster_", k, "_ly"))))]
acs[,   paste0("kmeans_cluster_", k, "_current") :=
      as.numeric(gsub(",", "", get(  paste0("kmeans_cluster_", k, "_current"))))]

acs[,paste0("kmeans_cluster_", k, "_conc") := ifelse(get(paste0("kmeans_cluster_", k, "_current")) ==
                                                       get(paste0("kmeans_cluster_", k, "_ly")),1,0)]

acs[, paste0("source_kmeans_", k, "_N") := .N, by = get(paste0("kmeans_cluster_", k, "_ly"))]

acs[, paste0("kmeans_cluster_", k, "_current") := factor(get(paste0("kmeans_cluster_", k, "_current"))
                                                         levels = centers_dist[get(paste0("kmeans_clu
                                                           .[order(get(paste0("geo_means_", k, "_dist
                                                           .[,get(paste0("kmeans_cluster_", k, "_ly")
acs[, paste0("kmeans_cluster_", k, "_ly") := factor(get(paste0("kmeans_cluster_", k, "_ly")),
                                                    levels = centers_dist[get(paste0("kmeans_cluster_
                                                      .[order(get(paste0("geo_means_", k, "_dist")))]
                                                      .[,get(paste0("kmeans_cluster_", k, "_ly"))])]
}
```

Check concordance with meso/micro/macro class schedules

4, 9, and 67 correspond with micro, meso, and macro respectively. While perfect concordance is less good, if we extend the reach of the algorithm to people in either of the top 3 closest skills categories, we see improved performance.

