Years in office, promotion incentives, and fiscal behavior

Siwei Dai

December 1, 2021

```
library(tidyverse)
library(estimatr)
library(modelsummary)
options(modelsummary_format_numeric_latex = "mathmode")
datapath <- './ps_data/FullData.csv'
df <- as_tibble(read.csv(datapath))
biodatapath <- './ps_data/BiographicalData.csv'
biodf <- as_tibble(read.csv(biodatapath))
fiscdatapath <- './ps_data/fiscal.dta'
fiscdf <- haven::read_dta(fiscdatapath)
```

Data description

This project is based on two datasets. The first dataset is the "Chinese Political Elite Database" (CPED), compiled by Junyan Jiang (accessed at https://www.junyanjiang.com/data.html). This dataset includes the biographical and career information of all mayors and municipal party secretaries from 2000 to 2015, provincial governors and party secretaries (1995 - 2015).

Another dataset documents county-level fiscal revenue and expenditure during 1994 to 2008. The dataset comes from Xu Xu (2021) who purchased and digitized the original "Fiscal Statistics of Cities and Counties" compiled by the Budget Department of the Ministry of Finance of China. The original paper concerns the effect of implementing digital surveillance programs on public security spending.

In this project, I merged the two datasets to produce a leader-year dataset that captures the

Data cleaning

Before runnning analyses, I tidied the data.

On the CPED dataset, I first created the key variables needed for analysis; I then converted the CPED dataset to a county-year level dataset.

On the county-level fiscal data, I dealt with certain coding discrepancies within the data. In some cases, missing data is coded `NA` and in other cases, it's coded 0. I recoded the missing data points to `NA`.

Finally, I merged the datasets together. As some politicians may only start their position closer to December, they may not have sufficient control over the fiscal expenditure and income that particular year. For politicians who came into office between July to December, I coded the first year of their tenure the next year.

```
df_tidy <- df %>%
    # translate the variables into English
    # exper_num refers to the career stage
    rename('identifier' = 用户编码,
           'name' = 姓名,
           'job' = 标志位,
           'position_ori' = 级别,
           'exper_num' = 经历序号,
           'start_date' = 起始时间.YYYY.MM.DD.,
           'end_date' = 终止时间..YYYY.MM.DD..,
           'prefectural_code' = 二级关键词编码,
           'provincial_code' = 一级关键词编码,
           'position_code' = 职务一级关键词编码) %>%
    select(identifier, name, job, position_ori, exper_num, start_date, end_date,
           prefectural_code, provincial_code, position_code) %>%
    # impute prefectural code from provincial code if the prefectural code is missing due to being a pr
    mutate(prefectural_code = ifelse(is.na(prefectural_code), provincial_code, prefectural_code)) %>%
    # recode the position from strings into numeric; larger the number, higher the position
    mutate('position_numeric' = case_when(position_ori == '无级别' ~ 0,
                                          position_ori == '小于副处' ~ 1,
                                          position_ori == '副处' ~ 2,
                                          position_ori == '正处' ~ 3,
                                          position_ori == '副厅' ~ 4,
                                          position_ori == '正厅' ~ 5,
                                          position_ori == '副部' ~ 6,
                                          position_ori == '正部' ~ 7,
                                          position_ori == '副国' ~ 8,
                                          position_ori == '正国' ~ 9)) %>%
    # Code the term length
    mutate('term_length' = (as.numeric(as.Date(end_date) - as.Date(start_date)))/365,
           'start_year' = ifelse(as.numeric(format(as.Date(start_date), '%b')) >= 7, as.numeric(format(a
           'end_year' = as.numeric(format(as.Date(end_date), '%Y'))) %>%
    # Create a `leader` variable to code if the official is a municipal/provincial governor or party se
    mutate(governor = job %in% c(" 市长", " 省长"),
           party_secretary = job %in% c(" 市委书记", " 省委书记"),
           leader = governor | party_secretary) %>%
    # Create promotion or demotion
    group_by(name) %>%
    # rank the experiences in the temporal order
    arrange(exper_num, by_group = TRUE) %>%
    mutate(dem = (lag(position_numeric) > position_numeric),
           # or being moved to an inconsequential position
           prom = (lead(position_numeric) > position_numeric))
```

