Survival Analysis of the Florida Cancer Data System: A Data Science Project Using Stata

Anders Alexandersson
Florida Cancer Data System
Miami, FL
aalexandersson@med.miami.edu

Abstract

This technical report provides a survival analysis of the Florida Cancer Data System (FCDS). The technical report consists of two chapters, an appendix, and a separate supplement. Chapter 1, Data Management, states the problem of survival analysis and it creates the analysis dataset. The Human Mortality Database is used to create the population mortality file. The concept of net (not crude) survival in a relative (not cause-specific) framework is central to survival analysis of FCDS. Chapter 2, Survival Analysis, illustrates the four conceptual approaches to survival analysis of FCDS. An example of net survival in the relative framework is estimated 10-year survival of adult lung cancer patients diagnosed in Florida 1999-2003. The appendix provides a sensitivity analysis. The separate supplement (Alexandersson, 2017a) discusses alternative software such as SAS and SEER*Stat, and it includes code for running R in Stata.

This technical report is also a data science project using mostly Stata. Here, data science is defined as programming the workflow of data analysis. Important Stata commands are `odbc load` for importing data from the FCDS database, `stnet` and `strs` for survival analysis, and `texdoc` and `tabout` for reporting. FCDS follows the standards of the North American Association of Central Cancer Registries (NAACCR). Currently, the SAS macro "CalculateSurvivalTimeInMonths.sas" (http://seer.cancer.gov/survivaltime/) is required for creating survival analysis variables according to the NAACCR standards. The main advantage of data science is reproducibility. Stata is used for estimating net survival because only Stata has implemented a life-table (actuarial) version of the Pohar Perme estimator of net survival. Pohar Perme estimation of net survival is useful because other approaches tend to overestimate survival. A life-table version is useful because FCDS releases birth year only, not full birth dates, which affects the matching of the survival times against the life tables. A companion monograph (Alexandersson, 2017b) provides net survival rates for all ten FCDS cancer site groups, and those rates are calculated as explained in this technical report.

Keywords: net survival, relative framework, Pohar Perme, life table, Stata, stnet, data science.

Contents

Chapter 1

Data Management

1.1 Introduction

This technical report provides a survival analysis of the Florida Cancer Data System (FCDS). Survival analysis is just another name for time-to-event analysis. The point of survival analysis is to follow subjects over time and observe at which point in time they experience the event of interest. Population-based cancer survival analysis deals almost exclusively with the time from diagnosis of cancer to death. The concept of **net** (not crude) **survival in a relative** (not cause-specific) **framework** is central to survival analysis of FCDS.

FCDS is Florida's statewide cancer registry. In 1978, the Florida Department of Health (DOH) contracted with the Sylvester Comprehensive Cancer Center (SCCC) at the University of Miami School of Medicine to design and implement the registry. FCDS has been collecting incidence data since 1981. In 1994, FCDS became part of the National Program of Cancer Registries (NPCR) administered by the Centers for Disease Control (CDC). Through this program, CDC provides funding for states, such as Florida, to enhance their existing registry to meet national standards for completeness, timeliness and data quality. The standards are set forth by the North American Association of Central Cancer Registries (NAACCR), the American College of Surgeons, Commission on Cancer (ACoS/CoC) and the Surveillance Epidemiology and End Results (SEER) reporting program of the National Cancer Institute (NCI). Florida has one of the highest crude incidence rate of cancer in the nation with a 18.8 million population[1] residing in 67 counties.

Two hundred thirty hospitals report over 200,000 cases annually, which when unduplicated, translate into approximately 115,000 newly diagnosed cases per year. At this time, the FCDS database contains almost 4 million cancer incidence records.[2] FCDS also maintains a cancer mortality file based on data provided from the State of Florida Bureau of Vital Statistics. The mortality data are linked with the incidence data and provide access to "passive" follow-up data. NPCR-funded states such as Florida are not

[1] Data source: 2010 Census, see https://www.census.gov/quickfacts/table/PST045216/12#.

[2] The count in the so called materialized view (see below) was 3,889,838 on May 10, 2017.

funded for active follow-up. To obtain more complete death information, CDC funds Florida to conduct NDI linkage at least every other year. Cases not known to be deseased by the NDI linkages will be assumed alive and censored at the end of the study period. This is known as presumed alive.

The survival analysis of FCDS is also a data science project using Stata. In principle, the goal of scientific publication is to enable reproducibility of research findings. To meet this goal, one should share the underlying code and data. To work reproducibly, one programs the workflows of data analysis. The popular term for programming the workflow of data analysis is Data Science. See figure 1:

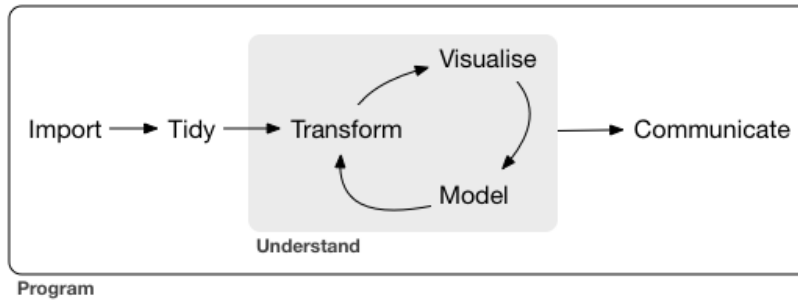

Figure 1.1: A standard model of data science (Wickham and Grolemund, 2016, ix).

The primary computing language is Stata because currently only Stata can estimate net survival in a relative framework using life tables. Stata is less popular for data science than R and Python because Stata is commercial, and it has few capabilities in machine learning and in document authoring. For this data science project, machine learning is irrelevant so the Stata issue is document authoring.

Stata is partially dependent on SAS for the project. This report uses Stata 14.2[3] and SAS 9.3. For Stata, type

```
. update query
```

in Stata and follow the instructions to ensure that you are up to date. The "dot prompt" is Stata asking for something to do. Stata has excellent online help. To obtain help on a command (or function) type `help` *command_name*, which displays the help on a separate window called the *Viewer*. Or just select `Help|Stata Command...` on the menu system. If you do not know the name of the command you need you can search for it. Stata has a search command that will search the documentation and other resources, type `help search` to learn more. For the report, you will need the

[3]Stata 15 was released on June 6, 2017. The main new applicable command is `putpdf` to create PDFs with embedded Stata results using Markdoc instead of LaTeX. The user-written Stata Markdown command `markstat` was released on September 25, 2017 (Rodríguez, 2017). `markstat` can produce HTML and PDF documents with the same script and it has other distinctive features such as cleaner scripts and support of citations.

user-written programs `stnet` (Coviello et al., 2015) and `strs` (Dickman and Coviello, 2015) for survival analysis, and `texdoc` (Jann, 2016) for document authoring.

Although this report should be useful for anyone who wants to analyse FCDS survival data, there are several constraints. First, I use Stata (for Windows). The main software alternatives are R and SAS. Other alternatives are Python and SEER*Stat. A separate Supplement has code for R and some advice for users of Python, SEER*Stat, and SAS respectively (Alexandersson, 2017a). A second constraint is that the analysis dataset used for this report and your own analysis dataset will differ because the source data are dynamic. Finally, the raw data are confidential and may not be shared with the public. You need an approved data request if you are not affiliated with FCDS and want to similar ad hoc data.

This report is justified, it is argued here, for three reasons. First, there is a **lack of reproducible research on FCDS data**. The term *reproducible research* refers to the idea that the ultimate product of research is the paper along with the linked executable code and data. According to a 2016 poll of 1,500 scientists, 70% of them failed to reproduce another scientist's experiments and 50% failed to reproduce their own experiment (Baker, 2016). Reproducibility or re-analysis using the same data is easier than replication using new data. Public use datasets would help. NAACCR soon will provide a CINA public use (non-confidential) dataset for NCI-SEER data.[4] FCDS, which instead belongs to CDC-NPCR, does not provide a public use dataset at the record level. FCDS provides public use datasets but only at the aggregate, non-record level.[5] In addition to lack of data, a second big problem for reproducibility is lack of code. This FCDS technical report of FCDS data is reproducible by showing the code that was used and how to request the data.

Second, **FCDS data and SEER data differ** in some important ways. For survival analysis, the most important difference between FCDS and presumed-alive SEER data is that FCDS, unlike SEER, does not provide a variable for birth month. The variable is collected but it is not made available for release. There are other data differences too, for example in requirements and in quality evaluation. For instance, the SEER data completion method requires more historical data than does the NAACCR method. FCDS is NAACCR GOLD certified, not SEER certified.

Third, there is a **lack of standard approach to population-based cancer survival analysis**. SEER's guideline for measures of cancer survival distinguishes between the survival measure (crude or net) and the framework or estimation method (cause of death or expected survival) [6]. This report will instead apply the guideline in Dickman and Coviello (2015, 187). The two measures are the same: crude and net survival. Dickman and Coviello (2015) additionally distinguishes between the framework for estimating the chosen measure (cause-specific and relative) and the available estimators. Also, Dickman and Coviello (2015) but not SEER prioritizes the measures over the framework.

The technical report consists of two chapters, an appendix, and a supplement. Chapter 1, Data Management, outlines the four survival analysis approaches and it demon-

[4] See https://www.naaccr.org/cina-public-use-data-set/.

[5] See the interactive webpage https://fcds.med.miami.edu/inc/statistics.shtml.

[6] See https://surveillance.cancer.gov/survival/measures.html.

strates how to create an analysis dataset. The analysis dataset is for all adult cancer cases diagnosed in Florida in 1999-2003 with 10-year follow-up. Chapter 2, Survival Analysis, gives examples of the four survival analysis approaches using the FCDS analysis dataset from chapter 1. The main recommended approach for survival analysis of FCDS is net survival in a relative framework using life tables. A provided example is estimated 10-year net survival for lung cancer cases diagnosed in Florida from 1999-2003. The appendix provides a sensitivity analysis. The separate supplement discusses alternative software such as SAS and SEER*Stat, and it includes code for running R in Stata (Alexandersson, 2017a). A companion monograph (Alexandersson, 2017b) provides net survival rates for all ten FCDS cancer site groups, and those rates are calculated as explained in this technical report.

1.2 A typology of survival analysis

1.2.1 Basic concepts

Survival analysis is full of jargon: truncation, censoring, hazard rates, etc. The key to mastering survival analysis lies in grasping the jargon.

By **time**, we mean *analysis time* such as years of age or days since diagnosis. From now on, we will write *time* to mean time as you have it recorded in your data and t to mean analysis time. Analysis time is like time, except that 0 has a special meaning; t=0 is the time of onset of risk, the time when failure first became possible. Analysis time is usually not what is recorded in a dataset. A dataset of patients might record calendar time. Calendar time must then be mapped to analysis time. The letter t is reserved for time in analysis-time units. The term time is used for time measured in other units. The origin is the time corresponding to t=0, which can vary subject to subject. That is,

$$t = \frac{time - origin}{scale}$$

By **event**, we mean *failure event* such as death, disease, relapse, recovery or any designated event under analysis. The failure event is of special interest in survival analysis, but there are other important events, such as the exposure event, from which analysis time is defined.

Many concepts in survival analysis depend on some understanding of mathematical statistics. The three key mathematical functions in survival analysis are the survival function, S(t); the hazard function, h(t); and the cumulative hazard function, H(t). The three functions are essentially just transformations of one another.

Let the random variable T be the survival time (that is, time to event) since the origin of the study (t=0). We shall assume that T is continuous unless we specify otherwise. The **survival function**, S(t), is the probability of surviving beyond time t:

$$S(t) = P(T > t)$$

The survival function is equal to one at t=0 and decreases towards zero as t goes to infinity. As defined here, it is right-continuous, that is, estimated at the right-

hand endpoint. Sometimes, the survival function is instead defined as $S(t) = P(T >= t)$ which is left-continuous, that is, estimated at the left-hand endpoint. The issue arises with discrete (step-function) survival analysis, e.g., the Kaplan-Meier estimate discussed later. The two most common methods for estimating the survival function are the actuarial (life-table) method and the Kaplan-Meier (product-limit) method. A life table tabulates the general population mortality rate by various demographics usually age, sex, and calendar period, but also sometimes by sub-region, ethnicity, and socio-economic status. Life table methods are well-suited to cancer registry data, where datasets are large and exact survival times in days cannot be established with any precision.

The **hazard function**, h(t), is the instantaneous rate of failure, meaning that it has units 1/t. To consider a formal definition of the hazard, first consider an event occurring in a time interval from t to t + Δ (where Δ is positive), that is, $t < T \leq t + \Delta$. The hazard function is the probability that, given that a subject has survived beyond time t, he or she fails in the next small interval of time, divided by the length of that interval:

$$h(t) = \lim_{\Delta \to 0} \frac{P(t < T \leq t + \Delta \mid T > t)}{\Delta}$$

The **cumulative hazard function**, H(t), is the integral of the hazard function h(t), from 0 (the onset of risk) to t. It is the total amount of risk, that is, the area under the hazard function up to time t:

$$H(t) = \int_0^t h(u)d(u)$$

The relationship between the cumulative hazard function, H(t), and the survival function, S(t) is

$$S(t) = exp\{-H(t)\}$$
$$H(t) = -ln\{S(t)\}$$

Two common features in survival data are right-censoring and left-truncation. **Censoring** occurs when subjects are observed for the whole duration of a study, but the exact times of their failures are unknown. In common usage, censoring without a modifier means right-censoring. *Left-censoring* occurs when the exact time of failure is unknown because the event has not happened by the end of the observation period. An observation is *right-censored* when the exact time of failure is unknown because the event happened after the end of the observation period. A subject leaves the study before an event occurs (withdraws or is lost), or he study ends before the event has occurred. For example, a patient's time of death is right censored if the patient survives until the end of a study. *Interval-censoring* occurs between two known time points but we do not observe exactly when failure occurred.

