Sociology, Demograpy, and Ecocnomics Presidential Ages and Sex over time

Supplementary Materials

Setting the Workspace

First, we need to set up our working environment. This is a generic preamble and includes packages that might or might not be needed for this exact data visualization.

The first thing we do is clear our workspace to start from scratch and instigate garbage collection to improve performance.

```r
rm(list = ls()) # Remove Previous Workspace
gc(reset = TRUE) # Garbage Collection
```

The `pacman` package is a really great package management system for R. This will install and load pacman.

```r
# Install and Load pacman for easier package loading and installation
if (!require("pacman", character.only = TRUE)){
  install.packages("pacman", dep = TRUE)
  if (!require("pacman", character.only = TRUE))
    stop("Package not found")
}
```

We will need to load the packages to make the visualization. We create a vector of the packages we need in an object called `pkgs`.

```r
# Libraries
pkgs <- c(
  "tidyverse",      # Tidyverse
  "data.table",     # Data Management/Manipulation
  "openxlsx",       # Microsoft Excel Files
  "stringi",        # Character/String Editor
  "stringr",        # Character/String Editor
  "reshape2",       # Data Management/Manipulation
  "scales",         # Number formatting
  "cowplot",        # Plot Grids
  "IDPmisc",        # Quality na.rm
  "ggrepel",        # Repel Labels
  "lemon"           # Axis manipulation
)
```

We then call the `p_install()` function from the pacman package. This piece of code with install missing packages and will only run if at least one package is missing. The `pkgs` object is then removed from the workspace to keep it clean.

```
if(!sum(!p_isinstalled(pkgs))==0){
  p_install(
    package = pkgs[!p_isinstalled(pkgs)],
    character.only = TRUE
  )
}

# Load the packages
p_load(pkgs, character.only = TRUE)
rm(pkgs)
```

Data

Data for this visualization comes from a variety of sources. These sources are listed in the two columns of the data file: “Source1” and “Source2.” The table below shows some sample output.

```
set.seed(1)
head(sample_n(dat, 6))[,1:8]

## # A tibble: 6 x 8
##   ElectionYear President Association Sex   Birthyear BAyear Sourc~ Source2
##          <dbl> <chr>     <chr>       <chr>     <dbl>  <dbl> <chr>  <chr>
## 1         1908 William ~ ASA         Male       1840   1863 Wikip~ <NA>
## 2         1945  Kimball~ ASA         Male       1893   1915 ASA.n~ <NA>
## 3         2006  Cynthia~ ASA         Fema~      1933     NA Wikip~ <NA>
## 4         1987 Gary S. ~ AEA         Male       1930     NA Wikip~ <NA>
## 5         1956 Joseph A~ PAA         Male       1902   1926 http:~ <NA>
## 6         1982 Gardner ~ AEA         Male       1915     NA Wikip~ <NA>
```

The majority of association president’s birth years and bachelor’s graduation years are either chronicled on Wikipedia or chronicled on the respective association’s website in some form.

I manually entered these ages into a spreadsheet called `presidents.xlsx` that can be found online at github.

Calculation of Age

The calculation/estimation of age during presidential year is rather straightforward. For association presidents with a verified birth year, I simply calculated the age as `ElectionYear - BirthYear`. Out of the 314 presidents across the three associations, direct age is calculable in 291 presidents.

Of the 23 remaining presidents without birth years, I estimated their age as `ElectionYear - (BAYear - 22)`. The average age for presidents where I have both birth year and the year their bachelors was attained is `mean(dat$BAyear-dat$Birthyear, na.rm=T)` making age 22 a good approximation.

I could not locate either birth years nor bachelor graduation years for two presidents – Karen O. Mason (PAA president 1997) and Jacob S. Siegel (PAA president 1980). These two presidents were excluded from the analysis.

The following code chunk reads the `presidents.xlsx` worksheet into the R environment and performs the subsequent age calculations. The variable `AGE` selects the estimated age (`AgeAtPres_est`) if the direct age (`AgeAtPres_direct`) is not available.

