DS 223 Marketing Analytics

Survival Analysis and CLV

Karen Hovhannisyan

AUA

Objective

- Intro to Survival Analysis
- Kaplan-Meier Estimate
- Cox Proportional Hazard
- Accelerated Failure Time Model
- Multi-State Survival
- CLV with Survival Analysis

Objective

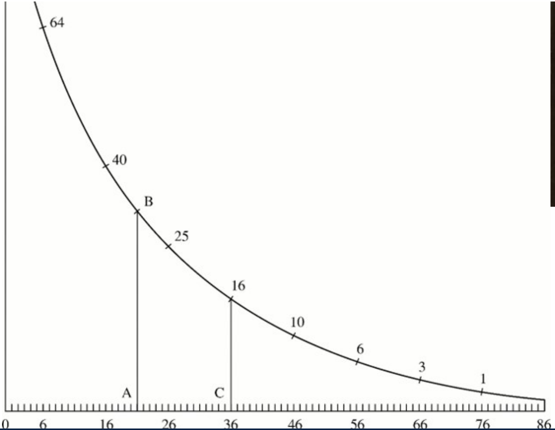
- Intro to Survival Analysis
- Kaplan-Meier Estimate
- Cox Proportional Hazard
- Accelerated Failure Time Model
- Multi-State Survival
- CLV with Survival Analysis

Intro to Survival Analysis

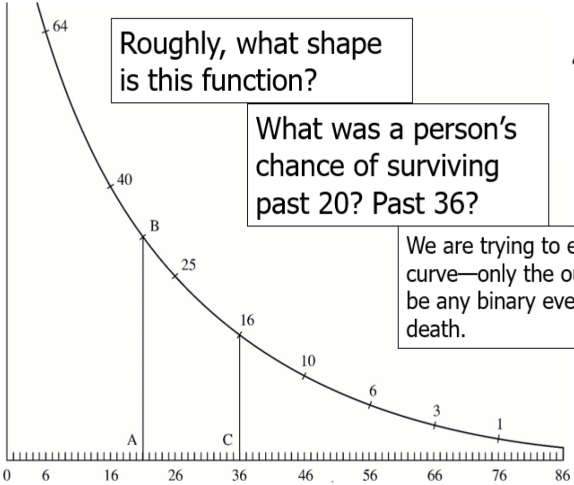
Origin

Christiaan Huygens's curve (in 1669)

The curve shows how many people out of 100 survive until 86 years.



Origin



Roughly, what shape is this function?

What was a person's chance of surviving past 20? Past 36?

We are trying to estimate this curve—only the outcome can be any binary event, not just death.

Death?

- Survival analysis is modeling of the time to death/event.
- Survival analysis has a much broader use in statistics. Any event can be defined as death:
  - age for marriage (!);
  - time for the customer to buy his first product after visiting the website for the first time;
  - time to attrition of an employee;

Applications

- **Business Planning:** profiling customers by survival rate and making respective strategies.
- **Lifetime Value Prediction:** engage with customers according to their lifetime value.
- **Active customers:** predict when the customer will be active for the next time and take interventions accordingly.
- **Campaign evaluation:** monitor the campaign's effect on customers' survival rate.
- **Employee attrition:** when will the employee leave?

Industry Specific Problems

- **Banking:** customer lifetime and time to mortgage redemption
- **Retail:** time to next purchase
- **Manufacturing:** a lifetime of a machine component
- **Telecom:** time to churn

Time to Event

Time to Event

Objectives of Survival Analysis | Medicine

- **Estimate time-to-event for a group of individuals:** time until the second heart attack for a group of MI patients.
- **To compare time-to-event between two or more groups:** treated vs. placebo MI (heart attack) patients in a randomized controlled trial.
- **To assess the relationship of co-variables to time-to-event:** does weight, insulin resistance, or cholesterol influence on survival time of patients?

Objectives of Survival Analysis | Marketing

- **Estimate time-to-event for a group of individuals:**
- **To compare time-to-event between two or more groups:**
- **To assess the relationship of co-variables to time-to-event:**

Objectives of Survival Analysis | Marketing

- **Estimate time-to-event for a group of individuals:** time until subscribers churn.
- **To compare time-to-event between two or more groups:**
- **To assess the relationship of co-variables to time-to-event:**

Objectives of Survival Analysis | Marketing

- **Estimate time-to-event for a group of individuals:** time until subscribers churn.
- **To compare time-to-event between two or more groups:** compare survival time between different subscribers with different education levels
- **To assess the relationship of co-variables to time-to-event:**

Objectives of Survival Analysis | Marketing

- **Estimate time-to-event for a group of individuals:** time until subscribers churn.
- **To compare time-to-event between two or more groups:** compare survival time between different subscribers with different education levels
- **To assess the relationship of co-variables to time-to-event:** does income, gender or age influence on survival time of subscribers?

Terms

- **Time-to-event:** the time from entry into a study until a subject has a particular outcome.

