Relationship Durations, NSFG 2006-2015

Jeanette Birnbaum

11 April, 2017

Contents

1 Quick Summary of Results

- We can use the median duration of active Spouse/Cohab ties to decently represent the Spouse/Cohab relationship length distribution with a geometric curve.
- We can do the same for the Other/Casual relationships, since the median of active ties matches the median we get after doing a Kaplan-Meier adjustment for censoring and left-truncation. The geometric will not capture the right tail of this distribution, but it will do a good job with the left half.
- We need a separate network to get the frequency of one-time relationships right. No geometric for the Other/Casual network will naturally yield the right order of magnitude of one-time relationships.
- We have yet to explore whether these findings vary significantly by race.

2 Detailed Summary of Results

2.1 All relationships

We use NSFG data 2006-2015 which contains 38,582 egos and 42,820 partnerships: 33,080 first partnerships, 6,722 and 3,018 third partnerships. If we categorize Spouse/Cohabs as "main" relationships and Others as "Casual", 49% of relationships among female egos are Casual versus 60% among male egos.

2.2 Active ties, raw data

Among the 28,486 active ties that are not one-times, 36% of ties among female egos are Casual versus 40% of ties among males. Mean ages of these ties in months are 33.22 and 29.01 among female and male Casual ties and 109.55 and 104.92 among female and male Spouse/Cohab ties.

We consider two ways of identifying "active" one-time ties - using those occuring in the month of the interview, or a monthly rate over the last 12 months. The resulting percentages of active ties that are one-time are 1.08 and 0.77 respectively. However, both of these estimates do not account for the left-truncation of the sampling scheme, i.e 2nd and 3rd partners were only reported if they were current within the 12 month period prior to the interview date.

Plots of the full distribution of relationship ages indicate that exponential distributions defined by the data median fit the data reasonably well. These plots do not take into account the left-truncation of the data, but as there is left truncation only for the 2nd and 3rd, they are minimally biased for the Spouse and Cohab relationships.

2.3 All ties, adjusting for censoring (active ties) and left-truncation

Since all Spouses/Cohabs are censored and only 4.46% of them are left-truncated, the adjusted analysis reflects the impact of left-truncation and censoring on the Other (Casual) partners.

The adjustment shifts the distribution towards shorter relationship lengths. The median relationship length in months decreases from 317 to 293 among all ties and from 21 to 14 among the Casual ties. Note that the (unadjusted) median duration among *active* Casual ties is 14!

Similarly, the percent of one-time ties increases from 5.6% to 6.1% among all ties and from 10.4% to 11.5% among Casual ties.

Defining exponential curves using the adjusted median and mean of all ties and, separately, Casual ties only, we see that the exponential distribution does not fit the empirical data very well. Both the distribution for all ties and the distribution for Casual ties clearly have non-constant hazards of relationship dissolution. In particularly, the exponentials expect fewer than 1% of all ties to be one-times, significantly underestimating the observed one-times.

2.4 Using the mean of active ties to simulate relationships and mimic NSFG sampling

Using Steve's simulation, we see that yes, the mean of active, sampled ties does represent the true mean of the full distribution well for geometrically-distributed relationships. In addition, the sampled active ties appear to preserve the 1st and 3rd quartiles as well.

We see again through sampling that the NSFG data are not perfectly geometrically distributed. However, the correspondance is pretty decent for Spouses/Cohabs when we look at coarse duration categories, especially when the exponential is based on the data median. For the Other/Casual relationships, the simulated data are quite good for relationships under 12 months. For longer durations, the NSFG has more longer-term (>5 years) relationships than in the simulated data.

Note All analyses are unweighted.

3 Data prep

3.1 Exclusions

- See R/data.R for the prep of the nsfg object
- We subset to "sexindi==1". Formerly we did "HADSEX=='YES, R EVER HAD INTERCOURSE' ", but unlike HADSEX, sexindi is not restricted to vaginal sex.

```
#-----------------------------------------------------------------------------
# Exclusions
#-----------------------------------------------------------------------------
data(nsfg)

nsfgs <- exclude_and_report(nsfg, list("sexindi==1"))
```

```
##   Selections Number of Rows Cases Deleted
## 1       None          43303             0
## 2 sexindi==1          38582          4721
```

```
Nego <- nrow(nsfgs)
```

3.2 Reshape into edgelist

Reshape into the edgelist, deleting empty partnerships based on missingness in the variables necessary for this analysis

```
#-----------------------------------------------------------------------------
# Reshape into edgelist
#-----------------------------------------------------------------------------
dfl <- reshape_edgelist(nsfgs, delete_empty = c("active", "rel", "len"), all = FALSE)
```

```
## Warning in reshape_edgelist(nsfgs, delete_empty = c("active", "rel", "len"), :
## In reshape_edgelist, creating ego ID variable using row number
```

```
Npship <- nrow(dfl)
```

4 All relationships

```
#-----------------------------------------------------------------------------
# Alters
#-----------------------------------------------------------------------------
(nalter <- table(dfl$alter))
```

```
##
##     1     2     3
## 33080  6722  3018
```

```r
#-------------------------------------------------------------------------------
# Distribution of alters conditional on relationship type
#-------------------------------------------------------------------------------
nalterrel <- table(dfl$rel, dfl$alter)
print(100 * prop.table(nalterrel, margin = 1), digits = 0)
```

```
##
##              1   2   3
##   Spouse    99   1   0
##   Cohab    100   0   0
##   Other     62  26  12
```

```r
#-------------------------------------------------------------------------------
# Group Spouses and Cohabs in a 'main' variable of relationship type
#-------------------------------------------------------------------------------
# Define main = Spouse or Cohab, and Casual = Other NEW 4/10/17: reassign Former
# Spouses/Cohabs to Spouse/Cohab for analyzing durations
dfl <- within(dfl, {
    main <- "NA"
    main[rel == "Other"] <- "Other"
    #### OLD CATEGORIES main[rel=='Spouse' | rel=='Cohab'] <- 'Spouse/Cohab' NEW
    #### CATEGORIES
    main[optype == "Current spouse" | optype == "Current cohab"] <- "Spouse/Cohab"
    main[(optype == "Former spouse" | optype == "Former cohab") & active == "not active"] <- "Spouse/Col

})

# Crosstabs
kable(with(dfl, table(optype, active, useNA = "ifany")))
```

	not active	active	DLS not in last yr
Current spouse	0	12809	0
Current cohab	0	4998	0
Former spouse	171	73	0
Former cohab	1606	1638	0
Other	12285	9240	0

```r
kable(with(subset(dfl, active == "not active"), table(main, optype, useNA = "ifany")))
```

	Current spouse	Current cohab	Former spouse	Former cohab	Other
Other	0	0	0	0	12285
Spouse/Cohab	0	0	171	1606	0

```r
kable(with(dfl, table(main, active)))
```

	not active	active	DLS not in last yr
Other	12285	10951	0
Spouse/Cohab	1777	17807	0

```
#-------------------------------------------------------------------------------
# Relationship type by sex
#-------------------------------------------------------------------------------
crosstab(factor(dfl$main), dfl$sex, prop.c = TRUE, dnn = c("Relationship Type", "Sex"),
    missing.include = TRUE, cell.layout = FALSE, total.c = TRUE, plot = FALSE)
```

```
##
## ===========================================
##                      Sex
## Relationship Type    female    male   Total
## -------------------------------------------
## Other                 11030   12206   23236
## col %                  49.3    59.7
## -------------------------------------------
## Spouse/Cohab          11360    8224   19584
## col %                  50.7    40.3
## -------------------------------------------
## Total                 22390   20430   42820
##                        52.3    47.7
## ===========================================
```

```
# Re-display the proportion of Other (Casual) relationships
(casuals <- tab_means(transform(dfl, cas = ifelse(main == "Other", 1, 0)), row = "sexindi",
    var = "cas"))
```

```
##   sexindi N_female N_male mean_female mean_male
## 1       1    22390  20430        0.49       0.6
```

```
casualsPerc <- casuals * 100
```

5 Active ties, unadjusted for left-truncation

5.1 Reminder of the active variable

You can see the original levels that have been recoded to NA.

```
pxtable(table(dfl$active))
```

not active	active	DLS not in last yr
14062	28758	0

5.2 Active, not-one-time ties

5.2.1 Define and examine

```
#-------------------------------------------------------------------------------
# Define population
#-------------------------------------------------------------------------------
act1 <- exclude_and_report(dfl, list("active=='active'", "once==0"))
```

```
##         Selections Number of Rows Cases Deleted
## 1             None          42820             0
## 2 active=='active'          28758         14062
## 3          once==0          28486           272
```

```r
Nact1 <- nrow(act1)
```

```r
#-------------------------------------------------------------------------------
# Relationship type by sex among active population
#-------------------------------------------------------------------------------
(casualsact1 <- crosstab(act1$main, act1$sex, prop.c = TRUE, dnn = c("Relationship type among actives",
    "Sex"), missing.include = TRUE, cell.layout = FALSE, total.c = TRUE, plot = FALSE))
```

```
##
## =========================================================
##                                    Sex
## Relationship type among actives    female    male   Total
## ---------------------------------------------------------
## Other                               5809    4874   10683
## col %                               35.8    39.8
## ---------------------------------------------------------
## Spouse/Cohab                       10417    7386   17803
## col %                               64.2    60.2
## ---------------------------------------------------------
## Total                              16226   12260   28486
##                                       57      43
## =========================================================
```

```r
casualsact1Perc <- round(100 * casualsact1$prop.col["Other", ])
```

