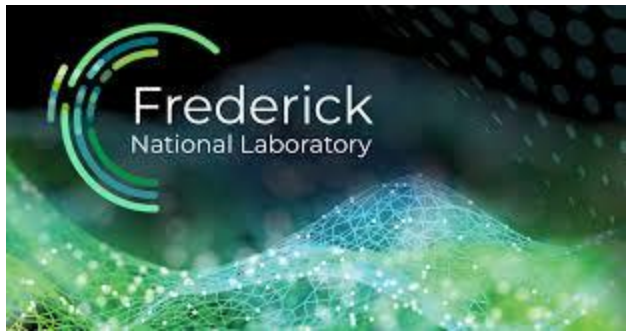


NIH NATIONAL CANCER INSTITUTE

Survival analysis in R with the “survival” package

Duncan Donohue, PhD

Duncan.donohue@nih.gov

Advanced Biomedical Computational Sciences (ABCS)/DMS/BRM*i*
Frederick National Laboratory for Cancer Research

BTEP Coding Club 15 January 2025

Learning Objectives

- Basics of survival analysis (recap of Alex Mitrophanov's talk)
  - https://bioinfo-abcc.ncifcrf.gov/training/series/statistics-for-lunch
- How to import and format data for survival analysis in R
  - Clinical data (from the web)
  - Basic research data (messy)
- Perform survival analysis in R
  - Kaplan-Meier - visualize
  - Log rank test – single categorical data
  - Cox regression
- Visualize and interpret survival analysis results in R

Basics of Survival Analysis

Definitions

- Event
  - Examples: Relapse, Progression, Death
  - Occurs at some probability/rate from initiating event (start of experiment, disease onset, diagnosis, surgery, etc.)
  - Time to event – start time does not have to be consistent
- Censoring (right)
  - Data collection ends before event of interest occurs
  - Really what makes survival analysis special
  - Left or interval censoring also possible

Methods

- Kaplan-Meier
  - Non-parametric
- Log-rank test
  - Nonparametric
  - Single categorical predictor
- Cox Proportional Hazard
  - Parametric (more assumptions)
  - Much more flexible

Setup and Data Wrangling

Packages

- “survival” for core survival analysis functionality
- “ggplot2” basic plotting
- “ggfortify” for plotting of survival fit objects
- “dplyer” and “dplyr” to pivot data from wide to long and other data wrangling
- “readxl” to read .xlsx files
- “readr” to read .tsv files
- “contsurvplot” to plot survival against continuous variables

- “npsurvSS” log rank test sample size determination
- “survminer” for fancier plotting

Raw data to “Surv” objects

- Data is generally not in the format that R packages expect
- Survival needs “time” as a continuous numerical
- Survival expects 0 for right censored and 1 for “event” (death)
  - There are other options including notation for left censoring

Formula notation for the survival package

- Response variable, to the left of the “~” needs to be a Surv() object
- Predictor variable(s) to the right of “~” are treated as factors
  - I would recommend still casting factors with as.factor() to avoid confusion
- Variables with spaces in the name must be fully referenced with data object name and quotes, not backslashes
  - E.g. ,my.data$”my var with spaces” instead of:

(`my var with spaces`, data=my.data)

Clinical Data

Clinical data example

- We will download a random example dataset from cBioPortal
  - https://www.cbioportal.org/datasets
  - Cerami et al. The cBio Cancer Genomics Portal: An Open Platform for Exploring Multidimensional Cancer Genomics Data. Cancer Discovery. May 2012 2; 401. PubMed.
  - Gao et al. Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal. Sci. Signal. 6, pl1 (2013). PubMed.
  - de Bruijn et al. Analysis and Visualization of Longitudinal Genomic and Clinical Data from the AACR Project GENIE Biopharma Collaborative in cBioPortal. Cancer Res (2023). PubMed.
- TCGA, 206 patients, adult soft tissue sarcoma

Re-introducing the cBioPortal Newsletter! Subscribe via LinkedIn or Google Groups

