00_NLSY79_explore

Hunter York

6/14/2021

```
# Remove the '#' before the following lines to rename variables using Qnames instead of Reference Numbe
data <- qnames(data)
setnames(data, gsub("-", "_", names(data), fixed = T))
setnames(data, gsub("~", "_", names(data), fixed = T))

# create dataset of reasons left each job to use later
data[,.SD, .SDcols = names(data)[names(data) %like% "EMPLOYERS_ALL_WHYLEFT_MOST_RECENT|CASEID"]] %>%
  melt(id.vars = "CASEID_1979") -> reasons_quit

data <- data[,.SD, .SDcols = names(data)[names(data) %like% "CASEID|RACE|SEX|SAMPLE_ID|OCCALL|INDALL|HR

rubric <- expand.grid(c.job = 1:5,
                      c.year = c(1980:1992, seq(1994,2016,2))) %>%
  data.table(., stringsAsFactors = F)

for(c.job in 1:5){
  for(c.year in  c(1980:1992, seq(1994,2016,2))) {
    data[,paste0("start_", c.job,"_", c.year) := get(paste0("START_WK#_", c.year, "_JOB#0", c.job, "_XR
    data[,paste0("stop_", c.job,"_", c.year) := get(paste0("STOP_WK#_", c.year, "_JOB#0", c.job, "_XRND
    data[,paste0("hrp_", c.job,"_", c.year) := get(paste0("HRP", c.job, "_", c.year))]
    data[,paste0("occ_", c.job,"_", c.year) := get(paste0("OCCALL_EMP.0", c.job, "_", c.year))]
    data[,paste0("ind_", c.job,"_", c.year) := get(paste0("INDALL_EMP.0", c.job, "_", c.year))]
    #
    data[,paste0("reason_left_", c.job,"_", c.year) := get(paste0("QES_23A.0", c.job, "_", c.year))]
    # data[,paste0("task_shrt_rpt_", c.job,"_", c.year) := get(paste0("QES_PDIIA.0", c.job, "_000001",
    # data[,paste0("task_phsyical_", c.job,"_", c.year) := get(paste0("QES_PDIIA.0", c.job, "_000002",
    # data[,paste0("task_superv_", c.job,"_", c.year) := get(paste0("QES_PDIIA.0", c.job, "_000003", c.
    # data[,paste0("task_prob_solv_", c.job,"_", c.year) := get(paste0("QES_PDIIB.0", c.job, "_", c.yea
    # data[,paste0("task_math_prob_solv_", c.job,"_", c.year) := get(paste0("QES_PDIIC.0", c.job, "_",
    # data[,paste0("task_longest_doc_read_", c.job,"_", c.year) := get(paste0("QES_PDIID.0", c.job, "_"
    # data[,paste0("task_freq_pers_contact_others_", c.job,"_", c.year) := get(paste0("QES_PDIIE.0", c.
    # data[,paste0("task_freq_pers_contact_custs_", c.job,"_", c.year) := get(paste0("QES_PDIIF.0", c.j
    # data[,paste0("job_stress_past_year", c.job,"_", c.year) := get(paste0("QES_JSWD1.0", c.job, "_",
    # data[,paste0("job_effect_emot_mental_health", c.job,"_", c.year) := get(paste0("QES_JSWD2.0", c.j
    # data[,paste0("job_effect_phys_health", c.job,"_", c.year) := get(paste0("QES_JSWD3.0", c.job, "_"
    # data[,paste0("can_work_shorter_hours", c.job,"_", c.year) := get(paste0("QES_JSWD4A.0", c.job, "_
    # data[,paste0("can_work_longer_hours", c.job,"_", c.year) := get(paste0("QES_JSWD4B.0", c.job, "_"
    # data[,paste0("can_work_more_flex_hours", c.job,"_", c.year) := get(paste0("QES_JSWD4C.0", c.job,


    if(c.job == 1 & c.year <= 1993){
      data[is.na(get(paste0("occ_", c.job,"_", c.year))),
           paste0("occ_", c.job,"_", c.year) := get(paste0("CPSOCC70_", c.year))]
      data[is.na(get(paste0("ind_", c.job,"_", c.year))),
           paste0("ind_", c.job,"_", c.year) := get(paste0("CPSIND70_", c.year))]
    }
  }
}

# loop over all employers ids and put in same format
data_names <- names(data)
for (c.job in 1:65){
  for(c.year in  c(1980:1992, seq(1994,2016,2))) {
    if (paste0("EMPLOYERS_ALL_ID_", c.year, ".", formatC(c.job, flag = "0", width = 2), "_XRND") %in% da
      data[get(paste0("EMPLOYERS_ALL_ID_", c.year, ".", formatC(c.job, flag = "0", width = 2), "_XRND")
      data[get(paste0("EMPLOYERS_ALL_ID_", c.year, ".", formatC(c.job, flag = "0", width = 2), "_XRND")
      data[get(paste0("EMPLOYERS_ALL_ID_", c.year, ".", formatC(c.job, flag = "0", width = 2), "_XRND")
      data[get(paste0("EMPLOYERS_ALL_ID_", c.year, ".", formatC(c.job, flag = "0", width = 2), "_XRND")
      data[get(paste0("EMPLOYERS_ALL_ID_", c.year, ".", formatC(c.job, flag = "0", width = 2), "_XRND")
    }
  }
}

data <- data[,.SD, .SDcols = sort(names(data))]

melt(data,
     id.vars = c("CASEID_1979",
                 "SAMPLE_RACE_78SCRN",
                 "SAMPLE_SEX_1979",
                 "SAMPLE_ID_1979",
                 "HGC_EVER_XRND"),
     measure.vars = patterns(start_week = "start_",
                             stop_week = "stop_",
                             occ = "occ_",
                             ind = "ind_",
                             hourly_pay = "hrp_",
                             reason_left = "reason_left_",
                             emp_id = "emp_id_")) -> data_long

# merge on rubric
rubric <- rubric[order(c.job, c.year)]
rubric[, variable := 1:nrow(rubric)]
data_long[, variable := as.numeric(variable)]
data_long <- merge(data_long, rubric, by = "variable")

## only include 1996 onwards :(
## https://www.nlsinfo.org/content/cohorts/nlsy79/other-documentation/codebook-supplement/nlsy79-append

data_long <- data_long[!is.na(occ)]


# now merge on reasons for quitting jobs from earlier
library(stringr)
reasons_quit[, emp_id := as.numeric(str_sub(variable, -7, -6))]
reasons_quit[, reason_quit_recent := value]
```

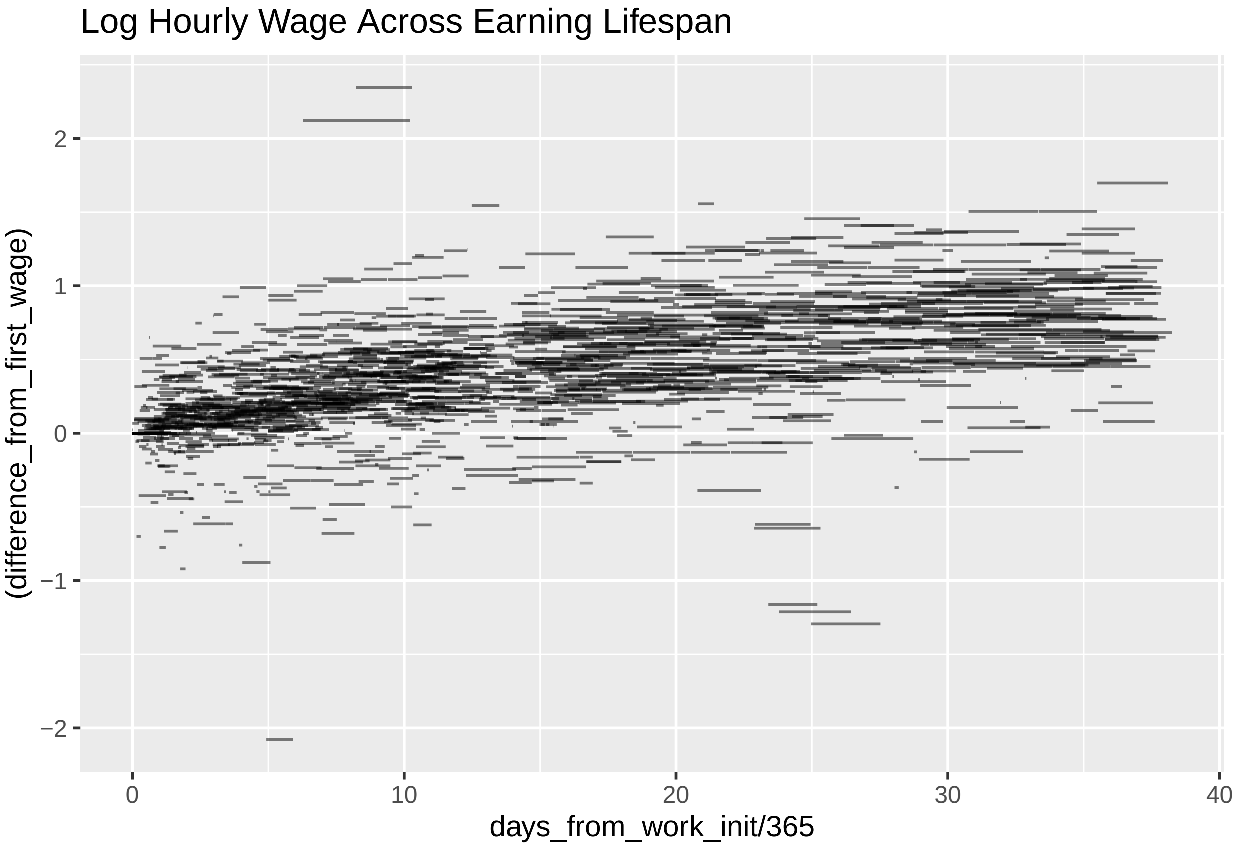
```
data_long <- merge(data_long, reasons_quit[,.(CASEID_1979, reason_quit_recent, emp_id)],
                   by = c("CASEID_1979", "emp_id"), all.x = T)
```

```
data_long[, start_day := (start_week* 7)]
data_long[, stop_day := (stop_week* 7)]
data_long[, start_date := as.Date(start_day, origin = as.Date("01-01-1978", format = "%m-%d-%Y"))]
data_long[, stop_date := as.Date(stop_day, origin = as.Date("01-01-1978", format = "%m-%d-%Y"))]
data_long <- data_long[!is.na(occ)]
```

```
data_long[, days_from_work_init := start_date - min(start_date), by = CASEID_1979]
data_long[, hourly_pay_dollars := hourly_pay/100]
data_long[, initial_pay := min(hourly_pay_dollars[days_from_work_init == 0]), by = CASEID_1979]
data_long[, difference_from_first_wage := log10(hourly_pay_dollars) - log10(initial_pay)]
data_long[, stop_days_from_work_init := stop_date - min(start_date), by = CASEID_1979]
data_long[, employment_midpoint := (days_from_work_init + stop_days_from_work_init)/2]

ggplot(data_long[CASEID_1979 %in% sample(unique(data_long$CASEID_1979), 100)]) +
  geom_segment(aes(x = days_from_work_init/365,
                   xend = stop_days_from_work_init/365,
                   y = (difference_from_first_wage),
                   yend = (difference_from_first_wage),
                   group = CASEID_1979), alpha = .5) +
  ggtitle("Log Hourly Wage Across Earning Lifespan")
```

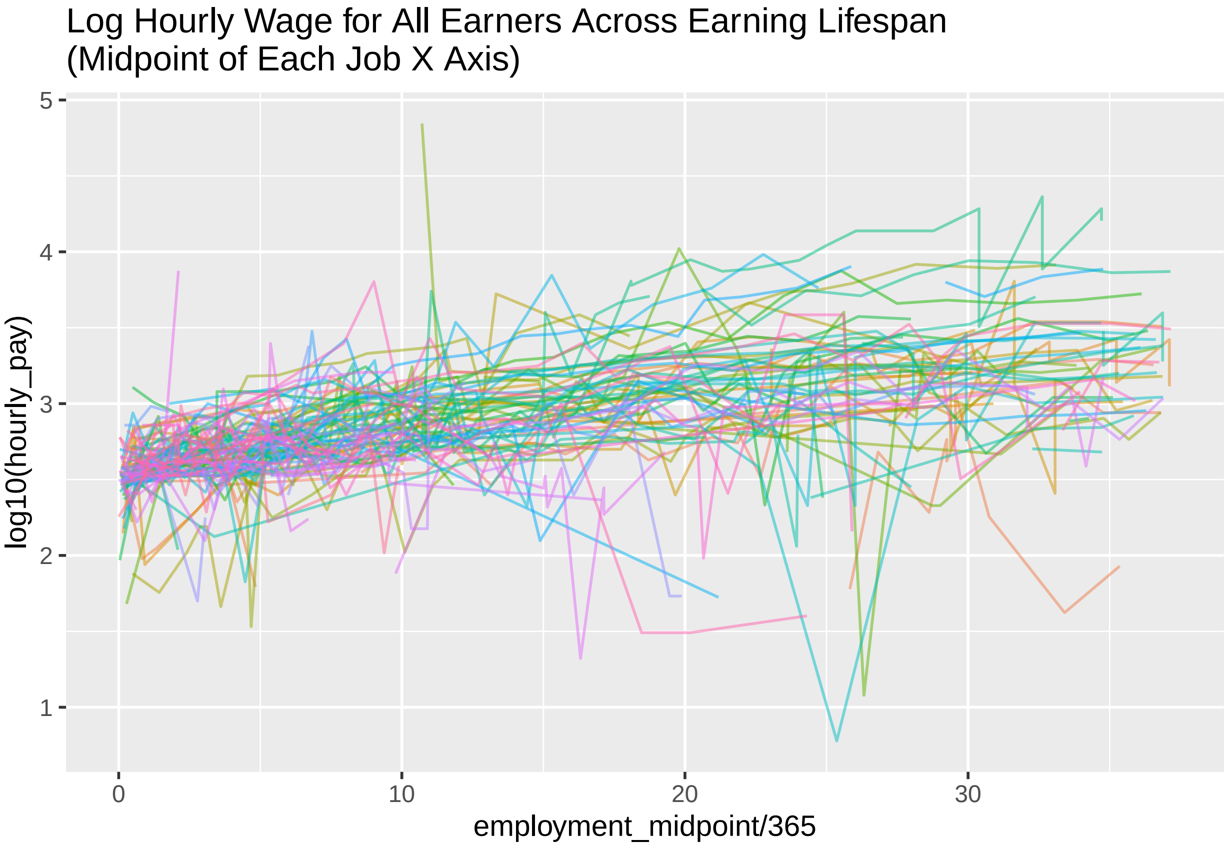
```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Removed 281 rows containing missing values (geom_segment).
```



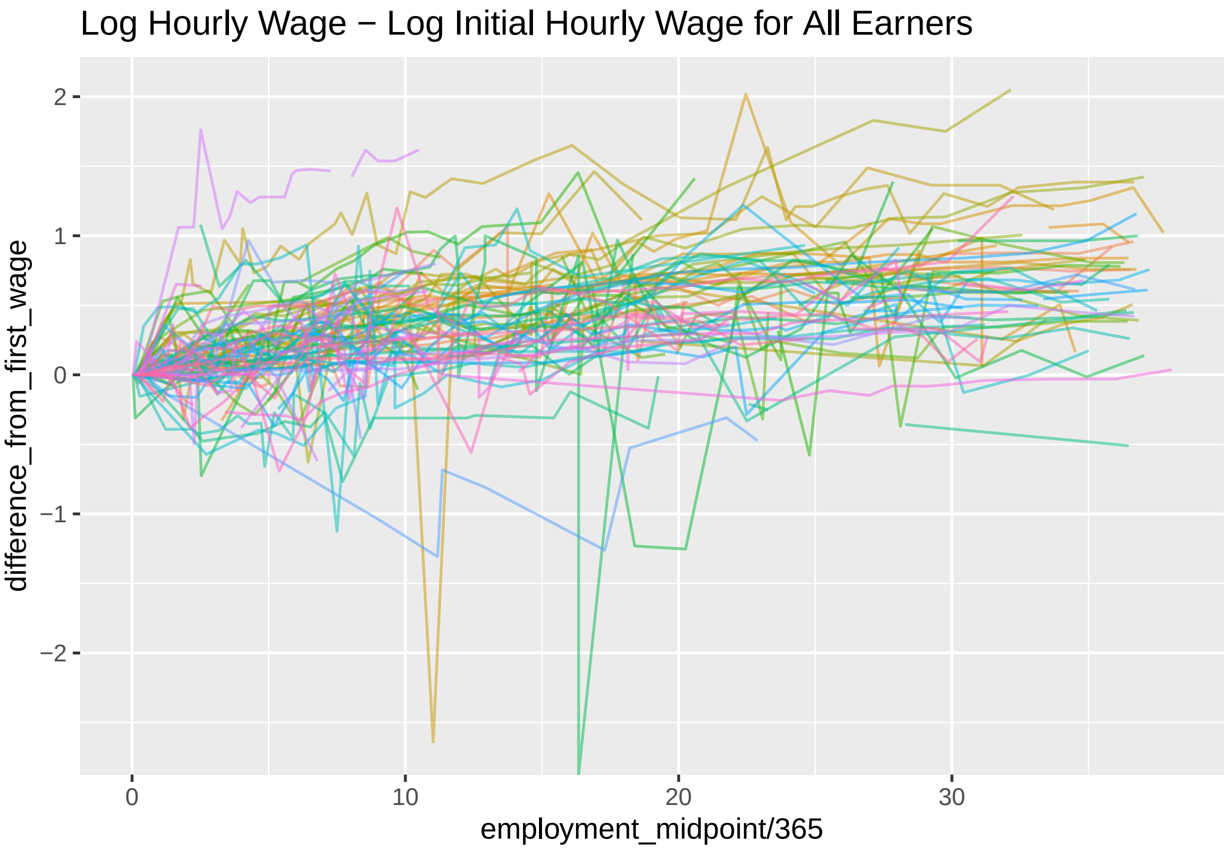
```
ggplot(data_long[CASEID_1979 %in% sample(unique(data_long$CASEID_1979), 100)]) +
  geom_line(aes(x = employment_midpoint/365, y = log10(hourly_pay),
                group = CASEID_1979, color = as.factor(CASEID_1979)), alpha = .5, show.legend = F) +
  ggtitle("Log Hourly Wage for All Earners Across Earning Lifespan\n(Midpoint of Each Job X Axis)")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.

## Warning: Removed 245 row(s) containing missing values (geom_path).
```



```
data_long%>%
  .[CASEID_1979 %in% sample(unique(CASEID_1979), 100)] %>%
  ggplot() +
  geom_line(aes(x = employment_midpoint/365, y = difference_from_first_wage,
                group = CASEID_1979, color = as.factor(CASEID_1979)), alpha = .5, show.legend = F)+
  ggtitle("Log Hourly Wage - Log Initial Hourly Wage for All Earners")
```

