Spend Down Analysis

Joe Mienko

Sources of Data

The data in this analysis draw from the FamLink data warehouse (FLDW) managed by the Department of Children Youth and Families (DCYF). These data are made available to the author under a data sharing agreement between DCYF and the University of Washington School of Social Work. As of the date of this writing, this analysis is still being reviewed by DCYF.

Population of Interest

We begin our analysis by establishing a population of placement episodes which is ostensibly court involved. This is accomplished by examining the most recent legal status of a given case (`tx_lgl_stat`), the discharge type (`tx_dsch_rsn`), and a series of checks back to the `LEGAL` datamart of the FLDW (as represented by the `fl_dep_exist` flag). Wherever possible, all records in this analysis make use of the "base reporting tables" developed by the business intelligence team at DCYF. As can be seen, we are limiting this analysis to the counties of Grant and Lewis - the two "treatment" counties for the legal representation pilot. We also restrict the sample to children who were under the age of 18 at the point of removal.

```sql
if object_id('tempdb..##court_episodes') is not null
    drop table ##court_episodes;

select
  *
  ,case
    when lf_dt is not null
      then lf_dt
    else discharge_dt
  end discharge_from_pilot
into ##court_episodes
from base.rptPlacement rp
where rp.tx_county in ('Grant', 'Lewis')
  and dbo.fnc_datediff_yrs(birthdate, removal_dt) < 18
  and (tx_lgl_stat in
      ('Closed - Adoption'
      ,'Closed - Dependency Guardianship No Supv'
      ,'Closed - Superior Court Guardianship'
      ,'Closed - Title 13 Guardianship'
      ,'Closed -Non-parental (3rd party) custody'
      ,'Dependency Guardianship'
      ,'Dependent'
      ,'Dependent - Legally Free'
      ,'EFC Dependent'
      ,'Non-parental (3rd Party) custody'
      ,'Parental Custody -Continued Court Action'
      ,'Protective Custody'
      ,'Shelter Care'
      ,'Superior Court Guardianship')
    or tx_dsch_rsn = 'Returned to Custody of Parents - Dependency Dismissed'
    or fl_dep_exist = 1)
```

Expected Time to Permanency

A critical component of this analysis concerns our expectations for the timing of permanency outcomes in the Counties of Grant and Lewis. There are a number of methods for making such estimates. However, in order to make use of the most recent data available for this analysis, we will employ a Kaplan-Meier estimator as implemented within the `survival` package by Terry M. Therneau and Patricia M. Grambsch (2000). This is accomplished by establishing subsetting the temporary table defined above into a standard event-history table.

```
select
  rp.child id_prsn_child
  ,rp.removal_dt
  ,iif(rp.discharge_from_pilot = '9999-12-31', '2018-04-27'
    ,rp.discharge_from_pilot) discharge_dt
  ,iif(rp.discharge_from_pilot = '9999-12-31', null
    ,rp.discharge_from_pilot) discharge_dt_null
  ,iif(rp.discharge_from_pilot = '9999-12-31', 0, 1) fl_discharge
  ,datediff(dd, rp.removal_dt,
  iif(rp.discharge_dt = '9999-12-31', '2018-04-27', rp.discharge_from_pilot)) + 1 los
  ,rp.id_case
  ,rp.tx_county
from ##court_episodes rp
where rp.removal_dt between '2010-01-01' and '2018-04-27'
```

The survival curves generated from the `survival` package are plotted below.

```
historical_surv_grant <- read_feather("historical_surv_grant.feather")

median_loc_grant <- which.min(abs(historical_surv_grant$surv - .5))
median_value_grant <- historical_surv_grant$surv[median_loc_grant]
median_days_grant <- historical_surv_grant$time[median_loc_grant]
median_label_grant <- paste0("Median Time to Exit = "
                             ,median_days_grant
                             ," Days")


ggplot(historical_surv_grant
       ,aes(y = surv
            ,x = time)) +
  geom_line(colour = "blue", linetype = "dashed") +
  ggplot2::geom_vline(xintercept = median_days_grant) +
  ggplot2::geom_hline(yintercept = round(median_value_grant, 2)) +
  geom_text(aes(median_days_grant*2
                ,median_value_grant
                ,label = median_label_grant
                ,vjust = -1)) +
  ylab("Percent Remaining in Dependency") +
  xlab("Days Since Entry") +
  ggtitle("Timing of Dependency Exits since 2010, Grant County") +
  theme_bw()
```

