Report: Becoming Father

Age at first birth among men in Germany based on the SOEP

Henrik Schubert

2023-06-15

Abstract

Men's fertility patterns deviate from women's, with a shift towards later ages and a wider age distribution of childbearing. However, limited information exists on the age distribution of first births among men. This study utilizes data from the Socio-ökonomisches Panel (SOEP) to investigate the transition to fatherhood. Non-parametric approaches and survival models are used to explore the impact of age, while considering socio-economic factors. Cohort shifts and East-West disparities are emphasized. This study contributes to the understanding of men's fertility by examining the age distribution of first births. Using SOEP data, insights are gained into the interplay between age, socio-economic factors, and men's fertility. This research aids decision-making on demographic challenges in modern societies.

Purpose

Fertility of men deviates from fertility of women. Research points at a wider age-distribution of childbearing and that fertility is more shifted towards the later ages. Despite the increasing evidence on sex differences with respect to age-specific fertility, the information on the age distribution of first births among men remains scarce. For that reason this study utilizes the *Socio-ökonomisches Panel* (SOEP) in order to describe the transition to fatherhood. We use non-parametric approaches as well as survival models to better investigate the effect of age net of other socio-econonmic factors. A focus of this study lies on cohort differences and differences between East and West.

Data wrangling

For the study we harness the *biobirth* questionnaire from SOEP. The questionnaire contains questions on biological children of the respondent. The Figure @ref(fig:interview-dates) below illustrates the distribution of interview years for that particular questionnaire. It becomes visible that the interviews were mostly executed after the year 2000 and they were biannually.

```
### Load residence data ---------------------------------------------
if(all(isFALSE(estimate) &
       file.exists("Data/spell_data.Rda") &
       file.exists("Data/person_data.Rda"))){

  # Load the data
  load("Data/spell_data.Rda")
  load("Data/person_data.Rda")

}else{



### Clean the bio-birth data  ----------------------------------------

# Load the birth data
```

```r
fert <- read_stata("SOEP_V36/Stata/biobirth.dta")

# Remove respondents that where no asked the question
fert <- fert |> filter(bioyear != -1 & gebjahr != -1)

# Filter men
fert <- fert |> filter(sex == 1)

# Remove unimportant variables
fert <- fert |> select(!starts_with("kidsex"))

# Make everything as double
fert <- fert |> mutate(across(where(is.factor), as.double))

# Make missing, where values are either -2 or -1
fert <- fert |> replace_with_na_all(condition = ~.x %in% c(-2, -1))

# Clean the names
names(fert) <- sub("(.*)(\\d{2})$", "\\1_\\2", names(fert))

# Make a life-course perspective
fert2 <- fert |> pivot_longer(cols = starts_with("kid"),
                              names_pattern = "([a-z]*)_([0-9]*)",
                              values_to = "Value",
                              names_to = c("Variable", "Number"))

# Filter first births
fert2 <- fert2 |> filter(Number == "01")



# Pivot wider
fert2 <- fert2 |> pivot_wider(names_from = c(Variable, Number),
                              values_from = Value)

# Create cohorts - split by 5 year groups
fert2 <- fert2 |> mutate(cohort = cut(gebjahr, breaks = seq(1900, 2020, by = 10), dig.lab = 4))

# Filter the data
fert2 <- fert2 |> filter(cohort %in% cohorts)

# Double check
fert2 <- fert2 |> filter(!is.na(gebjahr) & !is.na(bioyear))

# Create an event and censoring variable
fert2 <- fert2 |> mutate(Event = if_else(is.na(kidgeb_01), 0, 1),
                         Censoring = if_else(Event == 0, bioyear - gebjahr, kidgeb_01 - gebjahr))

### Split the data ----------------------------------------------

# Split the data
spell_data <- survSplit(fert2, cut = 15:55, end = "Censoring", event = "Event", start = "start")
```

```r
### Save the data
save(spell_data, file = "Data/spell_data.Rda")
save(fert2, file = "Data/person_data.Rda")


### Distribution of questionnaires
ggplot(fert2, aes(bioyear)) +
  geom_histogram() +
  scale_y_continuous(expand = c(0, 0)) +
  ylab("Year of biobirth interview")

}
```

Table @ref(table:data-structure1) displays the current shape of the data, when only showing the first 10 cases. Essentially, it is a single spell data set, which includes retrospective information on the fertility history.

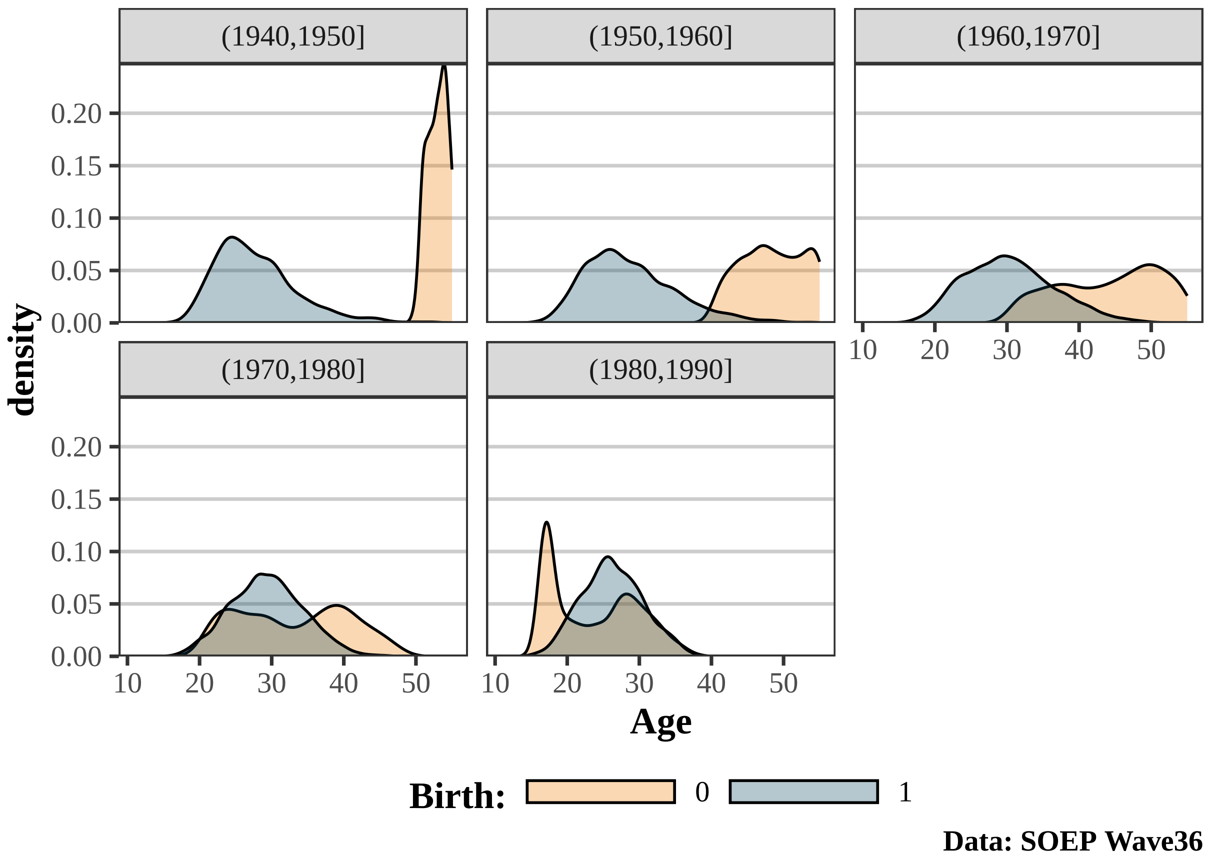
```r
# Make a table of the interview dates
fert2 |>
  arrange(persnr, bioage) |>
  slice_head(n = 10) |>
  select(pid, cohort, bioyear, bioage, Event) |>
  pander()
```

pid	cohort	bioyear	bioage	Event
604	(1980,1990]	2007	17	0
1603	(1980,1990]	2003	17	0
9403	(1980,1990]	2003	17	1
13404	(1980,1990]	2004	17	0
13405	(1980,1990]	2005	17	0
13406	(1980,1990]	2007	17	0
18704	(1980,1990]	2007	17	0
20104	(1950,1960]	2002	49	1
20503	(1980,1990]	2005	17	0
20504	(1980,1990]	2005	17	0

In @ref(fig:event-data) illustrates the distribution of censoring or event times across different cohorts. The x-axis of the plot represents the time variable, either the time of the event (first birth) or the time of censoring (such as loss to follow-up or end of the study). The y-axis represents the frequency or proportion of individuals who have experienced the event or remained uncensored at a given time.