Converting to county-year dataset

```
df_tidy_year <- df_tidy %>%
    # filter out NAs, otherwise there would be errors in mutating new variables
    filter(!is.na(start_year) & !is.na(end_year) & !is.na(prefectural_code)) %>%
    mutate('year' = map2(start_year, end_year, `:`)) %>%
    unnest(cols = c(year)) %>%
```

```
    # filter out the experiences after the fiscal data is available
    filter(year >= 1994) %>%
    # Create two dummies
    mutate('term_year' = year - start_year + 1,
           'first_year' = ifelse(year == start_year, 1, 0),
           'last_year' = ifelse(year == end_year, 1, 0))
```

Merge the dataset with spending data and biographical information

```
# merge with fiscal data using regional code and year
# first, filter only include those who have ascended to the position of making decisions
biodf_tidy <- biodf %>%
    rename('bio_identifier' = X,
           'birth_date' = 出生日期.YYYY.MM.DD.,
           'discipline_date' = '查处. 罢黜时间',
           'discipline_cause' = '查处. 罢黜原因') %>%
    select(bio_identifier, birth_date, discipline_date, discipline_cause) %>%
    mutate('birth_year' = as.numeric(format(as.Date(birth_date), '%Y')),
           'disciplined' = (discipline_cause == '贪污腐败' | discipline_cause == '违纪'),
           'discipline_year' = as.numeric(format(as.Date(discipline_date), '%Y')))

fiscdf_tidy <- fiscdf %>%
    filter(!is.na(admcode) & admcode %in% unique(df_tidy$prefectural_code)) %>%
    # Code the income and expenditure observations that are 0 to NA
    mutate(income_ttl = ifelse(income_ttl == 0, NA, income_ttl),
           exp_ttl = ifelse(exp_ttl == 0, NA, exp_ttl),
           exp_shbz = ifelse(exp_shbz == 0, NA, exp_shbz))

df_merged <- fiscdf_tidy %>%
    left_join(df_tidy_year, by = c('admcode' = 'prefectural_code',
                                   'year' = 'year')) %>%
    left_join(biodf_tidy, by = c('identifier' = 'bio_identifier')) %>%
    mutate('income_log' = log(income_ttl),
           'exp_log' = log(exp_ttl),
           'social_exp_log' = log(exp_shbz),
           'edu_exp_log' = log(exp_jy),
           'age' = year - birth_year,
           'post_discipline_time' = (year > discipline_year)) %>%
    # filter out the leaders
    filter(leader == 1) %>%
    arrange(admcode, year)
```