Truncation, unlike censoring, occurs when subjects are observed only if their failure times fall within a certain observational period of study. Truncation is deliberate and due to study design. *Left-truncation* occurs when subjects have been at risk before entering the study (a.k.a. "delayed entry"). *Right-truncation* occurs when the entire

study population has already experienced the event of interest. Right-truncated data typically occurs in registries. FCDS includes only subjects who developed cancer, and therefore survival data obtained from FCDS will be right-truncated. Right-truncation is much more difficult to accommodate than left truncation. Therefore, right-truncation is almost always ignored in models on cancer registry data. *Interval-truncation* occurs when gaps in the data exist and the researcher does not know whether an event or not has occurred in the time gap in question.

Due to the presence of right-censored and left-truncated data, most models and estimation methods are based on the hazard function. Examples are the Kaplan-Meier estimate of the survival function and the Cox proportional regression model.

The problem with ordinary least-squares (OLS) linear regression for survival analysis is with the assumed normality of the residuals, e_j (Cleves et al., 2016, 2). Substituting a more reasonably distributional assumption for e_j leads to *parametric* survival analysis. *Semiparametric* survival analysis make assumptions about covariates but not about the distribution of failure times. *Nonparametric* survival analysis make no assumptions about neither the covariates nor the distribution of failure times.

In nonparametric analysis, the effects of covariates are not modeled. The Kaplan-Meier function is, basically, the life table method where the interval size is decreased towards zero so that the number of intervals tends to infinity. The Kaplan-Meier function is also known as the product-limit method since it is a limit of the life-table method where S(t) is estimated as a product of interval-specific proportions. In small samples, the Kaplan-Meier product-limit estimator is better when estimating the survival function and the Nelson-Aalen estimator is better when estimating the cumulative hazard function. In very large samples, it does not matter whether you use the Kaplan-Meier estimator or the Nelson-Aalen estimator. The log-rank test is perhaps the most commonly used nonparametric test for comparing two survival curves. Several nonparametric tests are available, including weighted log-rank tests. Nonparametric analysis is always a useful starting point.

In most real-world applications, you will be forced into parametric or semiparametric analysis. The Cox proportional hazards model, which assumes that covariates multiplicatively shift the baseline hazard function, is by far the most popular semiparametric model. When subjects are tied (fail at the same time) and the exact ordering of failure is unclear, the situation requires special treatment. There are many ways to test the proportional hazards assumption. The hazard-ratio estimate is almost routinely used to summarize the difference between two groups for the Cox proportional hazards model. Alternative summary measures are based on the restricted mean survival time (RMST).

In parametric models, time plays a real role. The exponential model is the simplest of the parametric survival models because it assumes that the baseline hazard is constant. Flexible parametric models are more popular because they are more flexible. For example, flexible parametric models can include cure models. When net survival is estimated in the relative framework, the cure fraction is the proportion of cancer patients whose survival experience is equivalent to the general cancer-free population. Another example, flexible parametric models can be on the log-hazard scale; see `strcs`.

The event of interest can occur more than once in a participant, for example recurrence of cancer. The majority of survival analyses focus only on time to the first

event. Several statistical models have been proposed for analysing multiple events. The models are widely known as **multistate** or *semi-competing risk* models. When death is the outcome, then clearly it is not possible to have more than one event.

Competing risks are a special case of multistate models in which each of the different events are absorbing states. The two most important measures for competing risks are the cause-specific hazard and the cumulative incidence function (CIF). The cause-specific hazard is used instead of the hazard or cumulative hazard. The CIF is used instead of the survival function. The CIF is also known as crude probability. In contrast, net probability for standard non-competing risks is also known as marginal probability.

Some research questions are complicated for which **joint modeling** of longitudinal and survival data are appropriate. Three types of joint analysis may be considered: 1) evaluation of the effects of time-dependent covariates on the survival time; 2) adjustment for informative dropout in the analysis of longitudinal data; and 3) joint assessment of the effects of baseline covariates on the two types of outcomes. Multistate models, joint models, and other advanced models such as multilevel models are beyond the scope of the report.

For a Stata-specific introduction to survival analysis, see Cleves et al. (2016). In Stata, the command `snapspan` converts snapshot data to time-span (duration form) data. The command `stset` declares time-span data to be survival-time data. In Stata, events occur at the end of the recorded time span. The option `failure()` specifies the failure event. The option `id()` specifies the ID variable if you have multiple-record data. *Getting the `stset` right is the key to survival analysis in Stata.*

The four most common **measures** of survival in the literature are overall survival, relative survival ratio, crude survival, and net survival (Perme et al., 2016, 2). Overall (a.k.a. observed or all-cause) survival is the probability that a patient is still alive at a certain time point t after the diagnosis. The most frequently used method for calculating the observed survival is the Kaplan-Meier method. The relative survival ratio compares the overall survival to the expected survival from the general population. Crude survival is the survival in presence of competing risks. Net survival is the survival in absence of competing risks. Crude and net survival distinguish between two causes of death: death due to cancer and death due to other causes. Overall survival and relative survival ratio do **not** make this distinction.

A more mathematical way to distinguish between net survival and relative survival is to distinguish between an average ratio and a ratio of averages. The order of calculation likely produces different results when you calculate an average ratio or a ratio of averages. For example, with two values 1/2 and 3/4, the average ratio is 62.5% and the ratio of averages is 67%. Yule (1934) noted that a relative death or mortality rate can be either an average ratio or a ratio of averages. Similarly, Perme et al. (2012) noted that net survival is the average ratio of overall and population survival whereas relative survival ratio is the ratio of averages of overall and population survival.

1.2.2 Approaches

It is important to settle on a typology since there are many synonyms for the same concept. This technical report uses the typology in Dickman and Coviello (2015, 187). The typology distinguishes between the measures (crude and net probabilities), the framework (cause-specific or relative) for estimating the chosen measure, and the estimators available within the chosen framework. Based on the research question, we estimate either crude survival which accommodates the competing risks or net survival which ignores the competing risks. It is a 3-step approach: First, one determines the measure. Second, one determines the framework. Third, one determines the estimator.

Perme et al. (2016) proposed a similar typology and survival analysis approach. The Perme typology uses data setting instead of framework. The Perme approach has only two steps: First, one determines the measure. Second, one determines the estimator within a given data setting. The author prefers the Dickman typology because both data settings or frameworks require assumptions.

The Dickman typology and the Perme typology both differ sharply from the traditional NCI typology on the web page https://surveillance.cancer.gov/survival/measures.html. The NCI typology uses estimation method (cause of death and expected survival) instead of framework or data setting. The resulting cells, for example "relative survival" and "cause-specific survival", are "survival statistics". Each typology needs to be evaluated on its own merits. The author does not use the NCI typology primarily because it ignores the Pohar Perme estimator of net survival.

The cause-specific framework requires accurate classification of cause-of-death (COD). At FCDS, the COD variable is NAACCR item #1910 which requires additional approval from the Florida Office of Vital Statistics. The relative framework requires appropriate estimation of expected survival, which can be done using either life tables or modeling. Figure 2 is a 2*2 table of the four survival approaches (Dickman and Coviello, 2015, 187), and the recommended FCDS usage.

		Framework	
		Cause-specific	Relative
Measure	Crude	Registry-based randomized controlled trial (RRCT)	Risk communication
	Net	Causality with observational data	Life tables

Figure 1.2: Survival analysis approaches (Dickman and Coviello, 2015, 187) and recommended FCDS usage.

Crude survival in a cause-specific framework typically is useful for registry-based randomized controlled trials (RRCTs). RRCTs are discussed in, for example, Li et al. (2016). For FCDS, an RRCT would probably require a linkage data request and possibly also follow-back investigation of the matched patients. An alternative to RRCT is FCDS data enhanced for Comparative Effectiveness Research (CER). FCDS has a CER dataset but it is only for the diagnosis year 2011 for five counties.[7] Stata commands for this approach are `stcompet`, `stcrreg`, `stpm2`, `stpm2cif` and `stcrprep`.

Crude survival in a relative framework is useful for patient-risk communication. For example, if patients do not understand hypothetical-world explanations, one should report crude (real world) survival rather than estimate net survival and then describe it as something else. Another example, cancer patients diagnosed today may be more interested in actual risk than in hypothetical risk. Stata commands for this approach are `strs` with the `cuminc` option which uses life tables, and `stpm2cm` which is model based.

Net survival in a cause-specific framework is useful for causal inference with observational data. The standard estimators, for example as in Kaplan-Meier and in Cox regression, are for this approach. In addition, Stata 14 has the command `steffects` for treatment-effects estimators.

Net survival in a relative framework (Dickman and Coviello, 2015) is the primary focus in this report. The Stata commands are textttstns, `strs` and `stnet`. Age standardization is recommended. There are three international cancer survival standards (ICSS) for age standardization according to cancer site. ICSS 1 is for cancer sites with increasing incidence by age. This covers most cancer sites, including lung cancer. ICSS 2 is for cancer sites with broadly constant incidence by age. ICSS 3 is for cancer sites that mainly affect young adults.

The age variable is often split into categories. For example, SEER*Stat provides population weights by 5-year and 10-year groups.[8] The FCDS annual reports use four larger age groups: 0-14, 15-39, 40-64, and 65+. Unfortunately, these wider age-categories may not describe the data well.

The often preferred estimator of net survival in a relative framework is the Pohar-Perme estimator, which was developed for continuous survival times because it is unbiased. In contrast, a model-based estimator requires a high degree of experience and expertise (UKIACR, 2016, 8). One possible workaround is to standardize the modeling so that the result is a cancer survival index (CSI). Johnson et al. (2016b) did this for SEER in the CINA Survival 2016 (Johnson et al., 2016a) as "All Sites (Standardized)". The CSI has no clinical interpretation. Therefore, the monograph will *not* display survival rates for cancers combined. Stata can use the Pohar Perme estimator on discrete survival times using a life-table approach. The life-table approach is less sensitive than the time-continuous approach to the precision of survival times (Seppä et al., 2015). A life-table approach seems prudent because FCDS has only discrete birth dates (i.e., birth years) because DOH allows FCDS to release birth year only, not full date of birth. See the appendix for a brief comparison with full birth data.

Four different approaches for relative survival calculation are cohort, complete, pe-

[7] See https://fcds.med.miami.edu/inc/cer.shtml.

[8] See http://seer.cancer.gov/stdpopulations/survival.html.

riod, hybrid. The major difference in the the four approaches is in the case selection. The cohort approach is recommended when publishing standard and routine data tables, and when making international comparisons (UKIACR, 2016, 9). Therefore, this report will use the cohort approach despite that it is the least up-to-date.

It is difficult to recommend typical FCDS usage for each survival analysis approach as has been done in this report. For example, you typically use life tables in the relative framework not only for net survival but also for crude survival. However, the typical usage or way in which the life tables are used differs in the two approaches depending on the core idea: being technically correct as in life tables or reducing the technical jargon as in risk communication.

1.3 Data management

1.3.1 "Import": How to access the FCDS data

All new data requests must be submitted to FCDS via the Data Request Automated Management System (DREAMS). There are specific procedures and fees for data release based on the category of request. For more information about FCDS data requests, click on the link "Data Requests" on the FCDS website.[9]

The data requirements for survival analysis are the same regardless of type of data request. As a data requestor, you need an approved ad hoc data request or an approved linkage data request for doing survival analysis. See the FCDS webpage for details about FCDS data requests. The FCDS database is in Oracle 11. To make internal queries easier, FCDS has developed a materialized view named "mv_datarequest". A materialized view is a table that stores a snapshot of the query. By materializing the view, the Stata query below runs on my PC in 10-15 seconds instead of 10-15 minutes. The Stata SQL statement is easier to maintain in a separate local macro, which here is named "sql_statement". The materialized view specifies the NAACCR variable names, for example "PATIENT_ID_NUMBER_N20". FCDS recently developed a tab in DREAMS named "Extract Criteria" for selecting rows (a.k.a. records in a database or observations in a dataset). Three types of extract criteria are possible for new data requests from the tab:

- **Demographics - Required** Sex, Ethnicity, Race, Vital Status, County of Residence at DX, Age at DX
- **Tumor characteristics - Required** Years of DX, Primary Site, ICDO3 Morphologies, Stage at DX
- **Any Additional Characteristics - Optional** <Variable Name> <Parameters>

The FCDS website lists the number of new cancer cases each year. The structure of the published annual reports has remained the same since 2009. The published yearly counts for 2009-2013 are 103,783 (2009), 103,855 (2010), 107,082 (2011), 106,166 (2012) and 108,829 (2013). The counts in the analysis dataset will not match the published

[9] See https://fcds.med.miami.edu/inc/datarequest.shtml.

