

Supplement to *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 supplement, because of space constraints, was not included in [Alexandersson \(2017\)](#). The supplement provides a subset of Stata-equivalent R code using `Rcall` (GitHub). An appendix has advice for users of Python, SEER*Stat, and SAS. A second appendix contains monograph tables with added 95% confidence intervals.

Keywords: Stata, R, Rcall, net survival, relative survival, data science.

1 Introduction

This supplement, because of space constraints, was not included in [Alexandersson \(2017\)](#). The supplement provides a subset of Stata-equivalent R code. The supplement uses the user-written Stata command `Rcall` (GitHub), version 1.4.2, for calling R. `Rcall` has more features than the user-written command `rsource` (SSC).

The supplement uses Stata 14.2, R 3.3.2, and SAS 9.3. To update R on Windows, type `install.packages("installr")` followed by `installr::updateR()`. To load the package `installr`, type `library(installr)`. An extremely popular integrated development environment (IDE) for R is RStudio.

Run-time tests can be performed in R using the `assertive` package. Development-time testing can be done in R using the `testthat` package. Because [Alexandersson \(2017\)](#) focuses on Stata, I will call R from Stata rather than run R directly or call Stata from R.

2 The problem of survival analysis

The problem of survival analysis is the same regardless of software. [Alexandersson \(2017\)](#) used the survival analysis typology in [Dickman and Coviello \(2015\)](#). The focus was on net survival in a relative framework as in [Coviello et al. \(2015\)](#).

3 Data management using R

3.1 “Import”: How to access the FCDS data

R has three popular ways of accessing an Oracle database. RODBC is an R package for ODBC, which is the technology that Stata uses. The R package RJDBC is an implementation of JDBC. The R package ROracle is an implementation using an Oracle driver. In general, ROracle is the fastest and ODBC is the slowest. Although there are standards for SQL, all the major DBMSs have their own dialects. Below, I use RODBC. The function `odbcConnect()` opens the connection to the ODBC database. In `odbcConnect()`, I started out the password to reduce any potential concern by readers about FCDS security. The function `sqlQuery()` submits an SQL query to the ODBC database, and retrieves the results. In the query, a problem is that R interprets the whitespace as a newline. My solution creates a SQL statement using the function `paste()`. The query works for R but not for Stata’s Rcall. The Rcall problem with ODBC is ignored since there is no need for it in practice. The practical use with the SQL query below is to illustrate how FCDS can use R with ODBC for SQL queries.

```
library(RODBC)
ch <- odbcConnect("Oracle64", uid = "webuser", pwd = "*****")
options(max.print=20) # reduce print on screen
sql_statement <- paste("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,
  SEER_Summary_Stage_1977_N760,
  Derived_SS2000_Flag_N3050,
  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
  (Date_of_Diagnosis_N390 between '19990101' and '20131231') and
  Derived_SS1977_Flag_N3040 in ('1','2') and
  Derived_SS2000_Flag_N3050 in ('1','2') and
  FCDS_Site_Group_N2220 = 36 and
  EXISTS (SELECT abshist_patient_id FROM ABSHIST
  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')))"
temp_odbc <- sqlQuery(channel, sql_statement)
close(channel)
```

3.2 “Tidy”: How to clean the FCDS data

To create log files similar to Stata using R, you can type `install.packages("TeachingDemos")` followed by `txtStart("mylogfile.txt")`, the R code, and `txtStop`. In R, however, literate programming is more common than such log files. The `rmarkdown` package is excellent for literate programming. To get started with a “Hello World” example, simply click the menu `File -> New File -> R Markdown` in RStudio. Use the new `bookdown` package to write relatively complicated documents such as books, long-form articles, or reports. To call SAS from R, use `system("sas -SYSIN path/to/sasscript.sas")`. To read and write SEER*Stat export files, use the `SEER2R` package.

In R, the potential for errors is greater than in Stata because R is much more flexible. For example, the data table is very efficient. It can create new variables by reference using the `:=` operator. You can select variables using “\$ notation” (e.g, `temp$date_dx_390`) but it is more typing and it has no equivalent in selecting observations.

Another example, `markdoc` requires Unicode UTF-8 character encoding because the underlying program `pandoc` requires it. This is a problem with the package `tibble` which is for data frames in `dplyr` style. For example, the `tibble` for `read_csv()` outputs a multiplication sign that is invalid UTF-8 (this has been reported as bug 216). `read_csv()` is faster than `read.csv()` in Base R. The fastest method is `fread()` in the library `data.table`. Data tables is an inheritance of data frames.