5.2.2 Mean age of active ties that are not one-times, in months

```r
#-------------------------------------------------------------------------------
# Mean age of active ties by rel and sex
#-------------------------------------------------------------------------------
print(tab_means(act1, row = "rel", var = "len"), digits = 0)
```

```
##      rel N_female N_male mean_female mean_male
## 1 Spouse     7487   5320         128       122
## 2  Cohab     2930   2066          62        60
## 3  Other     5809   4874          33        29
```

```r
#-------------------------------------------------------------------------------
# Mean age of active ties by main and sex
#-------------------------------------------------------------------------------
age1 <- tab_means(act1, "len")
print(age1, digits = 0)
```

```
##           main N_female N_male mean_female mean_male
## 1        Other   6.e+03   4874          33        29
## 2 Spouse/Cohab   1.e+04   7386         110       105
```

5.3 Active, one-time ties

5.3.1 Using last-month definition

```
#-------------------------------------------------------------------------------
# Define active one-times using the last-month definition
#-------------------------------------------------------------------------------
act0 <- exclude_and_report(dfl, list("once==1", "dls==0"))

##   Selections Number of Rows Cases Deleted
## 1       None          42820             0
## 2    once==1           2669         40151
## 3     dls==0            311          2358

Nact0 <- nrow(act0)
```

5.3.2 Using average over last 12 months definition

```
act0r <- exclude_and_report(dfl, list("once==1", "dls<=12"))

##   Selections Number of Rows Cases Deleted
## 1       None          42820             0
## 2    once==1           2669         40151
## 3    dls<=12           2669             0

# Divide by 12
(Nact0rate <- round(nrow(act0r)/12))

## [1] 222
```

5.4 Create population of all active ties

```
#-------------------------------------------------------------------------------
# Create active population = active not-one-times + active one-times (last-month
# def)
#-------------------------------------------------------------------------------
pop <- rbind(act0, act1)

# N active ties total
(Nact <- nrow(pop))

## [1] 28797
```

5.5 All active ties

5.5.1 Proportion that are one-times

```
#-------------------------------------------------------------------------------
# Compare two definitions
#-------------------------------------------------------------------------------
(propot_unadj <- data.frame(`Last month` = round(100 * Nact0/Nact, 2), `Rate over last 12 mos` = round(
    Nact0rate/Nact, 2)))
```

```
##   Last.month Rate.over.last.12.mos
## 1       1.08                  0.77
```

5.5.2 Mean age in months

```
#--------------------------------------------------------------------------------
# Mean age of active ties by sex
#--------------------------------------------------------------------------------
(meanall <- tab_means(act1, row = "sexindi", var = "len"))
```

```
##   sexindi N_female N_male mean_female mean_male
## 1       1    16226  12260       82.22     74.74
```

```
#--------------------------------------------------------------------------------
# Mean age of active ties by main and sex
#--------------------------------------------------------------------------------
(meanmain <- tab_means(act1, var = "len"))
```

```
##           main N_female N_male mean_female mean_male
## 1        Other     5809   4874       33.22     29.01
## 2 Spouse/Cohab    10417   7386      109.55    104.92
```

```
#--------------------------------------------------------------------------------
# Mean age of active ties by race and sex
#--------------------------------------------------------------------------------
(meanrace <- tab_means(act1, row = "race", var = "len"))
```

```
##                 race N_female N_male mean_female mean_male
## 1           Hispanic     3750   2878       89.91     78.41
## 2 Non-Hispanic White     8163   6127       85.55     77.38
## 3 Non-Hispanic Black     3375   2517       65.38     63.54
## 4 Non-Hispanic Other      938    738       83.15     76.77
```

5.5.3 Full distribution of relationship ages (in months)

The following plots have relationship ages plotted as bars, with x's indicating the expected counts from an exponential defined by the data median.

```
#--------------------------------------------------------------------------------
# Bin duration (1st bin will be one-times only)
#--------------------------------------------------------------------------------
pop <- transform(pop, lencat1 = cut(len, breaks = c(0, 0.5, 1, 6, 12, 60, 120, 999),
    right = FALSE))
with(pop, table(lencat1, once, useNA = "ifany"))
```

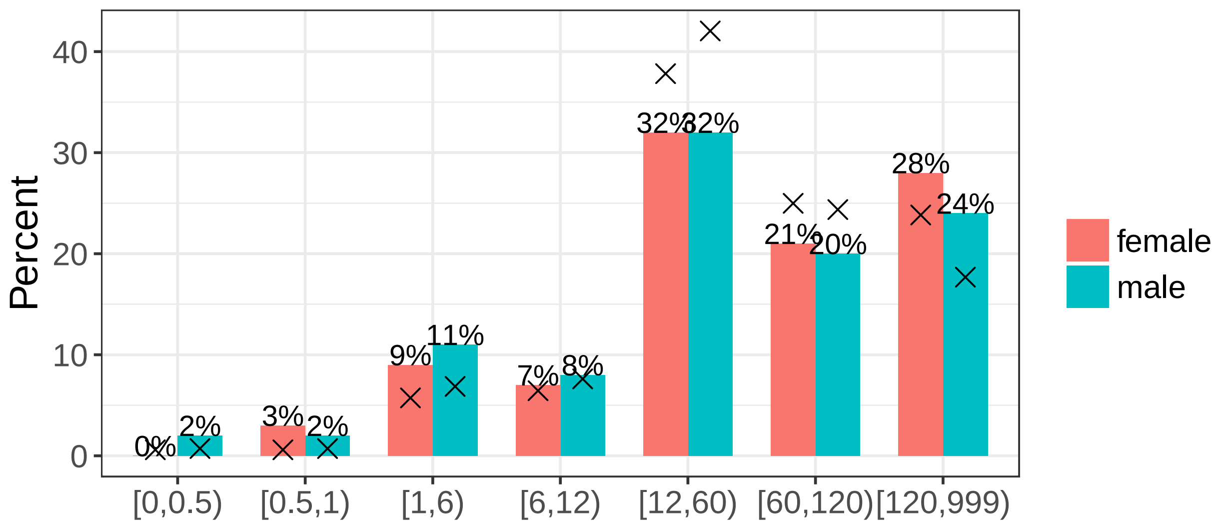
```
##            once
## lencat1        0    1
##   [0,0.5)      0  311
##   [0.5,1)    767    0
##   [1,6)     2775    0
##   [6,12)    2059    0
##   [12,60)   9295    0
##   [60,120)  6037    0
##   [120,999) 7553    0
```

5.5.3.1 All ties

```
p <- plot_categorical(pop, var = "lencat1", group = "sex", panel = NULL, yperc = TRUE,
    ylab = "Percent")
```

```
##
## Sum of bin proportions is  1
## Sum of bin proportions is  1
```

```
p
```

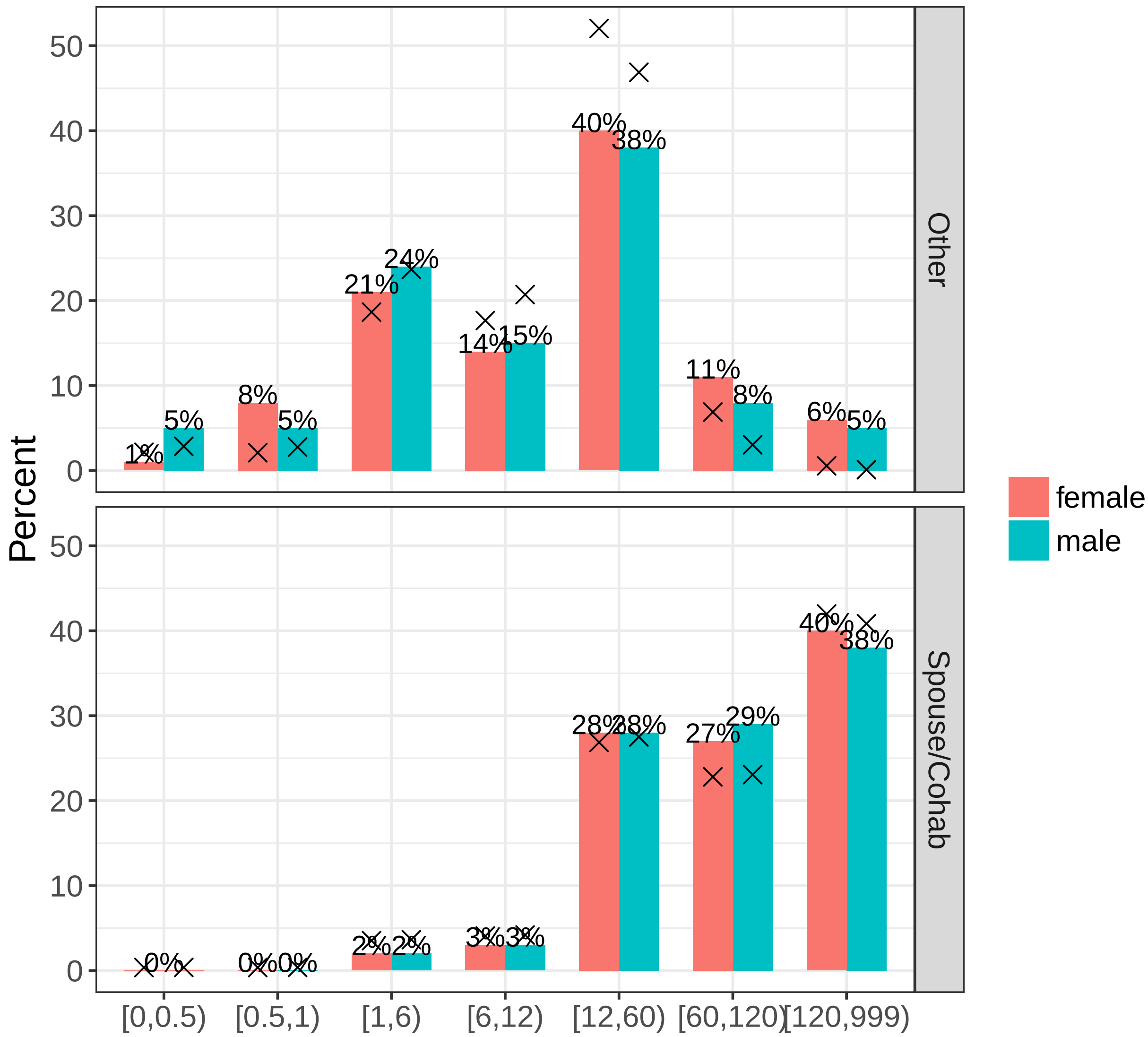


5.5.4 By Main (Spouse/Cohab) vs Other (Casual)

```
p <- plot_categorical(pop, "lencat1", "sex", "main", yperc = TRUE, ylab = "Percent") +
    facet_grid(panel ~ .)
```

```
##
## Sum of bin proportions is  1
## Sum of bin proportions is  1
## Sum of bin proportions is  1
## Sum of bin proportions is  1
```

```
p
```