This graphical representation provides valuable insights into the survival experience of a population or a specific group, illustrating the probability of experiencing the event at a specific time point.

```r
### Descriptive data ---------------------------------------------

# Plot discriptively
ggplot(subset(fert2, Censoring <= 55), aes(Censoring, fill = as.factor(Event))) +
  geom_density(alpha = 0.3) +
  guides(colour = guide_legend(nrow = 2, byrow = TRUE)) +
  facet_wrap(~ cohort) +
  scale_y_continuous(expand = c(0, 0)) +
  labs(caption = "Data: SOEP Wave36") +
  scale_fill_manual(name = "Birth:", values = c(MPIDRorange, MPIDRblue)) +
  xlab("Age")
```




```
# Save
ggsave(last_plot(), filename = "Figures/descriptive_age_firstbirth.pdf")
```

Survival analysis

As the data exists already in form to proceed with survival analysis, we make some descriptive estimations. First, we estimate kaplan-meier curves using the following estimator:

$$\hat{S}(t) = \prod_{t_i \leq t} [1 \frac{d_i}{Y_i}]$$

Population

First, we look at the cases which experience the event and which are censored.

```
### Prepare the survival data ------------------------------------

# Look at the survival times
with(fert2, Surv(Censoring, Event))[1:100]
```

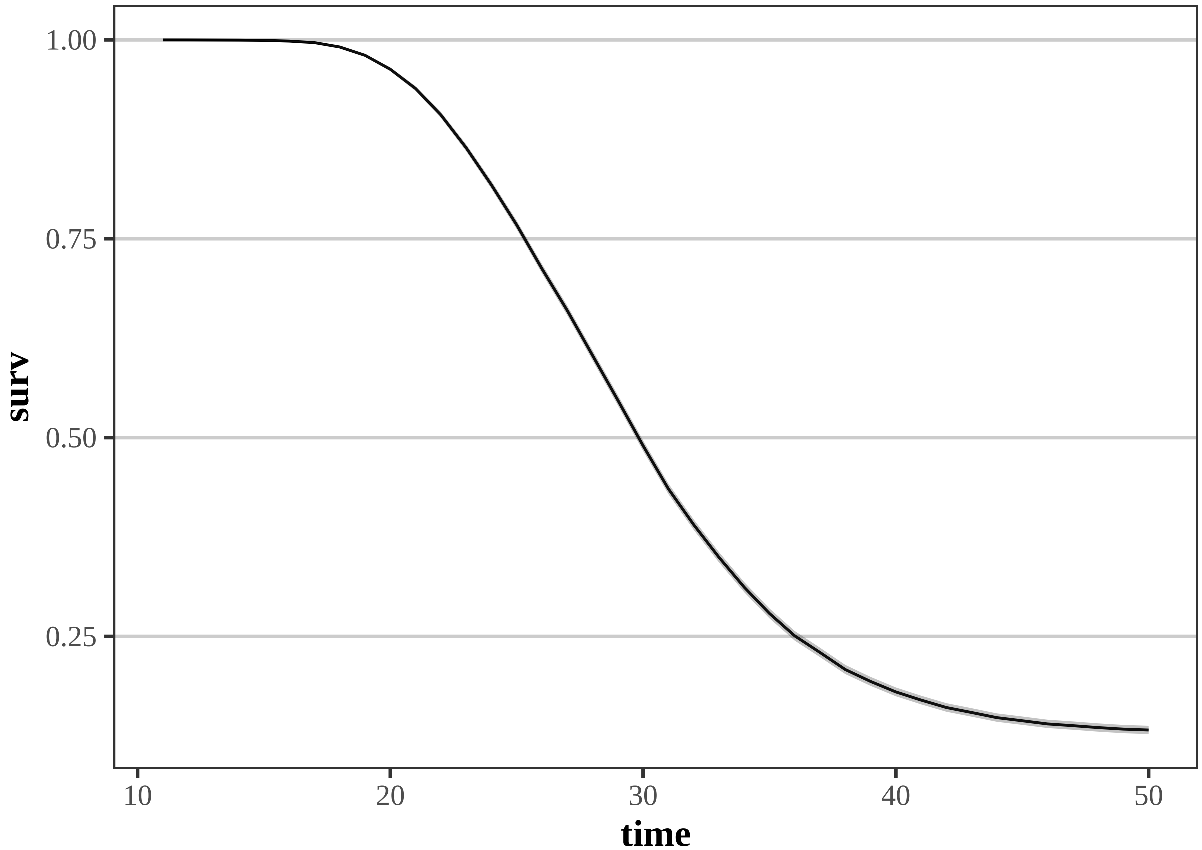
```
##   [1] 17+ 17+ 27+ 17+ 32+ 38  31  28+ 30  20+ 17+ 17+ 17+ 29  25  22  32  17+
##  [19] 35+ 27+ 34  23  27  17+ 17+ 17+ 17+ 24  17+ 17+ 32  39+ 28  21  17+ 17+
##  [37] 34  21  17+ 27+ 25  34+ 27+ 28  26+ 28+ 17+ 25  25  17+ 30+ 43  17+ 28
##  [55] 17+ 34+ 22+ 53+ 25+ 17+ 17+ 19+ 35  38+ 33  17+ 24  24+ 17+ 28  17+ 22+
##  [73] 40  30+ 26  30  25  17+ 26+ 17+ 31  17+ 32+ 25  21  54+ 17+ 17+ 17+ 17+
##  [91] 22  30+ 17+ 35  30  30  26  35+ 21+ 17+
```

```r
# Make the Kaplan-Meier
km <- survfit(Surv(Censoring, Event) ~ 1, conf.type = "log",
              conf.int = 0.95, type = "kaplan-meier", error = "greenwood",
              data = fert2)

# Plot the kaplan meier
km_result <- with(km, data.frame(time, n.risk, n.event, surv, n.censor, cumhaz, std.chaz, lower, upper))
  filter(time <= 50)

# Print the km-table
pander(km_result)
```

time	n.risk	n.event	surv	n.censor	cumhaz	std.chaz	lower	upper
11	24986	2	0.9999	0	8.004e-05	5.66e-05	0.9998	1
12	24984	1	0.9999	0	0.0001201	6.932e-05	0.9997	1
13	24983	3	0.9998	0	0.0002402	9.804e-05	0.9996	1
14	24980	2	0.9997	0	0.0003202	0.0001132	0.9995	0.9999
15	24978	8	0.9994	0	0.0006405	0.0001601	0.999	0.9997
16	24970	23	0.9984	3	0.001562	0.0002501	0.9979	0.9989
17	24944	49	0.9965	1118	0.003526	0.0003759	0.9957	0.9972
18	23777	130	0.991	162	0.008993	0.0006093	0.9898	0.9922
19	23485	249	0.9805	124	0.0196	0.000907	0.9788	0.9823
20	23112	413	0.963	152	0.03747	0.001263	0.9606	0.9654
21	22547	568	0.9387	168	0.06266	0.001647	0.9357	0.9418
22	21811	769	0.9056	169	0.09791	0.002081	0.9019	0.9094
23	20873	953	0.8643	166	0.1436	0.002553	0.8599	0.8687
24	19754	1067	0.8176	190	0.1976	0.003042	0.8127	0.8226
25	18497	1139	0.7673	177	0.2592	0.003547	0.7618	0.7727
26	17181	1233	0.7122	177	0.3309	0.004094	0.7064	0.7181
27	15771	1161	0.6598	275	0.4045	0.004629	0.6536	0.666
28	14335	1227	0.6033	291	0.4901	0.005234	0.5969	0.6097
29	12817	1195	0.547	293	0.5834	0.005888	0.5405	0.5536
30	11329	1186	0.4898	232	0.6881	0.006627	0.4832	0.4964
31	9911	1097	0.4356	237	0.7987	0.007422	0.429	0.4422
32	8577	887	0.3905	203	0.9022	0.008194	0.384	0.3971
33	7487	785	0.3496	177	1.007	0.009008	0.3432	0.3561
34	6525	703	0.3119	128	1.115	0.009882	0.3056	0.3183
35	5694	603	0.2789	144	1.221	0.01078	0.2728	0.2851
36	4947	500	0.2507	122	1.322	0.01169	0.2447	0.2568
37	4325	360	0.2298	120	1.405	0.01249	0.224	0.2358
38	3845	359	0.2084	116	1.498	0.01342	0.2027	0.2142
39	3370	242	0.1934	115	1.57	0.0142	0.1878	0.1991
40	3013	205	0.1802	107	1.638	0.01497	0.1748	0.1859
41	2701	154	0.17	109	1.695	0.01566	0.1646	0.1755
42	2438	134	0.1606	104	1.75	0.01636	0.1553	0.1661
43	2200	86	0.1543	90	1.789	0.0169	0.1491	0.1598
44	2024	85	0.1479	105	1.831	0.0175	0.1427	0.1532
45	1834	48	0.144	93	1.857	0.0179	0.1388	0.1494
46	1693	47	0.14	96	1.885	0.01836	0.1349	0.1453
47	1550	24	0.1378	101	1.901	0.01863	0.1327	0.1432
48	1425	25	0.1354	84	1.918	0.01895	0.1303	0.1407
49	1316	19	0.1335	90	1.933	0.01924	0.1283	0.1388

time	n.risk	n.event	surv	n.censor	cumhaz	std.chaz	lower	upper
50	1207	10	0.1324	85	1.941	0.01942	0.1272	0.1377