Timing of Dependency Exits since 2010, Grant County
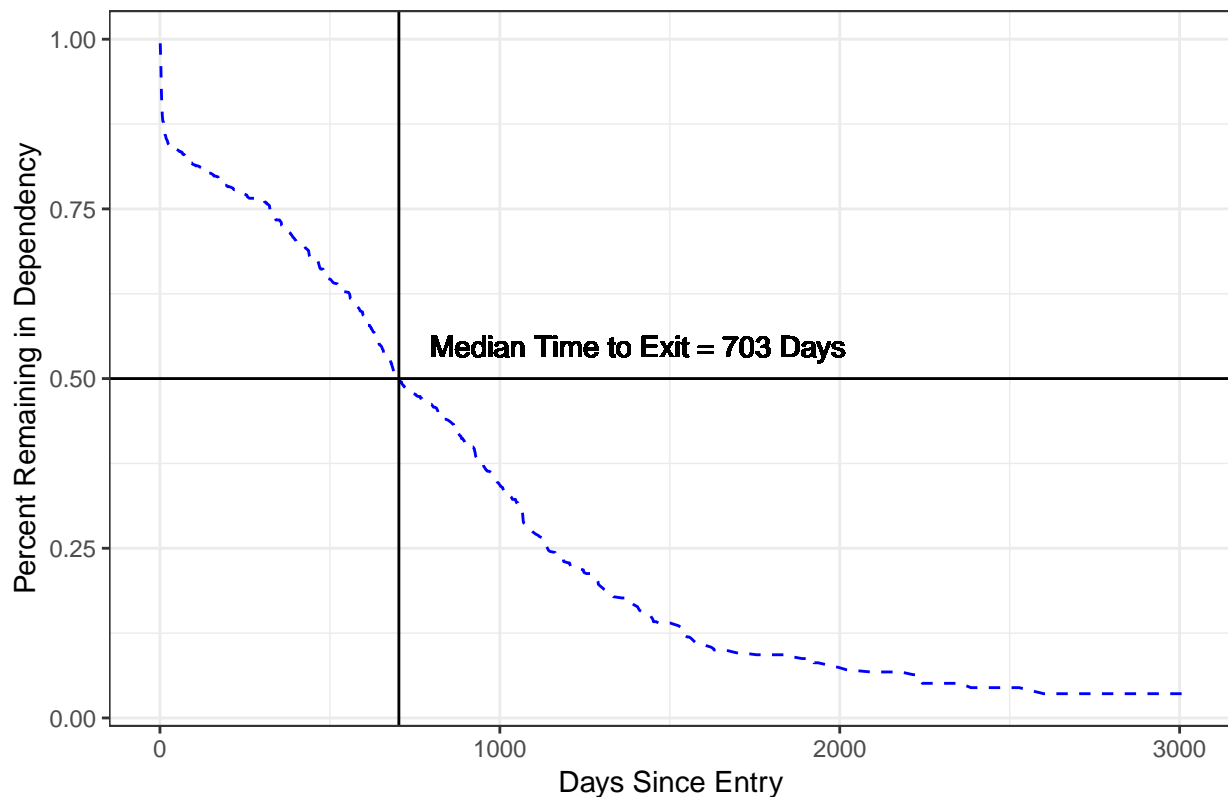

```r
historical_surv_lewis <- read_feather("historical_surv_lewis.feather")

median_loc_lewis <- which.min(abs(historical_surv_lewis$surv - .5))
median_value_lewis <- historical_surv_lewis$surv[median_loc_lewis]
median_days_lewis <- historical_surv_lewis$time[median_loc_lewis]
median_label_lewis <- paste0("Median Time to Exit = "
                             ,median_days_lewis
                             ," Days")

ggplot(historical_surv_lewis
       ,aes(y = surv
            ,x = time)) +
  geom_line(colour = "blue", linetype = "dashed") +
  ggplot2::geom_vline(xintercept = median_days_lewis) +
  ggplot2::geom_hline(yintercept = round(median_value_lewis, 2)) +
  geom_text(aes(median_days_lewis*2
                ,median_value_lewis
                ,label = median_label_lewis
                ,vjust = -1)) +
  ylab("Percent Remaining in Dependency") +
  xlab("Days Since Entry") +
  ggtitle("Timing of Dependency Exits since 2010, Lewis County") +
  theme_bw()
```