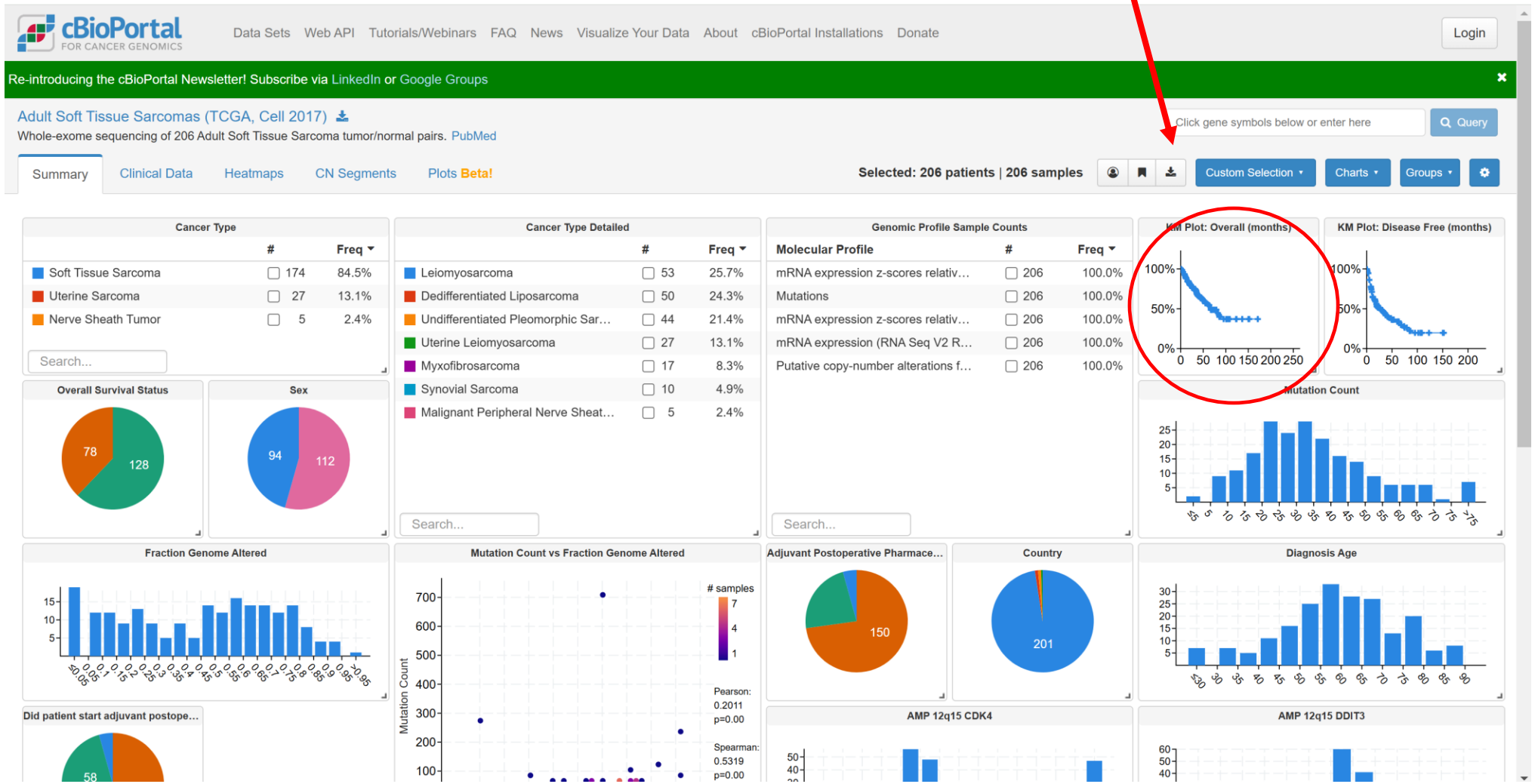
Datasets

The table below lists the number of available samples per cancer study and data type. It also provides links to download the data for each study. For alternative ways of downloading, see the Download Documentation.