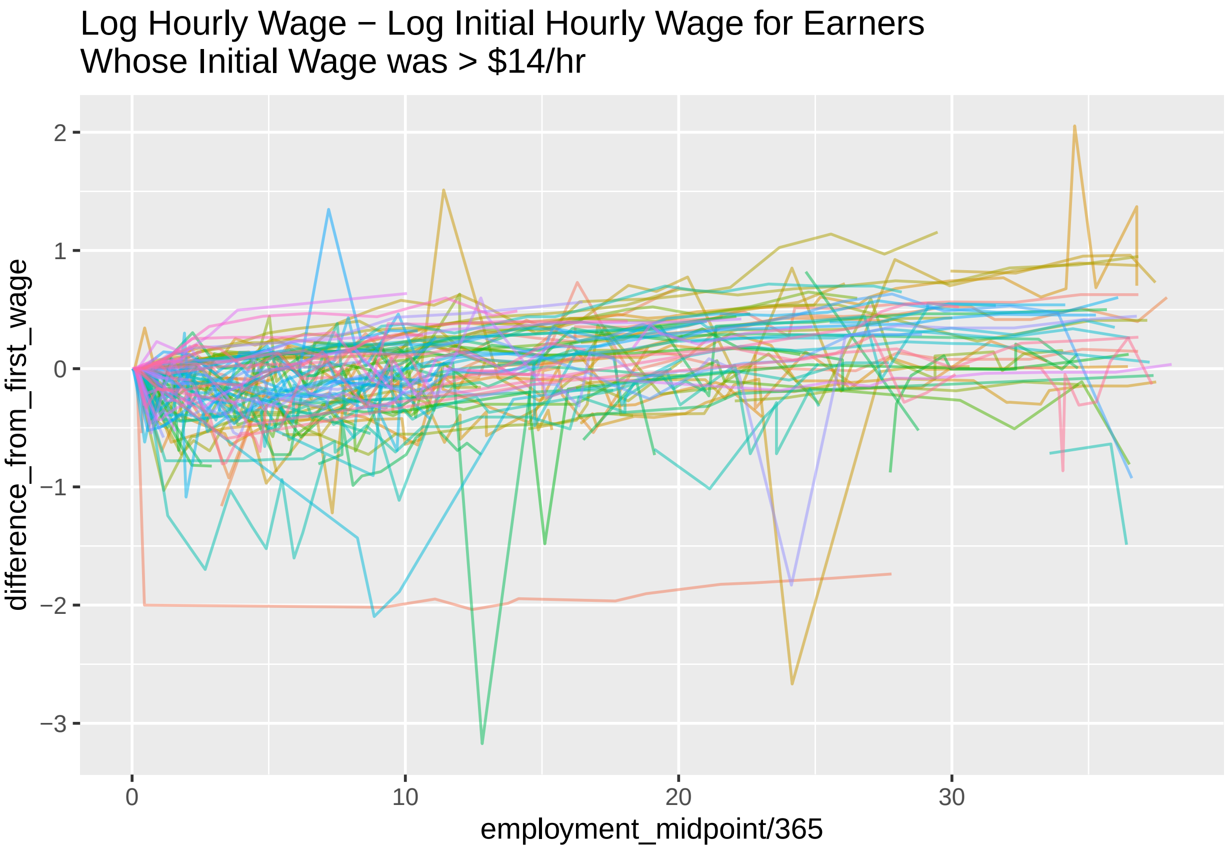
```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.

## Warning: Removed 275 row(s) containing missing values (geom_path).
```



```
data_long[initial_pay > 8]%>%
  .[CASEID_1979 %in% sample(unique(CASEID_1979), 100)] %>%
  ggplot() +
  geom_line(aes(x = employment_midpoint/365, y = difference_from_first_wage,
                group = CASEID_1979, color = as.factor(CASEID_1979)), alpha = .5, show.legend = F) +
  ggtitle("Log Hourly Wage - Log Initial Hourly Wage for Earners\nWhose Initial Wage was > $14/hr")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Removed 14 row(s) containing missing values (geom_path).
```



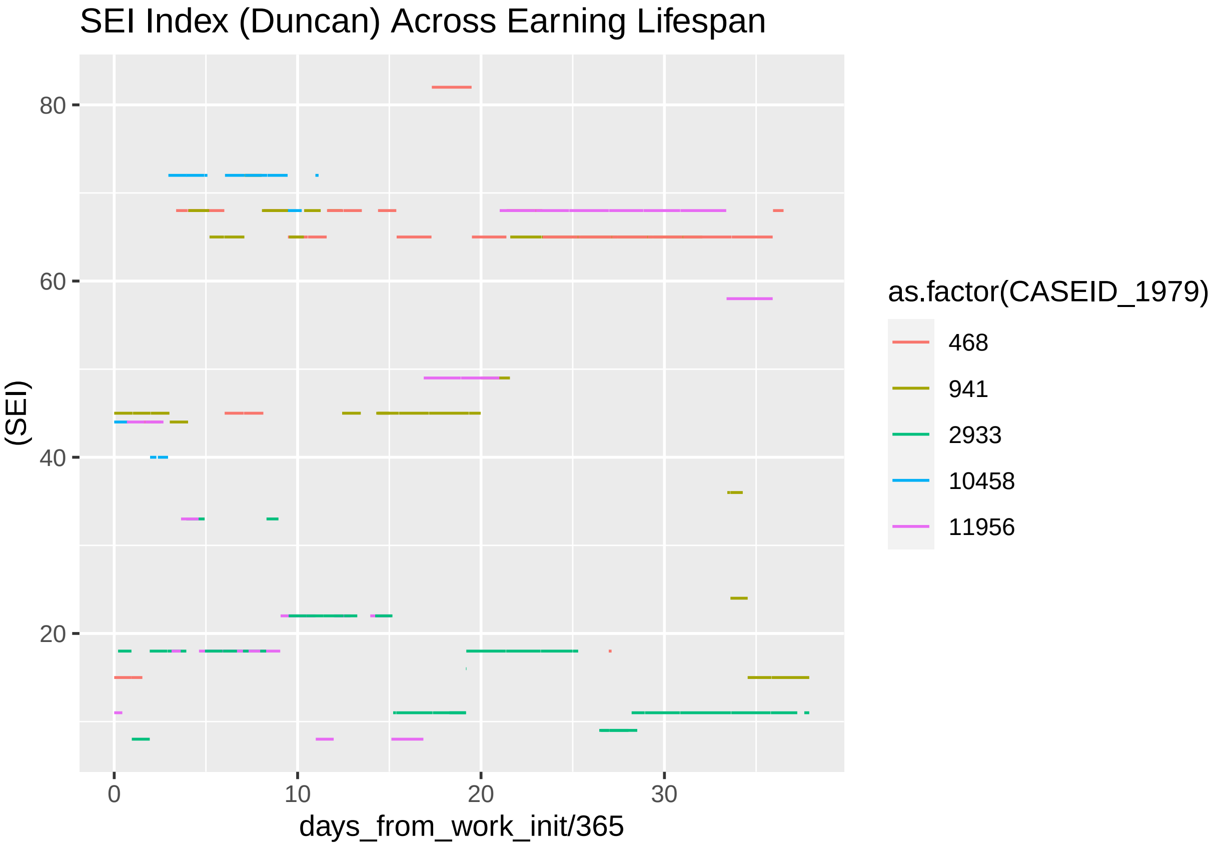
see how prestige tracks across time

```
data_long[nchar(occ) == 4, occ := as.numeric(substr(occ, 1,3))]


# load xwalk for occ1970
occ_1970 <- fread("../ref/usa_00013.csv")
occ_1970[,.(OCC, OCC1990, OCCSCORE, SEI, HWSEI, PRESGL, PRENT, ERSCOR90, EDSCOR90, NPBOSS90)] %>%
  unique() -> occ_1970
occ_1970 <- occ_1970[!(OCC == 0 & OCC1990 == 905)]

# load xwalk for occ2005
occ_2000 <- fread("../ref/usa_00015.csv")
occ_2000[,.(OCC, OCCSCORE,OCC1990, SEI, HWSEI, PRESGL, PRENT, ERSCOR90, EDSCOR90, NPBOSS90)] %>%
  unique() -> occ_2000
occ_2000 <- occ_2000[!(OCC == 0)]
occ_2000 <- occ_2000[!duplicated(OCC)]

data_long_1 <- merge(data_long[c.year <2002], occ_1970, by.x = "occ", by.y = "OCC" , all.x = T)

data_long_2 <- merge(data_long[c.year >=2002], occ_2000, by.x = "occ", by.y = "OCC" , all.x = T)

data_long <- rbind(data_long_1, data_long_2, fill = T)
```

```
data_long[occ != 0]%>%
  .[CASEID_1979 %in% sample(unique(CASEID_1979), 5)] %>%
ggplot() +
  geom_segment(aes(x = days_from_work_init/365,
                   xend = stop_days_from_work_init/365,
                   y = (SEI),
                   yend = (SEI),
                   group = CASEID_1979,
                   color = as.factor(CASEID_1979)), alpha = 1) +
  ggtitle("SEI Index (Duncan) Across Earning Lifespan")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

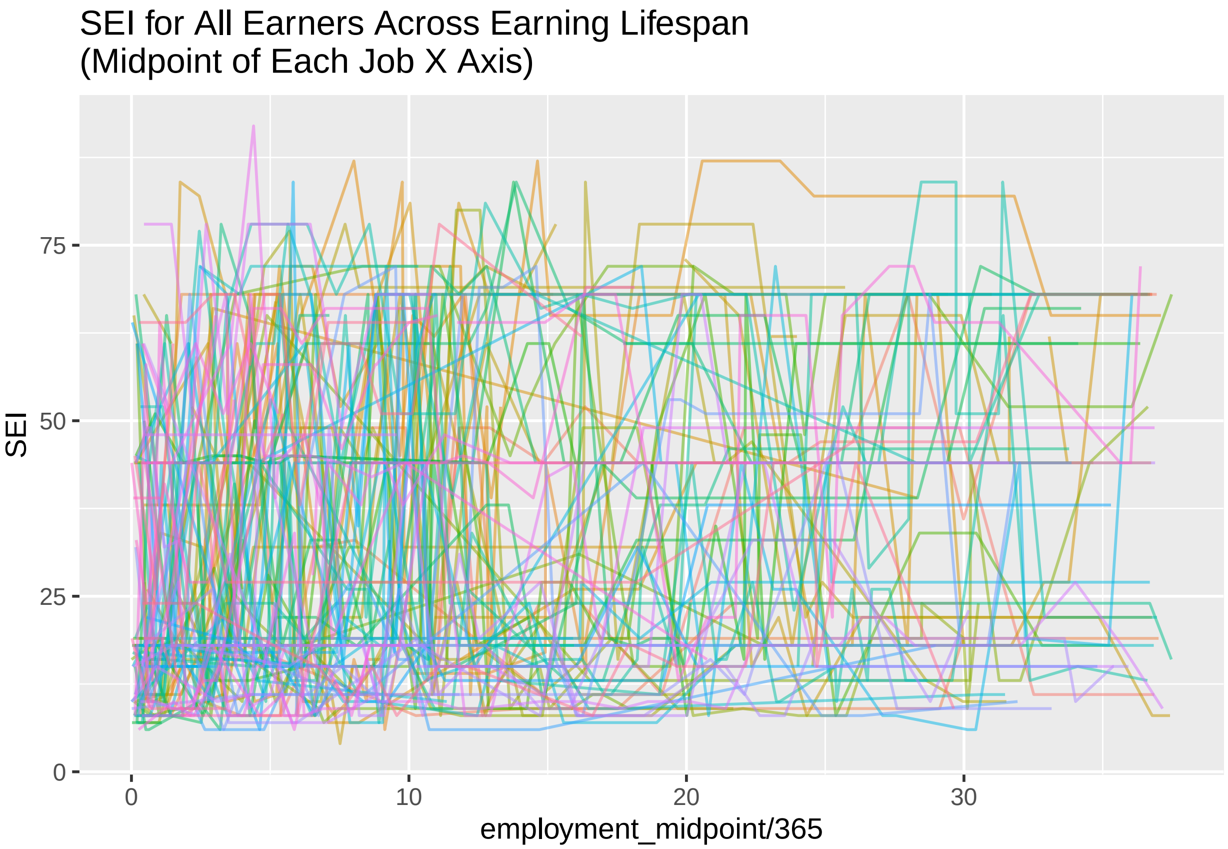
```
## Warning: Removed 5 rows containing missing values (geom_segment).
```



```
data_long[, sei_delta := SEI - SEI[days_from_work_init == 0][1], by = .(CASEID_1979)]
data_long[, presgl_delta := PRESGL - PRESGL[days_from_work_init == 0][1], by = .(CASEID_1979)]


data_long[occ != 0]%>%
  .[CASEID_1979 %in% sample(unique(CASEID_1979), 100)] %>%
  ggplot() +
  geom_line(aes(x = employment_midpoint/365, y = SEI,
                group = CASEID_1979, color = as.factor(CASEID_1979)), alpha = .5, show.legend = F) +
  ggtitle("SEI for All Earners Across Earning Lifespan\n(Midpoint of Each Job X Axis)")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

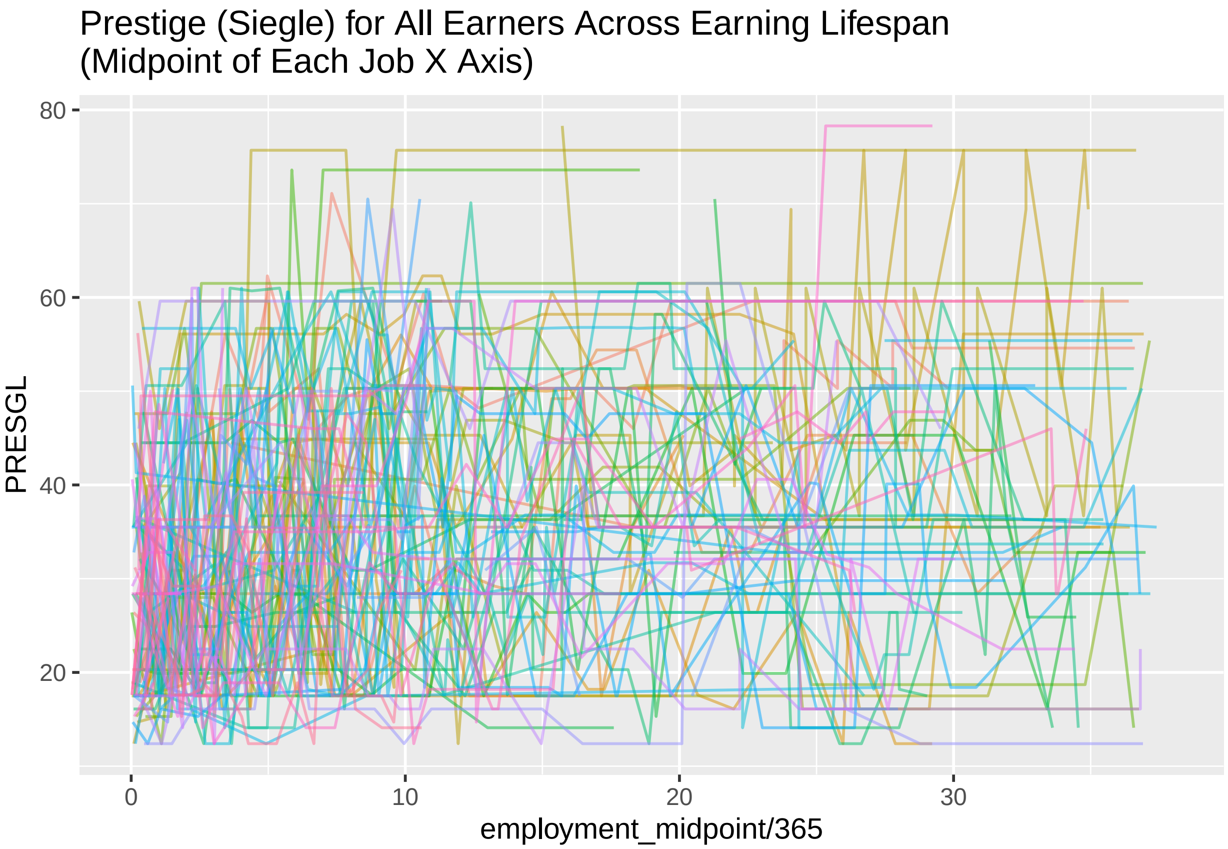
```
## Warning: Removed 27 row(s) containing missing values (geom_path).
```



```r
data_long[occ != 0]%>%
  .[CASEID_1979 %in% sample(unique(CASEID_1979), 100)] %>%
  ggplot() +
  geom_line(aes(x = employment_midpoint/365, y = PRESGL,
                group = CASEID_1979, color = as.factor(CASEID_1979)), alpha = .5, show.legend = F) +
  ggtitle("Prestige (Siegle) for All Earners Across Earning Lifespan\n(Midpoint of Each Job X Axis)")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

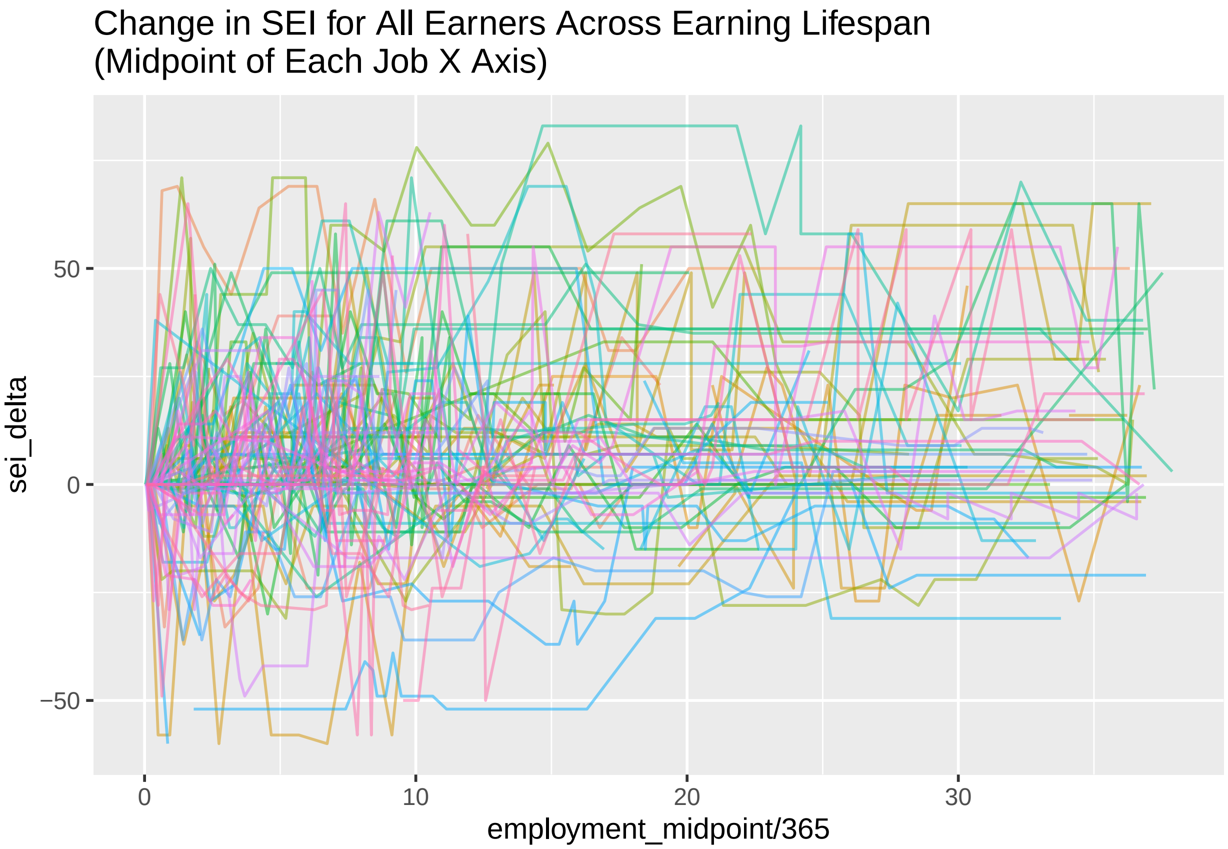
```
## Warning: Removed 98 row(s) containing missing values (geom_path).
```



