HNET analysis

Philipp Chapkovski

05 January, 2021

```
library('pacman')
p_load('dplyr', 'estimatr', 'tidyverse', 'readxl', 'sjPlot', 'emmeans', 'olsrr', 'writexl', 'gtsummary'
       'hrbrthemes','kableExtra', 'skimr','lubridate', 'data.table', 'gganimate','transformr', 'png','g
```

```
con <- dbConnect(odbc::odbc(),
                 driver = "/usr/local/lib/libsqlite3odbc.dylib",
                 database = "/Users/chapkovski/archive/hnet_scraper/db.sqlite3")
dm <- dm_from_src(con)

dm<-dm %>%
  dm_add_pk(hnet_structuredpost,id) %>%
  dm_add_pk(hnet_position,id) %>%
  dm_add_fk(hnet_structuredpost_positions,structuredpost_id, hnet_structuredpost)%>%
  dm_add_fk(hnet_structuredpost_positions,position_id, hnet_position)%>%
  dm_add_pk(hnet_category,id) %>%
  dm_add_fk(hnet_structuredpost_primary_categories,structuredpost_id, hnet_structuredpost)%>%
  dm_add_fk(hnet_structuredpost_primary_categories,category_id, hnet_category)%>%
    dm_add_fk(hnet_structuredpost_secondary_categories,structuredpost_id, hnet_structuredpost)%>%
  dm_add_fk(hnet_structuredpost_secondary_categories,category_id, hnet_category)




drop.cols=c('created_at', 'updated_at' )
position_tab <- dm %>%
  dm_select(hnet_position, -one_of(drop.cols)) %>%
  dm_select(hnet_structuredpost, -one_of(drop.cols))%>%
  dm_disambiguate_cols(sep='.')%>%
  dm_flatten_to_tbl(hnet_structuredpost_positions) %>%
  as_tibble()%>%
  rename(position=description)%>%
  mutate(position=str_trim(position)) %>%
    mutate(year=lubridate::year(posting_date),
          month=lubridate::month(posting_date),
          day=lubridate::day(posting_date),
          weekday=lubridate::wday(posting_date),
          week=lubridate::isoweek((posting_date)),
          monthdate = as.Date(paste("2020", month, '01', sep = "-")),
          fixeddate = as.Date(paste("2020", month, day, sep = "-"))
          )

# Preparing primary category analysis
```

```
#

primary_cat_tab<-dm %>%
  dm_select(hnet_category, -one_of(drop.cols)) %>%
  dm_select(hnet_structuredpost, -one_of(drop.cols))%>%
  dm_disambiguate_cols(sep='.')%>%
  dm_flatten_to_tbl(hnet_structuredpost_primary_categories) %>%
  as_tibble()%>%
  rename(category=description)%>%
  mutate(category=str_trim(category)) %>%
    mutate(year=lubridate::year(posting_date),
          month=lubridate::month(posting_date),
          day=lubridate::day(posting_date),
          weekday=lubridate::wday(posting_date),
          week=lubridate::isoweek((posting_date)),
          monthdate = as.Date(paste("2020", month, '01', sep = "-")),
          fixeddate = as.Date(paste("2020", month, day, sep = "-"))
          )

# Preparing secondary category analysis
#

secondary_cat_tab<-
  dm %>%
  dm_select(hnet_category, -one_of(drop.cols)) %>%
  dm_select(hnet_structuredpost, -one_of(drop.cols))%>%
  dm_disambiguate_cols(sep='.')%>%
  dm_flatten_to_tbl(hnet_structuredpost_secondary_categories) %>%
  as_tibble() %>%
  rename(category=description)%>%
  mutate(category=str_trim(category)) %>%
    mutate(year=lubridate::year(posting_date),
          month=lubridate::month(posting_date),
          day=lubridate::day(posting_date),
          weekday=lubridate::wday(posting_date),
          week=lubridate::isoweek((posting_date)),
          monthdate = as.Date(paste("2020", month, '01', sep = "-")),
          fixeddate = as.Date(paste("2020", month, day, sep = "-"))
          )
```

Introduction

This data was downloaded from https://www.h-net.org/jobs and parsed into a relational database using Python and Django. Parser is available here and the most recent version of the database is available here. In total for the last 20 years (starting from 2000 till December 31, 2020) a bit more than 38.000 job ads were posted at H-Net: on average about 1800 position per year.

Till recently about 2/3 of them were professorship positions (from Assistant to Full professor), but for the last 5 years these positions are in decline and in 2020 this amount dropped to below 45%.

Number of posts per year

```
posts<-as_tibble(tbl(con, "hnet_structuredpost") )%>%
  mutate(year=lubridate::year(posting_date),
```

