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.

```r
### Load bio-birth 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)

# Filter only relevant cohorts
fert <- fert |> filter(gebjahr %in% birthyears)

# 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(min(birthyears), max(birthyears), by = 10),

# 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))

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

}
```

```
# Plot the distribution of cohorts
ggplot(fert2, aes(gebjahr, fill = cohort)) +
  geom_histogram(binwidth = 2, colour = "white") +
  scale_x_continuous(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0)) +
  guides(fill = guide_legend(nrow = 3, byrow = 2))
```

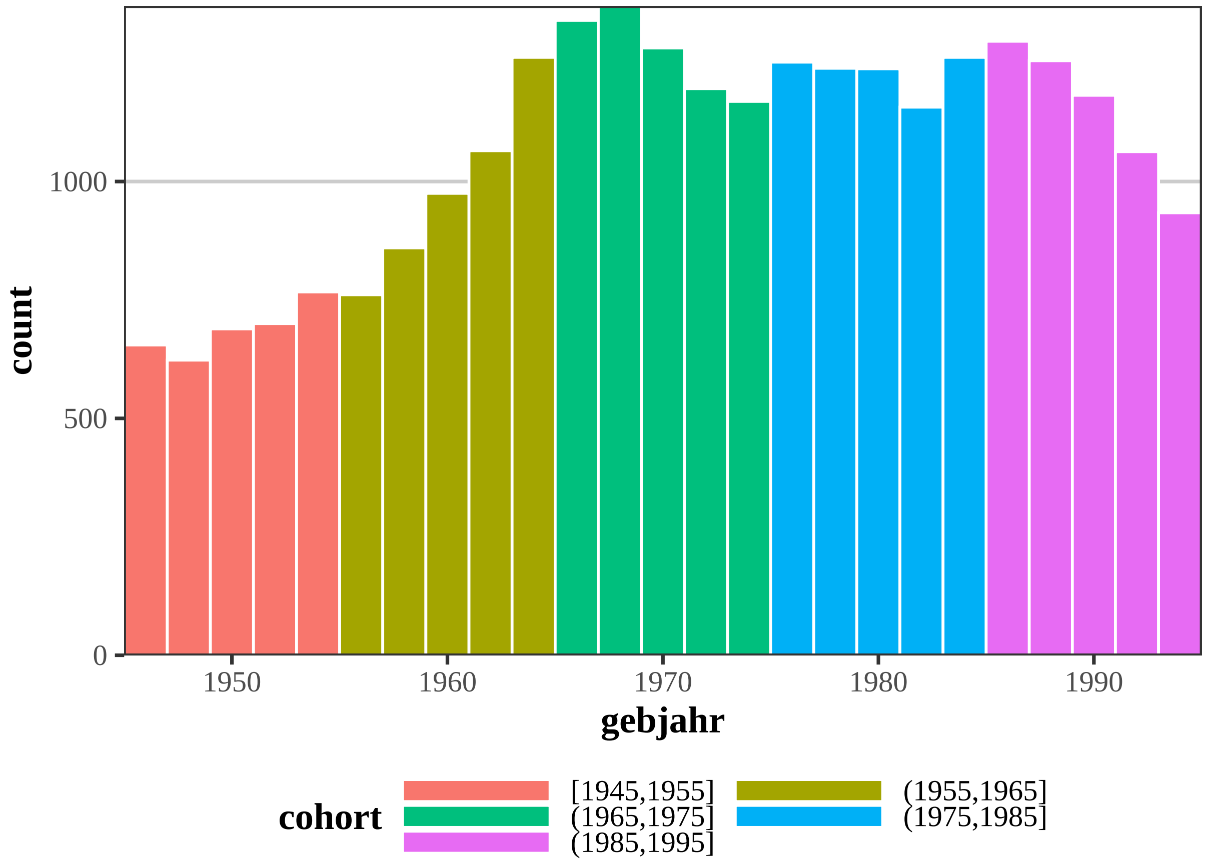


Figure 1: Distribution of biobirth interviews in the SOEP

```
# Save the cohort distribution
ggsave(last_plot(), filename = "Figures/birthyear_distribution.pdf")
```

As can be derived from the last plot, a 10-year cohort aggregation makes sense. Moreover, we split from 1945 every 10 years.

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

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

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


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

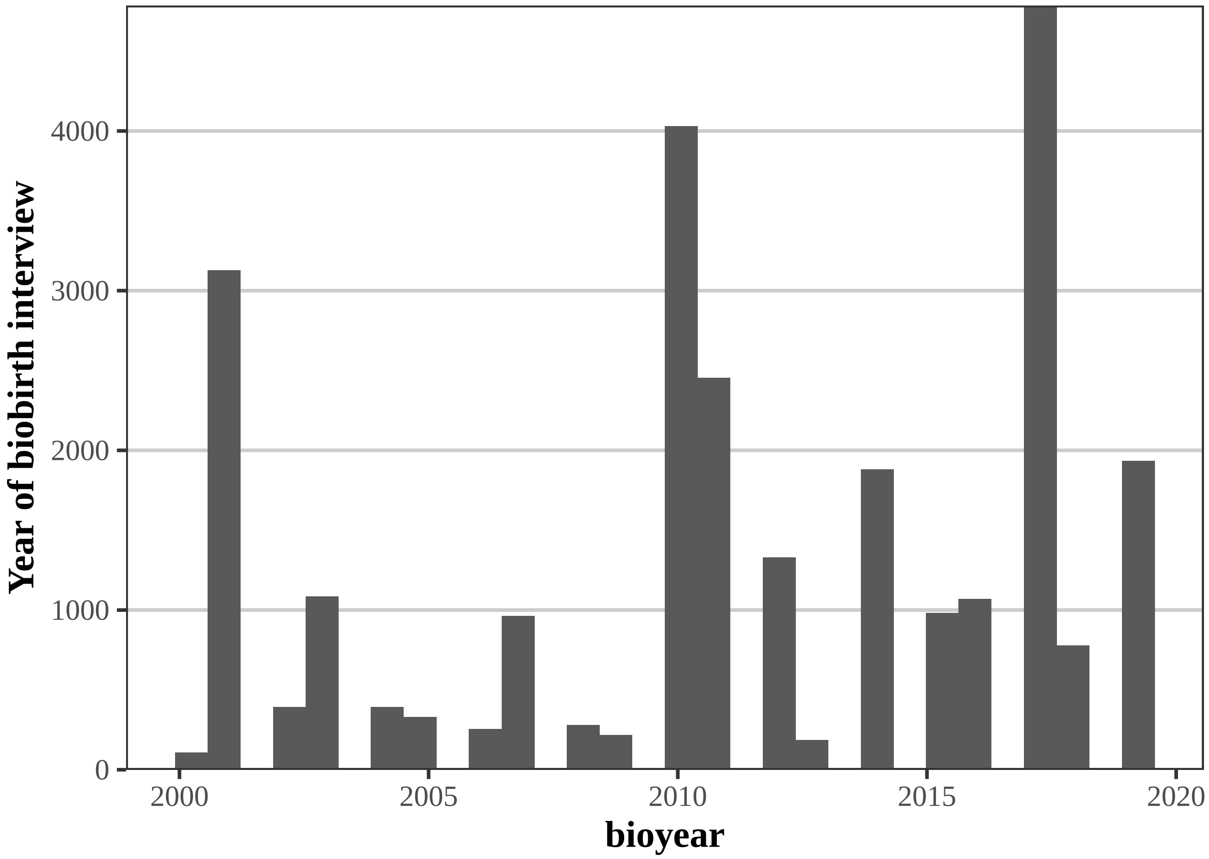


Figure 2: Distribution of biobirth interviews in the SOEP

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.