The variable `l` is also used to label the youngest presidents for each association by sex.

```
dat <- read.xlsx("DATA/presidents.xlsx", sheet = 2) %>%
  mutate(
    # Calculating the Direct Age
         AgeAtPres_direct = ElectionYear - Birthyear,
    # Estimating the age based on BA Year completion.
         AgeAtPres_est = ElectionYear - (BAyear-22),
    # Selecting BA Age if Direct Age is not present.
         AGE = if_else(is.na(AgeAtPres_direct), AgeAtPres_est,
AgeAtPres_direct)) %>%
  # Grouping to find youngest pres by sex in each Association. Used for
labeling.
  group_by(Association, Sex) %>%
  mutate(l = if_else(AGE ==  min(AGE, na.rm =T),
                     paste0(President, " (", ElectionYear, "): ", AGE),
                     "")) %>%
  ungroup()
```

Female Presidents

The figure also includes a summary of female presidents for each assocation. I used the reported pronoun (he/she) for each association president as it appeared in their online material. To include this in the figure I create a new object and `tally()` the number of presidents by sex for each association and calculate the simple percentage that are female then create a string (`label`) that will print the association, the number of female presidents, and the percentage. Finally, I pull just the `label` variable into a vector and paste all three associations into a single string.

```
# New object to calculate the % of presidents who are female.
fems <- dat %>%
  group_by(Association, Sex) %>%
  tally() %>% # Getting the # by Association and sex
  group_by(Association) %>%
  mutate(percentage = percent(n / sum(n)), # Calculating the % of each
Associationg
         label = paste0(Association,
                        " Female Presidents: ",
                        n,
                        " (", percentage,")\n")) %>% # Creating the label
  ungroup() %>%
  filter(Sex == "Female") # Selecting just Female rows.

fems2 <- paste(pull(fems, label)) # Turning the vector into a single text string.
```

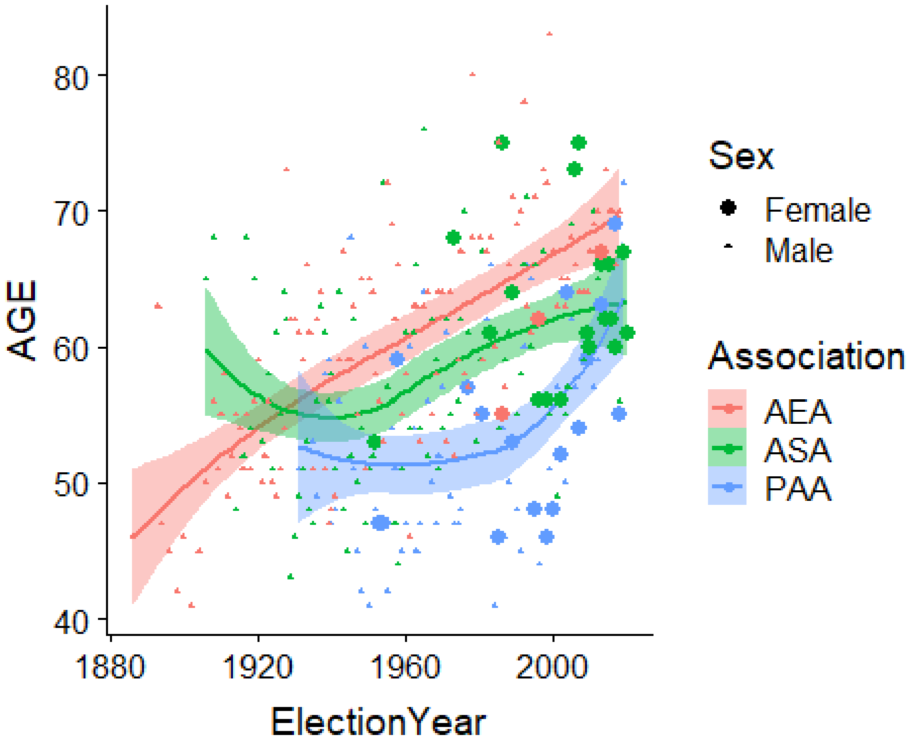
The Figure

We can then create the full figure.