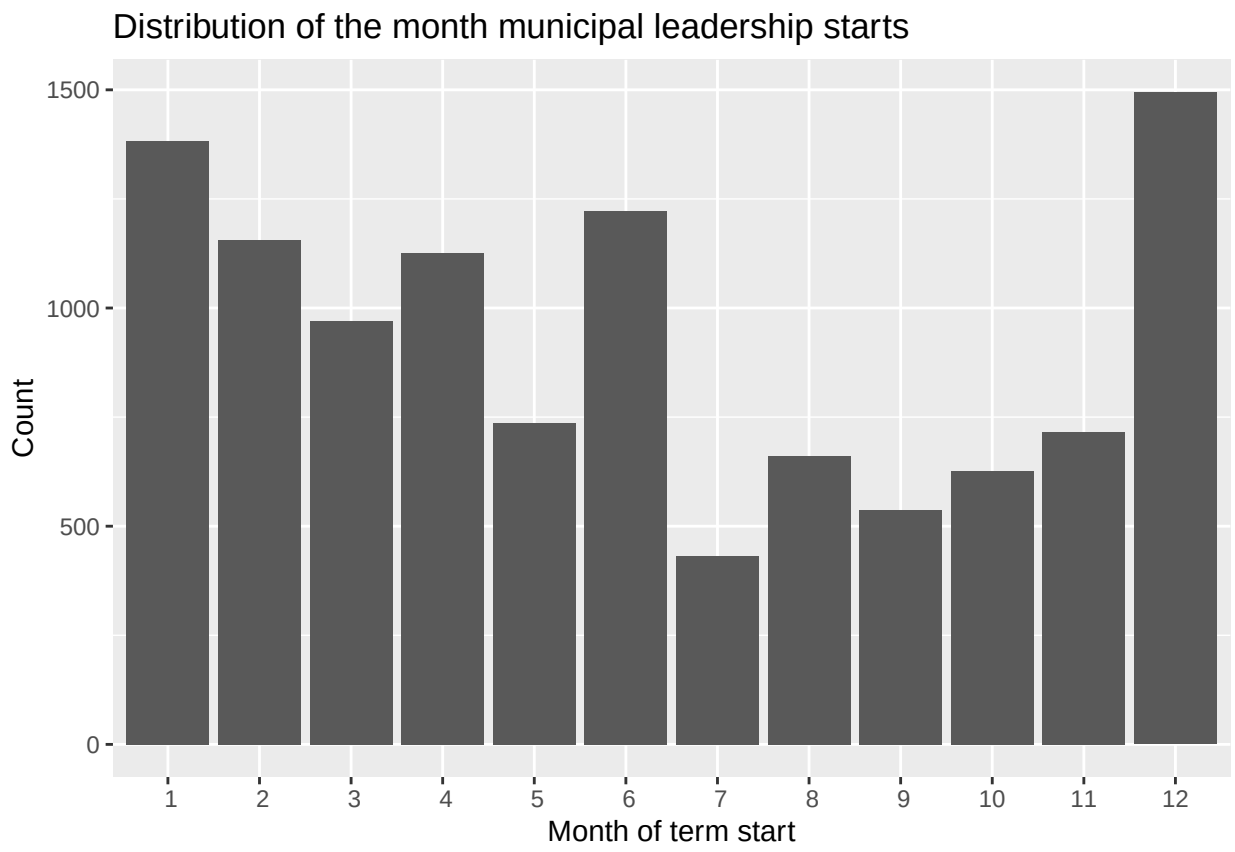
Summary Statistics & Visualizations

How many observations

Distribution of start and end date

One concern with merging fiscal data with political cycles is whether the leader can still decide

```
fig_distofstartdate <- df_merged %>%
    filter(!is.na(format(as.Date(start_date), '%b'))) %>%
    ggplot(aes(x = format(as.Date(start_date), '%b'))) +
    geom_bar() +
    labs(x = 'Month of term start',
         y = 'Count',
         title = "Distribution of the month municipal leadership starts")
fig_distofstartdate
```




```
ggsave("./FinalPJ/fig_distofstartdate.png")
```