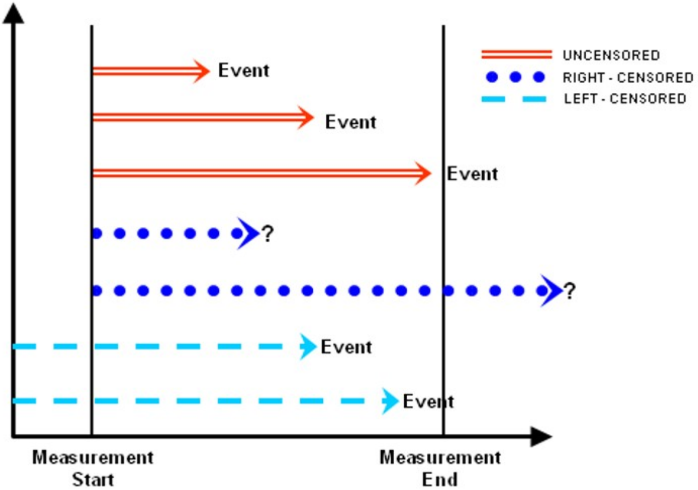
Terms

- **Time-to-event:** the time from entry into a study until a subject has a particular outcome.
- **Censoring:** subjects are said to be censored if they are lost to follow up or drop out of the study or if the study ends before they die or have an outcome of interest. They are counted as alive or disease-free for the time they were enrolled in the study.

Terms

- **Time-to-event:** the time from entry into a study until a subject has a particular outcome.
- **Censoring:** subjects are said to be censored if they are lost to follow up or drop out of the study or if the study ends before they die or have an outcome of interest. They are counted as alive or disease-free for the time they were enrolled in the study.
- **Hazard:** the event of interest occurring:*(death, churn, attrition etc.)*
  - *Hazard rate:* the instantaneous probability of the given event occurring at any point in time. It can be plotted against time on the X axis, forming a graph of the hazard rate over time.
  - *Hazard function:* the equation that describes plotted line in the next slide is the hazard function.

Censoring


UNCENSORED
RIGHT - CENSORED
LEFT - CENSORED
Event
Event
Event
?
?
Event
Event
Measurement
Start
Measurement
End

Hazard Function

Survival analysis focuses on the hazard function

Suppose a person is alive at the time t and that the probability of dying in the short time interval $(t, t + \Delta t)$ is $\lambda(t)\Delta t$

Then $\lambda(t)\Delta t$ is called a hazard function, sometimes it is given as $h(t)$

More precisely:

$$\lambda(t) = \frac{P(\text{Person dies by time } t + \Delta t | \text{Patient alive at time } t)}{\Delta t}$$

For a very large population:

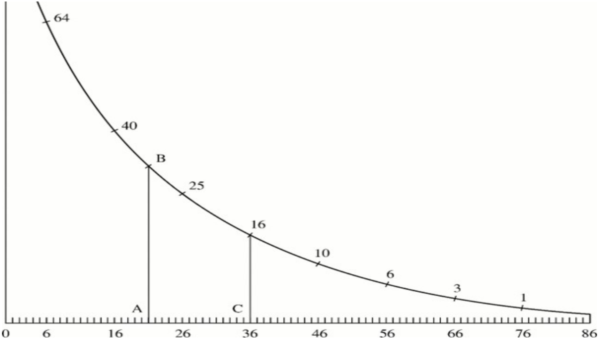
$$\lambda[t]\Delta t \cong \frac{\text{The number of deaths in the interval } (t, t + \Delta t)}{\text{Number of people alive at time } t}$$

Survival Curve

Let's try to find out what is the distribution of the survival curve?

One way to describe the survival distribution:

- P(T>76)=.01
- P(T>36)=.16

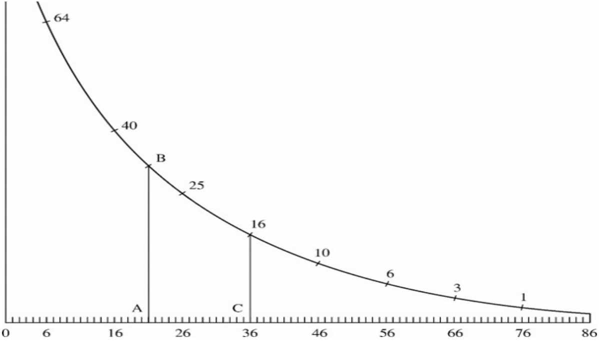


Survival Function

Assuming that the survival is an exponential distribution:

$$\Downarrow$$

If $T \sim exp(\lambda)$, then $P(T = t) = \lambda e^{-\lambda t}$
λ is a constant rate



Survival Function and Hazard

$\lambda(t) = 0.01$deaths per person quad: Incident rate (constant)

Probability of dying at age 10 $P(T = t) = \lambda e^{-\lambda t}$

$$\Downarrow$$

$$P(t = 10) = 0.1e^{-0.1\cdot(10)} = 0.0368$$

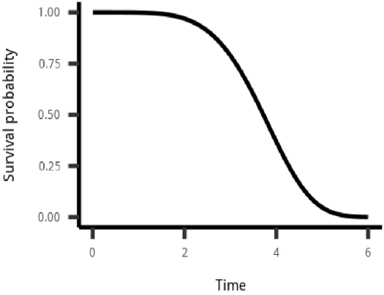
Probability of surviving past year 10 $P(T > t) = e^{-\lambda t}$

$$\Downarrow$$

$$S(t = 10) = e^{-0.1\cdot(10)} = 0.3679$$

Survival Function

$S(T) = 1 - F(T) = P(T > t)$
where $F(T)$ is the cumulative distribution function:



Survival: PDF, CDF

To sum up:

- Survival density function (PDF):

$$f(t) = \lim_{\Delta_t \to 0} \frac{P(t \leq T < t + \Delta t)}{\Delta t}$$

- Survival Cumulative Density Function (CDF):

$$F(t) = P(T \leq t) = \int_0^t f(u)du$$

- Survival Function:

$$S(t) = P(T > t) = 1 - F(t)$$

Hazard Function

$$h(t) = lim_{\Delta_t \to 0} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t}$$

The Hazard Function can be expressed as the probability density function divided by the survival function:

$$h(t) = \frac{f(t)}{S(t)}$$

In contrast to the survival function, which focuses on not failing, the hazard function focuses on failing, that is, on the event occurring.

Data Structure

Two-variable outcome:

- Time variable: $t_i =$ time at last disease-free observation or time at event
- Censoring variable: $c_i = 1$ if had the event; $c_i = 0$ no event by time t_i

Model Types

- **Non-parametric:** no assumption about the shape of the Hazard Function. It is estimated by empirical data:
  - Kaplan-Meier Estimate
- **Semi-parametric:** no assumption about the shape of the hazard function however, makes assumptions on the effect of covariates to Hazard Function:
  - Cox Regression
- **Parametric Model:** specifies the shape of the baseline hazard function and covariates effects on the hazard function:
  - Accelerated Failure Time (AFT) Model

Libraries

```r
libs<-c("ggplot2","dplyr","ggpubr","knitr","zoo","Hmisc","survival",
        "simsurv", "survminer","pec","SurvRegCensCov","flexsurv",
        "mstate","igraph","tidyr")

load_libraries<-function(libs){
  new_libs <- libs[!(libs %in% installed.packages()[,"Package"])]
  if(length(new_libs)>0) {install.packages(new_libs)}
  lapply(libs, library, character.only = TRUE)
}
load_libraries(libs)
```

Kaplan-Meier Estimate

What Is It?

- **The Kaplan-Meier** (K-M) procedure is a method of estimating time-to-event models in the presence of censored cases.
- A descriptive procedure for examining the distribution of time-to-event variables. We can also compare the distribution by levels of a factor variable or produce separate analyses by levels of a stratification variable.
- Censored cases (right-censored cases) are those for which the event of interest has not yet happened.

The Assumptions

- Probabilities for the event of interest should depend **only on time after the initial event without covariates (independent numeric variables) effects.**
- Cases that enter the study at different times (for example, patients who begin treatment at different times) should behave similarly.
- Censored and uncensored cases behave the same. If significant amount of the censored cases are patients with more serious conditions, your results may be biased.

The Estimator

Survival Estimator:

$$S(T) = P(T > t) = \prod_{t_i \leq T} \left(1 - \frac{d_i}{n_i}\right)$$

where:

- d_i: the number of deaths (events) at the time t_i
- n_i: the number of cases at risk at the time period t_i
- $S(t)$: the probability of surviving **at least** to time t

The Intuition

If the person has survived after period 3, then the person has survived at period 1, AND at period 2 AND at period 3, thus probability to survive after period 3:

$$P(T > 3) = P(1) \cdot P(2) \cdot P(3)$$

Survival Data

The variables we will model are:

1. **tenure**: the number of months the customer is with the company;
2. **churn**: if the customer has churned

```
load("Data/telco.Rda"); summary(telco)
    region          tenure            age                marital          address
 Zone 1:322   Min.   : 1.00    Min.   :18.00   Unmarried:505   Min.   : 0.00
 Zone 2:334   1st Qu.:17.00    1st Qu.:32.00   Married  :495   1st Qu.: 3.00
 Zone 3:344   Median :34.00    Median :40.00                   Median : 9.00
 Zone 4:   0  Mean   :35.53    Mean   :41.68                   Mean   :11.55
 Zone 5:   0  3rd Qu.:54.00    3rd Qu.:51.00                   3rd Qu.:18.00
              Max.   :72.00    Max.   :77.00                   Max.   :55.00

     income                                         ed          employ         retir
 Min.   :   9.00   Did not complete high school:204   Min.   : 0.00   No :9
 1st Qu.:  29.00   High school degree          :287   1st Qu.: 3.00   Yes:
 Median :  47.00   Some college                :209   Median : 8.00
 Mean   :  77.53   College degree              :234   Mean   :10.99
 3rd Qu.:  83.00   Post-undergraduate degree   : 66   3rd Qu.:17.00
 Max.   :1668.00                                      Max.   :47.00
```