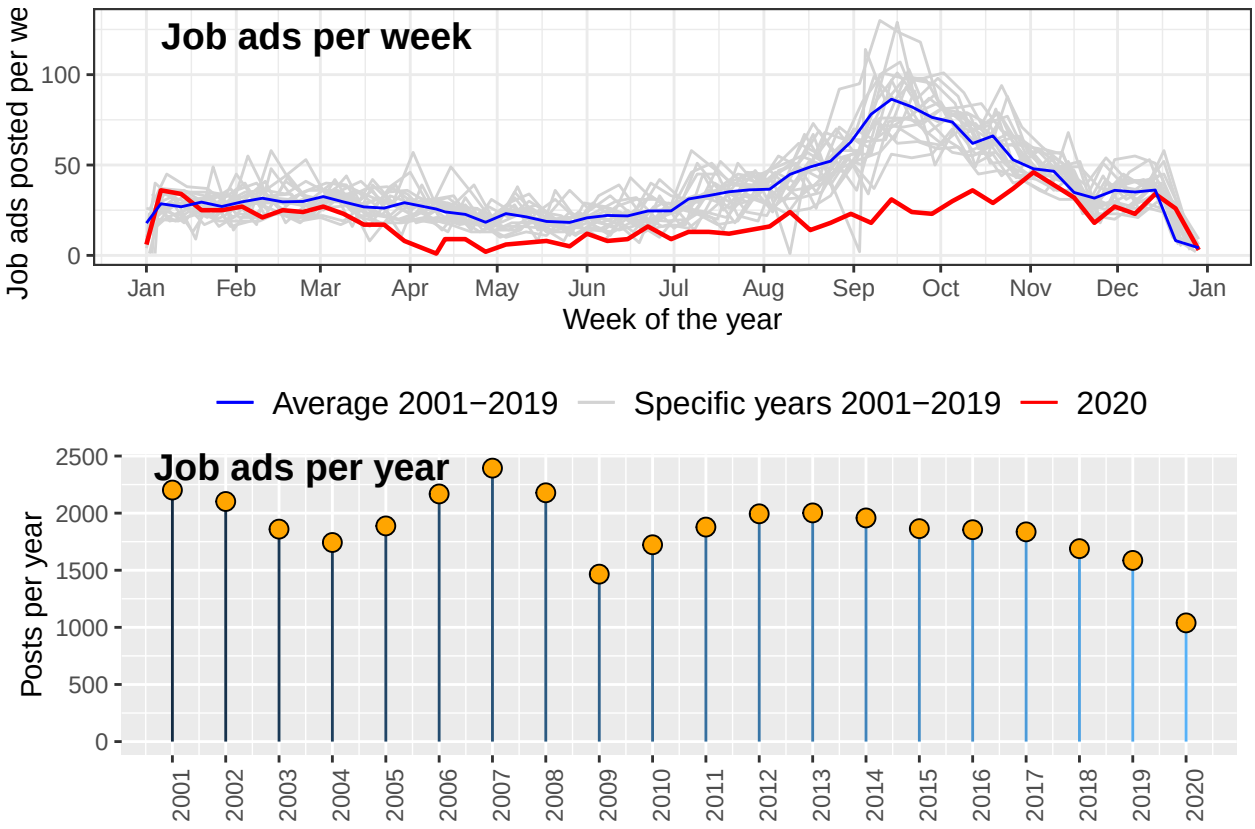
```
         month=lubridate::month(posting_date),
         day=lubridate::day(posting_date),
         weekday=lubridate::wday(posting_date),
         week=lubridate::isoweek((posting_date)),
         monthdate = as.Date(paste("2020", month, '01', sep = "-")),
         fixeddate = as.Date(paste("2020", month, day, sep = "-"))
         ) %>%
  group_by(week,year)%>%
  mutate(weekdate=min(posting_date),
         fixedweek=ymd(format(weekdate, "2020-%m-%d")))%>%
  ungroup()
```

```
theme_ipsum<-function(){list()}

posts_per_year_graph<-posts %>%
  filter(year>2000)%>%
  mutate(monthdate=ymd(glue('{year}-{month}-01')),
         yeardate=ymd(glue('{year}-01-01')))%>%
  group_by(yeardate)%>%
  summarise(posts=n())%>%
  ggplot(aes(x=yeardate, y=posts, fill=yeardate))+
      geom_segment(aes(
      x = yeardate,
      xend = yeardate,
      y = 0,
      yend = posts,
      color=yeardate
    )) +
  geom_point(size=3, fill='orange',color='black', pch = 21 )+
  # geom_label(aes(label=posts, y=posts ),colour='white', size=3)+
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
   theme(axis.text.x = element_text(angle = 90))+xlab('')+ theme(legend.position = "none") +ylab('Posts
```

Posts per week per year

```
averages<-posts %>%
  filter(year>2000,year<2020)%>%
  group_by(week, year) %>%
  summarise(posts=n()) %>%
  group_by(week)%>%
  summarise(averages=mean(posts))


byweek<-posts %>%
  filter(year>2000)%>%
  group_by(year,fixedweek,week)%>%
  summarise(posts= n())


polar_graph<-  full_join(byweek,averages) %>%
  mutate(col=case_when(year==2020~'red',TRUE~'lightgray'),size=case_when(year==2020~0.2,TRUE~0.1))%>%
  ggplot(aes(x=fixedweek, y=posts, group=year)) +
  # ylim(0,NA) +
```

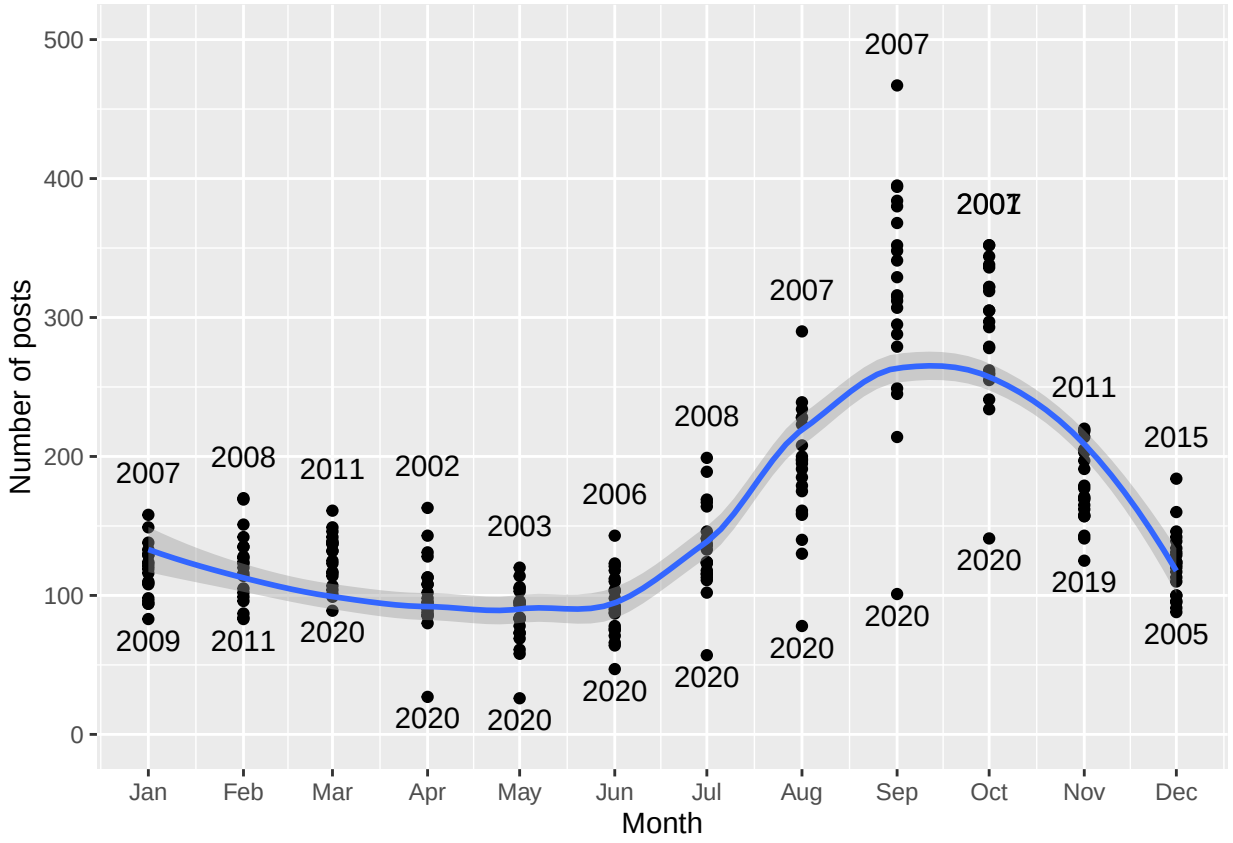
```
  geom_line(aes(color=col,size=size))+
  geom_line(data=full_join(byweek,averages)%>%filter(year==2020),aes(x=fixedweek, y=averages, group=NA,
  # coord_polar(start =0) +
  # scale_color_identity() +
  scale_x_date(date_breaks = "1 month", date_labels = "%b") +
  xlab('Week of the year')+
  ylab('Job ads posted per week')+
    scale_size(range = c(0.5, 0.8))+
  # labs(title = "Academic job ads in history: 2001-2020")+
  scale_colour_manual(name = element_blank(),
         values =c('blue'='blue','red'='red', 'lightgray'='lightgray'), labels = c('Average 2001-2019',
  theme_bw()+
  # hrbrthemes::theme_ipsum()+
  theme(legend.position = 'bottom',
        legend.text=element_text(size=12))+
  theme(
                    plot.margin=unit(c(0,0,0,0), "cm"),
                    )

ggarrange( polar_graph, posts_per_year_graph,
           labels = c("Job ads per week", "Job ads per year"),
           ncol = 1, nrow = 2)
```