The first R program using `Rcall`, `rprog1` below, reads the import-delimited text data from [Alexandersson \(2017\)](#), and then it runs the SAS macro described in [Alexandersson \(2017\)](#).

```
. capture program drop rprog1
. program define rprog1
  1. syntax
  2. Rcall vanilla: ///
> setwd("F:/DOH"); /// cd
> library(data.table); /// load package for fread()
> library(testthat); /// load package for expect_equal()
> library(gdata); /// load package for write.fwf()
> dohDT <- fread("2017_DOH Dataset.txt"); /// import delimited
> doh <- as.data.frame(dohDT[, c("tmp1","tmp2","tmp3","tmp4","tmp5","tmp6", ///
>   "tmp7","tmp8","tmp9","tmp10","tmp11","tmp12","tmp13") := list(c(""))]); ///
> expect_equal(anyDuplicated(doh[, c("Patient_Id_Number_N20", /// isid
>   "Sequence_Number_Central_N380")]), 0); ///
> setcolorder(doh ,c("tmp1", "Patient_Id_Number_N20", /// order 41,8
>   "tmp2", "Addr_At_Dx_State_N80", "tmp3", "County_At_Dx_N90", ///
>   "tmp4", "Race_1_N160", "tmp5", "Sex_N220", "Age_At_Diagnosis_N230", ///
>   "Birth_Year_N240", "tmp6", "Sequence_Number_Central_N380", ///
>   "Date_Of_Diagnosis_N390", "tmp7", "Type_Of_Reporting_Source_N500", ///
>   "tmp8", "Seer_Summary_Stage_2000_N759", "Seer_Summary_Stage_1977_N760", ///
>   "tmp9", "Derived_Ss2000_Flag_N3050", "tmp10", ///
>   "Date_Of_Last_Contact_N1750", "tmp11", "Vital_Status_N1760", "tmp12", ///
>   "Fcds_Site_Group_N2220", "tmp13")); ///
> write.fwf(doh, file="input.txt", sep = "", /// outfixt
>   width = c(41,8, 95,2, 9,3, 18,2, 13,1, 3,4, 328,2, 8,25, ///
>   1,340, 1,1,254, 1,955, 8,2, 1,213, 2,998), formatInfo = TRUE); ///
> system2("C:/SAS/x86/SASFoundation/9.3/sas.exe", /// saswrapper
```

```

>   args = c("-SYSIN", "CalculateSurvivalTimeInMonths.sas"); ///
>
>   3. end
. rprog1

```

It is often useful to test the R program in R before running it in Stata with `Rcall`. `Rcall` does not provide the R console output. This is an issue with package `tibble` and with `var_label()` in package `labelled`. Perhaps the most fundamental feature missing from base R is support for variable labels. In R, `Hmisc`, `memisc`, `prettyR` and `sjmisc` are a few packages that support variable labels. One solution to create variable labels is to use the package `labelled`. Another solution is to create variable labels as attributes of the data frame rather than of the columns. You can then create a data dictionary from that. I use both methods. You would usually select variables by name and observations by logic. The `subset` function lets us use that combination easily. The program below, `rprog2`, reads in the fixed-width dataset from SAS, modifies the data frame as was done in Stata, and then it saves the data frame.

```

. capture program drop rprog2
. program define rprog2
1. syntax
2. Rcall vanilla: ///
> library(readr); /// load package for read_fwf()
> library(labelled); /// load package for variable labels
> library(foreign); /// load package for write.dta and read.dta
> library(data.table); ///
> library(testthat); /// load package for expect_equal
> library(car); /// load package for recode() -- or use nested ifelse()
> temp <- read_fwf("myoutputfile.txt", fwf_positions( /// file
>   c(42, 145, 156, 177, 192, 193, 196, 528, 530, 563, /// start
>     904, 905, 1160, 2116, 2126, 2305, 2313, 2314, 2318, 2340, 2510), ///
>   c(49, 146, 158, 178, 192, 195, 199, 529, 537, 563, /// end
>     904, 905, 1160, 2123, 2126, 2312, 2313, 2317, 2325, 2341, 2511), ///
>   c("Patient_Id_Number_N20", "Addr_at_DX_State_N80", /// col_names
>     "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", ///
>     "Derived_SS2000_Flag_N3050", ///
>     "Date_of_Last_Contact_N1750", "Vital_Status_N1760", "date_lc_1785", ///
>     "surv_flag_1786", "surv_mon_1787", "date_dx_1788", ///
>     "FCDS_Site_Group_N2220", "record_order")), ///
>   col_types = "iciiiiiciiiiiiiiiii", na = c("", "NA")); /// col_types
> temp <- as.data.frame(temp); ///
> summary(temp); ///
> temp <- set_variable_labels(temp, /// label variable
>   Patient_Id_Number_N20 = "Patient_Id_Number_N20", ///
>   Addr_at_DX_State_N80 = "Addr_at_DX_State_N80", ///
>   County_at_DX_N90 = "County_at_DX_N90", ///
>   Race_1_N160 = "Race_1_N160", ///
>   Age_at_Diagnosis_N230 = "Age_at_Diagnosis_N230", ///
>   Birth_Year_N240 = "Birth_Year_N240", ///
>   Sequence_Number_Central_N380 = "Sequence_Number_Central_N380"); ///
> var_label(temp); /// (only shown in R console)
> var_label(temp$Patient_Id_Number_N20); /// (only shown in R console)
> write.dta(temp, tempfile <- tempfile()); ///
> a <- read.dta(tempfile); ///
> var.labels <- attr(a, "var.labels"); ///
> data.key <- data.frame(var.name=names(a),var.labels); /// data dictionary