Survival Data

```
head(telco)
  region tenure age   marital address income
1 Zone 2     13  44   Married       9     64               College d
2 Zone 3     11  33   Married       7    136    Post-undergraduate d
3 Zone 3     68  52   Married      24    116 Did not complete high s
4 Zone 2     33  33 Unmarried      12     33            High school d
5 Zone 2     23  30   Married       9     30 Did not complete high s
6 Zone 2     41  39 Unmarried      17     78            High school d
  employ retire gender reside tollfree equip callcard wireless longm
1      5     No   Male      2       No    No      Yes       No    3.
2      5     No   Male      6      Yes    No      Yes      Yes    4.
3     29     No Female      2      Yes    No      Yes       No   18.
4      0     No Female      1       No    No       No       No    9.
5      2     No   Male      4       No    No       No       No    6.
6     16     No Female      1      Yes    No      Yes       No   11.
  equipmon cardmon wiremon longten tollten equipten cardten wireten
1        0    7.50     0.0   37.45    0.00        0     110    0.00
2        0   15.25    35.7   42.00  211.45        0     125  380.35
```

Data Preparation

To run K-M estimates, first, we need to transform the indicator variable into numeric, with 0,1

```
telco$churn=ifelse(telco$churn=='Yes',1,0)
```

Call `survival` library and create **a survival object** with two parameters [time, event]

```
surv_obj=Surv(time=telco$tenure, event=telco$churn)
surv_obj[1:10]
 [1] 13  11  68+ 33  23+ 41+ 45  38+ 45+ 68+
```

Data Preparation

To run K-M estimates, first, we need to transform the indicator variable into numeric, with 0,1

```
telco$churn=ifelse(telco$churn=='Yes',1,0)
```

Call `survival` library and create **a survival object** with two parameters [time, event]

```
surv_obj=Surv(time=telco$tenure, event=telco$churn)
surv_obj[1:10]
 [1] 13  11  68+ 33  23+ 41+ 45  38+ 45+ 68+
```

+ sign indicates that the case was censored

Model | No Covariates

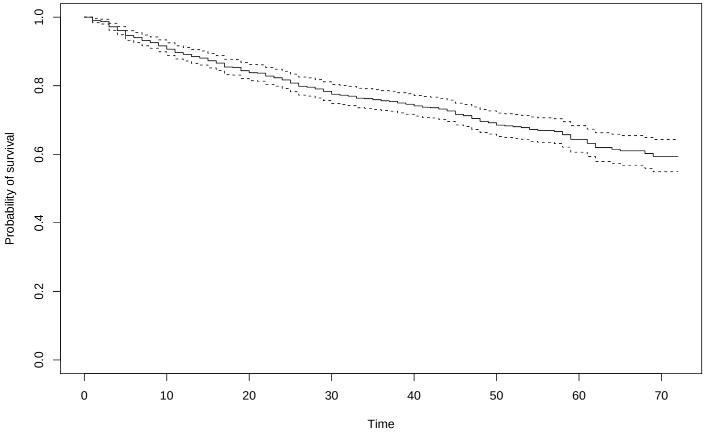
Create a Kaplan-Meier model with no covariates ($\sim$1)

```
km = survfit(surv_obj~1, data=telco)
```

Note how the formula is defined: `surv_object~1`, where 1 indicates that we are running a baseline model without **predictors**.

Survival Curve (1/3)

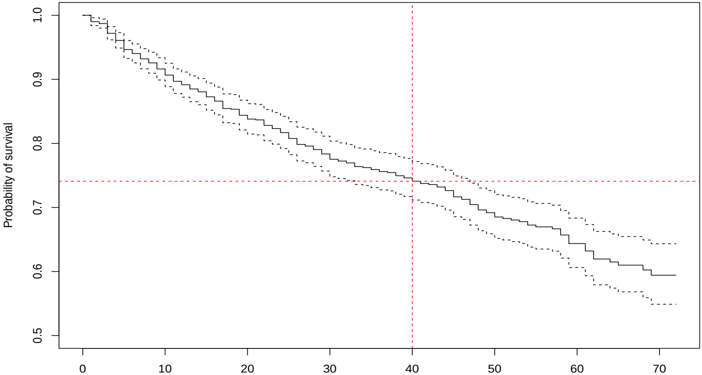
```
plot(km, xlab="Time", ylab="Probability of survival")
```



Survival Curve (2/3)

The probability of survival after month 40 is about 0.75.

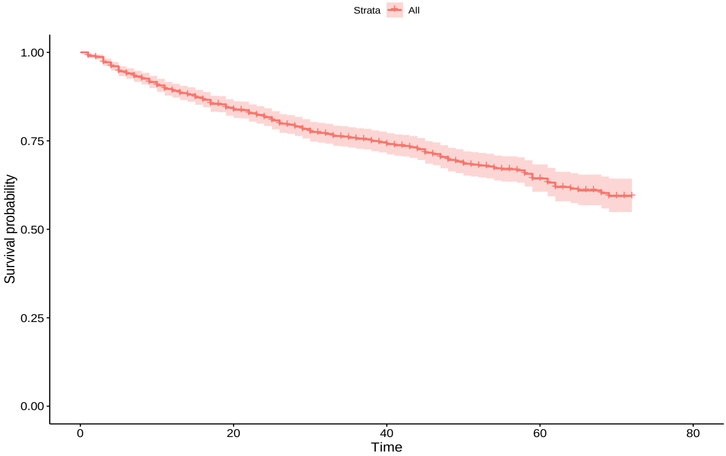
```
plot(km, xlab="Time", ylab="Probability of survival", ymin=0.5)
abline(v=40,lty=2,h = km$surv[which(km$time==40)],
       col=c("red","red"))
```