Timing of Dependency Exits since 2010, Lewis County

Median Time to Exit = 618 Days
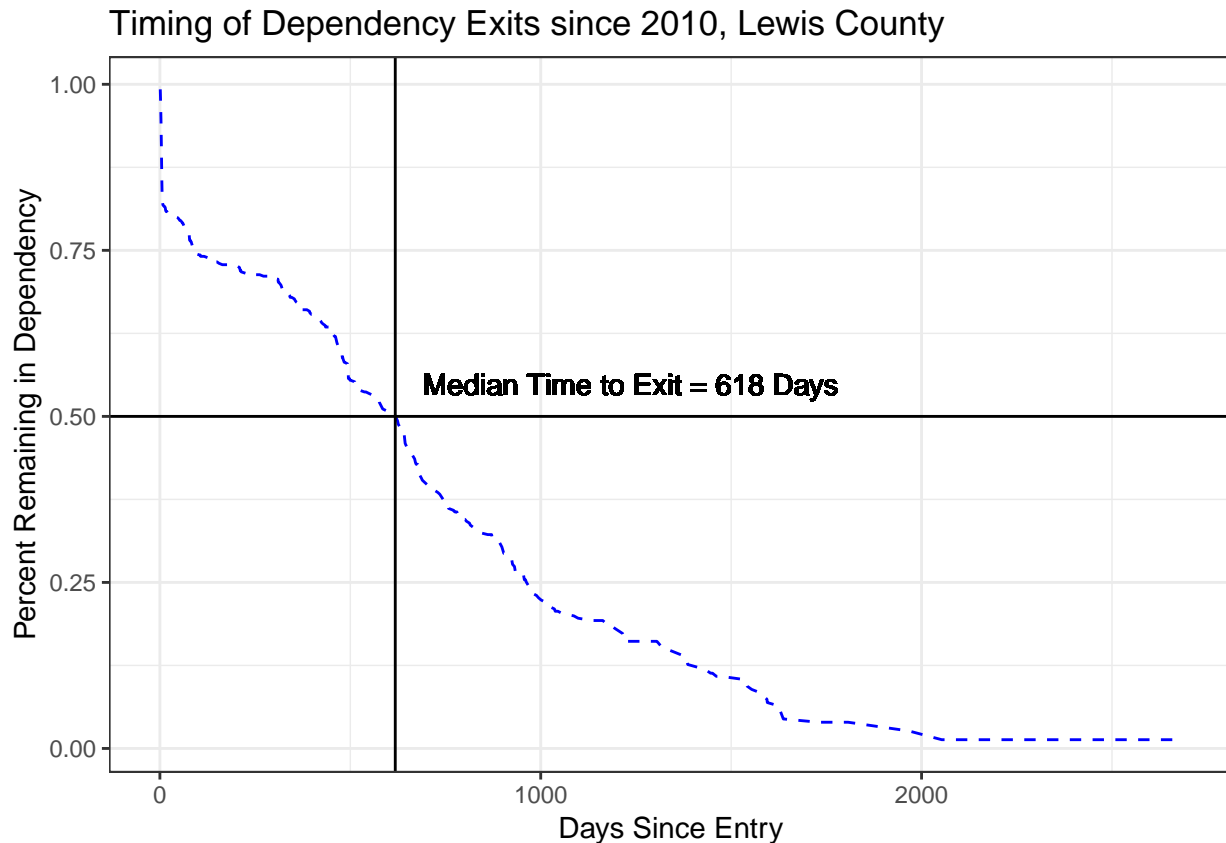
As can be seen, the median time to exit for both Grant and Lewis is nearly two years (703 days and 618 days respectively). In other words, this is the number of days it takes for half of a "typical" cohort of dependent children to exit from the system. This is, however, only one point on the estimated survival function for each county. In the following sections, we will make use of the entire estimated function for each jurisdiction.