```
### check concordance of micro meso and macro first

num_correct <- acs[OCC1950 != OCC50LY & !is.na(kmeans_cluster_30_current),
                   colSums(.SD, na.rm = T), .SDcols = names(acs)[names(acs) %like% "conc"]]
total <- acs[OCC1950 != OCC50LY , nrow(.SD)]
num_correct <- num_correct/total
num_cats <- acs[OCC1950 != OCC50LY, lapply(.SD, FUN = function(x){length(unique(x))}), .SDcols = names(

num_correct[names(num_correct) %like% "meso|macro|micro|4_|9_|67_"] -> out
out <- melt(out) %>% data.table(., keep.rownames = T)
out[,id := c(2,1,3,1,1,2,2,3,3)]
out[,method := c("Traditional","Traditional","Traditional",
                 "Skills-Based (Fuzzy)","Skills-Based",
                 "Skills-Based (Fuzzy)","Skills-Based",
                 "Skills-Based (Fuzzy)","Skills-Based")]
out <- dcast(out, method ~ id)
setnames(out, c("Method", "Micro (4)", "Meso (9)", "Macro (67)"))
xtable::xtable(out)
```

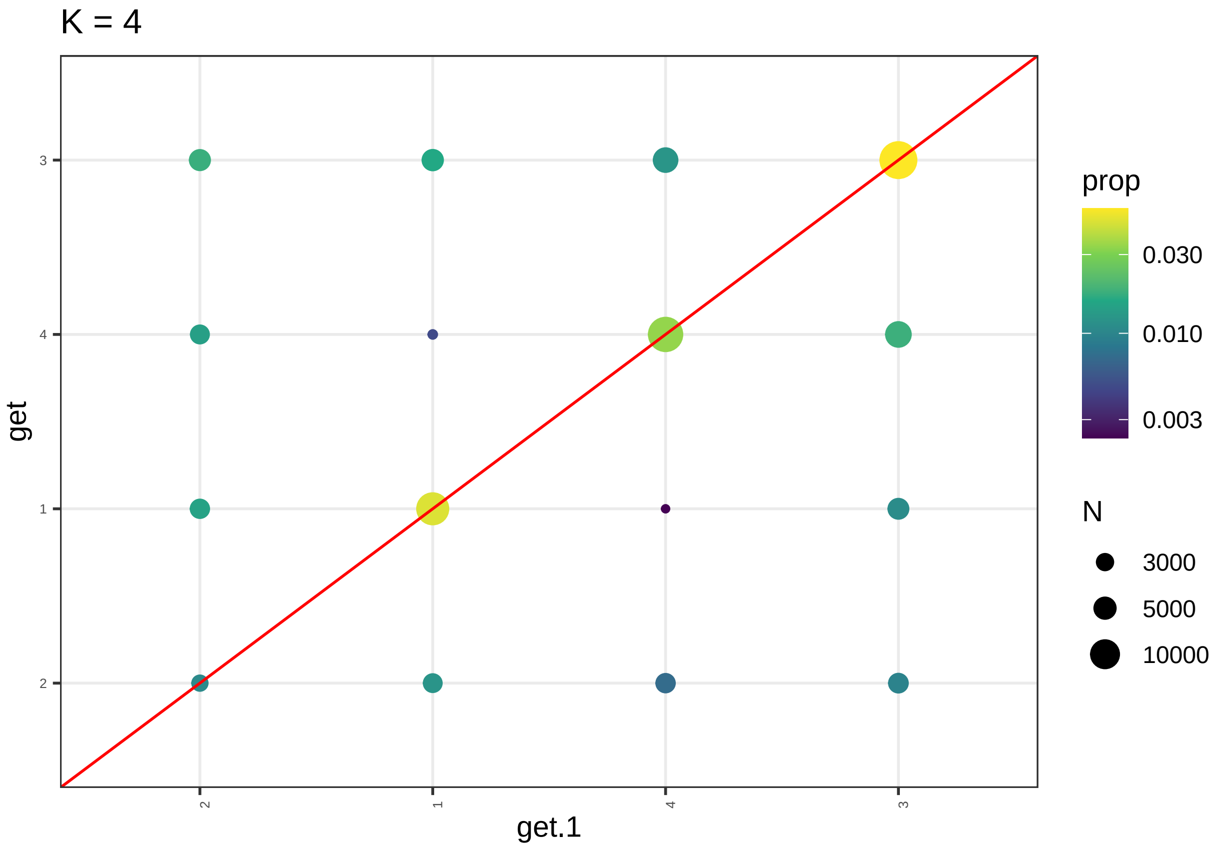
% latex table generated in R 4.0.2 by xtable 1.8-4 package % Fri Dec 4 13:00:38 2020

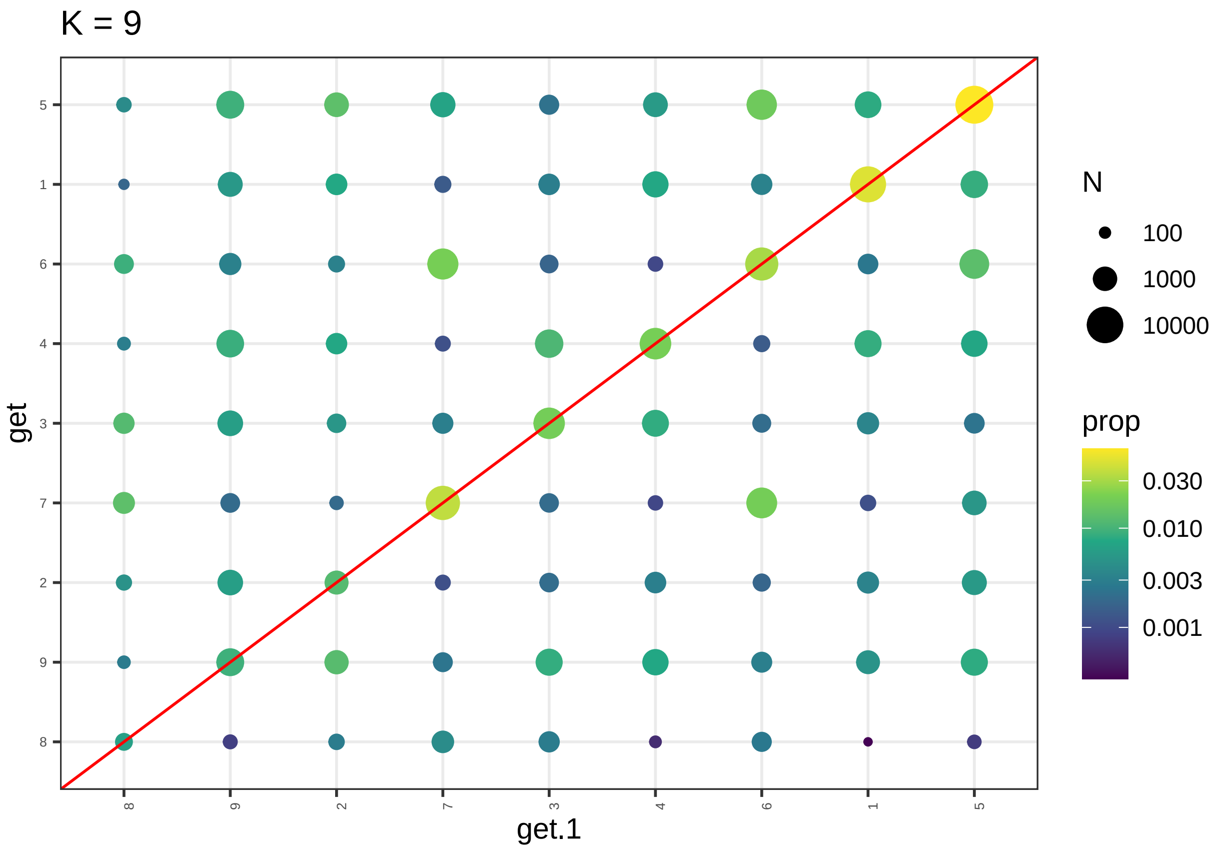
	Method	Micro (4)	Meso (9)	Macro (67)
1	Skills-Based	0.51	0.37	0.12
2	Skills-Based (Fuzzy)	0.85	0.63	0.26
3	Traditional	0.70	0.44	0.16

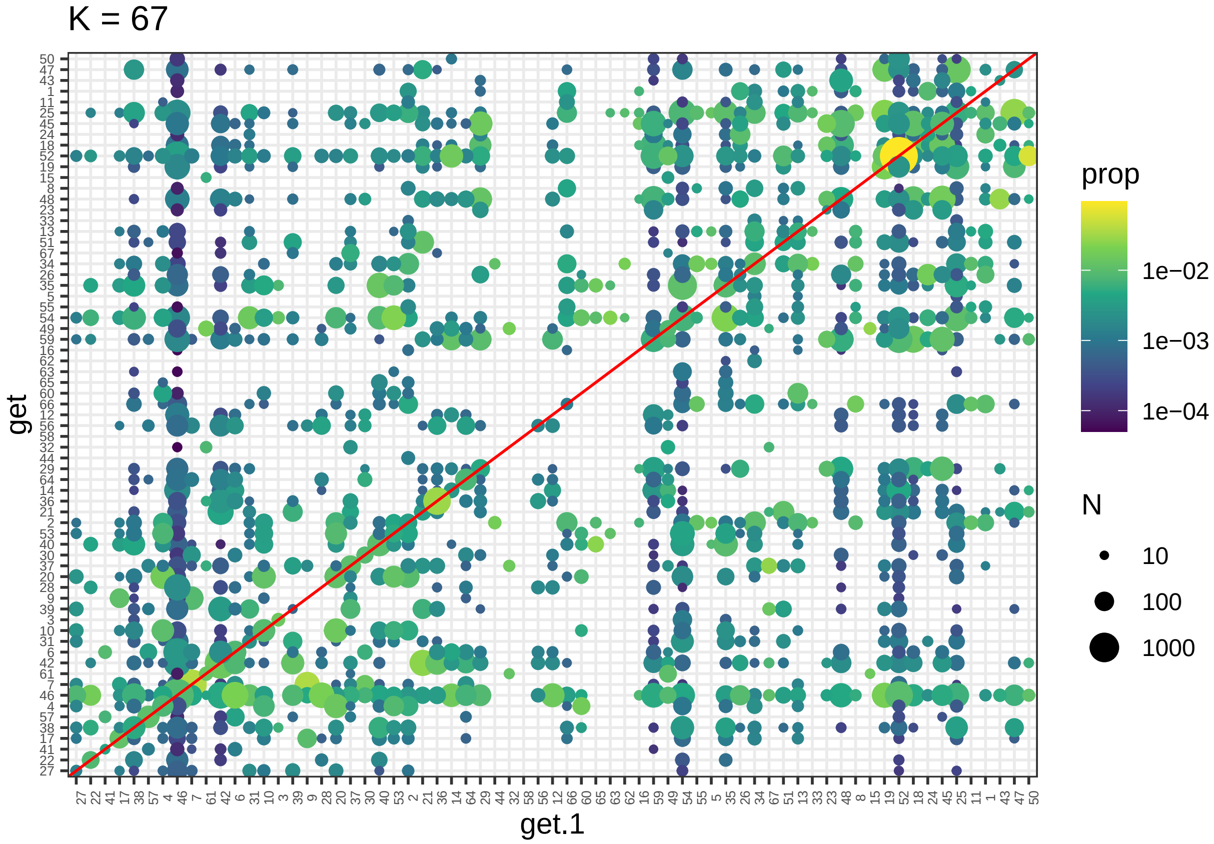
show overall job migration

These graphs are 2D versions of the 3d graphs used in the Grusky Microclass paper

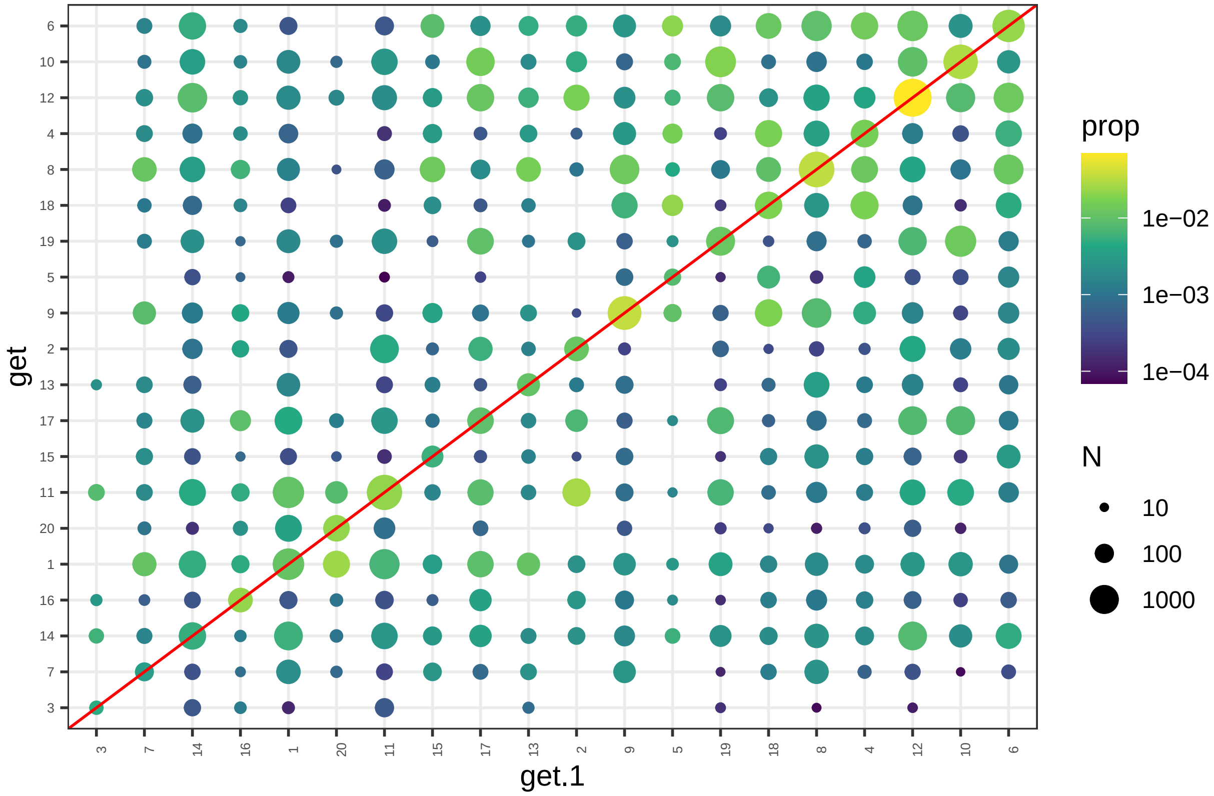
```
for(k in k_cand){