Survival Curve (3/3)

Survival curve with ggplot style with `survminer`:

```
ggsurvplot(km, data=telco)
```



Model Summary (1/3)

The probability that the customer will not leave the company after 4 months is 0.961

```
summary(km)
Call: survfit(formula = surv_obj ~ 1, data = telco)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1   1000      10    0.990 0.00315        0.984        0.996
    2    987       3    0.987 0.00358        0.980        0.994
    3    980      15    0.972 0.00524        0.962        0.982
    4    960      11    0.961 0.00616        0.949        0.973
    5    941      14    0.946 0.00716        0.933        0.961
    6    922       6    0.940 0.00754        0.926        0.955
    7    907       8    0.932 0.00802        0.916        0.948
    8    889       6    0.926 0.00837        0.909        0.942
    9    875       9    0.916 0.00886        0.899        0.934
   10    860       9    0.907 0.00933        0.888        0.925
   11    842       9    0.897 0.00977        0.878        0.916
   12    830       5    0.892 0.01001        0.872        0.911
   13    819       6    0.885 0.01028        0.865        0.905
   14    800       4    0.881 0.01047        0.860        0.901
   15    787       7    0.873 0.01079        0.852        0.894
   16    773       6    0.866 0.01105        0.845        0.888
   17    754      10    0.854 0.01149        0.832        0.877
   18    737       1    0.853 0.01153        0.831        0.876
   19    724       8    0.844 0.01187        0.821        0.867
   20    707       5    0.838 0.01209        0.815        0.862
   21    688       1    0.837 0.01213        0.813        0.861
   22    682       7    0.828 0.01243        0.804        0.853
```

Model Summary (2/3)

If you just run a table command, you will get the same results:

```
table(telco$tenure, telco$churn)[1:5,]; summary(km)


     0  1
  1  3 10
  2  4  3
  3  5 15
  4  8 11
  5  5 14
Call: survfit(formula = surv_obj ~ 1, data = telco)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1   1000      10    0.990 0.00315        0.984        0.996
    2    987       3    0.987 0.00358        0.980        0.994
    3    980      15    0.972 0.00524        0.962        0.982
    4    960      11    0.961 0.00616        0.949        0.973
    5    941      14    0.946 0.00716        0.933        0.961
    6    922       6    0.940 0.00754        0.926        0.955
    7    907       8    0.932 0.00802        0.916        0.948
    8    889       6    0.926 0.00837        0.909        0.942
```

Model Summary (3/3)

- You can access values from the model object **km**
- This will give you the survival probabilities

```
km$surv
 [1] 0.9900000 0.9869909 0.9718839 0.9607477 0.9464539 0.9402948 0.9
 [8] 0.9257109 0.9161893 0.9066013 0.8969107 0.8915077 0.8849765 0.8
[15] 0.8727195 0.8659455 0.8544608 0.8533014 0.8438727 0.8379047 0.8
[22] 0.8280991 0.8231030 0.8167617 0.8076722 0.7984642 0.7957982 0.7
[29] 0.7835508 0.7752004 0.7723917 0.7695363 0.7636954 0.7621950 0.7
[36] 0.7559656 0.7543741 0.7494649 0.7460889 0.7409078 0.7374129 0.7
[43] 0.7319027 0.7262291 0.7165718 0.7125686 0.7043781 0.6961156 0.6
[50] 0.6850622 0.6827161 0.6802951 0.6777661 0.6724710 0.6696922 0.6
[57] 0.6665918 0.6569311 0.6435924 0.6435924 0.6320308 0.6194739 0.6
[64] 0.6148510 0.6098522 0.6098522 0.6098522 0.6024150 0.5941627 0.5
[71] 0.5941627 0.5941627
```

Hazard vs Survival

Hazard is the inverse of survival:

$$S(T) = P(T > t) = \prod_{t_i \leq T} \left(1 - \frac{d_i}{n_i}\right)$$

where:

- $\frac{d_i}{n_i}$ is called **the instantaneous hazard function** $h(t_i)$
- Cumulative hazard at the time point t_i is the accumulated risk of failure (event to happen) at time t_i given that it didn't happen before.
- The cumulative hazard describes the accumulated risk up to the time t_i

Cumulative hazard is calculated with the following function:

$$H(T) = -In(S(T))$$

Hazard Function Interpretation (1/2)

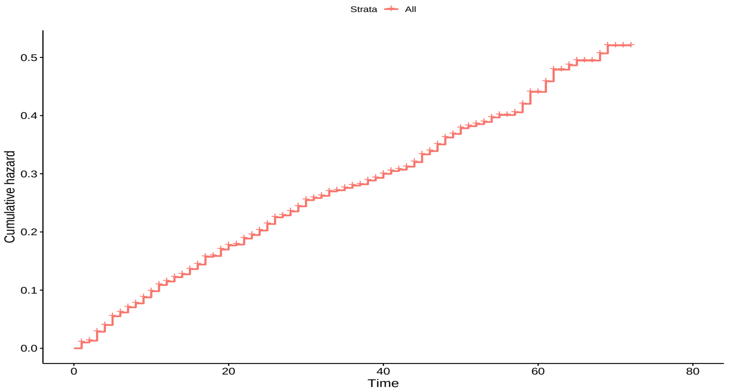
Suppose T denotes time from surgery for breast cancer until recurrence. Then when a patient who had received surgery visits her physician, she is interested in conditional probabilities such as:

"Given that I haven't had a recurrence yet, what are my chances of having one in the next year?"