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

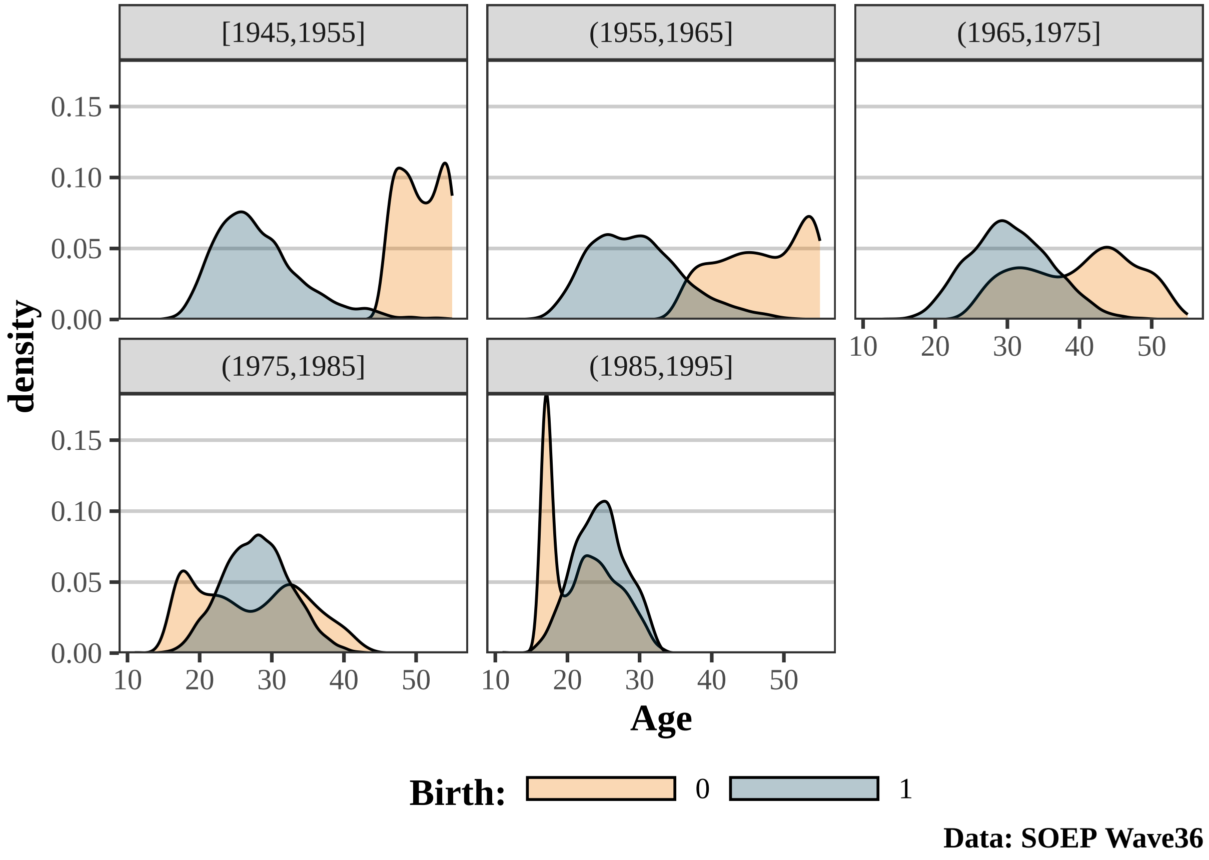
pid	cohort	bioyear	bioage	Event
604	(1985,1995]	2007	17	0
1603	(1985,1995]	2003	17	0
9403	(1985,1995]	2003	17	1
9805	(1985,1995]	2011	17	0
11303	(1985,1995]	2008	17	0
13404	(1985,1995]	2004	17	0
13405	(1985,1995]	2005	17	0
13406	(1985,1995]	2007	17	0

pid	cohort	bioyear	bioage	Event
13903	(1985,1995]	2011	17	0
18704	(1985,1995]	2007	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.

```
### 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]
```

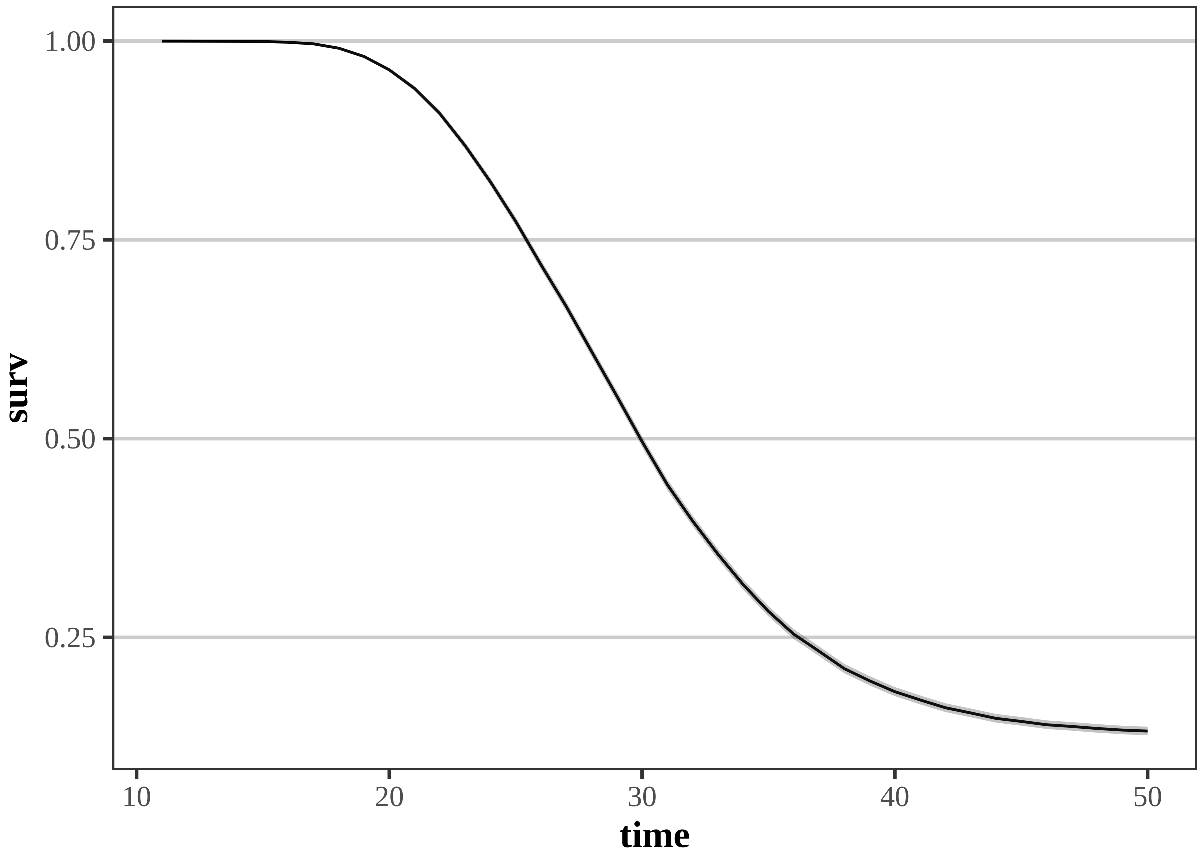
```
##   [1] 17+ 17+ 27+ 17+ 32+ 38  31  17+ 28+ 30  17+ 20+ 17+ 17+ 17+ 17+ 29  25
##  [19] 22  32  17+ 35+ 27+ 34  23  17+ 17+ 27  17+ 17+ 17+ 17+ 24  17+ 17+ 32
##  [37] 17+ 39+ 28  17+ 21  17+ 17+ 17+ 34  21  17+ 27+ 25  34+ 27+ 28+ 28  26+
##  [55] 17+ 17+ 28+ 17+ 17+ 25  25  17+ 17+ 17+ 17+ 30+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