  acs[OCC90LY != OCC1990,.(N = .N, source_kmeans_N = unique(get(paste0("source_kmeans_", k, "_N") ))),
      by= .(get(paste0("kmeans_cluster_", k, "_current")), get(paste0("kmeans_cluster_", k, "_ly")))] %
    .[, total := sum(N), by = get.1] %>%
    .[, prop := N/source_kmeans_N] %>%
    .[N >= 10] %>%
    ggplot() +
    geom_point(aes(y = get,
                   x = get.1, color = prop, size = N)) +
    scale_color_viridis_c(trans = "log10") +
    scale_radius(trans = "log10")+
    geom_abline(yintercept = 0, slope = 1, color= "red") +
    theme(axis.text.x = element_text(angle = 90),
          axis.text = element_text(size = 5)) +
    scale_x_discrete(drop = F)+
    scale_y_discrete(drop = F) +
    ggtitle(paste0("K = ", k))-> gg
  print(gg)
}
```



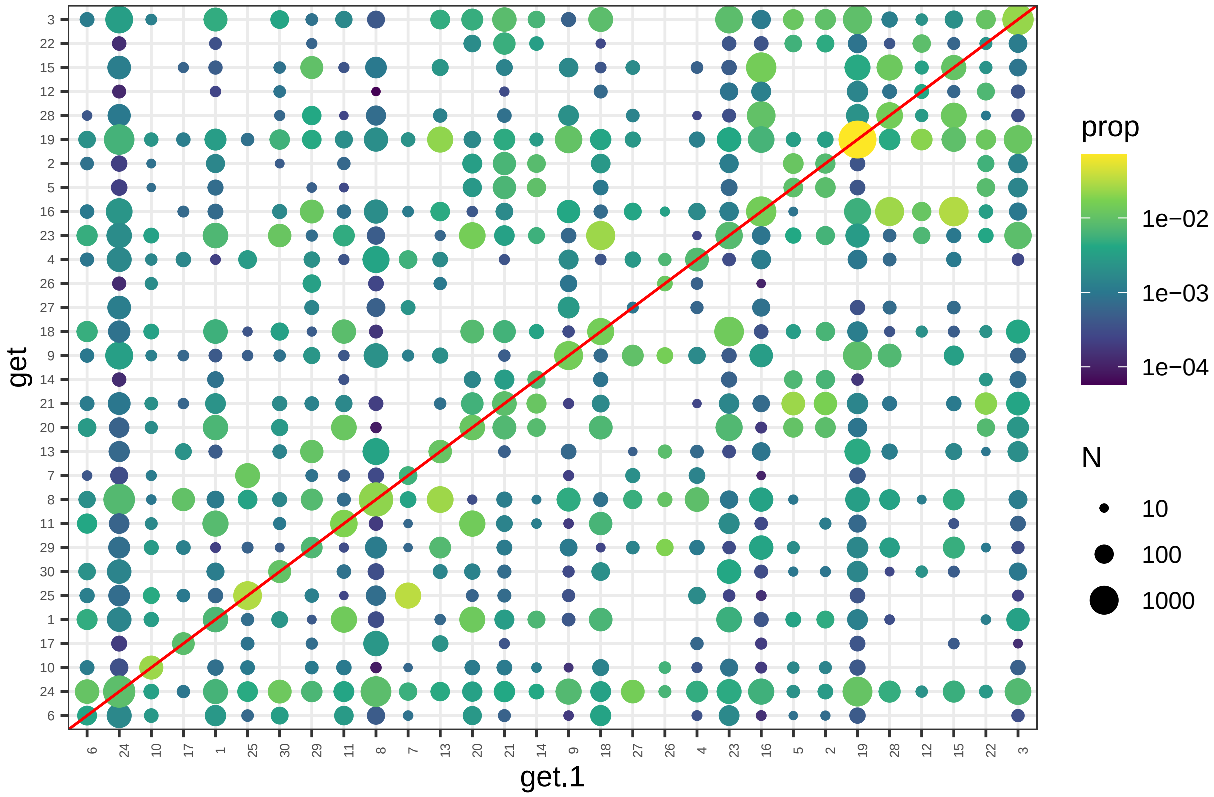

K = 9
get
5
1
6
4
3
7
2
9
8
get.1
8
9
2
7
3
4
6
1
5
N
100
1000
10000
prop
0.030
0.010
0.003
0.001

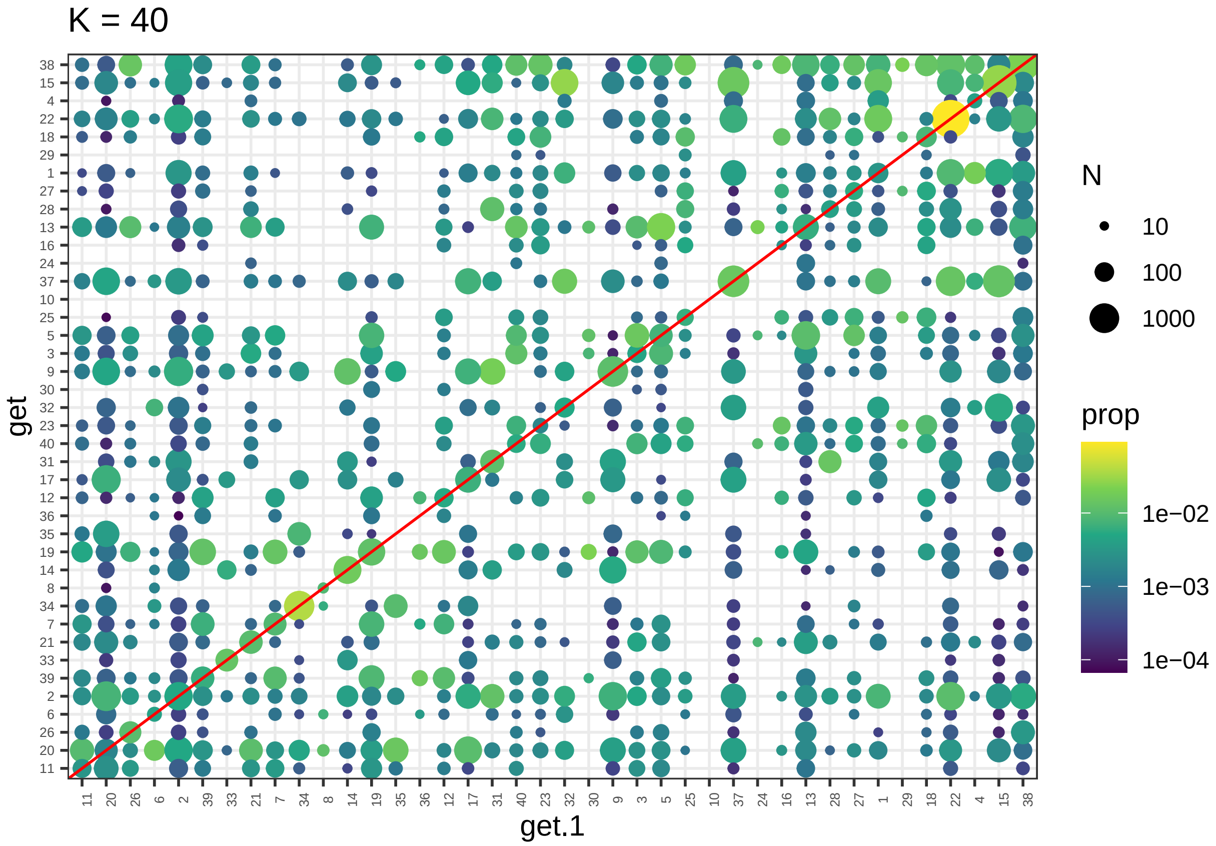

K = 67
get
50
47
43
1
11
25
45
24
18
52
19
15
8
48
23
33
13
51
67
34
26
35
5
55
54
49
59
16
62
63
65
60
66
12
56
58
32
44
29
64
14
36
21
2
53
40
30
37
20
28
9
39
3
10
31
6
42
61
7
46
4
57
38
17
41
22
27
get.1
27
22
41
17
38
57
4
46
7
61
42
6
31
10
3
39
9
28
20
37
30
40
53
2
21
36
14
64
29
44
32
58
56
12
66
60
65
63
62
16
59
49
54
55
5
35
26
34
67
51
13
33
23
48
8
15
19
52
18
24
45
25
11
1
43
47
50
prop
1e−02
1e−03
1e−04
N
10
100
1000

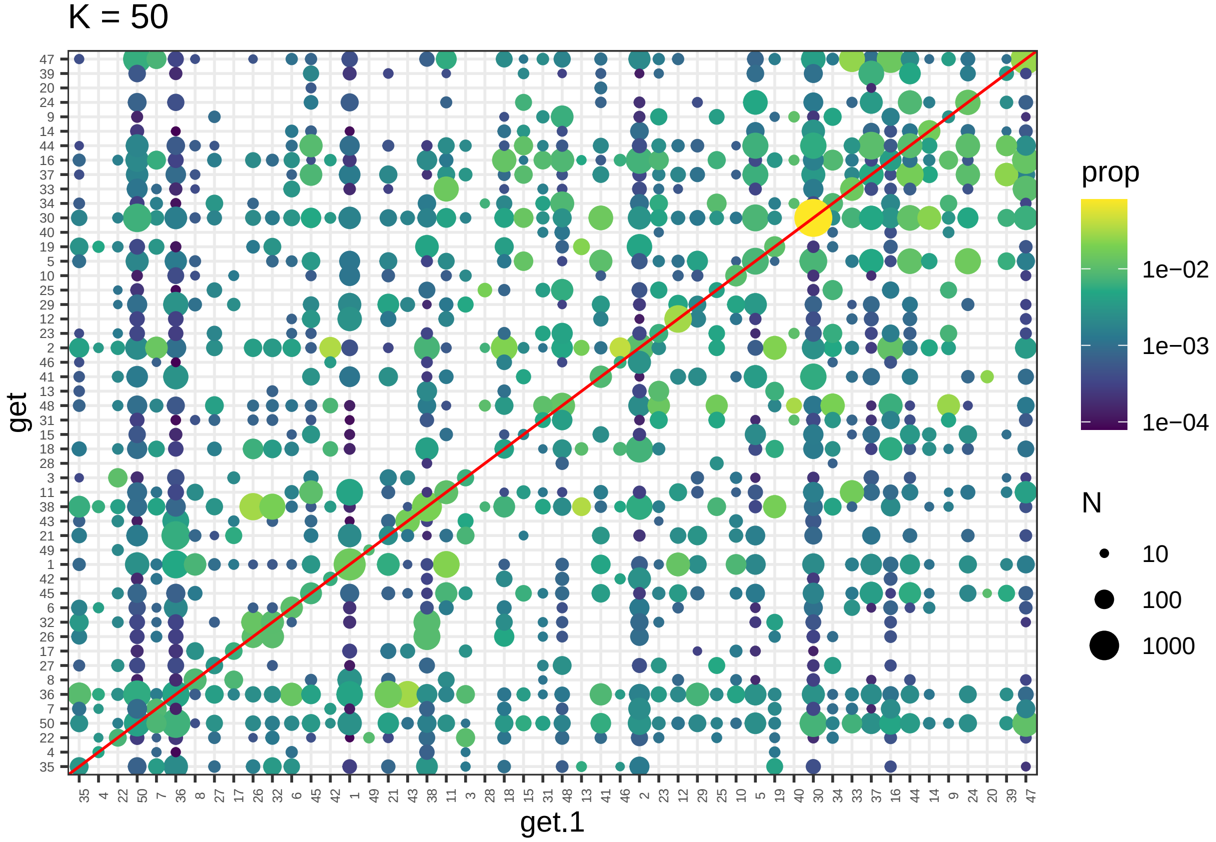
K = 20

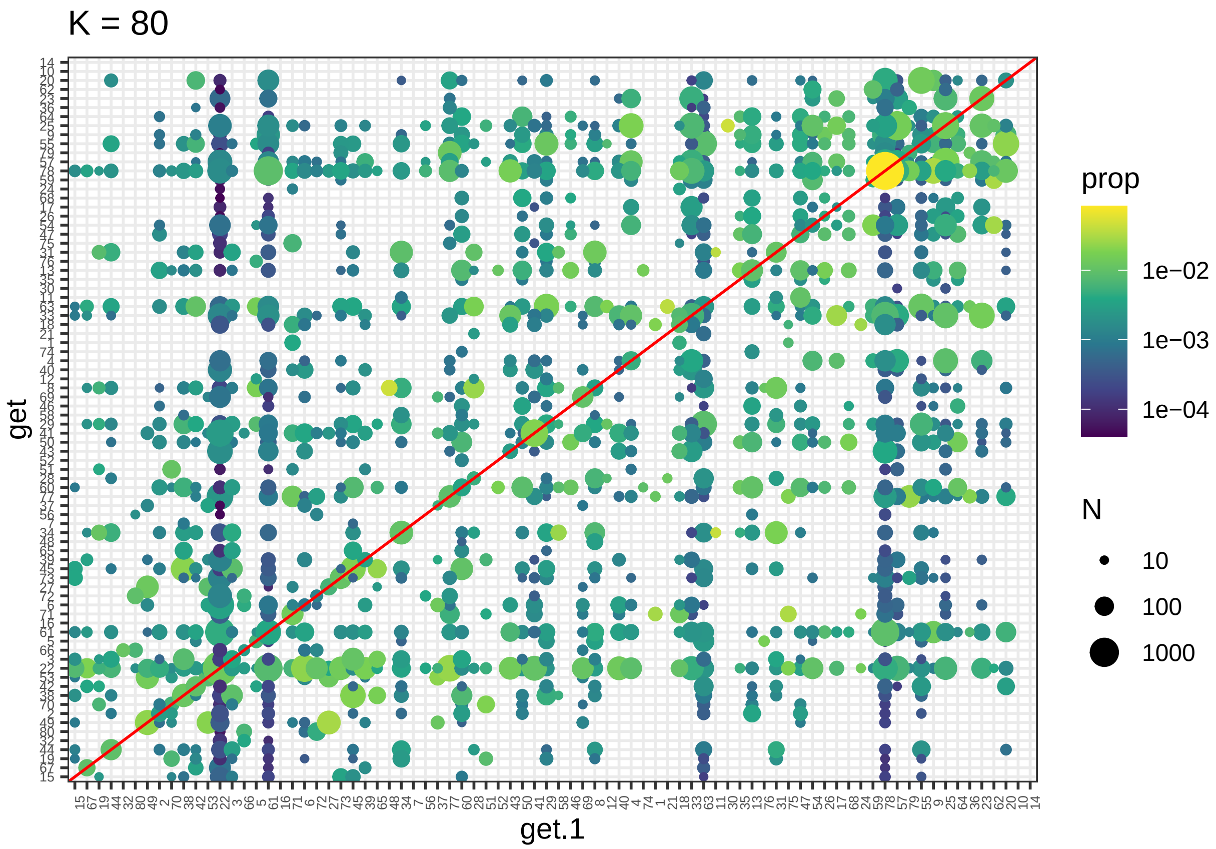

get
6
10
12
4
8
18
19
5
9
2
13
17
15
11
20
1
16
14
7
3
3
7
14
16
1
20
11
15