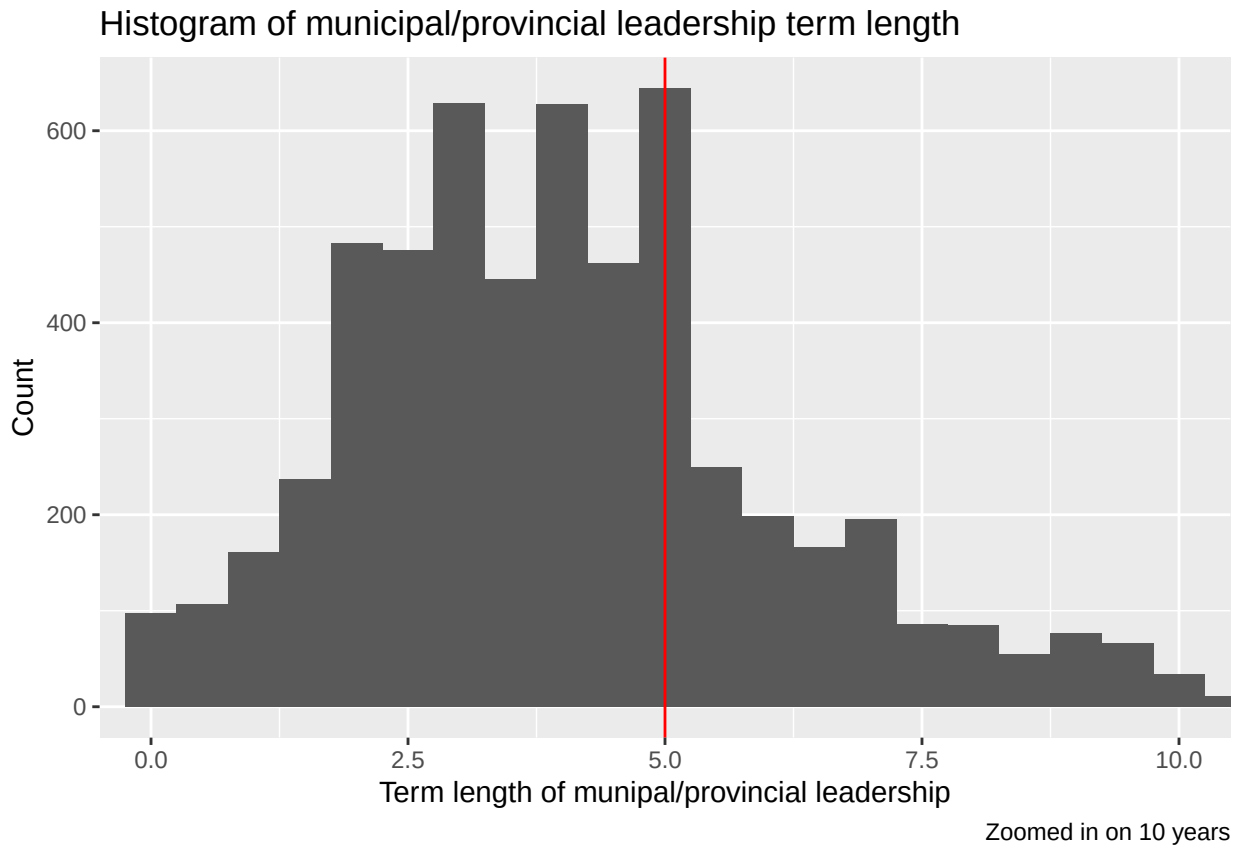
```
## Saving 6.5 x 4.5 in image
```

Average Term length

There is a formal term limit for Chinese mayors and provincial governors: 5 years. Yet this formal term limit is not strictly enforced. As shown in the following figure, most of Chinese mayors are promoted or transferred to other positions before reaching the 5-year term limit(the red vertical line marks the fifth year in office). This means that politicians face substantial uncertainty on how long they could stay at a particular position.

```
fig_tenuretime <- df_merged %>%
    group_by(name) %>%
    filter(governor)  %>%
```

```
    ggplot(aes(x = term_length)) +
    geom_histogram(binwidth = 0.5) +
    coord_cartesian(xlim = c(0, 10)) +
    geom_vline(color = "red", xintercept = 5) +
    # Change x coordinates
    labs(x = "Term length of munipal/provincial leadership",
         y = "Count",
         title = "Histogram of municipal/provincial leadership term length",
         caption = "Zoomed in on 10 years")
fig_tenuretime
```




```
ggsave("./FinalPJ/fig_tenuretime.png")
```