Columns

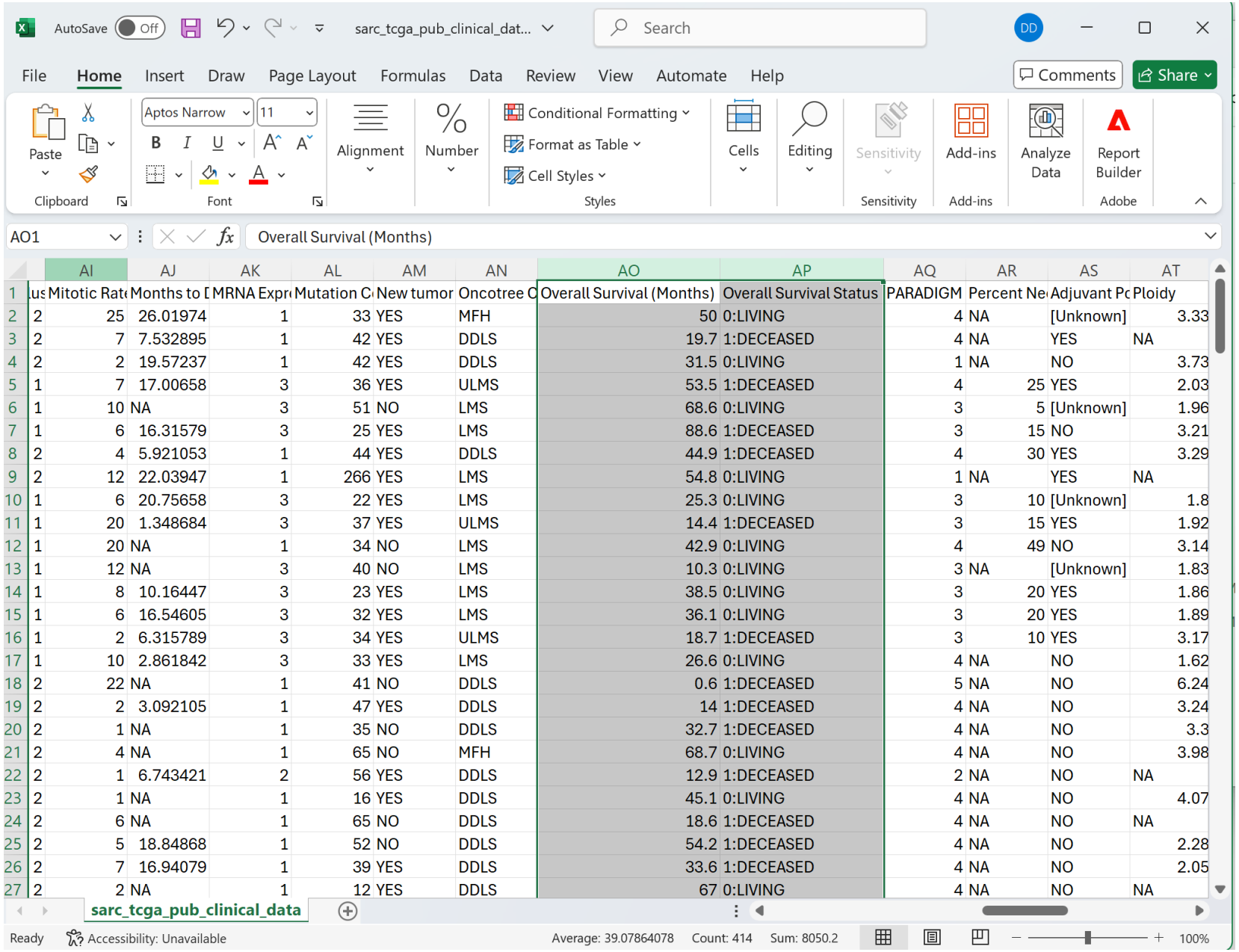
Name	Reference	All	Mutations	CNA	RNA-Seq
Acinar Cell Carcinoma of the Pancreas (JHU, J Pathol 2014)	Jial et al. J Pathol 2014	23	23	0	0
Acral Melanoma (TGEN, Genome Res 2017)	Liang et al. Genome Res 2017	38	38	38	36
Acute Leukemias of Ambiguous Lineage - Phase III (TARGET, GDC)		233	117	0	0
Acute Lymphoblastic Leukemia (St Jude, Nat Genet 2015)	Andersson et al. Nat Genet 2015	93	93	0	0
Acute Lymphoblastic Leukemia (St Jude, Nat Genet 2016)	Zhang et al. Nat Genet 2016	73	73	0	0
Acute Myeloid Leukemia (OHSU, Cancer Cell 2022)	Bottomly et al. Cancer Cell 2022	942	903	0	698
Acute Myeloid Leukemia (OHSU, Nature 2018)	Tyner et al. Nature 2018	672	622	0	451
Acute Myeloid Leukemia (TARGET, GDC)		2330	497	68	0
Acute Myeloid Leukemia (TCGA, Firehose Legacy)		200	197	191	173
Acute Myeloid Leukemia (TCGA, GDC)		200	72	190	0
Acute Myeloid Leukemia (TCGA, NEJM 2013)	TCGA, NEJM 2013	200	200	191	173
Acute Myeloid Leukemia (TCGA, PanCancer Atlas)	TCGA, Cell 2018	200	200	191	173
Acute myeloid leukemia or myelodysplastic syndromes (WashU, 2016)	Welch et al. N Engl J Med. 2016	136	136	0	0
Adenoid Cystic Carcinoma (FMI, Am J Surg Pathl. 2014)	Ross et al. Am J Surg Pathl 2014	28	28	28	0
Adenoid Cystic Carcinoma (JHU, Cancer Prev Res 2016)	Rettig et al, Cancer Prev Res 2016	25	25	0	0
Adenoid Cystic Carcinoma (MDA, Clin Cancer Res 2015)	Mitani et al. Clin Cancer Res 2015	102	65	0	0
Adenoid Cystic Carcinoma (MGH, Nat Gen 2016)	Drier et al. Nature Genetics 2016	10	10	0	0
Adenoid Cystic Carcinoma (MSK, Nat Genet 2013)	Ho et al. Nat Genet 2013	60	60	60	0
Adenoid Cystic Carcinoma (Sanger/MDA, JCI 2013)	Stephens et al. JCI 2013	24	24	0	0
Adenoid Cystic Carcinoma of the Breast (MSK, J Pathol. 2015)	Martelotto et al. J Pathol 2015	12	12	12	0
Adenoid Cystic Carcinoma Project (J Clin Invest 2019)	Allen et al. J Clin Invest 2019	1049	1049	928	0
Adrenocortical Carcinoma (TCGA, Firehose Legacy)		92	90	90	79