17
13
2
9
5
19
18
8
4
12
10
6
get.1
prop
1e−02
1e−03
1e−04
N
10
100
1000

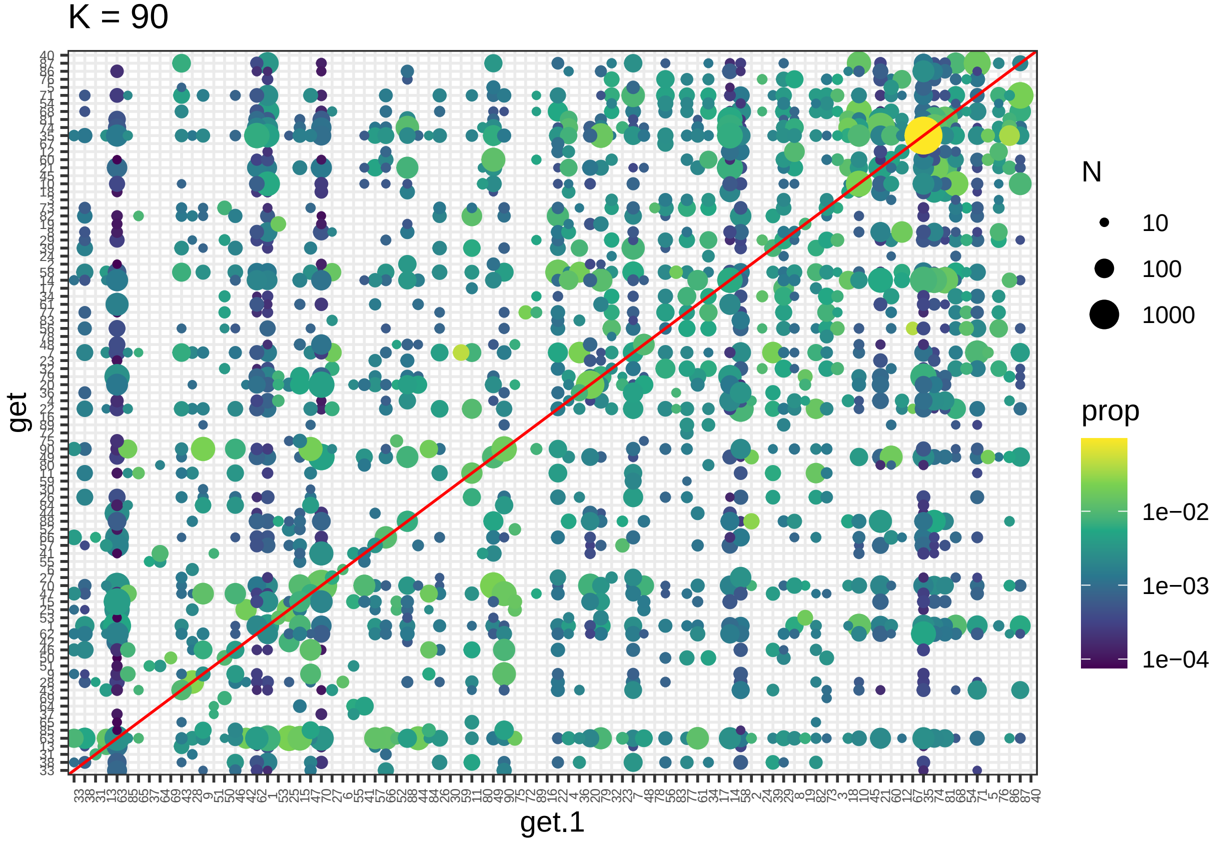
K = 30

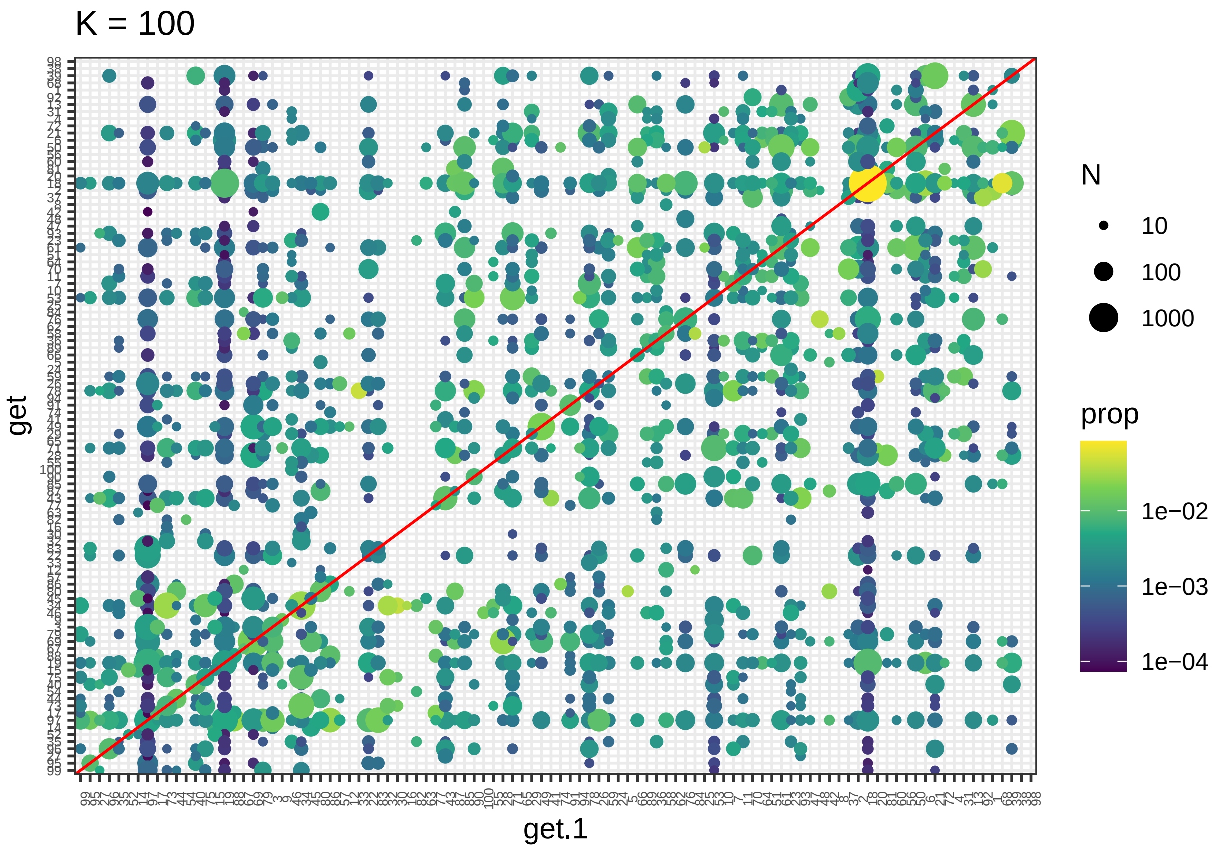

get
3
22
15
12
28
19
2
5
16
23
4
26
27
18
9
14
21
20
13
7
8
11
29
30
25
1
17
10
24
6
6
24
10
17
1
25
30
29
11
8
7
13
20
21
14
9
18
27
26
4
23
16
5
2
19
28
12
15
22
3
get.1
prop
1e−02
1e−03
1e−04
N
10
100
1000


K = 40
get
38
15
4
22
18
29
1
27
28
13
16
24
37
10
25
5
3
9
30
32
23
40
31
17
12
36
35
19
14
8
34
7
21
33
39
2
6
26
20
11
get.1
11
20
26
6
2
39
33
21
7
34
8
14
19
35
36
12
17
31
40
23
32
30
9
3
5
25
10
37
24
16
13
28
27
1
29
18
22
4
15
38
N
10
100
1000
prop
1e−02
1e−03
1e−04

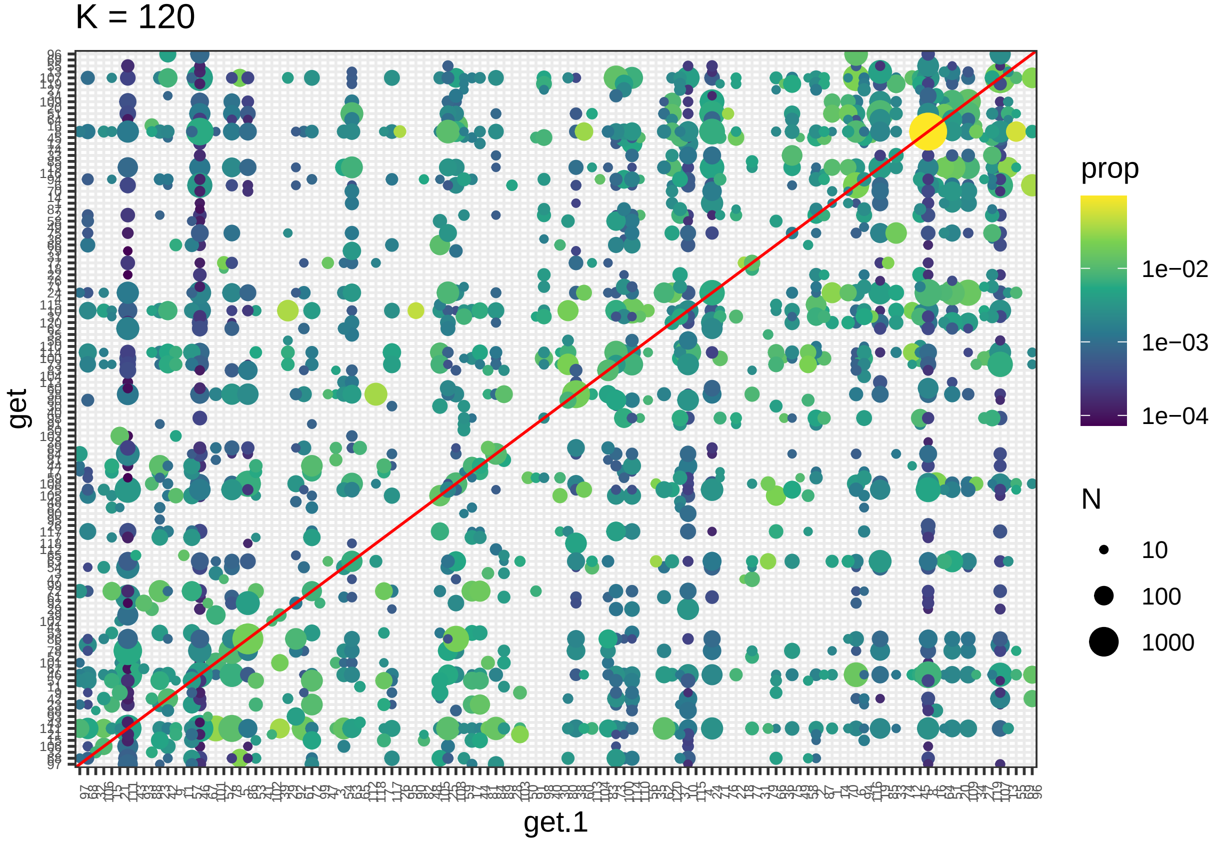

K = 50
get
47
39
20
24
9
14
44
16
37
33
34
30
40
19
5
10
25
29
12
23
2
46
41
13
48
31
15
18
28
3
11
38
43
21
49
1
42
45
6
32
26
17
27
8
36
7
50
22
4
35
get.1
35
4
22
50
7
36
8
27
17
26
32
6
45
42
1
49
21
43
38
11
3
28
18
15
31
48
13
41
46
2
23
12
29
25
10
5
19
40
30
34
33
37
16
44
14
9
24
20
39
47
prop
1e−02
1e−03
1e−04
N
10
100
1000


K = 80
get
14 10 20 62 23 36 64 25 9 55 79 57 78 59 24 68 17 26 54 47 75 31 76 13 35 30 11 63 33 18 21 1 74 4 40 12 8 69 46 58 29 41 50 43 52 51 28 60 77 37 56 7 34 48 65 39 45 73 27 72 6 71 16 61 5 66 3 22 53 42 38 70 2 49 80 32 44 19 67 15
get.1
15 67 19 44 32 80 49 2 70 38 42 53 22 3 66 5 61 16 71 6 72 27 73 45 39 65 48 34 7 56 37 77 60 28 51 52 43 50 41 29 58 46 69 8 12 40 4 74 1 21 18 33 63 11 30 35 13 76 31 75 47 54 26 17 68 24 59 78 57 79 55 9 25 64 36 23 62 20 10 14
prop
1e-02