We create a `ggplot` object based on the `dat` data file where the x-axis is the Election Year, the y-axis is Age. We color each observation using the Association variable and we label certain points using the l variable.

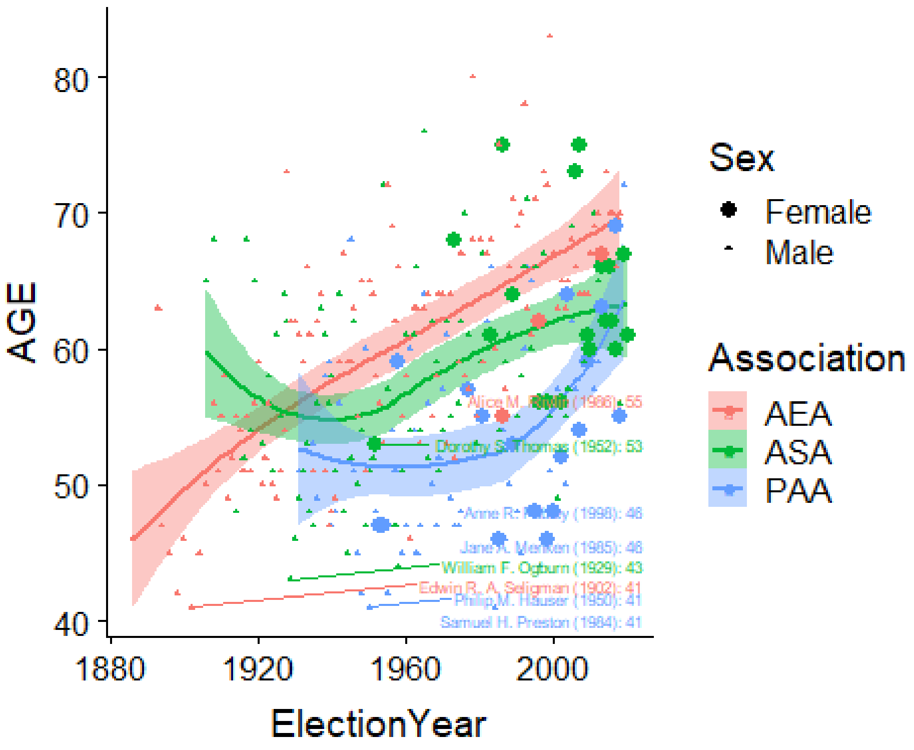
`geom_smooth` creates a locally estimated scatterplot smoothing (LOESS), sometimes called local regression or moving regression. We set the span to 1 to have 100% smoothing. Since the figure is a scatterplot, `geom_point` populates the figure with points and then we manually rescale the points based on sex, making female presidents larger – and thus more prominent. The LOESS is plotted first since ggplot layers graphics on top of each other. This ensures that the points and everything else sits on top of the LOESS lines.

```
(a <- ggplot(data = dat, aes(x = ElectionYear, y = AGE, color = Association, label = l)) +
  geom_smooth(span = 1, aes(fill = Association)) +
  geom_point(aes(shape = Sex, size = Sex)) +
  scale_size_manual(values = c(2.5,.75)))
```