```

```

> class(temp); ///
> class(data.key); ///
> temp <- data.table(temp); ///
> setnames(temp, ///
>   c("Patient_Id_Number_N20", "Addr_at_DX_State_N80", /// rename
>     "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"), ///
>   c("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")); ///
> expect_equal(anyDuplicated(temp[, c("pid_20", "record_order")]), 0); /// isid
> temp[, c("dx_year") := as.numeric(substr(temp$date_dx_390,1,4))]; /// dx_year
> temp[, c("male") := temp$sex_220]; /// male
> temp$male <- recode(temp$male, "2=0"); /// 0=Female, 1=Male
> summary(temp); ///
> set.seed(123); ///
> options(max.print=100); ///
> n <- nrow(subset(temp, subset = !is.na(doby_240))); /// if !mi(doby_240)
> temp[, c("dobmr_240") := sample(12, n, replace=T)]; /// gen dobmr_240
> temp <- subset(temp, sex_220==1 | sex_220==2); /// sex exclusion
> temp <- subset(temp, age_dx_230>=0 & age_dx_230<=126); /// age_dx exclusion
> save(temp, file="doh.Rda"); /// save R workspace
>
3. end
. rprog2

```

3.3 “Transform”: How to create the FCDS analysis dataset

The short program below, rprog3, merely displays the variable label for pid_20.

```

. capture program drop rprog3
. program define rprog3
1. syntax
2. Rcall vanilla: ///
> library(labelled); /// load package for variable labels
> load("doh.Rda"); /// load R workspace
> st.load(temp); /// load R data frame in Stata
> var_label(temp); /// di variable labels (fails! only shown in R)
> attr(temp$pid_20, "label"); /// di varlabel for pid_20
>
3. end
. rprog3
[1] "Patient_Id_Number_N20"

```

Verify that the R dataset is the same as the Stata dataset. readstata13 cannot save empty strings. The error message is “Error in save.dta13(temp, file = “_load.data.dta“, version = 11) NAs are not allowed in subscripted assignments.” The easiest fix is to use Stata. Save date_lc_1785 and date_dx_1788 as numeric, and then convert the two numeric variables to string variables:

```

. tostring date_lc_1785, replace

```

```

date_lc_1785 was long now str8
. tostring date_dx_1788, replace
date_dx_1788 was long now str8
. replace date_lc_1785="" if date_lc_1785=="."
(34 real changes made)
. replace date_dx_1788="" if date_dx_1788=="."
(34 real changes made)
. save doh_r, replace
file doh_r.dta saved
.
. use doh_r, clear
(Written by R)
. drop dohmr_240
. cf _all using doh, verbose

```

For relative survival analysis, we need the population mortality file. The equivalent R code for creating the population mortality file is omitted to enable a focus on the main text and Stata. In R, the population mortality table has to be specified in the argument `ratetable`, as defined in the package `survival`. The package `relsurv` simplifies the process. For an example, see [Pohar and Stare \(2007, 1745–1748\)](#).

4 Survival analysis

4.1 Estimating net survival

The [CRAN Task View of survival analysis](#) gives an overview of available R commands. It describes many more options than here. As explained in [Dickman and Coviello \(2015\)](#), net survival can be estimated in either the cause-specific or relative framework. The cause-specific framework has censored survival times of those who die of other causes than cancer, and standard estimates apply.