1e-03
1e-04
N
10
100
1000


K = 90
get
40 87 86 76 5 71 54 68 81 74 35 67 12 60 21 45 10 18 3 73 82 19 8 29 39 24 2 58 14 17 34 61 77 83 56 78 48 7 23 32 79 20 36 4 22 16 89 72 75 90 49 80 11 59 30 26 84 44 88 52 66 57 41 55 6 27 70 47 15 25 53 1 62 42 46 50 51 9 28 43 69 64 37 65 85 63 13 31 38 33
get.1
33 38 31 13 63 85 65 37 64 69 43 28 9 51 50 46 42 62 1 53 25 15 47 70 27 6 55 41 57 66 52 88 44 84 26 30 59 11 80 49 90 75 72 89 16 22 4 36 20 79 32 23 7 48 78 56 83 77 61 34 17 14 58 2 24 39 29 8 19 82 73 3 18 10 45 21 60 12 67 35 74 81 68 54 71 5 76 86 87 40
N
10
100
1000
prop
1e-02
1e-03
1e-04


K = 100

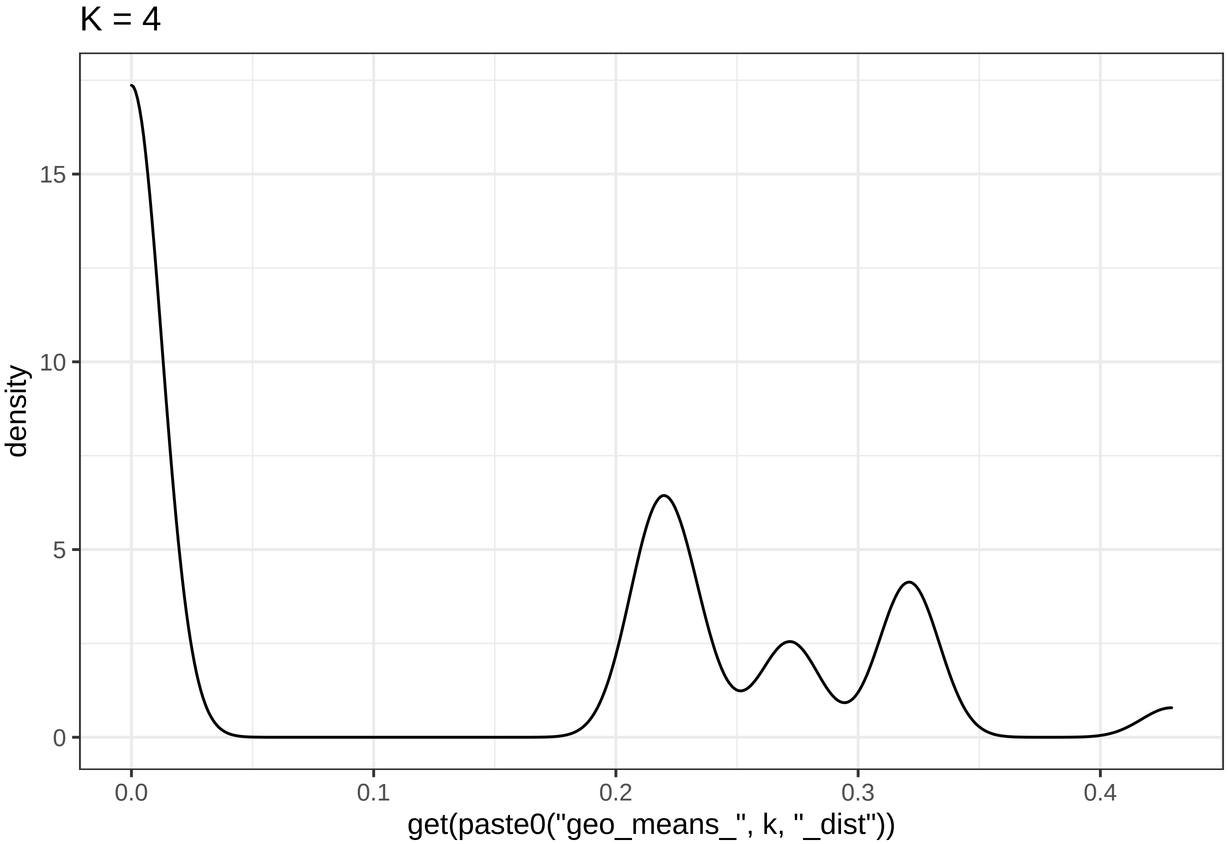
get
get.1
N
10
100
1000
prop
1e−02
1e−03
1e−04

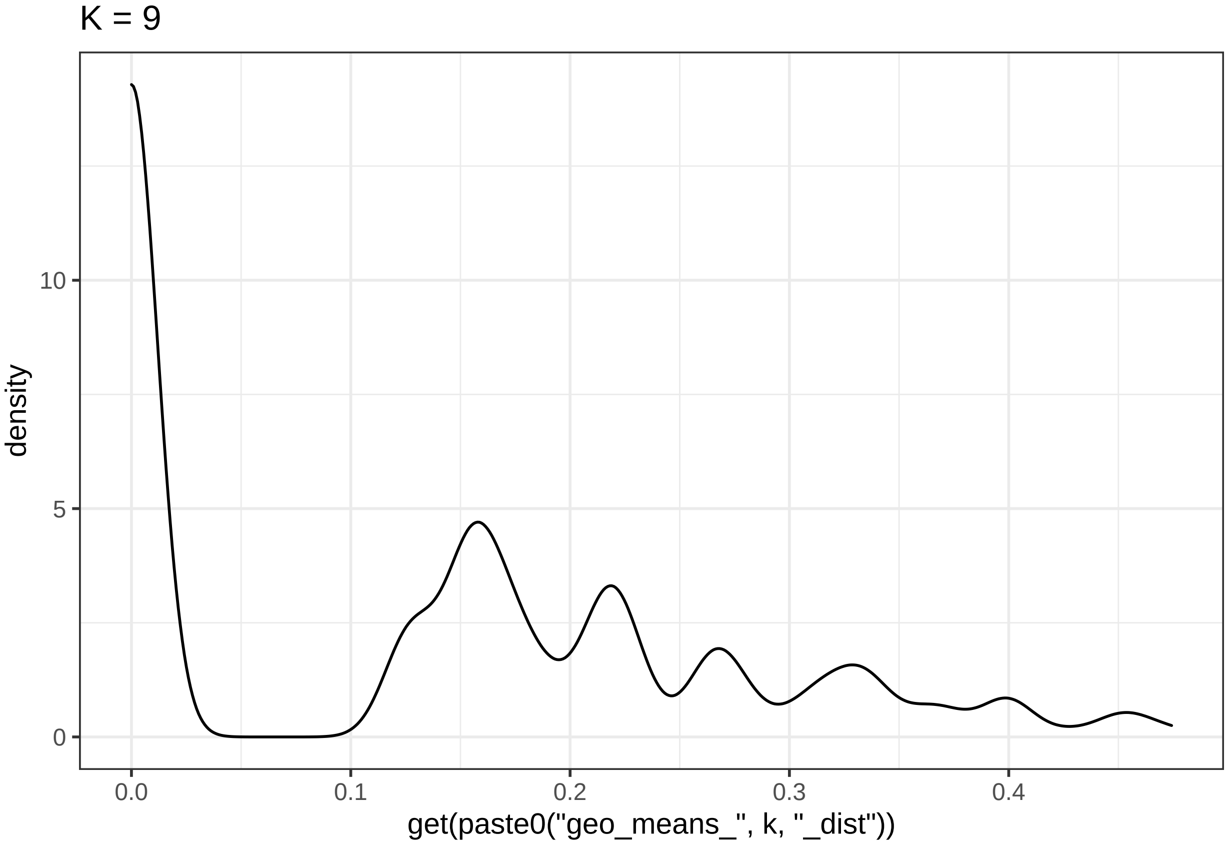

K = 120

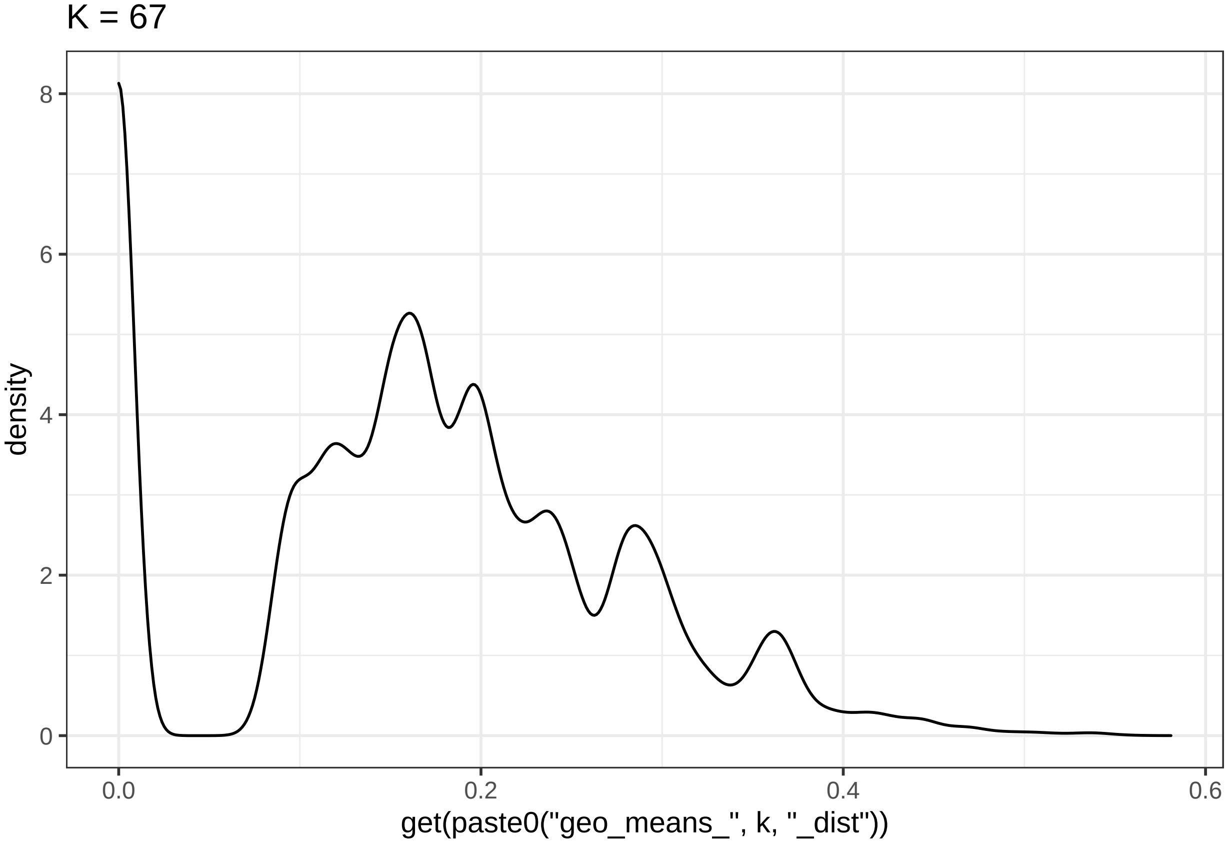
get
get.1
prop
1e−02
1e−03
1e−04
N
10
100
1000

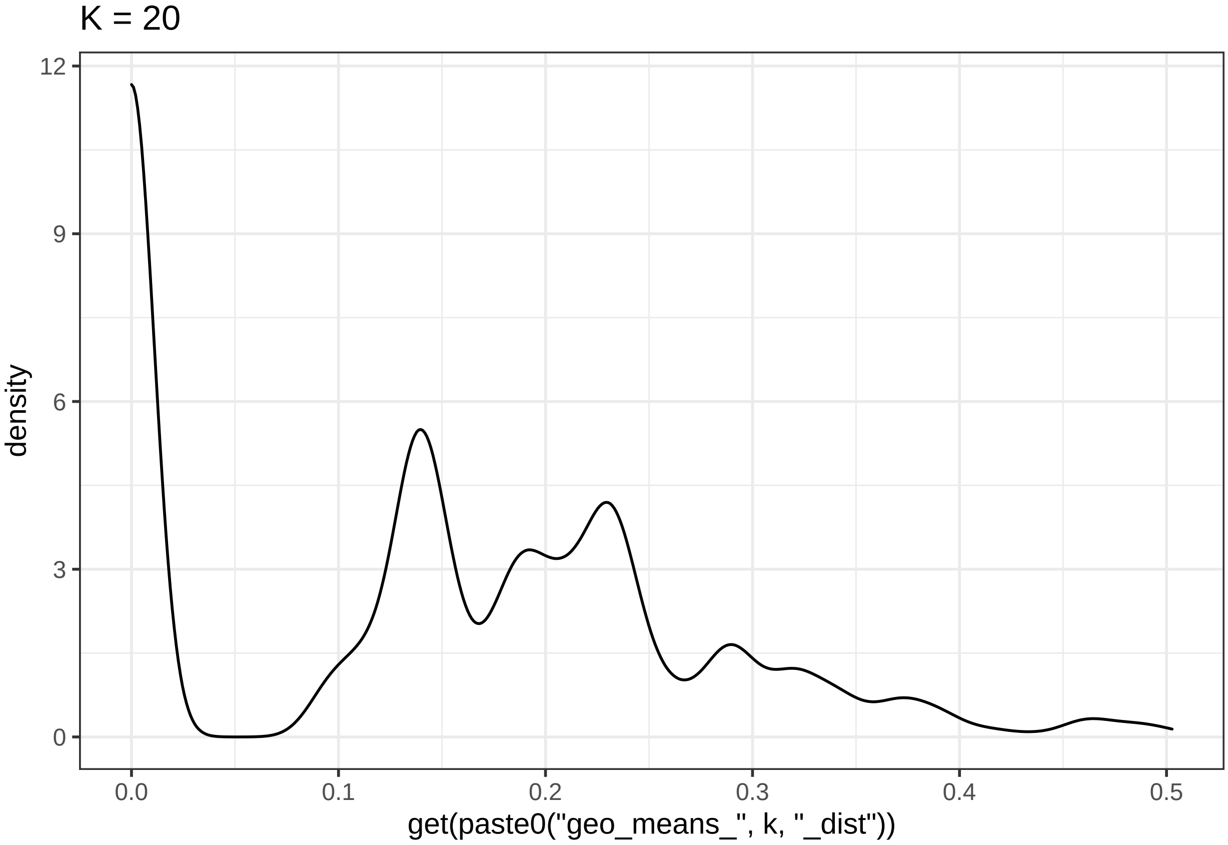
These graphs show the distribution of skills distances for all occupational migrants

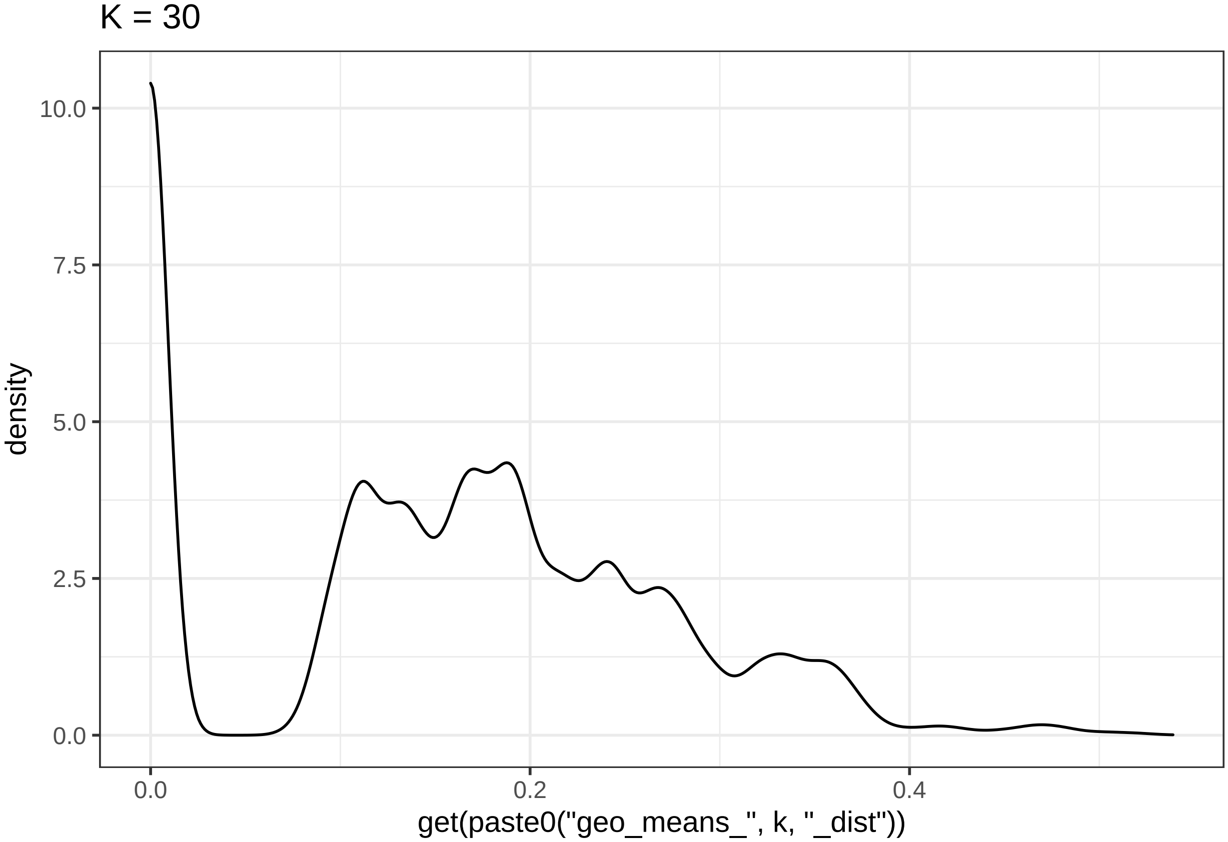
```r
for(k in k_cand){
  ggplot(acs[OCC90LY != OCC1990])+
    geom_density(aes(x = get(paste0("geo_means_", k, "_dist")))) +
    ggtitle(paste0("K = ", k)) -> gg
  print(gg)
}
```

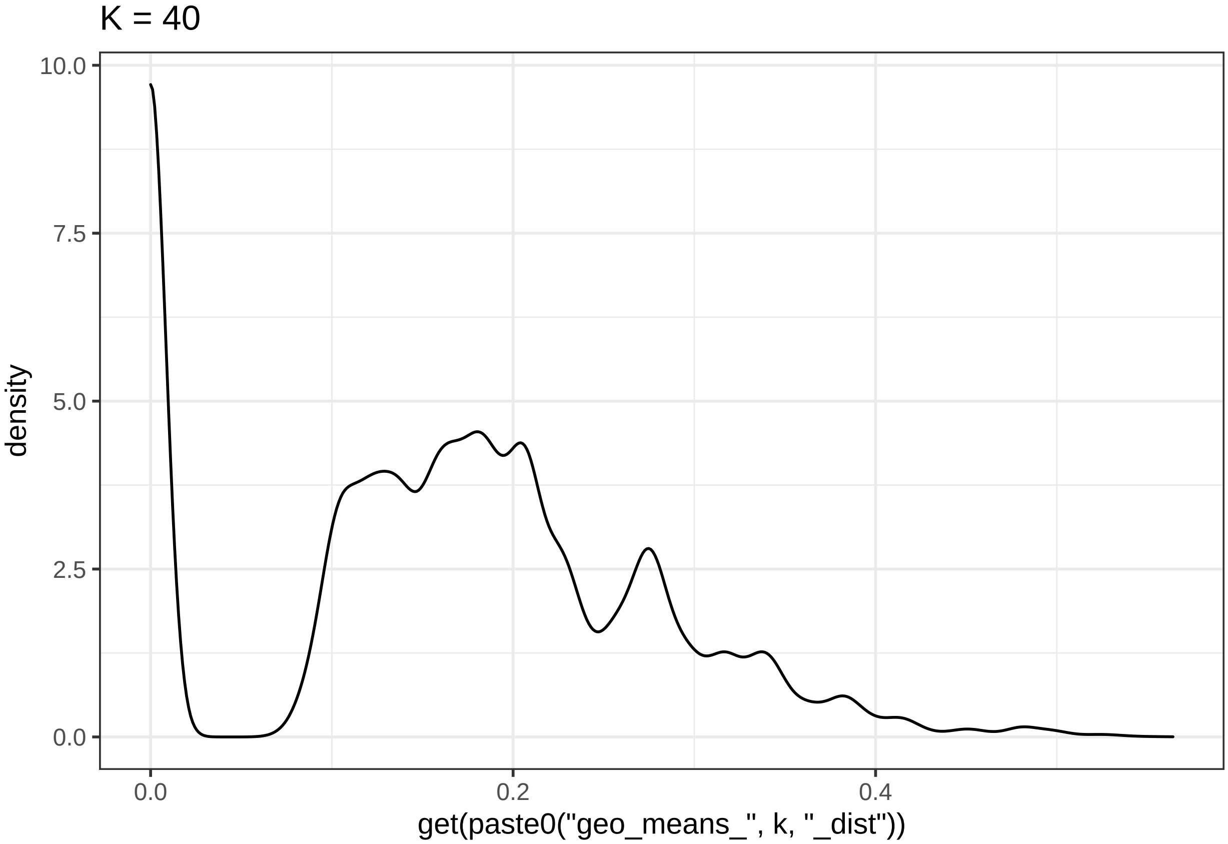


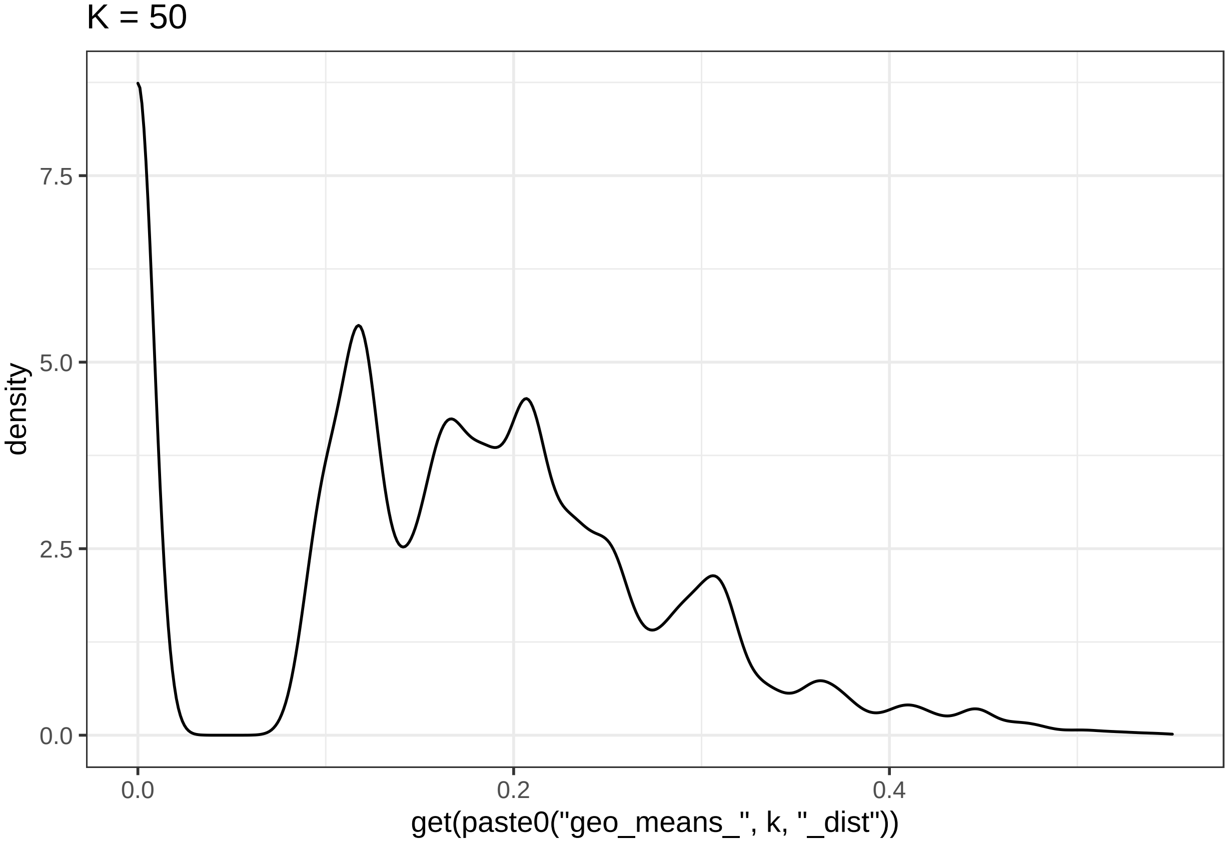

K = 9
density
10
5
0
0.0
0.1
0.2
0.3
0.4
get(paste0("geo_means_", k, "_dist"))

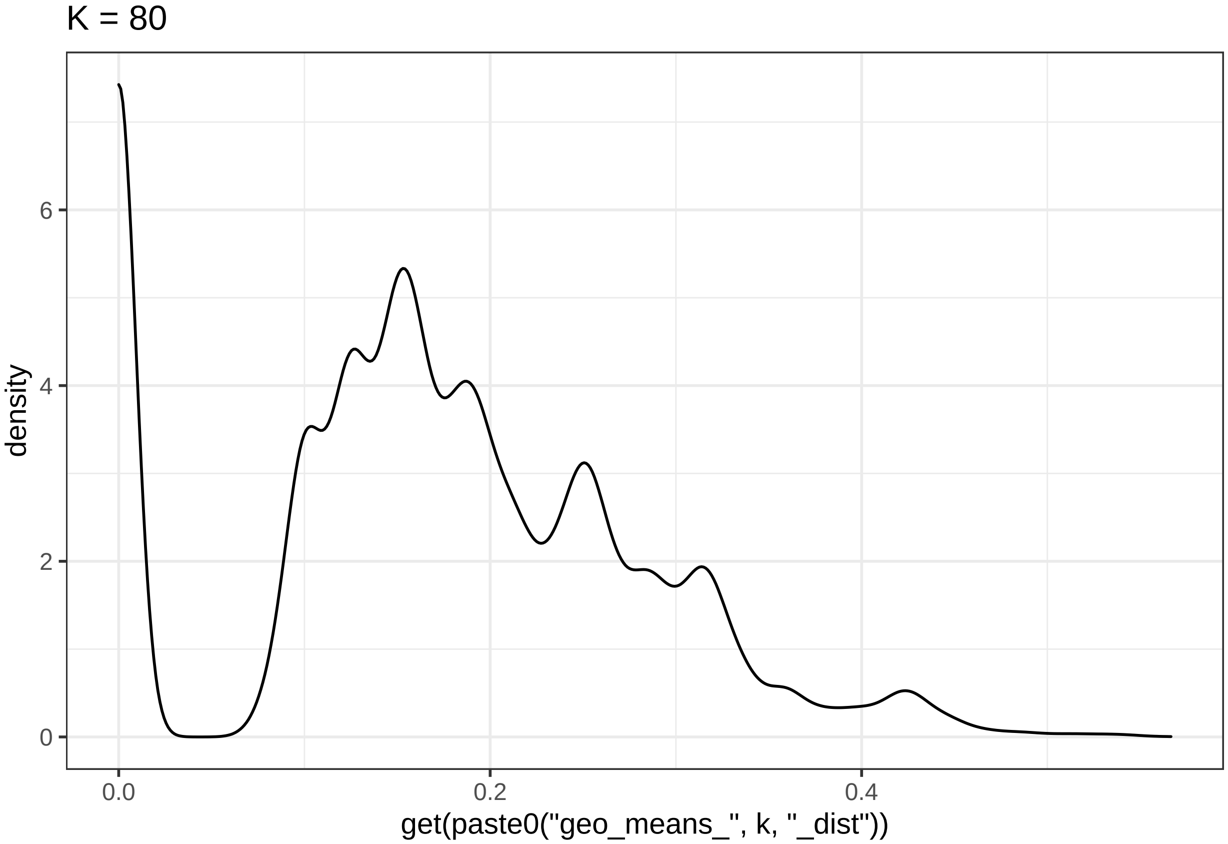

K = 67
density
8
6
4
2
0
0.0
0.2
0.4
0.6
get(paste0("geo_means_", k, "_dist"))

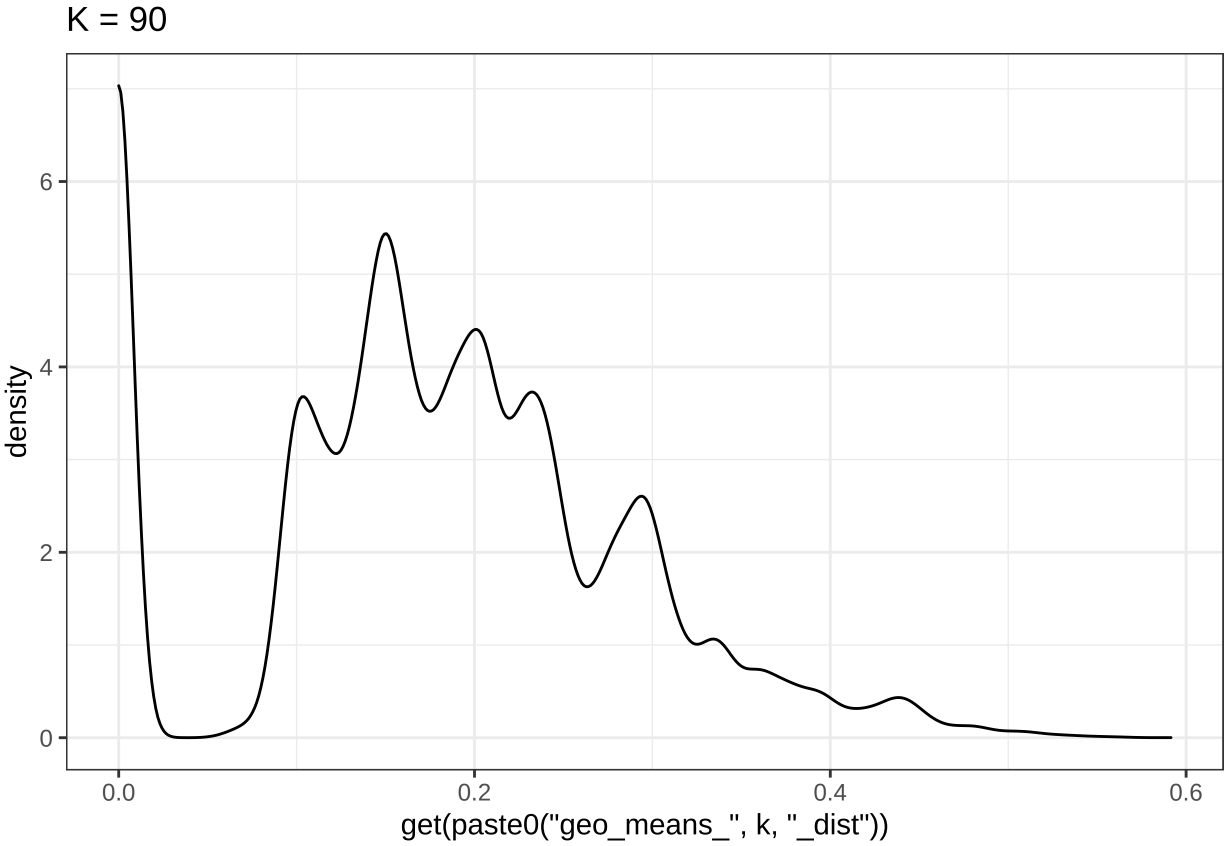