counts for three reasons: 1) I will remove summary stage errors, 2) I will exclude single non-Florida observations, and 3) the published numbers are typically not updated. The published counts and rates instead use the static table RATES_ABSTRACT. Here is the Stata code for the SQL statement in this report:

```
. local sql_statement ///
> SELECT ///
>     Patient_Id_Number_N20, ///
>     Addr_at_DX_State_N80, ///
>     County_at_DX_N90, ///
>     Race_1_N160, ///
>     Sex_N220, ///
>     Age_at_Diagnosis_N230, ///
>     Birth_Year_N240, ///
>     Sequence_Number_Central_N380, ///
>     Date_of_Diagnosis_N390, ///
>     Type_of_Reporting_Source_N500, ///
>     SEER_Summary_Stage_2000_N759, /// for dx_year >= 2001
>     SEER_Summary_Stage_1977_N760, /// for dx_year 1981-2000
>     Derived_SS2000_Flag_N3050, /// _N3040 is omitted as a mistake on purpose
>     Date_of_Last_Contact_N1750, ///
>     Vital_Status_N1760, ///
>     FCDS_Site_Group_N2220 ///
> FROM mv_datarequest ///
> WHERE  (County_at_DX_N90 between 1 and 133) and /// FL counties (remove 998, 999)
>     (Date_of_Diagnosis_N390 between ´19990101´ and ´20131231´) and ///
>     ( (substr(Date_Of_Diagnosis_N390,1,4) < ´2001´ and ///
>         SEER_Summary_Stage_1977_N760 > 0 ) or  ///
>     (substr(Date_Of_Diagnosis_N390,1,4) >= ´2001´ and  ///
>         SEER_Summary_Stage_2000_N759 > 0) or  ///
>     FCDS_Site_Group_N2220 = 55 or SEER_Summary_Stage_2000_N759 is null) and  ///
>     FCDS_Site_Group_N2220 <= 80 and Age_at_Diagnosis_N230 <> 999 and  ///
>     Derived_SS1977_Flag_N3040 in (´1´,´2´) and  /// no SS_1977 error
>     Derived_SS2000_Flag_N3050 in (´1´,´2´) and  /// no SS_2000 error
>     EXISTS (SELECT abshist_patient_id /// exclude single non-FL obs
>         FROM abshist /// table
>         WHERE abshist_patient_id = Patient_Id_Number_N20 and ///
>         abshist_central_seq = Sequence_Number_Central_N380 and ///
>         (abshist_medical_facility between ´1100´ and ´9999´ or ///
>         abshist_medical_facility in (´0510´)));
```

For the required demographics, I only selected all Florida counties. For the required tumor characteristics, I only selected years of diagnosis 1999-2013; the code differs slightly from the preset SQL script for FCDS staff. The rest of the WHERE clause in the SQL code above is an example of possible optional additional characteristics translated into SQL. The code "FCDS_Site_Group_N2220 <= 80" above selects only malignant sites. Summary stage is at least localized (values 1-9) except urinary bladder cancer can be in situ (value 0); this is both FCDS and SEER standard for reporting.

The final part of the SQL WHERE clause is the SQL statement EXISTS which uses the table ABSHIST. Therefore, it is not available in DREAMS. It removes records that are consolidated entirely on out of records. Arguably the code should be added to all data requests and for routine reporting (unless DOH says otherwise) or until the code is implemented in the materialized view itself.

The easiest way to hide the ODBC password is to put it in a local macro and then

hide the command but keep the output.[10]

```
texdoc stlog, cmdstrip
local password "*******"
texdoc stlog close
```

The `timer` command starts, stops, and reports interval timers. This can be useful for SQL queries because theu can take a long time to run. In Stata, you create SQL queries using the `odbc load` command. The `exec()` option allows you to select only the wanted rows. Stata is case sensitive.

```
. timer clear
. timer on 1
. odbc load, ///
>     user(webuser) password(`password´) dsn(Oracle64) /// connect_options
>     datestring clear  exec("`sql_statement´") // load_options
. timer off 1
. rename _all, proper
```

The easiest approach to organizing project files is to start with a carefully designed directory structure. The first step in naming files and directories is to pick a short mnemonic for your project such as DOH. My working directory is `F:/DOH/`. Stata uses the forward-slash to separate directory levels on all platforms, even Windows. You can set the working directory with the command `cd` but I do not recommend it. It is better to use relative paths, not absolute paths.

The data requestor does not have to worry about the workings of DREAMS or SQL queries. FCDS should provide the data requestor a comma-delimited dataset. The dataset in this report is from 2017 and for FDOH, so I name it "2017_DOH Dataset.txt".

```
. pwd
F:\doh
. export delimited using "2017_DOH Dataset.txt", replace quote
file 2017_DOH Dataset.txt saved
```

The data requestor can use `import delimited` to import the comma-delimited text dataset. The default is to read variable names as lowercase. Use the option `case(preserve)` to preserve the case. Use option `stringcols()` to read dates as strings.[11]

```
. import delimited using "2017_DOH Dataset.txt", clear case(preserve) ///
>     stringcols(7 9)
(16 vars, 1,607,209 obs)
```

1.3.2 "Tidy": How to clean the FCDS data

The `isid` command checks for unique identifiers.

```
. isid Patient_Id_Number_N20 Sequence_Number_Central_N380
```

[10] In `texdoc`, this means using the `cmdstrip` option. In `markdoc`, you would instead use the notation marker `/**/`. The following posting on Statalist gives more advice about passwords for Stata: http://www.statalist.org/forums/forum/general-stata-discussion/general/6323-encoding-odbc-call-passwords.

[11] Most statistics software can handle this. But sometimes you need specialized software. A very useful software for converting datasets is `Stat/Transfer` at www.stattransfer.com.

Data cleaning is often an iterative process. Here, the data cleaning process is iterative due mostly to the need for using a SAS program; there is no available equivalent Stata program. The background context is that population-based cancer survival analysis in the United States often is done on SEER data. The SEER data documentation contains a lot of information.[12] The most important link is "Months Survived Based on Complete Dates".[13] It contains a SAS program "CalculateSurvivalTimeInMonths.sas" that uses day information for the survival calculation. The SAS program creates these five variables for presumed-alive data:

- **1785** Surv-Date Presumed Alive (date_lc)
- **1786** Surv-Flag Presumed Alive (pa_surv_flag)
- **1787** Surv-Months Presumed Alive (pa_surv_mon); This is *complete* months or survival
- **1788** Surv-Date DX Recode (date_dx)
- **2220** Record order (record_order)

The date presumed alive variable derives the survival variables. The flag presumed alive variable will enable analysts to easily select a subset of cases. The months presumed alive variable is used for the survival analysis.[14] The survival date of diagnosis recode is calculated using the date of diagnosis (NAACCR item #390) with imputed values if the day or month is unknown or not available. The variable for record order addresses a problem with the variable sequence number central (NAACCR item #380). The problem is that sequence number central is not chronological if non-federally reportable tumors are included. That is, to make the sequence numbers 60-89 chronological, you need a complete date to sort by. Rather than saving a complete date variable, the SAS program saves a record order variable. Stata needs to run the SAS macro to create the NAACCR variables because there is no equivalent Stata code.[15]

It is more efficient to run the SAS program "CalculateSurvivalTimeInMonths.sas" *before* you clean the data in Stata. The reason is that the SAS program creates new variables which also need to be cleaned. The SAS program requires text data with fixed format, at specific column numbers, with a length of 3339 characters.

The easiest solution to create the data that the SAS program needs is to use the user-written program `outfixt`. You need the undocumented `cap` option, which captures errors that arise with very long lines, and a buffer variable to fill up the observation ("record" in SAS terminology). The command `order` can change the variable order but it is not needed here.

[12] See http://seer.cancer.gov/analysis/.

[13] See http://seer.cancer.gov/survivaltime/.

[14] The variable is calculated as: Survival months = FLOOR((endpoint – date of diagnosis) / days in a month) The FLOOR function always rounds down, e.g., FLOOR(1.68) = 1. The actual length of the year is 365.2422 days. Days in a month is assigned to 365.24/12. For comparison, the Gregorian calendar averages 365.2425 days, and the former Julian calendar averaged 365.25 days.

[15] The problem is not specific to Stata. For instance, SEER*Prep and SEER*Stat expect the tumors for a patient to be sorted chronologically.

Text files are large and time consuming to read and write. You may have to split the file with the `if` or `in` qualifier, if for example you have over a million records. For early drafts, I selected only lung cancer data. The lung cancer dataset was <250,000 observations and the disk space wwas <1GB which is manageable for repeated runs.

```
. gen str1 buffer = "z"
. timer on 2
. outfixt _all using "input.txt", ///
>     cols(  42 145 156 177 192 193 196 528 530 563 904 905 1160 2116 2126 2340 3339) ///
>     flist(%8s %2s %3s %2s %1s %3s %4s %2s %8s %1s %1s %1s  %1s %8s  %1s  %2s   %1s) ///
>     replace cap dct("2017_DOH Fixed.dct", replace)
(note: file input.txt not found)
. timer off 2
```

The user-written command `saswrapper` runs SAS code in Stata. The SAS program creates five new variables. Two changes are required in the SAS program "CalculateSurvivalTimeInMonths.sas":

(1) Add the working directory. I added `x "cd F:/DOH/";` at the top of the program.
(2) Add the %INCLUDE option `LRECL`.[16]

```
. timer on 3
. qui saswrapper using CalculateSurvivalTimeInMonths.sas
. timer off 3
```

The output, unless you change the SAS program more, is the same fixed-format text data with the filename "myoutputfile.txt". To read in the dataset, use `infix`. `timer list` lists the times of the timers.

```
. timer on 4
. infix Patient_Id_Number_N20 42-49 str Addr_at_DX_State_N80 145-146 ///
>     County_at_DX_N90 156-158 Race_1_N160 177-178 ///
>     Sex_N220 192-192 Age_at_Diagnosis_N230 193-195 ///
>     Birth_Year_N240 196-199 Sequence_Number_Central_N380 528-529 ///
>     str Date_of_Diagnosis_N390 530-537 Type_of_Reporting_Source_N500 563-563 ///
>     SEER_Summary_Stage_2000_N759 904-904 SEER_Summary_Stage_1977_N760 905-905 ///
>     Derived_SS2000_Flag_N3050 1160-1160 ///
>     str Date_of_Last_Contact_N1750 2116-2123 Vital_Status_N1760 2126-2126 ///
>     str date_lc_1785 2305-2312 ///
>     surv_flag_1786 2313 ///
>     surv_mon_1787 2314-2317 ///
>     str date_dx_1788 2318-2325 ///
>     FCDS_Site_Group_N2220 2340-2341 ///
>     record_order 2510-2511 ///
>     using "myoutputfile.txt", clear
(1,607,209 observations read)
. timer off 4
. timer list
   1:     32.65 /        1 =      32.6520
   2:   1373.39 /        1 =    1373.3910
   3:   2241.35 /        1 =    2241.3530
   4:    118.29 /        1 =     118.2900
```

Recall what the timers are for: 1 runs the SQL query, 2 creates the text dataset, 3 runs

[16] Specify a value of at least 3339. The SAS program runs as a defaults to a record length of 256 characters. The length limitation is documented in SAS Usage Note 15883 at http://support.sas.com/kb/15/883.html. I added `option LRECL=32767;` in the program.

the SAS program, and 4 reads in the SAS-modified text dataset. The timers show that the slowest timers are 2 and 3. That is, the bottleneck is creating the text dataset and running the SAS program which takes about 2 hours in total. The time is reduced to about 8 minutes if we instead would run the program only on the lung cancer data.

A disadvantage of text files is that variable labels are lost. Every variable should have a variable label. The variable names as variable labels is better than no variable labels.

```
. foreach v of varlist * {
  2.     label variable `v´ "`v´"
  3. }
```

Be careful with capitalizations. The Stata convention is lowercase. In general, rename long variable names to shorter, yet still informative names. A number of commands abbreviate long variable names to something that can be difficult to read.

```
. rename (Patient_Id_Number_N20 Addr_at_DX_State_N80 County_at_DX_N90 ///
>     Race_1_N160 Sex_N220 Age_at_Diagnosis_N230 Birth_Year_N240 ///
>     Sequence_Number_Central_N380 Date_of_Diagnosis_N390 ///
>     Type_of_Reporting_Source_N500 SEER_Summary_Stage_2000_N759 ///
>     SEER_Summary_Stage_1977_N760  ///
>     Date_of_Last_Contact_N1750 Vital_Status_N1760 FCDS_Site_Group_N2220 ///
>     Derived_SS2000_Flag_N3050) ///
>     (pid_20 state_80 county_90 race_160 sex_220 age_dx_230 doby_240 ///
>     seq_380 date_dx_390 rpt_src_500 ss2000_759 ss1977_760 date_lc_1750 ///
>     vital_1760 site_group_2220 ss2000_fl_3050)
```

You can easily check for unique ID variables. You typically want to sort by the ID variables. As mentioned in the introduction, sequence order is not chronological for values 60-89. Therefore, `seq_380` is no longer the within-person identifier after running the SAS program. Instead `record_order` is the within-person identifier.

```
. isid pid_20 record_order
. sort pid_20 record_order
```

1.3.3 "Transform": How to create the FCDS analysis dataset

Create a variable for the ten cancer site groups that FCDS reports on. Create a variable for year of diagnosis. Verify that the years are 1999-2013. Document the new variable with a note and label. Binary variables should always be valued 0 for negative outcomes and 1 for positive outcomes. Therefore, I recode the variable sex_220 into a new variable male.

```
. recode site_group_2220  (36 = 1) (51 = 2) (43 = 3 ) (14/24 = 4) (55 = 5 ) ///
>     (1/10 34 35 = 6) (66/67 = 7) (41 = 8) (47 = 9 ) (44 = 10) ///
>     (11/13 25/33 37/40 42 45/46 48/50 52/54 56/65 68/80 = 11 "Other")  ///
>     (99 = .) , ///
> gen(site_10group)
(1588417 differences between site_group_2220 and site_10group)
. label variable site_10group "RECODE of site_group_2220 (1-10)"
. label define sitelab 1 "Lung & Bronchus" 2 "Prostate" ///
>     3 "Breast" 4 "Colorectal" 5 "Bladder" 6 "Head & Neck" ///
>     7 "Non-Hodgkin" 8 "Melanoma" 9 "Ovary" 10 "Cervix" 11 "Other"
. label values site_10group sitelab
. gen dx_year = substr(date_dx_390,1,4)
. destring dx_year, replace
```

```
dx_year: all characters numeric; replaced as int
. assert inrange(dx_year,1999,2013)
. notes dx_year: substr(date_dx_390,1,4)
. label variable dx_year "Year of Diagnosis"
. gen male = sex_220
. replace male = 0 if sex_220 == 2
(754,598 real changes made)
. replace male = . if !inlist(sex_220,1,2)
(740 real changes made, 740 to missing)
. assert male==1 if sex_220==1  // error check
. label variable male "0=Female, 1=Male"
```