Download data: sarc_tcga_pub_clinical_data.tsv



Preview in Excel

- Lots of variables
- Overall Survival and Survival Status



	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT
1	lus	Mitotic Rate	Months to D	MRNA Expr	Mutation C	New tumor	Oncotree C	Overall Survival (Months)	Overall Survival Status	PARADIGM	Percent Ne	Adjuvant Pc	Ploidy
2	2	25	26.01974	1	33	YES	MFH	50	0:LIVING	4	NA	[Unknown]	3.33
3	2	7	7.532895	1	42	YES	DDLS	19.7	1:DECEASED	4	NA	YES	NA
4	2	2	19.57237	1	42	YES	DDLS	31.5	0:LIVING	1	NA	NO	3.73
5	1	7	17.00658	3	36	YES	ULMS	53.5	1:DECEASED	4	25	YES	2.03
6	1	10	NA	3	51	NO	LMS	68.6	0:LIVING	3	5	[Unknown]	1.96
7	1	6	16.31579	3	25	YES	LMS	88.6	1:DECEASED	3	15	NO	3.21
8	2	4	5.921053	1	44	YES	DDLS	44.9	1:DECEASED	4	30	YES	3.29
9	2	12	22.03947	1	266	YES	LMS	54.8	0:LIVING	1	NA	YES	NA
10	1	6	20.75658	3	22	YES	LMS	25.3	0:LIVING	3	10	[Unknown]	1.8
11	1	20	1.348684	3	37	YES	ULMS	14.4	1:DECEASED	3	15	YES	1.92
12	1	20	NA	1	34	NO	LMS	42.9	0:LIVING	4	49	NO	3.14
13	1	12	NA	3	40	NO	LMS	10.3	0:LIVING	3	NA	[Unknown]	1.83
14	1	8	10.16447	3	23	YES	LMS	38.5	0:LIVING	3	20	YES	1.86
15	1	6	16.54605	3	32	YES	LMS	36.1	0:LIVING	3	20	YES	1.89
16	1	2	6.315789	3	34	YES	ULMS	18.7	1:DECEASED	3	10	YES	3.17
17	1	10	2.861842	3	33	YES	LMS	26.6	0:LIVING	4	NA	NO	1.62
18	2	22	NA	1	41	NO	DDLS	0.6	1:DECEASED	5	NA	NO	6.24
19	2	2	3.092105	1	47	YES	DDLS	14	1:DECEASED	4	NA	NO	3.24
20	2	1	NA	1	35	NO	DDLS	32.7	1:DECEASED	4	NA	NO	3.3
21	2	4	NA	1	65	NO	MFH	68.7	0:LIVING	4	NA	NO	3.98
22	2	1	6.743421	2	56	YES	DDLS	12.9	1:DECEASED	2	NA	NO	NA
23	2	1	NA	1	16	YES	DDLS	45.1	0:LIVING	4	NA	NO	4.07
24	2	6	NA	1	65	NO	DDLS	18.6	1:DECEASED	4	NA	NO	NA
25	2	5	18.84868	1	52	NO	DDLS	54.2	1:DECEASED	4	NA	NO	2.28
26	2	7	16.94079	1	39	YES	DDLS	33.6	1:DECEASED	4	NA	NO	2.05
27	2	2	NA	1	12	YES	DDLS	67	0:LIVING	4	NA	NO	NA

Load .tsv file into R

```
#use full file path if not in working dir
in.file.sarc<-"sarc_tcga_pub_clinical_data.tsv"

#use read_tsv (readr package), not read.table which will put data in a dataframe and mess up the column names
data.sarc.orig<-read_tsv(in.file.sarc)

#examine results to make sure it looks how we want
head(data.sarc.orig)
dim(data.sarc.orig)
names(data.sarc.orig)
```

Basic Research Data

Data can be messy

- Spread over multiple files or worksheets
- Vital information is not always easily machine readable
  - Color coded, notes, lab conventions, etc.
- Often geared towards other analyses
- Batch effects and other concerns

Read the data into R

- in.file.basic<-"survival data.xlsx“
- in.sheet.basic<-"For R“
- #we know that batch 1 ended on day 14 and batch 2 ended on day 16
- last.day<-c(Run1=14,Run2=16)

- #read the worksheet into a tibble
- data.basic.oirg<-read_xlsx(in.file.basic,in.sheet.basic)

