

Conducting Survival Analysis in SAS® using Medicare Claims as a Real-world data source

Jay Iyengar, Data Systems Consultants LLC

ABSTRACT

Applications of Survival analysis as a statistical technique extend to longitudinal studies, and other studies in health research. The SAS/STAT® package contains multiple procedures for performing and running survival analysis. The most well-known of these are PROC LIFETEST and PROC PHREG. As a data source, Medicare claims are often used in Real-world evidence studies and observational research. In this paper, survival analysis and the SAS® procedures for performing it will be explored, and survival analyses will be conducted using Medicare claims data sets to assess patient's prognosis amongst Medicare beneficiaries.

INTRODUCTION

Survival analysis is a technique used in clinical and longitudinal studies which concentrates on the time or duration until a specific event occurs. The specific event might be the fatality of a patient, or the readmission of a patient into a healthcare facility for further treatment. In survival analysis, patients are monitored and followed up until they experience the event of interest. However, not every patient will experience the event of interest. Taking account of the patients who don't experience the event of interest is known as censoring. One of the first steps in survival analysis is the estimation of the Survival function. There are several types of survival functions, as well as survival distributions. The survival functions can be estimated using survival curves, which you can produce using one or more of the survival analysis procedures. To perform survival analysis, there are non-parametric and parametric methods. The method used to conduct a survival analysis depends on the SAS/STAT procedure you use.

SURVIVAL ANALYSIS CONCEPTS

In survival analysis, one of the main tasks is to compute a distribution of survival times for a defined population. Survival times are referred to as failure times, and event times are called uncensored survival times. Procedurally, to compute the distribution of survival times, you estimate the distribution, using one of several survival functions.

The survival distribution function, SDF, also known as the survivor function, describes the lifetimes of the population of interest. The SDF is represented in the equation below by $S(t)$

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

The survival distribution function is the probability that an experimental unit from the population will have a lifetime that exceeds t , that is, $Pr(T > t)$. Where T is the lifetime of a randomly selected experimental unit.

Other types of survival functions are the cumulative distribution function, CDF, the probability density function, PDF, and The Hazard function. The equation for the CDF is provided below.

$$F(t) = 1 - S(t)$$

The CDF, which is $F(t)$ is defined as $1 - S(t)$ and is the probability that a lifetime does not exceed t . The PDF, $f(t)$, is defined as the derivative of CDF, or $F(t)$.

The hazard function is another survival function used by SAS/STAT procedures. It is defined as the ratio of the PDF to SDF, that is, $f(t) / S(t)$.

CENSORING

In survival analysis, patients are followed up in a study from the point they experience a specific event, such as being admitted to a hospital, being diagnosed with a specific medical condition, or being discharged from a hospital. However, not every patient in the study will be able to be tracked or monitored. For example, some patients might leave or drop out of the study. Other patients might move and not provide up to date contact information. For these patients, the exact survival time can't be calculated, because this data can't be collected. These individuals are considered censored.

Censored patients have no actual time to event data. For study participants who are censored, the time to event data has to be estimated. There are different methods for doing this. The methods include right-censoring, left-censoring, and interval censoring. Regardless of the method, censored observations cannot be ignored and must be taken into account when performing survival analysis.

The various types of censoring are illustrated in Figure 1 below.

Right-Censored



Left-Censored



Interval Censored



Figure 1. Different types of censoring diagram

The above continuum illustrates and compares the different kinds of censoring. Right-censoring is when the event is unobserved for some individuals because the event date is greater than the event date for individuals with observed events. Left-censoring is when the event date is known to be less than or before the event date. Interval-censoring is where an event falls outside of an interval between 2 dates.

SAS PROCEDURES FOR SURVIVAL ANALYSIS

The primary procedures for conducting survival analysis in SAS/STAT are the LIFETEST procedure and the PHREG procedure, known as PROC LIFETEST and PROC PHREG, respectively. There are also other survival analysis procedures in other SAS modules.

The SAS/STAT module includes 8 different procedures for performing survival analysis, according to version 15.3 of SAS/STAT. The survival analysis procedures are listed and characterized by different attributes.

Besides PROC LIFETEST and PROC PHREG, the survival analysis procedures include variations of these PROCs for interval-censored data, procedures which use parametric estimates on any type of censored data, procedures which use quantile regression, and restricted mean survival time, as well as procedures using the Cox Proportional Hazards model for survey sample data. The SAS/STAT survival analysis procedures are listed in Figure 2.

SAS Procedure	Type of Model	Type of Censoring
LIFETEST	Non-parametric	Right-Censored
PHREG	Semi-Parametric	Right-Censored
ICLIFETEST	Non-parametric	Interval-Censored
ICPHREG	Parametric	Interval-Censored
LIFEREG	Parametric	Left-Censored, Right-Censored, or Interval-Censored
QUANTLIFE	Parametric	Right-Censored
RMSTREG	Parametric	Right-Censored
SURVEYPHREG	Semi-Parametric	Right-Censored

Figure 2. Survival Analysis procedures in SAS/STAT

For each survival analysis procedure listed, the table displays the type of model and type of censoring encompassed. Most of the procedures have a parametric form for the underlying model or function, with the exception of the LIFETEST and ICLIFETEST procedures.

Most of the procedures work with only right-censored data, or data with unobserved failure times. The ICLIFETEST and ICPHREG procedures are versions of LIFETEST and PHREG procedures specifically for interval-censored data. The LIFEREG procedure can be applied to any type of censored data; left, right, or interval.

THE LIFETEST PROCEDURE

With PROC LIFETEST, you perform a non-parametric survival analysis. This means that the procedure doesn't produce estimates of population parameters. PROC LIFETEST uses two different methods to perform survival analysis; the product-limit method, and the life-table method. The product-limit method is known as the Kaplan-Meier method, and the life-table method as the actuarial method, respectively. The life-table estimate is a grouped-data analog of the Kaplan-Meier estimate.

Besides these two methods, PROC LIFETEST can also compute the Breslow estimate, or the Fleming-Harrington estimate. We won't dive into these methods in this paper, as they're outside its scope.

The basic syntax of the LIFETEST Procedure is provided below.

```
Proc Lifetest <Options>;  
    Time Variable / Option>;  
    Strata Variable </ Options>;  
    Test Variable;  
Run;
```

Besides the PROC LIFETEST statement, the primary statements in PROC LIFETEST are the TIME statement and the STRATA statement. The TEST statement is where you include any covariates. Whereas the STRATA statement is for specifying stratification variables. PROC LIFETEST uses right-censoring as the censoring method.

THE PHREG PROCEDURE

The PHREG procedure conducts a survival analysis using the Cox Proportional Hazards Model. PHREG stands for Proportional Hazards Regression.

PROC PHREG is a semiparametric procedure which models survival time using the Hazard function. It's defined as a semiparametric procedure because it assumes a parametric form for the effects of explanatory variables, but it permits an unspecified form for the underlying survival function.

Other distinct features of PROC PHREG are the use of time-dependent explanatory variables, variable selection methods, and methods for handling ties in failure times. PROC PHREG allows the inclusion of time-dependent explanatory variables as covariates. The procedure uses four different variable selection methods; Forward, Backward, Stepwise and Best subset. It also provides four methods for dealing with ties in failure times.

The basic syntax of the PHREG Procedure is provided below.

```
Proc Phreg Options;  
    Class Variable / Options;  
    Model Response*Censor(List) = Effects / Options;  
    Strata Variable (List) / Option;  
Run;
```

Similar to PROC LIFETEST, PROC PHREG involves specifying the censor or status variable and the response variable, in this case, survival time. The MODEL statement is used to specify these variables along with the covariates in a regression-like equation, where the covariates are the effects. PROC PHREG also uses right-censoring as the censoring method, the same as PROC LIFETEST.

THE DATA SOURCE – MEDICARE CLAIMS

For this project, the primary data source is a Medicare claims data set containing healthcare utilization data for Medicare beneficiaries.

The file is an Inpatient Medicare claims data set, specific to an inpatient facility. This claims file contains claims for inpatient hospital visits. The claims file is a header file filetype, which contains summary information about each claim. The file has one record per claim, and multiple records per beneficiary since some beneficiaries have more than one claim.

There are approximately a dozen variables on the file, including diagnosis code, procedure code, admission and discharge dates, claim payment amounts, and other administrative variables.

In Figure 3 below, is a snapshot of the Inpatient Medicare claims data set.