Seasonality (posts per month)

```
posts %>%
  group_by(monthdate, year) %>% summarise(n = n()) %>% group_by(monthdate) %>%
```

```
  mutate(lab = case_when(n == max(n) | n == min(n) ~ year),
         vj= case_when(n == max(n) ~ -1.5 , n == min(n) ~ 1.5)) %>%
  ggplot(aes(x = monthdate, y = n)) +
  geom_point(, stat = 'identity') +
  geom_smooth(method = "loess") +
  scale_x_date(date_breaks = "1 month", date_labels = "%b") +
  xlab('Month') +
  ylab('Number of posts') + geom_text(aes(label = lab, vjust = vj))+
  ylim(0,500)
```



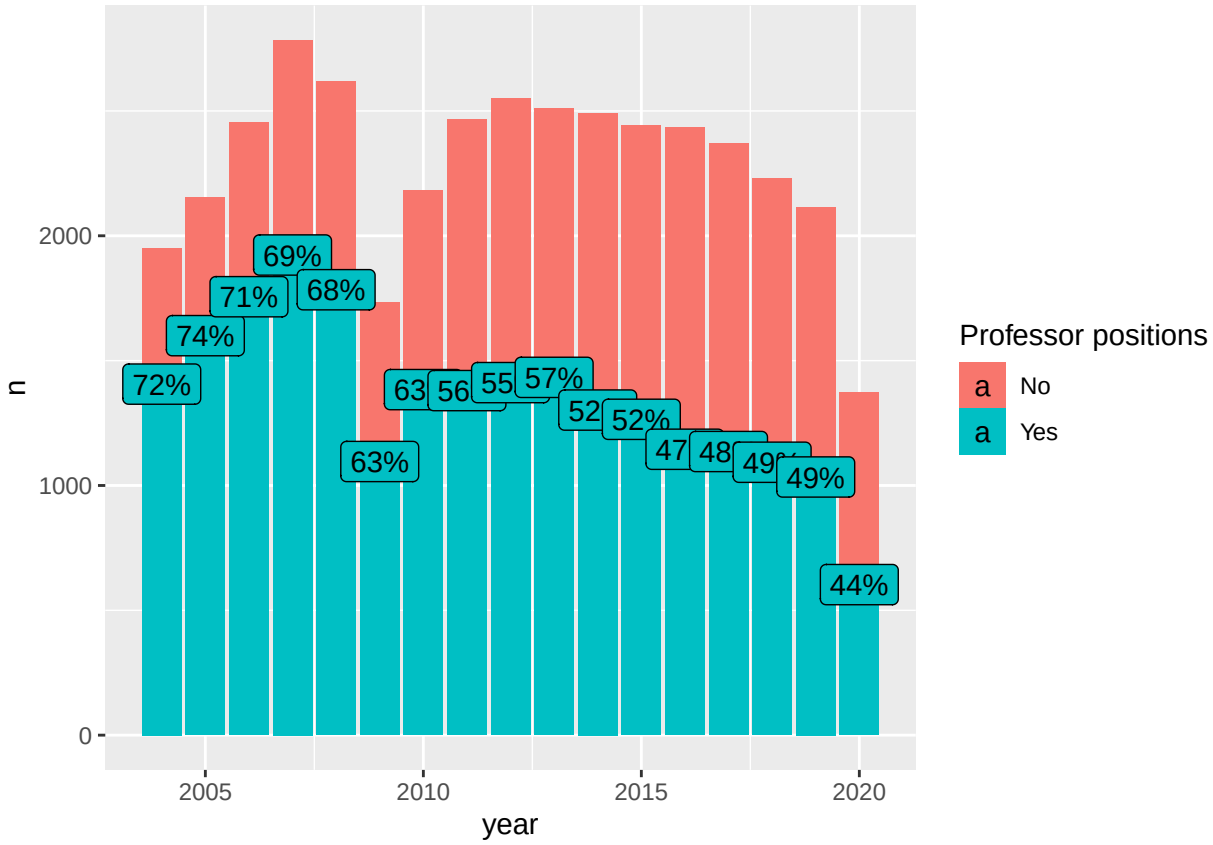
Dynamics of professorship positions vs other positions

In the plot below percentages is a share of professorship positions in total number of job ads posted per year.

For the last 10 years share of professorship positions are in decline. It substantially fell in 2020 to less than a half. We might believe it's because other positions are posted more often, but in fact total number of posts per year is also in decline (we of course cannot guarantee that it's not due to falling popularity of H-Net website as an advertising platform)

```
position_tab %>%
  mutate(prof=str_detect(position, 'Professor')) %>%
  mutate(prof = if_else(prof,  "Yes", "No")) %>%
  group_by(prof, year)%>%
  tally() %>%
  filter(year>2003)%>%
  group_by(year)%>%
  mutate(freq=if_else(prof=='Yes', n/sum(n), NA_real_))%>%
  ggplot(aes(x=year, y=n, fill=prof))+
  geom_col()+
```