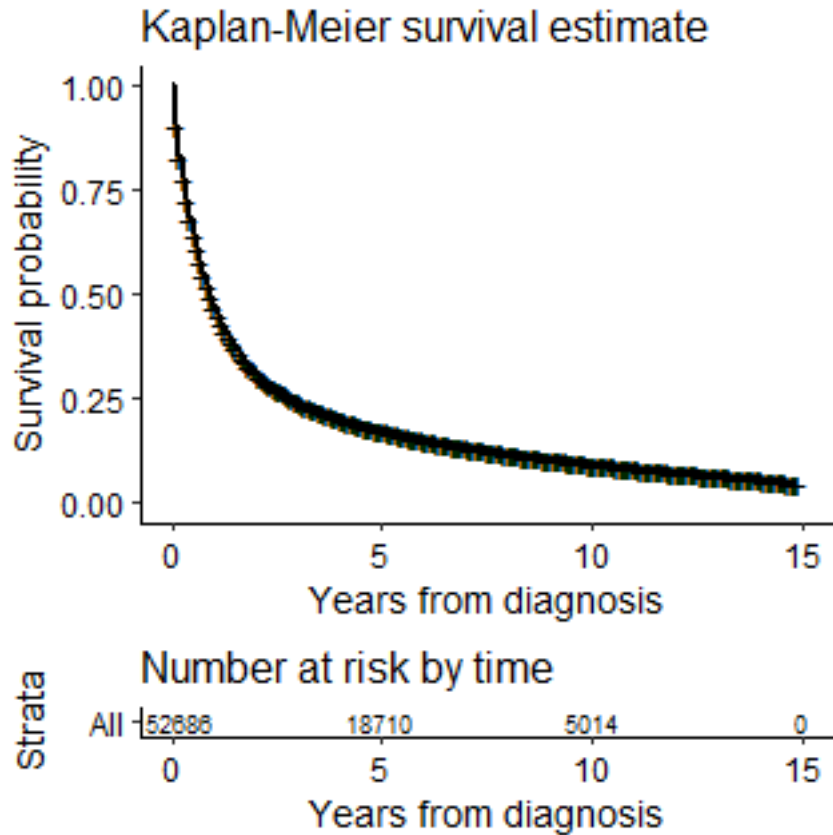
[Moore \(2016, 25-42\)](#) estimates the Kaplan-Meier survivor function in R using the `survfit()` function in the package `survival`, and the `survfit plot()` method which is basic. More advanced graphics features using base R are available in the function `npsurv()` in the package `rms`. However, R's overall best graphics system is `ggplot2`. One reason is that `ggplot2` has themes which give you fine control over the non-data elements of the plot. Two alternatives for graphing the Kaplan-Meier survivor function based on `ggplot2` are the function `ggsurv()` in the package `GGally`, and the function `ggsurvplot()` in the package `survminer`. Compared with Stata, the graph in R still looks basic. To make the graph in R similar to Stata, we would need to use `ggplot2` directly. We would first need to get the data into a suitable data frame and then we can use the Stata theme from the package `gthemes`. For Kaplan-Meier, we could use the package `RcmdrPlugin.KMggplot2`. It is a plug-in for the graphical interface R Commander (`Rcmdr`), which integrates with RStudio. The R output code can be copy-pasted into an R script. The graph would be publication quality but the code would be complicated. The Stata `Rcall` program below, `rprog4`, graphs a Kaplan-Meier survival estimate in R using instead the function `ggsurvplot()` in the package `survminer`. Then it runs the Cox regression model as in Stata. Since the model is the same, the hazard rate of 124% is the same.

```

. capture program drop rprog4
. program define rprog4
  1. syntax
  2. Rcall vanilla: ///
> library(survival); /// load package for survfit()
> library(survminer); /// load package for ggsvurvplot()
> load("doh.Rda"); /// load R workspace
> nrow(temp); ///
> library(ply); ///
> count(temp, `dx_year`); ///
> temp\%fail <- 1 - temp\%vital_1760; /// gen fail
> temp\%surv_year <- temp\%surv_mon_1787 / 12; /// gen surv_year
> temp <- subset(temp, age_dx_230<100); /// survival exclusions as in Stata
> temp <- subset(temp, seq_380==0 | seq_380==1); ///
> temp <- subset(temp, fail==0 | fail==1); ///
> temp <- subset(temp, surv_flag_1786==1); ///
> temp <- subset(temp, surv_mon_1787>0); ///
> fit <- survfit(Surv(surv_year, fail) ~ 1, data = temp); /// was surv_mon_1787
> myplot <- ggsvurvplot(fit, risk.table = TRUE, con.int = TRUE, ///
>   xlim = c(0,15), break.time.by = 5, /// was (0,180) and 60
>   xlab = "Years from diagnosis", main = "Kaplan-Meier survival estimate", ///
>   risk.table.fontsize = 3); /// reduce font size
> png("rplot.png", width=320, height=320); /// open a graphics device
> print(myplot); /// generate the plot
> invisible(dev.off()); /// close the device
> coxph(Surv(surv_year, fail) ~ male, /// Cox PH regression
>   method = c("breslow"), data = temp); ///
>
  3. end
. rprog4
Call:
coxph(formula = Surv(surv_year, fail) ~ male, data = temp, method = c("breslow"))
      coef exp(coef) se(coef)      z      p
male 0.21683  1.24213  0.00564 38.4 <2e-16

Likelihood ratio test=1484 on 1 df, p=0
n= 152686, number of events= 127317

```



Next logical steps for the cause-specific framework are to compare groups of survival times followed by regression analysis, as in [Moore \(2016, 43-72\)](#).

For the relative framework, the packages `rstpm2` and `flexsurv` use model-based estimators whereas the package `relnsurv` use the Pohar Perme estimator.

There is no Pohar Perme estimator in R for discrete survival times. In Stata, you can use `stnet` and `strs`. It is a major strength of Stata for survival analysis of FCDS, as illustrated in [Alexandersson \(2017\)](#).

4.2 Estimating crude survival

As explained in [Dickman and Coviello \(2015\)](#), not only net survival but also crude survival can be estimated in either the cause-specific or the relative framework. In the cause-specific framework, we can use the packages `timereg` and `msstate`; [Moore \(2016, 121-134\)](#) only illustrates `msstate`.

In the relative framework, we can use the `cmp.rel()` function in the package `relnsurv` to estimate crude probabilities of death. The functionality is similar to Stata's `strs` using the `cuminc` function; R creates a graph easier but the R table is less structured and less detailed.

5 Discussion and conclusion

For literate programming in R, I recommend the package `bookdown` (Xie, 2016) which is based on the markup language R Markdown and on the R package `rmarkdown`.

6 Appendix 1. Advice for Python, SEER*Stat, and SAS users

6.1 Advice for Python users

Python is the dominant programming language for data science next to R. Python is more general-purpose and less statistical than R. The Python package `lifelines` implements survival analysis in Python. I am not aware of a Python package that implements relative survival analysis. Therefore, Python users should call Stata or for relative survival analysis. For Stata, you would need to run a Python subprocess or define a Python function. In a posting on Stack Overflow, Roberto Ferrer shows how to run a Stata do-file from Python: <http://stackoverflow.com/questions/21263668/run-stata-do-file-from-python>

The Python package `rpy2` is for calling R from within Python. `PythonTeX` and `Pweave` are two Python packages for literate programming. `PythonTeX` is for LaTeX only, whereas `Pweave` is a multi-language weaver package.