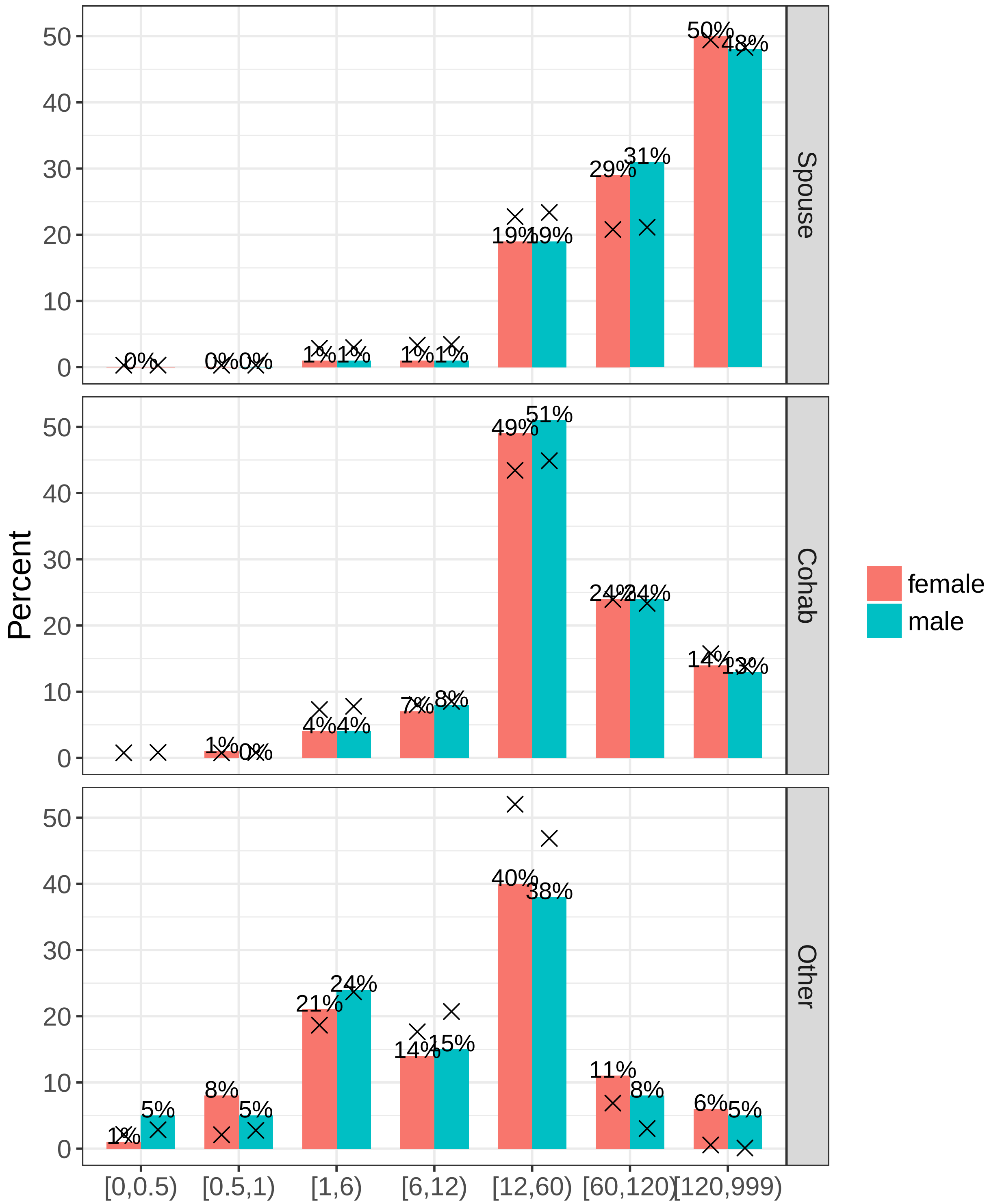


5.5.5 By Spouse vs Cohab vs Other

```
p <- plot_categorical(pop, "lencat1", "sex", "rel", yperc = TRUE, ylab = "Percent") +
    facet_grid(panel ~ .)
```

```
##
## Sum of bin proportions is  1
## Sum of bin proportions is  1
## Sum of bin proportions is  1
## Sum of bin proportions is  1
## Sum of bin proportions is  1
## Sum of bin proportions is  1
```

```
p
```



6 Kaplan-Meier adjustment for censoring and left-truncation

To adjust for left-truncation, we look not at only active ties but all partnerships and compare the unadjusted non-parametric Kaplan Meier (K-M) curve to the K-M curve adjusted for left-truncation. See SHAMP Issue 21 for a more thorough discussion and references: https://github.com/statnet/SHAMP/issues/21.

In the NSFG, the 1st partnership was documented without any left truncation. The 2nd and 3rd partnerships, however, were recorded only if the repspondent reported more than 1 sexual partner in the last 12 months. So we consider those partnerships left-truncated, as the recall window excludes partnerships that were already completed by 12 months prior to the interview date.

- **Time scale** The time scale will be relationship length. *Time=0* indicates relationship start and *Time=x* is relationship end.
- **Censoring** Active partnerships (including the one-times) will be considered censored. Note that the input *nsfg* dataset has been prepared such that self-reported active partnerships with a *dls>12* mos prior to interview are considered inactive.
- **time variable, unadjusted analysis** This is the *len* variable, representing the observed partnership duration
- **time variable, adjusted analysis (t1)** This represents the age of the partnership at the start of measurement time, and is 0 for the 1st partnerships since they were not left-censored. For the 2nd and 3rd partnerships, it is *len-12*, because *time=12* months prior to the interview date is the age of the partnership at the truncation time point
- **time2 variable, adjusted analysis (t2)** This is the *len* variable, representing the observed partnership duration, except for the one-times or those with *len=12*. In both cases, *len-t1=0* and that is not allowed for the adjusted analysis. As a workaround, we add a length of 1 day (or 1/30.5 months) to the len variable for these cases.

6.1 Prepare t1, t2 and censoring indicator ("event")

```
library(survival)
#------------------------------------------------------------------------------
# Create t1, t2 and event (requires no missingness in active and len variables)
#------------------------------------------------------------------------------
dfl <- transform(dfl, event = ifelse(active == "active", 0, 1), t1 = ifelse(alter ==
    1 | len < 12, 0, len - 12), t2 = len)
#------------------------------------------------------------------------------
# Inspect when t2-t1=0: it's only the one-times
#------------------------------------------------------------------------------
with(subset(dfl, t2 - t1 == 0), table(once, exclude = NULL))
```

```
## once
##    1 <NA>
## 2669    3
```

```
#------------------------------------------------------------------------------
# t1==t2 for one-time relationships But t2 must be greater than t1, so add 1 day
# to t2 for the zeroes (t1==t2)
#------------------------------------------------------------------------------
dfl[(dfl$t2 - dfl$t1) == 0, "t2"] <- 1/30.5
#------------------------------------------------------------------------------
# Examine left trunction: only alters 2 and 3 are left-truncated
#------------------------------------------------------------------------------
with(dfl, table(alter, truncated = (t1 != 0)))
```

```
##      truncated
## alter FALSE  TRUE
##     1 33080     0
##     2  4432  2290
##     3  2141   877
```

```r
#-------------------------------------------------------------------------------
# Left truncation is primarily among Casual (Other) relationships
#-------------------------------------------------------------------------------
with(dfl, table(main, truncated = (t1 != 0)))
```

```
##                truncated
## main             FALSE  TRUE
##   Other          20943  2293
##   Spouse/Cohab 18710   874
```

```r
(perc.main.truncated <- round(mean(subset(dfl, main == "Spouse/Cohab")$t1 != 0) *
    100, 2))
```

```
## [1] 4.46
```

```r
#-------------------------------------------------------------------------------
# Censoring (event==0) applies to all Spouses/Cohabs, because they were presumed
# active
#-------------------------------------------------------------------------------
with(dfl, table(main, event))
```

```
##                event
## main                0     1
##   Other         10951 12285
##   Spouse/Cohab 17807  1777
```

All Spouse and Cohab relationships are active/censored, which means that we can't get a K-M estimate of the length.