Historical Daily Counts and Flows of Dependent Children

The previous section can provide us with good information regarding how long we might expect children in Grant and Lewis to remain in care once they have entered. In order to determine how long the pilot should plan to serve the children on its caseload, it will also be useful for us to understand the rate at which children enter and exit care in these counties.

The next chunk of code selects information (using the same temporary table as a starting point) concerning the monthly counts of entries and exits into dependency for Grant and Lewis.

```
with calendar_dates as
(
select
    cd.*
    ,iif(i = 1, 'Lewis', 'Grant') tx_county
from calendar_dim cd, dbo.NumbersTable(1, 2, 1) cty
where cd.calendar_date between '2000-01-01' and '2018-04-27'
), placements as
(
select
    rp.removal_dt
    ,id_removal_episode_fact
```

```sql
      ,iif(rp.discharge_from_pilot = '9999-12-31', '2018-04-27'
        ,rp.discharge_from_pilot) discharge_dt
      ,iif(rp.discharge_from_pilot = '9999-12-31', null
        ,rp.discharge_from_pilot) discharge_dt_null
    ,rp.tx_county
from ##court_episodes rp
  where rp.tx_county in ('Grant', 'Lewis')
), inflow_counts as
(
select distinct
  calendar_date
  ,cd.tx_county
  ,count(distinct id_removal_episode_fact) inflow_of_placements
from calendar_dates cd
  left join placements p
    on cd.calendar_date = removal_dt
      and cd.tx_county = p.tx_county
group by
  calendar_date
  ,cd.tx_county
), outflow_counts as
(
select distinct
  calendar_date
  ,cd.tx_county
  ,count(distinct id_removal_episode_fact) outflow_of_placements
from calendar_dates cd
  left join placements p
    on cd.calendar_date = discharge_dt_null
      and cd.tx_county = p.tx_county
group by
  calendar_date
  ,cd.tx_county
), daily_counts as
(
select distinct
  ic.calendar_date