K = 20
density
12
9
6
3
0
0.0
0.1
0.2
0.3
0.4
0.5
get(paste0("geo_means_", k, "_dist"))

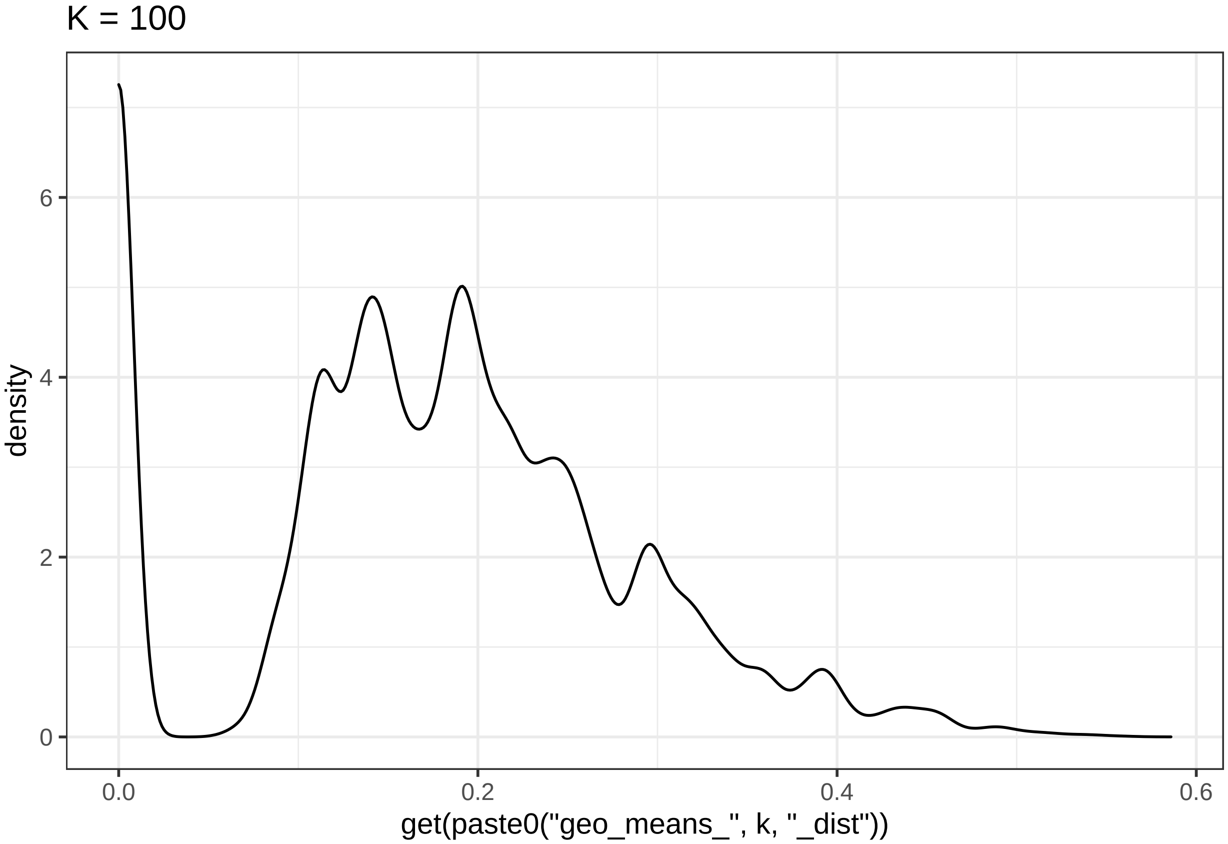

K = 30
density
10.0
7.5
5.0
2.5
0.0
0.0
0.2
0.4
get(paste0("geo_means_", k, "_dist"))

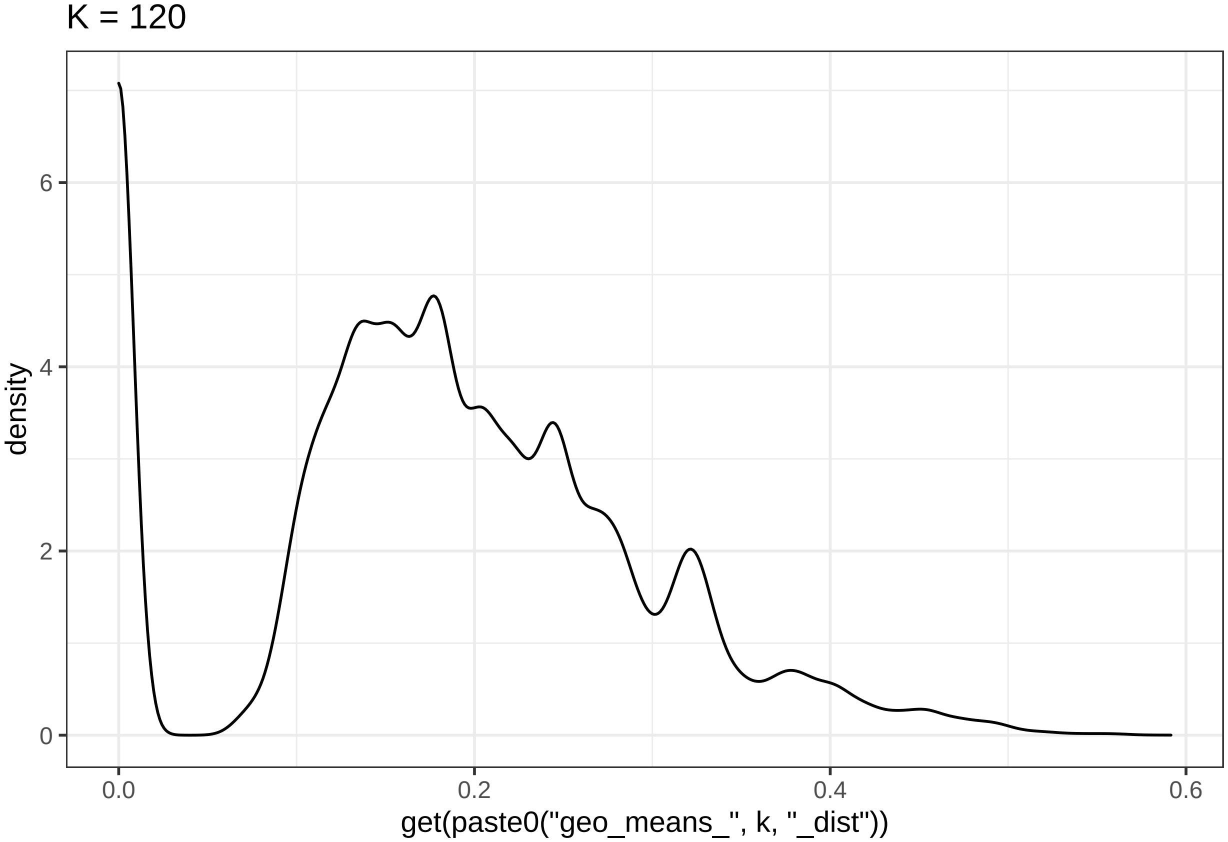

K = 40
density
10.0
7.5
5.0
2.5
0.0
0.0
0.2
0.4
get(paste0("geo_means_", k, "_dist"))


K = 50
density
7.5
5.0
2.5
0.0
0.0
0.2
0.4
get(paste0("geo_means_", k, "_dist"))


K = 80
density
0
2
4
6
0.0
0.2
0.4
get(paste0("geo_means_", k, "_dist"))


K = 90
density
0
2
4
6
0.0
0.2
0.4
0.6
get(paste0("geo_means_", k, "_dist"))

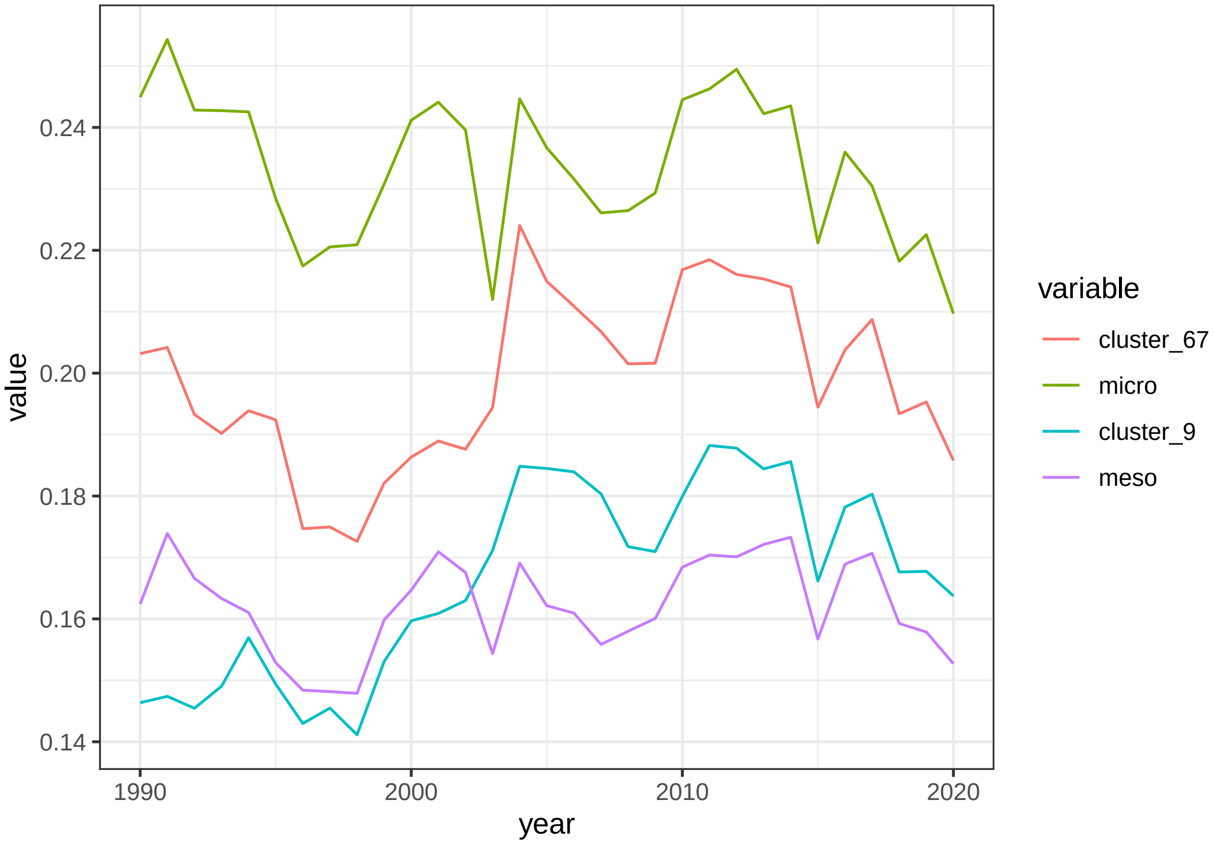

K = 100
density
0
2
4
6
0.0
0.2
0.4
0.6
get(paste0("geo_means_", k, "_dist"))


K = 120
density
0
2
4
6
0.0
0.2
0.4
0.6
get(paste0("geo_means_", k, "_dist"))

```
# df_temp <- data.table(OCC2010 = df_occ, kmeans_cluster = temp$cluster)
#
# acs <- merge(acs, df_temp, by = "OCC2010")
# setnames(acs, "kmeans_cluster", "kmeans_cluster_current")
#
# df_temp <- data.table(OCC10LY = df_occ, kmeans_cluster = temp$cluster)
# acs <- merge(acs, df_temp, by = "OCC10LY")
# setnames(acs, "kmeans_cluster", "kmeans_cluster_ly")
#
# # optimize k
# optr <- function(k){
#   stats::kmeans(df, centers = k) -> temp2
#   return(temp2$tot.withinss)
# }


# gg_list <- list()
# q <- 0
# for(k in seq(10,110,20)){
#   q <- q + 1
#   print(k)
#   df <- data.table(rbind(acs[, .SD, .SDcols = names(acs)[names(acs) %like% "LV_current" & !names(acs)
#   stats::kmeans(df, centers = k) -> temp
#
#   # assign