```
# 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	26592	3	0.9999	0	0.0001128	6.513e-05	0.9998	1
12	26589	1	0.9998	0	0.0001504	7.521e-05	0.9997	1
13	26588	3	0.9997	0	0.0002633	9.95e-05	0.9995	0.9999
14	26585	2	0.9997	0	0.0003385	0.0001128	0.9994	0.9999
15	26583	8	0.9994	0	0.0006394	0.0001551	0.9991	0.9997
16	26575	25	0.9984	4	0.00158	0.0002438	0.9979	0.9989
17	26546	52	0.9965	1746	0.003539	0.000365	0.9958	0.9972
18	24748	137	0.9909	364	0.009075	0.0005974	0.9898	0.9921
19	24247	255	0.9805	257	0.01959	0.0008892	0.9788	0.9822
20	23735	407	0.9637	277	0.03674	0.00123	0.9614	0.9661
21	23051	560	0.9403	288	0.06103	0.001602	0.9373	0.9433
22	22203	747	0.9087	433	0.09468	0.00202	0.905	0.9123
23	21023	934	0.8683	407	0.1391	0.002489	0.864	0.8726
24	19682	1030	0.8229	389	0.1914	0.002976	0.818	0.8278
25	18263	1104	0.7731	371	0.2519	0.003488	0.7677	0.7785
26	16788	1183	0.7186	325	0.3224	0.004045	0.7128	0.7245
27	15280	1115	0.6662	303	0.3953	0.004598	0.66	0.6724
28	13862	1176	0.6097	306	0.4802	0.005221	0.6033	0.6162

time	n.risk	n.event	surv	n.censor	cumhaz	std.chaz	lower	upper
29	12380	1141	0.5535	293	0.5723	0.005891	0.5469	0.5601
30	10946	1132	0.4962	232	0.6757	0.006645	0.4896	0.503
31	9582	1048	0.442	237	0.7851	0.007454	0.4353	0.4488
32	8297	856	0.3964	203	0.8883	0.008246	0.3897	0.4031
33	7238	761	0.3547	177	0.9934	0.009084	0.3481	0.3614
34	6300	681	0.3164	128	1.102	0.009984	0.3099	0.3229
35	5491	584	0.2827	144	1.208	0.01091	0.2764	0.2891
36	4763	483	0.254	122	1.309	0.01185	0.2479	0.2603
37	4158	351	0.2326	120	1.394	0.01267	0.2266	0.2388
38	3687	347	0.2107	116	1.488	0.01364	0.2049	0.2167
39	3224	234	0.1954	115	1.56	0.01445	0.1897	0.2013
40	2875	201	0.1818	107	1.63	0.01526	0.1761	0.1875
41	2567	147	0.1713	109	1.688	0.01598	0.1658	0.1771
42	2311	132	0.1616	104	1.745	0.01673	0.1561	0.1672
43	2075	84	0.155	90	1.785	0.01731	0.1496	0.1606
44	1901	84	0.1482	105	1.829	0.01797	0.1428	0.1537
45	1712	44	0.1444	93	1.855	0.01838	0.139	0.1499
46	1575	46	0.1401	96	1.884	0.01888	0.1349	0.1456
47	1433	22	0.138	101	1.9	0.01916	0.1327	0.1435
48	1310	24	0.1355	84	1.918	0.01952	0.1302	0.1409
49	1202	18	0.1334	90	1.933	0.01984	0.1282	0.1389
50	1094	10	0.1322	85	1.942	0.02005	0.1269	0.1377

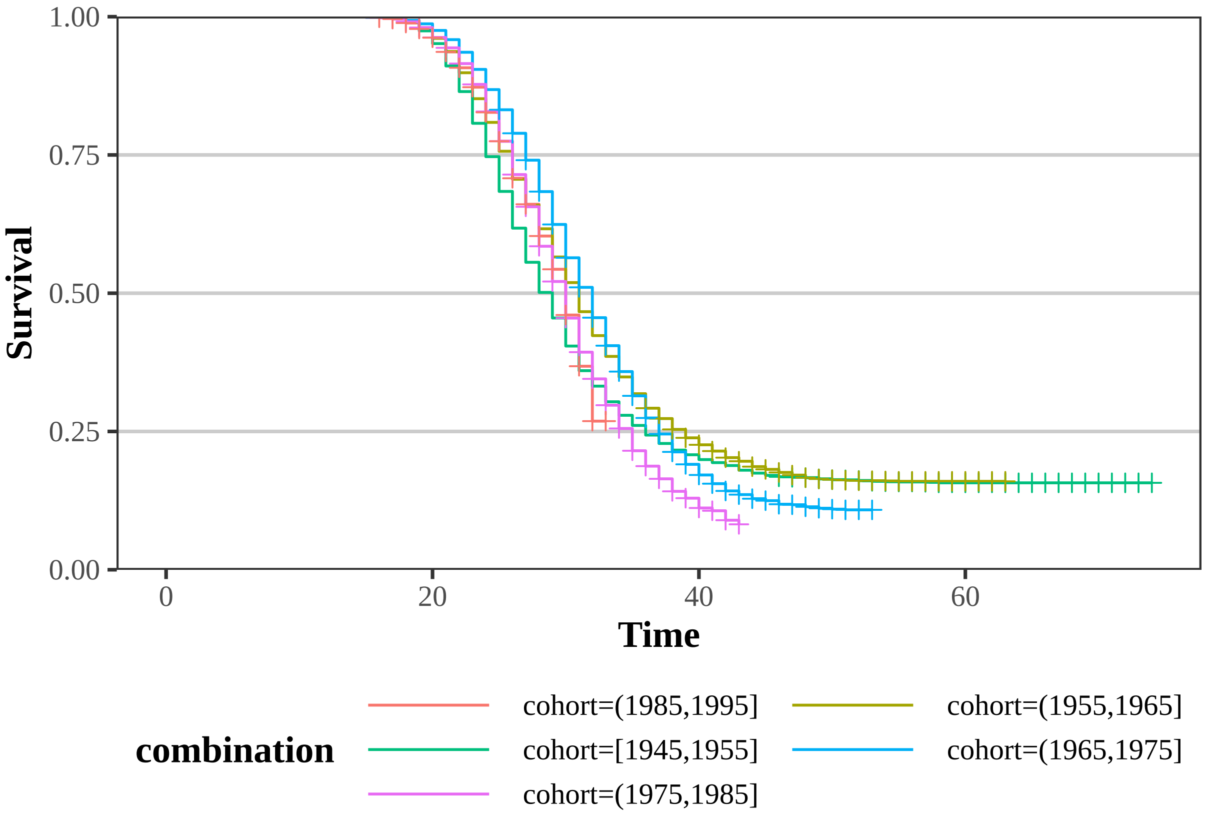
```r
ggplot(km_result, aes(time, y=  surv, ymin = lower, ymax = upper)) +
  geom_line() +
  geom_ribbon(alpha = .3)
```



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", error = "greenwood")