Hazard Function Interpretation (2/2)

- Hazard Function: describes the probability of churn
- set `fun="cumhaz"`

```
ggsurvplot(km, fun="cumhaz", conf.int = F, pval = T)
```



Model with Covariates | Gender (1/3)

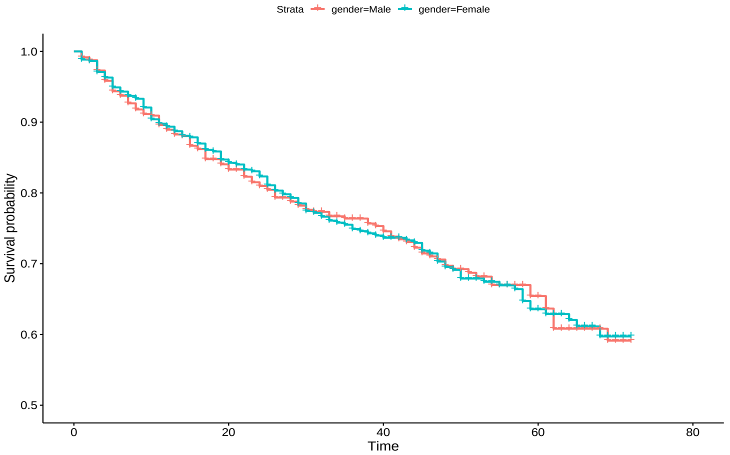
- With K-M estimates, we can also use covariates, thus creating survival curves for different groups.
- We want to see if survival is different by gender.

```
km_gender = survfit(surv_obj~gender, data=telco);
summary(km_gender)
Call: survfit(formula = surv_obj ~ gender, data = telco)

                gender=Male
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1    483       4    0.992 0.00412        0.984        1.000
    2    477       2    0.988 0.00505        0.978        0.998
    3    474       7    0.973 0.00739        0.959        0.988
    4    466       7    0.958 0.00912        0.941        0.976
    5    455       7    0.944 0.01054        0.923        0.965
    6    446       3    0.937 0.01109        0.916        0.959
    7    439       5    0.927 0.01195        0.903        0.950
    8    429       4    0.918 0.01259        0.894        0.943
    9    421       3    0.911 0.01306        0.886        0.937
   10    417       1    0.909 0.01321        0.884        0.935
   11    411       6    0.896 0.01408        0.869        0.924
   12    403       3    0.889 0.01449        0.861        0.918
   13    398       3    0.883 0.01489        0.854        0.912
   14    389       1    0.880 0.01503        0.851        0.910
   15    384       6    0.867 0.01581        0.836        0.898
   16    373       2    0.862 0.01606        0.831        0.894
   17    364       6    0.848 0.01681        0.815        0.881
```

Model with Covariate | Gender (2/3)

Plot the survival curves for males and females. Is there any difference?

```
ggsurvplot(km_gender, conf.int=F, ylim=c(0.5,1), pval = TRUE)
```



Model with Covariate | Gender (3/3)

In order to find out, let's test a hypothesis:

- H_0: survival curves are not different
- H_1: survival curves are different

```
survdiff(surv_obj~gender, data=telco)
Call:
survdiff(formula = surv_obj ~ gender, data = telco)

                N Observed Expected (O-E)^2/E (O-E)^2/V
gender=Male   483      131      131  0.000855   0.00165
gender=Female 517      143      143  0.000780   0.00165

 Chisq= 0  on 1 degrees of freedom, p= 1
```

Model with Covariate | Gender (3/3)

In order to find out, let's test a hypothesis:

- H_0: survival curves are not different
- H_1: survival curves are different

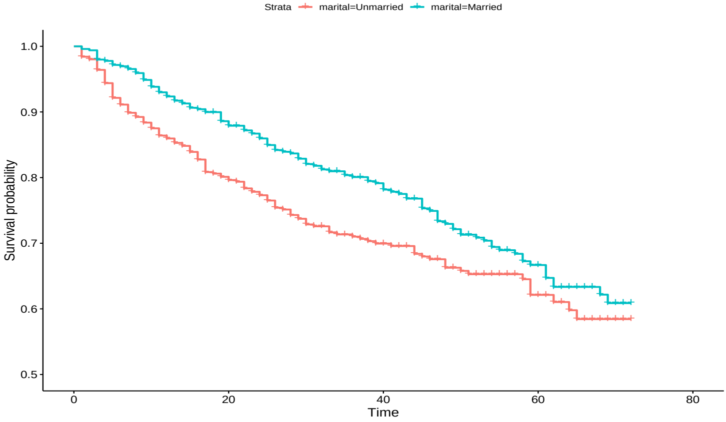
```
survdiff(surv_obj~gender, data=telco)
Call:
survdiff(formula = surv_obj ~ gender, data = telco)

                N Observed Expected (O-E)^2/E (O-E)^2/V
gender=Male   483      131      131  0.000855   0.00165
gender=Female 517      143      143  0.000780   0.00165

 Chisq= 0  on 1 degrees of freedom, p= 1
```

In this case, p-value is equal to 0.968, hence there is no statistically significant difference.