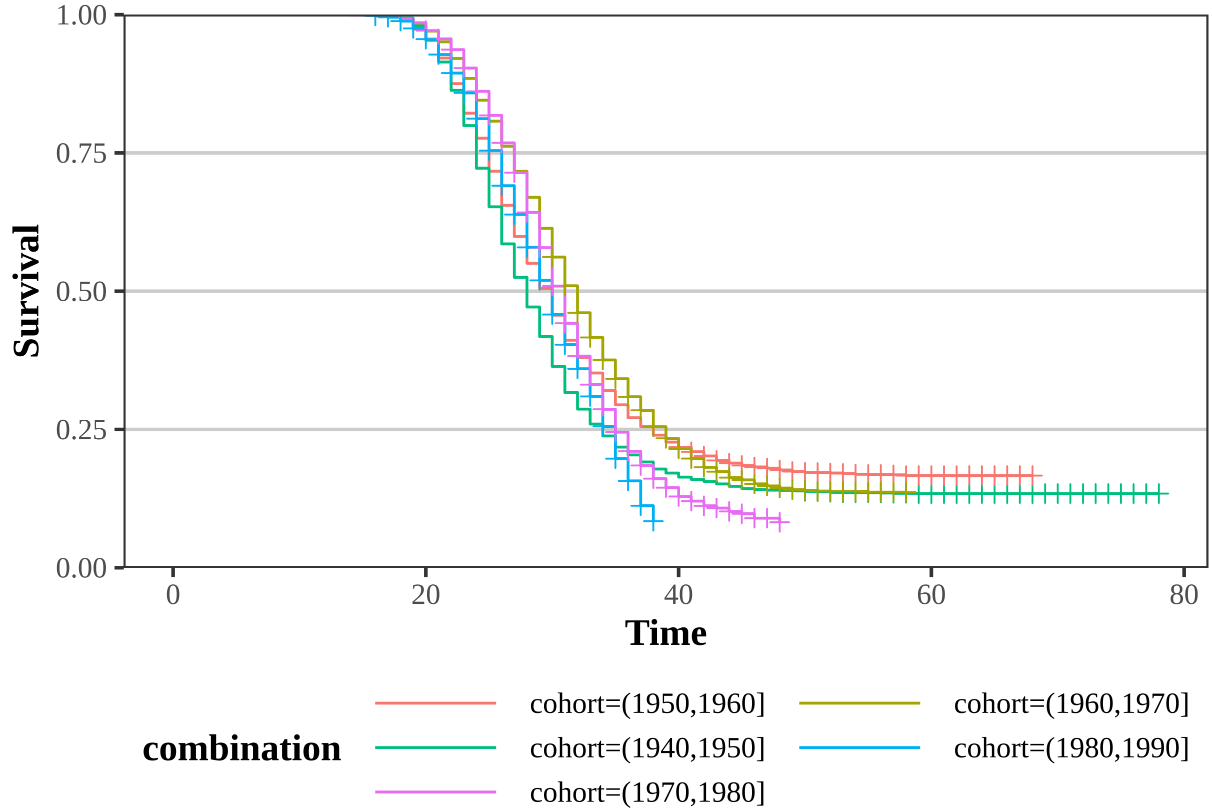
```
ggplot(km_result, aes(time, y=  surv, ymin = lower, ymax = upper)) +
  geom_line() +
  geom_ribbon(alpha = .3)
```



Cohort specific

In Figure @ref(fig:cohort-km), the kaplan-meier curves for specific cohorts are displayed.

```
# Fit by cohort
km_coh <- survfit(Surv(Censoring, Event) ~ cohort, data = fert2, conf.int = 0.95, type = "kaplan-meier"

# Plot
ggsurv(km_coh) +
  scale_y_continuous(expand = c(0, 0), limits = c(0, 1)) +
  guides(colour =  guide_legend(nrow = 3, byrow = TRUE))
```



Smoothed hazard models

Beyond describing the survival process using the Kaplan-Meier estimator, we also estimate smoothed hazard models. The results from the smoothed hazard model are displayed in table.

```
if(all( isFALSE(estimate) & file.exists("Results/smoothed.Rda"))){
  # Load the model
  load("Results/smoothed.Rda")

}else{
  # Run the regression model
mod_cb <- fitSmoothHazard(Event ~ ns(log(Censoring), knots = knots) * cohort,
                          data = fert2,
                          time = "Censoring")

  # Save the model
  save(mod_cb, file = "Results/smoothed.Rda")
}

# Display the results
pander(mod_cb)
```

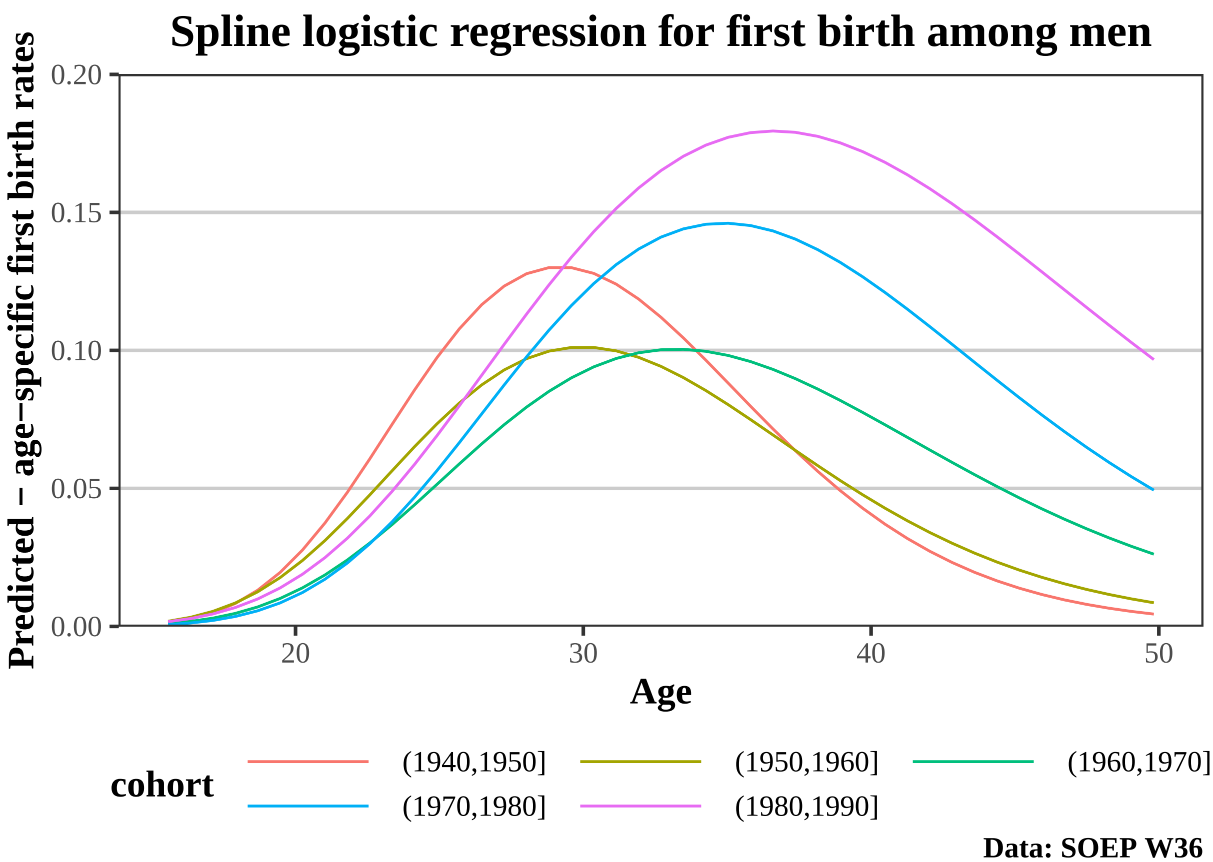
Table 3: Fitting generalized (binomial/logit) linear model: Event ~ ns(log(Censoring), knots = knots) * cohort