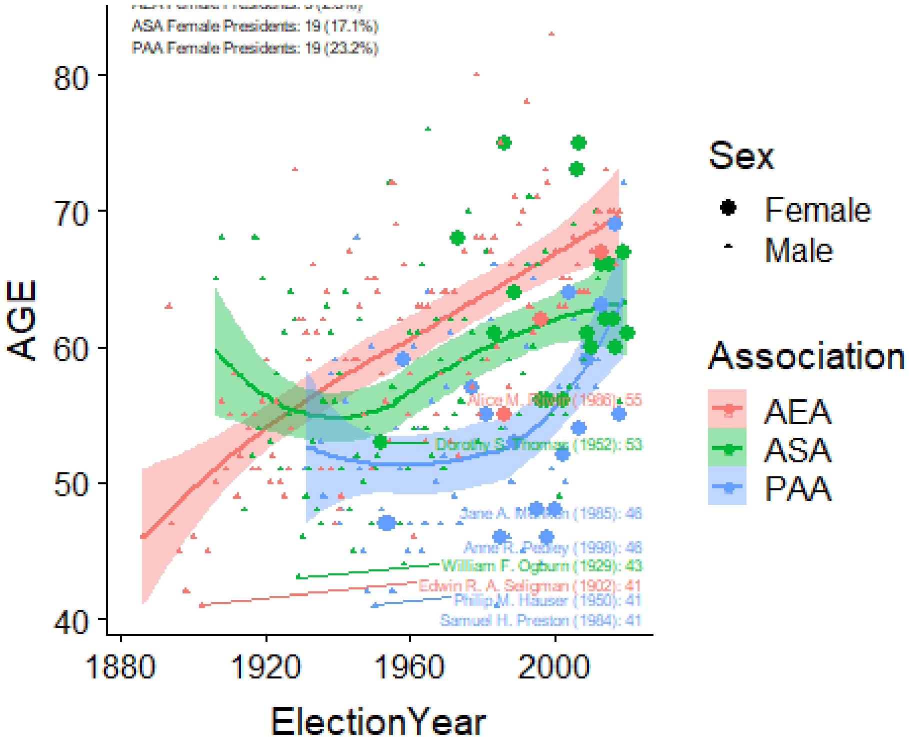
Next we add the labels for only the youngest presidents. This is done by subsetting the data where the label is not equal to a blank expression (`!l == ""`). Since the labels are all on the far right of the figure, I "nudge" the labels along the x-axis such that they are moved x years to the right where x is `2060 - ElectionYear` for each observation with a label. Thus Edwin R. A. Seligman will get a larger nudge (2060-1902) than Samuel H. Presont (2060-1984).

```
(a <- a + geom_text_repel(data = subset(dat, !l == ""),
                  size=2,
                  nudge_x = 2060 - subset(dat, !l == "")$ElectionYear,
                  direction    = "y"))
```



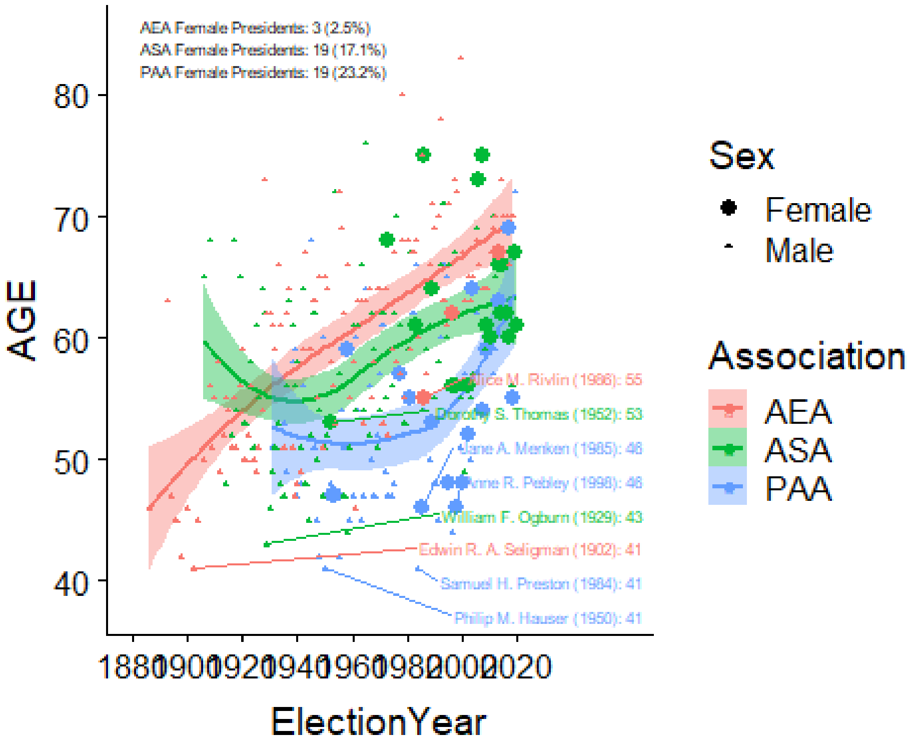
Next, I create an annotation layer for the female president summary in the top-left corner of the figure. I set this to the coordinates (1883, 83) and using `hjust=0` the annotation is left-aligned.

```
(a <- a +  annotate("text", x=1883, y = 83,
                    label = paste(fems2, sep="", collapse=""),
                    size = 2,
                    hjust = 0))
```