Obs	BENE_ID	CLM_ID	ADMSN_DT	DSCHRGDT	DRG	ICD_DIAG1	ICD_PRCDCD1	ADMTG_DGNS_CD	NPI	PMT_AMT	DED_AMT
1	00013D2EFD8E45D1	196661176988405	20100312	20100313	217	7802		4580	3139083564	4000.00	1100.00
2	00016F745862898F	196261176983265	20100626	20100701	983	3559		5819	6108100173	16000.00	1100.00
3	0007F12A492FD25D	196551177025145	20100522	20100612	950	V5789		V5789	1907446990	14000.00	.
4	0007F12A492FD25D	196831177025774	20100602	20100606	205	49121		49392	5838958805	5000.00	1100.00
5	0007F12A492FD25D	196831176966961	20100616	20100619	456	7356	0073	99641	4959466403	29000.00	1100.00
6	0013E139F1F37264	196141176989106	20100630	20100907	238	4280		873E	3771255921	3000.00	1100.00
7	00196F07C2489342	196751177019091	20100318	20100324	178	41519		41519	314996464E	7000.00	1100.00
8	00196F07C2489342	196991177014269	20100114	20100119	229	42731	4516	7802	7087360414	8000.00	1100.00
9	001EA2F4DB30F105	196131176973569	20100316	20100319	508	99541	0082	99677	7872369581	0.00	1100.00
10	002354398A00234E	196731176991192	20100820	20100831	282	45341	8848	45341	8198589276	10000.00	1100.00
11	00271F7DF9C2B88A	196711177025513	20100113	20100114	854	0389	8345	78060	7774053843	28000.00	1100.00
12	0028A82FE0CA0802	196241177011101	20100514	20100617	880	29533		30350	4199341016	9000.00	1100.00
13	00292D3DBB23CE44	196121176990349	20100710	20100715	683	5849	4516	40351	4903645485	8000.00	.
14	00292D3DBB23CE44	196721177025054	20100703	20100712	251	7802		7802	1499149511	3000.00	1100.00
15	002B6203F086ABA	196871176990954	20100125	20100129	466	71536	8152	71536	4037832287	12000.00	1100.00
16	008AE4429AC2BDE7	196121177011276	20100222	20100302	245	41401	3722	4111	7115813911	10000.00	1100.00
17	008AF4429AC2BDE7	196891176990986	20100121	20100127	188	485		99591	7115813911	7000.00	1100.00
18	0045838358ECA530	196691176959586	20100608	20100612	500	1985	8154	71535	2950647400	12000.00	1100.00
19	0049CB2A111F2225	196081176975386	20100129	20100208	664	5849	3995	78097	6962810697	6000.00	1100.00
20	004B36495271F56	196111176994552	20100708	20100710	285	42731		42731	7319323665	7000.00	1100.00
21	00534E60E989C69E	196141177016557	20100214	20100218	167	7381	8107	7242	3780108997	26000.00	1100.00
22	00534E60E989C69E	196621176976944	20100202	20100204	475	71536	8154	71535	1949223467	12000.00	1100.00
23	00565644D5332C24	196481176979455	20100602	20100611	025	1919	8891	78097	2628405021	4000.00	1100.00
24	005715FA817344A7	196181177011825	20100121	20100123	485	71536	8154	71536	2413038045	13000.00	1100.00
25	00641FD6499D535C	196381177017239	20101212	20101213	867	0389		7802	6452216373	10000.00	1100.00

Figure 3. Medicare Inpatient claims data set

Listed below are the primary variables we're interested in for this project, along with their definitions.

BENE_ID is the Medicare Beneficiary Identifier, which identifies the patient or beneficiary.

ADMSN_DT is Admission date. The date the patient was admitted to the hospital.

DRG is Diagnosis Related Group. It's a diagnosis code which can be used to select patients who received treatment for a specific medical condition.

ICD_DIAG1 is an ICD9 diagnosis code. ICD9 codes are similar to DRG codes, except they're more granular, providing more detail about a condition.

OTHER MEDICARE DATA FILES

Besides the main claims file, the membership file and the diagnosis code file contain useful variables for our analysis. These files can be used as lookup tables to extract additional useful data for Medicare beneficiaries.

The membership file contains beneficiary-level information on Medicare beneficiaries and is known as the Medicare Beneficiary Summary File (MBSF). The MBSF contains demographic variables which can be used as group or subgroup variables in the analysis, and other variables which are essential in survival analysis. The MBSF contains one record per Medicare beneficiary.

The Diagnosis Code file is a reference lookup table for ICD9 Diagnosis Codes. The data set provides descriptions for diagnosis codes. The table is used to focus on specific medical diagnoses in the analysis, such as congestive heart failure, kidney disease and diabetes. This SAS data set contains only two variables; Code and description.

In Figure 4 below is a screenshot of the MBSF (left) with the Diagnosis Code File (right)

Obs	BENE_ID	BENE_DOB	DEATH_DT	RACE	SEX	CNTY_CD	STATE_CD	BENE_HI_CVRAGE_TOT_MONS
1	00013D2EFD8E45D1	19230501		.1	1	950	26	12
2	00016F745862898F	19430101		.1	1	230	39	12
3	0001FDD721E223DC	19360901		.1	2	280	39	12
4	00021CA6FFC3E670	19410601		.5	1	290	06	12
5	00024E3D2352D2D0	19360801		.1	1	590	52	9
6	0002DAE1C81CC70D	19431001		.2	1	400	33	12
7	0002F28CE057345B	19220701		.1	1	270	39	12
8	00030843E3E5B76	19350901		.1	1	680	24	12
9	000345A39D4157C9	19760901		.1	2	810	23	12
10	00036A21B65B0206	19381001		.2	2	570	01	12
11	000489E7EAD463F	19340201		.1	2	140	15	12
12	00048EF1F4791C88	19290601		.1	1	230	44	12
13	0004FCABD505251D	19360701		.1	2	030	41	12
14	00052705243EA128	19340501		.1	1	982	14	12
15	000706637459E497	19360301		.1	2	270	13	12
16	0007E57CC13CE88C	19340101	20101201	.1	1	140	06	12
17	0007F12A492FD25D	19190901		.2	2	400	34	12
18	000A065BA0EED3EA	19191001		.2	2	160	50	12
19	000B4662348C35B4	19420701		.1	2	170	46	12
20	000B97BA2314E971	19380401		.1	1	020	22	12
21	000C7486B11E7030	19320801		.2	1	350	25	12
22	000D6D89463D8A76	19420801		.5	1	000	32	0

Obs	Code	Description
1	001.0	CHOLERA DUE TO VIBRIO CHOLERAE
2	001	CHOLERA
3	001.1	CHOLERA DUE TO VIBRIO CHOLERAE EL TOR
4	001.9	UNSPECIFIED CHOLERA
5	002.0	TYPHOID FEVER
6	002	TYPHOID AND PARATYPHOID FEVERS
7	002.1	PARATYPHOID FEVER A
8	002.2	PARATYPHOID FEVER B
9	002.3	PARATYPHOID FEVER C
10	002.9	UNSPECIFIED PARATYPHOID FEVER
11	003.0	SALMONELLA GASTROENTERITIS
12	003	OTHER SALMONELLA INFECTIONS
13	003.1	SALMONELLA SEPTICEMIA
14	003.2	LOCALIZED SALMONELLA INFECTIONS
15	003.20	UNSPECIFIED LOCALIZED SALMONELLA INFECTION
16	003.21	SALMONELLA MENINGITIS
17	003.22	SALMONELLA PNEUMONIA
18	003.23	SALMONELLA ARTHRITIS
19	003.24	SALMONELLA OSTEOMYELITIS
20	003.29	OTHER LOCALIZED SALMONELLA INFECTIONS
21	003.8	OTHER SPECIFIED SALMONELLA INFECTIONS
22	003.9	UNSPECIFIED SALMONELLA INFECTION
23	004.0	SHIGELLA DYSENTERIAE
24	004	SHIGELLOSIS
25	004.1	SHIGELLA FLEXNERI

Figure 4. Medicare Beneficiary Summary File and Diagnosis Code File

The two variables in the diagnosis code file are Code and Description, where Code is a list of ICD9 diagnosis codes and description is the text description of the code. The key variables from the MBSF are defined below.

BENE_DOB – Beneficiary or Patient’s Date of Birth.

DEATH_DT - Death date of the Medicare Beneficiary. Defines the event of interest allowing us to compute the survival times for uncensored observations.

RACE- Race of the Beneficiary. Demographic variable defining racial category.

SEX- Gender of the Beneficiary. Demographic variable defining Gender.

Other key variables include date of birth, state of residence, number of months of part A coverage, number of months of part B coverage, and flag variables for specific chronic diseases.

DATA MANAGEMENT TASKS PRIOR TO THE ANALYSIS

Prior to running a survival analysis, we need to perform data manipulation in order to create an analysis data set.

We're going to focus on specific medical diagnoses in our analysis, so we need to access the diagnosis code descriptions in our diagnosis code lookup table to subset inpatient claims to specific medical conditions. We also need to retrieve demographic and other variables from our beneficiary summary file to use these variables as group or subgroup variables in the analysis.

We have our three files as SAS data sets; Inpatient claims file, diagnosis code file, and the beneficiary summary file. The first step is to merge or join the Inpatient claims data set with the diagnosis code reference table to obtain the diagnosis code description. The subsequent task is to merge the resulting data set with the beneficiary summary file to obtain demographic and other variables.

```
/* Add DX Description - Join IP Claims File with DXCode Lookup Table */
Data ip2010claim;
    Length DXCD1-DXCD10 $6;

    Set SURVMEDI.ip2010claim;

    Array ICDDX{10} $ ICD_DGNS_CD1-ICD_DGNS_CD10;
    Array DXCode {10} $ DXCD1-DXCD10;

    Do I = 1 to 10;
        DXCode{I} = Substr(ICDDX{I}, 1, 3)||'|'||Substr(ICDDX{I}, 4, 2);
    End;

    Drop ICD_DGNS_CD1-ICD_DGNS_CD10;
Run;

Proc Sql;
    Create Table IPClaim_v2 as
    Select A.*, B.Description
    From ip2010claim as A Left Join SurvMedi.ICD9DX as B
    On A.DXCD1=B.Code;
Quit;

Proc Freq Data=IPClaim_v2 Order=FREQ;
    Tables Description / List Missing;
Run;

/* Join New IP Claim File with Membership File */
Data mbsf_ab_2010;
    Set SurvMedi.mbsf_ab_2010(Rename=(Race=RaceGrp));

    /* Create Race and Gender variables with formatted values */
    Race = Put(RaceGrp, $RaceCat.);
    Gender = Put(Sex, $Gender.);

    If Death_Dt^= . Then
        DeathStatus='Y';
    Else
        DeathStatus='N';

    Drop RaceGrp Sex;
Run;

Proc Sort Data = mbsf_ab_2010 Nodupkey;
    By Bene_ID;
Run;

Proc Sql;
    Create Table Claim_MBSF as
    Select A.*, B.Race, B.Gender, B.State_Cd, B.Cnty_Cd, B.Death_Dt, B.DeathStatus,
        B.Bene_Dob
    From IPClaim_v2 as A, mbsf_ab_2010 as B
    Where A.Bene_ID=B.Bene_ID;
Quit;
```

Figure 5. SAS Code to perform data manipulation tasks

The SAS code used to perform both of these tasks is displayed in Figure 5 above. The SAS logs from running the full program is located in the appendix.