```
data_long[occ != 0]%>%
  .[CASEID_1979 %in% sample(unique(CASEID_1979), 100)] %>%
  ggplot() +
  geom_line(aes(x = employment_midpoint/365, y = sei_delta,
                group = CASEID_1979, color = as.factor(CASEID_1979)), alpha = .5, show.legend = F) +
  ggtitle("Change in SEI for All Earners Across Earning Lifespan\n(Midpoint of Each Job X Axis)")
```

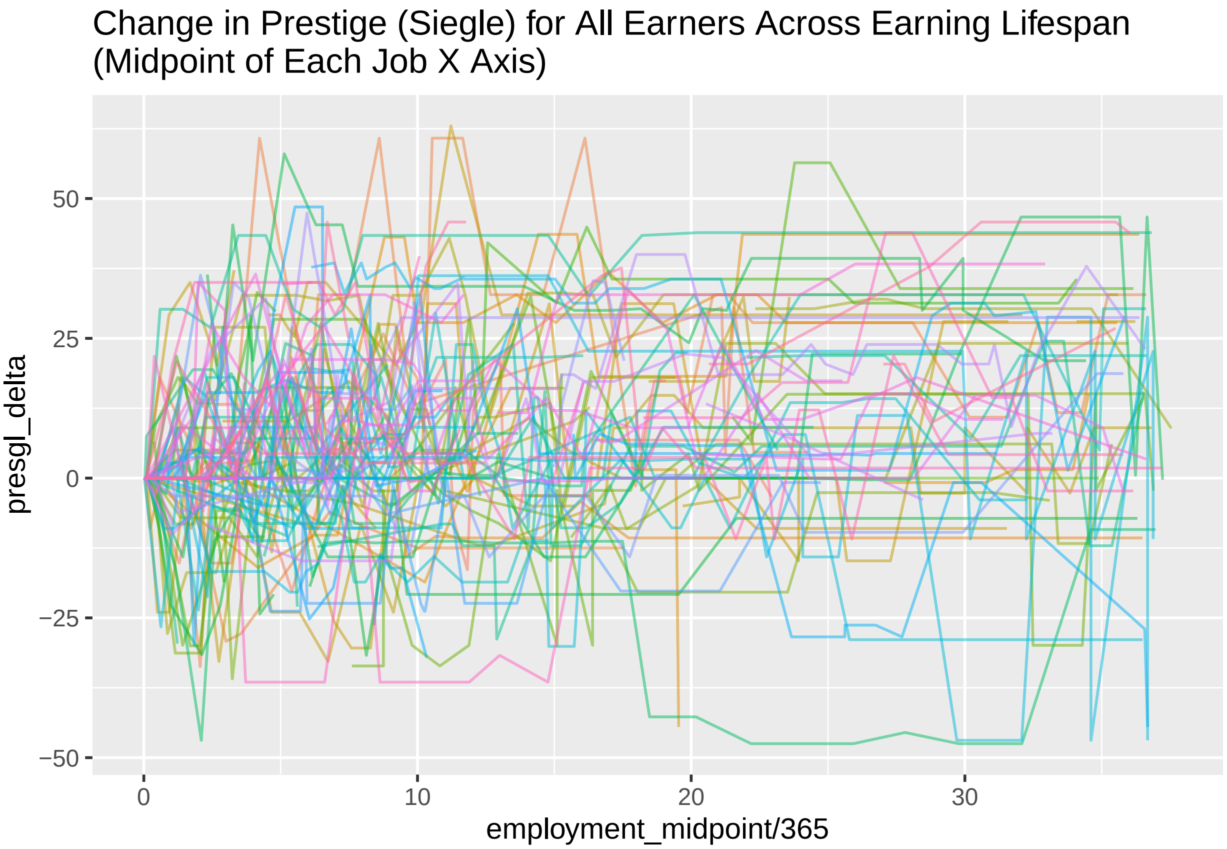
```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Removed 221 row(s) containing missing values (geom_path).
```



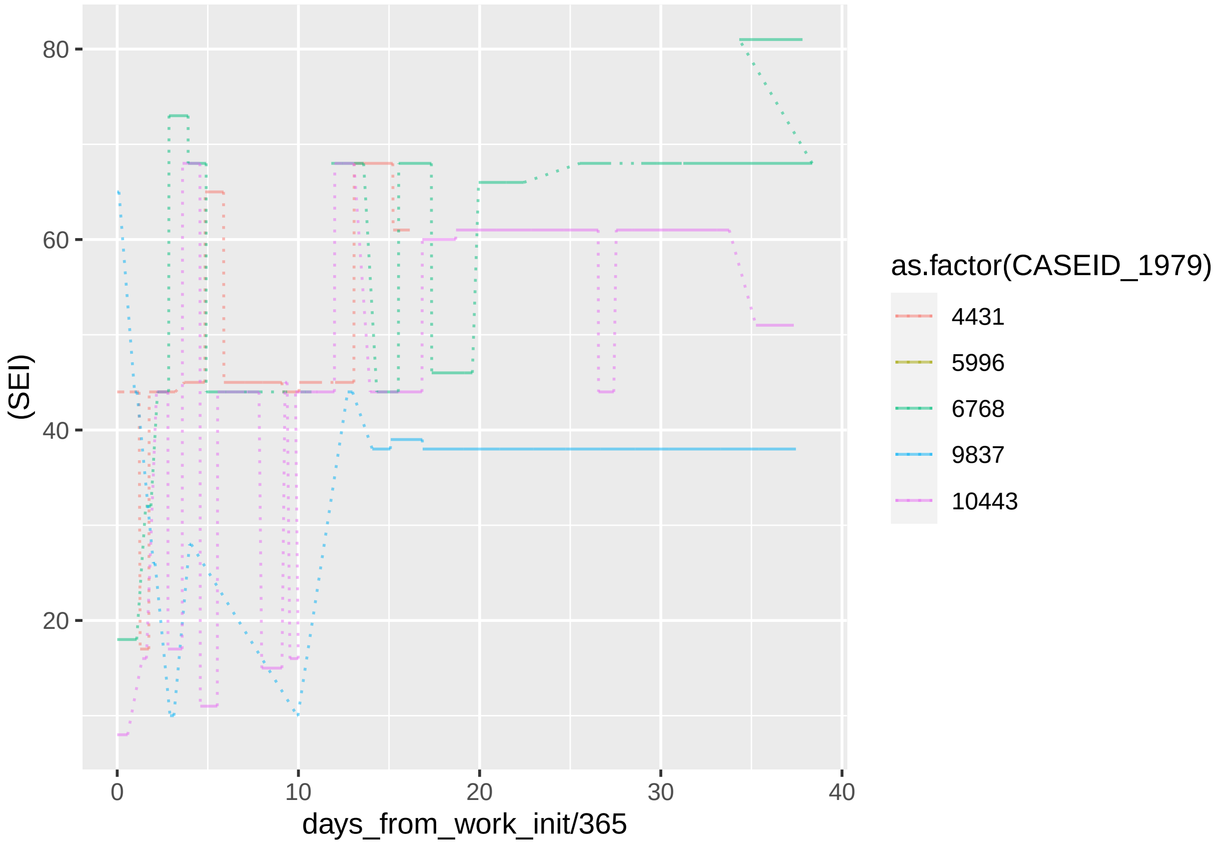
```
data_long[occ != 0]%>%
  .[CASEID_1979 %in% sample(unique(CASEID_1979), 100)] %>%
  ggplot() +
  geom_line(aes(x = employment_midpoint/365, y = presgl_delta,
                group = CASEID_1979, color = as.factor(CASEID_1979)), alpha = .5, show.legend = F) +
  ggtitle("Change in Prestige (Siegle) for All Earners Across Earning Lifespan\n(Midpoint of Each Job X
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.

## Warning: Removed 339 row(s) containing missing values (geom_path).
```



```
data_long <- data_long[order(CASEID_1979, start_date)]
data_long[, lag_SEI := lag(SEI), by = CASEID_1979]
data_long[, lag_PRESGL := lag(PRESGL), by = CASEID_1979]
data_long[, lag_hourly_pay_dollars := lag(hourly_pay_dollars), by = CASEID_1979]

data_long[occ != 0]%>%
  .[CASEID_1979 %in% sample(unique(CASEID_1979), 5)] %>%
  ggplot() +
  geom_segment(aes(x = days_from_work_init/365,
                   xend = stop_days_from_work_init/365,
                   y = (SEI),
                   yend = (SEI),
                   group = CASEID_1979,
                   color = as.factor(CASEID_1979)), alpha = .5) +
  geom_segment(aes(x = lag(stop_days_from_work_init)/365,
                   xend = days_from_work_init/365,
                   y = lag_SEI,
                   yend = (SEI),
                   group = CASEID_1979,
                   color = as.factor(CASEID_1979)), linetype = "dotted", alpha = .5)
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.

## Warning: Removed 11 rows containing missing values (geom_segment).

## Warning: Removed 17 rows containing missing values (geom_segment).
```



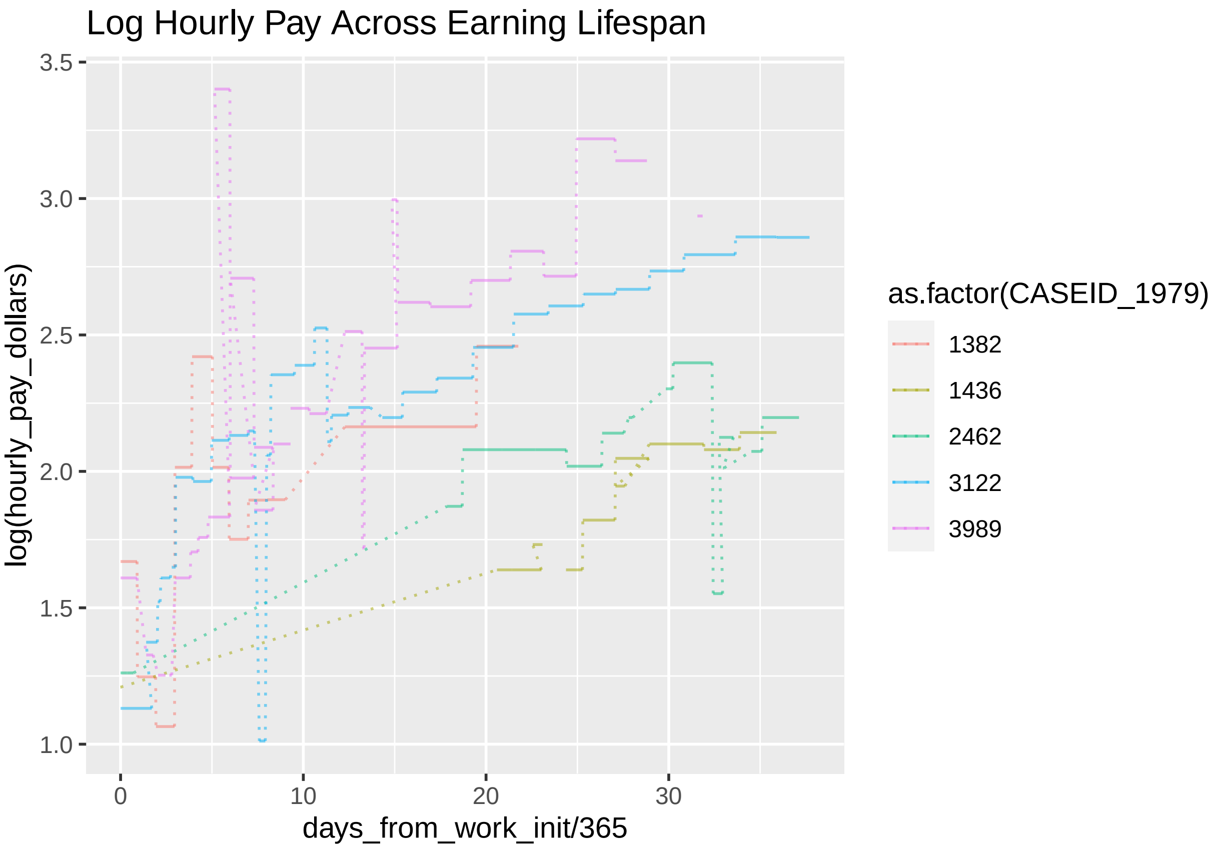
```
ggtitle("SEI Index (Duncan) Across Earning Lifespan")
```

```
## $title
## [1] "SEI Index (Duncan) Across Earning Lifespan"
##
## attr(,"class")
## [1] "labels"
```

```
data_long[occ != 0]%>%
  .[CASEID_1979 %in% sample(unique(CASEID_1979), 5)] %>%
  ggplot() +
  geom_segment(aes(x = days_from_work_init/365,
                   xend = stop_days_from_work_init/365,
                   y = log(hourly_pay_dollars),
                   yend = log(hourly_pay_dollars),
                   group = CASEID_1979,
                   color = as.factor(CASEID_1979)), alpha = .5) +
  geom_segment(aes(x = lag(stop_days_from_work_init)/365,
                   xend = days_from_work_init/365,
                   y = log(lag_hourly_pay_dollars),
                   yend = log(hourly_pay_dollars),
                   group = CASEID_1979,
                   color = as.factor(CASEID_1979)), linetype = "dotted", alpha = .5)+
  ggtitle("Log Hourly Pay Across Earning Lifespan")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Removed 3 rows containing missing values (geom_segment).
## Warning: Removed 10 rows containing missing values (geom_segment).
```

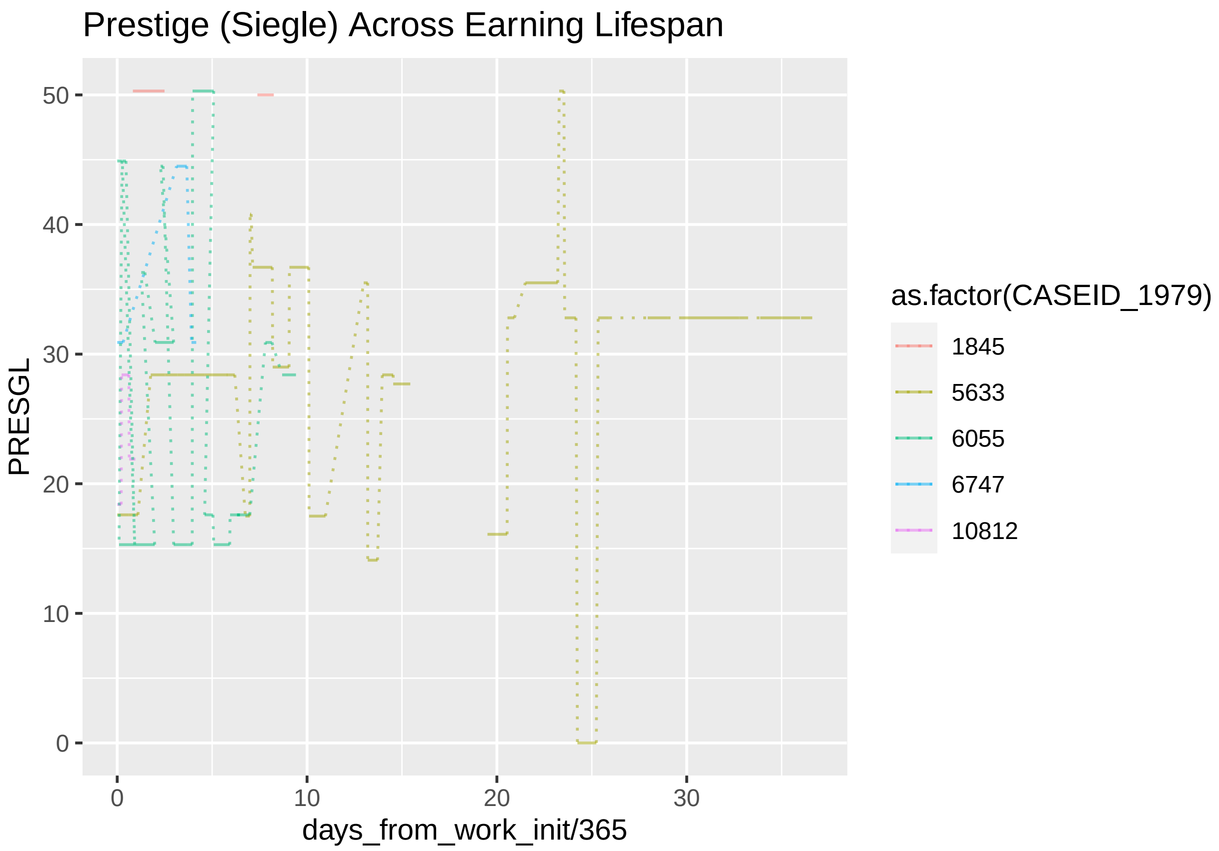


```r
data_long[occ != 0]%>%
  .[CASEID_1979 %in% sample(unique(CASEID_1979), 5)] %>%
  ggplot() +
  geom_segment(aes(x = days_from_work_init/365,
                   xend = stop_days_from_work_init/365,
                   y = PRESGL,
                   yend = PRESGL,
                   group = CASEID_1979,
                   color = as.factor(CASEID_1979)), alpha = .5) +
  geom_segment(aes(x = lag(stop_days_from_work_init)/365,
                   xend = days_from_work_init/365,
                   y = lag_PRESGL,
                   yend = PRESGL,
                   group = CASEID_1979,
                   color = as.factor(CASEID_1979)), linetype = "dotted", alpha = .5)+
  ggtitle("Prestige (Siegle) Across Earning Lifespan")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```

```
## Warning: Removed 4 rows containing missing values (geom_segment).
```

```
## Warning: Removed 11 rows containing missing values (geom_segment).
```