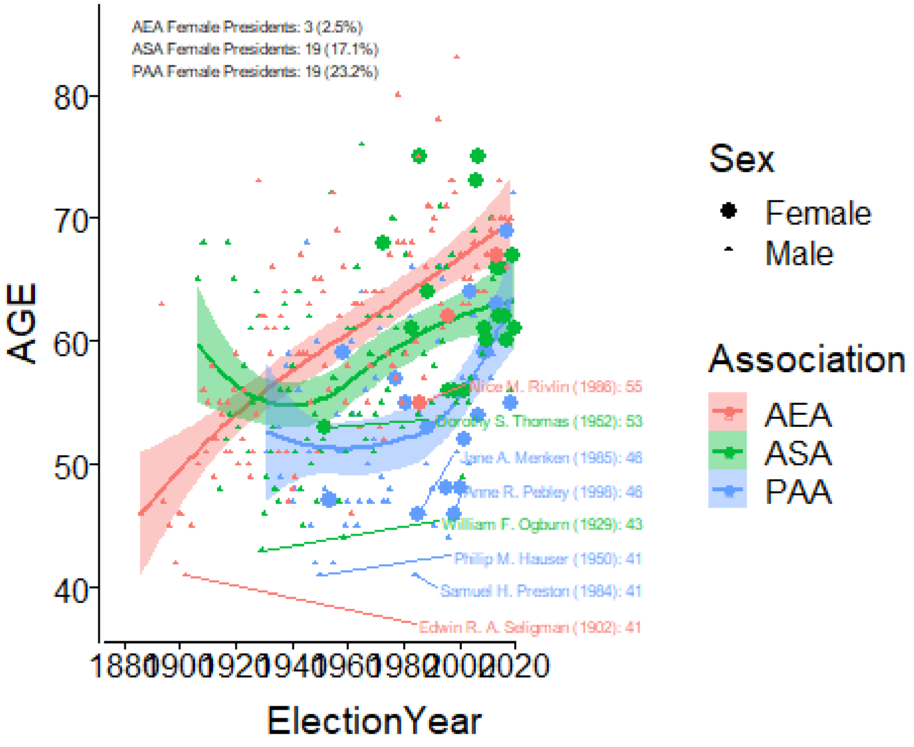
Finally, I clean the figure up. I set the y-axis limits to ages 38 and 85, so the annotation layer is fully shown. The labeled presidents are sitting on top of the points in the previous graphs and I really want them out to the right. This is accomplished by setting the x-axis scale limits to 180 and 2060 and setting my axis tick marks from 1880 to 2020 in 20-year intervals.

```
(a <- a + ylim(38,85) +
  scale_x_continuous(breaks = seq(1880, 2020, 20), limits = c(1880,2060)))
```