6.2 Survivor functions

```r
#-------------------------------------------------------------------------------
# Set the undajusted versus adjusted survival objects
#-------------------------------------------------------------------------------
unS <- with(dfl, Surv(time = len, event = event))
adS <- with(dfl, Surv(time = t1, time2 = t2, event = event, type = "counting"))
adSsc <- with(subset(dfl, main == "Spouse/Cohab"), Surv(time = t1, time2 = t2, event = event,
    type = "counting"))
adSo <- with(subset(dfl, main == "Other"), Surv(time = t1, time2 = t2, event = event,
    type = "counting"))

#-------------------------------------------------------------------------------
# All ties: survivor functions
#-------------------------------------------------------------------------------
unadj <- survfit(Surv(time = len, event = event) ~ 1, data = dfl)
unadj <- survfit(unS ~ 1, data = dfl)
adj <- survfit(Surv(time = t1, time2 = t2, event = event) ~ 1, data = dfl)
adj <- survfit(adS ~ 1, data = dfl)
#-------------------------------------------------------------------------------
# Spouse/Cohab vs Other: survivor functions
#-------------------------------------------------------------------------------
unadj.sco <- survfit(unS ~ strata(main), data = dfl)
```

```r
adj.sco <- survfit(adS ~ strata(main), data = dfl)
#-------------------------------------------------------------------------------
# Spouse/Cohab only
#-------------------------------------------------------------------------------
unadj.sc <- survfit(Surv(time = len, event = event) ~ 1, data = dfl, subset = (main ==
    "Spouse/Cohab"))
adj.sc <- survfit(Surv(time = t1, time2 = t2, event = event, type = "counting") ~
    1, data = dfl, subset = (main == "Spouse/Cohab"))
#-------------------------------------------------------------------------------
# Other only (since Spouse/Cohabs are censored)
#-------------------------------------------------------------------------------
unadj.other <- survfit(Surv(time = len, event = event) ~ 1, data = dfl, subset = (main ==
    "Other"))
adj.other <- survfit(Surv(time = t1, time2 = t2, event = event, type = "counting") ~
    1, data = dfl, subset = (main == "Other"))
```

6.3 Median ages of ties

```r
#-------------------------------------------------------------------------------
# All ties: median relationship ages
#-------------------------------------------------------------------------------
(kmmed.all <- data.frame(`Unadjusted Median` = quantile(unadj, probs = 0.5)$quantile,
    `Adjusted Median` = quantile(adj, probs = 0.5)$quantile, check.names = FALSE))

##    Unadjusted Median Adjusted Median
## 50               317             293

#-------------------------------------------------------------------------------
# Spouses/Cohabs
#-------------------------------------------------------------------------------
(kmmed.sc <- data.frame(`Unadjusted Median` = quantile(unadj.sc, probs = 0.5)$quantile[1],
    `Adjusted Median` = quantile(adj.sc, probs = 0.5)$quantile[1], check.names = FALSE))

##    Unadjusted Median Adjusted Median
## 50                NA              NA

#-------------------------------------------------------------------------------
# Other/Casual: median relationship ages (can't get it for Spouse/Cohabs b/c
# censoring)
#-------------------------------------------------------------------------------
(kmmed.other <- data.frame(`Unadjusted Median` = quantile(unadj.other, probs = 0.5)$quantile[1],
    `Adjusted Median` = quantile(adj.other, probs = 0.5)$quantile[1], check.names = FALSE))

##    Unadjusted Median Adjusted Median
## 50                21              14
```

6.4 Proportion of one-time ties

```r
#-------------------------------------------------------------------------------
# Compute proportion one-times (remember they have duration 1 day = 1/30.5 mos)
#-------------------------------------------------------------------------------
(prop.ot.km.all <- c(1 - summary(unadj, times = c(1/30.5))$surv, 1 - summary(adj,
    times = c(1/30.5))$surv))
```

```
## [1] 0.05616534 0.06065115
```

```r
(prop.ot.km.sc <- c(1 - summary(unadj.sc, times = c(1/30.5))$surv, 1 - summary(adj.sc,
    times = c(1/30.5))$surv))
```

```
## [1] 0 0
```

```r
(prop.ot.km.other <- c(1 - summary(unadj.other, times = c(1/30.5))$surv, 1 - summary(adj.other,
    times = c(1/30.5))$surv))
```

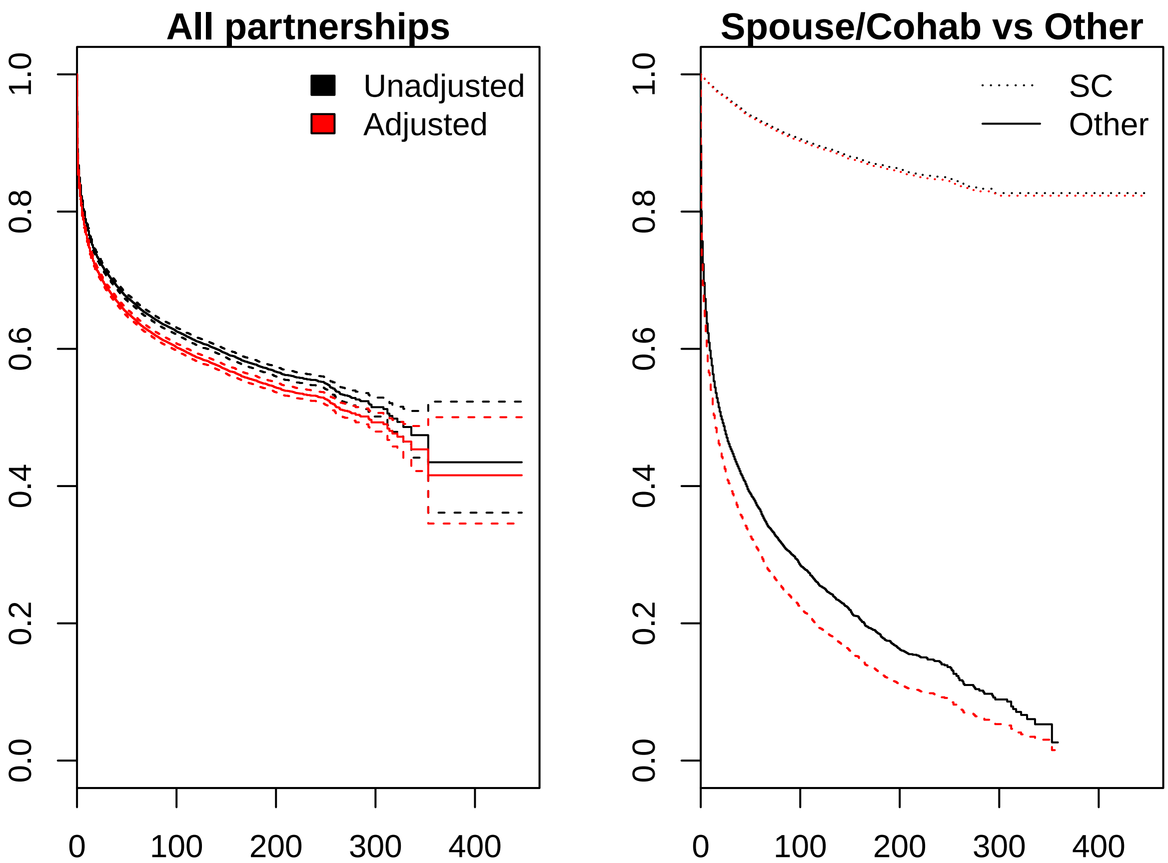
```
## [1] 0.1035032 0.1148355
```

```r
#--------------------------------------------------------------------------------
# Display percent one-times
#--------------------------------------------------------------------------------
(perc.ot.km <- data.frame(Estimate = c("Unadjusted", "Adjusted"), `All ties` = round(100 *
    prop.ot.km.all, 1), `Other/Casual` = round(100 * prop.ot.km.other, 1), check.names = FALSE))
```

```
##     Estimate All ties Other/Casual
## 1 Unadjusted      5.6         10.4
## 2   Adjusted      6.1         11.5
```

6.5 Plots of full duration distribution

```r
#--------------------------------------------------------------------------------
# All ties
#--------------------------------------------------------------------------------
par(mfrow = c(1, 2), mar = c(2, 3, 1, 1) + 0.1)
plot(unadj, main = "All partnerships")
lines(adj, col = "red")
legend("topright", legend = c("Unadjusted", "Adjusted"), fill = c("black", "red"),
    bty = "n")
#--------------------------------------------------------------------------------
# Spouse/Cohab vs Other
#--------------------------------------------------------------------------------
plot(unadj.sco, main = "Spouse/Cohab vs Other", lty = c(1, 3))
lines(adj.sco, col = "red", lty = 2:3)
legend("topright", legend = c("SC", "Other"), lty = c(3, 1), bty = "n")
```