clean up even more so that occupations match if tenure is constant

```
data_long <- data_long[order(CASEID_1979, start_week, occ, emp_id)]
data_long[, date_stop_previous := lag(stop_week), by = .(CASEID_1979, emp_id)]
data_long[, date_start_next := lead(start_week), by = .(CASEID_1979, emp_id)]

data_long[, cont_emp_from_prev := ifelse(abs(date_stop_previous - start_week) < 2, 1, 0)]
data_long[, cont_emp_to_next := ifelse(abs(date_start_next - stop_week) < 2, 1, 0)]
data_long[ stop_week == max(stop_week), cont_emp_to_next  := 1, by = .(CASEID_1979, emp_id)]

data_long[, run_length_id := rleid(cont_emp_to_next), by = .(emp_id, CASEID_1979)]

Mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}



job_sum <- data_long[,.(start_week = min(start_week),
                        stop_week = max(stop_week),
                        start_date = min(start_date),
                        stop_date = max(stop_date),
                        occ = Mode(occ),
                        occ_1990 = Mode(OCC1990),
                        ind = Mode(ind),
```

```r
                        c.year = Mode(c.year),
                        low_wage = min(hourly_pay_dollars),
                        high_wage = max(hourly_pay_dollars),
                        last_wage = hourly_pay_dollars[start_week == max(start_week)],
                        reason_left = Mode(reason_quit_recent),
                        race = Mode(SAMPLE_RACE_78SCRN),
                        sex = Mode(SAMPLE_SEX_1979),
                        job_id = Mode(c.job),
                        highest_grade = Mode(HGC_EVER_XRND),
                        sample_id = Mode(SAMPLE_ID_1979)), by = .(CASEID_1979, emp_id,run_length_id)]

# merge on skills


occ_1990_2010_xwalk <- fread("../ref/usa_00017.csv")
occ_1990_2010_xwalk[,.(OCC, OCC2010)] %>% unique() -> occ_1990_2010_xwalk
setnames(occ_1990_2010_xwalk, "OCC", "OCC1990")

occ_2010_soc_xwalk <- fread("../ref/usa_00018.csv")
occ_2010_soc_xwalk[,.(OCC, OCCSOC)] %>% unique() -> occ_2010_soc_xwalk
setnames(occ_2010_soc_xwalk, "OCC", "OCC2010")

occ_1990_soc_xwalk <- merge(occ_1990_2010_xwalk, occ_2010_soc_xwalk, by = "OCC2010")

########################


library(haven)
library(readxl)

# load the data

# now create skills dataset
skills_2009 <- read.delim('/Users/hyork/Documents/projects/occupation/ref/db_14_0 2009.7/Skills.txt') %
skills_2013 <-  read.delim('/Users/hyork/Documents/projects/occupation/ref/db_18_0_2013.7/Skills.txt') %
skills_2018 <- read_excel("/Users/hyork/Documents/projects/occupation/ref/db_22_2_excel 2018.2/Skills.x
skills_2009[, year := 2009]
skills_2013[, year := 2013]
skills_2018[, year := 2018]
setnames(skills_2018, names(skills_2018), gsub(" ", ".", names(skills_2018), fixed = T))
skills_2018[, `O.NET.SOC.Code` := `O*NET-SOC.Code`]
skills1 <- rbindlist(list(skills_2009, skills_2013, skills_2018), fill = T)
skills1 <- skills1[, .(O.NET.SOC.Code, Element.Name, Scale.ID, Data.Value, Standard.Error, year)]
skills1[, Element.Name := paste0("skl_", Element.Name)]
# add abilities
abilities_2009 <- read.delim('/Users/hyork/Documents/projects/occupation/ref/db_14_0 2009.7/Abilities.t
abilities_2013 <-  read.delim('/Users/hyork/Documents/projects/occupation/ref/db_18_0_2013.7/Abilities.
abilities_2018 <- read_excel("/Users/hyork/Documents/projects/occupation/ref/db_22_2_excel 2018.2/Abilit
abilities_2009[, year := 2009]
abilities_2013[, year := 2013]
abilities_2018[, year := 2018]
setnames(abilities_2018, names(abilities_2018), gsub(" ", ".", names(abilities_2018), fixed = T))
abilities_2018[, `O.NET.SOC.Code` := `O*NET-SOC.Code`]
```

```
abilities <- rbindlist(list(abilities_2009, abilities_2013, abilities_2018), fill = T)
abilities <- abilities[, .(O.NET.SOC.Code, Element.Name, Scale.ID, Data.Value, Standard.Error, year)]
abilities[, Element.Name := paste0("abl_", Element.Name)]

# add knowledge
knowledge_2009 <- read.delim('/Users/hyork/Documents/projects/occupation/ref/db_14_0 2009.7/Knowledge.t
knowledge_2013 <-  read.delim('/Users/hyork/Documents/projects/occupation/ref/db_18_0_2013.7/Knowledge.
knowledge_2018 <- read_excel("/Users/hyork/Documents/projects/occupation/ref/db_22_2_excel 2018.2/Knowl
knowledge_2009[, year := 2009]
knowledge_2013[, year := 2013]
knowledge_2018[, year := 2018]
setnames(knowledge_2018, names(knowledge_2018), gsub(" ", ".", names(knowledge_2018), fixed = T))
knowledge_2018[, `O.NET.SOC.Code` := `O*NET-SOC.Code`]
knowledge <- rbindlist(list(knowledge_2009, knowledge_2013, knowledge_2018), fill = T)
knowledge <- knowledge[, .(O.NET.SOC.Code, Element.Name, Scale.ID, Data.Value, Standard.Error, year)]
knowledge[, Element.Name := paste0("knl_", Element.Name)]

# add work activities
workactivities_2009 <- read.delim('/Users/hyork/Documents/projects/occupation/ref/db_14_0 2009.7/Work A
workactivities_2013 <-  read.delim('/Users/hyork/Documents/projects/occupation/ref/db_18_0_2013.7/Work 
workactivities_2018 <- read_excel("/Users/hyork/Documents/projects/occupation/ref/db_22_2_excel 2018.2/
workactivities_2009[, year := 2009]
workactivities_2013[, year := 2013]
workactivities_2018[, year := 2018]
setnames(workactivities_2018, names(workactivities_2018), gsub(" ", ".", names(workactivities_2018), fi
workactivities_2018[, `O.NET.SOC.Code` := `O*NET-SOC.Code`]
workactivities <- rbindlist(list(workactivities_2009, workactivities_2013, workactivities_2018), fill =
workactivities <- workactivities[, .(O.NET.SOC.Code, Element.Name, Scale.ID, Data.Value, Standard.Error
workactivities[, Element.Name := paste0("act_", Element.Name)]

# add work styles
workstyles_2009 <- read.delim('/Users/hyork/Documents/projects/occupation/ref/db_14_0 2009.7/Work Style
workstyles_2013 <-  read.delim('/Users/hyork/Documents/projects/occupation/ref/db_18_0_2013.7/Work Styl
workstyles_2018 <- read_excel("/Users/hyork/Documents/projects/occupation/ref/db_22_2_excel 2018.2/Work
workstyles_2009[, year := 2009]
workstyles_2013[, year := 2013]
workstyles_2018[, year := 2018]
setnames(workstyles_2018, names(workstyles_2018), gsub(" ", ".", names(workstyles_2018), fixed = T))
workstyles_2018[, `O.NET.SOC.Code` := `O*NET-SOC.Code`]
workstyles <- rbindlist(list(workstyles_2009, workstyles_2013, workstyles_2018), fill = T)
workstyles <- workstyles[, .(O.NET.SOC.Code, Element.Name, Scale.ID, Data.Value, Standard.Error, year)]
workstyles[, Element.Name := paste0("sty_", Element.Name)]

# add work context
context_2009 <- read.delim('/Users/hyork/Documents/projects/occupation/ref/db_14_0 2009.7/Work Context.
context_2013 <-  read.delim('/Users/hyork/Documents/projects/occupation/ref/db_18_0_2013.7/Work Context
context_2018 <- read_excel("/Users/hyork/Documents/projects/occupation/ref/db_22_2_excel 2018.2/Work Co
context_2009[, year := 2009]
context_2013[, year := 2013]
context_2018[, year := 2018]
setnames(context_2018, names(context_2018), gsub(" ", ".", names(context_2018), fixed = T))
context_2018[, `O.NET.SOC.Code` := `O*NET-SOC.Code`]
context <- rbindlist(list(context_2009, context_2013, context_2018), fill = T)
context <- context[, .(O.NET.SOC.Code, Element.Name, Scale.ID, Data.Value, Standard.Error, year)]
context <- context[Scale.ID %in% c("CX", "CT")]
context[, Element.Name := paste0("ctx_", Element.Name)]

# add work values
values_2009 <- read.delim('/Users/hyork/Documents/projects/occupation/ref/db_14_0 2009.7/Work Values.tx
values_2013 <-  read.delim('/Users/hyork/Documents/projects/occupation/ref/db_18_0_2013.7/Work Values.t
values_2018 <- read_excel("/Users/hyork/Documents/projects/occupation/ref/db_22_2_excel 2018.2/Work Valu
values_2009[, year := 2009]
values_2013[, year := 2013]
values_2018[, year := 2018]
setnames(values_2018, names(values_2018), gsub(" ", ".", names(values_2018), fixed = T))
values_2018[, `O.NET.SOC.Code` := `O*NET-SOC.Code`]
values <- rbindlist(list(values_2009, values_2013, values_2018), fill = T)
values <- values[, .(O.NET.SOC.Code, Element.Name, Scale.ID, Data.Value, year)]
values <- values[Scale.ID %in% c("EX")]
values[, Element.Name := paste0("vlu_", Element.Name)]

# add education training, etc
education_2009 <- read.delim('/Users/hyork/Documents/projects/occupation/ref/db_14_0 2009.7/Education, 
education_2013 <-  read.delim('/Users/hyork/Documents/projects/occupation/ref/db_18_0_2013.7/Education,
education_2018 <- read_excel("/Users/hyork/Documents/projects/occupation/ref/db_22_2_excel 2018.2/Educa
education_2009[, year := 2009]
education_2013[, year := 2013]
education_2018[, year := 2018]
setnames(education_2018, names(education_2018), gsub(" ", ".", names(education_2018), fixed = T))
education_2018[, `O.NET.SOC.Code` := `O*NET-SOC.Code`]
education <- rbindlist(list(education_2009, education_2013, education_2018), fill = T)
education <- education[,.(Data.Value = stats::weighted.mean(Category, Data.Value)), by = .(O.NET.SOC.Co
education <- education[, .(O.NET.SOC.Code, Element.Name, Scale.ID, Data.Value, year)]
education[, Element.Name := paste0("edu_", Element.Name)]


# # add tasks
# task_2009 <- read.delim('../ref/db_14_0 2009.7/Task Ratings.txt') %>% data.table()
# task_2013 <-  read.delim('../ref/db_18_0_2013.7/Task Ratings.txt') %>% data.table()
# task_2018 <- read_excel("../ref/db_22_2_excel 2018.2/Task Ratings.xlsx") %>% data.table()
# task_2009[, year := 2009]
# task_2013[, year := 2013]
# task_2018[, year := 2018]
# setnames(task_2018, names(task_2018), gsub(" ", ".", names(task_2018), fixed = T))
# task_2018[, `O.NET.SOC.Code` := `O*NET-SOC.Code`]
# task <- rbindlist(list(task_2009, task_2013, task_2018), fill = T)
# task <- task[, .(O.NET.SOC.Code, Task, Scale.ID, Data.Value, Standard.Error, year)]
# task <- task[Scale.ID %in% c("IM")]
# task[, Element.Name := paste0("tsk_", Task)]
# task <- task[!is.na(Task)]
#
skills <- rbindlist(list(skills1, knowledge, abilities, workstyles,
                         workactivities,context, values, education), fill = T)
#skills <- rbindlist(list(skills, knowledge, abilities, workstyles, workactivities), fill = T)

# standardize
```

```
skills[Scale.ID == "LV", Data.Value := Data.Value/7]
skills[Scale.ID == "EX", Data.Value := (Data.Value-1)/6]
skills[Scale.ID == "IM", Data.Value := (Data.Value-1)/4]
skills[Scale.ID == "CX", Data.Value := (Data.Value-1)/4]
skills[Scale.ID == "CT", Data.Value := (Data.Value-1)/2]
skills[Scale.ID == "RW", Data.Value := (Data.Value-1)/9]
skills[Scale.ID == "PT", Data.Value := (Data.Value-1)/7]
skills[Scale.ID == "OJ", Data.Value := (Data.Value-1)/7]
skills[Scale.ID == "RL", Data.Value := (Data.Value-1)/11]

skills[Element.Name %like% "sty_", Scale.ID := "LV"]
skills[Element.Name %like% "ctx", Scale.ID := "LV"]
skills[Element.Name %like% "vlu", Scale.ID := "LV"]
skills[Element.Name %like% "edu_", Scale.ID := "LV"]

#
skills <- skills[Scale.ID == "LV"]
# reformat onet codes to merge
skills[,Element.Name2 := paste0(substr(Element.Name, 1, 15), "_\n", str_sub(Element.Name, -15, -1))]
skills_xwalk <- skills[,.(Element.Name2, Element.Name)] %>% unique()

#fwrite(skills_xwalk, "../ref/skills_xwalk.csv")

skills[,Element.Name := paste0(substr(Element.Name, 1, 15), "_\n", str_sub(Element.Name, -15, -1))]

skills[, OCCSOC := gsub("-", "", substr(O.NET.SOC.Code,1,7))]
skills[, Standard.Error := as.numeric(Standard.Error)]
```

```
## Warning in eval(jsub, SDenv, parent.frame()): NAs introduced by coercion
```

```
skills <- skills[,.(Data.Value = mean(Data.Value),
                    Standard.Error = mean((Standard.Error))), by = .(Element.Name,Scale.ID, OCCSOC)]



########################

####################################################
occ_1990_soc_xwalk[!OCCSOC %in% unique(skills$OCCSOC) & !is.na(OCCSOC), unique(OCCSOC)] -> fixes
for(c.occsoc in fixes){
  keeper <- "continue"
  for(i in 1:4){
    # print(c.occsoc)
    # print(i)
    if(keeper == "continue"){
      fixes_substr <- str_sub(c.occsoc,1,-1-i)
      candidates <- unique(skills$OCCSOC)
      matches <- candidates[str_sub(candidates, 1, -1-i) == fixes_substr]
      if(length(matches > 1)){
        temp <- skills[OCCSOC %in% matches]
        temp[, OCCSOC := c.occsoc]
        temp <- temp[,.(Data.Value = mean(Data.Value),
                        Standard.Error = mean(Standard.Error)), by = .(Element.Name, Scale.ID, OCCSOC)]
        skills <- rbind(temp, skills, fill = T)
        keeper <- "stop"
      }
    }
  }
}
# #create a map of all children
# parents <- data.table(parents = unique(skills$OCCSOC)[unique(skills$OCCSOC) %like% "X$|0$"])
# parents_out <- data.table()
# for(c.parent in parents$parents){
#     substr_parent <- gsub("0$|00$|000$|0000$|X$|XX$|XXX$|XXXX$", "", c.parent, perl = TRUE)
#     children <- data.table(parents = c.parent, children = unique(skills$OCCSOC)[unique(skills$OCCSOC)
#     temp <- merge(parents, children, allow.cartesian = T)
#     if(nrow(temp)>1){
#         parents_out <- rbind(parents_out, temp, fill = T)
#     }
# }
# parents_out[, sub :=  gsub("0$|00$|000$|0000$|X$|XX$|XXX$|XXXX$", "", parents, perl = TRUE)]
# parents_out[, level := 6-nchar(sub)]
#
# # see how many children occsocs aren't in every year
# acs[,.(year, OCCSOC)] %>% unique -> all_children
#
# # get least common set
# all_children[year == 2009, unique(OCCSOC)][all_children[year == 2009, unique(OCCSOC)] %in%
#                                                all_children[year == 2013, unique(OCCSOC)] &
#                                                all_children[year == 2009, unique(OCCSOC)] %in%
#                                                all_children[year == 2018, unique(OCCSOC)]] -> common_
#
# children_fixes <- parents_out[!children %in% common_set]
# children_fixes[,N := .N, by = .(children)]
# children_fixes[,maxN := max(N), by = .(children)]
# children_fixes[,maxlevel := max(level), by = .(children)]
#
# children_fixes <- children_fixes[ N == 1 | level == maxlevel]
#
# children_fixes <- children_fixes[!duplicated(children_fixes$children)]
#
# acs <- merge(acs, children_fixes[,.(parents, children)],
#               by.x = "OCCSOC", by.y = "children", all.x = T)
# acs[!is.na(parents), OCCSOC := parents]
#standardize by percent
skills <- skills[OCCSOC %in% unique(occ_1990_soc_xwalk$OCCSOC)]
skills[,Element.Name := paste0(Element.Name,".", Scale.ID)]

skills_wide <- dcast(skills, OCCSOC   ~Element.Name, value.var = "Data.Value")



####################################################
vars <- names(skills_wide)[names(skills_wide) %like% ".LV"]
```

```
skills_wide <- merge(skills_wide, occ_1990_soc_xwalk, by = "OCCSOC")

skills_final <-skills_wide[,lapply(.SD, mean), by = .(OCC1990), .SDcols = vars]
```

create transition data set

```
######################
job_sum <- merge(job_sum, skills_final,by.x = "occ_1990", by.y = "OCC1990")

setnames(job_sum, vars, paste0(substr(vars,1,15), "\n", str_sub(vars, -15, -1)))
vars <- paste0(substr(vars,1,15), "\n", str_sub(vars, -15, -1))
######################
job_sum <- job_sum[order(CASEID_1979, stop_date)]


job_trans <- job_sum[job_id == 1,.(
  stop_date = stop_date,
  next_start_date = lead(start_date),
  emp_id_old = emp_id,
  emp_id_new = lead(emp_id),
  occ_old = occ_1990,
  occ_new = lead(occ_1990),
  ind_old = ind,
  ind_new = lead(ind),
  wage_old = last_wage,
  wage_new = lead(low_wage),
  reason_left = reason_left,
  race = race,
  sex = sex,
  sample_id = sample_id), by = .(CASEID_1979, highest_grade)]

job_trans_skills <- job_sum[job_id == 1,.SD,
                            .SDcols = vars]

job_trans_skills_new <- job_sum[job_id == 1,lapply(.SD, lead),
                                .SDcols = vars, by = .(CASEID_1979,highest_grade )]

job_trans_skills_new <- job_trans_skills_new[,.SD,
                                             .SDcols = vars]

setnames(job_trans_skills_new, paste0(names(job_trans_skills_new), "_new"))
setnames(job_trans_skills, paste0(names(job_trans_skills), "_old"))

job_trans <- cbind(job_trans, job_trans_skills, job_trans_skills_new)

job_trans <- job_trans[next_start_date - stop_date < 90 & complete.cases(job_trans[,.(wage_old, wage_new

job_trans[reason_left %in% c(1,2,3,4,5,7, 8,9,10),reason_left_binned := "involuntary"]
job_trans[reason_left %in% c(7),reason_left_binned := "fired"]

job_trans[reason_left %in% c(12:32, 39),reason_left_binned := "voluntary"]
```

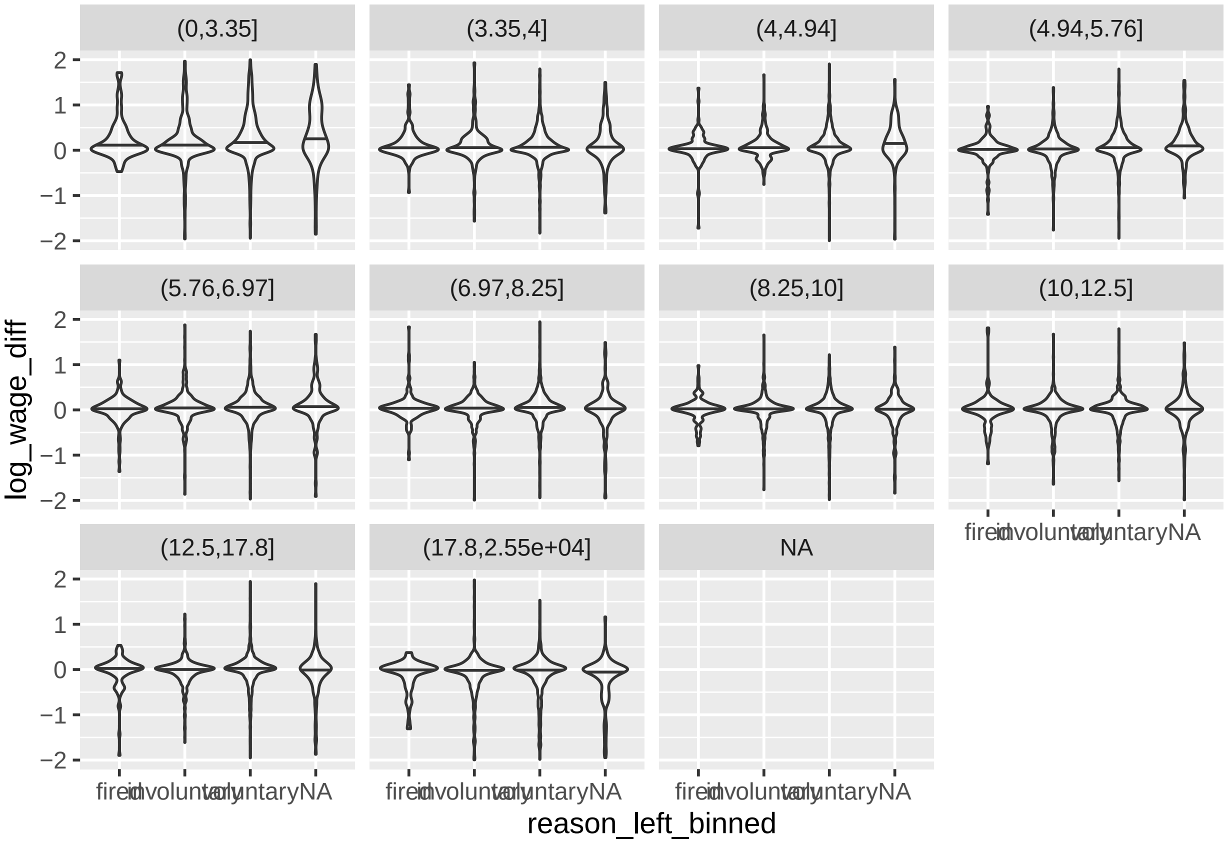
```r
job_trans[, log_wage_diff := log(wage_new) - log(wage_old)]
job_trans[, wage_diff := (wage_new) - (wage_old)]

job_trans[, highest_grade_bin := cut(highest_grade, breaks = c(0,11,12,15, 16, 20))]

job_trans_long <- melt(job_trans, id.vars = names(job_trans)[!str_sub(names(job_trans), 1, -5) %in% vars
job_trans_long[, old_new := str_sub(variable, -3, -1)]
job_trans_long[, variable := str_sub(variable, 1, -5)]
job_trans_long <- dcast(job_trans_long, ... ~ old_new, value.var =  "value")
```

```
## Aggregate function missing, defaulting to 'length'
```

```r
job_trans_skills_diff <- job_trans_long[,.(old_top_5_skills = mean(old[order(old,decreasing = T) &
                                                                         variable %like% "skl_"][1:5]),
                                           new_top_5_skills = mean(new[order(old,decreasing = T) &
                                                                         variable %like% "skl_"][1:5]),
                                           old_mean_skills = mean(old[variable %like% "skl_"]),
                                           new_mean_skills = mean(new[variable %like% "skl_"]),
                                           old_managerial = mean(old[tolower(variable) %like% "anagemen
                                           new_managerial = mean(new[tolower(variable) %like% "anagemen
                                           old_mean_activities = mean(old[variable %like% "act_"]),
                                           new_mean_activities = mean(new[variable %like% "act_"])),
                                        by = c(names(job_trans)[!str_sub(names(job_trans), 1, -5) %in% 

job_trans_skills_diff[, top_5_skills_diff := new_top_5_skills - old_top_5_skills]
job_trans_skills_diff[, mean_skills_diff := new_mean_skills - old_mean_skills]
job_trans_skills_diff[, mean_activities_diff := new_mean_activities - old_mean_activities]
job_trans_skills_diff[, managerial_diff := new_managerial - old_managerial]
```