- #examine results to make sure it looks how we want
- head(data.basic.oirg)
- dim(data.basic.oirg)
- names(data.basic.oirg)

Wrangle data (general best practices)

- Don’t skimp on variable names
- Don’t skimp on comments
- Create new variables instead of overwriting existing ones
- Check results after each step

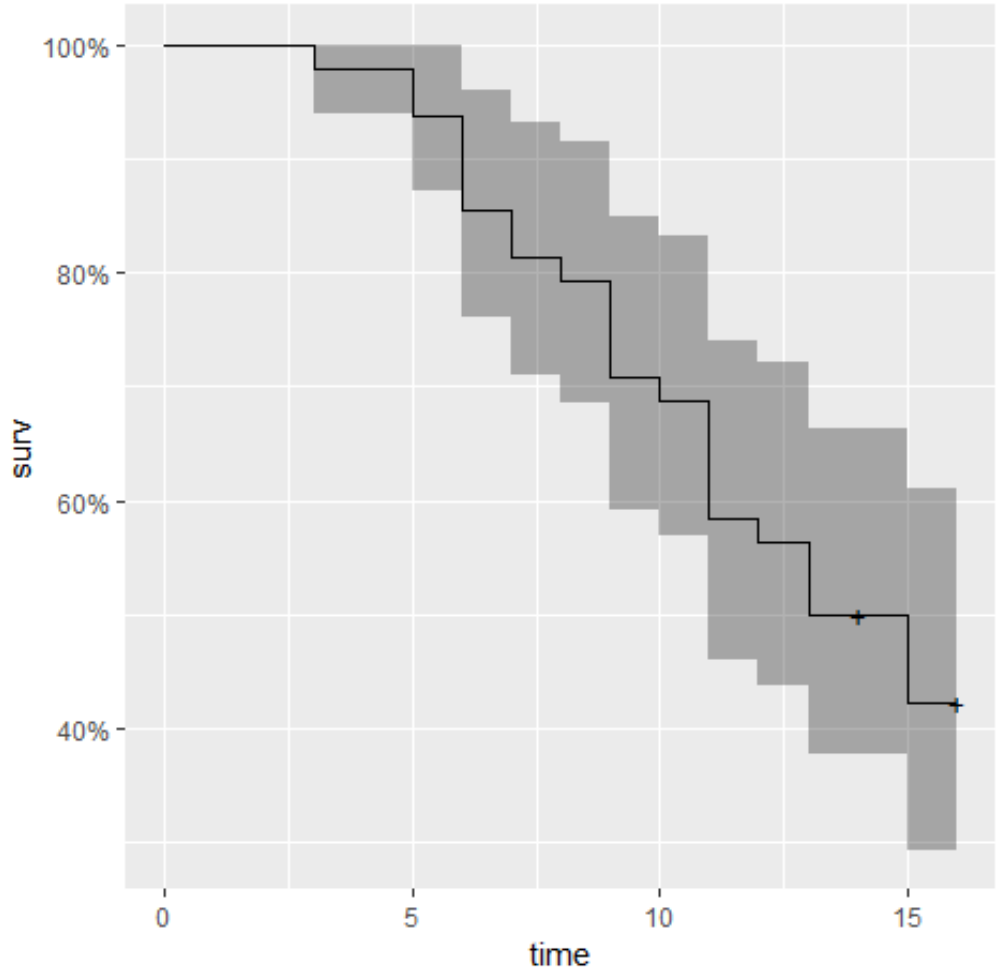
Visualizing KM Results

Fit Kaplan-Meier curves

```
km.basic <- survfit(Surv(time,event) ~ 1,data=data.basic.long.lastTime)
autoplot(km.basic)
summary(km.basic)
```

Call: survfit(formula = Surv(time, event) ~ 1, data = data.basic.long.lastTime)

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
3	48	1	0.979	0.0206	0.940	1.000
5	47	2	0.938	0.0349	0.871	1.000
6	45	4	0.854	0.0509	0.760	0.960
7	41	2	0.812	0.0563	0.709	0.931
8	39	1	0.792	0.0586	0.685	0.915
9	38	4	0.708	0.0656	0.591	0.849
10	34	1	0.687	0.0669	0.568	0.832
11	33	5	0.583	0.0712	0.459	0.741
12	28	1	0.562	0.0716	0.438	0.722
13	27	3	0.500	0.0722	0.377	0.663
15	13	2	0.423	0.0789	0.293	0.610



autoplot()

- Requires ggplot2 and ggfortify packages
- Kaplan-Meier survival curve, step plot (not smooth, not parametric)
- Describes the probability of survival at given time since initiating event for one or more groups

Add a categorical predictor variable (Group)

```
km.basic.group <-
survfit(Surv(time,event) ~
Group,data=data.basic.long.lastTime)

summary(km.basic.group)

summary(km.basic.group)$table

autoplot(km.basic.group)
```

summary() gives tables for each condition showing what changes at each time point where an event takes place