Woods et al. (2012) made a compelling case that full dates (day, month, year) rather than partial dates (month and year) should be used in population-based cancer survival studies. FCDS has the additional problem for birth date that only birth year is available for release due to confidentiality concerns. Birth months and birth days need to be randomly generated without drawing nonsensical combinations such as 2/30. Days are more difficult to generate due to leap years. The diagnosis date should also be in a date format. Survival years is a more natural unit than survival months.

```
. set seed 12345
. gen dobmr_240 = runiformint(1,12) if !mi(doby_240)
. gen dobdr_240 = runiformint(1,31) if inlist(dobm,1,3,5,7,8,10,12)
(668,292 missing values generated)
. replace dobdr_240 = runiformint(1,30) if inlist(dobm,4,6,9,11)
(534,656 real changes made)
. replace dobdr_240 = runiformint(1,28) if dobm==2  // what if leap year?
(133,636 real changes made)
. gen dob = mdy(dobmr_240, dobdr_240, doby_240)
. format dob %d
. label variable dobmr_240 "Birth month (random)"
. label variable dobdr_240 "Birth day (random)"
. label variable dob "Date of birth (random month and day)"
. gen date_dx = date(date_dx_390,"YMD")  // create date format
(3,699 missing values generated)
. label variable date_dx "date_dx_390 in date format"
. gen surv_year = surv_mon_1787 / 12  // create survival year
. label variable surv_year "Survival year"
. format date_dx %td
```

Recode age at diagnosis into age groups and age-standardized weights. The variables for age groups are agegr, agegr_prostate, agegr_seer, and agegr_fcds. The variables for age-standardized weights are icss1, icss_prostate, and icss2. The Stata code for icss1 is the same as in Coviello et al. (2015, 181) except that the variable name here is icss1 instead of standw.

```
. egen agegr = cut(age_dx_230), at(0 45(10)75 100) icodes
(910 missing values generated)
. egen agegr_prostate = cut(age_dx_230), at(0 55(10)85 100) icodes
(910 missing values generated)
```

```
. recode agegr 0=0.07 1=0.12 2=0.23 3=0.29 4=0.29, gen(icss1)
(1606299 differences between agegr and icss1)

. recode agegr_prostate 0=0.19 1=0.23 2=0.29 3=0.23478 4=0.05522, gen(icss1_prostate)
(1606299 differences between agegr_prostate and icss1_prostate)

. recode agegr 0=0.28 1=0.17 2=0.21 3=0.20 4=0.14, gen(icss2)
(1606299 differences between agegr and icss2)

. label variable agegr "ICSS standard age groups (15-44, 45-54, 55-64, 65-74, 75-99)"

. label variable agegr_prostate "ICSS prostate age groups (15-55, 55-64, 65-74, 75-84, 85-
99)"

. label variable icss1 "ICSS 1"

. label variable icss1_prostate "ICSS 1, age-adjusted for prostate"

. label variable icss2 "ICSS 2"

. recode age_dx_230 ///
>     (0/4 = 1 "0-4") (5/9 = 2 "5-9") ///
>     (10/14 = 3 "10-14") (15/19 = 4 "15-19") ///
>     (20/24 = 5 "20-24") (25/29 = 6 "25-29") ///
>     (30/34 = 7 "30-34") (35/39 = 8 "35-39") ///
>     (40/44 = 9 "40-44") (45/49 = 10 "45-49") ///
>     (50/54 = 11 "50-54") (55/59 = 12 "55-59") ///
>     (60/64 = 13 "60-64") (65/69 = 14 "65-69") ///
>     (70/74 = 15 "70-74") (75/79 = 16 "75-79") ///
>     (80/84 = 17 "80-84") (nonmissing = 18 "85+ years") ///
>     , gen(agegr_seer)
(1606427 differences between age_dx_230 and agegr_seer)

. recode age_dx_230 ///
>     (0/14 = 1 "0-14") (15/39 = 2 "15-39") ///
>     (40/64 = 3 "40-64") (nonmissing = 4 "65+ years") ///
>     , gen(agegr_fcds)
(1606427 differences between age_dx_230 and agegr_fcds)

. label variable agegr_seer "SEER age groups (18 groups)"

. label variable agegr_fcds "FCDS age groups (4 groups)"
```

Create a combined variable for stage group.

```
. assert ss2000_759==. if inlist(dx_year,1999,2000)

. count if dx_year>=2001 & !mi(ss1977_760) // should be a small number, not used
  2,386

. gen stage = ss1977_760 if dx_year<2001
(1,395,031 missing values generated)

. replace stage = ss2000_759 if dx_year>=2001
(1,395,031 real changes made)

. recode stage 0/1=1 2/5=2 7=3 8/9=4, gen(stagegr)
(831414 differences between stage and stagegr)

. label variable stage "Stage (from N760 and N759)"

. label variable stagegr "Stage Group"

. label define stagelab 1 "Localized" 2 "Regional" 3 "Distant" 4 "Unknown"

. label values stagegr stagelab
```

Exclusions: *Data quality exclusions.* Verify that there are only valid SS2000. This is implied from the SQL "WHERE clause" but the researcher does not see the SQL code and it is best practice to test for serious errors. It is not possible to test ss1977_fl_3040 since it was not selected.

```
. assert inlist(ss2000_fl_3050,1,2)
```

Standard exclusions: The most controversial standard survival analysis exclusion is to omit children (ages 0-14). Cancer for children is much more difficult to analyze than cancer for adults for two reasons: The classification of cancer for adults (ICD-O-3) emphasizes primary site whereas the International Classification of Childhood Cancer (ICCC) emphasizes morphology, which is more complicated.[17] The main survival analysis approach for FCDS for adults is age-standardized net survival in the relative framework. The approach can be used for adults only because age-standardized population weights (ICSS) are only available for adults.

```
. drop if !inlist(sex_220,1,2)  // Select only male or female
(740 observations deleted)

. drop if age_dx_230<15  // drop children
(8,224 observations deleted)

. drop if !inrange(age_dx_230,0,126)  // Select only known age
(0 observations deleted)
```

Expected survival table exclusions:

```
. drop if inrange(age_dx_230,100,126) // drop invalid age year
(908 observations deleted)

. assert inrange(race_160,1,32) | inrange(race_160,96,99)  // Race W, B, O

. recode race 3/98=3 99=4, gen(racegr)
(31750 differences between race_160 and racegr)

. label variable racegr "Race Group"

. label define racelab 1 "White" 2 "Black" 3 "Other" 4 "Unknown"

. label values racegr racelab
```

Multiple primary selection: Sequence number central is used to define first vs. multiple primary cancers. Cancer cases with a sequence number value of 0 or 1 are classified as first primary cancers, while cancers with a sequence number value of 2 or higher are classified as multiple primary cancers. Since this technical report is an introduction to survival analysis, only the first primary is used. In contrast, CINA Survival Johnson et al. (2016a, 15) allowed for multiple primary cancers per patient but only one record per patient was included in each survival estimate.

```
. drop if !inlist(seq_380,0,1)  // First Primary Only (Sequence Number 0 or 1)
(314,743 observations deleted)
```

Survival Calculation Exclusions: The Stata message "observations end on or before enter()" is the same as "Alive with no survival time" in SEER*Stat.

```
. drop if !inlist(vital_1760,0,1)  // invalid vital status
(0 observations deleted)

. tab surv_flag_1786, mi
```

surv_flag_1786	Freq.	Percent	Cum.
0	2,092	0.16	0.16
1	1,238,598	96.57	96.73
2	135	0.01	96.74
3	2,849	0.22	96.97
8	38,200	2.98	99.94
9	720	0.06	100.00

[17] FCDS uses the ICCC WHO 2008 specification for child cancers. See https://seer.cancer.gov/iccc/iccc-who2008.html for the definition of the ICCC variable.

```
------------+-----------------------------------
      Total |   1,282,594      100.00
. drop if surv_flag_1786!=1  // unknown survival duration
(43,996 observations deleted)
```

Because of the population mortality file (see below), it makes sense to rename variable sex_220 to sex. When you save a dataset, you should add a dataset label, a note, and a data signature.

```
. rename sex_220 sex
. label variable sex "Sex"
. label define sexlab 1 "Male" 2 "Female"
. label values sex sexlab
. notes: Dataset was created from draft_DOH20170630.do
. label data "Dataset for Survival Analysis, DX Years 1999-2013"
. datasignature set
  1238598:39(121503):786573920:3146713717       (data signature set)
. save doh, replace
file doh.dta saved
```

The following is an overview of the DOH analysis dataset.

```
. notes
_dta:
  1.  Dataset was created from draft_DOH20170630.do
dx_year:
  1.  substr(date_dx_390,1,4)
. datasignature confirm
  (data unchanged since 09may2017 15:42)
```

(*Continued on next page*)

```
. desc
Contains data from doh.dta
  obs:     1,238,598                          Dataset for Survival Analysis, DX Years 1999-
                                                2013
 vars:            39                          9 May 2017 15:42
 size:   182,073,906                          (_dta has notes)
```

variable name	storage type	display format	value label	variable label
pid_20	float	%9.0g		Patient_Id_Number_N20
state_80	str2	%9s		Addr_at_DX_State_N80
county_90	float	%9.0g		County_at_DX_N90
race_160	float	%9.0g		Race_1_N160
sex	byte	%8.0g	sexlab	Sex
age_dx_230	float	%9.0g		Age_at_Diagnosis_N230
doby_240	float	%9.0g		Birth_Year_N240
seq_380	float	%9.0g		Sequence_Number_Central_N380
date_dx_390	str8	%9s		Date_of_Diagnosis_N390
rpt_src_500	byte	%8.0g		Type_of_Reporting_Source_N500
ss2000_759	byte	%8.0g		SEER_Summary_Stage_2000_N759
ss1977_760	byte	%8.0g		SEER_Summary_Stage_1977_N760
ss2000_fl_3050	byte	%8.0g		Derived_SS2000_Flag_N3050
date_lc_1750	str8	%9s		Date_of_Last_Contact_N1750
vital_1760	byte	%8.0g		Vital_Status_N1760
date_lc_1785	str8	%9s		date_lc_1785
surv_flag_1786	byte	%8.0g		surv_flag_1786
surv_mon_1787	float	%9.0g		surv_mon_1787
date_dx_1788	str8	%9s		date_dx_1788
site_group_2220	float	%9.0g		FCDS_Site_Group_N2220
record_order	float	%9.0g		record_order
site_10group	float	%15.0g	sitelab	RECODE of site_group_2220 (1-10)
dx_year	int	%10.0g		* Year of Diagnosis
male	float	%9.0g		0=Female, 1=Male
dobmr_240	float	%9.0g		Birth month (random)
dobdr_240	float	%9.0g		Birth day (random)
dob	float	%d		Date of birth (random month and day)
date_dx	float	%td		date_dx_390 in date format
surv_year	float	%9.0g		Survival year
agegr	float	%9.0g		ICSS standard age groups (15-44, 45-54, 55-64, 65-74, 75-99)
agegr_prostate	float	%9.0g		ICSS prostate age groups (15-55, 55-64, 65-74, 75-84, 85-99)
icss1	float	%9.0g		ICSS 1
icss1_prostate	float	%9.0g		ICSS 1, age-adjusted for prostate
icss2	float	%9.0g		ICSS 2
agegr_seer	float	%9.0g	agegr_seer	SEER age groups (18 groups)
agegr_fcds	float	%9.0g	agegr_fcds	FCDS age groups (4 groups)
stage	float	%9.0g		Stage (from N760 and N759)
stagegr	float	%9.0g	stagelab	Stage Group
racegr	float	%9.0g	racelab	Race Group
				* indicated variables have notes

```
Sorted by: pid_20  record_order
```

```
. su
```

Variable	Obs	Mean	Std. Dev.	Min	Max
pid_20	1,238,598	2353780	476199.8	117110	3451577
state_80	0				
county_90	1,238,598	71.22879	37.16115	1	133
race_160	1,238,598	2.809342	12.58903	1	99
sex	1,238,598	1.471745	.4992012	1	2
age_dx_230	1,238,598	65.65618	13.97542	15	99
doby_240	1,238,598	1939.899	14.72764	1899	1998
seq_380	1,238,598	.1461233	.3532299	0	1
date_dx_390	0				
rpt_src_500	1,238,598	1.206108	.9973482	1	8
ss2000_759	1,072,712	3.581854	3.019903	0	9
ss1977_760	167,714	3.575331	3.183914	0	9
ss2000_~3050	1,238,598	1.299857	.4581952	1	2
date_lc_1750	0				
vital_1760	1,238,598	.4938971	.499963	0	1
date_lc_1785	0				
surv_fl~1786	1,238,598	1	0	1	1
surv_mo~1787	1,238,598	52.82199	50.01748	0	179
date_dx_1788	0				
site_gr~2220	1,238,598	43.692	16.76404	1	80
record_order	1,238,598	1.000007	.0026956	1	2
site_10group	1,238,598	5.531535	3.853146	1	11
dx_year	1,238,598	2006.074	4.333879	1999	2013
male	1,238,598	.5282545	.4992012	0	1
dobmr_240	1,238,598	6.499873	3.45094	1	12
dobdr_240	1,238,598	15.7071	8.793045	1	31
dob	1,238,598	-7160.056	5380.394	-22246	14241
date_dx	1,238,598	17004.17	1587.236	14245	19722
surv_year	1,238,598	4.401832	4.168124	0	14.91667
agegr	1,238,598	2.578519	1.237947	0	4
agegr_pros_e	1,238,598	1.724996	1.207749	0	4
icss1	1,238,598	.2380996	.0743179	.07	.29
icss1_pros_e	1,238,598	.2280398	.059128	.05522	.29
icss2	1,238,598	.1873804	.0387052	.14	.28
agegr_seer	1,238,598	13.7087	2.764725	4	18
agegr_fcds	1,238,598	3.530696	.5809905	2	4
stage	1,238,598	3.579944	3.042588	0	9
stagegr	1,238,598	1.972466	1.055118	1	4
racegr	1,238,598	1.154275	.4598755	1	4

```
. tab dx_year, mi
```