6.2 Advice for SEER*Stat users

You need the SEER*Prep software for creating SEER*Stat datasets. There is no book documentation. The current version of SEER*Stat is 8.3.4 from March 2017. The main problem with SEER*Stat is that it lacks scripting capabilities. SEER*Prep also lacks scripting capabilities. SEER has a webpage with a basic [SEER*Prep example](#). The main advantage of SEER*Prep is that it enables SEER*Stat to ignore most data management features. In general, SEER*Prep is more difficult to use than SEER*Stat.

For export files, SEER*Stat has a checkbox “Generate SAS Code to Read Data”. There is no such checkbox for Stata or R.

SEER*Stat is easy to use for basic survival analysis. There are SEER*Stat [tutorials](#). FCDS maintains an analysis dataset in the SEER*Stat format. In the “Statistic” tab, SEER*Stat has a development version of the Pohar Perma estimator for estimating net survival in a relative survival framework.

6.3 Advice for SAS users

SAS can estimate many survival models. For example, SAS/STAT 14.1 has survival models for interval-censored data: `proc iclifetest` is for nonparametric estimation and `proc icphreg` is for proportional hazards regression models.

Another example, SAS/STAT 14.1 can estimate the competing-risk model of Fine and Gray in `proc phreg`.

The current version of SAS is the third maintenance release of Base SAS 9.4 (9.4M3) and SAS/STAT 14.1 from July 2015. FCDS uses the second maintenance release of Base SAS 9.3 (9.3M2) and SAS/STAT 12.1 from August 2012.

SAS recommends [StatRep](#) for literate programming. StatRep requires SAS 9.2 or later, and Latex as the markup language.

SAS users can call Stata with the macro `stata_wrapper.sas` http://www.cpc.unc.edu/research/tools/data_analysis/sas_to_stata from the UNC Carolina Population Center. SAS users can call R using a `SUBMIT` statement with the `R` option: `submit / r`. To write SEER*Prep files, there is no specific program but see the Notes section for “CalculateSurvivalTimeInMonths.sas”. To create SEER*Stat data, you can use the [NPCR SAS Tool](#) (Dong et al., 2016). To read SEER*Stat export files, check the checkbox “Generate SAS Code to Read Data” in SEER*Stat. The NPCR SAS Tool also computes Ederer II estimates of net survival. The NPCR SAS Tool was made publicly available on June 1, 2017.

Ron Dewar has two two new SAS macros, `rel_surv` and `stpm2` in the syntax files “CPAC macros.sas” and “stpm2 macros.sas”, which mimic the Stata commands `strs` and `stpm2`. The new SAS macros were introduced on June 16 at the NAACCR 2017 annual conference and they are expected to be publicly available on the NAACCR or SEER website in the coming year. The only published implementation is an early version of `stpm2` (Dewar and Khan, 2015). The new SAS macros are available on request either from Anders Alexandersson or from Ron Dewar.

7 Appendix 2. Monograph tables with added 95% confidence intervals

To be consistent with previous FCDS reports, Appendix 2 contains 95% confidence intervals in the tables (to the right of each rate). The confidence interval for a net survival rate is a range of plausible values. Values outside of the confidence interval are relatively implausible.

Sometimes the difference between two estimated net survival rates is of interest. The usual appropriate solution in Stata is to use the command `contrast` but `stnet` and `strs` do not support `contrast`. A direct comparison is difficult without the command `contrast`. A possible workaround is to use the guideline that the difference between two estimates is significant if there is a *large* amount of no overlap between the confidence intervals, otherwise the difference may or may not be significant. As with every guideline, there are problems. The author prefers graphs, not tables, for comparing the significance of the differences.

The 95% confidence intervals can easily be created from the variables `locns` and `upcns` in the Stata dataset `NS.dta`. However, it is difficult to programmatically add the confidence intervals to the tables. `tabout` does not have a fea-

ture to add confidence intervals other than for survey statistics, and `frmtable` would result in slightly different formatting. The author decided to copy-paste the confidence intervals into copied tables. There were 41 copy-pastes since each site table has four groups. As explained in the monograph, sometimes the estimated net survival rates are over 100%. Then, the confidence intervals cannot be computed. In the tables, the confidence intervals are displayed in parentheses and a missing confidence interval is symbolized with the period mark.

7.1 Summary table

Table 1: Net Survival by FCDS Cancer Site Group

FCDS Cancer Site Group	Net survival rate (%)		
	1-year	5-year	10-year
Lung & Bronchus	50.5 (50.0-50.9)	20.9 (20.5-21.3)	15.4 (14.9-15.8)
Prostate	100.6 (.)	101.1 (.)	104.0 (.)
Breast	99.1 (98.9-99.2)	93.3 (92.8-93.7)	90.1 (89.1-90.9)
Colorectal	88.6 (88.3-89.0)	71.1 (70.5-71.7)	66.6 (65.5-67.6)
Bladder	93.7 (93.2-94.1)	83.3 (82.4-84.1)	78.4 (76.9-79.9)
Head & Neck	84.4 (83.7-85.1)	61.0 (59.9-62.2)	51.3 (49.4-53.2)
Non-Hodgkin	82.2 (81.4-82.8)	66.1 (65.1-67.1)	58.1 (56.5-59.7)
Melanoma	97.6 (97.2-97.9)	90.7 (89.9-91.3)	89.0 (87.8-90.2)
Ovary	80.5 (79.3-81.6)	44.6 (43.1-46.1)	35.3 (33.2-37.5)
Cervix	86.4 (85.1-87.6)	65.7 (63.8-67.5)	61.1 (58.6-63.5)

Source: Dataset NS.dta from the FCDS 2017 Monograph. Age-standardized Pohar-Perme life-table estimates from `stnet`. Prostate uses prostate age groups and prostate ICSS 1 weights. Melanoma and Cervix use ICSS 2 weights. The seven other cancer sites use ICSS 1 weights.