- Call: survfit(formula = Surv(time, event) ~ Group, data = data2.long.lastTime)

- Group=a
- time n.risk n.event survival std.err lower 95% CI upper 95% CI
- 3 12 1 0.917 0.0798 0.773 1.000
- 6 11 1 0.833 0.1076 0.647 1.000
- 11 10 2 0.667 0.1361 0.447 0.995

- Group=b
- time n.risk n.event survival std.err lower 95% CI upper 95% CI
- 5 12 1 0.917 0.0798 0.773 1.000
- 6 11 2 0.750 0.1250 0.541 1.000
- 7 9 1 0.667 0.1361 0.447 0.995
- 9 8 2 0.500 0.1443 0.284 0.880
- 10 6 1 0.417 0.1423 0.213 0.814
- 13 5 1 0.333 0.1361 0.150 0.742

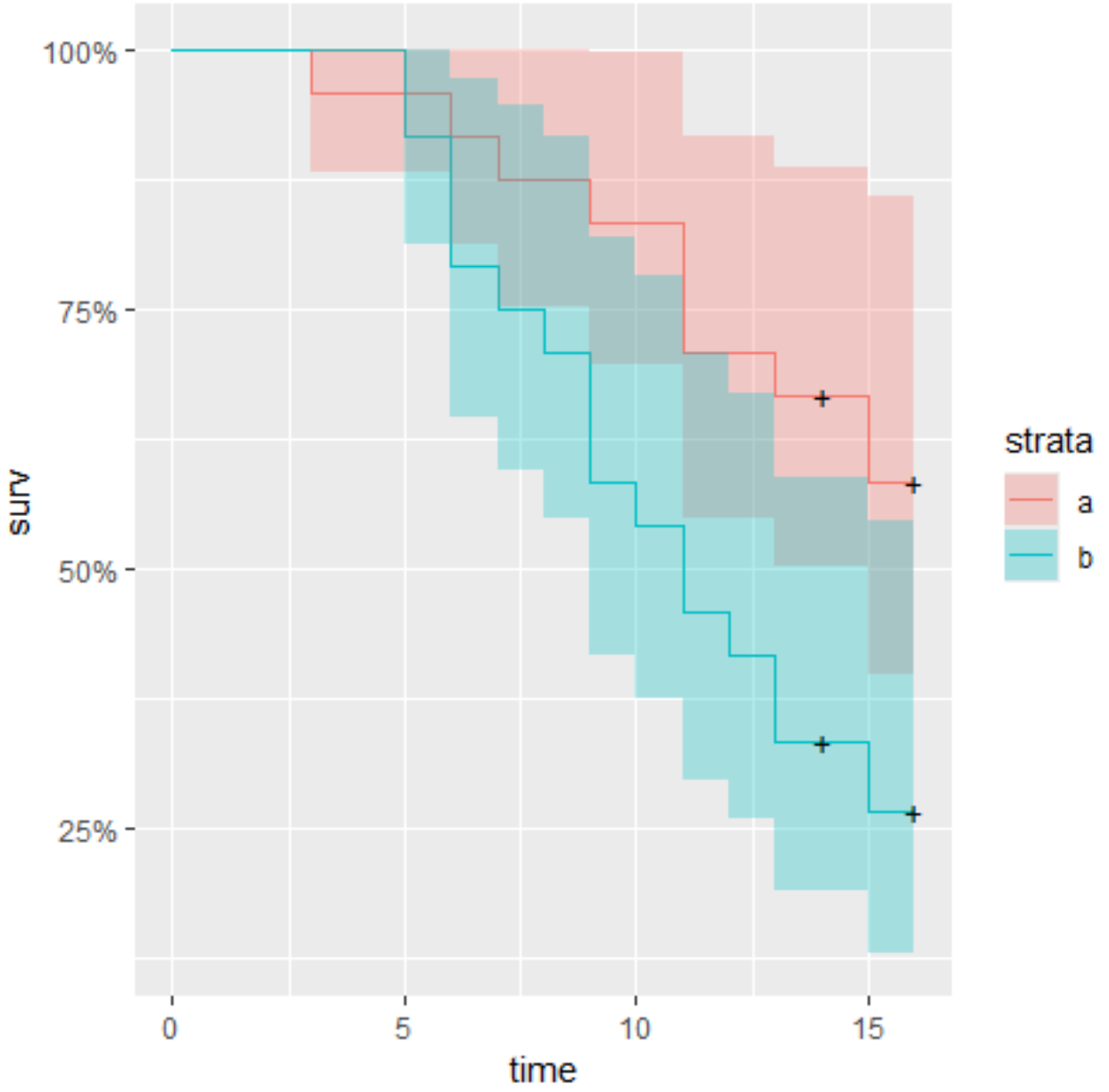
summary(km.basic.batch)$table

	records	n.max	n.start	events	rmean	se(rmean)	median	0.95LCL	0.95UCL
Batch=Run1	24	24	24	13	11.87500	0.8981317	13	10	NA
Batch=Run2	24	24	24	13	12.83333	0.7515416	15	11	NA