As shown in the code, I used a PROC SQL join to combine the inpatient claims file with the diagnosis code file. In a later step, PROC SORT with the NODUPKEY option is used to delete multiple beneficiary records from the beneficiary summary file (MBSF). Then PROC SQL was used again to merge the claims file with the beneficiary summary SAS data set.

One of the advantages of using the PROC SQL Join is it sorts data implicitly thus avoiding the need to sort explicitly with PROC SORT. Although its used here to delete duplicates, PROC SORT can be costly and resource-intensive in terms of performance, depending on the size of the data.

Now that all of our variables are in a single SAS data set, we need to subset the claims file to claims for specific medical conditions using ICD9 diagnosis codes. This step is often one of the first steps a lead study programmer performs in an observational research study.

Typically, an epidemiologist creates the specifications in a study protocol. They usually create a code list; a file containing the set of diagnosis, procedure, or drug codes to focus on for the study, contained in an excel spreadsheet. The programmer then imports the file into SAS, and subsets the real-world data sets they're using based on the medical codes.

For our analysis, we're interested in focusing on 3 specific medical conditions in our analysis; Congestive Heart Failure (CHF), Diabetes, and Chronic Kidney Disease (CKD). Using the diagnosis code description, we've located the relevant diagnosis codes for our 3 conditions of interest.

```
Data Claim_MBSF_v2;
  Length Condition $20;
  Set Claim_MBSF;

  Array DXCode {10} $ DXCD1-DXCD10;

  Do I=1 to 10;

    If Substr(DXCode{i}, 1, 3)='250' or DXCode{i}='253.5'
      Then Condition='Diabetes';

    Else If Substr(DXCode{i}, 1, 3)='428'
      Then Condition='CHF';

    Else If Substr(DXCode{i}, 1, 3) In ('580', '581', '582', '583', '584',
      '585', '586', '587', '588', '589',
      '590', '591', '592', '593')
      Then Condition='CKD';

  End;

  Keep Bene_ID Clm_ID From_Dt Thru_Dt Admsn_Dt DschrgDt Race Gender Bene_Dob
  State_Cd Cnty_Cd Death_Dt DeathStatus Description Condition DXCD1-DXCD10;

  If Condition In ('Diabetes', 'CHF', 'CKD');
Run;
```

Figure 6. Subsetting using a Diagnosis code list.

The code displayed in Figure 6 above shows this process. A new variable, condition, is created to flag claim records for each of the conditions, given the set of diagnosis codes. Each condition spans a range of diagnosis codes.

In claims files, there usually is a primary diagnosis code variable, with many secondary diagnosis codes, as many as 12. To include conditions in primary as well as secondary diagnoses an array is created to hold the 10 diagnosis code variables. An iterative do-loop is then used to cycle through each diagnosis as an element of the array.

Inside the loop, all the codes within a given range of codes are selected, by extracting only the first 3 characters of the code using the SUBSTR function. At the bottom of the step, a subsetting IF statement is used to limit the claims to conditions of interest based on the Condition variable. In a step not shown, the claims were further subsetting to Chronic Kidney Disease.

COMPUTING SURVIVAL TIME AND CENSORING

The next step in the project is to compute survival time and also to derive a binary variable to indicate censoring and other variables to use as stratification variables or covariates in the analysis.

To compute survival time, we use Death date and Admission date. Death date is our event date, with death being the event of interest. Admission date is our start date or index date. Where death date is missing, we estimate it using the last day of the current or following calendar year.

In Figure 7 below, is the DATA STEP code which computes three variables; Age, Survival time, and Censor. Age is computed from Beneficiary Date of Birth and either Admission Date or Death Date. If Death date is missing, then Admission date is used.

```
Data KidDis;
  Set Claim_MBSF_KidDis;

  *Use Death Date or Discharge Date to Compute Age;
  *Compute Survival Time Based on Death Date and Admission Date;
  *Create CENSOR flag variable for Censored patients;

  *Compute Survival time based on Death Date and Admission date;
  SurvTime=Death_Dt-Admsn_Dt;

  If DeathStatus='Y' Then Do; *Recorded Death Dates;
    Age=Floor((Death_Dt-Bene_Dob)/365.25);
    Censor=0;
  End;
  /* Death Date Missing;          */
  /* Patients lost to followup */
  Else Do;
    Age=Floor((DschrgDt-Bene_Dob)/365.25);
    Censor=1;
  End;

  If _N_ <= 10 Then Put ADMSN_DT= DSCHRGDT= DEATH_DT= BENE_DOB= AGE=
    SURVTIME= CENSOR=;

Run;
```

Figure 7. DATA STEP with IF-THEN-ELSE Conditional Logic

The variable CENSOR is created as a binary variable and assigned values of 0 and 1. For beneficiaries missing a date of death, values of 1 are assigned to censored observations, which didn't experience the event of interest.

For beneficiaries which have an actual date of death, values of 0 are assigned to uncensored observations which experienced the event of interest. Age is calculated from date of birth, and date of death. If death date is missing, then admission date is used as a proxy.

EXPLORATORY DATA ANALYSIS

Before actually running the survival analysis, we conducted exploratory data analysis to detect trends in the data and to view how survival time is distributed within specific categories of demographic variables.

Now that all the variables are in a single SAS data set, we're ready to produce descriptive statistics on survival time and our demographic group/subgroup variables. BASE SAS and ODS GRAPHICS contain many useful procedures and constructs for performing preliminary data analysis.

To that end, PROC MEANS was used to produce descriptive statistics for numeric variables. We also generated graphics and visualizations using PROC SGPLOT and PROC SGPANEL from the ODS GRAPHICS toolset. The SAS code to produce the descriptive statistics and graphics is presented below. Notice that only the code for chronic kidney disease (CKD) is displayed

```
Proc Means Data=KidDis N Mean Std Min Max;
  Var SurvTime;
  Class Race;
  Title'Chronic Kidney Disease by Race';
Run;

Proc SGPANEL Data=KidDis;
  Panelby Race;
  Histogram SurvTime;
  Title'Chronic Kidney Disease by Race';
Run;
```

In PROC MEANS, we specify race as our CLASS statement variable, since we're interested in analyzing survival time based on race.

For the task of visualizations, PROC SGPANEL was a better choice than PROC SGPLOT, because it generates multiple graphs based on the values of a categorical variable, which allows you to do a side-by-side visual comparison of the distributions. To analyze the distribution of survival time we choose a histogram as our chart type. The SAS output is presented below in Figure 8.

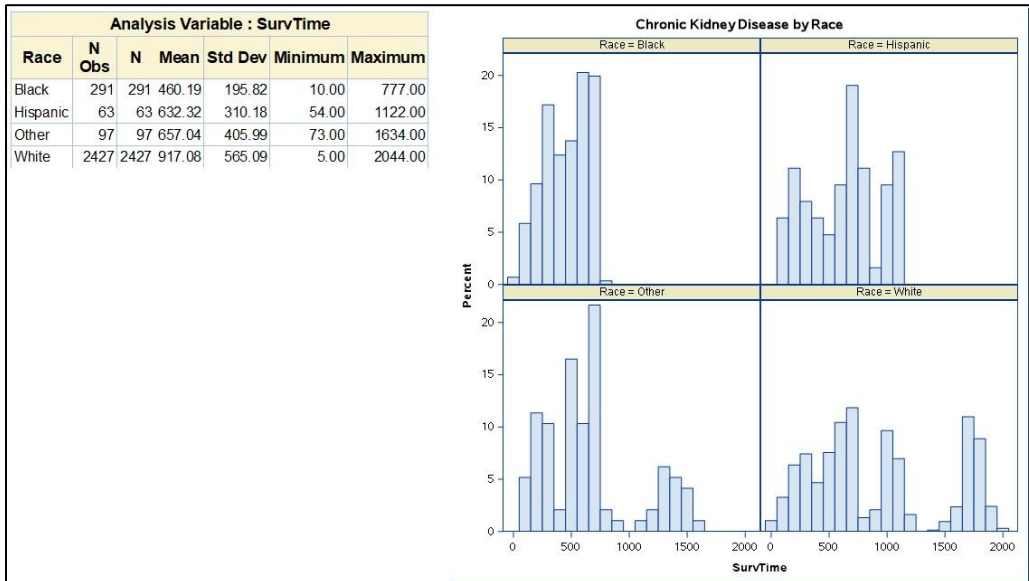


Figure 8. PROC MEANS and PROC SGPANEL Output

Reviewing the PROC MEANS output, we notice the small set of observations in the frequency counts for blacks (291), Hispanic (63) and other races (97) as compared to the large set for whites (2427) for Medicare beneficiaries diagnosed with chronic kidney disease.