# 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.

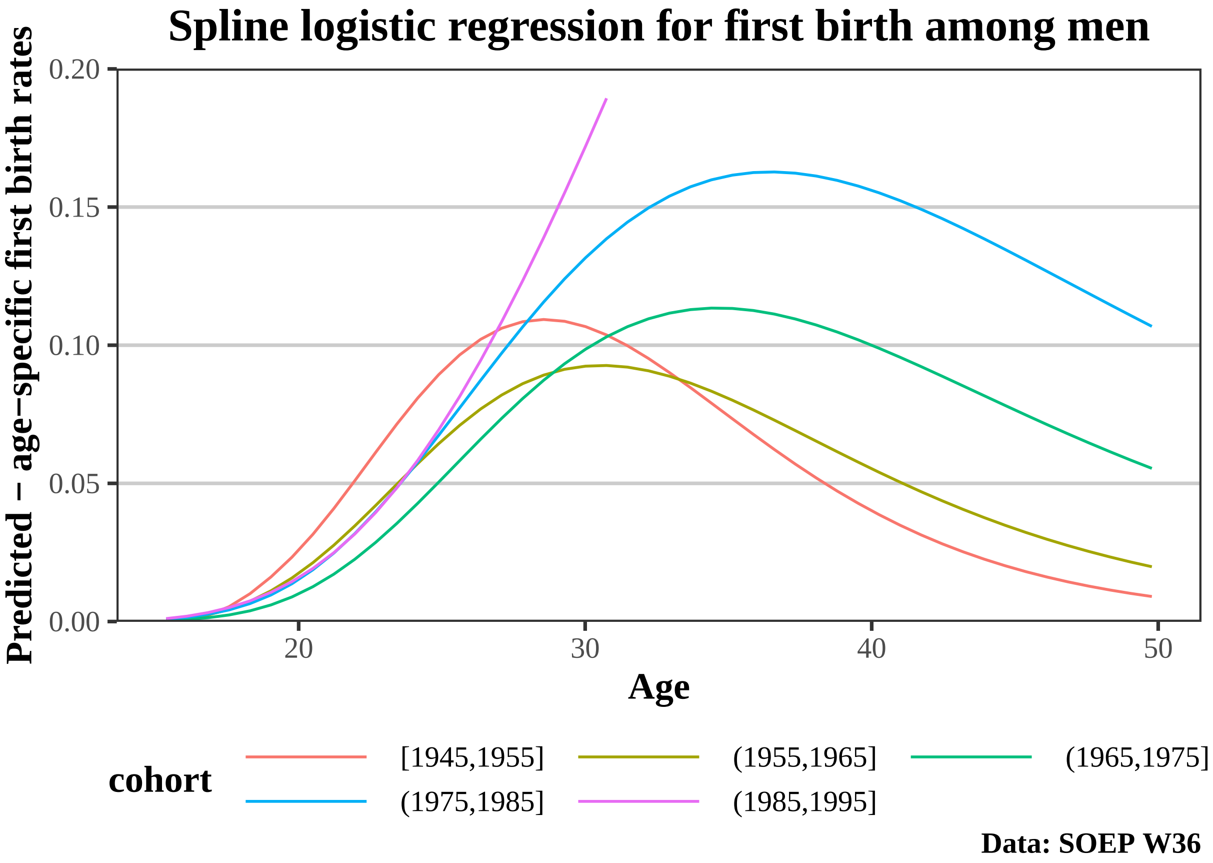
```r
if(all( isFALSE(estimate) & file.exists("Results/smoothed.Rda"))){
  # Load the model
  load("Results/smoothed.Rda")

}else{
  # Run the regression model
mod_cb <- fitSmoothHazard(Event ~ ns(log(Censoring), df = 3) * 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), df = 3) * cohort

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-26760	10704	-2.5	0.01242
ns(log(Censoring), df = 3)1	17316	6913	2.505	0.01225
ns(log(Censoring), df = 3)2	50790	20350	2.496	0.01257
ns(log(Censoring), df = 3)3	12165	4850	2.508	0.01214
cohort(1955,1965]	16380	12174	1.345	0.1785
cohort(1965,1975]	26767	10704	2.501	0.0124
cohort(1975,1985]	26762	10704	2.5	0.01242
cohort(1985,1995]	22864	11165	2.048	0.04058
ns(log(Censoring), df = 3)1:cohort(1955,1965]	-10583	7862	-1.346	0.1783
ns(log(Censoring), df = 3)2:cohort(1955,1965]	-31132	23144	-1.345	0.1786
ns(log(Censoring), df = 3)3:cohort(1955,1965]	-7424	5516	-1.346	0.1783
ns(log(Censoring), df = 3)1:cohort(1965,1975]	-17290	6913	-2.501	0.01238
ns(log(Censoring), df = 3)2:cohort(1965,1975]	-50881	20350	-2.5	0.01241
ns(log(Censoring), df = 3)3:cohort(1965,1975]	-12127	4850	-2.5	0.01241
ns(log(Censoring), df = 3)1:cohort(1975,1985]	-17291	6913	-2.501	0.01238
ns(log(Censoring), df = 3)2:cohort(1975,1985]	-50861	20350	-2.499	0.01244
ns(log(Censoring), df = 3)3:cohort(1975,1985]	-12129	4850	-2.501	0.01239
ns(log(Censoring), df = 3)1:cohort(1985,1995]	-14781	7210	-2.05	0.04037
ns(log(Censoring), df = 3)2:cohort(1985,1995]	-43436	21227	-2.046	0.04073
ns(log(Censoring), df = 3)3:cohort(1985,1995]	-10368	5058	-2.05	0.04039

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 ) |>
  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.

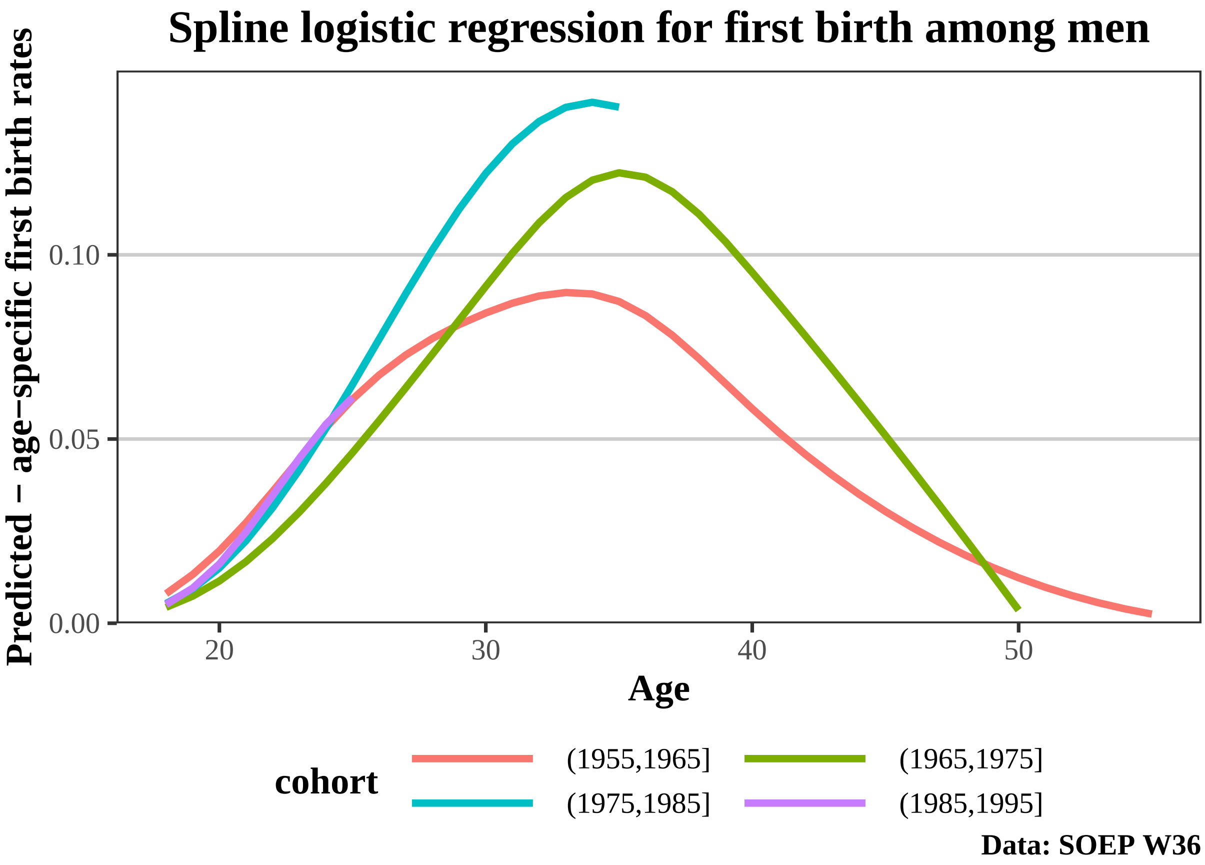
```
if(all(isFALSE(estimate) & file.exists("Results/discrete_eha_splines.Rda"))){
  # Load the data
  load("Results/discrete_eha_splines.Rda")