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-1794	57.81	-31.03	2.039e-211
ns(log(Censoring), knots = knots)1	1781	57.42	31.01	4.279e-211
ns(log(Censoring), knots = knots)2	783.7	92741	0.008451	0.9933
ns(log(Censoring), knots = knots)3	3065	52757	0.0581	0.9537
cohort(1950,1960]	341	71.41	4.775	1.793e-06
cohort(1960,1970]	531	68.07	7.8	6.197e-15
cohort(1970,1980]	439	77.42	5.67	1.426e-08
cohort(1980,1990]	790.6	96.11	8.226	1.938e-16
ns(log(Censoring), knots = knots)1:cohort(1950,1960]	-338.6	70.94	-4.773	1.815e-06
ns(log(Censoring), knots = knots)2:cohort(1950,1960]	883.3	184	4.801	1.582e-06
ns(log(Censoring), knots = knots)1:cohort(1960,1970]	-525.8	67.64	-7.774	7.619e-15
ns(log(Censoring), knots = knots)2:cohort(1960,1970]	1393	175.3	7.944	1.962e-15
ns(log(Censoring), knots = knots)1:cohort(1970,1980]	-433.2	76.9	-5.632	1.777e-08
ns(log(Censoring), knots = knots)2:cohort(1970,1980]	1167	199.5	5.85	4.903e-09
ns(log(Censoring), knots = knots)1:cohort(1980,1990]	-782.4	95.35	-8.206	2.287e-16
ns(log(Censoring), knots = knots)2:cohort(1980,1990]	2071	249.1	8.313	9.304e-17

In order to get a better understanding of the model, I visualized predicted probabilities of first birth by age and cohort in Figure @ref(fig: pred-smooth).

```
if(all(isFALSE(estimate) & file.exists("Results/predict_smoothed_eha.Rda"))){

  # Load the predicted data
  load("Results/predict_smoothed_eha.Rda")

}else{
  # Plot the result
plot_results <- plot(mod_cb,
                     hazard.params = list(xvar = "Censoring",
                                          by = "cohort",
                                          alpha = 0.10,
                                          ylab = "Hazard",
                                          plot = FALSE))
  # Save the predicted data
  save(plot_results, file = "Results/predict_smoothed_eha.Rda")

}
```

```
# Plot the predicted probabilities
plot_results$fit |>
  filter(Censoring >= 15 & Censoring <= 50 & cohort %in% cohorts) |>
  ggplot(aes(Censoring, visregFit, group = cohort, colour = cohort)) +
  geom_line() +
  scale_y_continuous(limits = c(0, 0.2), expand = c(0, 0)) +
  guides(colour = guide_legend(nrow = 2, byrow = TRUE)) +
  ylab("Predicted - age-specific first birth rates") +
  xlab("Age") +
  ggtitle("Spline logistic regression for first birth among men") +
  labs(caption = "Data: SOEP W36")
```



Discrete time survival model

While the parametric assumptions allow for more *degrees of freedom*, misspecification of the process may occur. In order to circumvent this issue, we have also estimated discrete time hazard models with splines for the age variables. We set the knots at 5-year age intervals. For this estimation, we use data in long-format, as is illustrated below.

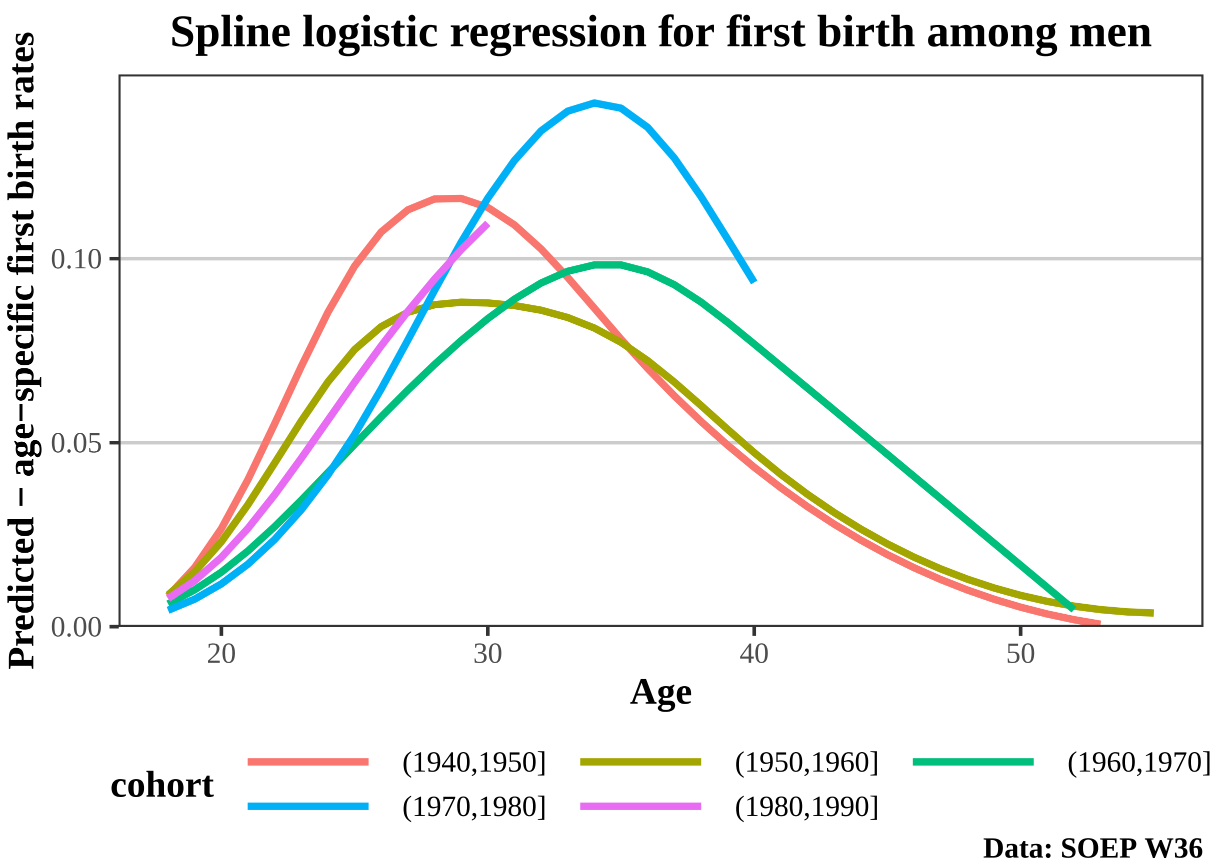
```
if(all(isFALSE(estimate) & file.exists("Results/discrete_eha_splines.Rda"))){

  # Load the data
  load("Results/discrete_eha_splines.Rda")

}else{
```

```r
### Discrete time model ------------------------------------------

# Estimate a logistic regression
logist <- glm(Event ~ ns(Censoring, knots = knots) * cohort, data = spell_data)

# Save the results
save(logist, file = "Results/discrete_eha_splines.Rda")

}

# Create the prediction data
pred_data <- expand.grid(Censoring = 15:55, cohort = unique(fert2$cohort))

# Predict the results
pred_data$prediction <- predict(logist, pred_data)

# Select the data
pred_data <- subset(pred_data, Censoring >= 18 )

# De-select data
pred_data <- pred_data |> filter((cohort == "(1970,1980]" & Censoring <= 40) |
                                  (cohort == "(1980,1990]" & Censoring <= 30 ) |
                                  cohort %in% c("(1950,1960]", "(1960,1970]", "(1940,1950]"))


# Print the spell data
spell_data |>
  arrange(pid, start) |>
  select(persnr, cohort, start, sumkids, Censoring, Event, kidgeb_01) |>
  slice_head(n = 15) |>
  pander()
```

persnr	cohort	start	sumkids	Censoring	Event	kidgeb_01
604	(1980,1990]	0	0	15	0	NA
604	(1980,1990]	15	0	16	0	NA
604	(1980,1990]	16	0	17	0	NA
1603	(1980,1990]	0	0	15	0	NA
1603	(1980,1990]	15	0	16	0	NA
1603	(1980,1990]	16	0	17	0	NA
9403	(1980,1990]	0	1	15	0	2017
9403	(1980,1990]	15	1	16	0	2017
9403	(1980,1990]	16	1	17	0	2017
9403	(1980,1990]	17	1	18	0	2017
9403	(1980,1990]	18	1	19	0	2017
9403	(1980,1990]	19	1	20	0	2017
9403	(1980,1990]	20	1	21	0	2017
9403	(1980,1990]	21	1	22	0	2017
9403	(1980,1990]	22	1	23	0	2017