Year of Diagnosis	Freq.	Percent	Cum.
1999	81,390	6.57	6.57
2000	84,496	6.82	13.39
2001	80,839	6.53	19.92
2002	79,921	6.45	26.37
2003	78,719	6.36	32.73
2004	78,728	6.36	39.08
2005	81,329	6.57	45.65
2006	82,077	6.63	52.28
2007	83,190	6.72	58.99
2008	84,798	6.85	65.84
2009	84,855	6.85	72.69
2010	84,498	6.82	79.51
2011	85,134	6.87	86.39
2012	84,046	6.79	93.17
2013	84,578	6.83	100.00
Total	1,238,598	100.00	

```
. tab site_10group, mi
```

RECODE of site_group_2220 (1-10)	Freq.	Percent	Cum.
Lung & Bronchus	173,575	14.01	14.01
Prostate	200,488	16.19	30.20
Breast	177,494	14.33	44.53
Colorectal	121,166	9.78	54.31
Bladder	58,760	4.74	59.06
Head & Neck	45,130	3.64	62.70
Non-Hodgkin	47,810	3.86	66.56
Melanoma	49,154	3.97	70.53
Ovary	17,750	1.43	71.96
Cervix	12,562	1.01	72.98
Other	334,709	27.02	100.00
Total	1,238,598	100.00	

You no longer need the large text data files. A 100% Stata solution could use temporary files with the macro `tempfile`. With the need for SAS, it is easier to use regular permanent files followed by the Stata command `erase`. The two text data files are about 5.5GB each.

```
. erase input.txt
. erase myoutputfile.txt
```

U.S. death rates are available from the Human Mortality Database.[18] Dickman et al. (2016, 97-98) provide instructions for how to create a population mortality file in Stata. You must convert the death rates to survival probabilities.

```
. infile _year _age female male total using "Mx_1x1.txt" ///
>     if (inrange(_year,1999,2013) & _age<100), clear
´Year´ cannot be read as a number for _year[1]
´Age´ cannot be read as a number for _age[1]
```

[18] See http://www.mortality.org/.

```
´Female´ cannot be read as a number for female[1]
´Male´ cannot be read as a number for male[1]
´Total´ cannot be read as a number for total[1]
(1,500 observations read)
. drop total
. rename male rate1
. rename female rate2
. reshape long rate, i(_year _age)
(note: j = 1 2)
Data                               wide   ->   long
-----------------------------------------------------------------------------
Number of obs.                     1500   ->    3000
Number of variables                   4   ->       4
j variable (2 values)                     ->   _j
xij variables:
                             rate1 rate2   ->   rate
-----------------------------------------------------------------------------
. rename _j sex
. gen prob=exp(-rate)
. label data "U.S. death rates 1999-2013 from http://www.mortality.org/"
. label variable rate "Death rate"
. label variable prob "Survival probability"
. label variable _year "Year of death"
. label variable _age "Age"
. label variable sex "Sex (1=Male, 2=Female)"
. sort _year sex _age
. save popmort9913, replace
file popmort9913.dta saved
```

The following is a list of the first five rows in the population mortality file.

```
. use popmort9913, clear
(U.S. death rates 1999-2013 from http://www.mortality.org/)
. list sex _year _age prob in 1/5, noobs
```

sex	_year	_age	prob
1	1999	0	.9920895
1	1999	1	.9994271
1	1999	2	.9995911
1	1999	3	.999693
1	1999	4	.999764

The Human Mortality Database produces a "standard" population mortality file. In contrast, CINA Survival uses a SEER-specific population mortality file but it requires SEER*Stat Database ID 01587 (Johnson et al., 2016b, 15, 18). A similar better matched population mortality file can be created by modeling cohort data (Dickman et al., 2016, 97-98). Here, the cohort would be the 1999-2003 FCDS analysis dataset. To create such non-standard population mortality files is beyond the scope of this technical report.

Chapter 2

Survival Analysis

2.1 "Model": Estimating crude survival

2.1.1 The cause-specific framework

Data analysis should be divided between data management and statistical analysis. With a saved dataset, it is time to begin the survival analysis. As explained in Dickman and Coviello (2015), crude (and net) survival can be estimated in either the cause-specific or the relative framework. The survival analysis commands `stcomlist`, `stcompet`, `stpm2cif`, `stpm2cr` and `stcrprep` are for estimating crude survival in the cause-specific framework. I do not provide an example because this approach is the most complicated, and it is the least likely to apply to FCDS; it is only valid if FCDS is used in the study context of RRCT or CER.

2.1.2 The relative framework

As shown in Dickman and Coviello (2015, 204–205), the command `strs` with the `cuminc` option can estimate crude survival in the relative framework using life tables. The first listed cancer site in the FCDS annual reports is lung cancer. Lung cancer, also known as lung carcinoma, is a malignant lung tumor characterized by uncontrolled cell growth in tissues of the lung. The vast majority (85%) cases of lung cancer are due to long-term tobacco smoking. Lung cancer is the most common cause of cancer-related death in men and second most common in women after breast cancer. This subsection follows the example of Dickman and Coviello (2015, 204–205) but it uses lung cancer instead of colon cancer. Here is a 10-year life table for cancer patients aged 75 or over:

```
. use doh if site_group==36, clear
(Dataset for Survival Analysis, DX Years 1999-2013)
. count if surv_mon_1787==0  // stset will drop alive with no survival time
  21,036
. stset surv_year, failure(vital_1760==0) id(pid_20) // or surv_mon_1787, scale(12)
                id:  pid_20
     failure event:  vital_1760 == 0
obs. time interval:  (surv_year[_n-1], surv_year]
 exit on or before:  failure
```

```
     173575  total observations
      21036  observations end on or before enter()
------------------------------------------------------------------------------
     152539  observations remaining, representing
     152539  subjects
     127205  failures in single-failure-per-subject data
 307344.167  total analysis time at risk and under observation
                                                at risk from t =         0
                                     earliest observed entry t =         0
                                          last observed exit t =  14.91667

. strs using popmort9913 if age_dx_230>74 , breaks(0(1)10) ///
>     diagage(age_dx_230) diagyear(dx_year) ///
>     cuminc mergeby(_year sex _age) list(cr_e2 ci_dc ci_do)

        failure _d:  vital_1760 == 0
  analysis time _t:  surv_year
                id:  pid_20

No late entry detected - p is estimated using the actuarial method
```

start	end	cr_e2	ci_dc	ci_do
0	1	0.4698	0.5125	0.0490
1	2	0.3141	0.6530	0.0735
2	3	0.2468	0.7096	0.0906
3	4	0.2077	0.7400	0.1043
4	5	0.1811	0.7592	0.1159
5	6	0.1618	0.7719	0.1259
6	7	0.1456	0.7817	0.1348
7	8	0.1311	0.7896	0.1426
8	9	0.1220	0.7940	0.1496
9	10	0.1114	0.7986	0.1559

In the output above, 1 minus cr_e2 is the net probability of death due to lung cancer. The columns ci_dc and ci_do are the crude probabilities of death (cumulative incidence) due to lung cancer and due to other causes, respectively. At 10-year follow-up, `strs` estimated that 80% will have died of lung cancer, 16% will have died of causes other than lung cancer, and 4% will be alive. In the hypothetical net survival scenario where patients can die only of lung cancer, `strs` estimated that 89% of patients will have died of lung cancer and 11% will not have died of lung cancer within 10 years.

2.2 "Model": Estimating net survival

2.2.1 The cause-specific framework

As explained in Dickman and Coviello (2015), net (and crude) survival can be estimated in either the cause-specific or the relative framework. The cause-specific framework has censored survival times of those who die of other causes than cancer, and standard estimates apply. To be able to focus on the statistical problem, the example remains lung cancer.

After you have `stset` your data, there are five things you should do: *1. Look at*

the stset output. Originally, I had the message "17 multiple records at same instant PROBABLE ERROR" "(surv_mon_1787[_n-1]==surv_mon_1787)" The reason was I mistakenly read sequence number as having only one character. The first piece of the output is a complete accounting for the records in the data.

2. List some of your data. `stset` does not change any existing data. All it does is define the new variables `_t0`, `_t`, `_d`, and `_st`. `_t0` and `_t` record the time span in analysis-time units. `_d` records the outcome at the end of each time span. `_st` records whether the observation is relevant to the current analysis. Often you have cleaned the data to the point where all observations should be relevant to the analysis, like here. Then, there is no need to list data.

3. Type ***stdescribe*** to describe the dataset. Everyone entered at time 0; there is no delayed entry. The values from `stdescribe` are the same as simply summarizing the time variable with the `summarize` command. In contrast, `stsum` reports summary statistics based on analytical methods that consider censoring, delayed entry, and gaps in history. Therefore, the median survival time is not the same for the commands `stdescribe` and `stsum`.

```
. stdescribe
        failure _d:  vital_1760 == 0
  analysis time _t:  surv_year
                id:  pid_20
```

		per subject			
Category	total	mean	min	median	max
no. of subjects	152539				
no. of records	152539	1	1	1	1
(first) entry time		0	0	0	0
(final) exit time		2.014856	.0833333	.8333333	14.91667
subjects with gap	0				
time on gap if gap	0	.	.	.	.
time at risk	307344.17	2.014856	.0833333	.8333333	14.91667
failures	127205	.8339179	0	1	1

```
. stsum
        failure _d:  vital_1760 == 0
  analysis time _t:  surv_year
                id:  pid_20
```

		incidence	no. of	Survival time		
	time at risk	rate	subjects	25%	50%	75%
total	307344.1743	.4138845	152539	.3333333	.9166667	2.916667

4. Type ***stvary*** if you have multiple-record (meaning multiple records per subject) data. I do not have this, according to `stdescribe`.

5. Fix any problems in step 4, perhaps using ***stfill*** *and* ***streset****.* Not needed. The command `sts generate` will create variables containing the Kaplan-Meier or Nelson-Aalen estimates, depending on which you request. The commands `sts list` and `sts graph` will list and graph respectively the survival, hazard, or cumulative hazard function.

```
. sts list, at(15)
        failure _d:  vital_1760 == 0
  analysis time _t:  surv_year
                id:  pid_20
```

Time	Beg. Total	Fail	Survivor Function	Std. Error	[95% Conf.	Int.]
0	0	0	1.0000	.	.	.
1	76086	76095	0.4943	0.0013	0.4918	0.4969
2	45583	24516	0.3227	0.0012	0.3203	0.3251
3	32427	9855	0.2486	0.0011	0.2463	0.2508
4	24551	5282	0.2057	0.0011	0.2035	0.2078
5	19110	3309	0.1762	0.0010	0.1742	0.1783
6	15063	2250	0.1541	0.0010	0.1521	0.1561
7	11801	1654	0.1360	0.0010	0.1340	0.1379
8	9157	1285	0.1200	0.0010	0.1181	0.1219
9	6958	899	0.1071	0.0010	0.1052	0.1090
10	5150	705	0.0951	0.0010	0.0933	0.0970
11	3612	559	0.0835	0.0010	0.0816	0.0853
12	2458	360	0.0740	0.0010	0.0721	0.0759
13	1482	226	0.0656	0.0010	0.0636	0.0676
14	686	144	0.0571	0.0011	0.0549	0.0593
15	57	66	.	.	.	.

```
Note: Survivor function is calculated over full data and evaluated at indicated
      times; it is not calculated from aggregates shown at left.
. sts graph, risktable(0(5)15) ci noorigin
        failure _d:  vital_1760 == 0
  analysis time _t:  surv_year
                id:  pid_20
. graph export survival_function.eps, as(eps) replace
(file survival_function.eps written in EPS format)
```

(*Continued on next page*)