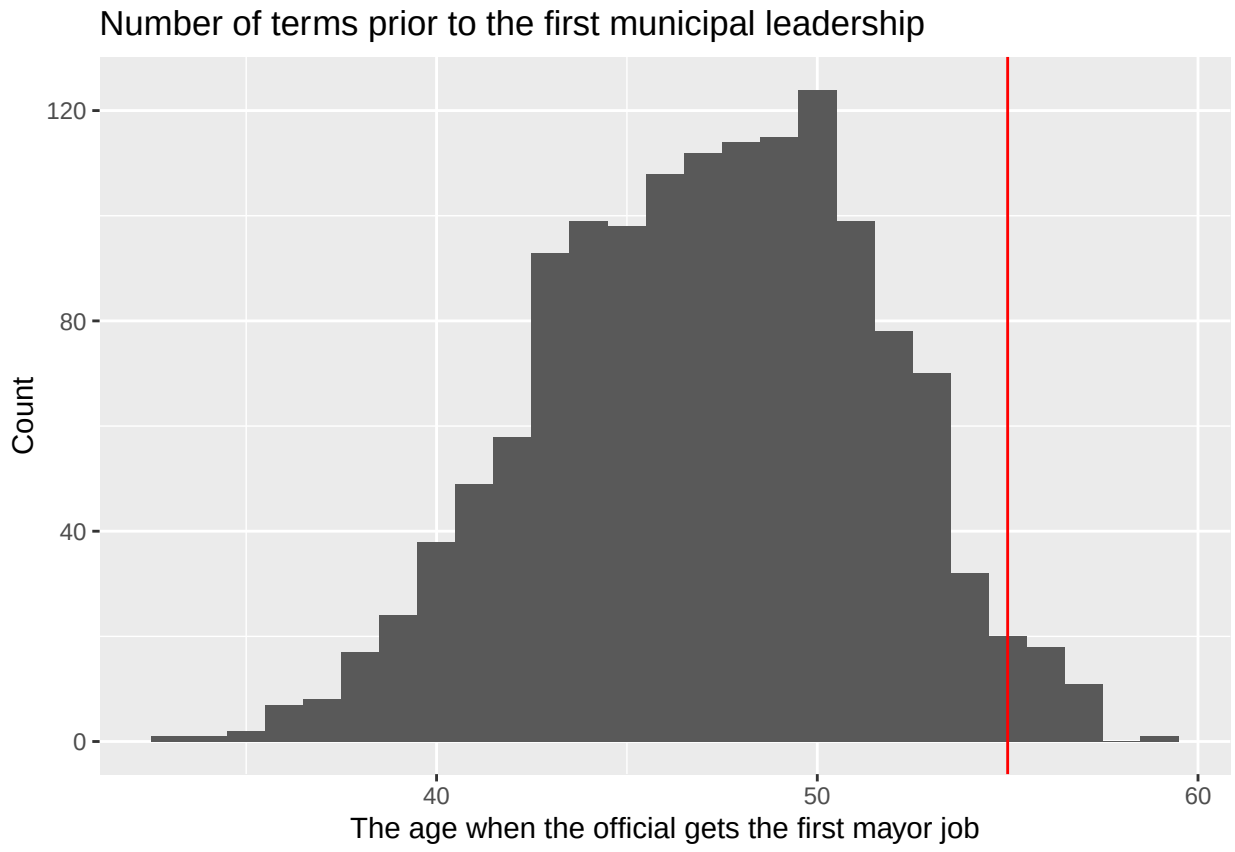
```
## Saving 6.5 x 4.5 in image
```

How much time does it take for one to become a municipal leader

Age also shapes promotion incentives.

```
fig_promotiontime <- df_merged %>%
    group_by(name) %>%
    filter(job == " 市长") %>%
    filter(age == min(age)) %>%
```

```
    ggplot(aes(x = age)) +
    geom_histogram(binwidth = 1) +
    geom_vline(color = "red", xintercept = 55) +
    labs(x = "The age when the official gets the first mayor job",
         y = "Count",
         title = "Number of terms prior to the first municipal leadership")
fig_promotiontime
```



Political cycles on spending

As there is an age constraint on future promotions and substantial uncertainty on how long a politician can serve as a mayor or provincial governor, we would predict that politicians are incentivised to produce good results and signal to the upper organizational department for early promotion. As a politician serves longer on a particular position, promotion incentives may decrease and fiscal necessity pushes the politician back to responsible behavior.

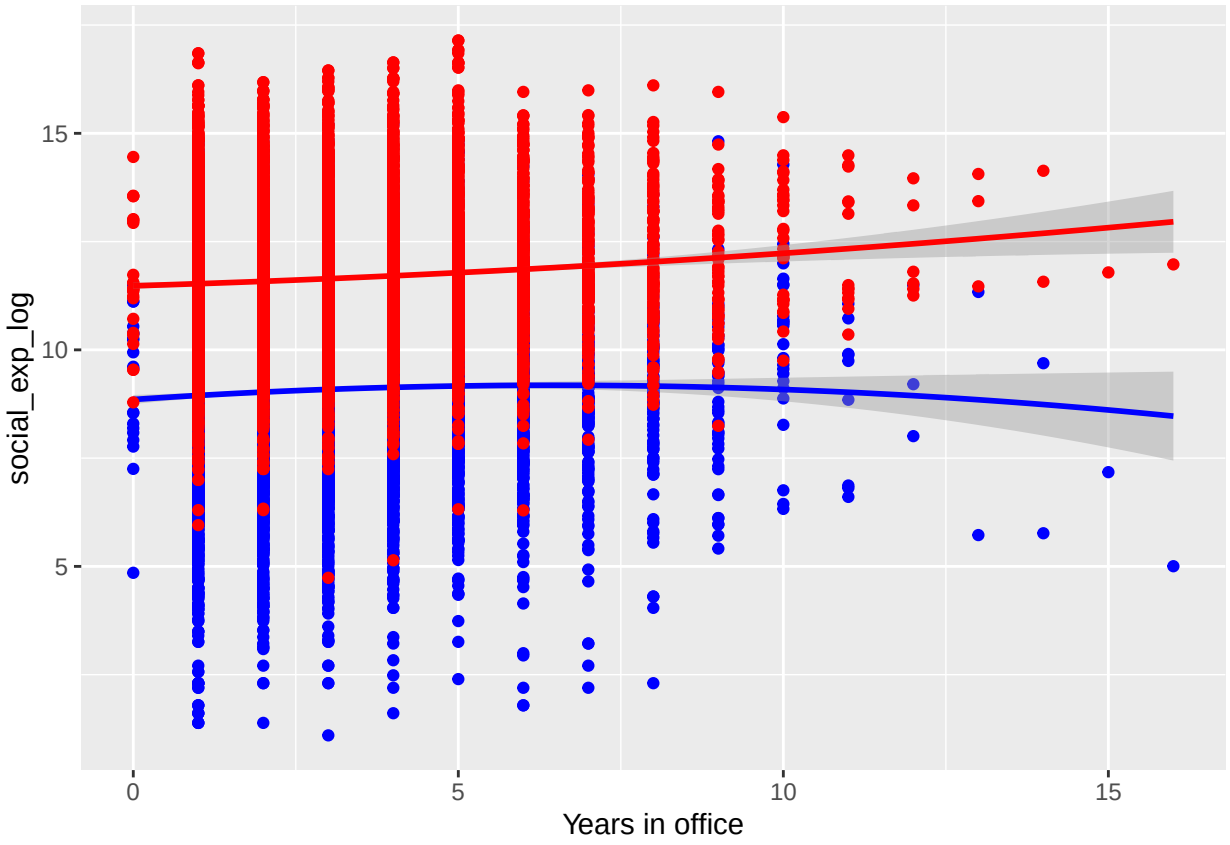
This difference in promotion incentives could be reflected in fiscal behaviors. To signal competence,

The key variables within the project are social expenditure and government income.