}else{

### 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 == "(1975,1985]" & Censoring <= 35) |
                                   (cohort == "(1985,1995]" & Censoring <= 25 ) |
                                   cohort %in% c("(1945,1955]", "(1955,1965]", "(1965,1975]"))


# 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	(1985,1995]	0	0	15	0	NA
604	(1985,1995]	15	0	16	0	NA
604	(1985,1995]	16	0	17	0	NA
1603	(1985,1995]	0	0	15	0	NA
1603	(1985,1995]	15	0	16	0	NA
1603	(1985,1995]	16	0	17	0	NA
9403	(1985,1995]	0	1	15	0	2017
9403	(1985,1995]	15	1	16	0	2017
9403	(1985,1995]	16	1	17	0	2017
9403	(1985,1995]	17	1	18	0	2017
9403	(1985,1995]	18	1	19	0	2017
9403	(1985,1995]	19	1	20	0	2017
9403	(1985,1995]	20	1	21	0	2017
9403	(1985,1995]	21	1	22	0	2017

persnr	cohort	start	sumkids	Censoring	Event	kidgeb_01
9403	(1985,1995]	22	1	23	0	2017

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

```
# 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))
```



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

A non-parametric approach

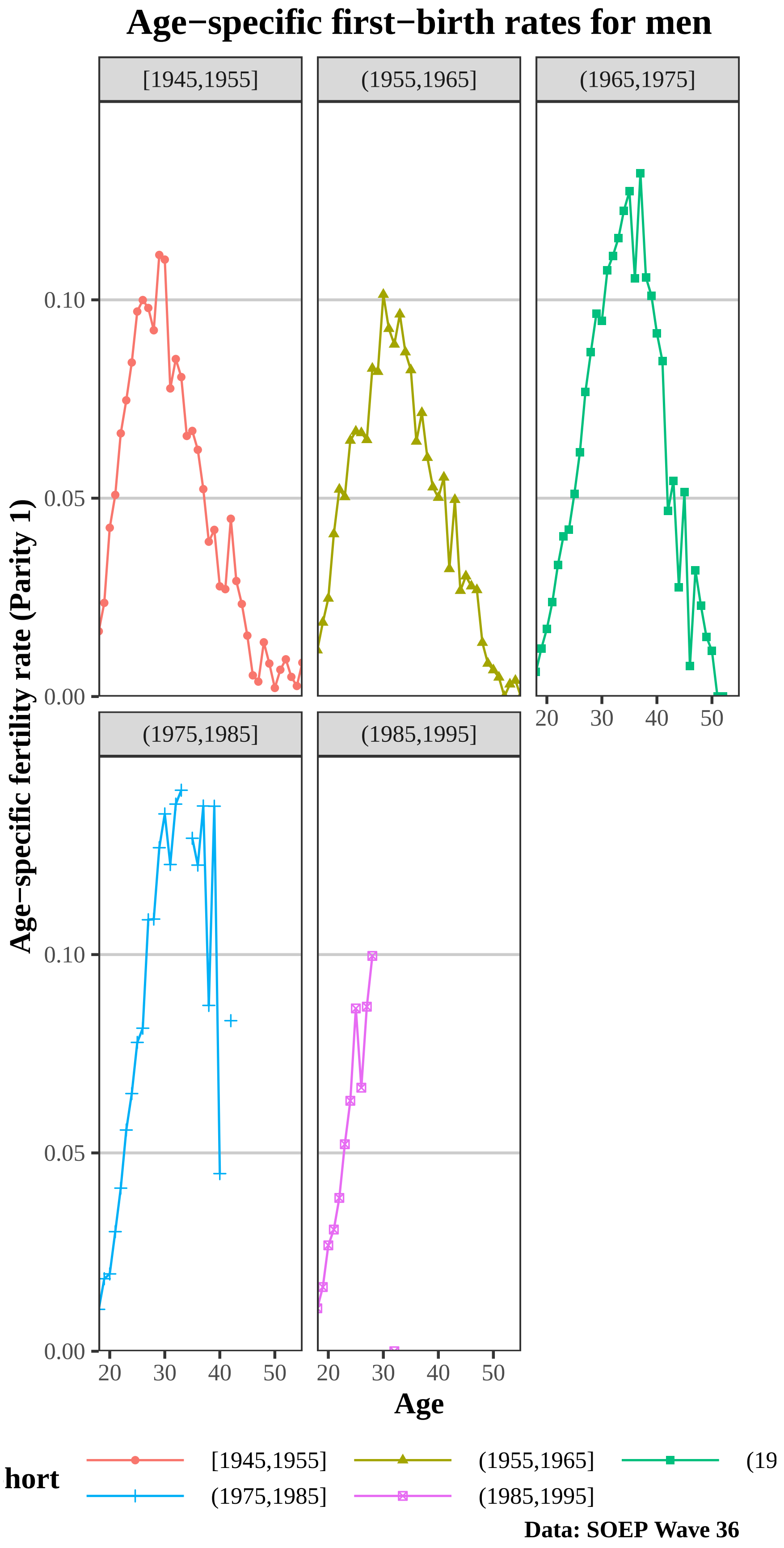
While the models are useful for incorporating covariates, they may rely on too restrictive assumptions. Therefore, we also used a non-parametric approach to estimate age-specific first birth rates.

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:

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

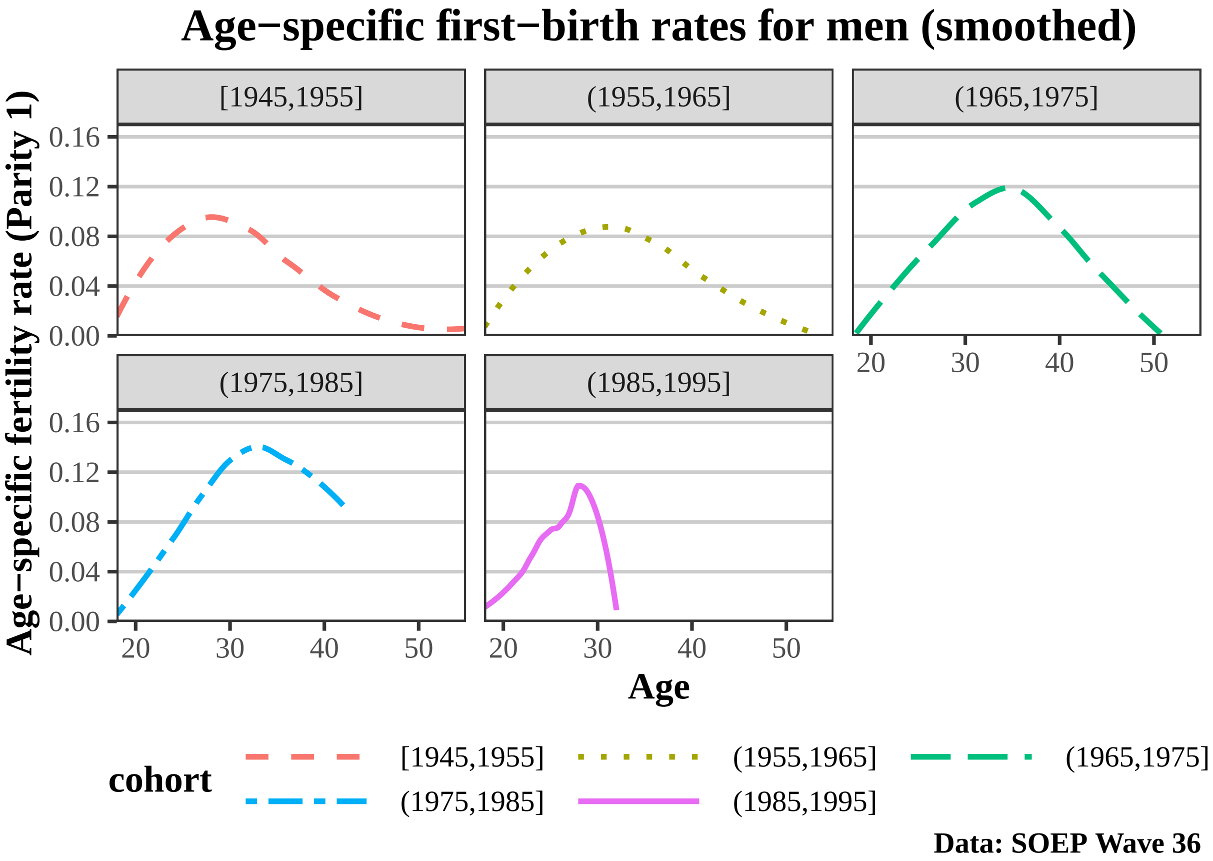
Figure @ref{fig:plot_raw} illustrates the raw age-specific first birth rates for different cohorts. Because the data is from a survey, the rates show an erratic pattern. Nonetheless, the expected bell-shape becomes apparent.

```
# 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 == "(1975,1985]" & start <= 35) |
                                   (cohort == "(1985,1995]" & start <= 25 ) |
                                   cohort %in% c("(1945,1955]", "(1955,1965]", "(1965,1975]"))

# Plot the result
plot_raw <- unparametric |>
  filter( start >= 18) |>
  ggplot(aes(start, rate, colour = cohort, group = cohort, shape = cohort)) +
    geom_line() +
    geom_point() +
    facet_wrap( ~ cohort) +
    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
[1945,1955]
(1955,1965]
(1965,1975]
(1975,1985]
(1985,1995]
Age−specific fertility rate (Parity 1)
0.00
0.05
0.10
20
30
40
50
Age
hort
[1945,1955]
(1955,1965]
(19
(1975,1985]
(1985,1995]
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(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) +
    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 result
plot_interpol
```



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.

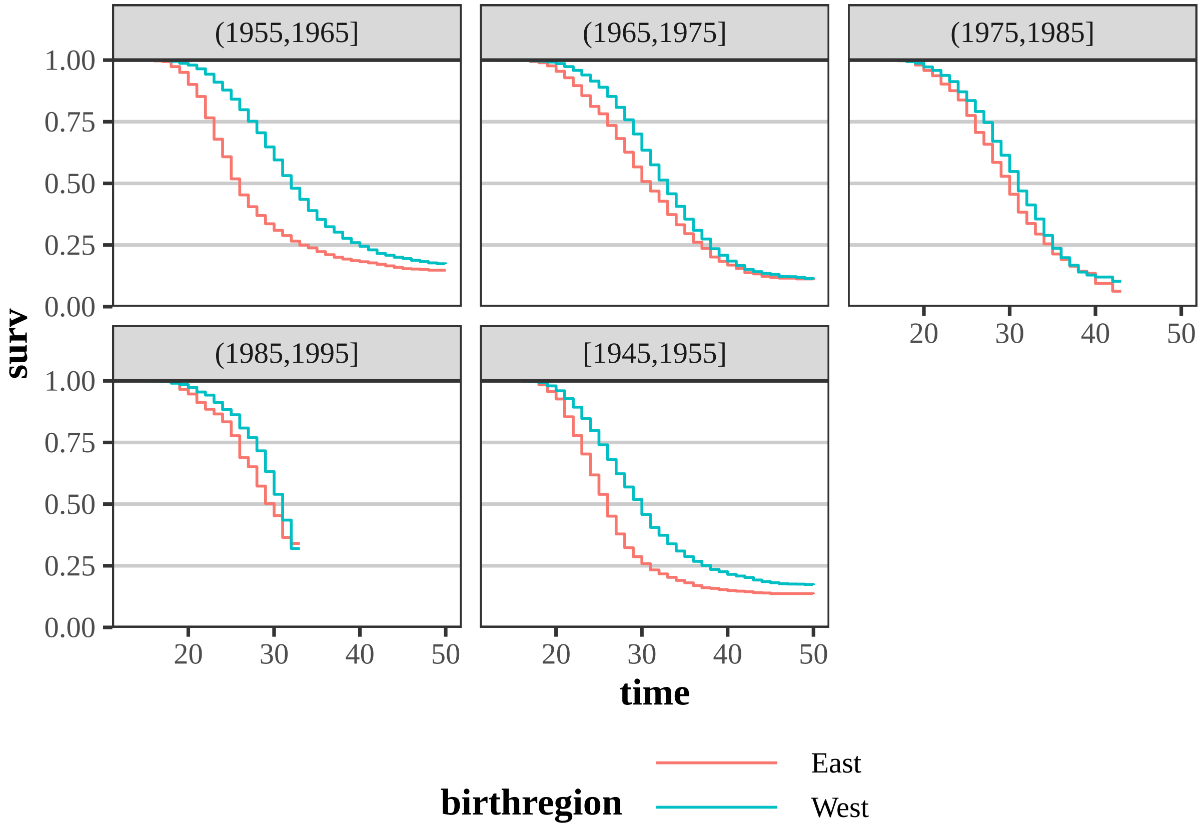
```
if(all(isFALSE(estimate) & file.exists("Data/region_spell_data.Rda"))){

  # 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, loc1989))

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

# Impute birth region if missing using location in 1989
id$birthregion <- ifelse(is.na(id$birthregion) & id$loc1989 == 2, "West",
       ifelse(is.na(id$birthregion) & id$loc1989 == 1, "East", id$birthregion))

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

# Join with birth region
fert2 <- left_join(fert2, id)

# Plot the share of missing values in birth region
ggplot(fert2, aes(x = birthregion, fill = birthregion)) +
  stat_count() +
  scale_y_continuous(expand = c(0, 0))

# Save the plot
ggsave(last_plot(), filename = "Figures/missing_birthregion.pdf")

# 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
```