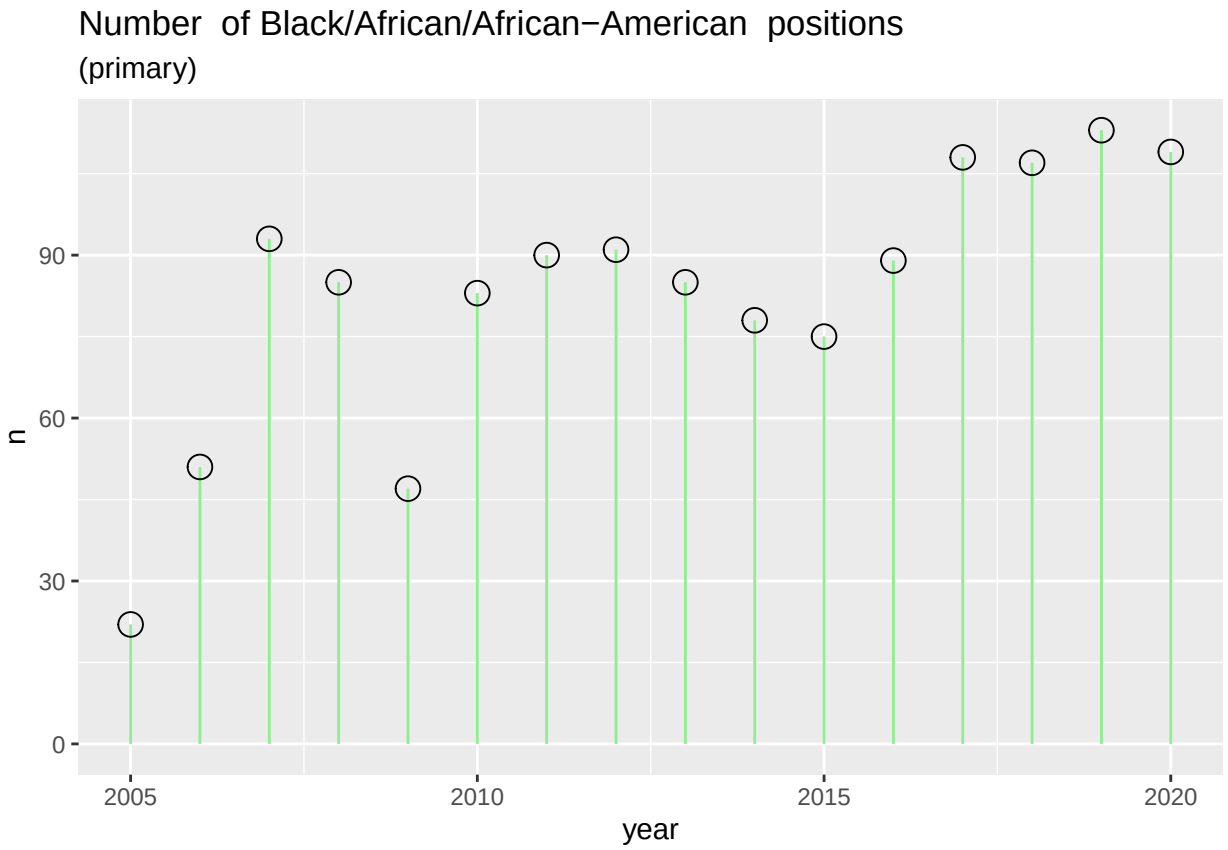
```
  geom_label(aes(label=label_percent(accuracy=1)(freq)))+
  labs(fill='Professor positions')+
  theme_ipsum()
```



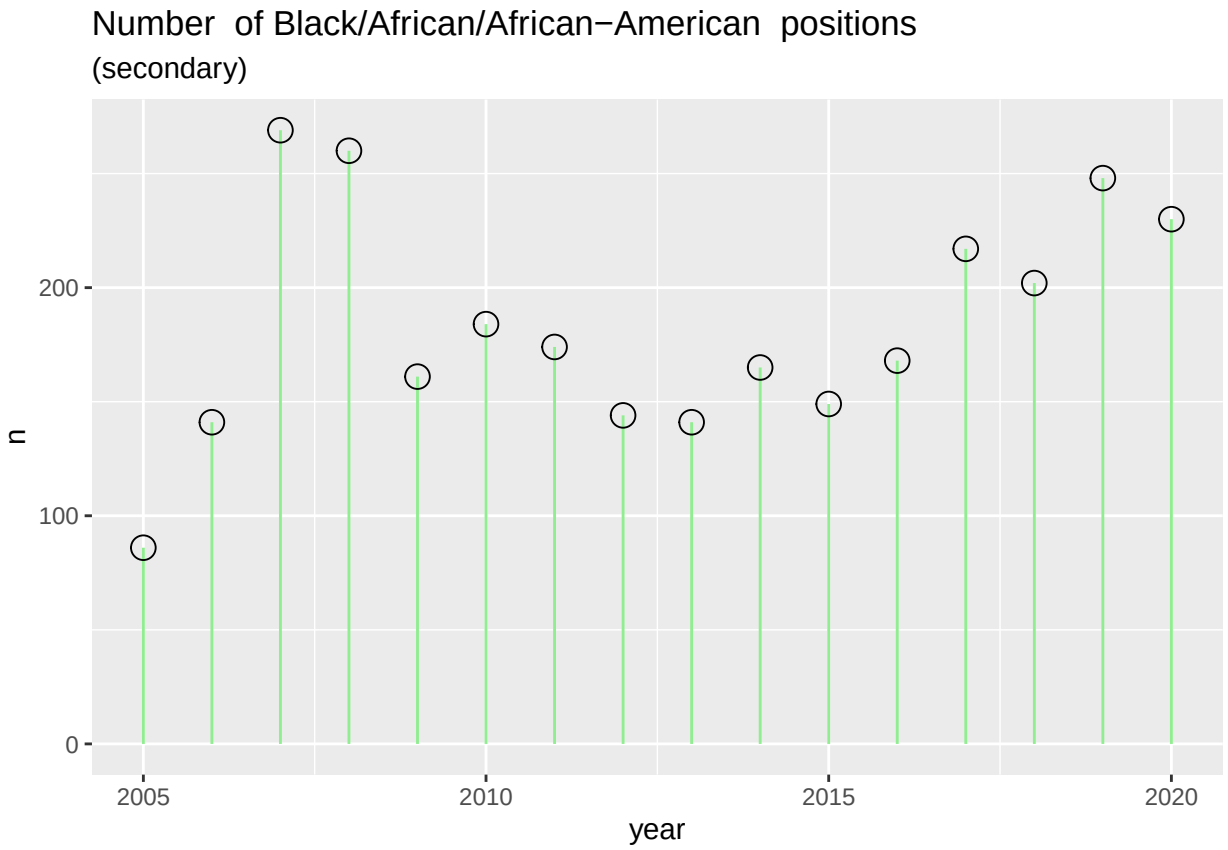
Black studies/Race studies positions

```
black_obj<-list(filter_str='frican|Race|lack',
                title='Black/African/African-American', col='navyblue')


graph_builder <- function(data, obj, ordertype, rel=F) {
  if (rel){
    reltitle <- 'Share'
    custom_scale<-  list(scale_y_continuous(labels = label_percent(accuracy=1)),
                         geom_line()
                         )
    segment_color<-"orange"

  } else {
    reltitle <- 'Number'
    custom_scale<-list()
    segment_color<-"lightgreen"
  }

  subchain <- . %>%
    group_by(year)%>%
```