clusters
#   acs[, kmeans_cluster_current := temp$cluster[1:nrow(acs)]]
#   acs[, kmeans_cluster_ly := temp$cluster[(nrow(acs) + 1):(nrow(acs)*2)]]
#
#   #
#   # get matrix of means and compute distances between each cluster
#   # geometric means
#   centers <- temp$centers
#   centers_long <- melt(centers)
#   centers_long <- merge(centers_long, centers_long, by = "Var2", allow.Cartesian = T)
#   centers_dist <- data.table(centers_long)[,.(geo_mean_dist = prod(abs(value.x - value.y)) ^ (1/lengt
#                                             by = .(Var1.x, Var1.y)]
#   setnames(centers_dist, c("kmeans_cluster_current", "kmeans_cluster_ly", "geo_mean_dist"))
#
#   #
#   acs[, geo_mean_dist := NULL]
#   acs <- merge(acs, centers_dist, by = c("kmeans_cluster_current", "kmeans_cluster_ly"))
#
#   #
#   acs[, source_kmeans_N := .N, by = kmeans_cluster_ly]
#   centers <- data.table(centers)
#   centers[,total_skills := rowSums(.SD), .SDcols = names(centers)]
#
#   acs[, kmeans_cluster_current := factor(kmeans_cluster_current,
#                                           levels = centers[, order(total_skills)])]
#   acs[, kmeans_cluster_ly := factor(kmeans_cluster_ly,
#                                      levels = centers[, order(total_skills)])]
#
#
#   gg <- acs[OCC10LY != OCC2010,.(N = .N, source_kmeans_N = unique(source_kmeans_N)),
#             by= .(kmeans_cluster_current, kmeans_cluster_ly)] %>%
#     .[, total := sum(N), by = kmeans_cluster_ly] %>%
#     .[, prop := N/source_kmeans_N] %>%
#     .