Kaplan-Meier by birthregion

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

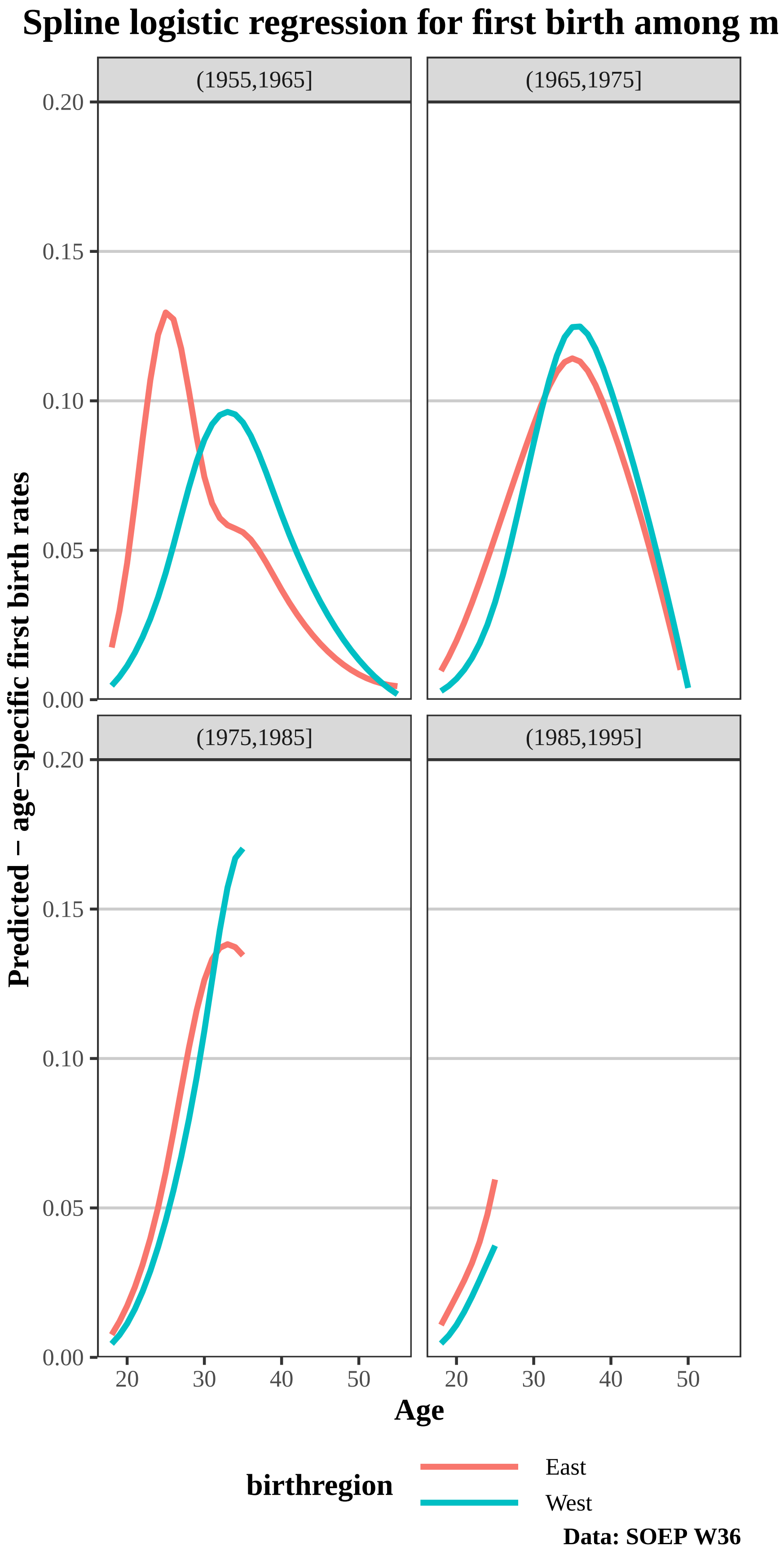
```
# Fit by cohort
km_coh_reg <- survfit(Surv(Censoring, Event) ~ cohort + birthregion,
                      data = fert2, conf.int = 0.95,
                      type = "kaplan-meier", error = "greenwood")

# 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 = birthregion, group = birthregion)) +
  geom_step() +
  scale_y_continuous(expand = c(0, 0), limits = c(0, 1)) +
  guides(colour =  guide_legend(nrow = 3, byrow = TRUE)) +
  facet_wrap( ~ cohort)
```



```
# 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 == "(1975,1985]" & Censoring <= 35) |
       (cohort == "(1985,1995]" & Censoring <= 25 ) |
       cohort %in% c("(1945,1955]", "(1955,1965]", "(1965,1975]"))

# Plot the result
ggplot(pred_data, aes(Censoring, prediction, colour = birthregion, group = birthregion)) +
  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_wrap( ~ cohort) +
  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 m

(1955,1965]

(1965,1975]

(1975,1985]

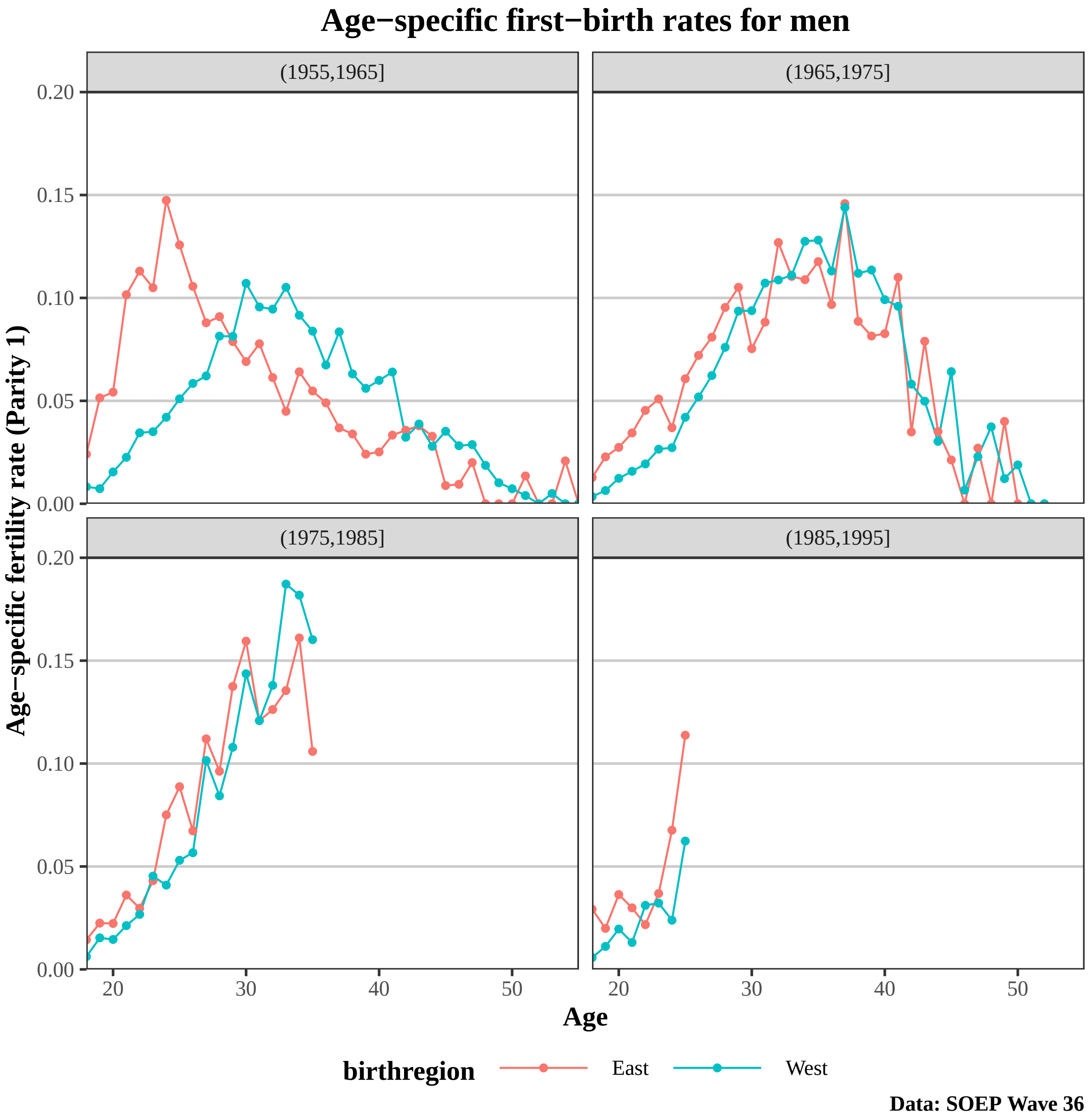
(1985,1995]

Data: SOEP W36

```
# 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).