The results for the discrete-time logistic regression in form of predicted probabilities are displayed below.

```r
# Plot the result
ggplot(pred_data, aes(Censoring, prediction, colour = cohort, group = cohort)) +
  geom_line(size = 1.3)  +
  scale_y_continuous(limits = c(0, 0.15), expand = c(0, 0)) +
  ylab("Predicted - age-specific first birth rates") +
  xlab("Age") +
  ggtitle("Spline logistic regression for first birth among men") +
  labs(caption = "Data: SOEP W36") +
  guides(colour = guide_legend(nrow = 2, byrow = TRUE))
```




```r
# Save the file
ggsave(last_plot(), filename = "Figures/logistic_splines_soep.pdf")
```

A non-parametric approach

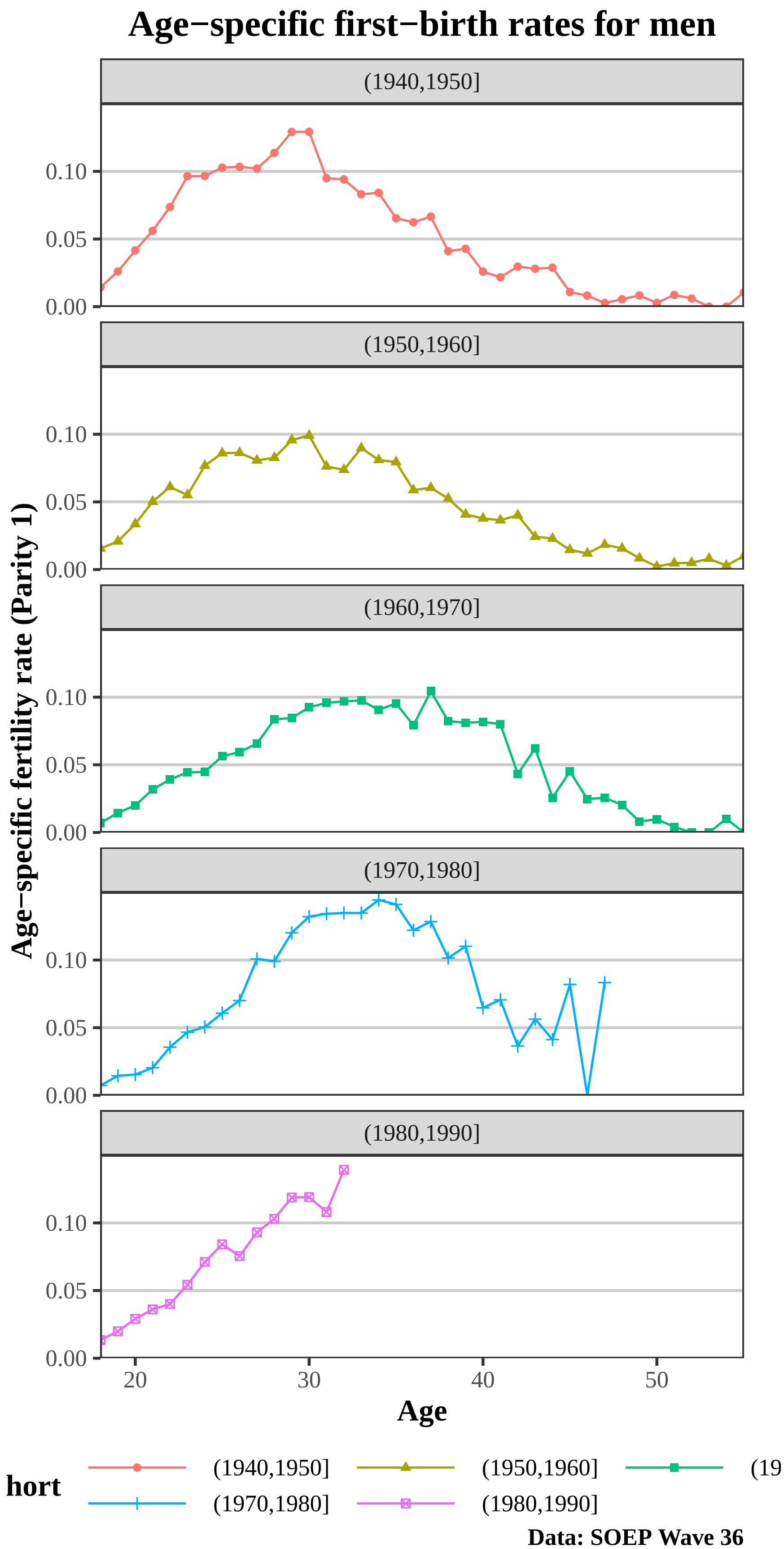
While the models are useful for incorporating covariates, they may rely on too restrictive assumptions. Therefore, we also used a non-parametric approach to estimate age-specific first birth rates.

We used the spell data and aggregated the exposures as well as the births by age. Than, we simply estimated the rates in the following way:

$$rate(x) = \frac{B_{firstbirth}(x)}{P_{childless}}$$

Figure @ref{fig:plot_raw} illustrates the raw age-specific first birth rates for different cohorts. Because the data is from a survey, the rates show an erratic pattern. Nonetheless, the expected bell-shape becomes apparent.

```r
# Estimate the exposures
exposures <- spell_data |> group_by(start, cohort) |> count()

# Count the events
births <- spell_data |> group_by(start, cohort) |> summarise(birth = sum(Event))

# Combine
unparametric <- inner_join(exposures, births) |> mutate(rate = birth / n)


# De-select data
pred_data <- unparametric |> filter((cohort == "(1970,1980]" & start <= 40) |
                                      (cohort == "(1980,1990]" & start <= 30 ) |
                                      cohort %in% c("(1950,1960]", "(1960,1970]", "(1940,1950]"))

# Plot the result
plot_raw <- unparametric |> filter(cohort %in% cohorts & start >= 18) |>
  ggplot(aes(start, rate, colour = cohort, group = cohort, shape = cohort)) +
    geom_line() +
    geom_point() +
    facet_wrap( ~ cohort, ncol = 1) +
    scale_x_continuous(expand = c(0, 0)) +
    scale_y_continuous(expand = c(0, 0), limits = c(0, 0.15)) +
    ggtitle("Age-specific first-birth rates for men") +
    labs(caption = "Data: SOEP Wave 36") +
    ylab("Age-specific fertility rate (Parity 1)") +
    xlab("Age") +
    guides(colour = guide_legend(nrow = 2, byrow = TRUE))


# Plot the result
plot_raw
```



Age−specific first−birth rates for men
(1940,1950]
(1950,1960]
(1960,1970]
(1970,1980]
(1980,1990]
0.10
0.05
0.00
Age−specific fertility rate (Parity 1)
20
30
40
50
Age
hort
(1940,1950]
(1950,1960]
(19
(1970,1980]
(1980,1990]
Data: SOEP Wave 36

In order to reduce the noise and random fluctuations, which result from limited case numbers and the spread of the interview dates, we have smoothed the age-specific first birth rates using a *locally estimated scatterplot smoothing* (loess). The results are presented in Figure @ref{fig:smoothed-rates}

```
# Plot interpolated
plot_interpol <- unparametric |>
  filter(cohort %in% cohorts & start >= 18) |>
  ggplot(aes(start, rate, colour = cohort, group = cohort, linetype = cohort, fill = cohort)) +
    geom_smooth(se = FALSE) +
    scale_x_continuous(expand = c(0, 0)) +
    scale_y_continuous(expand = c(0, 0), limits = c(0, 0.17)) +
    facet_wrap( ~ cohort, ncol = 1) +
    ggtitle("Age-specific first-birth rates for men (smoothed)") +
    labs(caption = "Data: SOEP Wave 36") +
    ylab("Age-specific fertility rate (Parity 1)") +
    xlab("Age") +
    guides(colour = guide_legend(nrow = 2, byrow = TRUE),
           linetype = guide_legend(nrow = 2, byrow = TRUE)) +
    scale_linetype_manual(values = c("dashed", "dotted", "longdash", "twodash", "solid"))
```