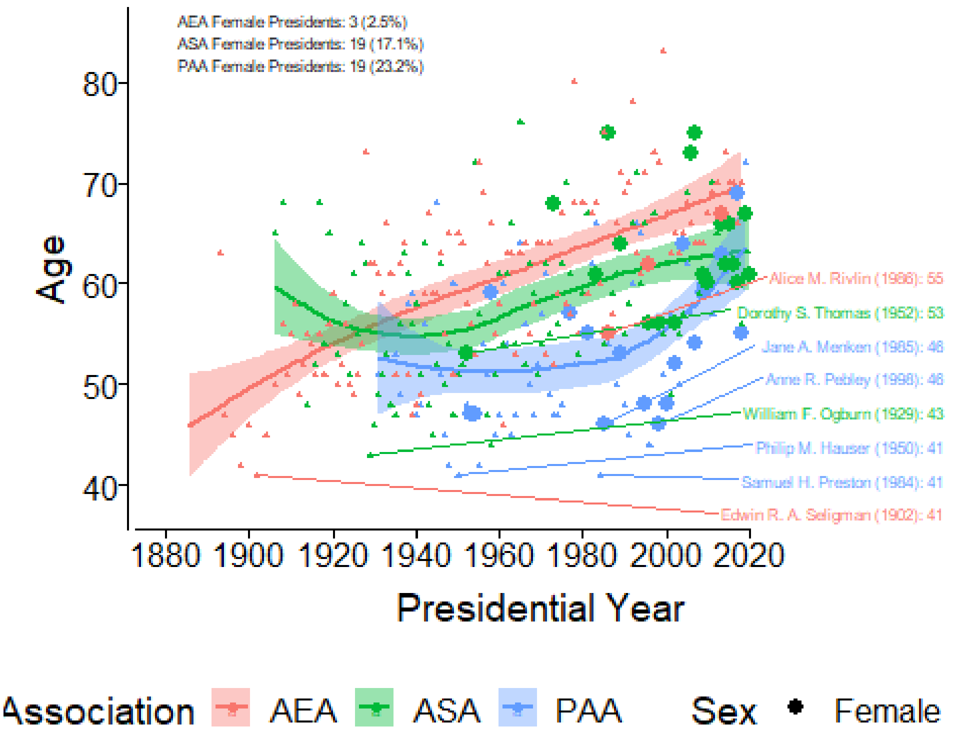
I don’t particularly like the way the x-axis continues from 2020-2060, so we “cap” the axis to the bottom so the labels appear to float outside of the figure area.

```
a +  coord_capped_cart(bottom='right')
```



I then change the legend from the right side of the figure to the bottom, and rename my axes for better presentation. Putting it all together. This forms the upper-panel of the figure. For simplicity sake we call this plot “a”.

```
(a<-ggplot(data = dat, aes(x = ElectionYear, y = AGE, color = Association,
label = l)) +
  geom_smooth(span = 1, aes(fill = Association)) +
  geom_point(aes(shape = Sex, size = Sex)) +
  scale_size_manual(values = c(2.5,.75)) +
  geom_text_repel(data = subset(dat, !l == ""),
                  size=2,
                  nudge_x = 2060 - subset(dat, !l == "")$ElectionYear,
                  direction    = "y") +
  annotate("text", x=1883, y = 83,
                   label = paste(fems2, sep="", collapse=""),
                   size = 2,
                   hjust = 0) +
  ylim(38,85) +
  scale_x_continuous(breaks = seq(1880, 2020, 20), limits = c(1880,2060)) +
  coord_capped_cart(bottom='right') +
    theme(legend.position="bottom") +
  labs(x = "Presidential Year",
       y = "Age"))
```