$table part of survfit object summarizes the results for each categorical group

Plots

- Time on x axis, “survival” on y
- Default shading is 95% confidence intervals, taken from “fit” object.
- Plus signs show right censoring
- Drops in lines show event(s)
- **https://cran.r-project.org/web/packages/ggfortify/vignettes/plot_surv.html**



What plot shows

- Probability of survival over time
- % still alive over time
- Questions to ask of the plot
  - Clear difference in groups (non overlapping CI)
  - Abrupt drops?
  - Censoring patterns
  - Confidence intervals
  - Hints about proportional hazard ratios (do the lines cross)

survminer

- ggsurvplot(fit, data = my.data)
- More easily allows inclusion of p-values, risk tables, etc. on the plot.
- https://rpkgs.datanovia.com/survminer/

Log Rank Test

Log rank test

- Compares survival functions, not curves
- Nonparametric
- 2 or more curves (categories of data)
- H0 = each curve from same dataset
- Need to use Holm’s correction for multiple pairwise
- Does not require proportional hazards
- One categorical predictor

Log rank test results

```
survdiff.group<-survdiff(Surv(time,event) ~ Group,data=data.basic.long.lastTime)
survdiff.group
Call:
survdiff(formula = Surv(time, event) ~ Group, data = data.basic.long.lastTime)
```

	N	Observed	Expected	(O-E)^2/E	(O-E)^2/V
Group=a	24	9	14.7	2.20	5.47
Group=b	24	17	11.3	2.86	5.47

Chisq= 5.5 on 1 degrees of freedom, p= 0.02

- Group A had less events than expected while group b had more.
- The two curves are unlikely to represent the same underlying data (p=.02)

Cox Regression

Cox Proportional Hazard regression

- Type of generalized liner model regression
- Output variable y is survival probability (log hazard ratio) in this case
  - Hazard ratio over baseline hazard
  - Does not change over time
  - Depends on predictors (x variables, right hand side of formula)
- Don’t need to know actual baseline hazard ratio for this to work

Cox **Proportional Hazard**

- “Proportional” means lines should not cross. If you are 2x likely to die at time a, you should also be 2x likely to die at time b.
- Hazard is prob of dying over short defined time interval
  - Rate function
- This can be checked with cox.zph() function
  - https://www.sthda.com/english/wiki/cox-model-assumptions
  - We want non-significant p-values to assure us that our hazards are proportional
  - Or. Plot log(-log(survival function)), look for parallel lines (=proportional)

Testing proportional hazards assumption

```
cox.zph(cox.basic.group)
```

	chisq	df	p
Group	0.00247	1	0.96
GLOBAL	0.00247	1	0.96

Results are not significant. We can assume that hazards are proportional.

Interpreting Cox model results

```
Call:
coxph(formula = Surv(time, event) ~ Group, data =
data.basic.long.lastTime)

  n= 48, number of events= 26

            coef exp(coef) se(coef)    z      Pr(>|z|)
Groupb 0.929    2.532    0.414     2.244    0.0248 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

      exp(coef) exp(-coef) lower .95 upper .95
Groupb    2.532     0.395    1.125      5.7

Concordance= 0.62  (se = 0.051 )
Likelihood ratio test= 5.35  on 1 df,   p=0.02
Wald test            = 5.04  on 1 df,   p=0.02
Score (logrank) test = 5.4  on 1 df,   p=0.02
```

- coef = regression coefficient. Positive means Group B has higher risk of death
- exp(coef) = hazard ratio (group b over 2x as likely to die as group 1)
- We get the 95% CI of this hazard ratio
- P-value of the difference in death risk between group a and group b is significant at 0.05 level
- Results match what we saw for log rank test