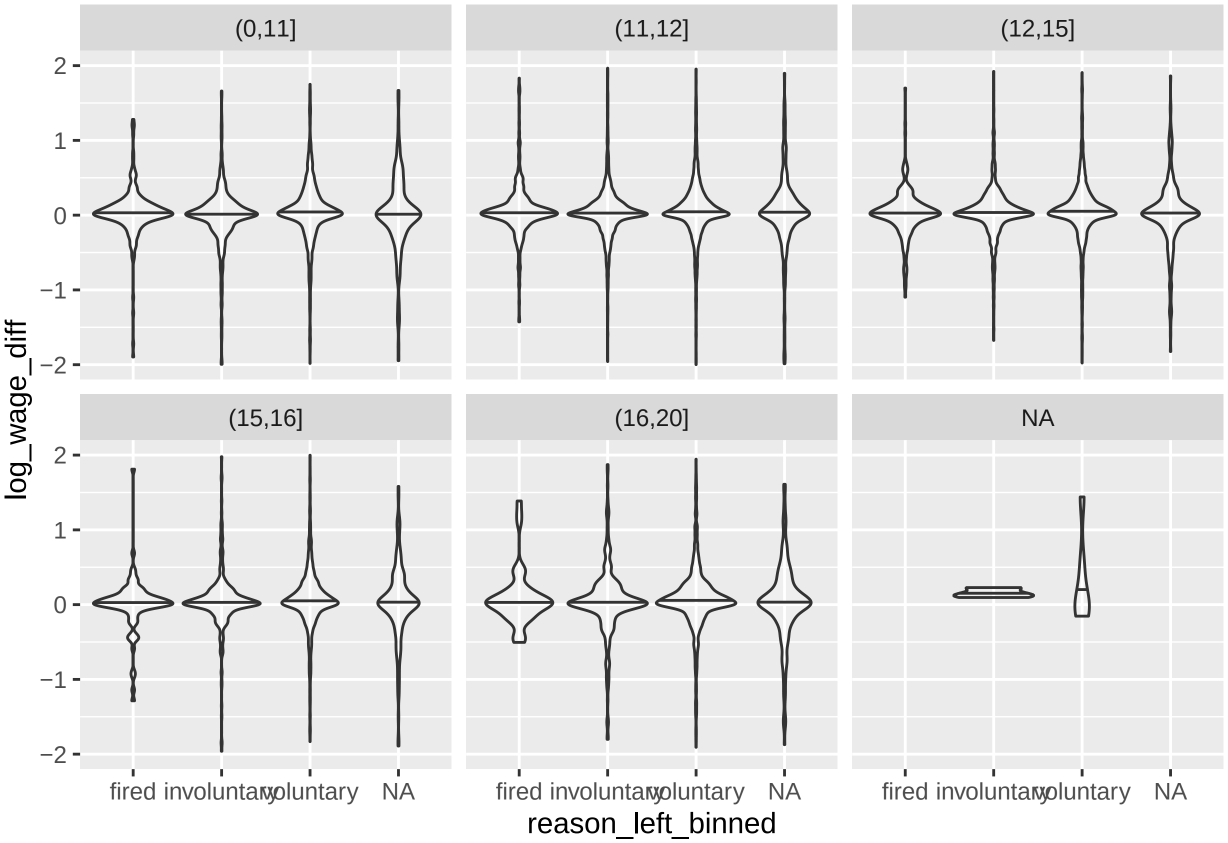
```r
job_trans[,wage_old_binned := cut(wage_old, breaks = quantile(wage_old, seq(0,1,.1)))]
job_trans_skills_diff[,wage_old_binned := cut(wage_old, breaks = quantile(wage_old, seq(0,1,.1)))]

job_trans_skills_diff[, highest_grade_bin := cut(highest_grade, breaks = c(0,11,12,15, 16, 20))]

job_trans_skills_diff[, log_wage_diff_bin := cut(log_wage_diff,c(-8,-1, -.5, -.1, 0, .1, .5, 1,8))]


ggplot(job_trans) +
  geom_violin(draw_quantiles = .5,aes(x = reason_left_binned, y = log_wage_diff), alpha = .2) +
  facet_wrap(~wage_old_binned) +
  ylim(-2, 2)
```

```
## Warning: Removed 370 rows containing non-finite values (stat_ydensity).

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



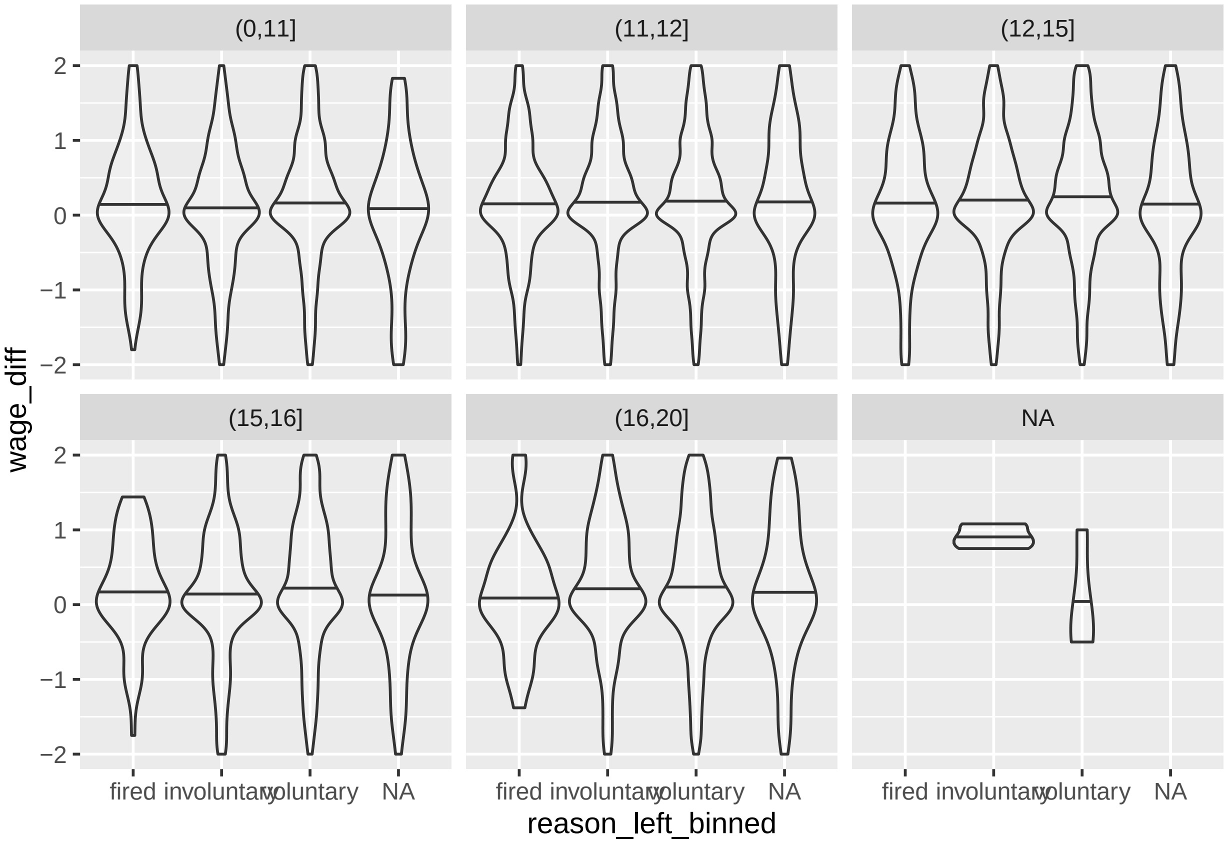
```
ggplot(job_trans) +
  geom_violin(draw_quantiles = .5,aes(x = reason_left_binned, y = log_wage_diff), alpha = .2) +
  facet_wrap(~highest_grade_bin) +
  ylim(-2, 2)
```

```
## Warning: Removed 370 rows containing non-finite values (stat_ydensity).

## Warning: collapsing to unique 'x' values
```



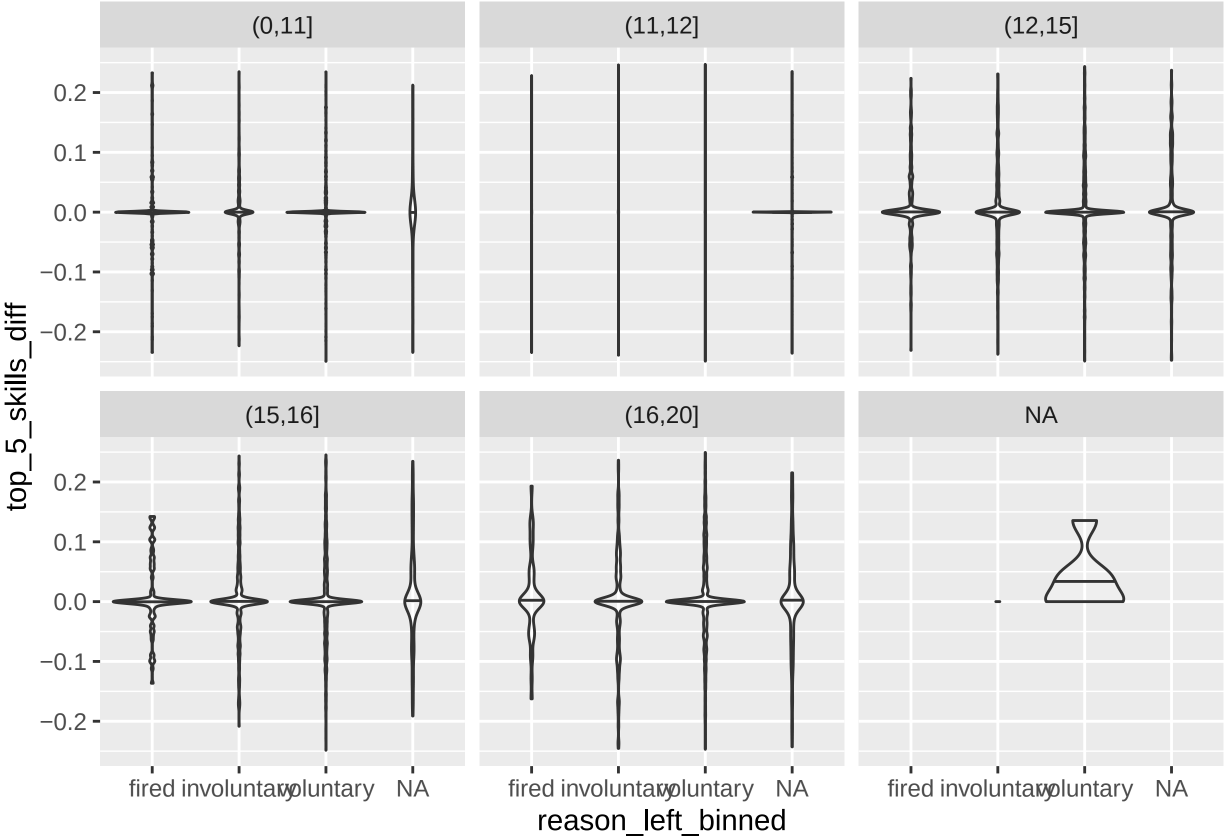
```r
ggplot(job_trans) +
  geom_violin(draw_quantiles = .5,aes(x = reason_left_binned, y = wage_diff), alpha = .2) +
  facet_wrap(~highest_grade_bin) +
  ylim(-2, 2)
```

```
## Warning: Removed 6868 rows containing non-finite values (stat_ydensity).
```



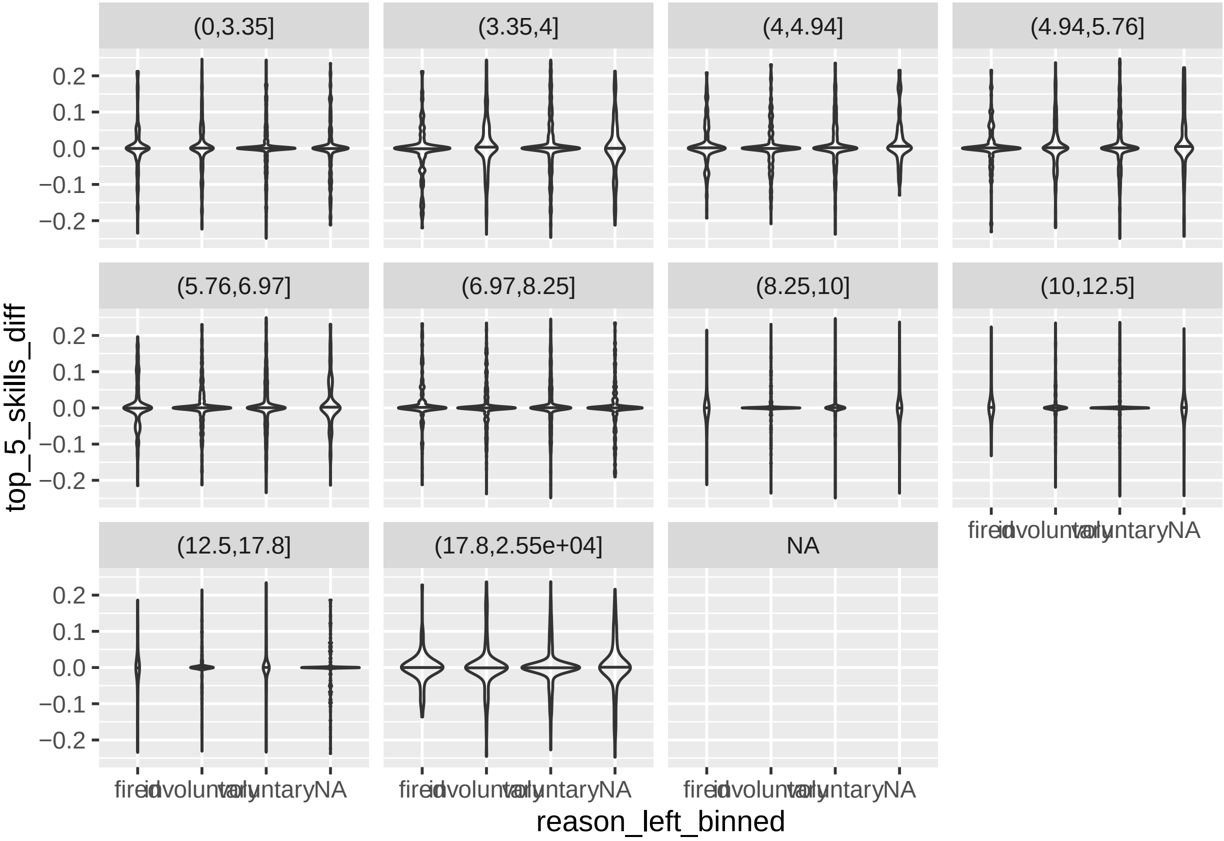
```
ggplot(job_trans_skills_diff) +
  facet_wrap(~highest_grade_bin) +
  geom_violin(draw_quantiles = .5,aes(x = reason_left_binned, y = top_5_skills_diff), alpha = .2) +
  ylim(-.25, .25)
```

```
## Warning: Removed 121 rows containing non-finite values (stat_ydensity).

## Warning: collapsing to unique 'x' values

## Warning: collapsing to unique 'x' values

## Warning: collapsing to unique 'x' values
```



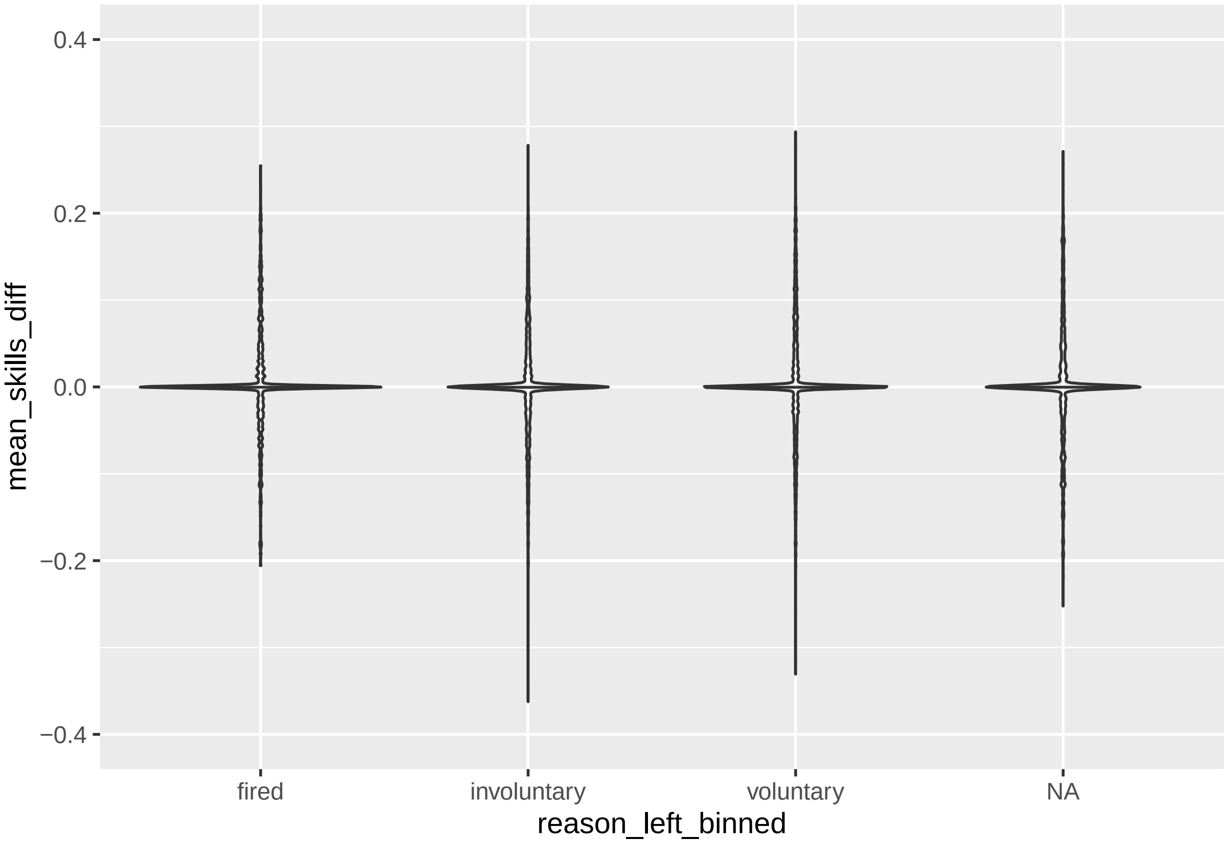
```r
ggplot(job_trans_skills_diff) +
  geom_violin(draw_quantiles = .5,aes(x = reason_left_binned, y = top_5_skills_diff), alpha = .2) +
  facet_wrap(~wage_old_binned) +
  ylim(-.25, .25)
```

```
## Warning: Removed 121 rows containing non-finite values (stat_ydensity).