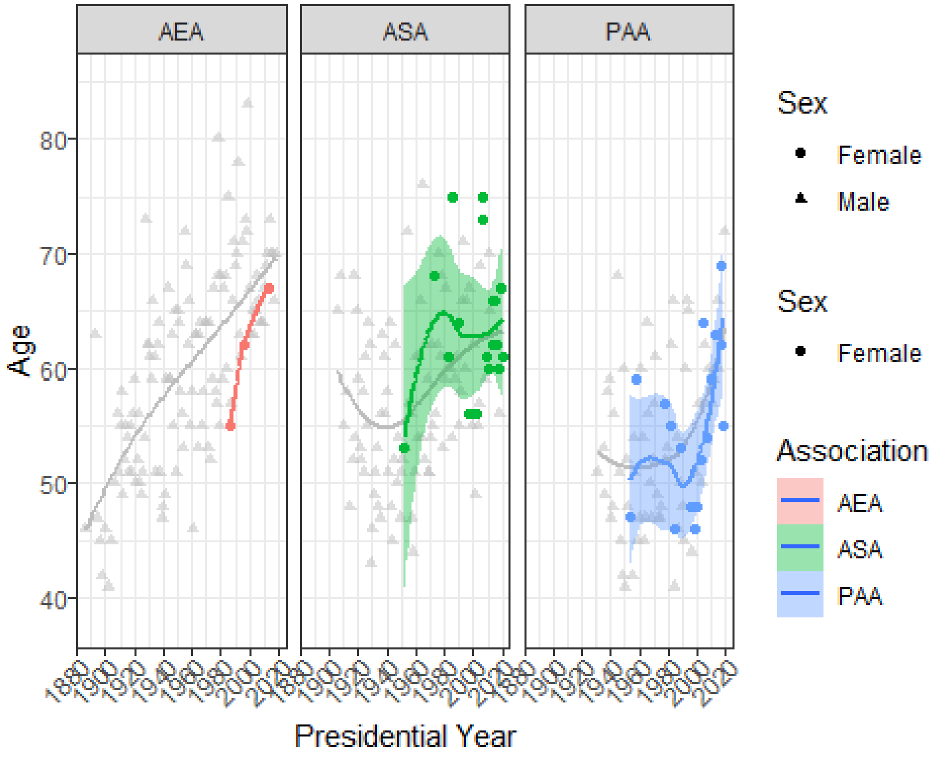


The bottom part of the panel uses much of the same code, but utilizes the `facet_wrap` command to highlight each association. The first thing we need to do is create a new dataframe that contains only women presidents.

```
dat2 <- dat %>% filter(Sex == "Female")
```

The code is largely the same, but I have commented out the pieces of code from panel (a) that are not necessary for the second panel. I've also added a duplicate `geom_smooth()` and `geom_point()` in gray. These two additional functions will plot the gray LOESS and points on each facet. `facet_wrap(~ Association)` creates three panels (one for each academic association). And finally, to ensure that the labels are readable, `theme(axis.text.x = element_text(angle = 45, hjust=1))` turns the x-axis labels 45-degrees.