Model with Covariate | Marital Status (1/2)

Is survival different by marital status?

```
km_marital = survfit(surv_obj~marital, data=telco)
ggsurvplot(km_marital, conf.int = F, ylim=c(0.5,1))
```



Model with Covariate | Marital Status (2/2)

Do the hypothesis testing:

- H_0: Survival curves are not different
- H_1: Survival curves are different

```
survdiff(surv_obj~marital, data=telco)
Call:
survdiff(formula = surv_obj ~ marital, data = telco)

                    N Observed Expected (O-E)^2/E (O-E)^2/V
marital=Unmarried 505      147      127      3.23      6.09
marital=Married   495      127      147      2.78      6.09

 Chisq= 6.1  on 1 degrees of freedom, p= 0.01
```

Model with Covariate | Marital Status (2/2)

Do the hypothesis testing:

- H_0: Survival curves are not different
- H_1: Survival curves are different

```
survdiff(surv_obj~marital, data=telco)
Call:
survdiff(formula = surv_obj ~ marital, data = telco)

                    N Observed Expected (O-E)^2/E (O-E)^2/V
marital=Unmarried 505      147      127      3.23      6.09
marital=Married   495      127      147      2.78      6.09

 Chisq= 6.1  on 1 degrees of freedom, p= 0.01
```

In this case, p-value is equal to 0.014, hence there is a statistically significant difference.

Loading Packages

```
import lifelines
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Loading Data

```
telco = pd.read_csv('Data/telco.csv')
telco.head()
   tenure  churn  gender   marital
0      13      1    Male   Married
1      11      1    Male   Married
2      68      0  Female   Married
3      33      1  Female Unmarried
4      23      0    Male   Married
```

Model Fitting

```
kmf = lifelines.KaplanMeierFitter()
kmf.fit(telco['tenure'], telco['churn'])
<lifelines.KaplanMeierFitter:"KM_estimate", fitted with 1000 total o
```

Event Table

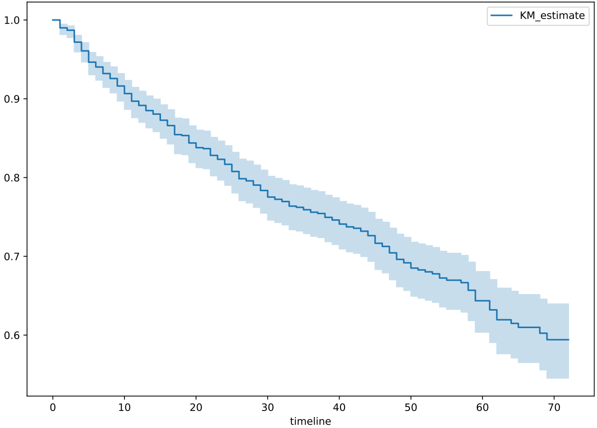
```
kmf.event_table
          removed  observed  censored  entrance  at_risk
event_at
0.0             0         0         0      1000     1000
1.0            13        10         3         0     1000
2.0             7         3         4         0      987
3.0            20        15         5         0      980
4.0            19        11         8         0      960
...           ...       ...       ...       ...      ...
68.0            9         1         8         0       82
69.0           14         1        13         0       73
70.0           11         0        11         0       59
71.0           17         0        17         0       48
72.0           31         0        31         0       31

[73 rows x 5 columns]
```

Kaplan-Meier estimate

```
kmf.survival_function_
          KM_estimate
timeline
0.0          1.000000
1.0          0.990000
2.0          0.986991
3.0          0.971884
4.0          0.960748
...               ...
68.0         0.602415
69.0         0.594163
70.0         0.594163
71.0         0.594163
72.0         0.594163

[73 rows x 1 columns]
```

Survival Curve

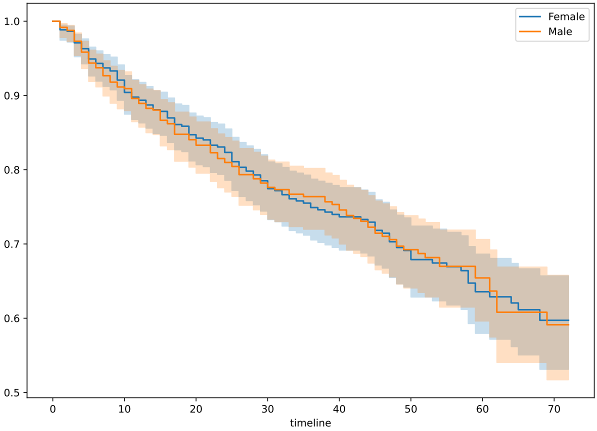
```
kmf.plot_survival_function()
```



Survival Curve by Gender (1/2)