## Warning in max(data$density): no non-missing arguments to max; returning -Inf

## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



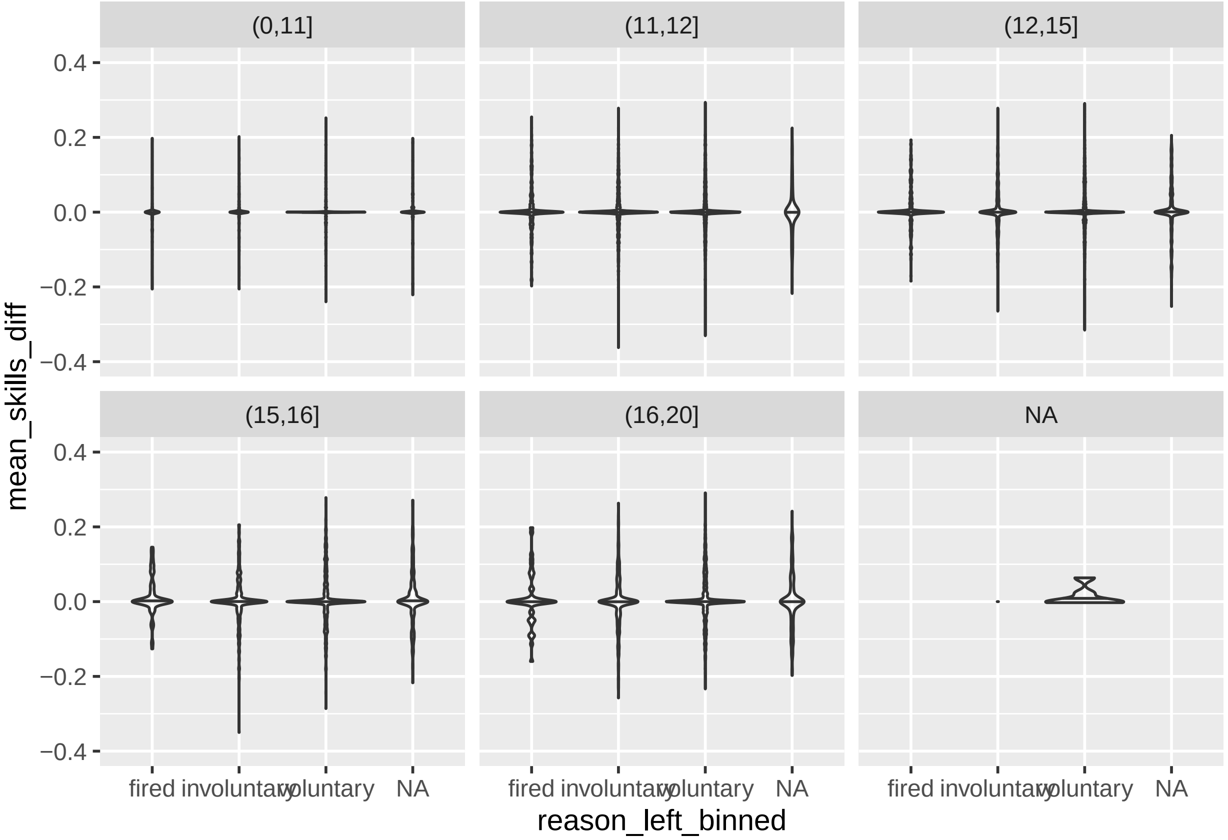
```
ggplot(job_trans_skills_diff) +
  geom_violin(draw_quantiles = .5,aes(x = reason_left_binned, y = mean_skills_diff), alpha = .2) +
  ylim(-.4, .4)
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



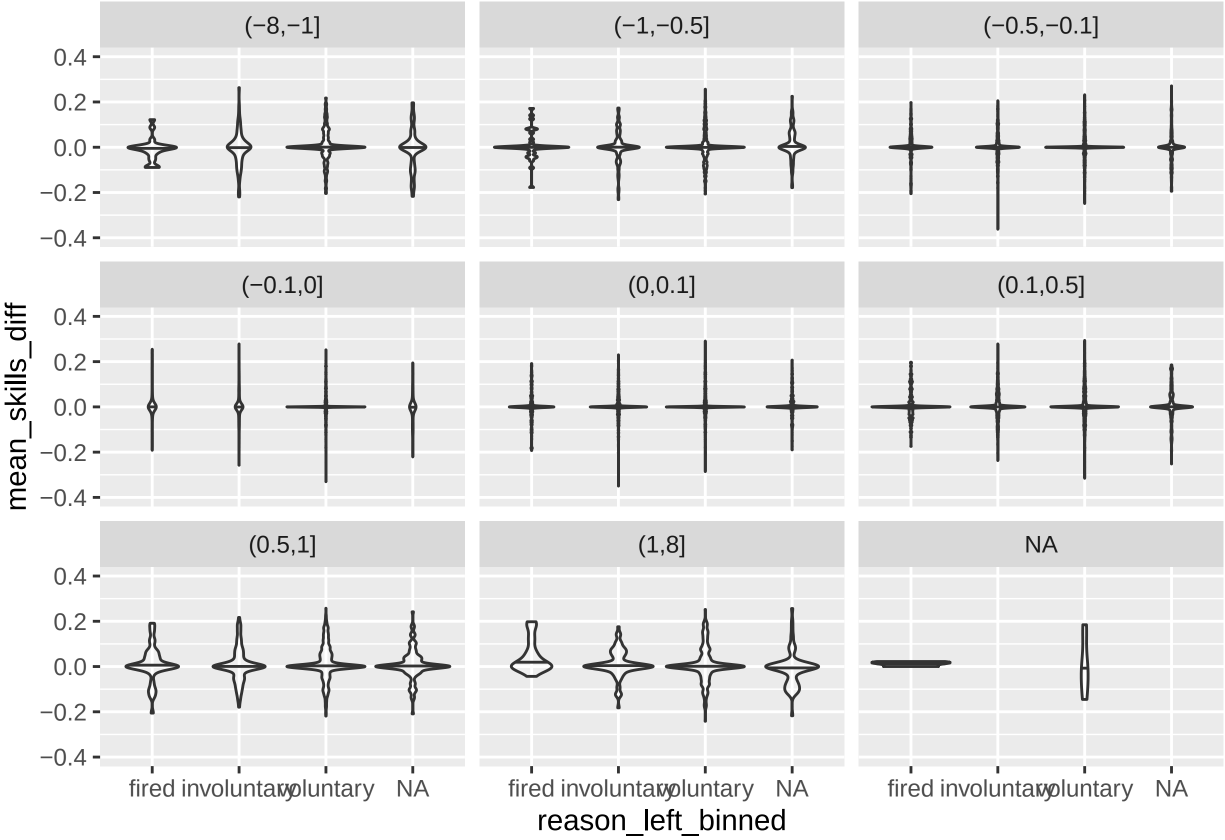
```r
ggplot(job_trans_skills_diff) +
  geom_violin(draw_quantiles = .5,aes(x = reason_left_binned, y = mean_skills_diff), alpha = .2) +
  facet_wrap(~highest_grade_bin) +
  ylim(-.4, .4)
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



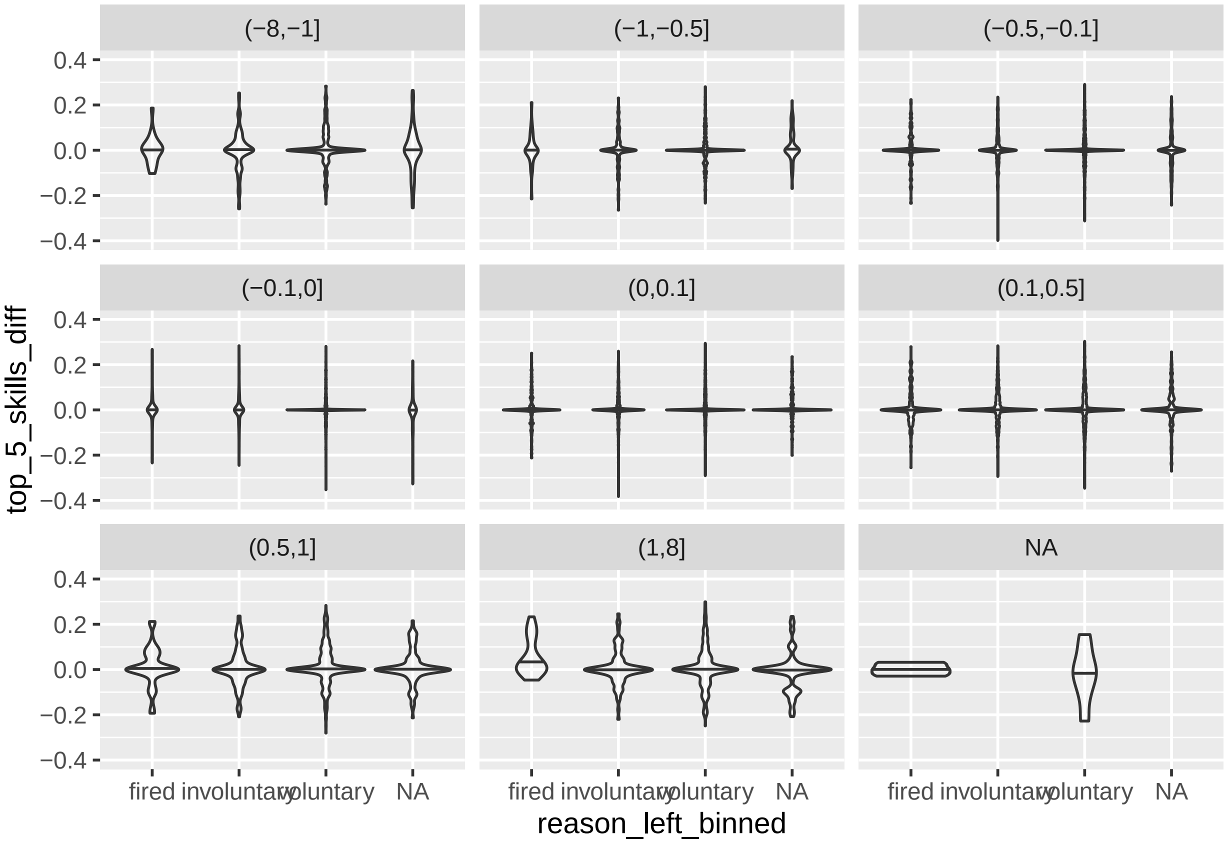
```
ggplot(job_trans_skills_diff) +
  geom_violin(draw_quantiles = .5,aes(x = reason_left_binned, y = mean_skills_diff), alpha = .2) +
  facet_wrap(~log_wage_diff_bin) +
  ylim(-.4, .4)
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



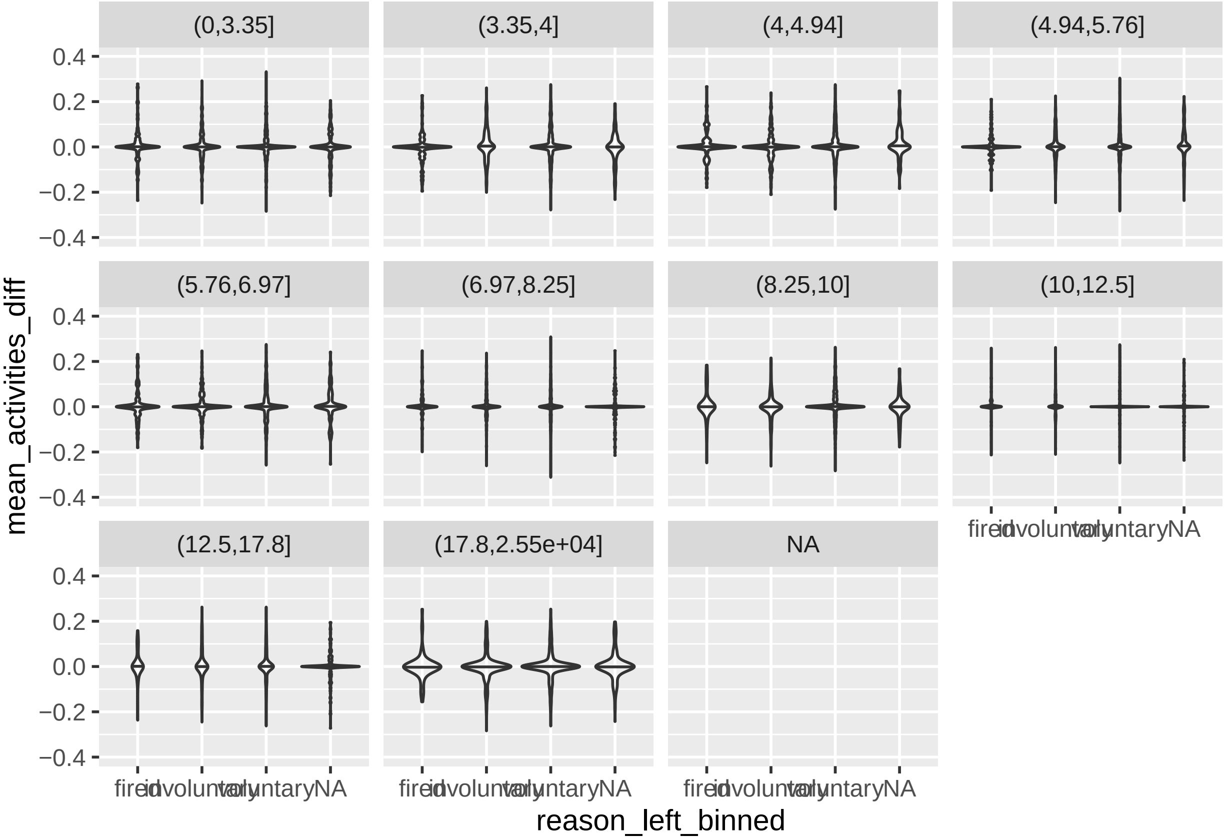
```
ggplot(job_trans_skills_diff) +
  geom_violin(draw_quantiles = .5,aes(x = reason_left_binned, y = top_5_skills_diff), alpha = .2) +
  facet_wrap(~log_wage_diff_bin) +
  ylim(-.4, .4)
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



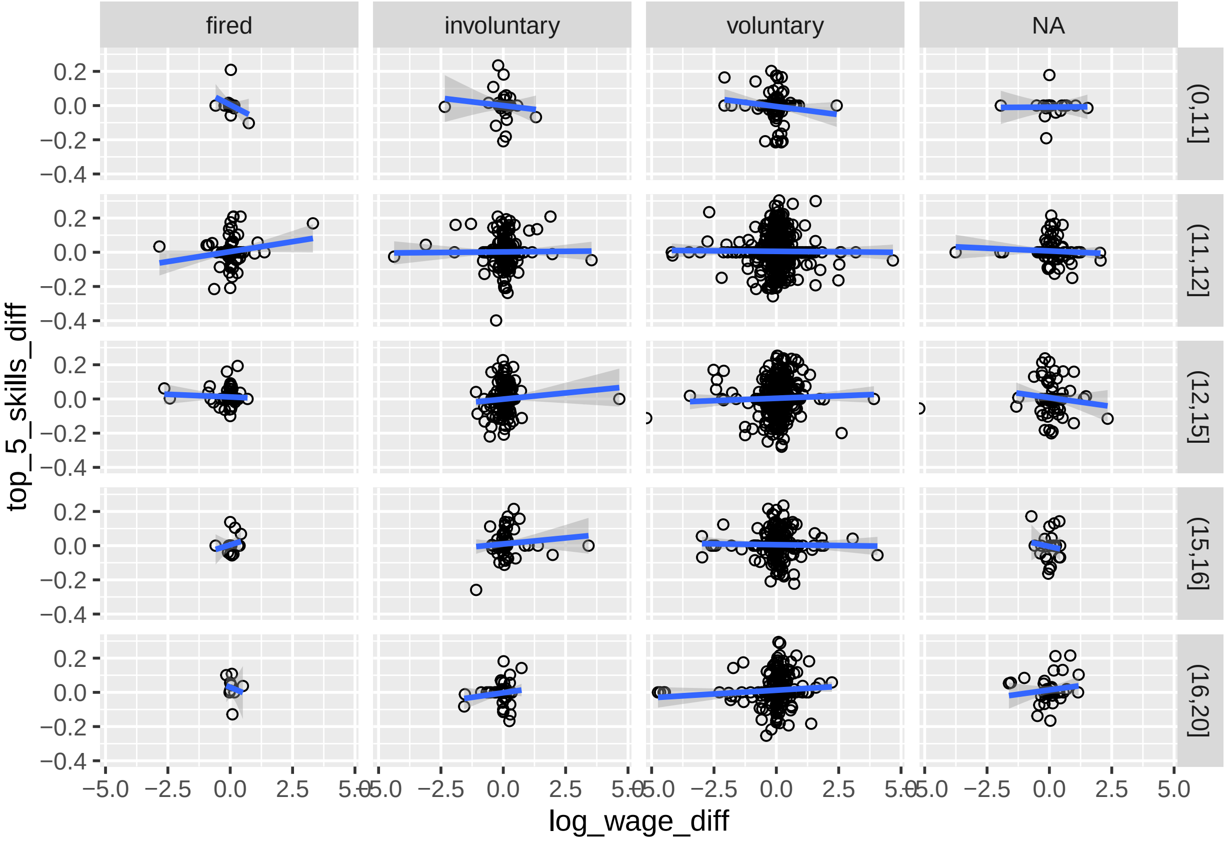
```r
ggplot(job_trans_skills_diff) +
  geom_violin(draw_quantiles = .5,aes(x = reason_left_binned, y = mean_activities_diff), alpha = .2) +
  facet_wrap(~wage_old_binned) +
  ylim(-.4, .4)
```

```
## Warning in max(data$density): no non-missing arguments to max; returning -Inf

## Warning: Computation failed in `stat_ydensity()`:
## replacement has 1 row, data has 0

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



```
ggplot(job_trans_skills_diff[sex ==2 & race == 2]) +
  geom_point(aes(x = log_wage_diff, y = top_5_skills_diff), shape = 1) +
  facet_grid(highest_grade_bin~reason_left_binned) +
  geom_smooth(aes(x =  log_wage_diff, y = top_5_skills_diff), method = "lm")
```

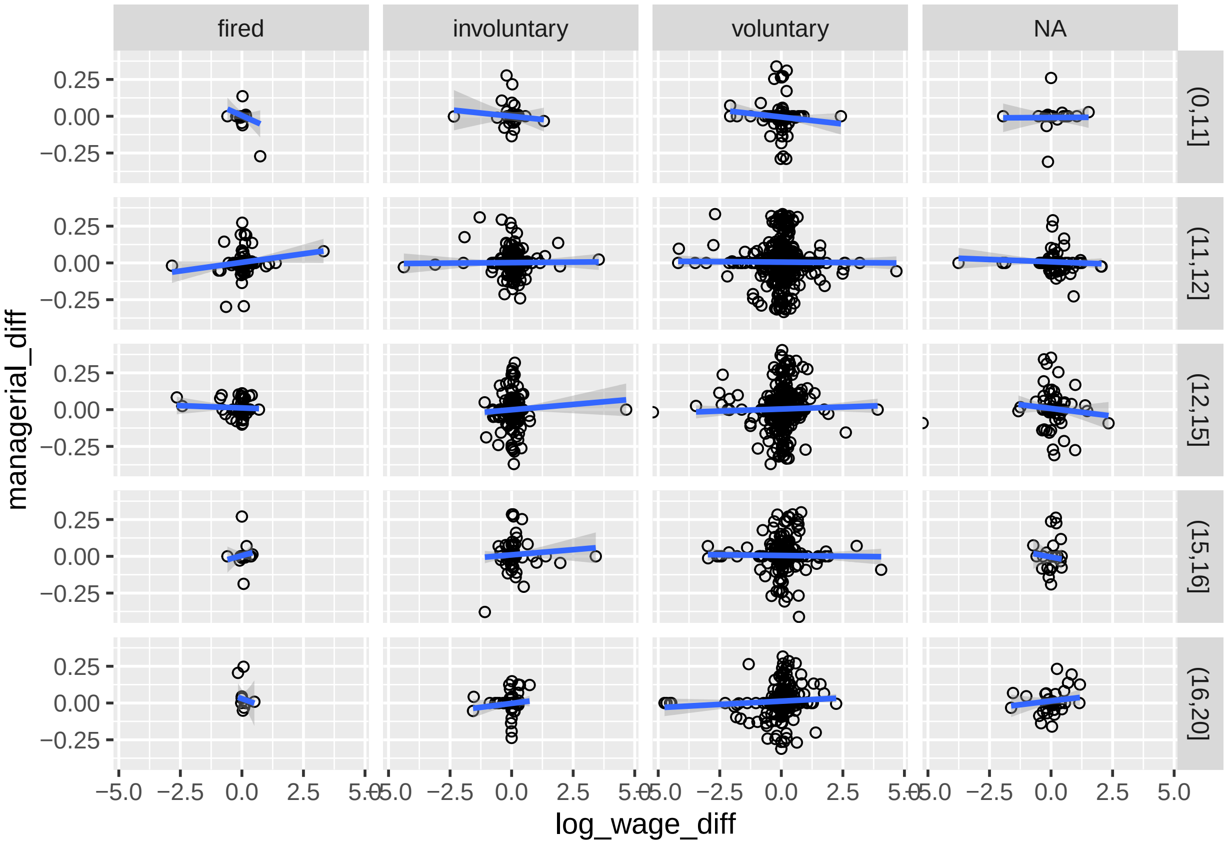
```
## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 2 rows containing non-finite values (stat_smooth).
```