Comparison by birth region

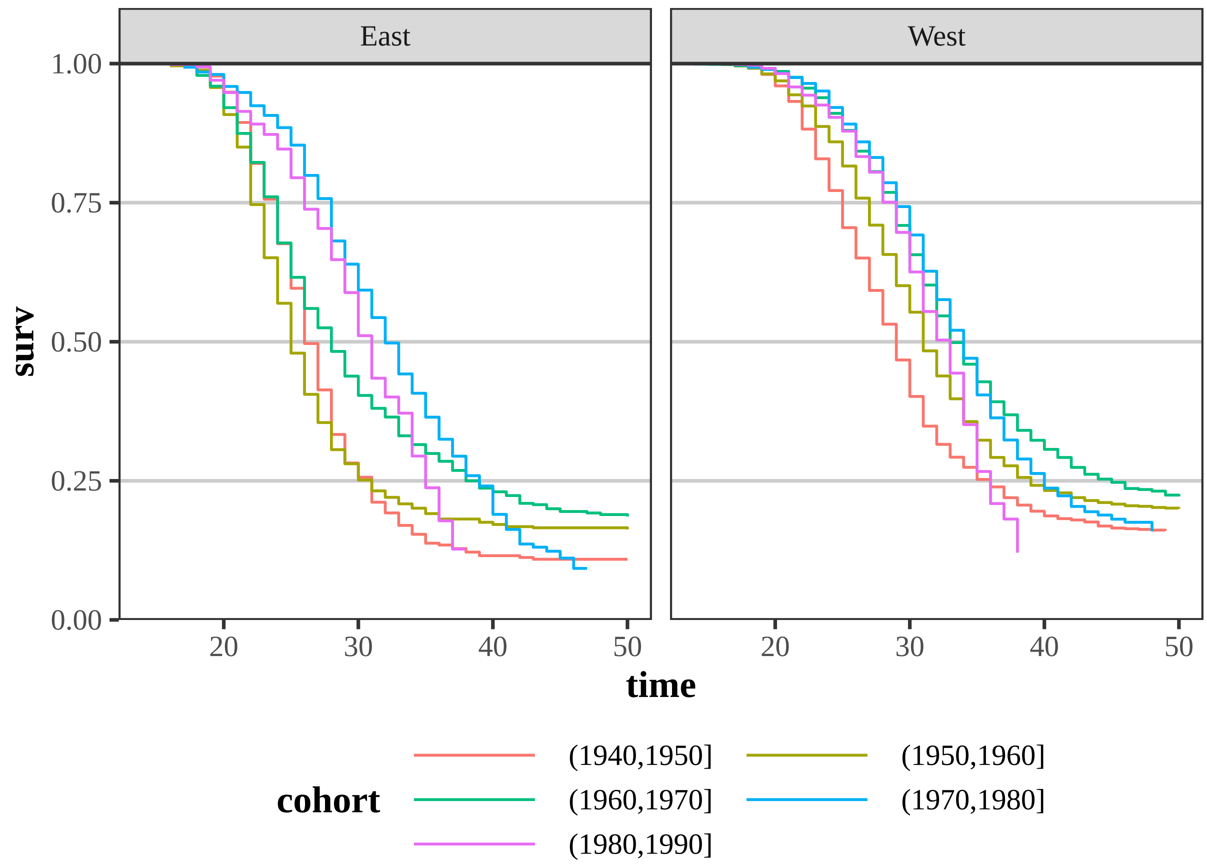
It is very likely that some of the change in the age distribution is driven by the impact reunification, which caused migration as well as fertility postponement. Thus, we estimated non-parametric age-specific first birth rates separately by birth region. The sample was split into persons who were born in East-Germany and respondents who were born in West Germany. Following common practice, respondents from Berlin were classified as East-German.

```
if(all(isFALSE(estimate) & file.exists("Data/region_spell_data.Rda"))){

  # Load the data
  load("Data/region_spell_data.Rda")


}else{

### Prepare the background data ---------------------------

# Load the data
id <- read_dta(file = "SOEP_V36/Stata/ppfad.dta")

# Select variables
id <- subset(id, select = c(persnr, pid, birthregion))

# Clean the birthregion
id$birthregion <- ifelse(id$birthregion %in% 11:16, "East",
       ifelse(id$birthregion %in% 1:10, "West", NA_character_))


### Combine with background variables ----------------------------

# Join with birthregion
fert2 <- left_join(fert2, id)
```

```r
# Filter respondents where the birth information are existent
fert2 <- fert2 |> filter(!is.na(birthregion))

# Create spell data
spell_data_reg <- survSplit(fert2, cut = 15:55, end = "Censoring", event = "Event", start = "start")

# Save the data
save(spell_data_reg, file = "Data/spell_data_reg.Rda")

}

# Create the prediction data
pred_data <- expand.grid(Censoring = 15:55, cohort = unique(fert2$cohort), birthregion = c("East", "Wes
```

Once we have prepared the data, we estimate Kaplan-Meier curves by region.

```r
# Fit by cohort
km_coh_reg <- survfit(Surv(Censoring, Event) ~ cohort + birthregion,
                      data = fert2, conf.int = 0.95,
                      type = "kaplan-meier", error = "greenwood")

# Transform into a data frame
km_coh_reg_data <- surv_summary(km_coh_reg, data = fert2) |>
  filter(time <= 50)

# Plot
ggplot(km_coh_reg_data, aes(x = time, y = surv,colour = cohort)) +
  geom_step() +
  scale_y_continuous(expand = c(0, 0), limits = c(0, 1)) +
  guides(colour =  guide_legend(nrow = 3, byrow = TRUE)) +
  facet_wrap( ~ birthregion)
```

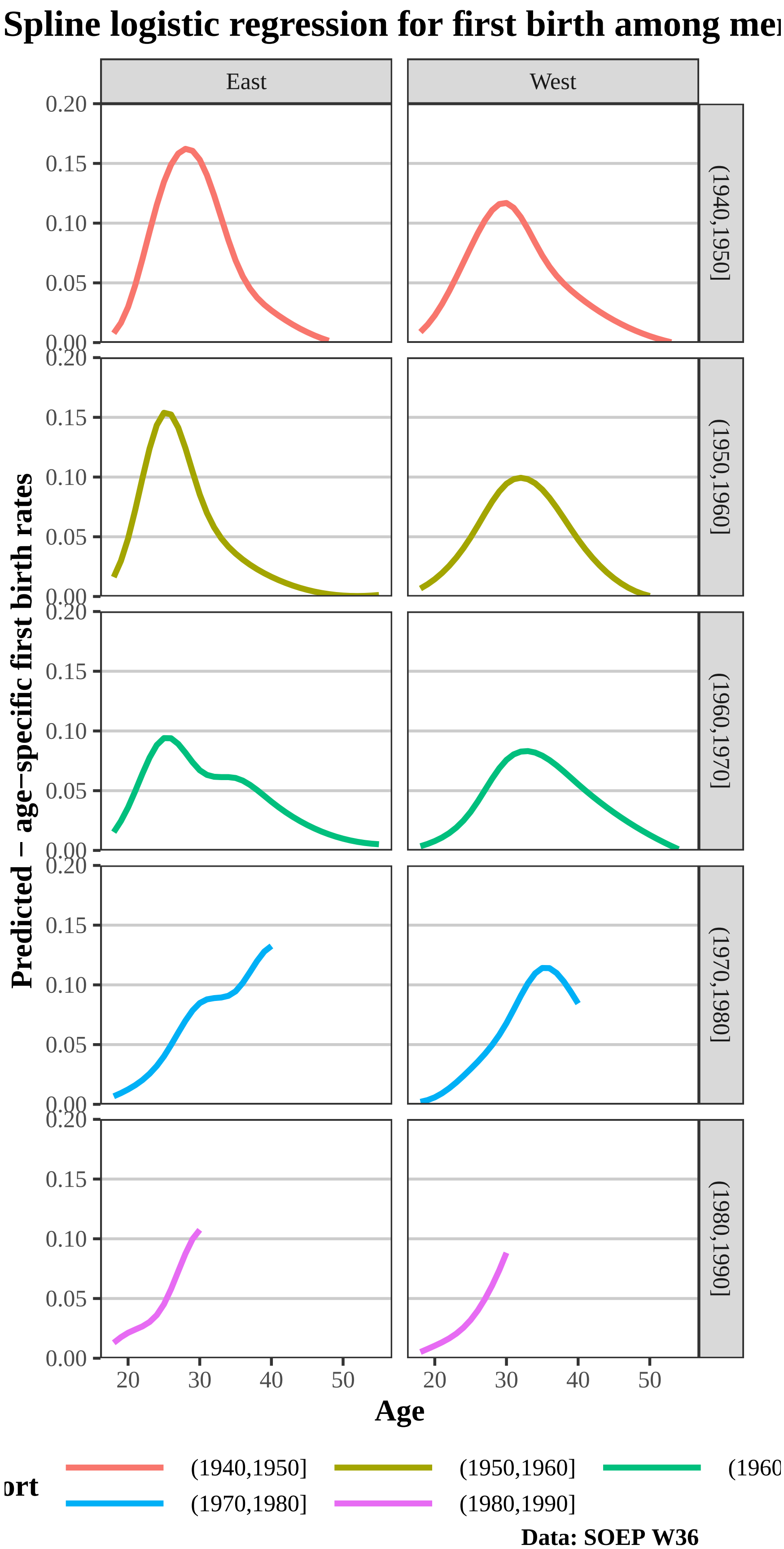



```
# Save the plot
ggsave(last_plot(), filename = "Figures/km_reg-coh.pdf")
```