  ,ic.tx_county
  ,isnull(ic.inflow_of_placements, 0) total_inflow_of_placements
  ,isnull(oc.outflow_of_placements, 0) total_outflow_of_placements
from inflow_counts ic
  left join outflow_counts oc
    on ic.calendar_date = oc.calendar_date
      and ic.tx_county = oc.tx_county
)

select
    sum(dc.total_inflow_of_placements) total_inflow_of_placements
    ,sum(dc.total_outflow_of_placements) total_outflow_of_placements
    ,cd.month
    ,cd.tx_county
from calendar_dates cd
    join daily_counts dc
```

```
        on cd.calendar_date = dc.calendar_date
            and cd.tx_county = dc.tx_county
where cd.month < '2018-04-01'
group by
    cd.month
    ,cd.tx_county
```

```r
dat_stock_and_flow <- read_feather("dat_stock_and_flow.feather")


flow_names <- list(
  "total_inflow_of_placements"="Monthly Entries",
  "total_outflow_of_placements"="Monthly Exits"
)

flow_labeller <- function(variable,value){
  return(flow_names[value])
}

dat_stock_and_flow %>%
  filter(tx_county == "Grant") %>%
  select(-tx_county) %>%
  gather(key = measure, value = count, -month) %>%
  ggplot(aes(y = count, x = month)) +
  geom_line() +
  geom_smooth() +
  facet_wrap(.~measure
              ,nrow = 2
              ,labeller = flow_labeller) +
  xlab("") +
  ylab("") +
  ggtitle("Monthly Entries and Exits Since 2000, Grant County") +
  theme_bw()
```
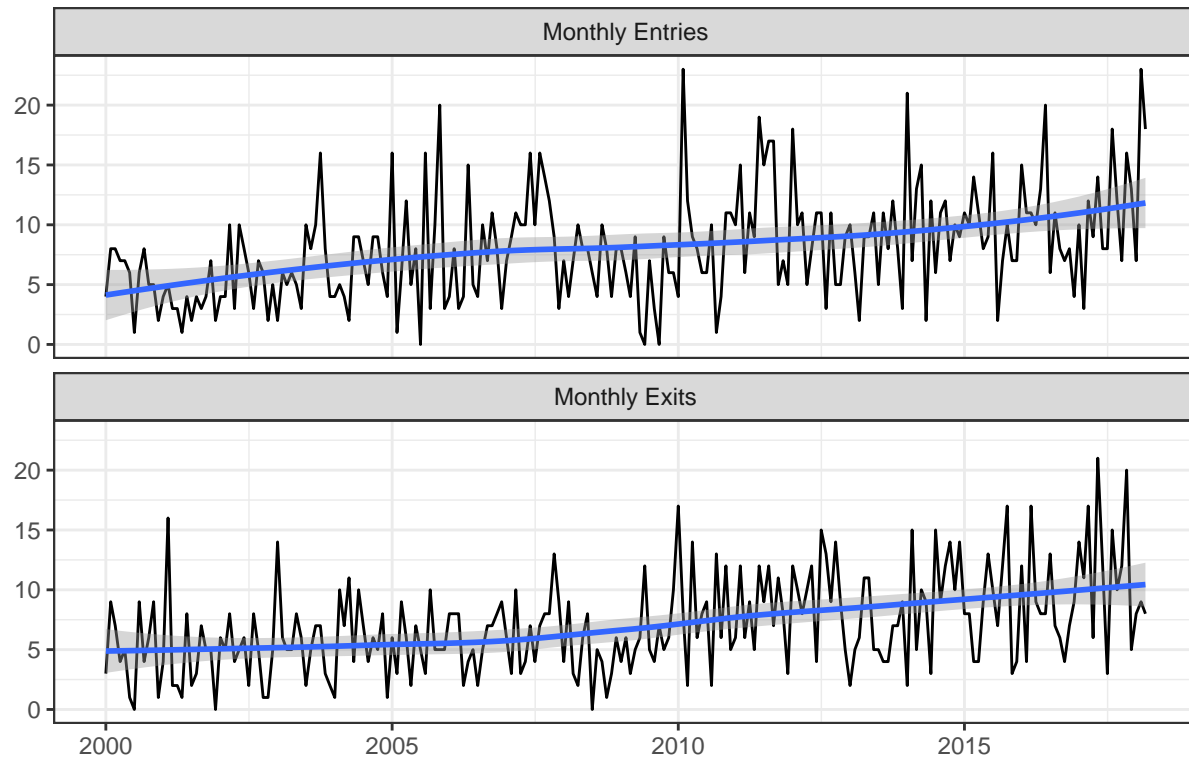
Monthly Entries and Exits Since 2000, Grant County


```
dat_stock_and_flow %>%
  filter(tx_county == "Lewis") %>%
  select(-tx_county) %>%
  gather(key = measure, value = count, -month) %>%
  ggplot(aes(y = count, x = month)) +
  geom_line() +
  geom_smooth() +
  facet_wrap(.~measure
             ,nrow = 2
             ,labeller = flow_labeller) +
  xlab("") +
  ylab("") +
  ggtitle("Monthly Entries and Exits Since 2000, Lewis County") +
  theme_bw()
```