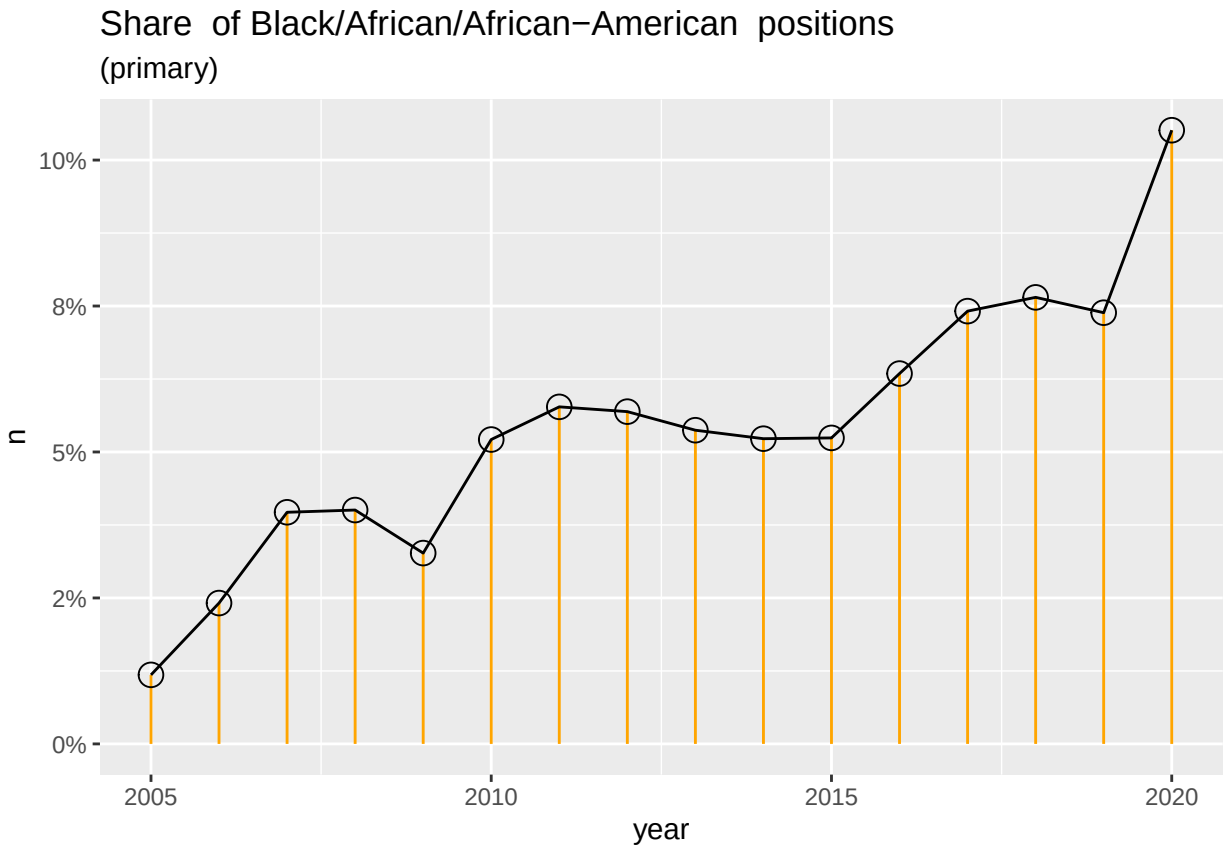
```
  mutate(n=n/sum(n))

  data %>%
    mutate(black = str_detect(category, obj$filter_str)) %>%
    group_by(black,year) %>%
    tally() %>%
   {if(rel) subchain(.) else .} %>%
    filter(black == T) %>%
    ggplot(aes(x = year,  y = n)) +
    geom_segment(aes(
      x = year,
      xend = year,
      y = 0,
      yend = n
    ), color = segment_color) +
    geom_point(
      aes(x = year, y = n),
      fill = NA, #obj$col
      colour = 'black',
      size = 4 ,
      pch = 21
    ) +
    custom_scale +
    ggtitle(glue("{reltitle}  of {obj$title}  positions"), subtitle=glue('({ordertype})'))
}

graph_builder(primary_cat_tab, black_obj, 'primary', rel=F)
```



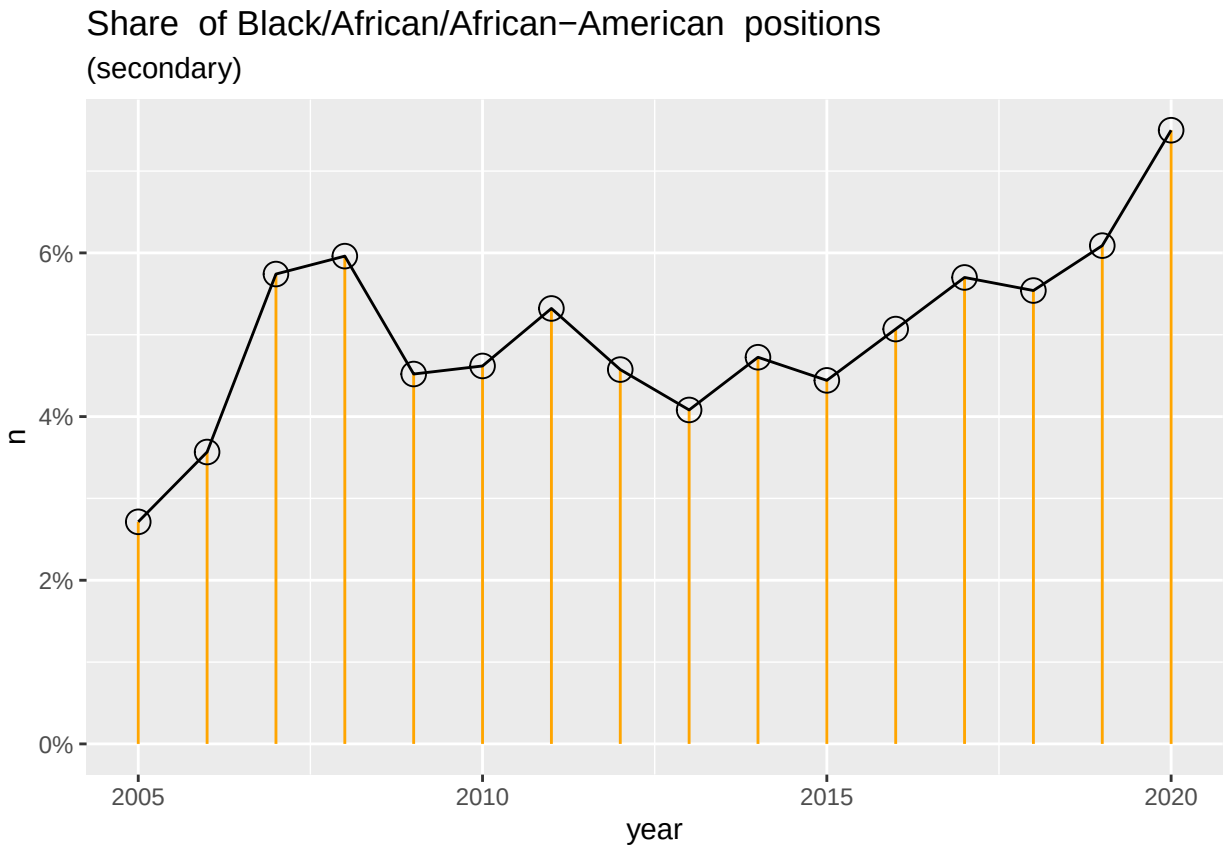

```
graph_builder(secondary_cat_tab, black_obj, 'secondary', rel=F)
```




```
graph_builder(primary_cat_tab, black_obj, 'primary', rel=T)
```