Taking a look at the other statistics, we notice the different ranges and distributions of survival time for each racial group, based on minimum, maximum and mean survival time values. The minimum and maximum survival time provide us with the distribution starting and ending point, which shows substantial differences in the range of survival times between the race categories

Specifically, blacks have the lowest mean (average) survival time, and the narrowest distribution out of the four race categories. Conversely, whites have the highest average survival time and the widest distribution. The other two groups, Hispanics and other races, fall between the distribution for black and whites, respectively.

The output of PROC SG-panel allows the visualization of survival time distributions by race. For blacks, the distribution of survival times is skewed to the left, with the highest percentage of beneficiaries having medium range survival times. For the other race categories, the distribution of survival time is more evenly distributed amongst the range.

Hispanics have a distribution closer to normal, although still skewed to the left. Whites and other races have multiple-peak distributions, with other races having a left-skewed distribution, and whites having an approximately normal distribution. For Hispanics, the highest percentage of beneficiaries have a mid-level survival time, same as for blacks and other races. For white, the highest percentages have a survival time below the middle of the range, with high percentages at mid-level and high-level survival times also.

Based on the results of our investigatory data analysis, there is sufficient evidence that the distribution of survival times for beneficiaries is dependent on race, and at this juncture proceed with the survival analysis.

PRODUCT-LIMIT ESTIMATES WITH PROC LIFETEST

The first step in executing the survival analysis is to run PROC LIFETEST to generate product-limit estimates of the survival distribution function. As stated earlier, product-limit estimates are Kaplan-Meier estimates of the probability of survival.

For this example, we're interested in computing survival probabilities for Medicare beneficiaries diagnosed with chronic kidney disease (CKD). Based on our exploratory data analysis, we want to see how survival distributions vary by categories of race, and perform statistical significance tests on them.

AGE is included as a covariate in our model to explore the impact which patient age has on the survival functions. The SAS code for this first example is provided below.

```
Proc LifeTest Data=KidDis Plots=(s, ls, lls) Maxtime=600;  
    Time SurvTime*Censor(1);  
    Strata Race;  
    Test Age;  
Run;
```

The primary variables are specified in the TIME statement; SURVTIME for survival time and CENSOR. One of the initial procedures in survival analysis is to generate survival curves for groups of interest. The survival curves are produced using the PLOTS=(s, ls, lls) option on the PROC LIFETEST statement.

The STRATA statement is used to specify RACE as the stratification variable, and AGE is included as a covariate on the TEST statement. PROC LIFETEST will produce separate survival curves for each category of the RACE.

PROC LIFETEST generates 3 graphs in the SAS output, survival time against survival probability, Survival Time against the Negative Log of Survival Probability, and The Log of Survival Time and the Log of the Negative Log of Survival Probability. Survival time is measured in days until the event is experienced. In Figure 9 are a set of three graphs containing survival curves which plot survival time against survival probabilities for Medicare beneficiaries by race.

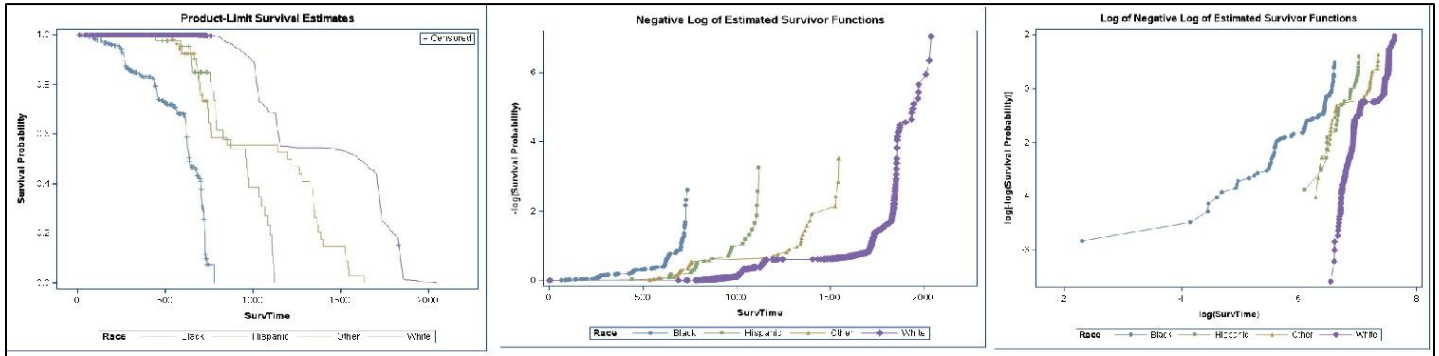


Figure 9. PROC LIFETEST OUTPUT – Survival Curves for Race Strata

In general, the survival functions show a decreasing probability of survival as survival time increases, regardless of racial group. However, the survival curves for blacks show the sharpest decrease in survival probability, with an angle closest to 90 degrees.

For Hispanics, the curves also show a sharp decrease, but with a higher probability of survival for longer durations.

For both other races and whites, the curves flatten out at a given probability level for a range of survival time before decreasing again. Both these groups have a higher probability of survival for longer durations than blacks and Hispanics.

Along with the survival curves, PROC LIFETEST produces a separate table with product-limit estimates for each value of the STRATA variable. The estimates are the probability of survival for computed survival times. Listed below is an excerpt from the product-limit table for black Medicare beneficiaries.

Product-Limit Survival Estimates					
SurvTime	Survival	Failure	Survival Standard Error	Number Failed	Number Left
0.00	1.0000	0	0	0	291
10.00	0.9966	0.00344	0.00343	1	290
19.00	*	.	.	1	289
61.00	*	.	.	1	288
63.00	0.9931	0.00690	0.00486	2	287
66.00	*	.	.	2	286
85.00	0.9896	0.0104	0.00596	3	285
86.00	0.9862	0.0138	0.00687	4	284
91.00	*	.	.	4	283
95.00	*	.	.	4	282
99.00	0.9827	0.0173	0.00769	5	281
100.00	*	.	.	5	280

PROC LIFETEST produces a series of additional tables containing results from other statistical tests and analytics. Not every table produced in PROC LIFETEST output is presented here due to reasons of practicality.

The procedure produces a summary table containing frequencies for each category of the strata variable, which experienced the event ('Failed'), and which did not experience the event ('Censored'). This table is presented below.

Summary of the Number of Censored and Uncensored Values					
Stratum	Race	Total	Failed	Censored	Percent Censored
1	Black	291	134	157	53.95
2	Hispanic	63	28	35	55.56
3	Other	97	37	60	61.86
4	White	2427	1153	1274	52.49
Total		2878	1352	1526	53.02

The summary data reveals that blacks were the group with the highest percentage of censored observations, followed by Hispanics, other races, and whites, respectively.

Of primary interest is the set of statistical tests PROC LIFETEST performs to determine the relationship between the stratification variable (race) and survival time. PROC LIFETEST conducts multiple tests to determine if there's a statistically significant difference in the survival functions based on this variable.

It initially computes a set of rank statistics, for each stratification level race category using two different methodologies; Log-Rank and Wilcoxon.

It conducts a set of three different significance tests; The Log-Rank, Wilcoxon, and Likelihood Ratio (LR) tests, to test for homogeneity over strata. It computes Chi-square test statistics and p-values for the three tests. Tables containing rank statistics and test of homogeneity are presented in Figure 10 below.

Rank Statistics			Test of Equality over Strata			
Race	Log-Rank	Wilcoxon	Test	Chi-Square	DF	Pr > Chi-Square
Black	124.99	244409	Log-Rank	1986.2729	3	<.0001
Hispanic	20.57	21926	Wilcoxon	1558.8424	3	<.0001
Other	20.56	18710	-2Log(LR)	45.5710	3	<.0001
White	-166.12	-285045				

Figure 10. Rank Statistics and Log-Rank, Wilcoxon and LR test results.

For each of the three significance tests, the chi-square test statistics are sufficiently large to generate corresponding p-values less than .05.

This meets the standard to reject the null hypothesis of no relationship between race and survival time. We have sufficient evidence to indicate a statistically significant difference in the survival curves according to race and to support the alternative hypothesis of a relationship between race and the survival time.

PROC LIFETEST also performs rank tests of the association between survival time and any covariates; variables listed on the TEST statement. In our example, Age was specified as the covariate.

By default, PROC LIFETEST performs two statistical tests; the Wilcoxon Test and the Log-Rank test for the covariates and outputs tables with test results for each, which are displayed in Figure 11.

Univariate Chi-Squares for the Wilcoxon Test					Univariate Chi-Squares for the Log-Rank Test				
Variable	Test Statistic	Standard Error	Chi-Square	Pr > Chi-Square	Variable	Test Statistic	Standard Error	Chi-Square	Pr > Chi-Square
Age	317.0	270.6	1.3727	0.2414	Age	430.9	445.5	0.9353	0.3335

Figure 11. Wilcoxon and Log-Rank Test Results for Age as a Covariate.

For the Wilcoxon test and the Log-Rank test, the Chi-square values are 1.37 and .93 respectively. These test statistics aren't large enough to produce p-values less than .05. Both the Wilcoxon test and the Log-Rank test provide no evidence of an association between age and survival time of Medicare beneficiaries diagnosed with chronic kidney disease.

PROC PHREG – PROPORTIONAL HAZARDS MODEL

We repeated the survival analysis of Medicare beneficiaries diagnosed with chronic kidney disease using the PHREG procedure to illustrate our results using a different underlying statistical methodology.

The features of PROC PHREG were discussed in an earlier section of this paper. To quickly review, PROC PHREG uses the Cox Proportional Hazards regression model. The underlying survival function is the hazard function which is unspecified. The procedure uses a parametric form for the covariates and explanatory variables, in contrast to PROC LIFETEST which is non-parametric.