7.2 Site-specific tables

Table 2: Net Survival of Lung & Bronchus Cancer

	Net survival rate (%)		
	1-year	5-year	10-year
Age Standardized			
Total	50.5 (50.0-50.9)	20.9 (20.5-21.3)	15.4 (14.9-15.8)
Sex			
Male	46.6 (46.0-47.2)	18.0 (17.5-18.6)	13.3 (12.6-13.9)
Female	55.2 (54.5-55.9)	24.4 (23.8-25.1)	18.0 (17.3-18.7)
Race Group			
White	50.7 (50.2-51.2)	21.1 (20.6-21.5)	15.4 (14.9-15.9)
Black	47.0 (45.2-48.7)	17.6 (16.2-19.2)	15.4 (13.2-17.8)
Stage Group			
Localized	84.1 (83.3-84.9)	57.2 (55.9-58.4)	43.0 (41.5-44.6)
Regional	59.3 (58.4-60.2)	22.1 (21.3-22.9)	14.4 (13.6-15.3)
Distant	28.6 (28.0-29.3)	5.7 (5.4-6.1)	4.9 (4.5-5.3)
Unknown	58.6 (57.3-59.9)	21.2 (20.0-22.4)	15.8 (14.5-17.1)

Source: NS.dta from FCDS 2017 Monograph. Age-standardized Pohar-Perme life-table estimates from `stnet`. Age groups are ICSS standard age groups (15-44, 45-54, 55-64, 65-74, 75-99). Weight is ICSS 1.

Table 3: Net Survival of Prostate Cancer

	Net survival rate (%)		
	1-year	5-year	10-year
Age Standardized			
Total	100.6 (.)	101.1 (.)	104.0 (.)
Sex			
Male	100.6 (.)	101.1 (.)	104.0 (.)
Race Group			
White	100.9 (.)	101.8 (.)	104.4 (.)
Black	98.6 (97.8-99.1)	95.4 (93.3-96.8)	101.3 (.)
Stage Group			
Localized	102.1 (.)	107.3 (.)	113.6 (.)
Regional	100.4 (.)	96.8 (93.3-98.5)	98.9 (0.6-100.0)
Distant	81.1 (79.2-82.8)	39.0 (36.4-41.5)	30.0 (26.5-33.6)
Unknown	99.2 (98.5-99.6)	88.8 (87.3-90.1)	80.1 (77.5-82.4)

Source: NS.dta from FCDS 2017 Monograph. Age-standardized Pohar-Perme life-table estimates from `stnet`. Age groups are ICSS prostate age groups (15-55, 55-64, 65-74, 75-84, 85-99). Weight is ICSS 1, age-adjusted for prostate.

Table 4: Net Survival of Breast Cancer

	Net survival rate (%)		
	1-year	5-year	10-year
Age Standardized			
Total	99.1 (98.9-99.2)	93.3 (92.8-93.7)	90.1 (89.1-90.9)
Sex			
Female	99.1 (98.9-99.2)	93.3 (92.8-93.7)	90.0 (89.1-90.9)
Race Group			
White	99.4 (99.2-99.5)	94.4 (93.9-94.8)	91.4 (90.4-92.3)
Black	95.0 (93.9-95.9)	78.8 (76.6-80.8)	73.4 (69.5-76.9)
Stage Group			
Localized	101.3 (.)	101.8 (.)	101.9 (.)
Regional	98.3 (97.9-98.7)	83.7 (82.6-84.7)	74.9 (73.0-76.6)
Distant	67.9 (65.7-70.0)	27.2 (25.1-29.4)	16.2 (14.0-18.6)
Unknown	96.6 (95.5-97.3)	79.7 (77.5-81.7)	67.6 (63.8-71.2)

Source: NS.dta from FCDS 2017 Monograph. Age-standardized Pohar-Perme life-table estimates from `stnet`. Age groups are ICSS standard age groups (15-44, 45-54, 55-64, 65-74, 75-99). Weight is ICSS 1.