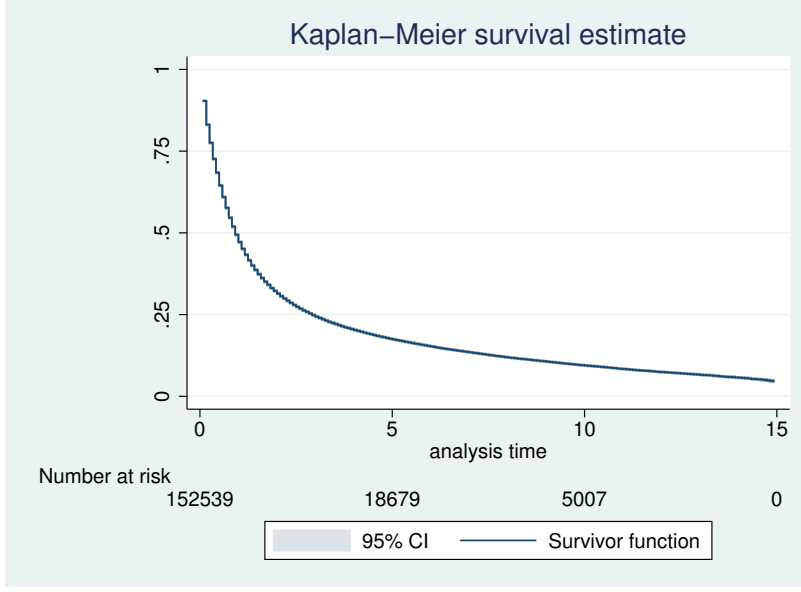

Figure 2.1: Kaplan-Meier survival function

Semiparametric survival results: Stata's `stcox` command fits the semiparametric Cox proportional hazards models. The command is flexible. For example, `stcox` can handle fractional polynomials, and it handles ties in four different ways. You type `stcox` followed by the independent variables. You can specify the `nohr` option for no hazard ratios if you need to compare results reported as coefficients. However, hazard ratios are easier to interpret. In the example model, a male lung cancer patient faces a 24% larger hazard rate than a female lung cancer patient.

```
. stcox male
        failure _d:  vital_1760 == 0
  analysis time _t:  surv_year
                id:  pid_20

Iteration 0:   log likelihood = -1421254.8
Iteration 1:   log likelihood = -1420513.4
Iteration 2:   log likelihood = -1420513.4
Refining estimates:
Iteration 0:   log likelihood = -1420513.4

Cox regression -- Breslow method for ties

No. of subjects =      152,539                 Number of obs    =     152,539
No. of failures =      127,205
Time at risk    =  307344.1743

                                               LR chi2(1)       =     1482.78
Log likelihood  =   -1420513.4                 Prob > chi2      =     0.0000

------------------------------------------------------------------------------
          _t | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        male |    1.24211   .0070129    38.40   0.000      1.22844    1.255931
------------------------------------------------------------------------------
```

A first test of the proportional-hazards assumption is the linktest. If our model really is specified correctly, then the linear predicted value `_hat` should be statistically significant, and the linear predicted value squared `_hatsq` would have no explanatory power. This is what `linktest` does:

```
. linktest
        failure _d:  vital_1760 == 0
  analysis time _t:  surv_year
                id:  pid_20

note: _hatsq omitted because of collinearity
Iteration 0:   log likelihood = -1421254.8
Iteration 1:   log likelihood = -1420513.4
Iteration 2:   log likelihood = -1420513.4
Refining estimates:
Iteration 0:   log likelihood = -1420513.4

Cox regression -- Breslow method for ties

No. of subjects =      152,539                 Number of obs    =     152,539
No. of failures =      127,205
Time at risk    =  307344.1743
                                               LR chi2(1)       =     1482.78
Log likelihood  =   -1420513.4                 Prob > chi2      =     0.0000

------------------------------------------------------------------------------
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        _hat |          1   .026041    38.40   0.000     .9489606    1.051039
      _hatsq |          0  (omitted)
------------------------------------------------------------------------------
```

The prediction squared is omitted from the output because of collinearity. I conclude that the prediction squared has explanatory power, so the model is specified wrongly.

Basic parametric models in Stata are fit using the `streg` command. `predict` after `streg` is used to generate a new variable containing predicted values or residuals. Flexible parametric models can be fit with the user-written commands `stgenreg` and `stpm2`. For multistate survival analysis, Stata has the user-written module `multistate` which includes the data management commands `msset`, the estimation command `stms`, and the post-estimation command `predictms`. For joint modeling of longitudinal and survival data, Stata has the user-written command `stjm`.

`sts test` tests equality of survivor functions. The user-written command `verswlr` is a versatile weighted log-rank test. The user-written command `strmst2` provides RMST measures as an alternative to hazard ratios.

2.2.2 The relative framework

Relative survival can be used with lung cancer data (Hinchcliffe et al., 2012). Stroup et al. (2014) examined the impact of state-specific life tables on relative survival. They found that differences between relative survival based on US life tables and state life tables were small, and state-based estimates were less reliable than US-based estimates for older populations aged 85+. For observed survival only, the Stata life-table command is `ltable` which treats censored observations as if they were withdrawn halfway through the interval. For estimating net survival using a life-table approach, Stata has two user-written commands: `strs` and `stnet`. `stnet` by Coviello et al. (2015) is slightly

faster because it is optimized for the Pohar-Perme estimator. Below is the estimated net survival in the relative framework using `stnet`. It is crude as opposed to age-standardized, rather than as opposed to net. Crude estimates are only useful as a possible intermediate step towards age-standardized estimates. All the estimates in the appendix are age standardized.

```
. stnet using popmort9913 if inrange(dx_year,1999,2003), ///
>     mergeby(_year sex _age) ///
>     breaks(0(0.083333333)10) diagdate(date_dx) birthdate(dob) listyearly
         failure _d:  vital_1760 == 0
   analysis time _t:  surv_year
                 id:  pid_20
Cumulative net survival according to Pohar Perme, Stare and Estève method.
```

start	end	n	d	cns	locns	upcns	secns
.9167	1	25813	1276	0.5023	0.4978	0.5068	0.0023
1.917	2	15752	0	0.3333	0.3289	0.3376	0.0022
2.917	3	12416	217	0.2670	0.2628	0.2712	0.0021
3.917	4	10233	116	0.2291	0.2250	0.2332	0.0021
4.917	5	8780	0	0.2061	0.2021	0.2102	0.0021
5.917	6	7698	0	0.1880	0.1840	0.1921	0.0021
6.917	7	6843	0	0.1740	0.1699	0.1781	0.0021
7.917	8	6141	0	0.1629	0.1587	0.1671	0.0021
8.917	9	5587	45	0.1553	0.1510	0.1597	0.0022
9.917	10	5041	34	0.1494	0.1449	0.1539	0.0023

It was a hassle to use the SAS macro for creating survival time in months. If one has recorded time t in completed years, then t+0.5 will approximate the person-time at risk. The main reason for adding 0.5 to all survival times is to avoid the zero survival times being ignored. Below is the estimated net survival using a simpler variable for survival time in months, `surv_mm`, that does not depend on the SAS macro for imputing missing date of diagnosis or date of last contact. The variable `surv_mm` has an average of 18.1 instead of 21.3 or about 3 months less than when the SAS macro is used. The `stnet` estimates are the same if survival times instead are grouped in days as in `exit, orig(dx)`; this is not shown here but see Coviello et al. (2015, 180) for an example.

```
. gen exit = date(date_lc_1750,"YMD")  //
(5 missing values generated)
. drop if mi(exit)
(5 observations deleted)
. gen surv_mm = floor((exit-date_dx)/365.24*12)+.5 // assumes record_order
. su surv_mon_1787 surv_mm
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
surv_mo~1787 |    173,570    21.24467    32.21551          0        179
     surv_mm |    173,570    18.15616    27.20197         .5      215.5
.
. stset surv_mm, failure(vital_1760==0) id(pid_20) scale(12)  // stset surv_mm
                id:  pid_20
     failure event:  vital_1760 == 0
obs. time interval:  (surv_mm[_n-1], surv_mm]
 exit on or before:  failure
```

```
    t for analysis:  time/12
-------------------------------------------------------------------------------
     173570  total observations
          0  exclusions
-------------------------------------------------------------------------------
     173570  observations remaining, representing
     173570  subjects
     147828  failures in single-failure-per-subject data
 262613.667  total analysis time at risk and under observation
                                             at risk from t =         0
                                  earliest observed entry t =         0
                                       last observed exit t =  17.95833
```

```
. stnet using popmort9913 if inrange(dx_year,1999,2003), ///
>     mergeby(_year sex _age) breaks(0(0.08333)10)  ///
>     diagdate(date_dx) birthdate(dob) listyearly

        failure _d:  vital_1760 == 0
  analysis time _t:  surv_mm/12
                id:  pid_20

Cumulative net survival according to Pohar Perme, Stare and Estève method.
```

start	end	n	d	cns	locns	upcns	secns
1	1.083	23062	1198	0.4100	0.4058	0.4142	0.0021
2	2.083	14064	411	0.2680	0.2641	0.2719	0.0020
3	3.083	10377	264	0.2080	0.2044	0.2117	0.0019
4	4.083	8153	133	0.1737	0.1702	0.1773	0.0018
5	5.083	6703	102	0.1507	0.1473	0.1542	0.0018
6	6.083	5509	80	0.1316	0.1282	0.1350	0.0017
7	7.083	4544	54	0.1156	0.1122	0.1190	0.0017
8	8.083	3721	63	0.1004	0.0971	0.1039	0.0017
9	9.083	3016	47	0.0879	0.0845	0.0913	0.0017

Use the `by()` option together with `standstrata()` to produce age-standardized estimates for each sex. surv_mm is still being used.

```
. stnet using popmort9913 if inrange(dx_year,1999,2003) [iw=icss1], ///
>     mergeby(_year sex _age) breaks(0(0.083333333)10)  ///
>     diagdate(date_dx) birthdate(dob) notables ///
>     standstrata(agegr) by(sex) savstand(agestand_sex__NS, replace)

        failure _d:  vital_1760 == 0
  analysis time _t:  surv_mm/12
                id:  pid_20
file agestand_sex__NS.dta saved
```

The following command produces figure 1, which illustrates age-standardized net survival (NS) and 95

```
. use agestand_sex__NS, clear

. twoway (rarea locns upcns end, col(gs10))  ///
>     (line cns end, lc(black) lw(medthick) lp(l)),  ///
>     by(sex, legend(off)) xlabel(0(2)10) xtitle("Years from diagnosis") ///
>     ytitle("Net survival") ylabel(0(0.2)1, format(%2.1f))  ///
>     saving(net_survival.gph, replace)
(file net_survival.gph saved)

. graph export net_survival.eps, replace  // export .gph graph to .eps
(file net_survival.eps written in EPS format)
```

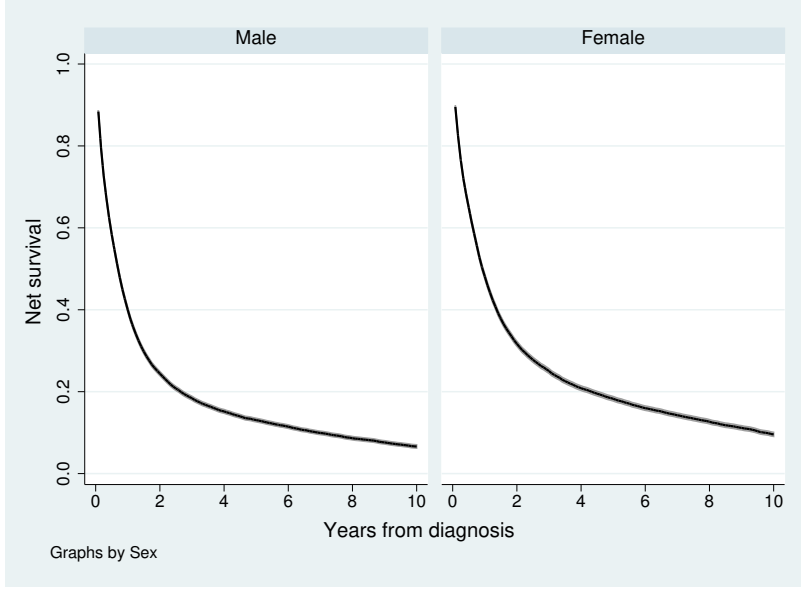

Figure 2.2: Net survival by sex for adult lung cancer cases diagnosed in Florida from 1999-2003

It is possible to do more up-to-date period and hybrid estimation. See Coviello et al. (2015, 181-183). However, again, the cohort approach is recommended when publishing standard and routine life tables, and when making international comparisons (UKIACR, 2016, 9).

2.3 "Visualize": How to construct publication quality tables and graphs

The FCDS annual reports list ten cancer sites or groups: lung and bronchus, prostate, breast, colectoral, bladder, head and neck, non-Hodgkin, melanoma, ovary, and cervix. The main demographic groups are sex (female, male), race (black, white), and age-group (0-14, 15-39, 40-64, and 65+). Another important variable is stage group (localized, regional, distant, unknown).

```
. use doh, clear
(Dataset for Survival Analysis, DX Years 1999-2013)
. stset surv_year, failure(vital_1760==0) id(pid_20)
                id:  pid_20
     failure event:  vital_1760 == 0
obs. time interval:  (surv_year[_n-1], surv_year]
 exit on or before:  failure
------------------------------------------------------------------------------
    1238598  total observations
```

```
     67587  observations end on or before enter()
------------------------------------------------------------------------------
    1171011  observations remaining, representing
    1171011  subjects
     564240  failures in single-failure-per-subject data
  5452100.5  total analysis time at risk and under observation
                                                at risk from t =         0
                                     earliest observed entry t =         0
                                          last observed exit t =  14.91667

. stsum, by(site_10group)
        failure _d:  vital_1760 == 0
  analysis time _t:  surv_year
                id:  pid_20
```

site_1~p	time at risk	incidence rate	no. of subjects	Survival time 25%	50%	75%
Lung & B	307344.1667	.4138845	152539	.3333333	.9166667	2.916667
Prostate	1266102.667	.044479	199017	7.25	13.08333	.
Breast	1085161.75	.0451656	175473	6.583333	14.25	.
Colorect	549577.6666	.1114383	114906	1.833333	6.25	14.41667
Bladder	288064	.0968153	57523	2.583333	7.583333	14.5
Head & N	188948.5833	.125341	44091	1.416667	5.583333	13.58333
Non-Hodg	204868.4167	.1046672	44918	1.583333	7.416667	14.91667
Melanoma	278795.9166	.0513709	48633	5.416667	14.08333	.
Ovary	64375.33334	.1563642	16487	1.25	3.666667	12.25
Cervix	65695.50001	.0730339	12323	2.083333	13.75	.
Other	1153166.5	.1459139	305101	.75	4.083333	14.91667
total	5452100.5	.1034904	1171011	1.5	7.5	.