6.6 Comparison of adjusted K-M distribution to parametric distributions

6.6.1 Exponentials based on mean verus median

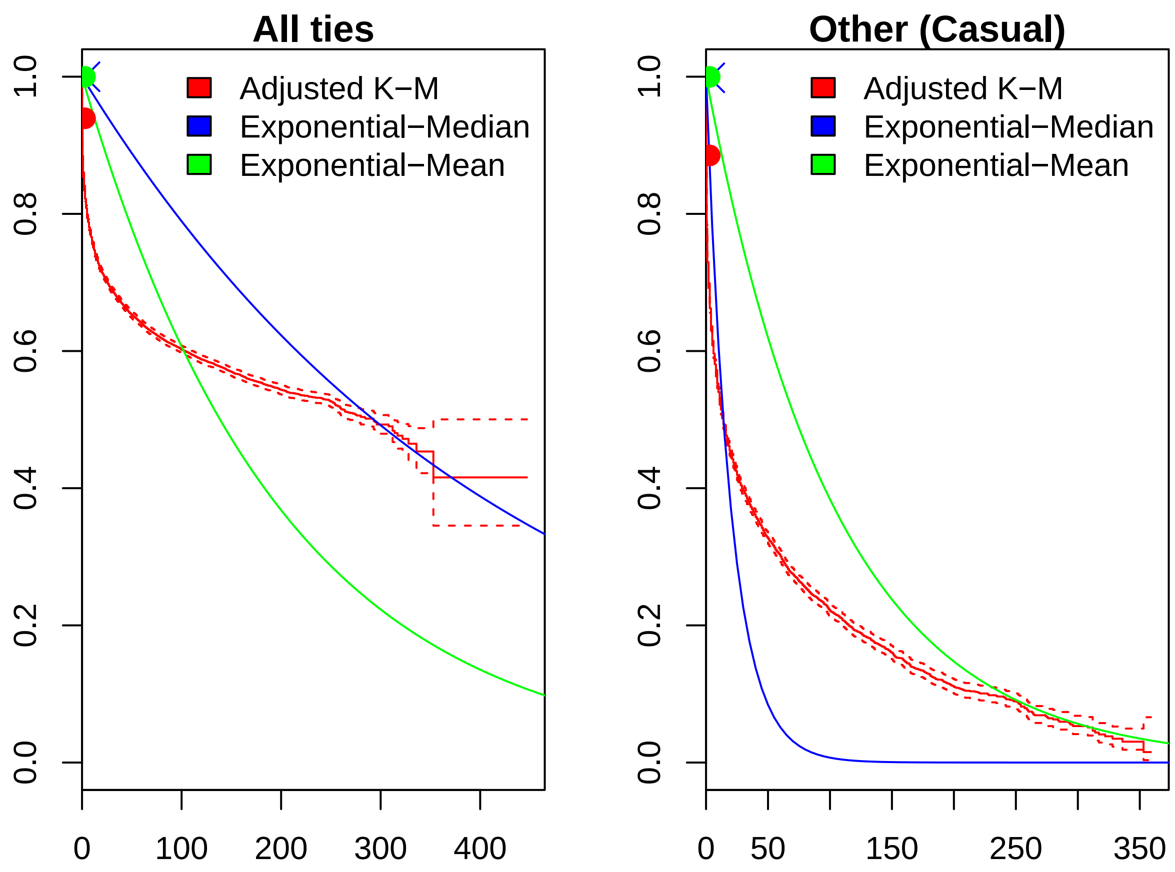
Two possible exponentials: one based on the adjusted median, and one based on the adjusted mean (which we compute assuming that all relationships are over by 1 month after the longest observed relationship, i.e. K-M curve goes to zero immediately).

We compare the full distribution as well as single out the proportion of one-times (see the large dots).

```
#--------------------------------------------------------------------------------
# Adjusted data means
#--------------------------------------------------------------------------------
time.steps <- diff(c(0, summary(adj)$time, max(summary(adj)$time + 1)))
survivals.all <- c(summary(adj)$surv, 0)
survivals.other <- c(summary(adj.other)$surv, 0)
# Mean
(adj.mean.all <- sum(time.steps * survivals.all))

## [1] 200.0655

(adj.mean.other <- sum(time.steps * survivals.other))

## Warning in time.steps * survivals.other: longer object length is not a multiple
## of shorter object length
```

```
## [1] 104.4122

#------------------------------------------------------------------------------
# Proportion one-times from the mean-based exponential
#------------------------------------------------------------------------------
(prop.ot.exp.all.mean <- pexp(1/30.5, rate = 1/adj.mean.all))

## [1] 0.0001638674

(prop.ot.exp.other.mean <- pexp(1/30.5, rate = 1/adj.mean.other))

## [1] 0.0003139647

#------------------------------------------------------------------------------
# Proportion one-times from the median-based exponential
#------------------------------------------------------------------------------
(prop.ot.exp.all.med <- pexp(1/30.5, rate = log(2)/kmmed.all["Adjusted Median"][1,
    1]))

## [1] 7.75606e-05

(prop.ot.exp.other.med <- pexp(1/30.5, rate = log(2)/kmmed.other["Adjusted Median"][1,
    1]))

## [1] 0.001621979

#------------------------------------------------------------------------------
# Plots of adjusted K-M versus exponential
#------------------------------------------------------------------------------
par(mfrow = c(1, 2), mar = c(2, 3, 1, 1) + 0.1)
#------------------------------------------------------------------------------
# All ties
#------------------------------------------------------------------------------
plot(adj, main = "All ties", col = "red")
curve(1 - pexp(x, rate = log(2)/kmmed.all["Adjusted Median"][1, 1]), add = TRUE,
    col = "blue", from = 0, to = 500)
curve(1 - pexp(x, rate = 1/adj.mean.all), add = TRUE, col = "green", from = 0, to = 500)
legend("topright", legend = c("Adjusted K-M", "Exponential-Median", "Exponential-Mean"),
    fill = c("red", "blue", "green"), bty = "n")
points(3, 1 - prop.ot.km.all[2], col = "red", cex = 1.5, pch = 16)
points(3, 1 - prop.ot.exp.all.med, col = "blue", cex = 2, pch = 4)
points(3, 1 - prop.ot.exp.all.mean, col = "green", cex = 1.5, pch = 16)
#------------------------------------------------------------------------------
# Other (Casual)
#------------------------------------------------------------------------------
plot(adj.other, main = "Other (Casual)", col = "red")
curve(1 - pexp(x, rate = log(2)/kmmed.other["Adjusted Median"][1, 1]), add = TRUE,
    col = "blue", from = 0, to = 500)
curve(1 - pexp(x, rate = 1/adj.mean.other), add = TRUE, col = "green", from = 0,
    to = 500)
legend("topright", legend = c("Adjusted K-M", "Exponential-Median", "Exponential-Mean"),
    fill = c("red", "blue", "green"), bty = "n")
```

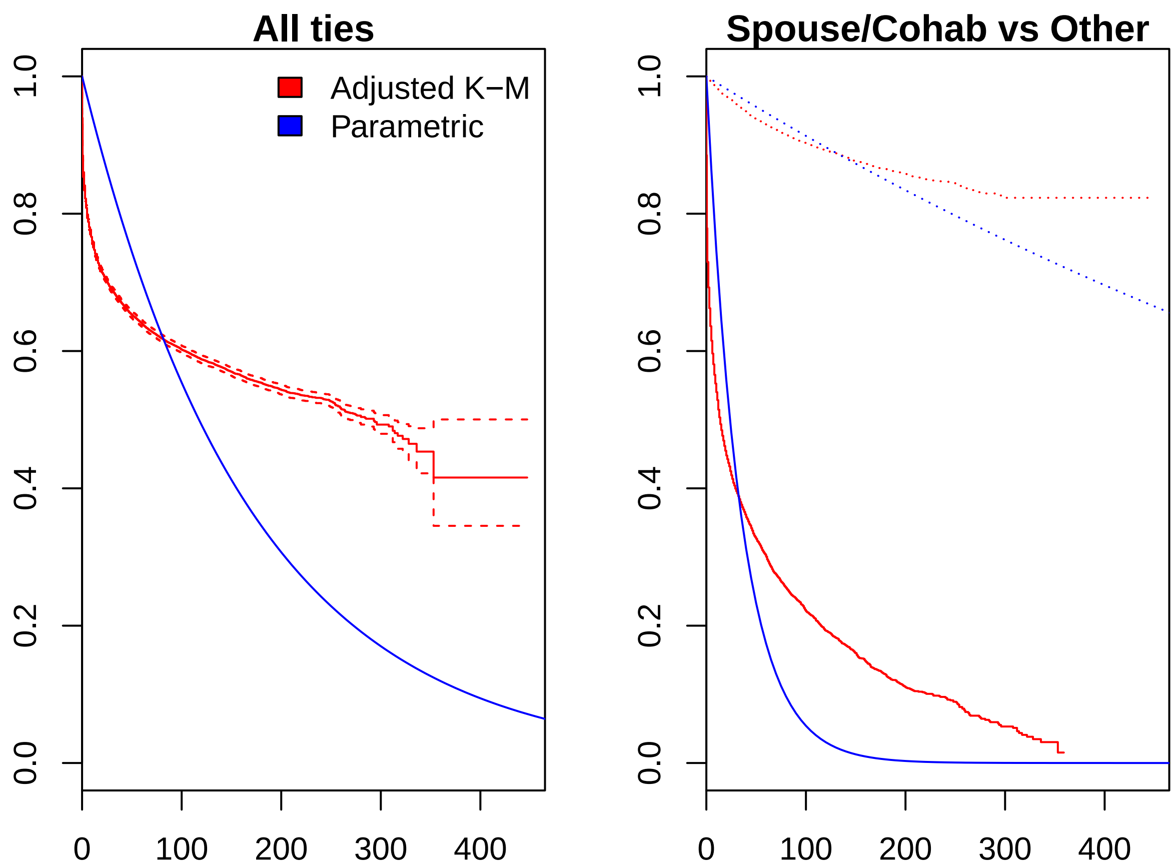
```r
points(3, 1 - prop.ot.km.other[2], col = "red", cex = 1.5, pch = 16)
points(3, 1 - prop.ot.exp.other.med, col = "blue", cex = 2, pch = 4)
points(3, 1 - prop.ot.exp.other.mean, col = "green", cex = 1.5, pch = 16)
```