One nice feature of PROC PHREG specification of reference categories for the stratification variables. This means that the statistical estimates which PROC PHREG produces can be interpreted in relation to specific categories of the strata variables.

The PROC PHREG code used for our project is provided below

```
Proc Phreg Data=KidDis Plots(Overlay)=Survival;
  Class Race(refno='Black') Gender(refno='Female');
  Model SurvTime*Censor(1) = Age Gender|Race;
Run;
```

In the CLASS statement, we specify explanatory variables which are classification variables. In contrast, PROC PHREG has a STRATA statement which is used to list any stratification variables. PROC PHREG makes a distinction between these two types of variables. PROC LIFETEST doesn't appear to have the same distinction, although any classification explanatory variables are listed in the TEST statement.

Another difference between PROC PHREG and PROC LIFETEST is that the volume of output is not as high for PROC PHREG as it is for PROC LIFETEST. PROC LIFETEST produces a higher number of output tables.

The CLASS LEVEL information table which displays information about the CLASS variables is displayed below. PROC PHREG creates dummy variables for categorical variables and the number of dummy variables it creates is dependent on the number of unique values in the categorical variables.

Class Level Information				
Class	Value	Design Variables		
Race	Black	0	0	0
	Hispanic	1	0	0
	Other	0	1	0
	White	0	0	1
Gender	Female	0		
	Male	1		

As appearing in the table, for the class variables, Race and Gender, 3 dummy variables are created for Race, and 1 dummy variable is created for Gender respectively.

In Figure 12, I've listed 2 additional output tables. In the first table are test results for testing the model validity as a whole, according to three different methods; Likelihood ratio, Score and Wald. The second table contains test results for individual covariates, effect and explanatory variables whether numeric or classification.

Testing Global Null Hypothesis: BETA=0				Joint Tests			
Test	Chi-Square	DF	Pr > ChiSq	Effect	DF	Wald Chi-Square	Pr > ChiSq
Likelihood Ratio	780.0253	8	<.0001	Age	1	1.0067	0.3157
Score	2012.6255	8	<.0001	Gender	1	1.9993	0.1574
Wald	558.2375	8	<.0001	Race	3	470.2797	<.0001
				Race*Gender	3	6.8372	0.0773

Figure 12. PHREG Output – Tests for Strata and Individual Covariates.

For the test of the model as a whole, regardless of which method you use, the model is statistically significant at the .05 level. That is, the chi-square test statistics are large enough to produce p-values less than .05.

For the individual effect tests, PROC PHREG computes the Wald Chi-Square test statistics. The first two explanatory variables, Age and Gender have low Wald Chi-square test statistics, and corresponding p-values greater than .05. Thus, they're both not statistically significant. The third variable, Race, is statistically significant at the .05 level, having a large Wald Chi-square statistic.

The individual effects table also supplies results for the interaction between Race and Gender as a separate variable, Race*Gender. The interaction term is found to be not statistically significant at the .05 level, with a p-value of .0773. Though, of the three statistically invalid variables, Race*Gender is the closest to statistical significance, based on its p-value.

In the last table of output, PROC PHREG produces the Analysis of Maximum Likelihood Estimates table which contains parameter estimates, standard errors, and Chi-square statistics which test the effect of specific categories of covariates in relation to the reference categories of those same covariates. P-values are provided to determine statistical significance.

In addition to these statistics, the table provides Hazard ratios for each parameter, for which the ratio is calculated. The Hazard ratio is defined as the ratio of Hazard rates in response to an increase in one unit of the covariate or explanatory variable.

In order to properly and directly interpret the Hazard Ratios, the provided Hazard ratio in the table must be plugged into the equation;

$$1 - (\text{HazardRatio})^{10}$$

The result is the percentage decrease in the hazard rate for an increase in 10 units of the covariate.

The maximum likelihood parameter estimates, and other statistics are provided in Figure 13 below.

Analysis of Maximum Likelihood Estimates									
Parameter			DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio	Label
Age			1	-0.00226	0.00225	1.0067	0.3157	0.998	
Gender	Male		1	-0.25305	0.17896	1.9993	0.1574		. Gender Male
Race	Hispanic		1	-3.52040	0.33392	111.1458	<.0001		. Race Hispanic
Race	Other		1	-4.36483	0.33527	169.4927	<.0001		. Race Other
Race	White		1	-5.30480	0.25614	428.9400	<.0001		. Race White
Race*Gender	Hispanic	Male	1	0.78430	0.44257	3.1404	0.0764		. Race Hispanic * Gender Male
Race*Gender	Other	Male	1	0.83113	0.37513	4.9087	0.0267		. Race Other * Gender Male
Race*Gender	White	Male	1	0.29827	0.18814	2.5134	0.1129		. Race White * Gender Male

Figure 13. PHREG Output – Maximum Likelihood Estimates.

In the table, Hazard ratios have been calculated for only one of the covariates, AGE.

Using Age as an example, and plugging the Hazard ratio for age into the equation, we get

$$1 - (.998)^{10} = 2\%$$

Thus, an increase in 10 years of Age results in a 2% reduction in the hazard rate for Medicare beneficiaries with chronic kidney disease.

The other covariates in the model, Race and Gender are categorical variables. Notice that the Maximum likelihood estimates table has rows for only specific values of the categorical variables, excluding the reference values. The estimates are computed for non-reference categories of the covariate.

As coded in PROC PHREG, our reference categories for race and gender are ‘Black’ and ‘Female’, respectively. Thus, for Race, estimates are computed for ‘Hispanic’, ‘Other’, and ‘White’. Likewise for Gender, estimates are computed for ‘Male’. For the interaction of Race and Gender, estimates are computed for the combination of ‘Hispanic’, ‘Other’, ‘White’, and ‘Male’.

Examining the test statistics and p-values for the classification variables, we notice that for Race, the effects of ‘Hispanic’, ‘Other’ and ‘White’, in relation to ‘Black’ are statistically significant at the .05 level, with p-values of <.0001.

For Race*Gender, Race=‘Other’ and Gender=‘Male’ in relation to ‘Black’ and ‘Female’ was statistically significant at the .05 level. The other variables (Age, etc.) and categorical values were not statistically significant at the .05 level.

Notice in Figure 13, that the hazard ratio was only calculated for the Age variable. In many scenarios, hazard ratios will not be computed for particular variables and variable values. However, you can override this outcome, using the HAZARDRATIO statement in PROC PHREG. The HAZARDRATIO statement computes hazard ratios for specific covariates, where they were missing from output.

CONCLUSION

The SAS/STAT package contains a set of procedures for performing survival analysis. To conduct a proper survival analysis, care must be taken to perform required steps of data manipulation, data cleaning, subsetting by applying a code list, and exploratory data analysis. The choice of a particular construct depends on the need to do a parametric vs. a semi-parametric or non-parametric analysis, as well as the underlying survival function. With this paper, the objective was to illustrate the steps involved in performing a survival analysis for a longitudinal study and show the capabilities of the survival analysis procedures on real-world data sets, specifically Medicare claims.

REFERENCES

SAS/STAT 15.3 User's Guide. Introduction to Survival Analysis Procedures. Survival Analysis Procedures. SAS Institute, 2025. https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/statug/statug_introsurv_sect002.htm.

David Shen, & Jane Lu. 2014 'Survival Analysis Approaches and New Developments using SAS'. Proceedings of the Pharmaceutical SAS Users Group Conference, 2014. Paper PO02. <https://www.lexjansen.com/pharmasug/2014/PO/PharmaSUG-2014-PO02.pdf>

David Shen, & Jane Lu. 2018 'Application of Survival Analysis in Multiple Events using SAS'. Proceedings of the Pharmaceutical SAS Users Group Conference. Paper PO02. <https://www.lexjansen.com/pharmasug/2018/EP/PharmaSUG-2018-EP02.pdf>

ACKNOWLEDGMENTS

The author would like to thank John LeBore, Operations Chair, David Corliss, Academic Chair and Analysis and Advanced Analytics Section Chair, and the MWSUG 2025 Executive Committee and Conference Team for accepting my abstract and paper and for organizing this conference.

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact the author at:

Jay Iyengar
Data Systems Consultants LLC
datasyscon@gmail.com
<https://www.linkedin.com/in/datasysconsult/>

Jay Iyengar is Director of Data Systems Consultants LLC. He's a SAS consultant, trainer, and SAS Certified Advanced Programmer. He's been an invited speaker at several SAS user group conferences (WILSU, WCSUG, SESUG) and has presented papers and training seminars at SAS Global Forum, Pharmaceutical SAS Users Group (PharmaSUG), and other regional and local SAS User Group conferences (MWSUG, NESUG, WUSS, MISUG). He was co-leader and organizer of the Chicago SAS Users Group (WCSUG) from 2015-19. He received his bachelor's degree from Syracuse University in Public Policy and Economics, and his master's degree from the American University.