Monthly Entries and Exits Since 2000, Lewis County
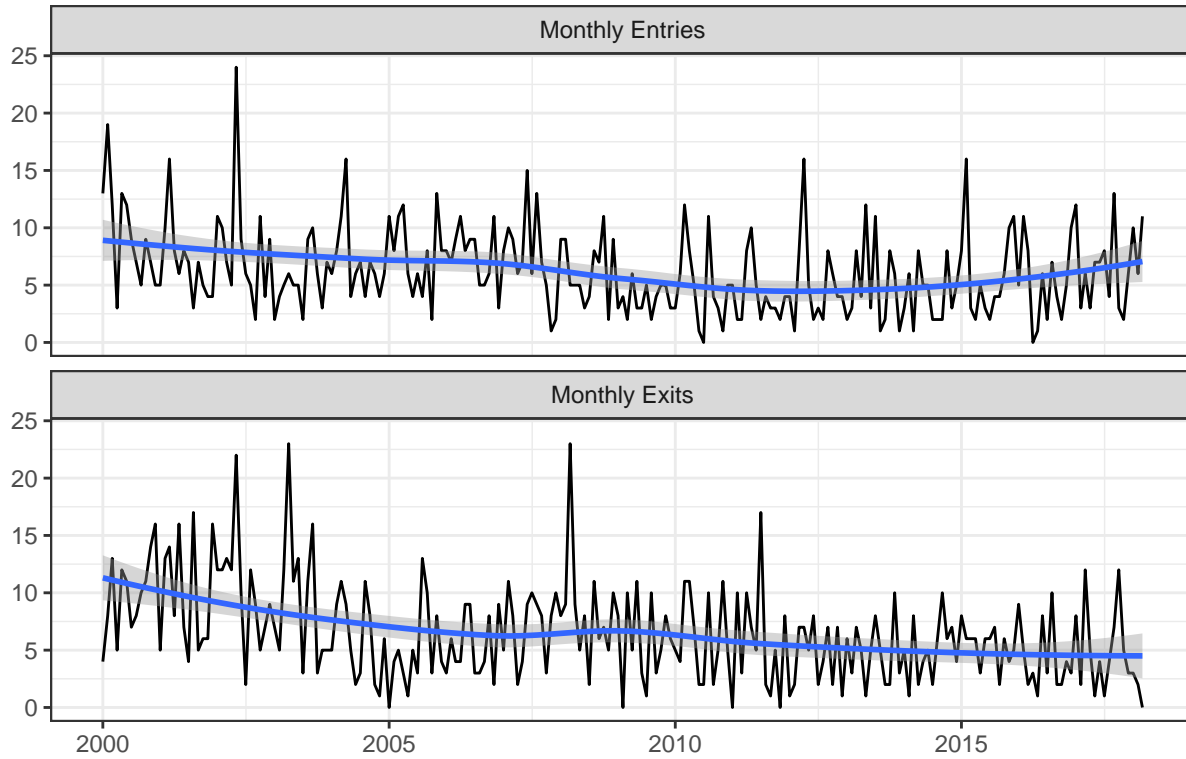
This historical information can be used in a variety of ways. For our current purposes, we want to make a reasonable estimate as to how many entries and exits can be expected for several months into the future. While a complete discussion of the forecasting models is beyond the scope of this analysis, we will proceed with a generalization of moving average models referred to as autoregressive integrated moving average (ARIMA) models, as implemented by Hyndman et al. (2018). Specifically, we will make use of a non-seasonal ARIMA(p,d,q) model in which p is defined as the number of time lags in the model (i.e. how far back the model looks for information), d is the number of times the have had past values subtracted (typically for stablization), and q is the order of the underlying moving-average model. In order to make use of this form of ARIMA, we will first "deseasonalize" our data. In other words, we will identify any patterns of seasonality in the data, and focus our analysis on the underlying trend of the data. We will then follow the "automatic ARIMA" algorithm developed by Hyndman, Khandakar, and others (2007) to obtain reasonable starting points for p, d, and q above. Using visual examination of the residuals generated from these parameters, we then adjust the q parameter. To be clear, we do not propose the selected ARIMA models to be recommended forecasting models beyond this analysis. Here, the chosen models are being selected as reasonable estimates - preferrable to simply averaging entries and exits over the observed time period.

References

Hyndman, Rob J, George Athanasopoulos, Christoph Bergmeir, Gabriel Caceres, Leanne Chhay, Mitchell O'Hara-Wild, Fotios Petropoulos, Slava Razbash, Earo Wang, and Farah Yasmeen. 2018. "Forecast: Forecasting Functions for Time Series and Linear Models."

Hyndman, Rob J, Yeasmin Khandakar, and others. 2007. *Automatic Time Series for Forecasting: The Forecast Package for R*. 6/07. Monash University, Department of Econometrics; Business Statistics.

Terry M. Therneau, and Patricia M. Grambsch. 2000. *Modeling Survival Data: Extending the Cox Model*. New York: Springer.