6.6.2 Exponentials based on parametric regression

```r
#--------------------------------------------------------------------------------
# Plots of adjusted K-M versus parametric exponential
#--------------------------------------------------------------------------------
par(mfrow = c(1, 2), mar = c(2, 3, 1, 1) + 0.1)
#--------------------------------------------------------------------------------
# All ties
#--------------------------------------------------------------------------------
plot(adj, main = "All ties", col = "red")
adjexpreg <- phreg(adS ~ 1, dist = "weibull", shape = 1, data = dfl)
curve(1 - pexp(x, rate = 1/exp(adjexpreg$coefficients)), add = TRUE, col = "blue",
    from = 0, to = 500)
legend("topright", legend = c("Adjusted K-M", "Parametric"), fill = c("red", "blue"),
    bty = "n")
#--------------------------------------------------------------------------------
# By rel
#--------------------------------------------------------------------------------
dflsc <- subset(dfl, main == "Spouse/Cohab")
dflo <- subset(dfl, main == "Other")
adjexpreg.sc <- phreg(adSsc ~ 1, dist = "weibull", shape = 1, data = dflsc)
```

```r
adjexpreg.o <- phreg(adSo ~ 1, dist = "weibull", shape = 1, data = subset(dfl, main ==
    "Other"))
plot(adj.sco, main = "Spouse/Cohab vs Other", col = "red", lty = c(1, 3))
curve(1 - pexp(x, rate = 1/exp(adjexpreg.sc$coefficients)), lty = 3, add = TRUE,
    col = "blue", from = 0, to = 500)
curve(1 - pexp(x, rate = 1/exp(adjexpreg.o$coefficients)), lty = 1, add = TRUE, col = "blue",
    from = 0, to = 500)
```

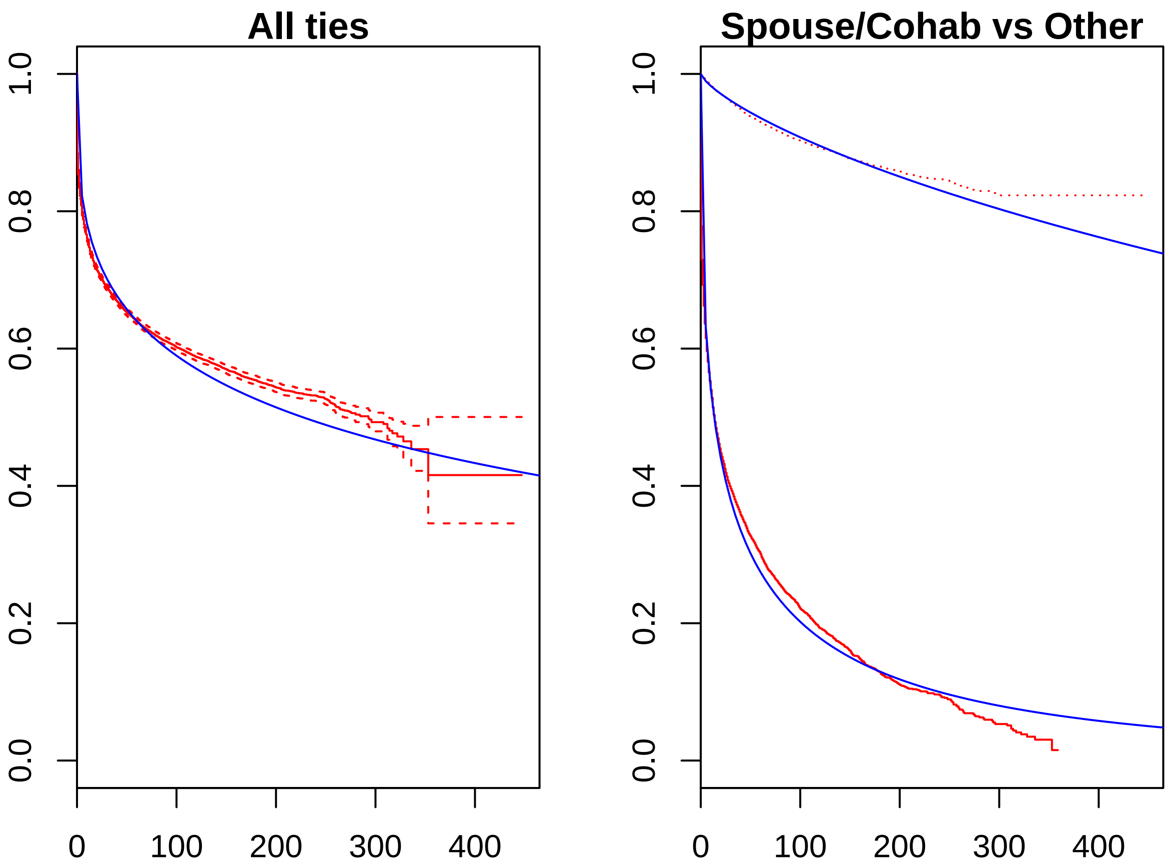



```r
#------------------------------------------------------------------------------
# Proportion one-times from the mean-based exponential
#------------------------------------------------------------------------------
(prop.ot.expreg <- pexp(1/30.5, rate = 1/exp(adjexpreg.o$coefficients)))

## [1] 0.0009554517
```

6.6.3 Weibulls based on parametric regression

```r
#------------------------------------------------------------------------------
# Plots of adjusted K-M versus parametric exponential
#------------------------------------------------------------------------------
par(mfrow = c(1, 2), mar = c(2, 3, 1, 1) + 0.1)
#------------------------------------------------------------------------------
# All ties
#------------------------------------------------------------------------------
plot(adj, main = "All ties", col = "red")
```

```r
adjweibreg <- phreg(adS ~ 1, dist = "weibull", data = dfl)
acoefs <- exp(adjweibreg$coefficients)
curve(exp(-((x/acoefs[1])^(acoefs[2]))), from = 0, to = 500, col = "blue", add = TRUE)
#------------------------------------------------------------------------------
# By rel
#------------------------------------------------------------------------------
dflsc <- subset(dfl, main == "Spouse/Cohab")
dflo <- subset(dfl, main == "Other")
adjweibreg.sc <- phreg(adSsc ~ 1, dist = "weibull", data = dflsc)
adjweibreg.o <- phreg(adSo ~ 1, dist = "weibull", data = dflo)
wcoefs <- exp(c(adjweibreg.sc$coefficients, adjweibreg.o$coefficients))
names(wcoefs) <- c("sc.sc", "sc.sh", "o.sc", "o.sh")
plot(adj.sco, main = "Spouse/Cohab vs Other", col = "red", lty = c(1, 3))
curve(exp(-((x/wcoefs["sc.sc"])^(wcoefs["sc.sh"]))), from = 0, to = 500, col = "blue",
    add = TRUE)
curve(exp(-((x/wcoefs["o.sc"])^(wcoefs["o.sh"]))), from = 0, to = 500, col = "blue",
    add = TRUE)
```



7 Simulation of true and sampled relationship lengths

Another way to assess the appropriateness of exponentially-distributed relationships is via simulation.

As of Dec 20, the function *sim_and_sample_durs* is based on Steve's code that he emailed. The only difference is that the seed is set to 98103 for replicability when using the default observation time. The default observation time is a randomly selected day that is not very close to the beginning or end of the window.

7.1 Simulation parameters

Property	Stats
Window Size	1000
Expected Duration	40
Number of relationships	5000

7.2 ASIDE: Simulation is impacted by relative sizes of windodw length and mean duration

When relationships are long relative to window length, sampling extant ties will underestimate the true mean. In the plots below, the panel title shows the true mean, and the blue line is a loess smoother across simulations that have unique seeds for the random selection of the observation day.

```
#------------------------------------------------------------------------------
# Simulate duration=10
#------------------------------------------------------------------------------
dur10 <- sapply(1:1000, FUN = function(x) {
    mean(sim_and_sample_durs(expected_dur = 10, obs_time_seed = x, verbose = FALSE)$extant_ages)
})
```

```
## Using geometric distributionUsing geometric distributionUsing geometric distributionUsing geometric d
```

```
#------------------------------------------------------------------------------
# Simulate duration=40
#------------------------------------------------------------------------------
dur40 <- sapply(1:1000, FUN = function(x) {
    mean(sim_and_sample_durs(expected_dur = 40, obs_time_seed = x, verbose = FALSE)$extant_ages)
})
```