The life tables in the monograph provided the estimated 1-, 5-, and 10-year standardized net survival rate (%) for four groups: overall, by sex, by race, and by stage. The reporting is similar to the 2007 SEER survival monograph (Ries et al., 2007) and to the 2016 CINA survival monograph (Johnson et al., 2016a). The SEER monograph has more details such as also 2-, 3- and 8–year survival rates but it only reports crude rates (i.e., not age-standardized). The newer CINA monograph reports age-standardized rates but only for 5-year survival. The CINA monograph reports crude rates by age group which, by not being age standardized, are difficult to interpret.[1]

What is not included is as important as what is included. Three types survival data are omitted on purpose. First, crude rates are omitted because standardized rates are more useful for comparison. Second, all cancer groups combined are omitted because there is no standard on how to do this. Third, cancer for children are omitted.

Excellent tables and graphs are nearly always multivariate which requires some more work. There are two possible approaches. One approach, which is not used but can be useful as a last resort, is to create a matrix of statistics using `frmttable`. The other approach, which I use because it is easier, is to create a dataset of the life tables and then use `tabout` (version 3.0.2 beta).

Because there are four separate survival analyses for each of the ten sites, we need to run `stnet` 40 times and combine the 40 life tables into ten datasets. The best solution is to omit output, and repeat over the ten site groups using `forvalues`. To access the

[1]The user-written Stata command `distrate` does not allow age-standardization by age; the output is the same as crude. Personal communication with Enzo Coviello on April 22, 2016.

programming code, please email the author. The ten datasets are then appended into one dataset named `NS.dta`. An appendix to the supplement (Alexandersson, 2017a) provides the monograph tables with added 95% confidence interval.

2.4 "Communicate": Discussion and conclusion

This monograph uses literate programming. The original idea behind literate programming (Knuth, 1984) was to have separate languages for the code ("tangle"), the documentation ("weave"), and the environment. It required learning three languages to write a program: Pascal for the code, TeX for documentation, and WEB for embedding the code. Very few people used literate programming, probably because very few people are willing to learn three languages just to get their program to work. Literate programming concepts have been extended and expanded in the area of reproducible research. The term reproducible research is usually credited to Claerbout (Schwab et al., 2000). Claerbout's framework centered around UNIX makefiles which describe file dependencies. Knuth's embedded-code approach and Claerbout's make-file approach need not be mutually exclusive. Modern implementations of literate programming in data analysis are more interactive and simpler. The main choice for the markup language is between Markdown and LaTeX. The point of Markdown is to achieve most (perhaps 80%?) of what can be done in LaTeX using much simpler syntax, and to not limit the output format to PDF only.

The Stata program used is `texdoc` (Jann, 2016) which is based on the markup language LaTeX and on the Stata program `sjlatex`. Stata has `markdoc` (Haghish, 2016) which is based on Markdown. But it has less features than R Markdown, and is less reliable. The largest feature missing is nice table output. The `markdoc` table output is in plain text rather than in the default Stata Markup Control Language (SMCL). Another useful feature missing in `markdoc` is footnotes. A Stata master do-file "DOH20170630.do" creates the tex files, including a master tex-file "DOH20170630.tex". Any mainstream LaTeX distribution such as MikTeX or TeX Live (PC) or ShareLaTeX (`www.sharelatex.com`) will typeset the PDF with the bibliography from the tex files. MikTex and TeX Live have the advantage of not requiring a file upload before compiling. I recommend ShareLaTex for LaTeX beginners and for LaTeX troubleshooting because it does not require installation, and warning and error messages are more helpful.

In addition to concerns about reproducibility, there are also reporting guidelines. The three most general relevant reporting guidelines for scientific journals and similar technical audiences are CONSORT, STROBE, and RECORD. CONSORT applies to randomized trials, that is, to studies that estimate crude survival in the cause-specific framework. STROBE applies to non-routine observational studies, that is, to studies that estimate net survival in the cause-specific framework or crude survival in the relative framework. RECORD applies to routine observational studies, that is, to studies that estimate net survival in the relative framework. Since the focus in this monograph is on routine reporting, it is worth referencing RECORD (Benchimol et al., 2015). RECORD was created as an extension to the STROBE statement to address reporting items specific to observational studies using routinely collected health data. STROBE

consists of a checklist of 22 items, and RECORD consists of a checklist of 13 related items.

To summarize, the technical report showed how to do survival analysis in Stata using FCDS data in a reproducible data science framework. The focus was on net survival in a relative framework using a life-table version of Pohar Perme estimates. The NAACCR-defined survival variables required a SAS macro, and it resulted in about 3 months longer survival time in an example of lung cancer. The companion monograph (Alexandersson, 2017b) will provide 1-, 5-, and 10-year net survival rates in the relative framework for ten major cancer site groups of adult patients diagnosed in Florida 1999-2003. The net survival rates in Florida are mostly somewhat better than in the U.S. combined.

The main advantage of data science is reproducibility. The main advantage of Stata here is that only Stata has implemented a life-table (actuarial) version of the Pohar Perme estimator of net survival. Life-table estimation of net survival is critical to FCDS because other approaches tend to overestimate survival.

This technical report and the companion monograph have several limitations. Here are ten known limitations, listed in approximate order of importance depending on interest:

1. Did not provide survival rate trends – Cancer trends should be interpreted by examining incidence, mortality, and survival simultaneously over the past several years. Therefore, for example, the Cancer Registry of Norway uses a three-year period window in the annual report "Cancer in Norway" (https://www.kreftregisteret.no/Generelt/Publikasjoner/Cancer-in-Norway/). A seemingly more common approach is to compute an annual percentage change (APC) or to describe survival trends in isolation.

2. No sensitivity analysis of using different life tables – Florida-specific tables are available in NAACCR's "CiNA Deluxe Analytic File". Angela Mariotto at NCI is working on county-SES life tables.

3. Dependency on SAS macro

4. Excluded children

5. Did not allow for multiple primary cancers per patient – Two dominant rules for multiple primary cancers are SEER and IARC/IACR. SEER MP rules are the standard in the U.S.

6. Did not create more updated estimates – This can be done with a period, rather than a cohort approach.

7. Did not do flexible parametric modeling – It is possible to create a yearly report with an in-depth 5-year survival analysis for a cancer of interest. Additional useful measures are loss of life expectancy (average survival time) and, for quality control, conditional net survival. Consider cause-specific survival for cancers with screening.

8. Did not discuss reporting guidelines in detail

9. Did not compare results with other than most recent CINA survival – Need to be more comparative whether with other states or with the U.S.

10. Did not compare alternatives to Stata beyond the supplement

From the strengths and limitations, all three main stakeholders can learn a lesson.

- **DOH:** DOH should allow FCDS to release complete birth dates or at least birth months.
- **FCDS:** FCDS should continue to automate data requests and to improve routine reporting. FCDS needs continuing education. The companion monograph showed that the life-table estimation of net survival in a relative framework is useful for routine reporting.
- **Data requestors:** Data requests must carefully balance the need for getting all needed data for survival analysis against overreach.

There are no books yet on population-based cancer survival analysis but, according to Paul Dickman, a book is expected in 2018 (Dickman et al., 2018). Though this technical report added a lot of details to the companion monograph, the main advantage is more reproducible research. Hopefully, this technical report will be helpful not only for DOH and FCDS but also for requestors wanting to do survival analysis of FCDS. Feedback on FCDS publications, interactive statistics, and on data request procedures is always welcomed. For questions about the monograph and this technical report, please contact the author.

Chapter 3

Acknowledgments

The Florida cancer incidence data used in this report were collected by the Florida Cancer Data System (FCDS), the statewide cancer registry funded by the Florida Department of Health (DOH) and the Centers for Disease Control and Prevention's National Program of Cancer Registries (CDC-NPCR). The views expressed herein are solely those of the author and do not necessarily reflect those of the DOH or CDC-NPCR.

This work was supported by FDOH (Contract CODJU) and CDC through the NPCR (DP003872-04). The author thanks Tara Hylton from FDOH, Stephen MacKinnon from the MacKinnon Group and the following FCDS people for helpful feedback: David Lee, Gary Levin, Brad Wohler, and Monique Hernandez. The author thanks Florida Vital Statistics, the National Death Index (NDI) and the Human Mortality Database (HMD) for providing the death information. Special thanks to Paul Dickman (Karolinska Institutet, Sweden) for feedback during the NAACCR 2017 workshop in June on how to address the main limitations in a future update.

Appendix A

Sensitivity analysis

A.1 Introduction

It is important to explain the sensitivity of the results or "what if?" questions. Wimberley et al. (2013) showed how to use Stata for thought experiments based on simulations. The key assumption of net survival in the relative framework is appropriate life tables. Schaffar et al. (2017) found that the use of different life tables did not compromise net survival in the relative framework for colorectal, lung, melanoma and breast cancer. By contrast, a relatively small error in cause of death led to a large change in the net survival estimate. Schaffar et al. (2017) used data of 4285 women in the Geneve Cancer Registry. Below, the sensitivity analysis of the net survival results in the monograph instead will use data directly on less critical assumptions and for one cancer site only which is more limiting but simpler. Four examples will be provided for lung cancer data. The first example uses full birth dates instead of birth year to illustrate a small bias of using birth year only. The second example uses `strs` instead of `stnet` to illustrate a small difference of using different life-table formulas. The third example uses survival time without imputed missing values to illustrate the importance of missing survival times. The final example uses incomplete survival months to illustrate the importance of using completed survival months.

A.2 Example 1: What if you use full birth dates?

To enable reproducible results for interested data requestors, the monograph and this technical report have used birth year only, nut full birth dates. However, it would be interesting to have some kind of measure of the impact of not being able to use full birthdates. The first step is to get the variable, which is stored in the PATIENT table, and to restrict the analysis dataset to lung cancer.

```
. odbc load PATIENT_ID PATIENT_DATE_OF_BIRTH, ///
>     user(webuser) password(`password´) dsn(Oracle64) /// connect_options
>     datestring clear  table("PATIENT")
```

```
. rename PATIENT_ID pid_20
. gen dob_orig = date(PATIENT_DATE_OF_BIRTH, "YMD")
(1,140 missing values generated)
. label variable dob_orig "Patient birthdate (date)"
. drop PATIENT_DATE_OF_BIRTH
. format dob_orig %d
. merge 1:m pid_20 using doh, keep(match) nogen
(label sexlab already defined)
(label racelab already defined)
(label stagelab already defined)
    Result                           # of obs.
    -----------------------------------------
    not matched                             0
    matched                         1,238,598
    -----------------------------------------
. keep if site_10group==1  // lung cancer
(1,065,023 observations deleted)
. gen exit = date(date_lc_1750,"YMD")
(5 missing values generated)
. format exit %d
. drop if mi(exit)
(5 observations deleted)
. gen surv_mm = floor((exit-date_dx)/365.24*12)+.5
. save temp, replace
file temp.dta saved
```

A seminal reference in favor of using full dates is (Woods et al., 2012). The article showed a 1-5% bias in 1-year net survival of colorectal, breast, and ovary cancer if survival times are only provided in months, not days. The problem of birth month and birth date not being releasable is much smaller because the survival times are continuous. The problem instead is that the matching to life tables may be off by a year of age. In the monograph, Tables A.1 and A.2 show that the 1-, 5-, and 10-year age-standardized net survival rates for lung cancer were 50.5%, 20.9%, and 15.3%. The only difference when using full birth dates is that the 10-year rate changes by 0.2%, from 15.33% to 15.35%. Restricted birth dates not only result in a small bias but also introduce operational problems, as Woods et al. (2012, E1121) noted, by introducing avoidable complexity into the quality control of the data, a crucial component of robust comparisons.

```
. use temp, clear
. stset surv_year, failure(vital_1760==0) id(pid_20)
                id:  pid_20
     failure event:  vital_1760 == 0
obs. time interval:  (surv_year[_n-1], surv_year]
 exit on or before:  failure
------------------------------------------------------------------------------
     173570  total observations
      21036  observations end on or before enter()
------------------------------------------------------------------------------
     152534  observations remaining, representing
     152534  subjects
     127205  failures in single-failure-per-subject data
   307286.5  total analysis time at risk and under observation
```

```
                                          at risk from t =         0
                                earliest observed entry t =         0
                                     last observed exit t =  14.91667
. tempfile temp1 temp2
. qui stnet using popmort9913 if inrange(dx_year,1999,2003) [iw=icss1], ///
>     mergeby(_year sex _age) breaks(0(0.083333333)10)  ///
>     diagdate(date_dx) birthdate(dob) notables standstrata(agegr)  ///
>     savstand(`temp1´, replace)  // source
. qui stnet using popmort9913 if inrange(dx_year,1999,2003) [iw=icss1], ///
>     mergeby(_year sex _age) breaks(0(0.083333333)10)  ///
>     diagdate(date_dx) birthdate(dob_orig) notables standstrata(agegr)  ///
>     savstand(`temp2´, replace)  // std total
. clear
. append using `temp1´ `temp2´, gen(source)
. list source end cns locns upcns if inlist(end,1,5,10), noobs
```

source	end	cns	locns	upcns
1	1	0.5046	0.4999	0.5093
1	5	0.2090	0.2048	0.2131
1	10	0.1536	0.1490	0.1583
2	1	0.5046	0.4999	0.5092
2	5	0.2089	0.2048	0.2131
2	10	0.1535	0.1489	0.1581

A.3 Example 2: What if you ignore birth dates?