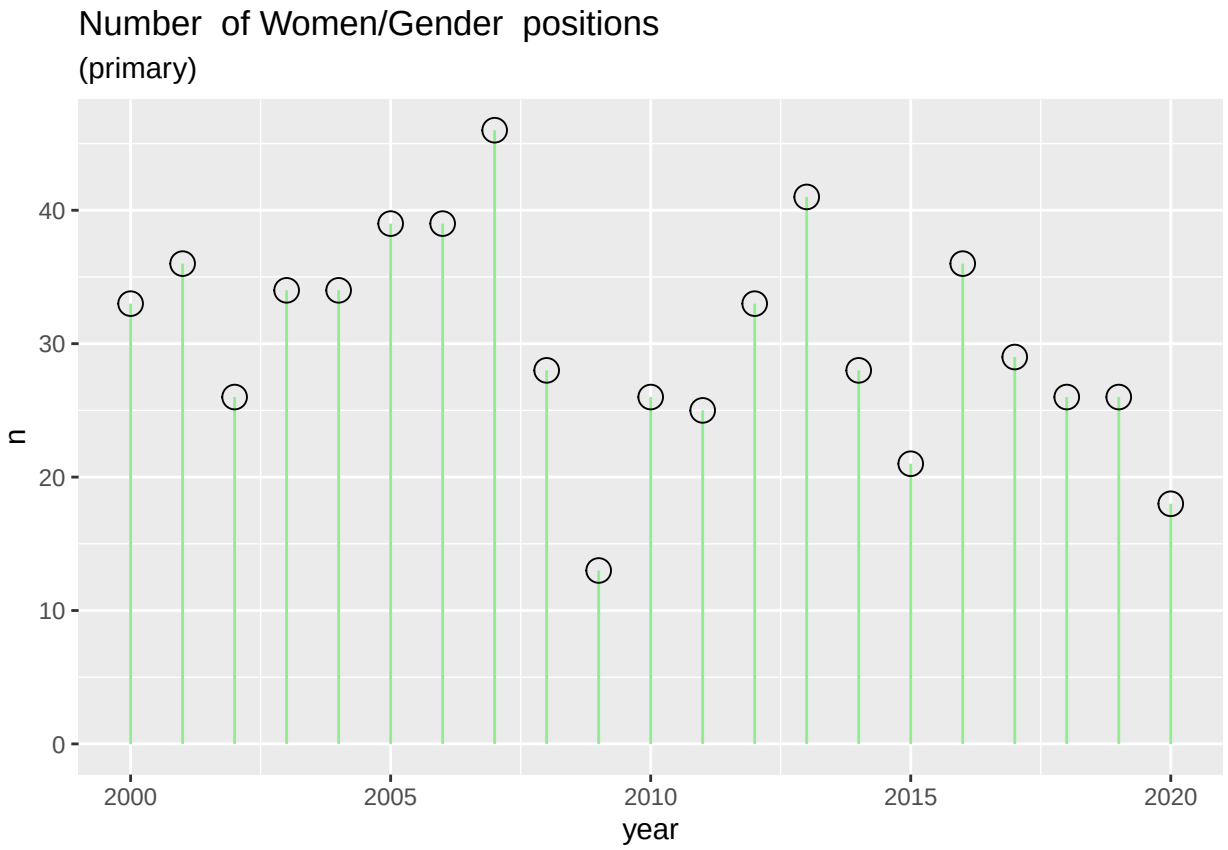



```
graph_builder(secondary_cat_tab, black_obj, 'secondary', rel=T)
```