TRADEMARK CITATION

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies

APPENDIX I – PROC CONTENTS OUTPUT

Data Set Name	SURVMEDI.IP2010CLAIM	Observations	13916
Member Type	DATA	Variables	36
Engine	V9	Indexes	0
Created	10/28/2013 10:10:52	Observation Length	232
Last Modified	10/28/2013 10:10:52	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	YES
Label			
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		

Engine/Host Dependent Information	
Data Set Page Size	16384
Number of Data Set Pages	200
First Data Page	1
Max Obs per Page	70
Obs in First Data Page	45
Number of Data Set Repairs	0
Filename	/home/iyenj/SAS Papers/Suvivial_Anal_Medicare/ip2010claim.sas7bdat
Release Created	9.0301M1
Host Created	W32_7PRO
Inode Number	2170931617
Access Permission	rw-r--r--
Owner Name	iyenj
File Size	3MB
File Size (bytes)	3277824

Alphabetic List of Variables and Attributes						
#	Variable	Type	Len	Format	Informat	Label
12	ADMSN_DT	Num	4	YYMMDDN8.	YYMMDD8.	Inpatient admission date
13	ADMTG_DGNS_CD	Char	5	\$5.		Claim Admitting Diagnosis Code
9	AT_NPI	Char	10	\$10.		Attending Physician - National Provider Identifier Number
1	BENE_ID	Char	16	\$16.		Beneficiary Code
17	BLDDEDAM	Num	8	12.2		NCH Beneficiary Blood Deductible Liability Amount
2	CLM_ID	Char	15	\$15.		Claim ID
16	COIN_AMT	Num	8	12.2		NCH Beneficiary Part A Coinsurance Liability Amount
15	DED_AMT	Num	8	12.2		NCH Beneficiary Inpatient Deductible Amount
20	DRG_CD	Char	3	\$3.		Claim Diagnosis Related Group Code
19	DSCHRGDT	Num	4	YYMMDDN8.	YYMMDD8.	Inpatient discharged date
4	FROM_DT	Num	4	YYMMDDN8.	YYMMDD8.	Claims start date
31	ICD9_PRCDR_CD_1	Char	5	\$5.		Claim Procedure Code 1
32	ICD9_PRCDR_CD_2	Char	5	\$5.		Claim Procedure Code 2

Alphabetic List of Variables and Attributes						
#	Variable	Type	Len	Format	Informat	Label
33	ICD9_PRCDR_CD_3	Char	5	\$5.		Claim Procedure Code 3
34	ICD9_PRCDR_CD_4	Char	5	\$5.		Claim Procedure Code 4
35	ICD9_PRCDR_CD_5	Char	5	\$5.		Claim Procedure Code 5
36	ICD9_PRCDR_CD_6	Char	5	\$5.		Claim Procedure Code 6
21	ICD_DGNS_CD1	Char	5	\$5.		Claim Diagnosis Code 1
22	ICD_DGNS_CD2	Char	5	\$5.		Claim Diagnosis Code 2
23	ICD_DGNS_CD3	Char	5	\$5.		Claim Diagnosis Code 3
24	ICD_DGNS_CD4	Char	5	\$5.		Claim Diagnosis Code 4
25	ICD_DGNS_CD5	Char	5	\$5.		Claim Diagnosis Code 5
26	ICD_DGNS_CD6	Char	5	\$5.		Claim Diagnosis Code 6
27	ICD_DGNS_CD7	Char	5	\$5.		Claim Diagnosis Code 7
28	ICD_DGNS_CD8	Char	5	\$5.		Claim Diagnosis Code 8
29	ICD_DGNS_CD9	Char	5	\$5.		Claim Diagnosis Code 9
30	ICD_DGNS_CD10	Char	5	\$5.		Claim Diagnosis Code 10
10	OP_NPI	Char	10	\$10.		Operating Physician - National Provider Identifier Number
11	OT_NPI	Char	10	\$10.		Other Physician - - National Provider Identifier Number
14	PER_DIEM	Num	8	12.2		Claim Pass Thru Per Diem Amount
7	PMT_AMT	Num	8	12.2		Claim Payment Amount
6	PROVIDER	Char	6	\$6.		Provider Institution
8	PRPAYAMT	Num	8	12.2		NCH Primary Payer Claim Paid Amount
3	SEGMENT	Num	3	2.		Claim Line Segment
5	THRU_DT	Num	4	YYMMDDN8.	YYMMDD8.	Claims end date
18	UTIL_DAY	Num	3	3.		Claim Utilization Day Count

Sort Information	
Sortedby	BENE_ID CLM_ID
Validated	YES
Character Set	ANSI

Data Set Name	SURVMEDI.MBSF_AB_2010	Observations	112754
Member Type	DATA	Variables	32
Engine	V9	Indexes	0
Created	10/28/2013 10:17:53	Observation Length	152
Last Modified	10/28/2013 10:17:53	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		

Engine/Host Dependent Information	
Data Set Page Size	12288
Number of Data Set Pages	1410
First Data Page	1
Max Obs per Page	80
Obs in First Data Page	44
Number of Data Set Repairs	0
Filename	/home/iyenj/SAS Papers/Suvivial_Anal_Medicare/mbsf_ab_2010.sas7bdat
Release Created	9.0301M1
Host Created	W32_7PRO
Inode Number	2174747054
Access Permission	rw-r--r--
Owner Name	iyenj
File Size	17MB
File Size (bytes)	17327104

Alphabetic List of Variables and Attributes						
#	Variable	Type	Len	Format	Informat	Label
2	BENE_DOB	Num	4	YYMMDDN8.	YYMMDD8.	Date of birth
9	BENE_HI_CVRAGE_TOT_MONS	Num	3	2.		Total number of months of part A coverage for the beneficiary.
11	BENE_HMO_CVRAGE_TOT_MONS	Num	3	2.		Total number of months of HMO coverage for the beneficiary.
1	BENE_ID	Char	16	\$16.		Beneficiary Code
10	BENE_SMI_CVRAGE_TOT_MONS	Num	3	2.		Total number of months of part B coverage for the beneficiary.
31	BENRES_CAR	Num	8	10.2		Carrier annual beneficiary responsibility amount
25	BENRES_IP	Num	8	10.2		Inpatient annual beneficiary responsibility amount
28	BENRES_OP	Num	8	10.2		Outpatient Institutional annual beneficiary responsibility amount
8	CNTY_CD	Char	3	\$3.		County Code
3	DEATH_DT	Num	4	YYMMDDN8.	YYMMDD8.	Date of death
6	ESRD_IND	Char	1	\$1.		End stage renal disease Indicator
30	MEDREIMB_CAR	Num	8	10.2		Carrier annual Medicare reimbursement amount
24	MEDREIMB_IP	Num	8	10.2		Inpatient annual Medicare reimbursement amount
27	MEDREIMB_OP	Num	8	10.2		Outpatient Institutional annual Medicare reimbursement amount
12	PLAN_CVRG_MOS_NUM	Char	2	\$2.		Total number of months of part D plan coverage for the beneficiary.
32	PPPYMT_CAR	Num	8	10.2		Carrier annual primary payer reimbursement amount

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat	Label
26	PPPYMT_IP	Num	8	10.2		Inpatient annual primary payer reimbursement amount
29	PPPYMT_OP	Num	8	10.2		Outpatient Institutional annual primary payer reimbursement amount
5	RACE	Char	1	\$1.		Beneficiary Race Code
4	SEX	Char	1	\$1.		Sex
13	SP_ALZHDMTA	Num	3	1.		Chronic Condition: Alzheimer or related disorders or senile
14	SP_CHF	Num	3	1.		Chronic Condition: Heart Failure
15	SP_CHRNKIDN	Num	3	1.		Chronic Condition: Chronic Kidney Disease
16	SP_CNCR	Num	3	1.		Chronic Condition: Cancer
17	SP_COPD	Num	3	1.		Chronic Condition: Chronic Obstructive Pulmonary Disease
18	SP_DEPRESSN	Num	3	1.		Chronic Condition: Depression
19	SP_DIABETES	Num	3	1.		Chronic Condition: Diabetes
20	SP_ISCHMCHT	Num	3	1.		Chronic Condition: Ischemic Heart Disease
21	SP_OSTEOPRS	Num	3	1.		Chronic Condition: Osteoporosis
22	SP_RA_OA	Num	3	1.		Chronic Condition: RA/OA
23	SP_STRKETIA	Num	3	1.		Chronic Condition: Stroke/transient Ischemic Attack
7	STATE_CD	Char	2	\$2.		State Code

APPENDIX II – SAS LOG

```
1  OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
72
73  Libname SurvMedi '/home/iyenj/SAS Papers/Suvivial_Anal_Medicare';
NOTE: Libref SURVMEDI was successfully assigned as follows:
      Engine:    V9
      Physical Name: /home/iyenj/SAS Papers/Suvivial_Anal_Medicare
74  Libname SurvFmt '/home/iyenj/SAS Papers/Suvivial_Anal_Medicare/FormatLib';
NOTE: Libref SURVFMT was successfully assigned as follows:
      Engine:    V9
      Physical Name: /home/iyenj/SAS Papers/Suvivial_Anal_Medicare/FormatLib
75
76  Proc Format Library=SurvFmt;
77  Value $RaceCat
78      '0'='Unknown'
79      '1'='White'
80      '2'='Black'
81      '3'='Other'
82      '4'='Asian'
83      '5'='Hispanic'
84      '6'='North American Native';
NOTE: Format $RACECAT is already on the library SURVFMT.FORMATS.
NOTE: Format $RACECAT has been written to SURVFMT.FORMATS.
85  Value $Gender
86      '1'='Male'
87      '2'='Female';
NOTE: Format $GENDER is already on the library SURVFMT.FORMATS.
NOTE: Format $GENDER has been written to SURVFMT.FORMATS.
87  !
88  Run;
```

NOTE: PROCEDURE FORMAT used (Total process time):

```
real time      0.00 seconds
user cpu time   0.00 seconds
system cpu time 0.00 seconds
memory         296.84k
OS Memory      31140.00k
Timestamp      04/07/2025 04:25:28 PM
```

```
89
90  Proc Contents Data = SurvMedi.ICD9DX;
```

NOTE: Data file SURVMEDI.ICD9DX.DATA is in a format that is native to another host, or the file encoding does not match the session encoding. Cross Environment Data Access will be used, which might require additional CPU resources and might reduce performance.