The user-written Stata commands `strs` with the option `pohar` and `stnet` have both implemented the Pohar Perme estimator of net survival. The `strs` command implements two different formulas, actuarial and hazard transformation, which give similar results. The Pohar Perme estimator with the hazard transformation approach, `ht` option, should be identical to the formula in `stnet` (Dickman and Coviello, 2015, 191). However, only `stnet` requires the date of birth (for creating age at diagnosis in years). Instead of the net survival rates 50.5%, 20.9%, and 15.3%, `strs` with the options `pohar ht` estimate the rates 49.1%, 21.3%, and 15.6%. Therefore, restricted birth dates are better than no birth dates for estimating net survival.

```
. use temp, clear
. tempfile temp1 temp2
. qui stset surv_year, failure(vital_1760==0) id(pid_20)
. qui strs using popmort9913 if inrange(dx_year,1999,2003) [iw=icss1], ///
>     mergeby(_year sex _age) diagage(age_dx_230) diagyear(dx_year) ///
>     breaks(0(0.083333333)10)  standstrata(agegr) pohar  ///
>     savstand(`temp1´, replace)  // actuarial as in stns
. qui strs using popmort9913 if inrange(dx_year,1999,2003) [iw=icss1], ///
>     mergeby(_year sex _age) diagage(age_dx_230) diagyear(dx_year) ///
>     breaks(0(0.083333333)10)  standstrata(agegr) pohar  ///
>     savstand(`temp2´, replace) ht // hazard transformation as in stnet
. clear
. append using `temp1´ `temp2´, gen(source)
```

```
. list source end cns_pp lo_cns_pp hi_cns_pp if inlist(end,1,5,10), noobs
```

source	end	cns_pp	lo_cns_p	hi_cns_p
1	1	0.5037	0.4990	0.5084
1	5	0.2075	0.2034	0.2116
1	10	0.1507	0.1463	0.1553
2	1	0.4912	0.4866	0.4959
2	5	0.2130	0.2088	0.2172
2	10	0.1559	0.1513	0.1605

A.4 Example 3: What if you ignore the SAS macro to create survival months?

What happens if you ignore the SAS macro "CalculateSurvivalTimeInMonths.sas" to create survival months? Recall that the formula for `surv_mm` added 0.5 to avoid the zero survival times being ignored, and that `surv_mon_1787` is for *completed* survival months. To get comparable results with `surv_mon_1787`, therefore, values with 0.5 should not be used. The 1-, 5-, and 10-year net survival rates when using `surv_mon_1787` are, as mentioned, 50.5%, 20.9%, and 15.3%. The rates when instead using `surv_mm` are 49.1%, 17.3%, and 8.9%. The SAS macro creates a record order variable, which can somewhat easily be created in Stata. More importantly, the SAS macro standardizes specification of missing months and days for date or diagnosis and date of last contact. To have equivalent or better Stata code would be a welcomed improvement.

```
. use temp, clear
. tempfile temp1
. stset surv_mm if surv_mm!=0.5, failure(vital_1760==0) id(pid_20) scale(12)
                id:  pid_20
     failure event:  vital_1760 == 0
obs. time interval:  (surv_mm[_n-1], surv_mm]
 exit on or before:  failure
    t for analysis:  time/12
           if exp:  surv_mm!=0.5
--------------------------------------------------------------------------------
     173570  total observations
      23057  ignored at outset because of -if <exp>-
--------------------------------------------------------------------------------
     150513  observations remaining, representing
     150513  subjects
     127570  failures in single-failure-per-subject data
 261652.958  total analysis time at risk and under observation
                                             at risk from t =         0
                                  earliest observed entry t =         0
                                       last observed exit t =  17.95833
. qui stnet using popmort9913 if inrange(dx_year,1999,2003) [iw=icss1], ///
>     mergeby(_year sex _age) breaks(0(0.083333333)10)  ///
>     diagdate(date_dx) birthdate(dob) notables standstrata(agegr)  ///
>     savstand(`temp1´, replace) listyearly // source
. use `temp1´, clear
```

```
. list end cns locns upcns if inlist(end,1,5,10), noobs
```

end	cns	locns	upcns
1	0.4905	0.4857	0.4953
5	0.1729	0.1689	0.1768
10	0.0886	0.0849	0.0924

A.5 Example 4: What if you include incomplete survival months?

What happens if you include incomplete survival months? The SAS macro creates completed survival months. You may want to instead include incomplete survival months. The answer is that the survival rates could drop dramatically. The 1-, 5-, and 10-year rates with incomplete survival months drop from 50.5%, 20.9%, and 15.3% to 43.6%, 15.3%, and 7.9%. It does not matter for the life-table approach if survival time is recorded in days or in months (Coviello et al., 2015, 180). The trick to avoid failures at time t=0 if you record survival time in days is that you have to move the days forward a little, like was done for the variable `surv_mm`; the code below uses a smaller adjustment number 0.125 instead of 0.5.[1]

```
. use temp, clear
. tempfile temp1
. stset exit, origin(date_dx) failure(vital_1760==0) id(pid_20) scale(365.24)
                id:  pid_20
     failure event:  vital_1760 == 0
obs. time interval:  (exit[_n-1], exit]
 exit on or before:  failure
    t for analysis:  (time-origin)/365.24
            origin:  time date_dx
------------------------------------------------------------------------------
     173570  total observations
        266  observations end on or before enter()
------------------------------------------------------------------------------
     173304  observations remaining, representing
     173304  subjects
     147828  failures in single-failure-per-subject data
 262445.795  total analysis time at risk and under observation
                                             at risk from t =         0
                                  earliest observed entry t =         0
                                       last observed exit t =  17.91972
. replace exit = exit + 0.125
(173,570 real changes made)
. stset exit, origin(date_dx) failure(vital_1760==0) id(pid_20) scale(365.24)
                id:  pid_20
     failure event:  vital_1760 == 0
```

[1] William Gould, head of Stata, discussed this trick in 2007 at http://www.stata.com/statalist/archive/2007-05/msg00124.html and it was discussed in more detail in 2014 at http://www.statalist.org/forums/forum/general-stata-discussion/general/306276-survival-analysis-failure-at-time-zero.

```
obs. time interval:  (exit[_n-1], exit]
 exit on or before:  failure
    t for analysis:  (time-origin)/365.24
            origin:  time date_dx
------------------------------------------------------------------------------
     173570  total observations
          0  exclusions
------------------------------------------------------------------------------
     173570  observations remaining, representing
     173570  subjects
     147828  failures in single-failure-per-subject data
 262505.197  total analysis time at risk and under observation
                                                at risk from t =         0
                                     earliest observed entry t =         0
                                          last observed exit t =  17.92007

. qui stnet using popmort9913 if inrange(dx_year,1999,2003) [iw=icss1], ///
>     mergeby(_year sex _age) breaks(0(0.083333333)10)  ///
>     diagdate(date_dx) birthdate(dob_orig) notables standstrata(agegr)  ///
>     savstand(`temp1´, replace) listyearly //

. use `temp1´, clear

. list end cns locns upcns if inlist(end,1,5,10), noobs
```

end	cns	locns	upcns
1	0.4357	0.4313	0.4402
5	0.1535	0.1499	0.1570
10	0.0789	0.0757	0.0822

References

Alexandersson, A. 2017a. Supplement to *Survival Analysis of the Florida Cancer Data System: A Data Science Project Using Stata*. FCDS Technical Report Supplement.

———. 2017b. *Cancer Survival in Florida 1999-2003 with 10-year Follow-up*. FCDS Monograph.

Baker, M. 2016. 1,500 scientists lift the lid on reproducibility. *Nature News* 533: 452–454.

Benchimol, E., L. Smeeth, A. Guttmann, K. Harron, D. Moher, and I. Petersen. 2015. The Reporting of studies Conducted using Observational Routinely-collected health Data (record) statement. *PLOS Medicine* 12(10): 1–22. URL http://researchonline.lshtm.ac.uk/2324719/1/pmed.1001885.pdf [Accessed: March 27, 2017].

Cleves, M., W. Gould, and Y. Marchenko. 2016. *An Introduction to Survival Analysis Using Stata*. Rev. 3rd ed. College Station, TX: Stata Press.

Coviello, E., P. Dickman, K. Seppä, and A. Pokhrel. 2015. Estimating net survival using a life-table approach. *Stata Journal* 15(1): 173–185.

Dickman, P., and E. Coviello. 2015. Estimating and modeling relative survival. *Stata Journal* 15(1): 186–215.

Dickman, P., P. Lambert, S. Eloranta, T. Andersson, M. Rutherford, A. Johansson, C. Weibull, S. Hinchcliffe, H. Bower, and M. Crowther. 2016. Statistical methods for population-based cancer survival analysis: Computing notes and exercises. URL http://www.pauldickman.com/survival/labs.pdf [Accessed: January 8, 2017].

Dickman, P., P. Lambert, and T. Hakulinen. 2018. Population-based cancer survival analysis.

Haghish, E. F. 2016. markdoc: Literate programming in Stata. *Stata Journal* 16(4): 964–988.

Hinchcliffe, S., M. Rutherford, M. Crowther, C. Nelson, and P. Lambert. 2012. Should relative survival be used with lung cancer data? *British Journal of Cancer* 106(11): 1854–1859.

Jann, B. 2016. Creating LaTeX documents from within Stata using texdoc. *Stata Journal* 16(2): 245–263.

Johnson, C., H. Weir, A. Mariotto, and D. N. et al., ed. 2016a. *Cancer in North America: 2009-2013. Volume Four: Cancer Survival in the United States and Canada 2006-2012*. Springfield, IL: North American Association of Central Cancer Registries, Inc. (NAACCR). URL https://www.naaccr.org/cancer-in-north-america-cina-volumes/#Vol4 [Accessed: March 14, 2017].

Johnson, C., H. Weir, A. Mariotto, R. Wilson, and D. Nishri. 2016b. Construction of a North American Cancer Survival Index to Measure Progress of Cancer Control Efforts. Presentation at NAACCR 2016 Annual Conference. URL http://www.naaccr.org/wp-content/uploads/2016/11/johnson.pdf [Accessed: March 16, 2017].

Knuth, D. 1984. Literate programming. *Computer Journal* 27: 97–111.

Li, G., T. Sajobi, B. Menon, and L. K. et al. 2016. Registry-based randomized controlled trials- what are the advantages, challenges, and areas for future research? *Journal of Clinical Epidemiology* 80: 16–24.

Perme, M. P., J. Estève, and B. Rachet. 2016. Analysing population-based cancer survival - settling the controversies. *BMC Cancer* 16(933): 1–8.

Perme, M. P., M. Stare, and J. Estève. 2012. On estimation in relative survival. *Biometrics* 68(1): 113–120.

Ries, L., J. Young, G. Keel, M. Eisner, Y. Lin, and M.-J. Horner, ed. 2007. *SEER Survival Monograph: Cancer Survival Among Adults: U.S. SEER Program, 1988-2001, Patient and Tumor Characteristics*. Bethesda, MD: National Cancer Institute, SEER Program, NIH. NIH Pub. No. 07-6215. URL https://seer.cancer.gov/archive/publications/survival/seer_survival_mono_highres.pdf [Accessed: March 20, 2017].

Rodríguez, G. 2017. Literate data analysis with Stata and Markdown. *Stata Journal* 17(3): 600–618.

Schaffar, R., B. Rachet, A. Belot, and L. M. Woods. 2017. Estimation of net survival for cancer patients: Relative survival setting more robust to some assumption violations than cause-specific setting, a sensitivity analysis on empirical data. *European Journal of Cancer* 72: 78–83.

Schwab, M., M. Karrenbach, and J. Claerbout. 2000. Making scientific computations reproducible. *Computing in Science & Engineering* 2(6): 61–67.

Seppä, K., T. Hakulinen, and A. Pokhrel. 2015. Choosing the net survival method for cancer survival estimation. *European Journal of Cancer* 51(9): 1123–1129.

Stroup, A., H. Cho, S. Scoppa, H. Weir, and A. Mariotto. 2014. The Impact of State-Specific Life Tables on Relative Survival. *Journal of National Cancer Institute Monograph* 2014(49): 218–227.

UKIACR. 2016. Standard Operting Procedure: Guidelines on population based cancer survival analysis. Word document "Cancer Survival SOP v11_0.docx" at URL http://ukiacr.org/sites/ukiacr/files/file-uploads/publication/ [Accessed: January 13, 2017].

Wickham, H., and G. Grolemund. 2016. *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. 1st ed. Sebastopol, CA: O'Reilly Media.

Wimberley, T., E. Parner, and H. Støvring. 2013. Stata as a numerical tool for scientific thought experiments: A tutorial with worked examples. *Stata Journal* 13(1): 3–20.

Woods, L., B. Rachet, L. Ellis, and M. Coleman. 2012. Full dates (day, month, year) should be used in population-based cancer survival studies. *International Journal of Cancer* 131: E1120–E1124.

Yule, G. U. 1934. On some points relating to vital statistics, more especially statistics of occupational mortality. *Journal of the Royal Statistical Society* 97(1): 1–84.

About the author

Anders Alexandersson is a Senior Research Associate in the statistical unit at the Florida Cancer Data System (FCDS) in Miami, Florida. He is a long-time Stata user and the author of the user-written Stata command `ellip` for graphing confidence ellipses.