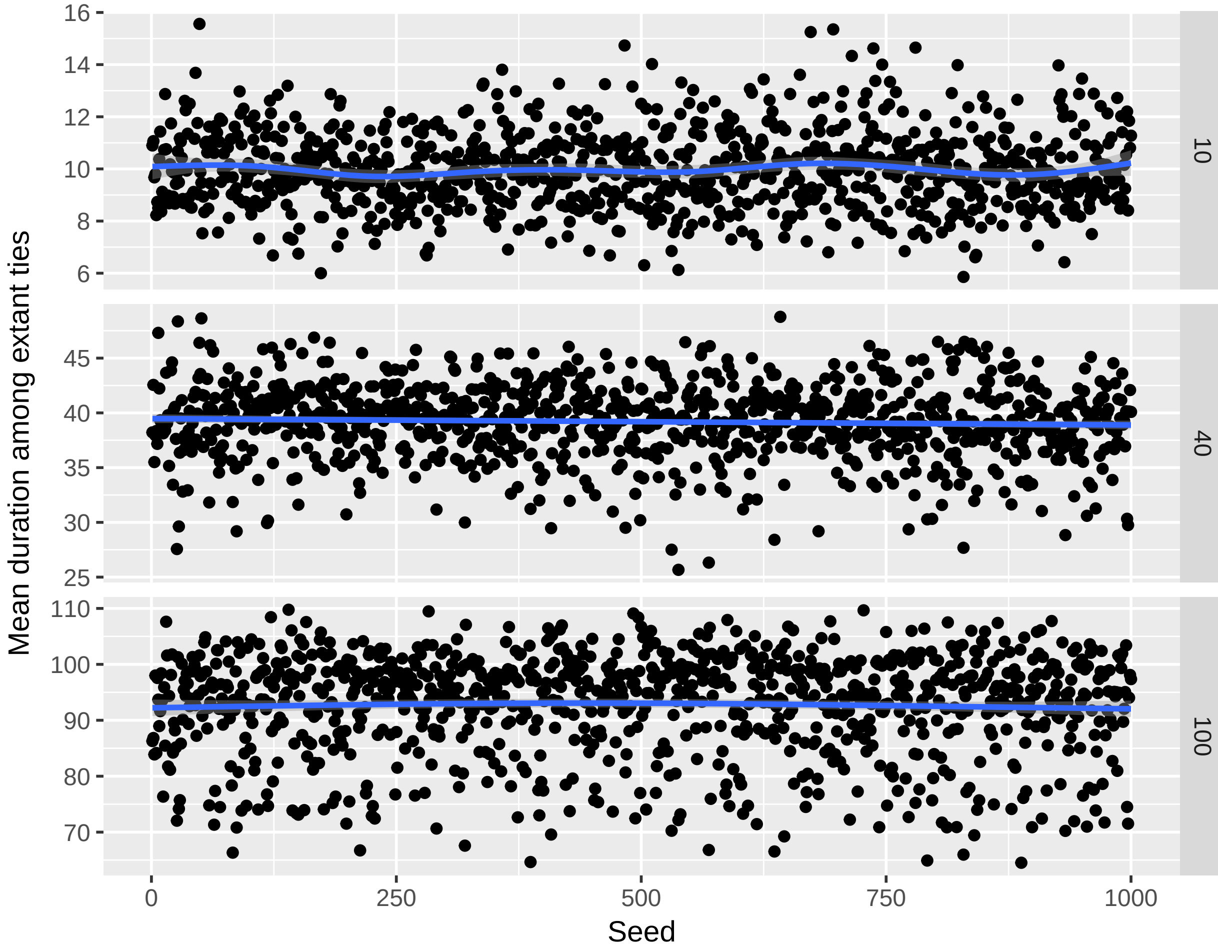
```
## Using geometric distributionUsing geometric distributionUsing geometric distributionUsing geometric d
```

```
#------------------------------------------------------------------------------
# Simulate duration=100
#------------------------------------------------------------------------------
dur100 <- sapply(1:1000, FUN = function(x) {
    mean(sim_and_sample_durs(expected_dur = 100, obs_time_seed = x, verbose = FALSE)$extant_ages)
})
```

```
## Using geometric distributionUsing geometric distributionUsing geometric distributionUsing geometric d
```

```
#------------------------------------------------------------------------------
# Plot
#------------------------------------------------------------------------------
simtoplot <- data.frame(Seed = rep(1:1000, 3), Dur = c(rep(10, 1000), rep(40, 1000),
    rep(100, 1000)), Mean = c(dur10, dur40, dur100))
ggplot(simtoplot, aes(x = Seed, y = Mean, group = Dur)) + geom_point() + scale_y_continuous("Mean durat
    geom_smooth() + facet_grid(Dur ~ ., scales = "free_y")
```

```
## `geom_smooth()` using method = 'gam'
```



7.3 Comparing active ties in simulated data to active ties in NSFG

7.3.1 Compute means and medians of active ties

```
#-----------------------------------------------------------------------------
# Spouses and cohabs: number of relationships, mean/med duration, and summary
# stats
#-----------------------------------------------------------------------------
n_sc <- nrow(subset(act1, main == "Spouse/Cohab"))
mean_sc <- mean(subset(act1, main == "Spouse/Cohab")$len)
med_sc <- median(subset(act1, main == "Spouse/Cohab")$len)
sc_summary <- summary(subset(act1, main == "Spouse/Cohab")$len)
#-----------------------------------------------------------------------------
# Other (Casual): number of relationships, mean/med duration, and summary
#-----------------------------------------------------------------------------
n_other <- nrow(subset(act1, main == "Other"))
mean_other <- mean(subset(act1, main == "Other")$len)
med_other <- median(subset(act1, main == "Other")$len)
other_summary <- summary(subset(act1, main == "Other")$len)
```

Summary:

Statistic	Spouses and Cohabs	Other
Number of relationships	17803	10683
Median Duration (Months)	95	14
Exponential Mean Based on Median	137	20
Mean Duration (Months)	108	31

7.3.2 Simulate, converting to a day time scale

Use a window of 30,000 days or 82 years, a plausible estimate for maximum relationship length.

Use number of relationships and mean relationship duration from the data. Note that Pavel proved that the mean of active ties is unbiased for geometrically-distributed data.

```
#-----------------------------------------------------------------------------
# Spouses/Cohabs: simulate
#-----------------------------------------------------------------------------
sc_sim_mean <- sim_and_sample_durs(expected_dur = round(mean_sc * 30), nrels = n_sc,
    verbose = FALSE, window_size = 30000)

## Using geometric distribution

sc_sim_med <- sim_and_sample_durs(expected_dur = round(meanfrommed_sc * 30), nrels = n_sc,
    verbose = FALSE, window_size = 30000)

## Using geometric distribution

#-----------------------------------------------------------------------------
# Other (Casual): simulate
#-----------------------------------------------------------------------------
other_sim_mean <- sim_and_sample_durs(expected_dur = round(mean_other * 30), nrels = n_other,
    verbose = FALSE, window_size = 30000)

## Using geometric distribution

other_sim_med <- sim_and_sample_durs(expected_dur = round(meanfrommed_other * 30),
    nrels = n_other, verbose = FALSE, window_size = 30000)

## Using geometric distribution
```

7.3.3 Compile simulated data

```
#-----------------------------------------------------------------------------
# Number of sampled ties
#-----------------------------------------------------------------------------
nsc1 <- length(sc_sim_mean$extant_ages)
nsc2 <- length(sc_sim_med$extant_ages)
noth1 <- length(other_sim_mean$extant_ages)
noth2 <- length(other_sim_med$extant_ages)
#-----------------------------------------------------------------------------
# Compile simulated data into one data frame
#-----------------------------------------------------------------------------
```

```r
simdf <- data.frame(data = rep("Sim", nsc1 + nsc2 + noth1 + noth2), main = c(rep("Spouse/Cohab",
    nsc1 + nsc2), rep("Other", noth1 + noth2)), stat_matched = c(rep("Mean", nsc1),
    rep("Median", nsc2), rep("Mean", noth1), rep("Median", noth2)), lendays = c(sc_sim_mean$extant_ages
    sc_sim_med$extant_ages, other_sim_mean$extant_ages, other_sim_med$extant_ages))
#--------------------------------------------------------------------------------
# Convert back to months and code <1 month relationships as one-times
#--------------------------------------------------------------------------------
simdf <- within(simdf, {
    len <- lendays/30
    len[lendays <= 1] <- 0.25
    lencat1 <- cut(len, breaks = c(0, 0.5, 1, 6, 12, 60, 120, 999), right = FALSE)
})
#--------------------------------------------------------------------------------
# Combine into a data frame with NSFG data
#--------------------------------------------------------------------------------
nsfg4sim <- transform(subset(pop, select = c("main", "len", "lencat1")), lendays = len *
    30, stat_matched = "Mean", data = "NSFG 2006-2010")
nsfg4sim2 <- transform(subset(pop, select = c("main", "len", "lencat1")), lendays = len *
    30, stat_matched = "Median", data = "NSFG 2006-2010")
simVnsfg <- rbind(simdf, nsfg4sim, nsfg4sim2)
```

7.3.4 Compare summary stats for all ties vs sampled active ties

```r
#--------------------------------------------------------------------------------
# Spouses/Cohabs, parameterized by data mean
#--------------------------------------------------------------------------------
sapply(sc_sim_mean, function(x) round(summary(x)/30, 1))

##         extant_ages all_ages
## Min.            0.2      0.0
## 1st Qu.        30.1     30.7
## Median         75.9     74.5
## Mean          106.3    108.2
## 3rd Qu.       145.6    150.1
## Max.          678.3   1285.3

#--------------------------------------------------------------------------------
# Spouses/Cohabs, parameterized by data median
#--------------------------------------------------------------------------------
sapply(sc_sim_med, function(x) round(summary(x)/30, 1))

##         extant_ages all_ages
## Min.            0.0      0.0
## 1st Qu.        39.9     39.0
## Median         93.7     95.7
## Mean          132.3    138.0
## 3rd Qu.       188.9    190.9
## Max.          639.3   1393.3

#--------------------------------------------------------------------------------
# Other, parameterized by data mean
#--------------------------------------------------------------------------------
sapply(other_sim_mean, function(x) round(summary(x)/30, 1))
```

```
##         extant_ages all_ages
## Min.           0.1      0.0
## 1st Qu.        8.9      9.1
## Median        22.1     22.0
## Mean          31.5     31.6
## 3rd Qu.       46.2     43.7
## Max.         183.7    348.0
```

```r
#-------------------------------------------------------------------------------
# Other, parameterized by data median
#-------------------------------------------------------------------------------
sapply(other_sim_med, function(x) round(summary(x)/30, 1))
```

```
##         extant_ages all_ages
## Min.           0.1      0.0
## 1st Qu.        7.4      5.8
## Median        14.9     14.3
## Mean          22.2     20.4
## 3rd Qu.       31.6     28.3
## Max.         185.3    223.8
```

The sampling really impacts the maximum relationship length observed, but the other quartiles appear very well represented by the sample.