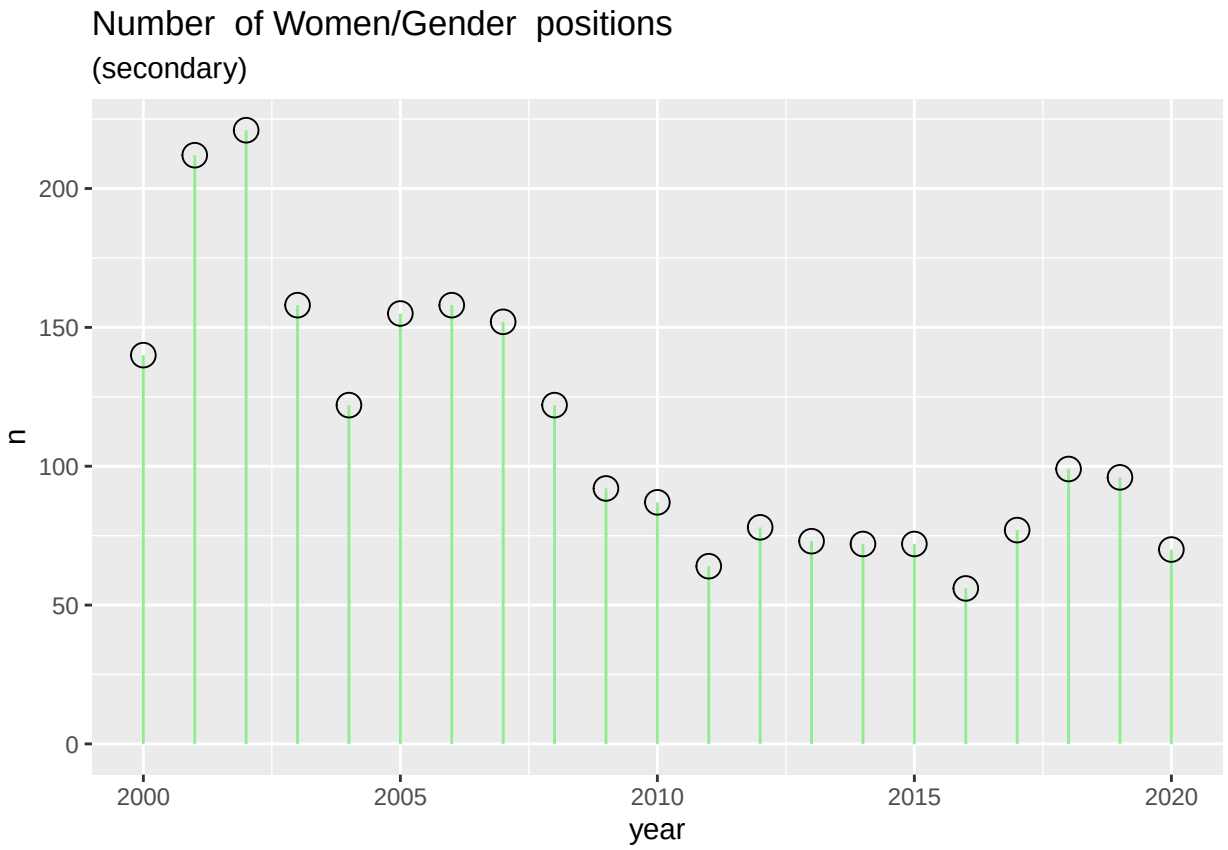


Feminism/Women studies positions

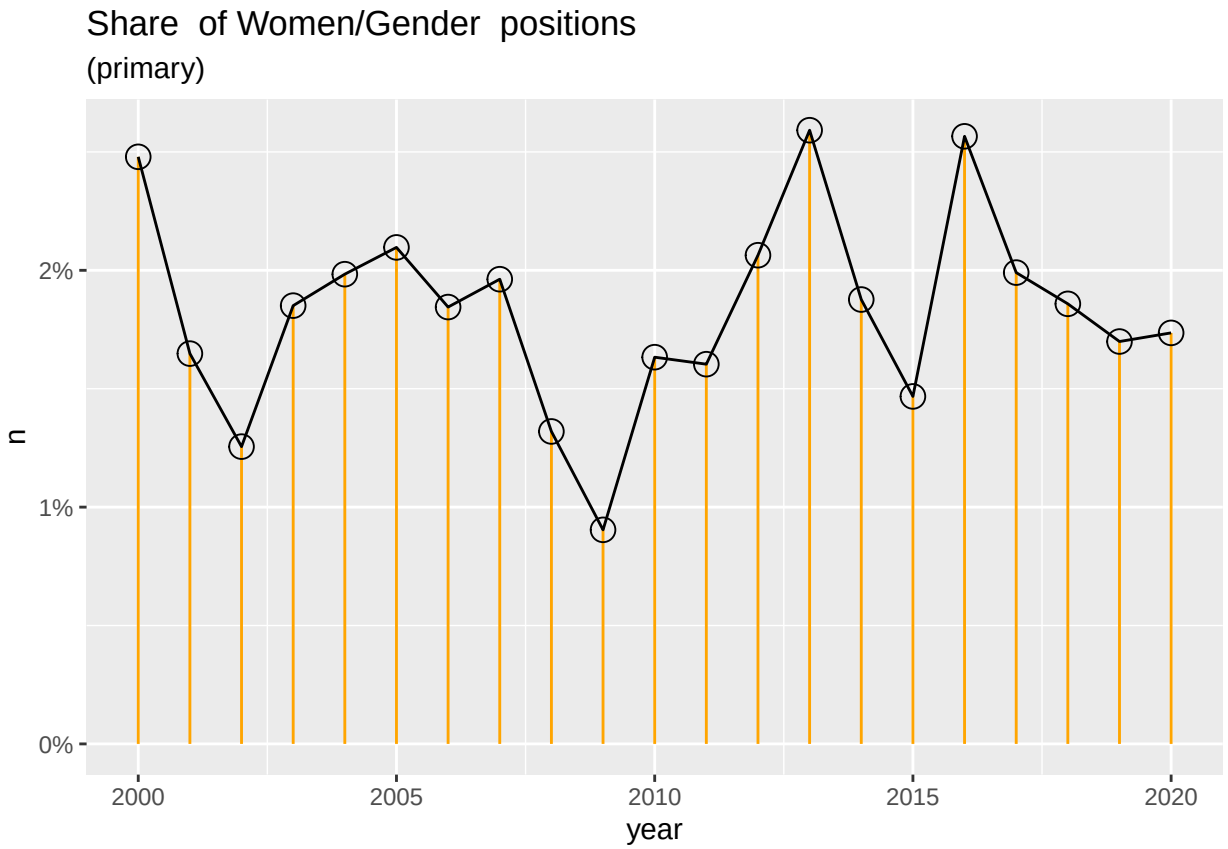
```
women_obj<-list(filter_str='wom|Wom|fem|Fem|Gend|gend',title='Women/Gender', col='pink')
graph_builder(primary_cat_tab, women_obj, 'primary', rel=F)
```



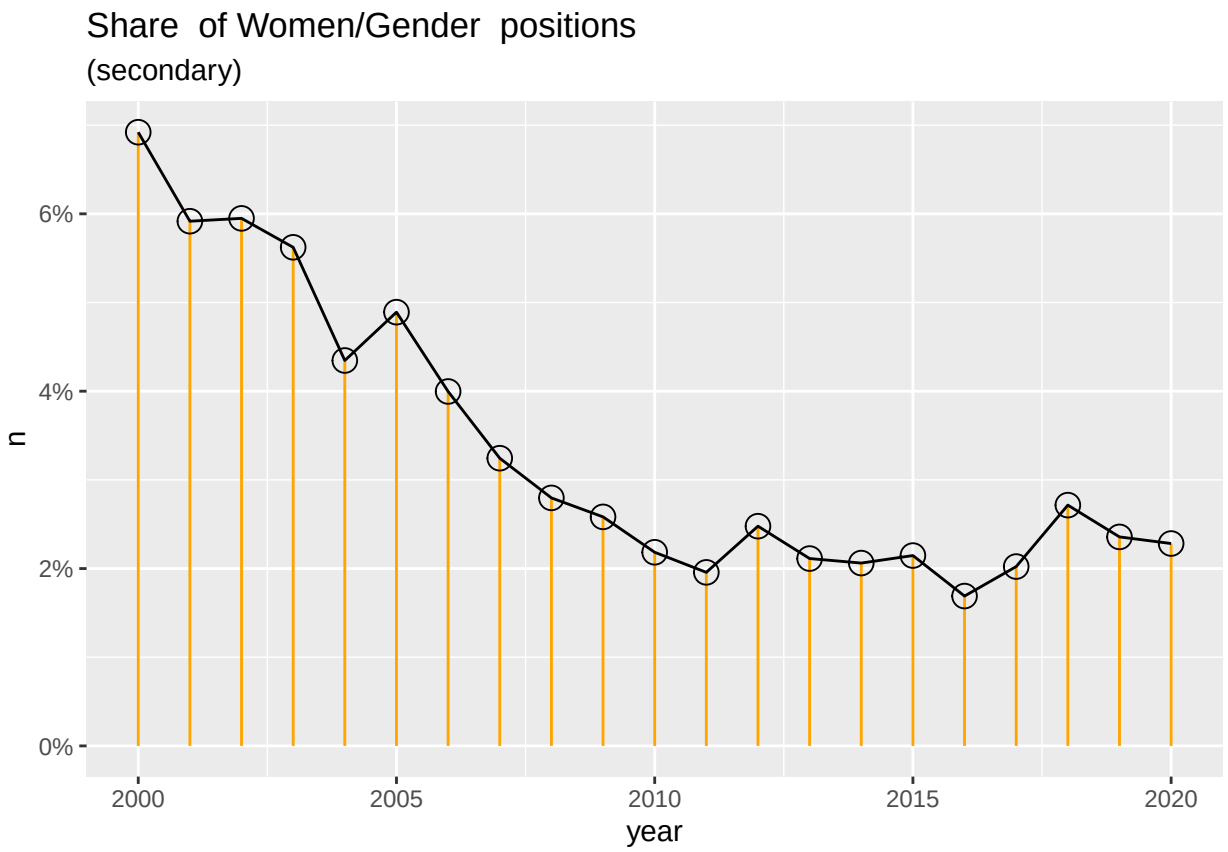

```
graph_builder(secondary_cat_tab, women_obj, 'secondary', rel=F)
```




```
graph_builder(primary_cat_tab, women_obj, 'primary', rel=T)
```