Table 5: Net Survival of Colorectal Cancer

	Net survival rate (%)		
	1-year	5-year	10-year
Age Standardized			
Total	88.6 (88.3-89.0)	71.1 (70.5-71.7)	66.6 (65.5-67.6)
Sex			
Male	88.9 (88.4-89.4)	70.6 (69.7-71.4)	65.3 (63.6-67.0)
Female	88.5 (88.0-88.9)	71.8 (71.0-72.6)	68.1 (66.7-69.3)
Race Group			
White	89.0 (88.7-89.4)	72.2 (71.5-72.8)	67.3 (66.2-68.4)
Black	83.8 (82.4-85.1)	58.3 (56.1-60.4)	57.6 (53.9-61.1)
Stage Group			
Localized	97.9 (97.5-98.2)	94.6 (93.7-95.4)	92.3 (90.3-93.9)
Regional	93.1 (92.6-93.5)	73.5 (72.6-74.5)	66.3 (64.7-67.9)
Distant	54.7 (53.4-55.9)	13.7 (12.8-14.7)	11.6 (10.3-12.9)
Unknown	86.6 (85.5-87.7)	62.9 (61.0-64.8)	57.5 (54.3-60.5)

Source: NS.dta from FCDS 2017 Monograph. Age-standardized Pohar-Perme life-table estimates from `stnet`. Age groups are ICSS standard age groups (15-44, 45-54, 55-64, 65-74, 75-99). Weight is ICSS 1.

Table 6: Net Survival of Bladder Cancer

	Net survival rate (%)		
	1-year	5-year	10-year
Age Standardized			
Total	93.7 (93.2-94.1)	83.3 (82.4-84.1)	78.4 (76.9-79.9)
Sex			
Male	94.5 (94.0-95.0)	84.1 (83.0-85.1)	78.8 (76.9-80.6)
Female	91.2 (90.3-92.1)	81.2 (79.6-82.8)	77.6 (74.9-80.0)
Race Group			
White	94.0 (93.5-94.4)	83.7 (82.8-84.5)	78.7 (77.1-80.2)
Black	81.5 (77.8-84.7)	65.5 (60.0-70.4)	54.5 (45.9-62.4)
Stage Group			
Localized	97.3 (96.9-97.7)	90.1 (89.2-91.0)	85.3 (83.5-86.9)
Regional	74.4 (71.9-76.7)	40.7 (37.6-43.8)	36.9 (32.2-41.6)
Distant	39.1 (34.3-43.9)	14.6 (11.1-18.6)	10.4 (6.9-14.8)
Unknown	93.1 (91.5-94.5)	78.7 (75.4-81.6)	72.0 (66.6-76.7)

Source: NS.dta from FCDS 2017 Monograph. Age-standardized Pohar-Perme life-table estimates from `stnet`. Age groups are ICSS standard age groups (15-44, 45-54, 55-64, 65-74, 75-99). Weight is ICSS 1.

Table 7: Net Survival of Head & Neck Cancer

	Net survival rate (%)		
	1-year	5-year	10-year
Age Standardized			
Total	84.4 (83.7-85.1)	61.0 (59.9-62.2)	51.3 (49.4-53.2)
Sex			
Male	84.6 (83.8-85.5)	61.2 (59.7-62.7)	50.7 (48.1-53.2)
Female	84.2 (82.9-85.4)	61.4 (59.5-63.3)	53.1 (50.1-55.9)
Race Group			
White	85.4 (84.6-86.1)	62.5 (61.3-63.6)	52.4 (50.4-54.4)
Black	72.6 (69.0-75.8)	42.7 (38.2-47.1)	34.5 (28.0-41.0)
Stage Group			
Localized	95.5 (94.7-96.2)	80.1 (78.3-81.7)	70.2 (66.8-73.3)
Regional	78.0 (76.7-79.3)	48.6 (46.8-50.4)	38.6 (35.9-41.3)
Distant	59.2 (56.1-62.1)	30.4 (27.2-33.7)	27.2 (21.5-33.3)
Unknown	83.3 (81.2-85.2)	55.2 (52.0-58.3)	42.9 (38.6-47.1)

Source: NS.dta from FCDS 2017 Monograph. Age-standardized Pohar-Perme life-table estimates from `stnet`. Age groups are ICSS standard age groups (15-44, 45-54, 55-64, 65-74, 75-99). Weight is ICSS 1.

Table 8: Net Survival of Non-Hodgkin Cancer

	Net survival rate (%)		
	1-year	5-year	10-year
Age Standardized			
Total	82.2 (81.4-82.8)	66.1 (65.1-67.1)	58.1 (56.5-59.7)
Sex			
Male	80.2 (79.2-81.2)	62.4 (60.8-63.9)	53.8 (51.3-56.2)
Female	84.4 (83.4-85.3)	70.3 (68.9-71.7)	62.9 (60.7-65.1)
Race Group			
White	82.9 (82.2-83.6)	67.2 (66.1-68.2)	59.2 (57.5-60.9)
Black	74.0 (69.9-77.6)	53.9 (48.7-58.9)	42.8 (35.0-50.4)
Stage Group			
Localized	89.2 (88.0-90.4)	79.8 (77.7-81.7)	74.2 (70.3-77.7)
Regional	83.3 (81.4-85.1)	68.1 (65.2-70.9)	59.6 (55.4-63.5)
Distant	74.1 (72.9-75.3)	54.4 (52.8-56.1)	47.4 (45.2-49.5)
Unknown	87.6 (86.2-88.9)	70.2 (68.0-72.4)	58.3 (54.3-62.0)

Source: NS.dta from FCDS 2017 Monograph. Age-standardized Pohar-Perme life-table estimates from `stnet`. Age groups are ICSS standard age groups (15-44, 45-54, 55-64, 65-74, 75-99). Weight is ICSS 1.