In the second model, I estimate the model with robust standard errors.

To account for the effect of age on promotion incentives, I have also Yet promotion incentives also has something to do with age, if

```
# filter out who the mayors/leaders are
df_merged %>%
    ggplot(aes(x = term_year)) +
    geom_point(aes(y = social_exp_log), color = 'blue') +
    geom_smooth(aes(y = social_exp_log), color = 'blue', method = 'lm', formula = y ~ poly(x, 2)) +
    geom_point(aes(y = income_log), color = 'red') +
    geom_smooth(aes(y = income_log), method = 'lm', color = 'red', formula = y ~ poly(x, 2)) +
    labs(x = "Years in office")
```



Regression analysis

```
# This regression is for mayors
model_socexp <- lm(social_exp_log ~ term_year +  factor(year),
                   data = df_merged %>% filter(governor))
# Add a non-linear term
model_socexp2 <- lm(social_exp_log ~ poly(term_year, 2) + factor(year),
                    data = df_merged %>% filter(governor))
# Produce robust standard errors
modelr_socexp2 <- lm_robust(social_exp_log ~ poly(term_year, 2) +factor(year),
                            data = df_merged %>% filter(governor))
# add an interaction with age
model_socexp_interage <- lm(social_exp_log ~ age*term_year + factor(year),
                            data = df_merged %>% filter(governor))
```

```
mdlist_socexp <- list(model_socexp, model_socexp2, modelr_socexp2, model_socexp_interage)
# Producing the regression tables
modelsummary(mdlist_socexp,
             stars = TRUE,
             coef_map = )
```

```
## Warning: In version 0.8.0 of the `modelsummary` package, the default significance markers produced by
## This warning is displayed once per session.
```

Interpretation part

Income trends

```
# This regression is for mayors
model_inc <- lm(income_log ~ term_year + factor(year), data = df_merged %>% filter(governor))
# Add
model_inc2 <- lm(income_log ~ poly(term_year, 2) + factor(year), data = df_merged %>% filter(governor))
# Produce robust standard errors
modelr_inc2 <- lm_robust(income_log ~ poly(term_year, 2) + factor(year), data = df_merged %>% filter(gov
# add an interaction with age
model_inc_interage <- lm(income_log ~ age*term_year + factor(year), data = df_merged %>% filter(governo
mdlist_inc <- list(model_inc, "2nd-degree" = model_inc2, "Robust standard errors" = modelr_inc2, model_
# Producing the regression tables
modelsummary(mdlist_inc,
             stars = TRUE)
```