7.3.5 Compare summary stats for sampled sim data vs NSFG

```r
#-------------------------------------------------------------------------------
# Spouses/Cohabs
#-------------------------------------------------------------------------------
data.frame(sim_mean = sapply(sc_sim_mean, function(x) round(summary(x)/30, 1))[,
    "extant_ages"], sim_med = sapply(sc_sim_med, function(x) round(summary(x)/30,
    1))[, "extant_ages"], nsfg = c(sc_summary))
```

```
##         sim_mean sim_med  nsfg
## Min.         0.2     0.0   0.5
## 1st Qu.     30.1    39.9  46.0
## Median      75.9    93.7  95.0
## Mean       106.3   132.3 107.6
## 3rd Qu.    145.6   188.9 158.0
## Max.       678.3   639.3 447.0
```

```r
#-------------------------------------------------------------------------------
# Other
#-------------------------------------------------------------------------------
data.frame(sim_mean = sapply(other_sim_mean, function(x) round(summary(x)/30, 1))[,
    "extant_ages"], sim_med = sapply(other_sim_med, function(x) round(summary(x)/30,
    1))[, "extant_ages"], nsfg = c(other_summary))
```

```
##         sim_mean sim_med  nsfg
## Min.         0.1     0.1   0.5
## 1st Qu.      8.9     7.4   4.0
## Median      22.1    14.9  14.0
```

```
## Mean         31.5    22.2  31.3
## 3rd Qu.      46.2    31.6  39.0
## Max.        183.7   185.3 359.0
```

7.3.6 Plot simulated data next to observed data

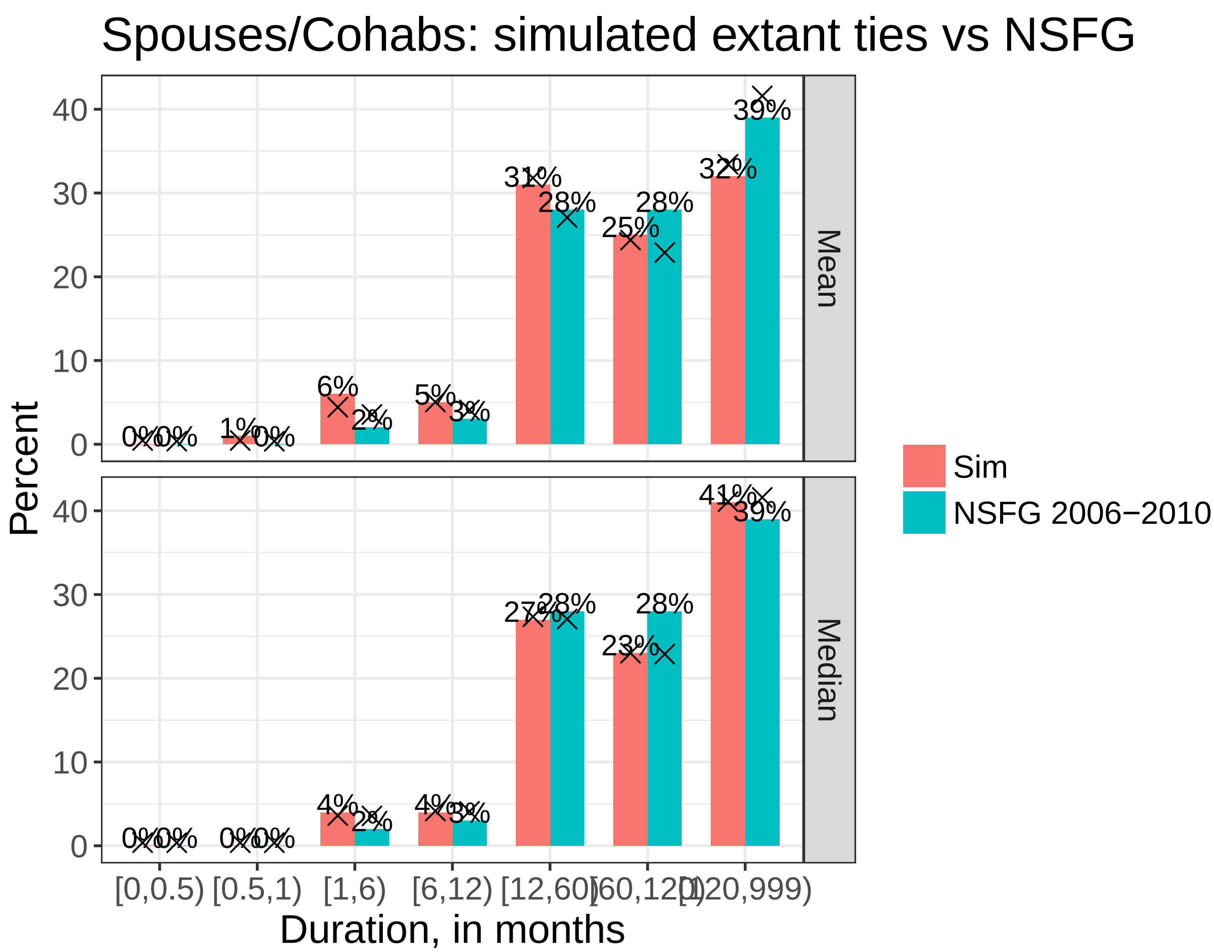
Note: I need to remove the Xs in the plots - they are not meaningful here. Please ignore them.

The bottom line here is that by coarse duration categories, the simulated data (Orange) match the NSFG data (green) decently well for Spouses/Cohabs and not as well for Other/Casual. However, the simulated data get the left half of the Other/Casual distribution quite well. For the right half, the simulated relationships are shorter than the ones in the data.

```
plot_categorical(subset(simVnsfg, main == "Spouse/Cohab"), var = "lencat1", group = "data",
    panel = "stat_matched", yperc = TRUE, ylab = "Percent") + facet_grid(panel ~
    .) + scale_x_discrete(name = "Duration, in months") + ggtitle("Spouses/Cohabs: simulated extant ties
```

```
##
## Sum of bin proportions is  1
## Sum of bin proportions is  1
## Sum of bin proportions is  1
## Sum of bin proportions is  1
```

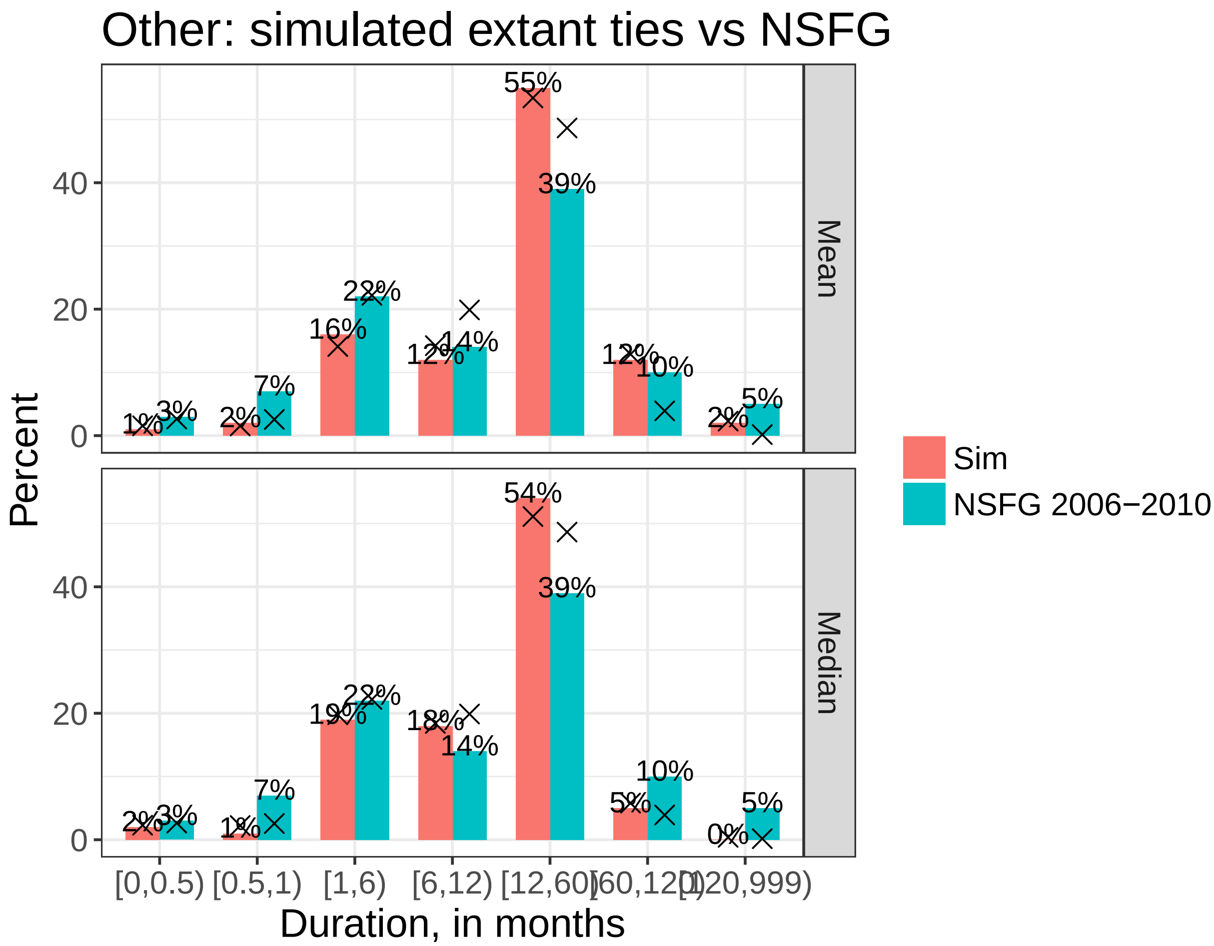
```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```




```
plot_categorical(subset(simVnsfg, main == "Other"), var = "lencat1", group = "data",
    panel = "stat_matched", yperc = TRUE, ylab = "Percent") + facet_grid(panel ~
    .) + ggtitle("Other: simulated extant ties vs NSFG") + scale_x_discrete(name = "Duration, in months
```

```
##
## Sum of bin proportions is  1
## Sum of bin proportions is  1
## Sum of bin proportions is  1
## Sum of bin proportions is  1
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```


Other: simulated extant ties vs NSFG
Mean
Median
Percent
Duration, in months
[0,0.5)
[0.5,1)
[1,6)
[6,12)
[12,60)
[60,120)
[120,999)
Sim
NSFG 2006−2010