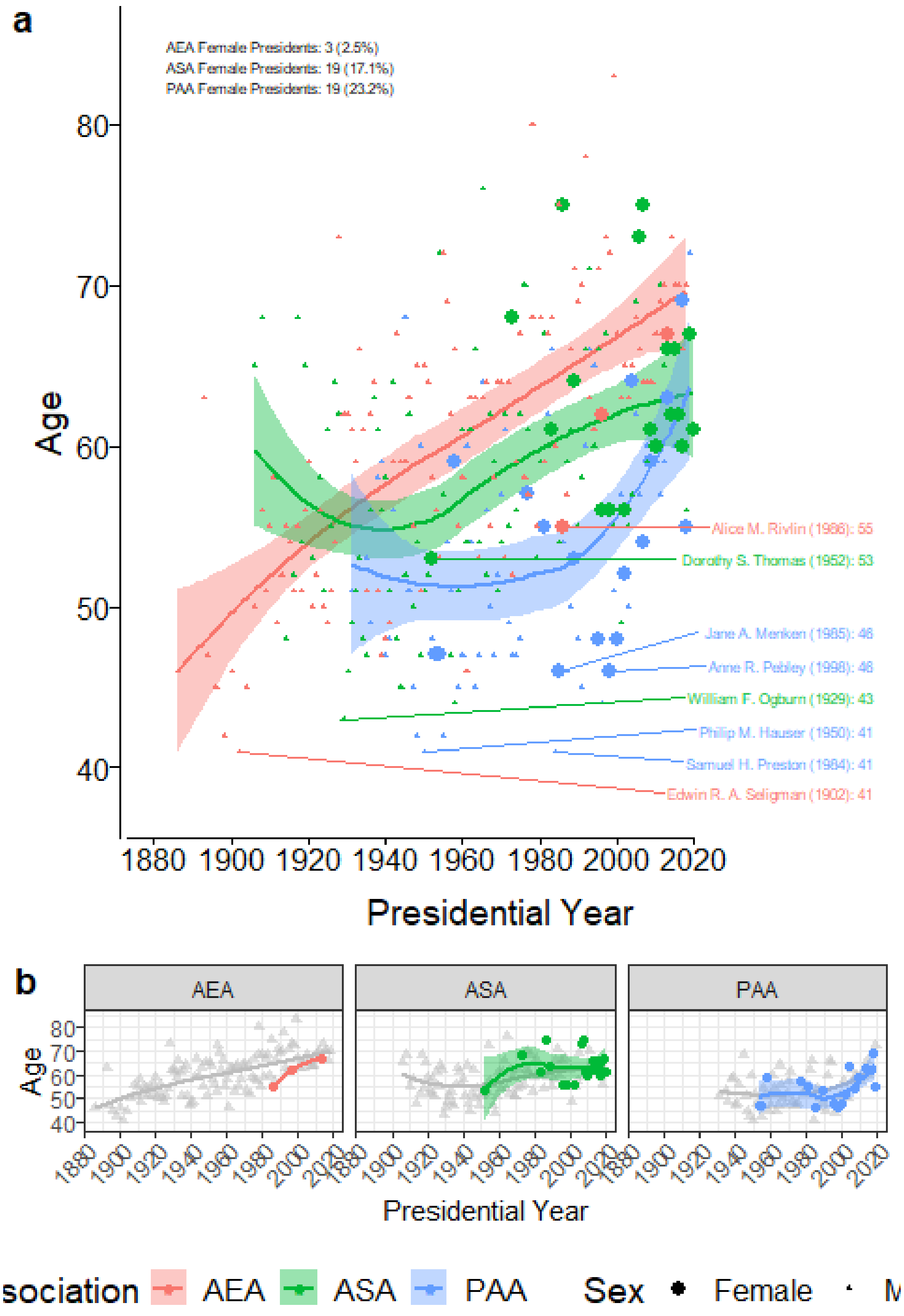
```
(b<- ggplot(data = dat2, aes(x = ElectionYear, y = AGE, color = Association,
label = l)) +
  geom_smooth(data=dat, span = 1, fill = "gray", color = "gray", se= FALSE) +
  geom_point(data = dat, aes(x = ElectionYear, y = AGE, shape = Sex), color =
"grey", alpha = 0.5) +
  geom_smooth(span = 1, aes(fill = Association)) +
  geom_point(aes(shape = Sex, size = Sex)) +
  facet_wrap(~ Association) +
  scale_size_manual(values = 1.5) +
  # geom_text_repel(data = subset(dat, !l == ""),
  #                 size=2,
  #                 nudge_x = 2060 - subset(dat, !l == "")$ElectionYear,
  #                 direction    = "y") +
  # annotate("text", x=1883, y = 83, label = paste(fems2, sep="",
collapse=""), size = 2, hjust = 0) +
  ylim(38,85) +
  scale_x_continuous(breaks = seq(1880, 2020, 20), limits = c(1880,2025),
expand= c(0,0)) +
  coord_capped_cart(bottom='right') +
  guides(color = FALSE) +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust=1)) +
  labs(x = "Presidential Year",
       y = "Age"))
```



Finally, to put both panels together, we use the `cowplot` package which is a fantastic package for multi-panel figures. We need to use `legend <- get_legend(a)` to get the legend from the top panel. The figure is actually 3 panels – one for the main figure (a), one for the facets (b), and one for the legend. We use `theme(legend.position="none")` for the top two panels since our legend will be its own unlabeled panel.

```
legend <- get_legend(a)

plot_grid(a+ theme(legend.position="none"),
          b+ theme(legend.position="none"),
          legend,
          ncol=1, rel_heights = c(1,0.3, 0.1),
          labels = c("a","b"))
```


a
AEA Female Presidents: 3 (2.5%)
ASA Female Presidents: 19 (17.1%)
PAA Female Presidents: 19 (23.2%)
Age
80
70
60
50
40
Alice M. Rivlin (1986): 55
Dorothy S. Thomas (1952): 53
Jane A. Menken (1985): 46
Anne R. Pebley (1998): 46
William F. Ogburn (1929): 43
Philip M. Hauser (1950): 41
Samuel H. Preston (1984): 41
Edwin R. A. Seligman (1902): 41
1880 1900 1920 1940 1960 1980 2000 2020
Presidential Year
b
AEA
ASA
PAA
Age
80
70
60
50
40
1880 1900 1920 1940 1960 1980 2000 2020
Presidential Year
sociation AEA ASA PAA
Sex ● Female ▴ M

Et voilà!