NOTE: PROCEDURE CONTENTS used (Total process time):

```
real time      0.05 seconds
user cpu time   0.05 seconds
system cpu time 0.00 seconds
memory         4151.18k
```

OS Memory 33452.00k
Timestamp 04/07/2025 04:25:28 PM

91 Proc Contents Data = SurvMedi.ip2010claim;

NOTE: Data file SURVMEDI.IP2010CLAIM.DATA is in a format that is native to another host, or the file encoding does not match the session encoding. Cross Environment Data Access will be used, which might require additional CPU resources and might reduce performance.

NOTE: PROCEDURE CONTENTS used (Total process time):

real time 0.07 seconds
user cpu time 0.07 seconds
system cpu time 0.01 seconds
memory 1951.50k
OS Memory 32936.00k
Timestamp 04/07/2025 04:25:29 PM

92 Proc Contents Data = SurvMedi.mbsf_ab_2010;

93 Run;

NOTE: PROCEDURE CONTENTS used (Total process time):

real time 0.06 seconds
user cpu time 0.06 seconds
system cpu time 0.00 seconds
memory 2349.21k
OS Memory 33964.00k
Timestamp 04/07/2025 04:25:29 PM

94

95 Proc Print Data = SurvMedi.ICD9DX(Obs=25);

NOTE: Data file SURVMEDI.ICD9DX.DATA is in a format that is native to another host, or the file encoding does not match the session encoding. Cross Environment Data Access will be used, which might require additional CPU resources and might reduce performance.

96 Run;

NOTE: There were 25 observations read from the data set SURVMEDI.ICD9DX.

NOTE: PROCEDURE PRINT used (Total process time):

real time 0.02 seconds
user cpu time 0.02 seconds
system cpu time 0.00 seconds
memory 1500.59k
OS Memory 33448.00k
Timestamp 04/07/2025 04:25:29 PM

98 /* Join IP Claims File with Diagnosis Code Lookup Table to add Diagnosis Description */

99 Data ip2010claim;

100 Length DXCD1-DXCD10 \$6;

101

102 Set SURVMEDI.ip2010claim;

NOTE: Data file SURVMEDI.IP2010CLAIM.DATA is in a format that is native to another host, or the file encoding does not match the session encoding. Cross Environment Data Access will be used, which might require additional CPU resources and might reduce performance.

103

104 Array ICDDX{10}\$ICD_DGNS_CD1-ICD_DGNS_CD10;

```

105 Array DXCode {10} $ DXCD1-DXCD10;
106
107 Do I = 1 to 10;
108 DXCode{i} = Substr(ICDDX{i}, 1, 3)||'|'||Substr(ICDDX{i}, 4, 2);
109 End;
110
111 Drop ICD_DGNS_CD1-ICD_DGNS_CD10;
112 Run;

```

NOTE: There were 13916 observations read from the data set SURVMEDI.IP2010CLAIM.

NOTE: The data set WORK.IP2010CLAIM has 13916 observations and 37 variables.

NOTE: DATA statement used (Total process time):

```

real time      0.06 seconds
user cpu time   0.06 seconds
system cpu time 0.00 seconds
memory         2423.53k
OS Memory      34472.00k
Timestamp      04/07/2025 04:25:29 PM

```

```

113
114 Proc Sql;
115 Create Table IPClaim_v2 as
116 Select A.*, B.Description
117 From ip2010claim as A Left Join SurvMedi.ICD9DX as B
118 On A.DXCD1=B.Code;

```

NOTE: Data file SURVMEDI.ICD9DX.DATA is in a format that is native to another host, or the file encoding does not match the session encoding. Cross Environment Data Access will be used, which might require additional CPU resources and might reduce performance.

NOTE: Table WORK.IPCLAIM_V2 created, with 13916 rows and 38 columns.

```

119 Quit;

```

NOTE: PROCEDURE SQL used (Total process time):

```

real time      0.04 seconds
user cpu time   0.03 seconds
system cpu time 0.02 seconds
memory         26113.78k
OS Memory      57784.00k
Timestamp      04/07/2025 04:25:29 PM

```

```

121 Proc Print Data=IPClaim_v2(Obs=25);
122 Var DXCD1 Description;
123 Run;

```

NOTE: There were 25 observations read from the data set WORK.IPCLAIM_V2.

NOTE: PROCEDURE PRINT used (Total process time):

```

real time      0.02 seconds
user cpu time   0.02 seconds
system cpu time 0.00 seconds
memory         2030.62k
OS Memory      35496.00k
Timestamp      04/07/2025 04:25:29 PM

```

```

124
125 Proc Freq Data=IPClaim_v2 Order=FREQ;

```

```
126 Tables Description / List Missing;
127 Run;
```

NOTE: There were 13916 observations read from the data set WORK.IPCLAIM_V2.

NOTE: PROCEDURE FREQ used (Total process time):

```
real time      1.20 seconds
user cpu time   1.20 seconds
system cpu time 0.00 seconds
memory         12701.68k
OS Memory      45740.00k
Timestamp      04/07/2025 04:25:30 PM
```

```
128
129 /* Join New IP Claim File with Membership File */
130 Data mbsf_ab_2010;
131 Set SurvMedi.mbsf_ab_2010(Rename=(Race=RaceGrp));
132
133 /* Create Race and Gender variables with formatted values */
134 Race = Put(RaceGrp, $RaceCat.);
135 Gender = Put(Sex, $Gender.);
136
137 If Death_Dt^=. Then
138     DeathStatus='Y';
139 Else
140     DeathStatus='N';
141
142 Drop RaceGrp Sex;
143 Run;
```

NOTE: There were 112754 observations read from the data set SURVMEDI.MBSF_AB_2010.

NOTE: The data set WORK.MBSF_AB_2010 has 112754 observations and 33 variables.

NOTE: DATA statement used (Total process time):

```
real time      0.05 seconds
user cpu time   0.03 seconds
system cpu time 0.02 seconds
memory         3704.28k
OS Memory      39596.00k
Timestamp      04/07/2025 04:25:30 PM
```

```
144
145 Proc Sort Data = mbsf_ab_2010 Nodupkey;
146 By Bene_ID;
147 Run;
```

NOTE: There were 112754 observations read from the data set WORK.MBSF_AB_2010.

NOTE: 0 observations with duplicate key values were deleted.

NOTE: The data set WORK.MBSF_AB_2010 has 112754 observations and 33 variables.

NOTE: PROCEDURE SORT used (Total process time):

```
real time      0.04 seconds
user cpu time   0.03 seconds
system cpu time 0.02 seconds
memory         26497.68k
OS Memory      61676.00k
Timestamp      04/07/2025 04:25:30 PM
```

```

149 Proc Sql;
150   Create Table Claim_MBSF as
151     Select A.*,
152           B.Race,
153           B.Gender,
154           B.State_Cd,
155           B.Cnty_Cd,
156           B.Death_Dt,
157           B.DeathStatus,
158           B.Bene_Dob
159   From IPClaim_v2 as A, mbsf_ab_2010 as B
160   Where A.Bene_ID=B.Bene_ID;

```

NOTE: Table WORK.CLAIM_MBSF created, with 13916 rows and 45 columns.

```
161 Quit;
```

NOTE: PROCEDURE SQL used (Total process time):

```

real time      0.03 seconds
user cpu time   0.02 seconds
system cpu time 0.02 seconds
memory         22728.31k
OS Memory      57528.00k
Timestamp      04/07/2025 04:25:30 PM

```

```

163 Proc Freq Data=Claim_MBSF;
164   Tables DeathStatus / List Missing;
165 Run;

```

NOTE: There were 13916 observations read from the data set WORK.CLAIM_MBSF.

NOTE: PROCEDURE FREQ used (Total process time):

```

real time      0.01 seconds
user cpu time   0.01 seconds
system cpu time 0.00 seconds
memory         2241.25k
OS Memory      37804.00k
Timestamp      04/07/2025 04:25:30 PM

```

```

167 Data Claim_MBSF_v2;
168   Length Condition $20;
169   Set Claim_MBSF;
170
171   Array DXCode {10} $ DXCD1-DXCD10;
172
173   Do I=1 to 10;
174
175     If Substr(DXCode{i}, 1, 3)='250' or DXCode{i}='253.5'
176       Then Condition='Diabetes';
177
178     Else If Substr(DXCode{i}, 1, 3)='428'
179       Then Condition='CHF';
180
181     Else If Substr(DXCode{i}, 1, 3) In ('580', '581', '582', '583', '584',
182                                       '585', '586', '587', '588', '589',
183                                       '590', '591', '592', '593')
184       Then Condition='CKD';

```

```

185 End;
186
187 Keep Bene_ID Clm_ID From_Dt Thru_Dt Admsn_Dt DschrgDt Race Gender Bene_Dob
188     State_Cd Cnty_Cd Death_Dt DeathStatus Description Condition DXCD1-DXCD10;
189
190 If Condition In ('Diabetes', 'CHF', 'CKD');
191 Run;

```

NOTE: There were 13916 observations read from the data set WORK.CLAIM_MBSF.

NOTE: The data set WORK.CLAIM_MBSF_V2 has 7714 observations and 25 variables.