```
ggplot(job_trans_skills_diff[sex ==2 & race == 2]) +
  geom_point(aes(x = log_wage_diff, y = managerial_diff), shape = 1) +
  facet_grid(highest_grade_bin~reason_left_binned) +
  geom_smooth(aes(x =  log_wage_diff, y = top_5_skills_diff), method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 2 rows containing non-finite values (stat_smooth).
```

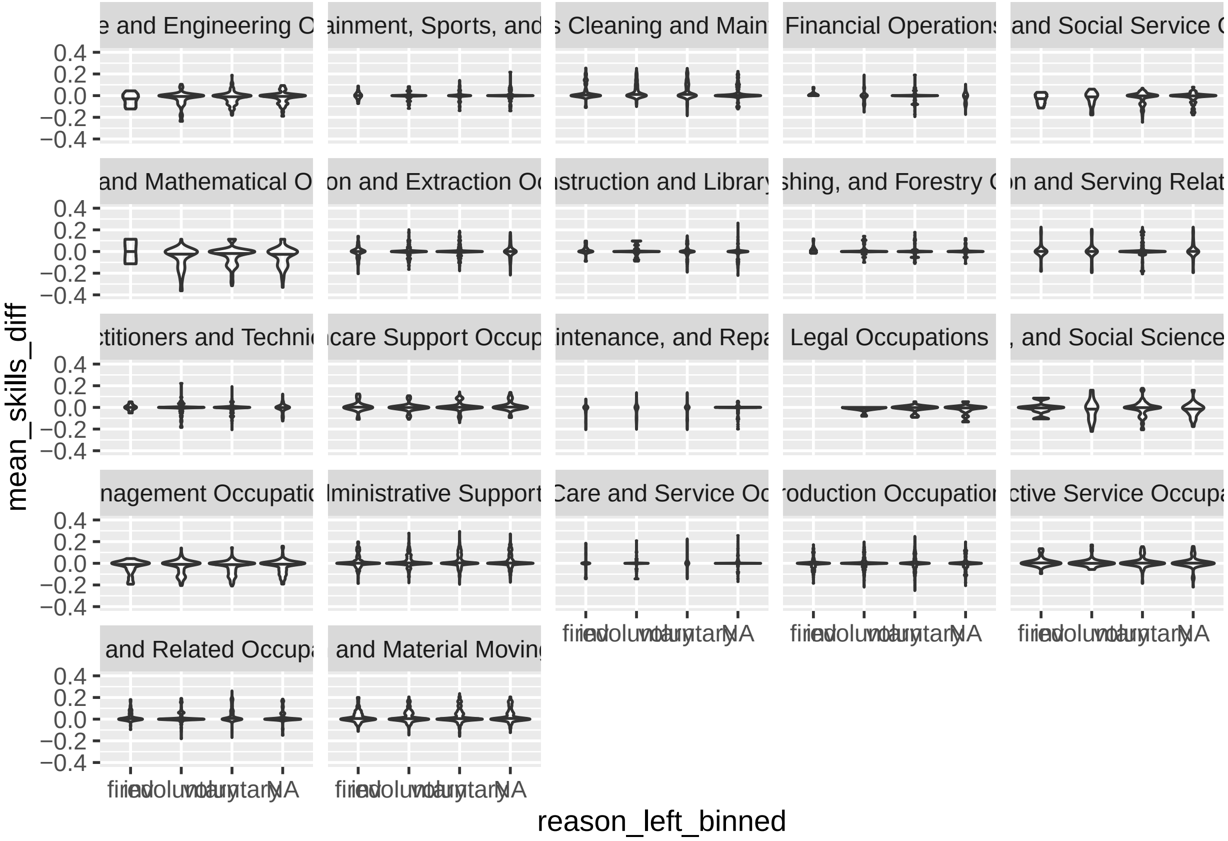


```
soc_xwalk <- fread("../../monster_jobs/inputs/soc_hierarchy_xwalk.csv")
occ_1990_soc_xwalk_new <- occ_1990_soc_xwalk[!duplicated(OCC1990)]
job_trans_skills_diff <- merge(occ_1990_soc_xwalk_new, job_trans_skills_diff, by.x = "OCC1990", by.y =
job_trans_skills_diff[, occ_soc_pref := as.numeric(substr(OCCSOC,1,2))]
job_trans_skills_diff <- merge(job_trans_skills_diff, soc_xwalk, by.x = "occ_soc_pref", by.y = "occ_agg
```

```
ggplot(job_trans_skills_diff) +
  geom_violin(draw_quantiles = .5,aes(x = reason_left_binned, y = mean_skills_diff), alpha = .2) +
  facet_wrap(~ soc_agg_group) +
  ylim(-.4, .4)
```

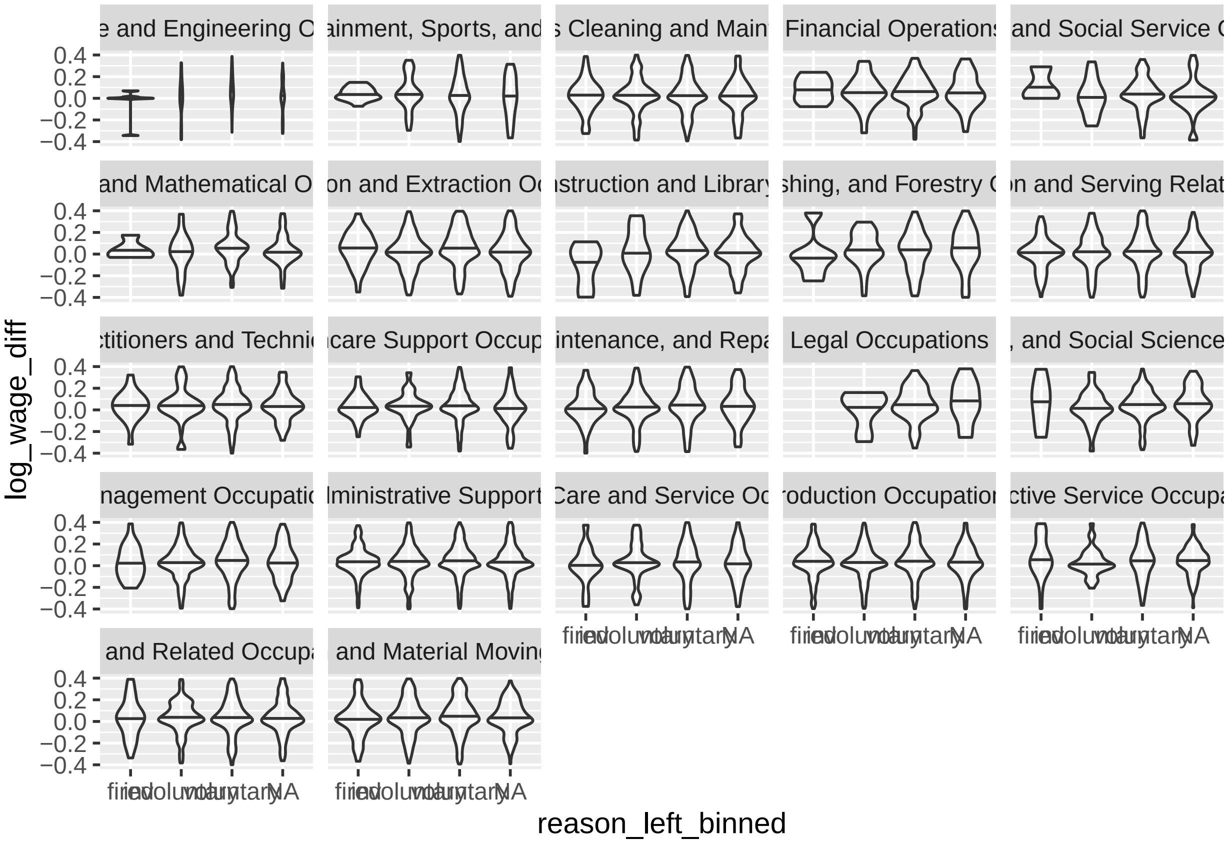
```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
```

```
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



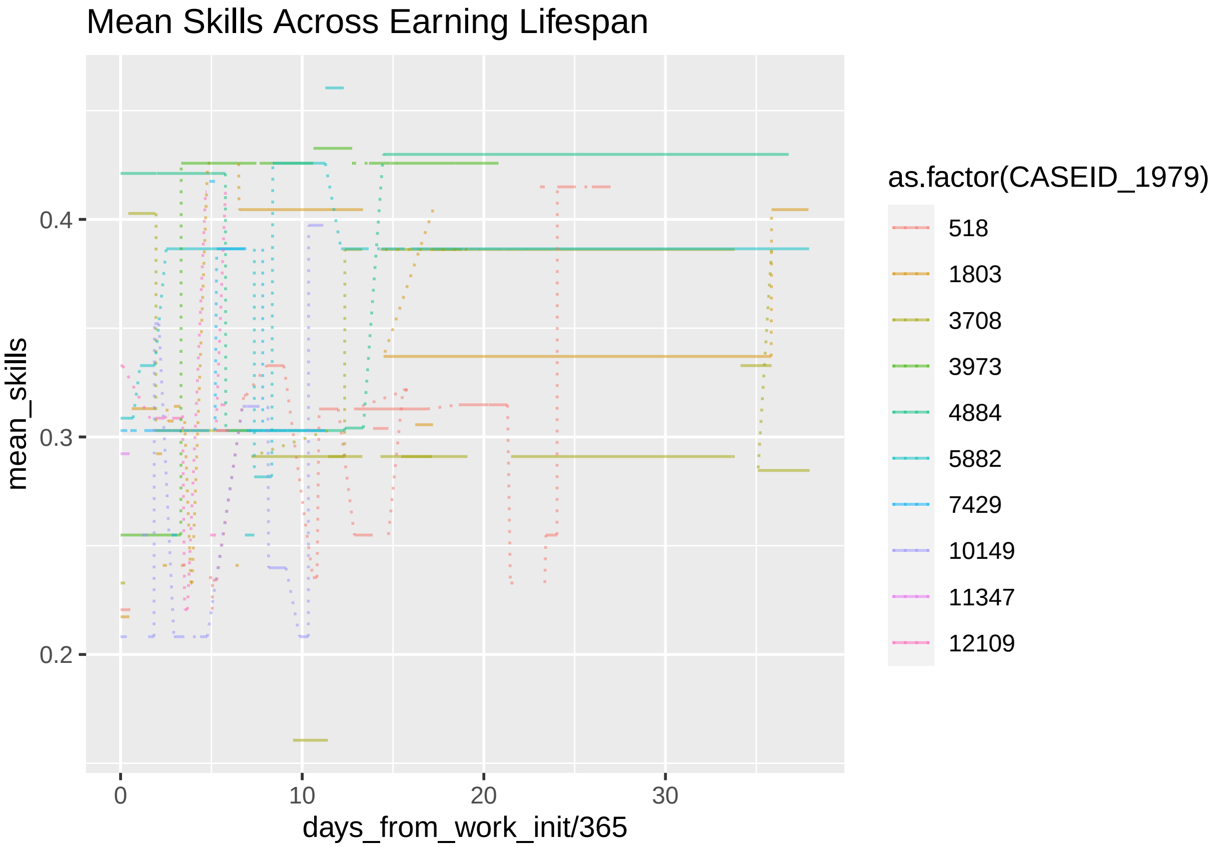
```r
ggplot(job_trans_skills_diff) +
  geom_violin(draw_quantiles = .5,aes(x = reason_left_binned, y = log_wage_diff), alpha = .2) +
  facet_wrap(~ soc_agg_group) +
  ylim(-.4, .4)
```

```
## Warning: Removed 4572 rows containing non-finite values (stat_ydensity).

## Warning: collapsing to unique 'x' values
```



```
job_trans_skills_diff[, length_of_job_loss := as.numeric(next_start_date - stop_date)]
#job_trans_skills_diff[length_of_job_loss >= 0]
job_trans_skills_diff[, length_of_job_loss_binned := cut(length_of_job_loss, c(-1,1,7,14, 30, 60))]
job_trans_skills_diff[, same_emp := ifelse(emp_id_old == emp_id_new,1,0)]




job_sum[,days_from_work_init := as.numeric(start_date - min(start_date)), by = .(CASEID_1979)]
job_sum[,days_from_work_init_stop :=  as.numeric(stop_date - min(start_date)), by = .(CASEID_1979)]


job_sum[,mean_skills := rowMeans(.SD), .SDcols = names(job_sum)[names(job_sum) %like% "skl"]]
job_sum[job_id == 1,lag_mean_skills := lag(mean_skills), by = .(CASEID_1979)]

job_sum[occ != 0]%>%
  .[CASEID_1979 %in% sample(unique(CASEID_1979), 10)] %>%
  ggplot() +
  geom_segment(aes(x = days_from_work_init/365,
                   xend = days_from_work_init_stop/365,
                   y = mean_skills,
                   yend = mean_skills,
                   group = CASEID_1979,
                   color = as.factor(CASEID_1979)), alpha = .5) +
  geom_segment(aes(x = lag(days_from_work_init_stop)/365,
                   xend = days_from_work_init/365,
                   y = lag_mean_skills,
                   yend = mean_skills,
                   group = CASEID_1979,
                   color = as.factor(CASEID_1979)), linetype = "dotted", alpha = .5)+
  ggtitle("Mean Skills Across Earning Lifespan")
```

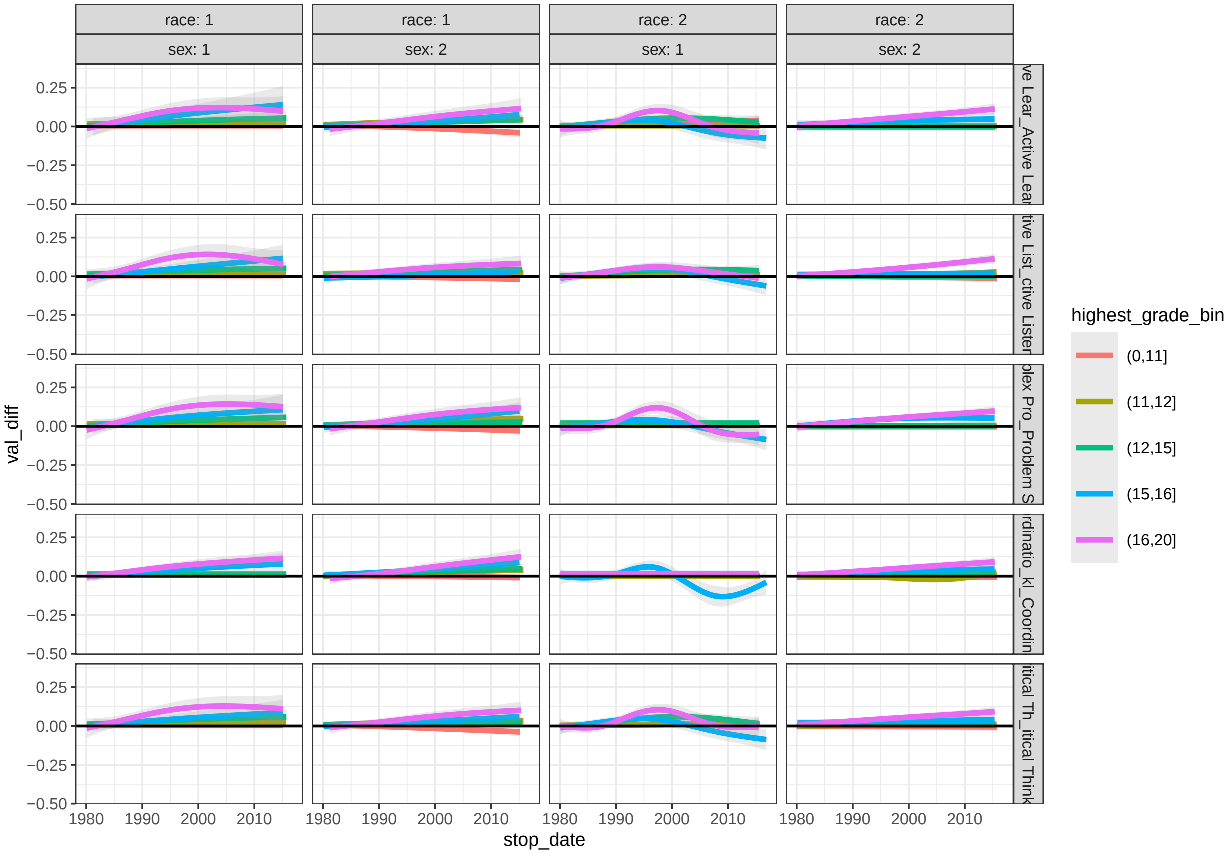
```
## Warning: Removed 62 rows containing missing values (geom_segment).
```