In the next step, we estimate a discrete time survival regression with knots in 5-year intervals, with interactions between cohort and birth region. We than plot the predicted probabilities from the model in @ref(fig:pred-reg)

```
# Estimate a logistic regression
logist <- glm(Event ~ ns(Censoring, knots = knots) * cohort * birthregion,
              data = spell_data_reg)

# Predict the results
pred_data$prediction <- predict(logist, pred_data)

# Select the data
pred_data <- subset(pred_data, Censoring >= 18 )

# De-select data
pred_data <- pred_data |>
  filter((cohort == "(1970,1980]" & Censoring <= 40) |
         (cohort == "(1980,1990]" & Censoring <= 30 ) |
          cohort %in% c("(1950,1960]", "(1960,1970]", "(1940,1950]"))

# Plot the result
ggplot(pred_data, aes(Censoring, prediction, colour = cohort, group = cohort)) +
  geom_line(size = 1.3)  +
  scale_y_continuous(limits = c(0, 0.2), expand = c(0, 0)) +
  ylab("Predicted - age-specific first birth rates") +
```

```
  xlab("Age") +
  facet_grid(cohort ~ birthregion) +
  ggtitle("Spline logistic regression for first birth among men") +
  labs(caption = "Data: SOEP W36") +
  guides(colour = guide_legend(nrow = 2, byrow = TRUE))
```



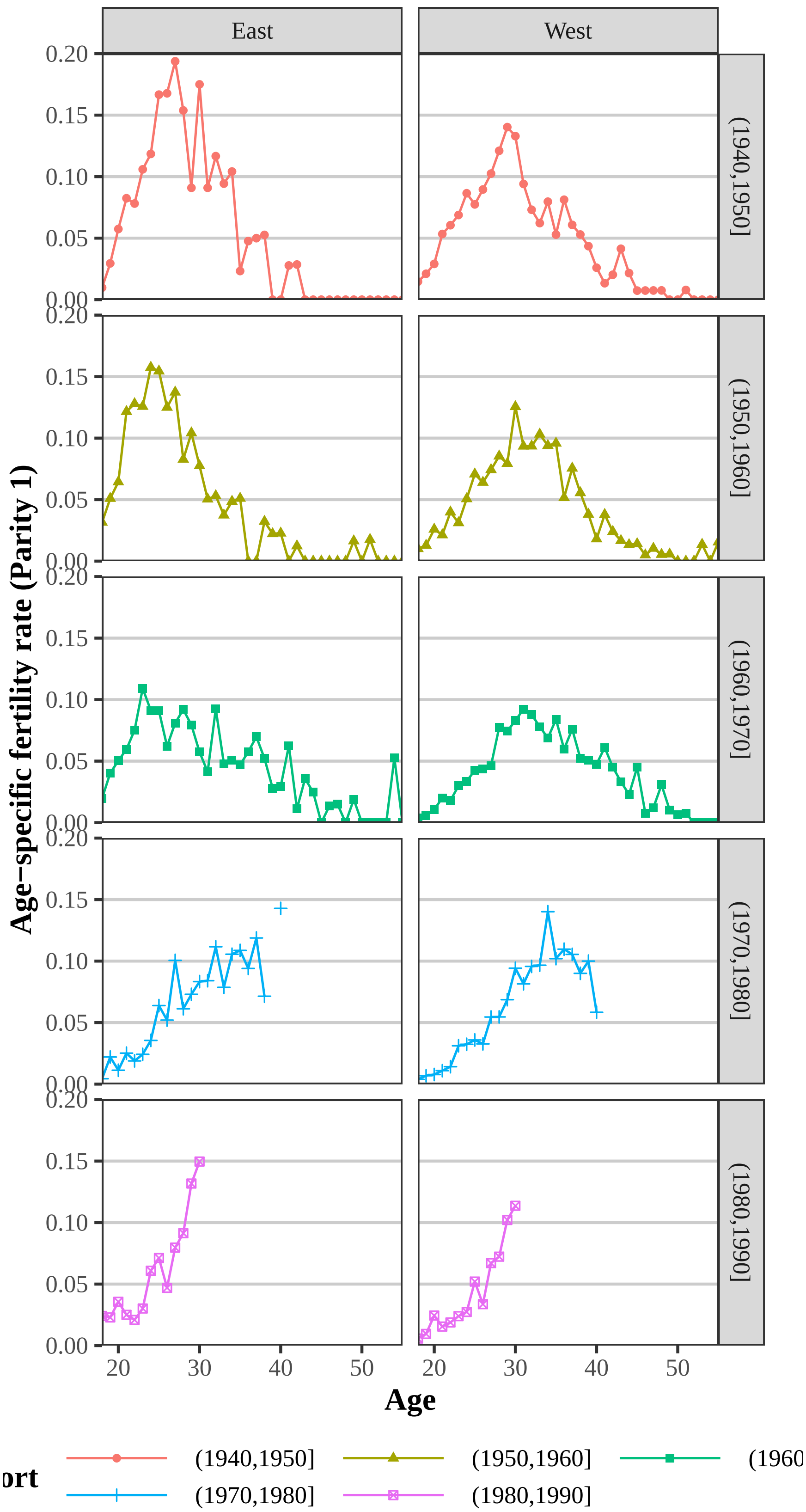
Spline logistic regression for first birth among me
East
West
(1940,1950]
(1950,1960]
(1960,1970]
(1970,1980]
(1980,1990]
0.00
0.05
0.10
0.15
0.20
Predicted – age−specific first birth rates
20
30
40
50
Age
ort
(1940,1950]
(1950,1960]
(1960
(1970,1980]
(1980,1990]
Data: SOEP W36

```r
# Save the file
ggsave(last_plot(), filename = "Figures/logistic_reg_soep.pdf")
```

As outlined earlier, the models may suffer from subjectivity and parametric assumptions, while they increase the degrees of freedom. We estimate the age-specific first birth rates using the non-parametric approach as well. The results with the raw birth rates is displayed in Figures @ref(fig:nonpara-reg).

```r
### Unparametric by birthregion ------------------------------------

# Estimate the exposures
exposures <- spell_data_reg |> group_by(start, cohort, birthregion) |> count()

# Count the events
births <- spell_data_reg |> group_by(start, cohort, birthregion) |> summarise(birth = sum(Event))

# Combine
unparametric_reg <- inner_join(exposures, births) |> mutate(rate = birth / n)

# De-select data
unparametric_reg <- unparametric_reg |>
  filter((cohort == "(1970,1980]" & start <= 40) |
         (cohort == "(1980,1990]" & start <= 30 ) |
         cohort %in% c("(1950,1960]", "(1960,1970]", "(1940,1950]"))

# Plot the result
plot_raw_reg <- unparametric_reg |>
  filter(cohort %in% cohorts & start >= 18) |>
  ggplot(aes(start, rate, colour = cohort, group = cohort, shape = cohort)) +
    geom_line() +
    geom_point() +
    facet_grid(cohort ~ birthregion) +
    scale_x_continuous(expand = c(0, 0)) +
    scale_y_continuous(expand = c(0, 0), limits = c(0, 0.2)) +
    ggtitle("Age-specific first-birth rates for men") +
    labs(caption = "Data: SOEP Wave 36") +
    ylab("Age-specific fertility rate (Parity 1)") +
    xlab("Age") +
    guides(colour = guide_legend(nrow = 2, byrow = TRUE))

# Print the result
plot_raw_reg
```