Different specification

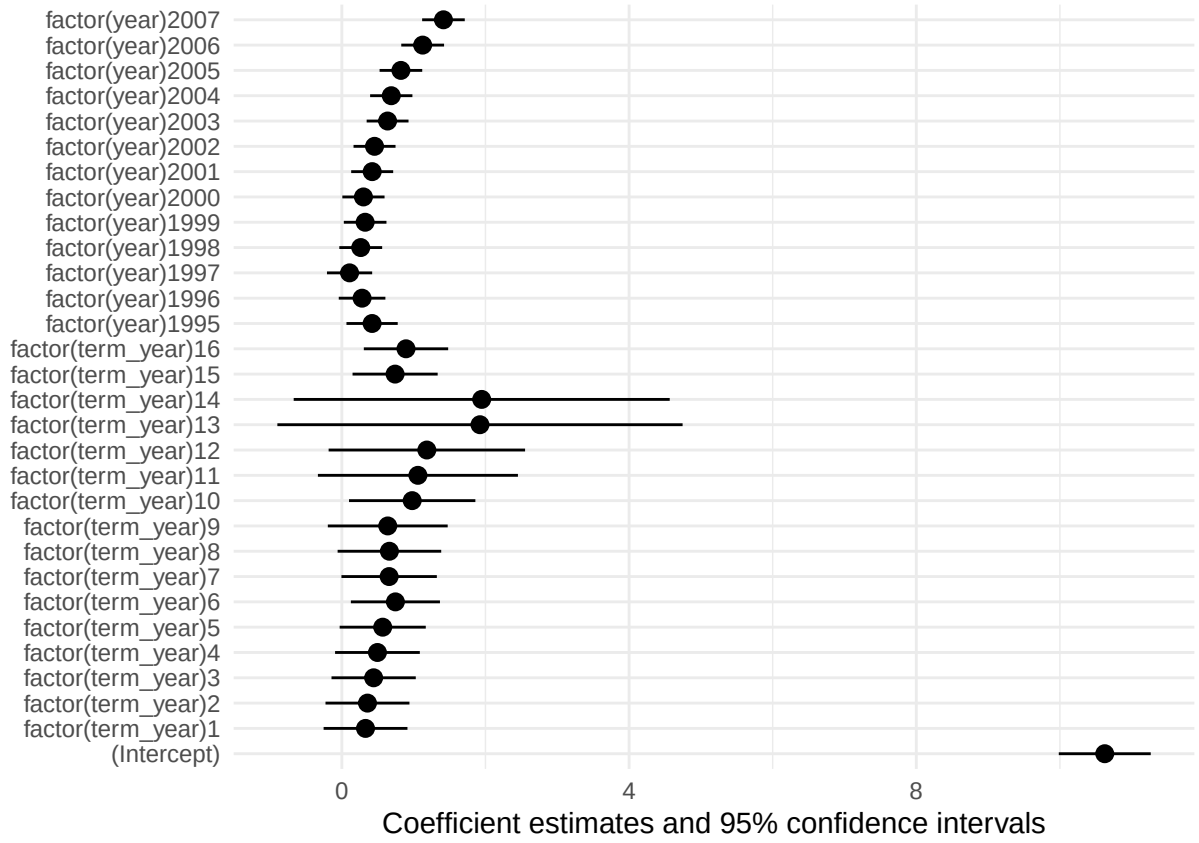
```
# Using indicator variables
model_fac_socexp <- lm_robust(social_exp_log ~ factor(term_year) + factor(year),
                              data = df_merged %>% filter(governor))
model_fac_income <- lm_robust(income_log ~ factor(term_year) + factor(year),
                              data = df_merged %>% filter(governor))
tbl_fac <- list(model_fac_socexp, model_fac_income)
modelplot(model_fac_income)
```

	Model 1	Model 2	Model 3	Model 4
(Intercept)	7.100***	7.179***	7.179***	2.355***
	(0.084)	(0.079)	(0.090)	(0.384)
term_year	0.024+			−0.524***
	(0.013)			(0.115)
factor(year)1999	0.858***	0.844***	0.844***	0.883***
	(0.112)	(0.112)	(0.122)	(0.111)
factor(year)2000	1.198***	1.180***	1.180***	1.192***
	(0.108)	(0.108)	(0.117)	(0.107)
factor(year)2001	1.507***	1.494***	1.494***	1.499***
	(0.105)	(0.105)	(0.113)	(0.103)
factor(year)2002	1.749***	1.739***	1.739***	1.758***
	(0.106)	(0.106)	(0.116)	(0.103)
factor(year)2003	1.967***	1.962***	1.962***	1.910***
	(0.102)	(0.102)	(0.111)	(0.098)
factor(year)2004	2.166***	2.146***	2.146***	2.124***
	(0.105)	(0.105)	(0.112)	(0.101)
factor(year)2005	2.214***	2.188***	2.188***	2.165***
	(0.105)	(0.106)	(0.114)	(0.101)
factor(year)2006	2.544***	2.519***	2.519***	2.507***
	(0.106)	(0.107)	(0.113)	(0.102)
factor(year)2007	4.065***	4.059***	4.059***	3.952***
	(0.105)	(0.105)	(0.105)	(0.100)
poly(term_year, 2)1		3.138+	3.138+	
		(1.705)	(1.838)	
poly(term_year, 2)2		−5.281**	−5.281*	
		(1.730)	(2.066)	
age				0.104***
				(0.008)
age × term_year				0.008***
				(0.002)
Num.Obs.	4771	4771	4771	4469
R2	0.296	0.297	0.297	0.404
R2 Adj.	0.295	0.296	0.296	0.403
AIC	17724.3	17716.9		15852.6
BIC	17801.9	17801.1		15942.3
Log.Lik.	−8850.137	−8845.470		−7912.321
F	200.165	183.133		252.143
Std.Errors			HC2	