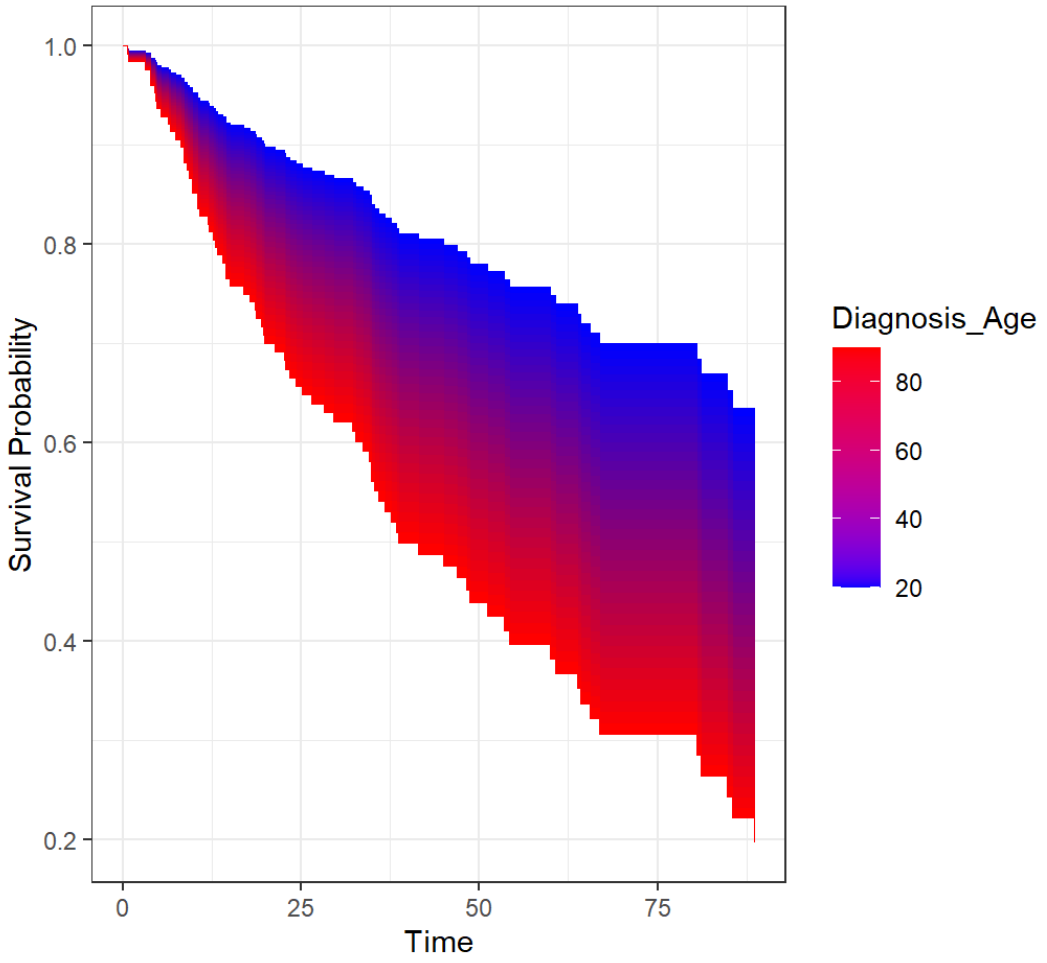
Continuous variables

```
data.sarc$Diagnosis_Age<-data.sarc$'Diagnosis Age' Call:

coxph(formula = Surv(time, event) ~ Diagnosis_Age, data = data.sarc,

    x = TRUE)

#plot_surv_area function also requries x=TRUE in coxph model

summary(cox.sarc.age2)
```

```
data.sarc$Diagnosis_Age<-data.sarc$'Diagnosis Age' Call:

coxph(formula = Surv(time, event) ~ Diagnosis_Age, data = data.sarc,

    x = TRUE)

  n= 206, number of events= 78

                    coef exp(coef) se(coef)     z Pr(>|z|)
Diagnosis_Age 0.017129  1.017277 0.008704 1.968   0.0491 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

              exp(coef) exp(-coef) lower .95 upper .95
Diagnosis_Age     1.017      0.983       1     1.035

Concordance= 0.565  (se = 0.041 )
Likelihood ratio test= 4  on 1 df,   p=0.05
Wald test            = 3.87  on 1 df,   p=0.05
Score (logrank) test = 3.87  on 1 df,   p=0.05
```

Visualize continuous variable (contsurvplot)

- plot_surv_area(data = data.sarc, model=cox.sarc.age2,time = "time", status = "event", variable = "Diagnosis_Age")



Cox proportional hazard regression

- Approach like you would any other regression problem
- One or more continues or categorical predictors depending on data and biological question
- Try different predictors, include those which (significantly) effect survival

Computational Science Support Available to NCI



- The Advanced Biomedical Computational Science (ABCS) group provides computational science support to the NCI community.

- ABCS computational science expertise includes bioinformatics, statistics, image analysis, protein/nucleotide modeling, AI, machine learning, etc.



- ABCS hosts weekly virtual office hours - opportunity to ask any questions and receive computational science support.
  - 12 – 1 p.m. every Wednesday on Teams.
  - Teams meeting information:

    Join the meeting now
    Meeting ID: 262 702 201 226
    Passcode: ko42PT

    Dial in by phone
    +1 240-800-7959,,249901770# United States, Bethesda
    Find a local number
    Phone conference ID: 249 901 770#

- Please reach out to natasha.pacheco@nih.gov with any additional questions.