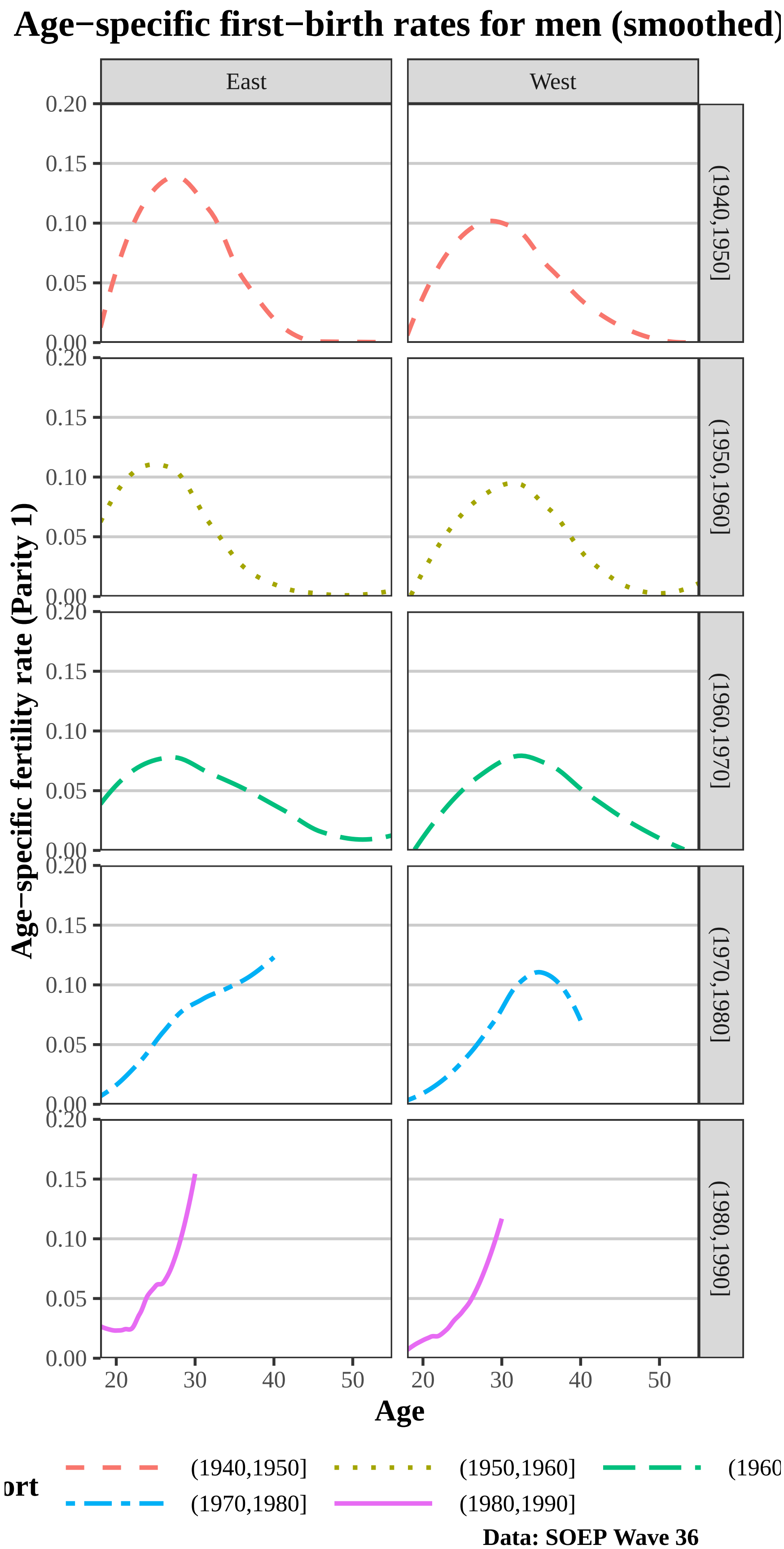
Age−specific first−birth rates for men



Data: SOEP Wave 36

Again, we used *loess* to smooth the rates and to yield a more schematic result. The result is displayed in Figure @ref(fig:nonpara-smooth-reg).

```
# Plot interpolated
plot_interpol_reg <- unparametric_reg |>
  filter(cohort %in% cohorts & start >= 18) |>
  ggplot(aes(start, rate, colour = cohort, group = cohort, linetype = cohort, fill = cohort)) +
    geom_smooth(se = FALSE) +
    facet_grid(cohort ~ birthregion) +
    scale_x_continuous(expand = c(0, 0)) +
    scale_y_continuous(expand = c(0, 0), limits = c(0, 0.2)) +
    ggtitle("Age-specific first-birth rates for men (smoothed)") +
    labs(caption = "Data: SOEP Wave 36") +
    ylab("Age-specific fertility rate (Parity 1)") +
    xlab("Age") +
    guides(colour = guide_legend(nrow = 2, byrow = TRUE),
           linetype = guide_legend(nrow = 2, byrow = TRUE)) +
    scale_linetype_manual(values = c("dashed", "dotted", "longdash", "twodash", "solid"))

# Plot the interpolated result
plot_interpol_reg
```



Age−specific first−birth rates for men (smoothed)
East
West
(1940,1950]
(1950,1960]
(1960,1970]
(1970,1980]
(1980,1990]
0.20
0.15
0.10
0.05
0.00
Age−specific fertility rate (Parity 1)
20
30
40
50
Age
ort
(1940,1950]
(1950,1960]
(1960
(1970,1980]
(1980,1990]
Data: SOEP Wave 36

Parametric regression models

To allow for the inclusion of covariates, we used parametric event-history models. In order to abstain from too restrictive assumptions regarding the parametric shape, we have estimated models with several parametric specifications and compared the results using log-rank tests.

Exponential model

```
### Make parametric hazard models --------------------------------

# Exponential
exp <- par_surv(distribution = "exponential")
stargazer(exp, header = FALSE, type = 'latex')
```

Weibull model

```
# Weibull
weib <- par_surv(distribution = "weibull")
stargazer(weib, header = FALSE, type = 'latex')
```

Gaussian model

```
# Gompertz
#gomp <- par_surv(distribution = "gompertz")

# Gaussian
gauss <- par_surv(distribution = "gaussian")
stargazer(gauss, header = FALSE, type = 'latex')
```

Log-normal model

```
# Lognormal
lognor <- par_surv(distribution = "lognormal")
stargazer(gauss, header = FALSE, type = 'latex')
```

Log-logistic model

```
# Log-logistic
loglog <- par_surv(distribution = "loglogistic")
stargazer(loglog, header = FALSE, type = 'latex')
```

Table 5:

	Dependent variable:
	Censoring
cohort(1910,1920]	
	(0.000)
cohort(1920,1930]	
	(0.000)
cohort(1930,1940]	
	(0.000)
cohort(1940,1950]	-0.643^{***}
	(0.052)
cohort(1950,1960]	-0.578^{***}
	(0.049)
cohort(1960,1970]	-0.480^{***}
	(0.048)
cohort(1970,1980]	-0.483^{***}
	(0.051)
cohort(1980,1990]	
	(0.000)
cohort(1990,2000]	
	(0.000)
cohort(2000,2010]	
	(0.000)
cohort(2010,2020]	
	(0.000)
Constant	4.295^{***}
	(0.041)
Observations	8,270
Log Likelihood	−27,336.590
χ^2	202.054^{***} (df = 11)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 6:

	Dependent variable:
	Censoring
log(scale):1	3.662***
	(0.015)
log(shape):1	0.759***
	(0.024)
log(scale):2	3.684***
	(0.011)
log(shape):2	0.862***
	(0.021)
log(scale):3	3.706***
	(0.008)
log(shape):3	1.111***
	(0.020)
log(scale):4	3.619***
	(0.006)
log(shape):4	1.572***
	(0.024)
log(scale):5	3.510***
	(0.006)
log(shape):5	1.933***
	(0.030)
Observations	8,270
Log Likelihood	−23,471.550
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 7:

	Dependent variable:
	Censoring
cohort(1910,1920]	
	(0.000)
cohort(1920,1930]	
	(0.000)
cohort(1930,1940]	
	(0.000)
cohort(1940,1950]	-2.175^{***}
	(0.532)
cohort(1950,1960]	-1.303^{***}
	(0.488)
cohort(1960,1970]	0.417
	(0.470)
cohort(1970,1980]	-0.405
	(0.508)
cohort(1980,1990]	
	(0.000)
cohort(1990,2000]	
	(0.000)
cohort(2000,2010]	
	(0.000)
cohort(2010,2020]	
	(0.000)
Constant	35.720^{***}
	(0.388)
Log Likelihood	$-24{,}196.430$
χ^2	40.838^{***} (df = 11)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 8:

	Dependent variable:
	Censoring
cohort(1910,1920]	
	(0.000)
cohort(1920,1930]	
	(0.000)
cohort(1930,1940]	
	(0.000)
cohort(1940,1950]	-2.175^{***}
	(0.532)
cohort(1950,1960]	-1.303^{***}
	(0.488)
cohort(1960,1970]	0.417
	(0.470)
cohort(1970,1980]	-0.405
	(0.508)
cohort(1980,1990]	
	(0.000)
cohort(1990,2000]	
	(0.000)
cohort(2000,2010]	
	(0.000)
cohort(2010,2020]	
	(0.000)
Constant	35.720^{***}
	(0.388)
Log Likelihood	$-24{,}196.430$
χ^2	40.838^{***} (df = 11)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 9:

	Dependent variable:
	Censoring
cohort(1910,1920]	
	(0.000)
cohort(1920,1930]	
	(0.000)
cohort(1930,1940]	
	(0.000)
cohort(1940,1950]	-0.068^{***}
	(0.014)
cohort(1950,1960]	-0.019
	(0.013)
cohort(1960,1970]	0.074^{***}
	(0.012)
cohort(1970,1980]	0.078^{***}
	(0.013)
cohort(1980,1990]	
	(0.000)
cohort(1990,2000]	
	(0.000)
cohort(2000,2010]	
	(0.000)
cohort(2010,2020]	
	(0.000)
Constant	3.446^{***}
	(0.010)
Observations	8,270
Log Likelihood	$-22,667.880$
χ^2	214.210^{***} (df = 11)
Note:	*p<0.1; **p<0.05; ***p<0.01