Table 9: Net Survival of Melanoma Cancer

	Net survival rate (%)		
	1-year	5-year	10-year
Age Standardized			
Total	97.6 (97.2-97.9)	90.7 (89.9-91.3)	89.0 (87.8-90.2)
Sex			
Male	96.7 (96.2-97.2)	88.4 (87.3-89.4)	86.2 (84.4-87.7)
Female	98.6 (98.1-99.0)	93.5 (92.4-94.4)	92.8 (91.0-94.2)
Race Group			
White	97.5 (97.2-97.9)	90.1 (89.4-90.8)	87.6 (86.3-88.7)
Black	90.2 (82.3-94.7)	80.8 (68.7-88.6)	73.8 (56.3-85.2)
Stage Group			
Localized	100.6 (.)	98.7 (97.8-99.2)	98.7 (96.5-99.6)
Regional	92.5 (90.6-94.0)	57.7 (54.5-60.9)	48.2 (44.3-52.0)
Distant	47.6 (42.9-52.0)	21.5 (17.7-25.6)	15.5 (12.0-19.4)
Unknown	96.6 (95.3-97.6)	81.1 (78.6-83.4)	75.4 (71.6-78.8)

Source: NS.dta from FCDS 2017 Monograph. Age-standardized Pohar-Perme life-table estimates from `stnet`. Age groups are ICSS standard age groups (15-44, 45-54, 55-64, 65-74, 75-99). Weight is ICSS 2.

Table 10: Net Survival of Ovary Cancer

	Net survival rate (%)		
	1-year	5-year	10-year
Age Standardized			
Total	80.5 (79.3-81.6)	44.6 (43.1-46.1)	35.3 (33.2-37.5)
Sex			
Female	80.5 (79.3-81.6)	44.6 (43.1-46.1)	35.3 (33.2-37.5)
Race Group			
White	81.2 (80.0-82.3)	45.0 (43.4-46.5)	35.5 (33.2-37.7)
Black	68.5 (62.8-73.6)	35.0 (29.4-40.7)	28.9 (22.7-35.5)
Stage Group			
Localized	97.9 (95.0-99.1)	91.9 (86.7-95.2)	91.1 (74.5-97.1)
Regional	87.8 (84.8-90.3)	61.5 (56.7-65.9)	50.7 (44.5-56.6)
Distant	74.4 (72.8-75.9)	29.8 (28.1-31.5)	19.2 (17.4-21.1)
Unknown	84.4 (81.1-87.1)	46.9 (42.3-51.4)	35.6 (28.5-42.7)

Source: NS.dta from FCDS 2017 Monograph. Age-standardized Pohar-Perme life-table estimates from `stnet`. Age groups are ICSS standard age groups (15-44, 45-54, 55-64, 65-74, 75-99). Weight is ICSS 1.

Table 11: Net Survival of Cervix Cancer

	Net survival rate (%)		
	1-year	5-year	10-year
Age Standardized			
Total	86.4 (85.1-87.6)	65.7 (63.8-67.5)	61.1 (58.6-63.5)
Sex			
Female	86.4 (85.1-87.6)	65.7 (63.8-67.5)	61.1 (58.6-63.5)
Race Group			
White	87.2 (85.8-88.5)	66.7 (64.6-68.6)	60.0 (57.4-62.5)
Black	82.1 (78.7-85.0)	60.3 (55.7-64.5)	65.1 (56.7-72.2)
Stage Group			
Localized	97.1 (95.5-98.2)	87.5 (84.7-89.9)	83.8 (79.4-87.3)
Regional	85.2 (83.0-87.2)	54.7 (51.6-57.7)	50.3 (45.9-54.6)
Distant	50.5 (45.4-55.3)	21.3 (17.1-25.8)	20.6 (14.2-27.8)
Unknown	85.7 (81.6-88.9)	59.6 (54.4-64.3)	47.1 (41.9-52.1)

Source: NS.dta from FCDS 2017 Monograph. Age-standardized Pohar-Perme life-table estimates from `stnet`. Age groups are ICSS standard age groups (15-44, 45-54, 55-64, 65-74, 75-99). Weight is ICSS 2.

References

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

- Coviello, E., P. Dickman, K. Seppä, and A. Pokhrel. 2015. Estimating net survival using a life-table approach. *Stata Journal* 15(1): 173–185.
- Dewar, R., and I. Khan. 2015. A new SAS macro for flexible parametric survival modeling: applications to clinical trials and surveillance data. *Clinical Investigation* 5(12): 1–12.
- Dickman, P., and E. Coviello. 2015. Estimating and modeling relative survival. *Stata Journal* 15(1): 186–215.
- Dong, X., K. Zhang, R. Wilson, and M. O’Neil. 2016. Development of National Program of Cancer Registries SAS Tool for Population-Based Cancer Relative Survival Analysis. *Journal of Registry Management* 43(2): 63–73.
- Moore, D. 2016. *Applied Survival Analysis Using R*. Springer.
- Pohar, M., and J. Stare. 2007. Making relative survival analysis relatively easy. *Computers in Biology and Medicine* 37: 1741–1749.
- Xie, Y. 2016. *bookdown: Authoring Books and Technical Documents with R Markdown*. Boca Raton, FL: CRC Press.