NOTE: DATA statement used (Total process time):

```

real time      0.02 seconds
user cpu time   0.02 seconds
system cpu time 0.01 seconds
memory         3666.81k
OS Memory      39596.00k
Timestamp      04/07/2025 04:25:30 PM

```

```

192
193 Data Claim_MBSF_KidDis;
194 Set Claim_MBSF_v2(Where=(Condition='CKD'));
195
196 If (225<=_N_<=275 or 325<=_N_<=340 or
197     401<=_N_<=422 or 1501<=_N_<=2000) Then Death_Dt='31DEC2010'D;
198
199 Else If (275<_N_<=291 Or 340<_N_<=354 Or
200     422<_N_<=451 Or 2001<_N_<=2878) Then Death_Dt='31DEC2011'D;
201
202 If (225<=_N_<=291 or 325<=_N_<=354 or 401<=_N_<=451 or 1501<=_N_<=2878)
203     Then DeathStatus='N';
204 Else DeathStatus='Y';
205
206 Run;

```

NOTE: There were 2878 observations read from the data set WORK.CLAIM_MBSF_V2.

WHERE Condition='CKD';

NOTE: The data set WORK.CLAIM_MBSF_KIDDIS has 2878 observations and 25 variables.

NOTE: DATA statement used (Total process time):

```

real time      0.00 seconds
user cpu time   0.01 seconds
system cpu time 0.00 seconds
memory         2686.06k
OS Memory      38316.00k
Timestamp      04/07/2025 04:25:30 PM

```

```

207
208 Proc Freq Data = Claim_MBSF_KidDis;
209 Tables Condition / List Missing;
210 Tables Condition*DeathStatus / List Missing;
211 Run;

```

NOTE: There were 2878 observations read from the data set WORK.CLAIM_MBSF_KIDDIS.

NOTE: PROCEDURE FREQ used (Total process time):

```

real time      0.02 seconds
user cpu time   0.02 seconds
system cpu time 0.00 seconds

```

memory 2216.46k
OS Memory 38200.00k
Timestamp 04/07/2025 04:25:30 PM

```
213 Proc Sql;
214 Create Table Claim_Sum As
215 Select Bene_ID, Count(Bene_ID) as Num_Visits
216 From Claim_MBSF_KidDis
217 Group By Bene_ID
218 Having Num_Visits>1;
```

NOTE: Table WORK.CLAIM_SUM created, with 96 rows and 2 columns.

```
219 Quit;
```

NOTE: PROCEDURE SQL used (Total process time):

real time 0.00 seconds
user cpu time 0.01 seconds
system cpu time 0.00 seconds
memory 6250.56k
OS Memory 42284.00k
Timestamp 04/07/2025 04:25:30 PM

```
220
221 Data KidDis;
222 Set Claim_MBSF_KidDis;
223
224 *Use Death Date or Discharge Date to Compute Age;
225 *Compute Survival Time Based on Death Date and Admission Date;
226 *Create CENSOR flag variable for Censored patients;
227
228 *Compute Survival time based on Death Date and Admission date;
229 SurvTime=Death_Dt-Admsn_Dt;
230
231 If DeathStatus='Y' Then Do; *Recorded Death Dates;
232 Age=Floor((Death_Dt-Bene_Dob)/365.25);
233 Censor=0;
234 End;
235 /* Death Date Missing; */
236 /* Patients lost to followup */
237 Else Do;
238 Age=Floor((DschrgDt-Bene_Dob)/365.25);
239 Censor=1;
240 End;
241
242 If _N_<=10 Then Put ADMSN_DT= DSCHRGDT= DEATH_DT= BENE_DOB= AGE= SURVTIME=
CENSOR=;
243 Run;
```

ADMSN_DT=20100522 DSCHRGDT=20100612 DEATH_DT=20110901 BENE_DOB=19190901 Age=92 SurvTime=467 Censor=0
ADMSN_DT=20100830 DSCHRGDT=20100907 DEATH_DT=20150901 BENE_DOB=19531201 Age=61 SurvTime=1828 Censor=0
ADMSN_DT=20100114 DSCHRGDT=20100119 DEATH_DT=20150101 BENE_DOB=19571101 Age=57 SurvTime=1813 Censor=0
ADMSN_DT=20100703 DSCHRGDT=20100712 DEATH_DT=20130101 BENE_DOB=19251001 Age=87 SurvTime=913 Censor=0
ADMSN_DT=20100222 DSCHRGDT=20100302 DEATH_DT=20121201 BENE_DOB=19620601 Age=50 SurvTime=1013 Censor=0
ADMSN_DT=20100129 DSCHRGDT=20100208 DEATH_DT=20130101 BENE_DOB=19410501 Age=71 SurvTime=1068 Censor=0
ADMSN_DT=20100708 DSCHRGDT=20100710 DEATH_DT=20150401 BENE_DOB=19431101 Age=71 SurvTime=1728 Censor=0
ADMSN_DT=20100518 DSCHRGDT=20100523 DEATH_DT=20150101 BENE_DOB=19370201 Age=77 SurvTime=1689 Censor=0
ADMSN_DT=20100713 DSCHRGDT=20100722 DEATH_DT=20150101 BENE_DOB=19370201 Age=77 SurvTime=1633 Censor=0
ADMSN_DT=20101002 DSCHRGDT=20101020 DEATH_DT=20151101 BENE_DOB=19420901 Age=73 SurvTime=1856 Censor=0

NOTE: There were 2878 observations read from the data set WORK.CLAIM_MBSF_KIDDIS.

NOTE: The data set WORK.KIDDIS has 2878 observations and 28 variables.

NOTE: DATA statement used (Total process time):

real time 0.00 seconds
user cpu time 0.00 seconds
system cpu time 0.00 seconds
memory 2337.12k
OS Memory 37420.00k
Timestamp 04/07/2025 04:25:30 PM

```
244  
245 /* Graph Survival time Group by Sex and Race for each Condition/Disease */  
246 /* use PROC SGPLOT */  
247  
248 Proc Freq Data=KidDis;  
249 Tables DeathStatus / List Missing;  
250 Run;
```

NOTE: There were 2878 observations read from the data set WORK.KIDDIS.

NOTE: PROCEDURE FREQ used (Total process time):

real time 0.01 seconds
user cpu time 0.01 seconds
system cpu time 0.00 seconds
memory 1492.09k
OS Memory 37548.00k
Timestamp 04/07/2025 04:25:30 PM

```
250  
251  
252 /* Kidney Disease */  
253 /* By Race, Gender, and Race & Gender */  
254  
255 Proc Means Data=KidDis N Mean Std Min Max Maxdec=2;  
256 Var SurvTime;  
257 Class Race;  
258 Title'Chronic Kidney Disease by Race';  
259 Run;
```

NOTE: There were 2878 observations read from the data set WORK.KIDDIS.

NOTE: PROCEDURE MEANS used (Total process time):

real time 0.02 seconds
user cpu time 0.02 seconds
system cpu time 0.00 seconds
memory 8654.34k
OS Memory 43452.00k
Timestamp 04/07/2025 04:25:30 PM

```
261 Proc SGPanel Data=KidDis;
262     Panelby Race;
263     Histogram SurvTime;
264     Title'Chronic Kidney Disease by Race';
265 Run;
```

NOTE: PROCEDURE SGPANEL used (Total process time):

```
real time      2.88 seconds
user cpu time   0.16 seconds
system cpu time 0.03 seconds
memory         21619.79k
OS Memory      55212.00k
Timestamp      04/07/2025 04:25:33 PM
```

NOTE: There were 2878 observations read from the data set WORK.KIDDIS.

```
266
267 Proc Means Data=KidDis N Mean Std Min Max;
268     Var SurvTime;
269     Class Gender;
270     Title'Chronic Kidney Disease by Gender';
271 Run;
```

NOTE: There were 2878 observations read from the data set WORK.KIDDIS.

NOTE: PROCEDURE MEANS used (Total process time):

```
real time      0.02 seconds
user cpu time   0.02 seconds
system cpu time 0.01 seconds
memory         9736.96k
OS Memory      63932.00k
Timestamp      04/07/2025 04:25:33 PM
```

```
271
272
273 Proc SGPanel Data=KidDis;
274     Panelby Gender;
275     Histogram SurvTime;
276     Title'Chronic Kidney Disease by Gender';
277 Run;
```

NOTE: PROCEDURE SGPANEL used (Total process time):

```
real time      0.38 seconds
user cpu time   0.09 seconds
system cpu time 0.01 seconds
memory         5439.07k
OS Memory      57776.00k
Timestamp      04/07/2025 04:25:34 PM
```

NOTE: There were 2878 observations read from the data set WORK.KIDDIS.

```
278
279 Proc Means Data=KidDis N Mean Std Min Max;
280     Var SurvTime;
281     Class Race Gender;
282     Title'Chronic Kidney Disease by Race and Gender';
```

283 Run;

NOTE: There were 2878 observations read from the data set WORK.KIDDIS.

NOTE: PROCEDURE MEANS used (Total process time):

real time	0.04 seconds
user cpu time	0.04 seconds
system cpu time	0.01 seconds
memory	8679.15k
OS Memory	62908.00k
Timestamp	04/07/2025 04:25:34 PM

284

285 Proc SGPanel Data=KidDis;

286 Panelby Race Gender;

287 Histogram SurvTime;

288 Title'Chronic Kidney Disease by Race and Gender';

289 Run;

NOTE: PROCEDURE SGPANEL used (Total process time):

real time	0.85 seconds
user cpu time	0.24 seconds
system cpu time	0.02 seconds
memory	5476.21k
OS Memory	57952.00k
Timestamp	04/07/2025 04:25:34 PM

NOTE: There were 2878 observations read from the data set WORK.KIDDIS.