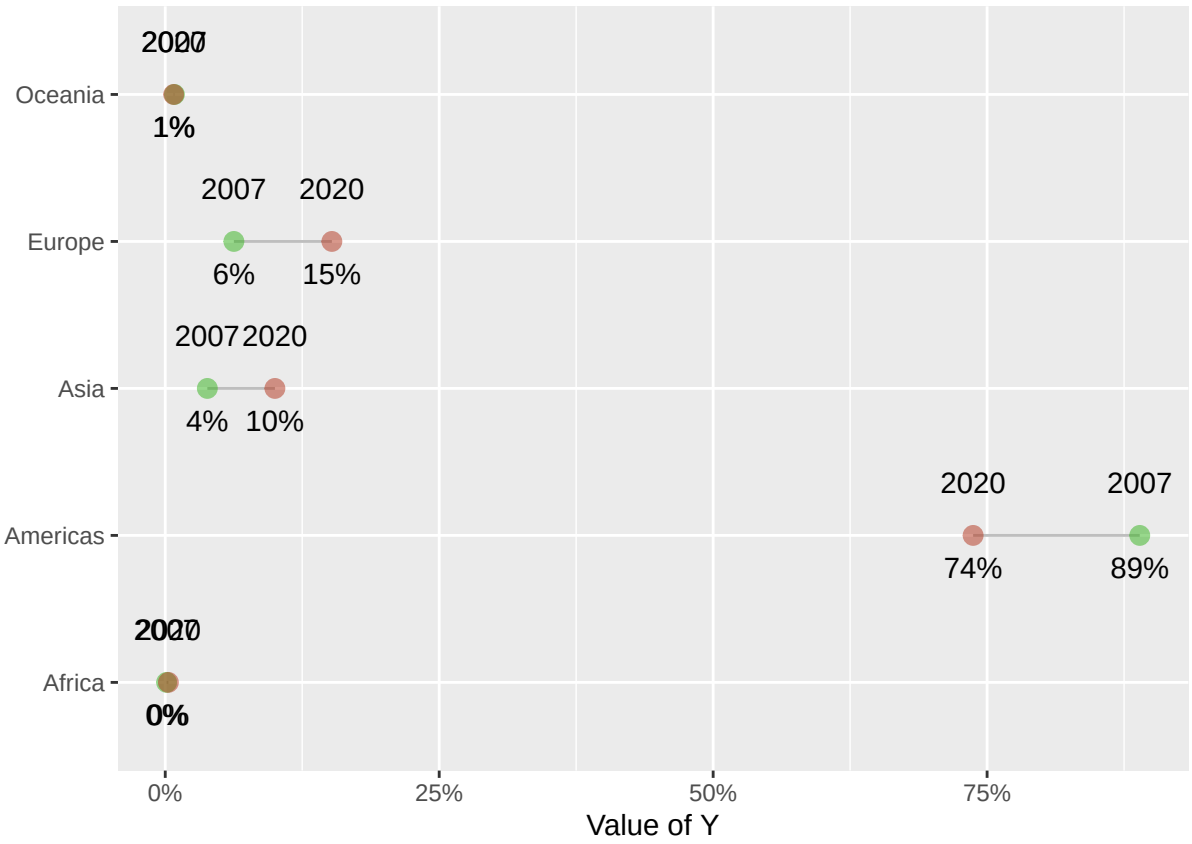

```
graph_builder(secondary_cat_tab, women_obj, 'secondary', rel=T)
```



Positions posted by continent and country

```
# RUN THE FOLLOWING JUST ONCE BECAUSE IT REQUESTS A BUNCH OF THINGS FROM GEOCODING API (*TWICE*); THE R
#
# p_load('ggmap','revgeo')
#
# api_key<-'YOUR API KEY'
# register_google(key = api_key)


# loc2country <- function(loc) {
#   latlon <-
#     ggmap::geocode(location = c(loc),
#                    output = "latlon",
#                    source = "google") %>% as_tibble()
#   revgeo::revgeo(
#     longitude = latlon$lon,
#     latitude = latlon$lat,
#     provider = 'google',
#     API = api_key,
#     output = 'hash',
#     item = 'country'
#   )$country
# }

# alllocs<-alllocs %>%
```

```
#   arrange(location)%>%
#   mutate(country=loc2country(location))

# alllocs$continent <- countrycode(sourcevar = alllocs$ country,
#                                  origin = "country.name",
#                                  destination = "continent")

# write.csv(alllocs, 'alllocs.csv')
alllocs<- read.csv('alllocs.csv')
alllocs <- alllocs %>% mutate(country=recode(country, `Yugoslavia`='Kosovo'))
left_join(posts, alllocs) %>%
  filter(year%in%c(2007,2020))%>%
  group_by(continent, year)%>%
  tally() %>%
  group_by(year)%>%
  mutate(freq=n/sum(n))%>%
  pivot_wider(names_from=year,
              values_from=freq,
              id_cols=continent) %>%
   mutate(mean = rowMeans(across(where(is.numeric))))%>%
  arrange(mean)%>%
  mutate(x=factor(continent),
         value1=`2007`,
         value2=`2020`)%>%
  ggplot()+
  geom_segment( aes(x=x, xend=x, y=value1, yend=value2), color="grey") +
  geom_point( aes(x=x, y=value1), color=rgb(0.2,0.7,0.1,0.5), size=3 ) +
  geom_point( aes(x=x, y=value2), color=rgb(0.7,0.2,0.1,0.5), size=3 ) +
  coord_flip()+
  theme_ipsum() +
  theme(
    legend.position = "none",
  ) +
  xlab("") +
  ylab("Value of Y")+
  geom_text(aes(x=x,   y=value1, vjust=2, label=label_percent(accuracy=1)(value1)))+
  geom_text(aes(x=x,   y=value2, vjust=2, label=label_percent(accuracy=1)(value2)))+
    geom_text(aes(x=x,   y=value1, vjust=-2, label='2007'))+
  geom_text(aes(x=x,   y=value2, vjust=-2, label='2020'))+
  ylim(0,1.2)+
  scale_y_continuous(labels = label_percent())
```


2007
2020
Oceania
1%
2007
2020
Europe
6%
15%
2007
2020
Asia
4%
10%
2020
2007
Americas
74%
89%
2007
2020
Africa
0%
0%
25%
50%
75%
Value of Y