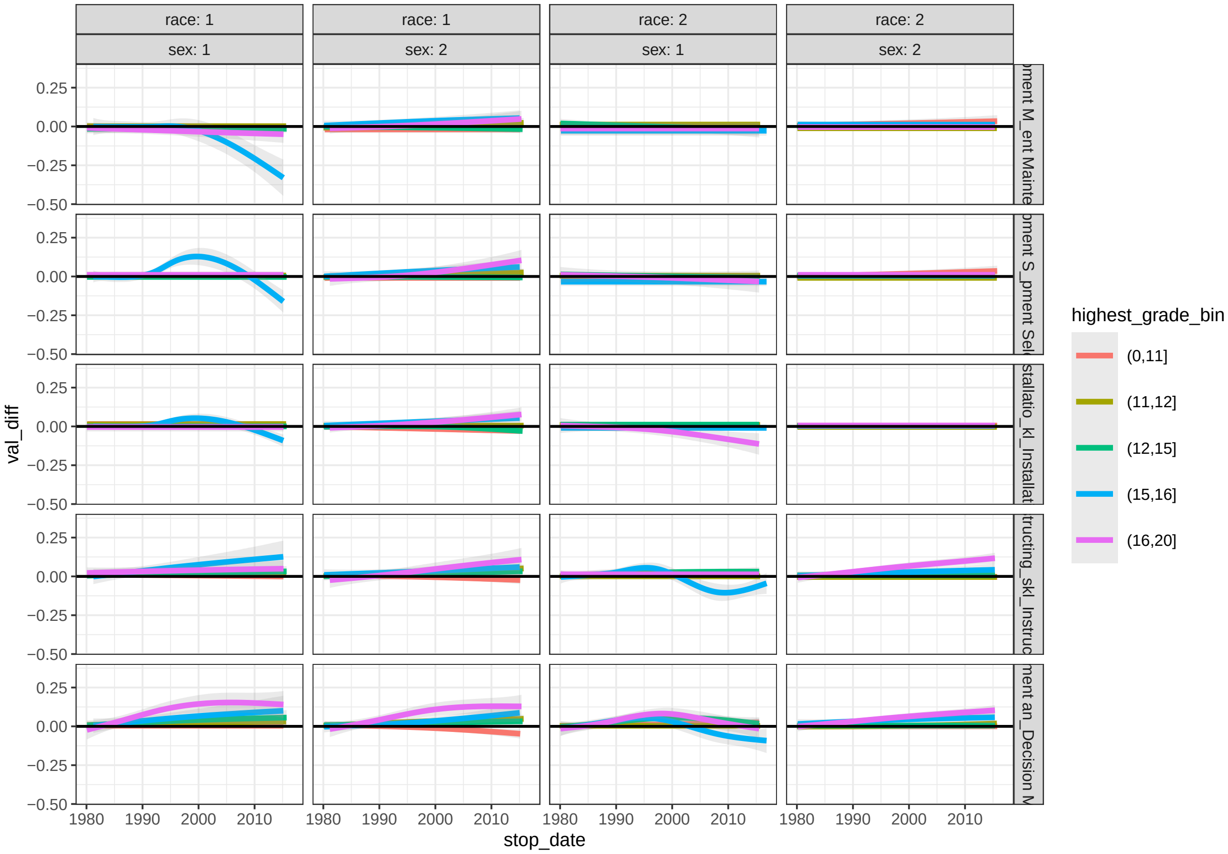
#
growth trajectories of skills

```
job_trans_long_2 <- melt(job_trans, id.vars = names(job_trans)[!str_sub(names(job_trans), 1, -5) %in% va
job_trans_long_2[, old_new := str_sub(variable, -3, -1)]
job_trans_long_2 <- job_trans_long_2[old_new == "old"]
job_trans_long_2[, variable := str_sub(variable, 1, -5)]
job_trans_long_2[, orig_val := mean(value[stop_date == min(stop_date)]), by = .(CASEID_1979, variable)]
job_trans_long_2[, val_diff := value - orig_val]

job_trans_long_2 <-job_trans_long_2[CASEID_1979 %in% sample(unique(CASEID_1979), 3000)] %>% copy()
library(ggrepel)
library(ggforce)
c.pre <- "skl"
ceiling(length(vars[substr(vars,1,3) == c.pre])/5)-> npages
for(c.page in 1:npages){
  job_trans_long_2[substr(variable,1,3) == c.pre & !is.na(highest_grade_bin) & race %in% 1:2] %>%
    ggplot(.) +
    geom_smooth(aes(x = stop_date, y = val_diff, color = highest_grade_bin,
                    group = highest_grade_bin), alpha = .2, method = "gam" ) +
    facet_grid_paginate(variable ~ paste0("race: ", race) + paste0("sex: ", sex),ncol = 4,
                        nrow = 5, page = c.page) +
    geom_hline(aes(yintercept = 0)) -> p
  print(p)
}
```

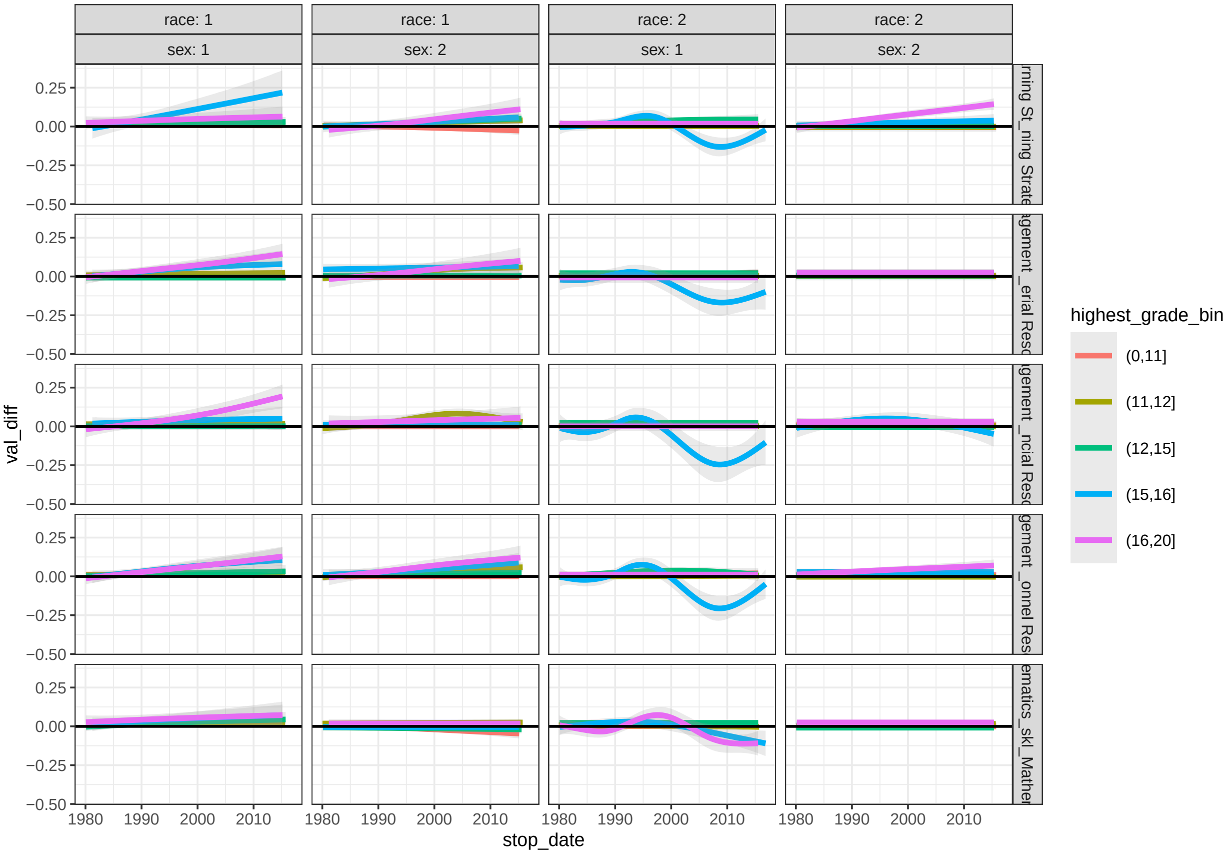
```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```



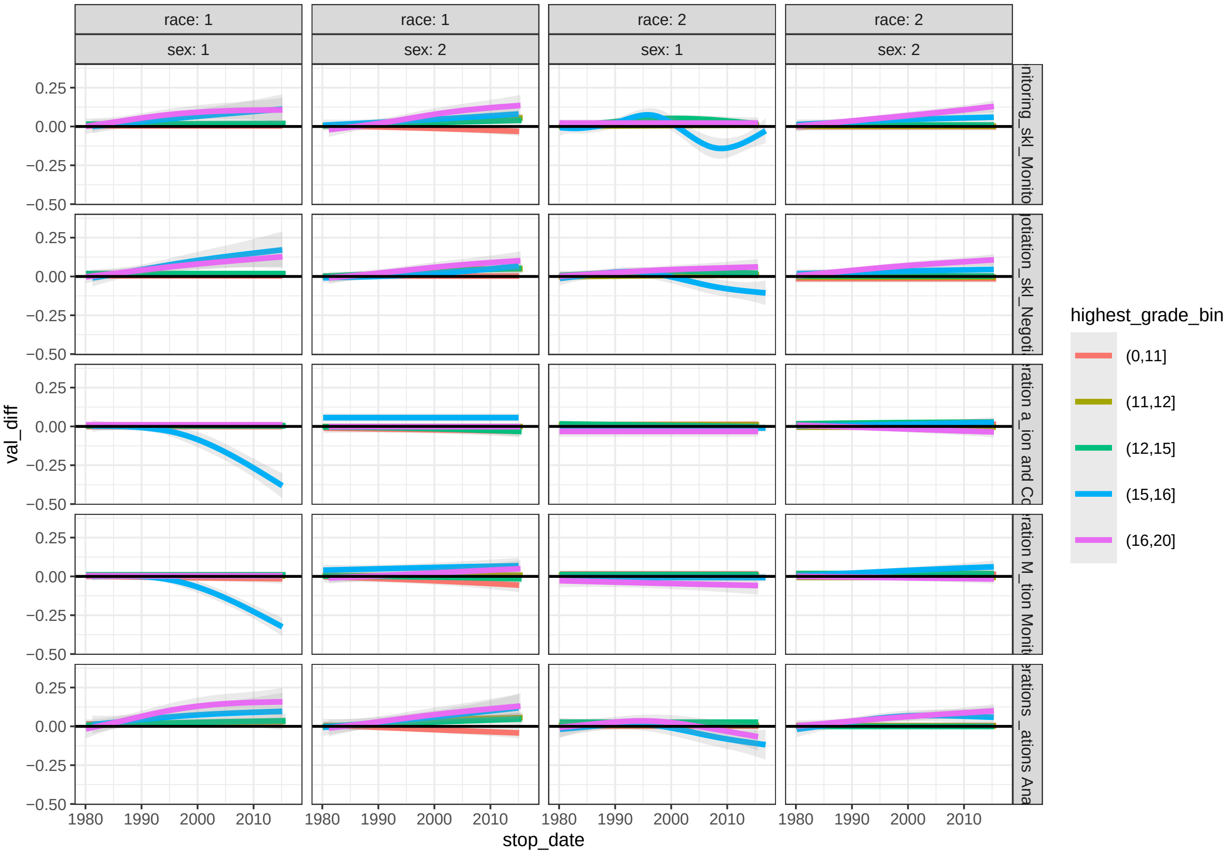
```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```



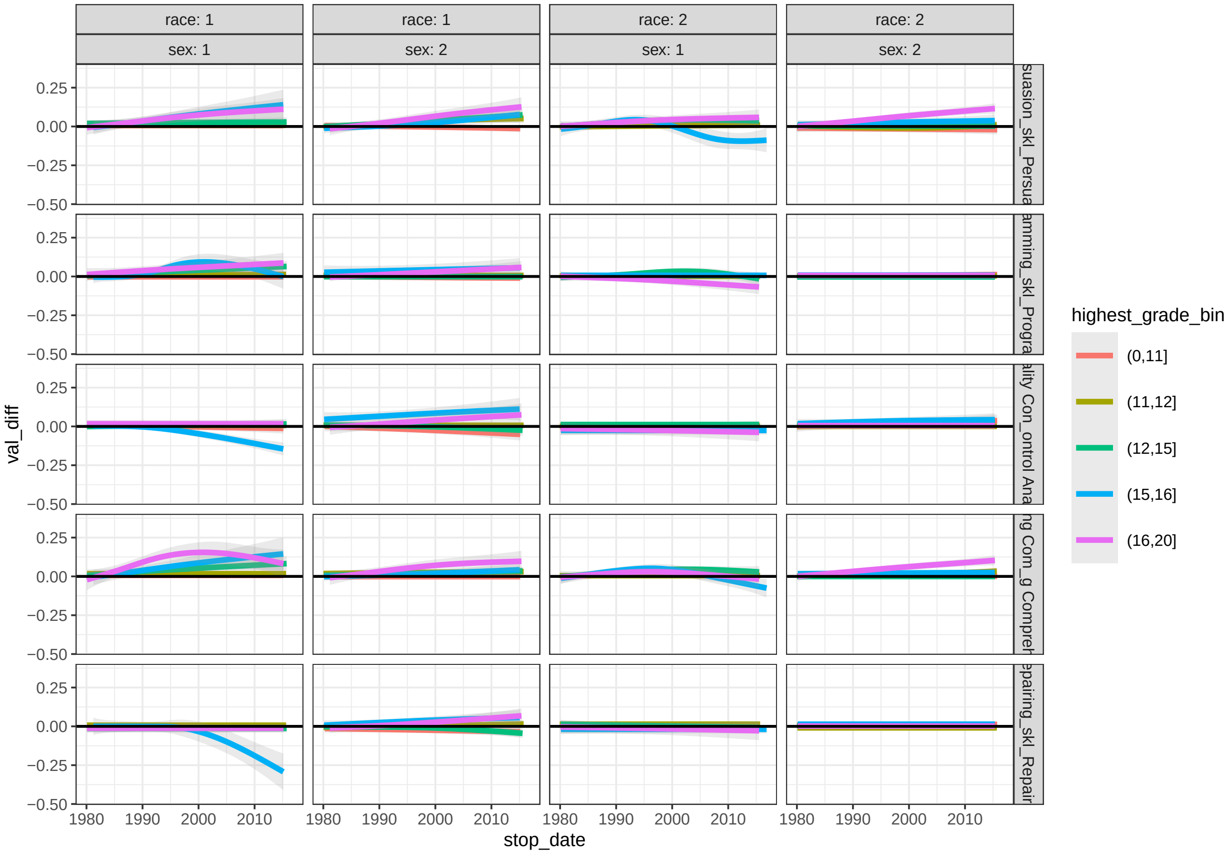
```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```



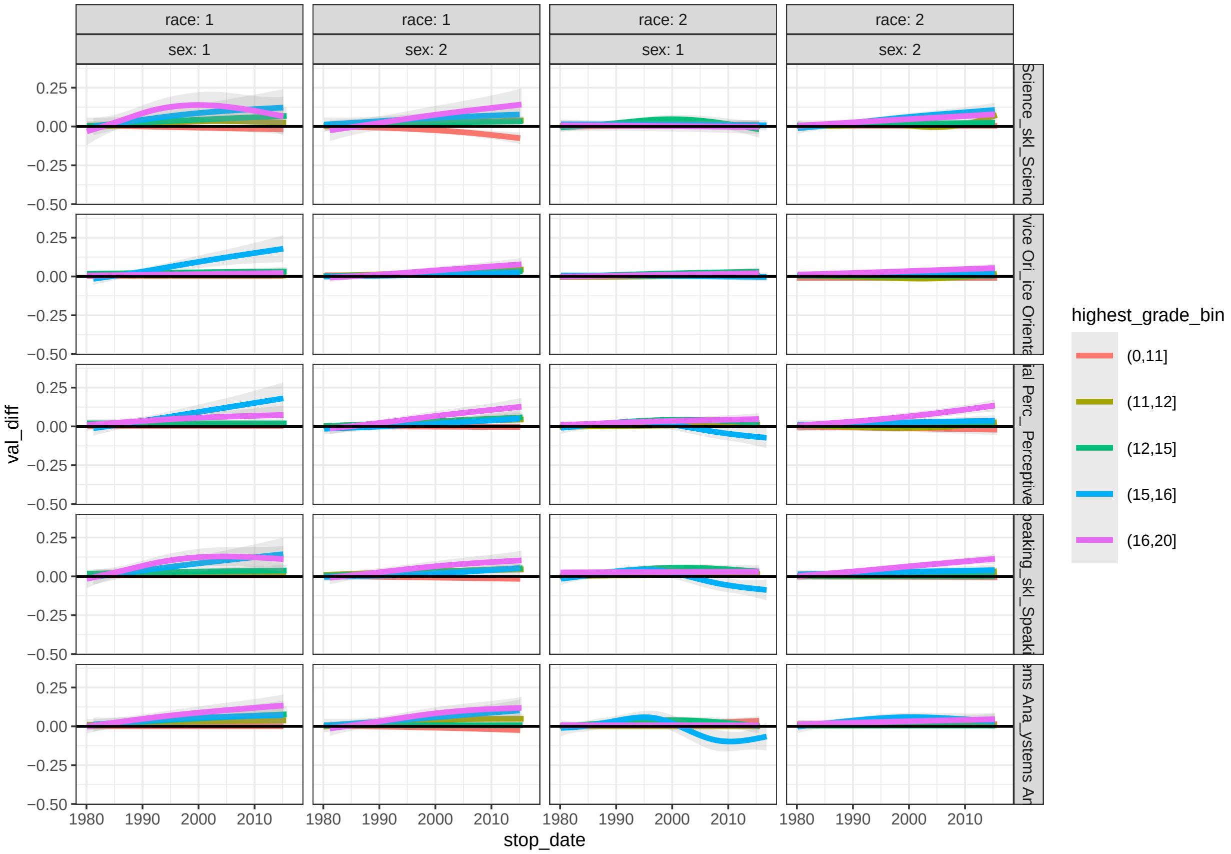
```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```



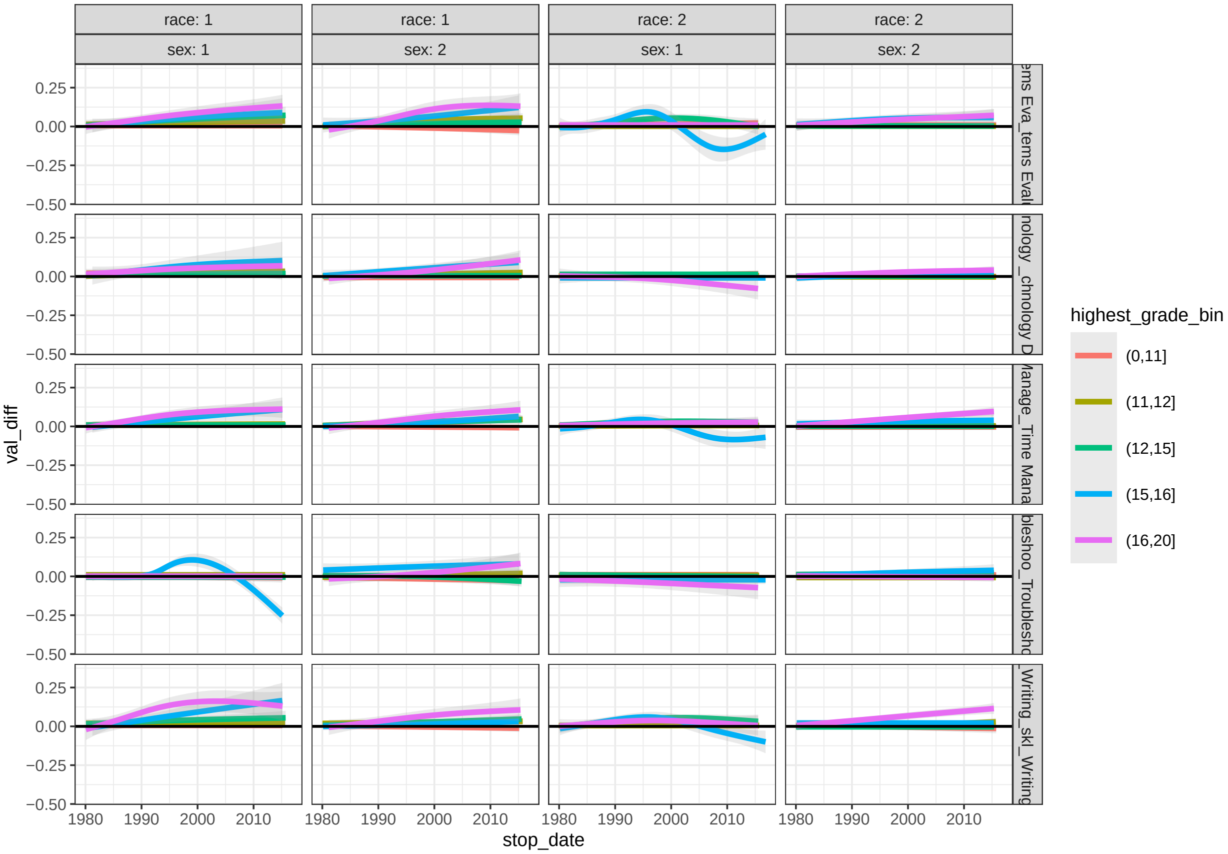
```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```



```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```



```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```


race: 1
sex: 1
race: 1
sex: 2
race: 2
sex: 1
race: 2
sex: 2
ems Eva_tems Eval
nology _chnology D
Manage_Time Mana
bleshoo_Troublesho
Writing_skl_Writing
val_diff
0.25
0.00
−0.25
−0.50
1980
1990
2000
2010
stop_date
highest_grade_bin
(0,11]
(11,12]
(12,15]
(15,16]
(16,20]