[N >= 10] %>%
#     ggplot()+
#       geom_point(aes(y = (kmeans_cluster_current), x = (kmeans_cluster_ly), color = prop, size = N))
#   scale_color_viridis_c(trans = "log10") +
#   scale_radius(trans = "log10")+
#   geom_abline(yintercept = 0, slope = 1, color= "red") +
#   scale_x_discrete(labels = centers[, order(total_skills)],
#                    breaks = centers[, order(total_skills)], drop = F)+
#   scale_y_discrete(labels = centers[, order(total_skills)],
#                    breaks = centers[, order(total_skills)], drop = F)+
#   theme(axis.text.x = element_text(angle = 90))
#
#   print(gg)
#
# }
```

See how well each scheme predicts log earnings

```
aicr <- function(c.year, data = acs){
  mod <- lm(log_incwage ~ as.factor(kmeans_cluster_67_current), data[year == c.year])
  mod2 <- lm(log_incwage ~ as.factor(microocc_current), data[year == c.year])
  mod3 <- lm(log_incwage ~ as.factor(kmeans_cluster_9_current), data[year == c.year])
  mod4<- lm(log_incwage ~ as.factor(mesoocc_current), data[year == c.year])
  return(data.table(year = c.year, cluster_67 = summary(mod)$r.squared,
             micro =  summary(mod2)$r.squared,
             cluster_9 = summary(mod3)$r.squared,
             meso = summary(mod4)$r.squared
             ))
}

lapply( unique(acs$year), aicr) %>% rbindlist() -> out
ggplot(melt(out, id.var = 'year'))+
  geom_line(aes(x = year, y = value, color = variable))
```


0.24
0.22
0.20
0.18
0.16
0.14
value
1990
2000
2010
2020
year
variable
cluster_67
micro
cluster_9
meso