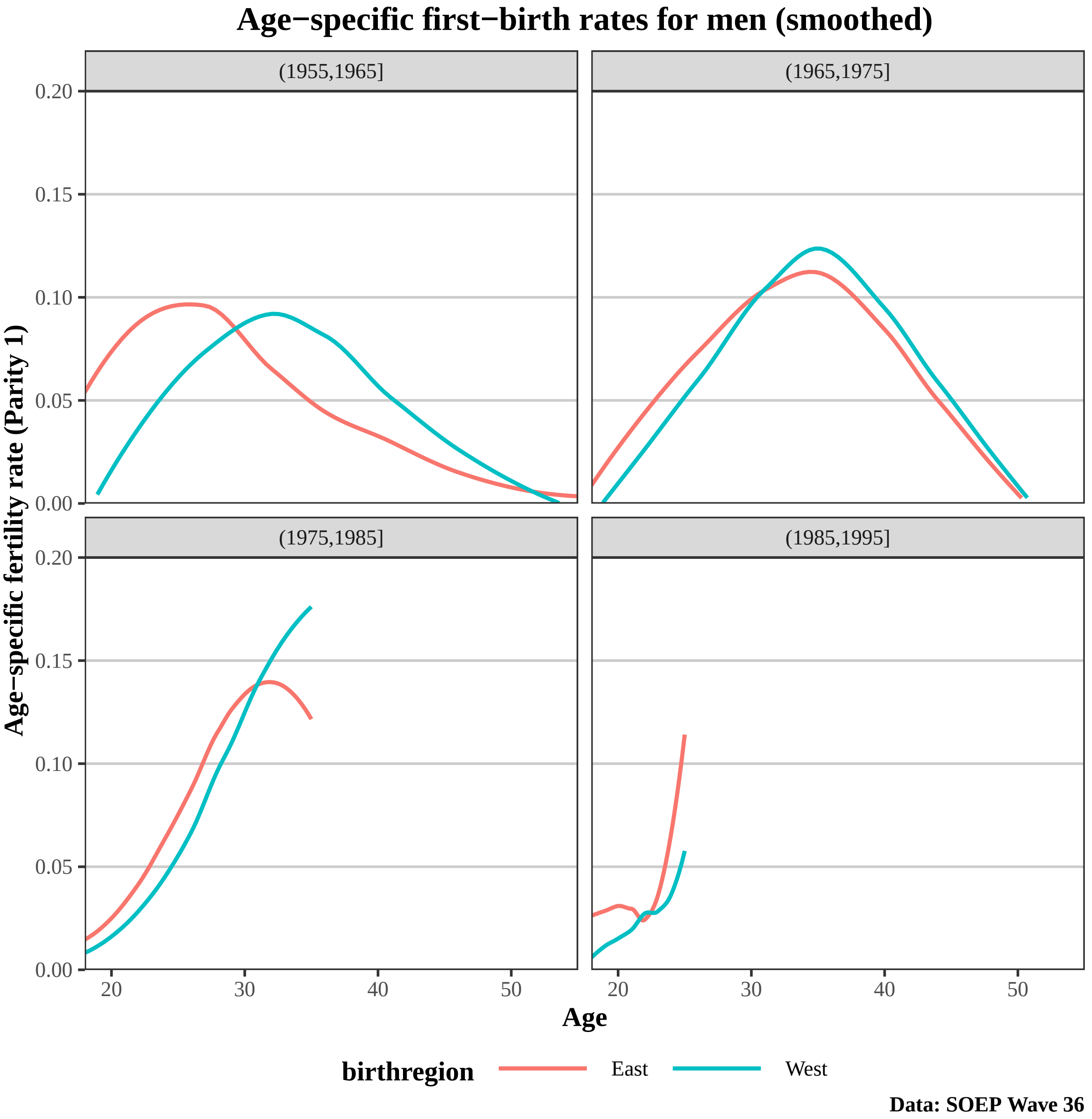
```
### Unparametric by birthregion ------------------------------------

# 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 == "(1975,1985]" & start <= 35) |
         (cohort == "(1985,1995]" & start <= 25 ) |
         cohort %in% c("(1945,1955]", "(1955,1965]", "(1965,1975]"))

# Plot the result
plot_raw_reg <- unparametric_reg |>
  filter(start >= 18) |>
  ggplot(aes(start, rate, colour = birthregion, group = birthregion)) +
    geom_line() +
    geom_point() +
    facet_wrap( ~ cohort) +
    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")


# Print the result
plot_raw_reg
```



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).

```r
# Plot interpolated
plot_interpol_reg <- unparametric_reg |>
  filter(start >= 18) |>
  ggplot(aes(start, rate, colour = birthregion, group = birthregion)) +
    geom_smooth(se = FALSE) +
    facet_wrap( ~ cohort) +
    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")
```

```
# Plot the interpolated result
plot_interpol_reg
```



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 --------------------------------
```

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

Table 5:

	Dependent variable:
	Censoring
cohort(1955,1965]	0.033
	(0.026)
cohort(1965,1975]	0.031
	(0.026)
cohort(1975,1985]	0.262***
	(0.030)
cohort(1985,1995]	1.355***
	(0.056)
Constant	3.670***
	(0.019)
Observations	17,791
Log Likelihood	−56,450.770
χ^2	983.534*** (df = 4)
Note:	*p<0.1; **p<0.05; ***p<0.01

Weibull model

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

Gaussian model

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

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

Log-normal model

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

Table 6:

	Dependent variable:
	Censoring
log(scale):1	3.662***
	(0.008)
log(shape):1	0.840***
	(0.015)
log(scale):2	3.652***
	(0.006)
log(shape):2	1.083***
	(0.013)
log(scale):3	3.606***
	(0.004)
log(shape):3	1.464***
	(0.013)
log(scale):4	3.508***
	(0.004)
log(shape):4	1.782***
	(0.017)
log(scale):5	3.448***
	(0.008)
log(shape):5	2.096***
	(0.036)
Observations	17,791
Log Likelihood	−46,933.110
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 7:

	Dependent variable:
	Censoring
cohort(1955,1965]	0.498^{**}
	(0.250)
cohort(1965,1975]	0.381
	(0.248)
cohort(1975,1985]	-0.783^{***}
	(0.281)
cohort(1985,1995]	2.509^{***}
	(0.405)
Constant	33.578^{***}
	(0.188)
Log Likelihood	−48,160.250
χ^2	72.600^{***} (df = 4)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 8:

	Dependent variable:
	Censoring
cohort(1955,1965]	0.498^{**}
	(0.250)
cohort(1965,1975]	0.381
	(0.248)
cohort(1975,1985]	-0.783^{***}
	(0.281)
cohort(1985,1995]	2.509^{***}
	(0.405)
Constant	33.578^{***}
	(0.188)
Log Likelihood	−48,160.250
χ^2	72.600^{***} (df = 4)
Note:	*p<0.1; **p<0.05; ***p<0.01

Log-logistic model

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

Table 9:

	Dependent variable:
	Censoring
cohort(1955,1965]	0.063^{***}
	(0.007)
cohort(1965,1975]	0.094^{***}
	(0.007)
cohort(1975,1985]	0.045^{***}
	(0.008)
cohort(1985,1995]	0.088^{***}
	(0.012)
Constant	3.395^{***}
	(0.005)
Observations	17,791
Log Likelihood	−45,486.930
χ^2	204.730^{***} (df = 4)
Note:	*p<0.1; **p<0.05; ***p<0.01