+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001

		2nd-degree	Robust standard errors	
(Intercept)	10.866***	11.042***	11.042***	5.992***
	(0.123)	(0.120)	(0.137)	(0.314)
term_year	0.065***			−0.449***
	(0.010)			(0.088)
factor(year)1995	0.431**	0.430**	0.430*	0.457**
	(0.158)	(0.158)	(0.182)	(0.151)
factor(year)1996	0.287+	0.286+	0.286+	0.337*
	(0.153)	(0.153)	(0.165)	(0.146)
factor(year)1997	0.116	0.116	0.116	0.208
	(0.148)	(0.148)	(0.160)	(0.142)
factor(year)1998	0.279*	0.278*	0.278+	0.300*
	(0.139)	(0.139)	(0.152)	(0.133)
factor(year)1999	0.328*	0.328*	0.328*	0.369**
	(0.139)	(0.139)	(0.152)	(0.134)
factor(year)2000	0.309*	0.310*	0.310*	0.324*
	(0.137)	(0.137)	(0.150)	(0.131)
factor(year)2001	0.427**	0.427**	0.427**	0.418**
	(0.135)	(0.135)	(0.149)	(0.129)
factor(year)2002	0.458***	0.458***	0.458**	0.440***
	(0.135)	(0.135)	(0.149)	(0.129)
factor(year)2003	0.651***	0.650***	0.650***	0.570***
	(0.133)	(0.133)	(0.148)	(0.126)
factor(year)2004	0.689***	0.690***	0.690***	0.629***
	(0.135)	(0.135)	(0.150)	(0.128)
factor(year)2005	0.829***	0.830***	0.830***	0.770***
	(0.135)	(0.135)	(0.152)	(0.128)
factor(year)2006	1.128***	1.129***	1.129***	1.074***
	(0.136)	(0.136)	(0.151)	(0.128)
factor(year)2007	1.424***	1.424***	1.424***	1.293***
	(0.135)	(0.135)	(0.151)	(0.127)
poly(term_year, 2)1		8.717***	8.717***	
		(1.407)	(1.486)	
poly(term_year, 2)2		0.464	0.464	
		(1.401)	(1.322)	
age				0.107***
				(0.006)
age × term_year				0.007***
				(0.002)
Num.Obs.	5566	5566	5566	5160
R2	0.075	0.075	0.075	0.268
R2 Adj.	0.073	0.073	0.073	0.265
AIC	19436.6	19438.5		16887.2
BIC	19542.6	19551.1		17005.0
Log.Lik.	−9702.305	−9702.250		−8425.582
F	32.172	30.030		117.487
Std.Errors			HC2	

+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001




```
# modelplot(model_fac_socexp)
```

Visualize the results of specification

```
# Do a CI graph
```

```
# Creating lagged DV in the dataset
df_lagged <- df_merged %>%
    group_by(admcode, year) %>%
    summarise(sumed_disciplined = any(disciplined == 1),
              sumed_exp_log = mean(exp_log),
              sumed_income_log = mean(income_log),
              sumed_social_exp_log = mean(social_exp_log)) %>%
    mutate(treatment = lag(sumed_disciplined),
           lag_exp = lag(sumed_exp_log),
           lag_income = lag(sumed_income_log),
           lag_socialexp = lag(sumed_social_exp_log)) %>%
    select(-sumed_disciplined, -sumed_exp_log) %>%
    right_join(df_merged, by = c('admcode' = 'admcode',
                                 'year' = 'year'))
```

```
## `summarise()` has grouped output by 'admcode'. You can override using the `.groups` argument.
```

	Model 1	Model 2
(Intercept)	1.228***	0.229***
	(0.076)	(0.065)
lag_socialexp	0.913***	
	(0.008)	
poly(term_year, degree = 2)1	−3.143***	−0.506+
	(0.953)	(0.290)
poly(term_year, degree = 2)2	0.566	0.270
	(1.652)	(0.294)
lag_income		0.994***
		(0.006)
Num.Obs.	4301	5209
R2	0.834	0.965
R2 Adj.	0.833	0.965
Std.Errors	HC2	HC2

+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001

```
# Add a lagged DV to the regression
model_lag_socexp <- lm_robust(social_exp_log ~ lag_socialexp + poly(term_year, degree = 2),
                      data = df_lagged %>% filter(governor))

model_lag_income <- lm_robust(income_log ~ lag_income + poly(term_year, degree = 2),
                      data = df_lagged %>% filter(governor))
tbl_lag <- list(model_lag_socexp, model_lag_income)
modelsummary(tbl_lag, stars = TRUE)
```

Bootstrap the standard errors