```
ax = plt.subplot(111)
kmf = lifelines.KaplanMeierFitter()
for name, grouped_df in telco.groupby('gender'):
    kmf.fit(grouped_df["tenure"], grouped_df["churn"], label=name)
    kmf.plot_survival_function(ax=ax)
```

Survival Curve by Gender (2/2)



Cox Proportional Hazard Model

Cox Proportional Hazard Method

What do we need to know ?

- In Cox Regression, the dependent variable is the hazard.
- The central statistical output is the **hazard ratio**.
- The Cox Regression procedure is useful for modeling the time to a specified event, based upon the values of given **covariates**.
- One or more covariates are used to predict a status (event).
- Data contains censored and uncensored cases. Similar to logistic regression, but Cox Regression the relationship between survival time and covariates.

Cox Proportional Hazard Method

What do we need to know ?

- In Cox Regression, the dependent variable is the hazard.
- The central statistical output is the **hazard ratio**.
- The Cox Regression procedure is useful for modeling the time to a specified event, based upon the values of given **covariates**.
- One or more covariates are used to predict a status (event).
- Data contains censored and uncensored cases. Similar to logistic regression, but Cox Regression the relationship between survival time and covariates.

About hazard function:

- It is popular as you do not have to specify the hazard function
- Even if the hazard function is not specified, the functional form is completely specified

Let's see why?

Hazard Function

Hazard function:

$$h(t|X_i) = h_0(t)exp(\sum_{j=1}^{n} \beta_j X_j)$$

- X_i - n predictors for an individual
- β_j - the coefficient for the independent variable
- $h_0(t)$ - baseline hazard function: the hazard when all predictors are equal to zero
    - Baseline hazard is free from predictors
    - Exp part is free from time
- We assume that predictors are time-independent
    - Extended versions of Cox regression allow for time varying predictors
- It is a semi-parametric model:
    - We assume that predictors have effect on the hazard
    - But we don't have to specify the $h_0(t)$

The Intuition

The basic Cox Model assumes that the hazard functions for two different levels of a covariate are proportional for all values of t.

The Intuition

The basic Cox Model assumes that the hazard functions for two different levels of a covariate are proportional for all values of t.

$$\Downarrow$$

For example, if men has twice the risk of heart attack compared to women at age 50, they also have twice the risk of a heart attack at age 60, or any other age.

Hazard Ratio

Let's say we have two observations X_1, X_2. Then the Hazard ratio will be defined as:

$$HR = exp(\sum_{j=1}^{n} \beta_j (X_2 - X_1))$$

Proportional Hazard assumption: hazard ratio is independent of time:

$$\frac{h(t|X_2)}{h(t|X_1)} = \frac{h_0(t)exp(\sum_{j=1}^{n} \beta_j X_2)}{h_0(t)exp(\sum_{j=1}^{n} \beta_j X_1)} = exp(\sum_{j=1}^{n} \beta_j (X_2 - X_1))$$

Hazard Ratio

Let's say we have two observations X_1, X_2. Then the Hazard ratio will be defined as:

$$HR = exp(\sum_{j=1}^{n} \beta_j (X_2 - X_1))$$

Proportional Hazard assumption: hazard ratio is independent of time:

$$\frac{h(t|X_2)}{h(t|X_1)} = \frac{h_0(t)exp(\sum_{j=1}^{n} \beta_j X_2)}{h_0(t)exp(\sum_{j=1}^{n} \beta_j X_1)} = exp(\sum_{j=1}^{n} \beta_j (X_2 - X_1))$$

Note, the hazard ratio is the same for all time periods.

Coefficient Interpretation

Hazard Ratio $= exp(\beta)$, a unit increase in predictor variable increases the hazard by a factor of $exp(\beta)$.

Hazard Ratio signs:

- HR$=$1: Predictor has no affect on the hazard
- HR$>$1: Predictor increases hazard, thus decreasing the survival time
- HR$<$1: Predictor reduces hazard, thus increasing the survival time

Model Building

Let's build a Cox Regression model:

```
mod_cox = coxph(surv_obj~gender+age+ed, data=telco)
summary(mod_cox)
Call:
coxph(formula = surv_obj ~ gender + age + ed, data = telco)

  n= 1000, number of events= 274

                                coef exp(coef)  se(coef)       z
genderFemale               -0.053574  0.947836  0.121310  -0.442
age                        -0.062278  0.939622  0.005953 -10.462
edHigh school degree        0.243694  1.275954  0.217627   1.120
edSome college              0.452029  1.571498  0.221010   2.045
edCollege degree            0.786123  2.194870  0.206590   3.805
edPost-undergraduate degree 0.856556  2.355035  0.259568   3.300
                                        Pr(>|z|)
genderFemale                            0.658756
age                          < 0.0000000000000002 ***
```

Model Summary

The summary also gives the exponent of the coefficients:

```
summary(mod_cox)$rsq
      rsq    maxrsq
0.1540968 0.9705242
```

The *R-square* here is called a **pseudo R-square** and shows the improvement of the model with predictors compared to the baseline model (no predictors). Note, for pseudo R square the upper limit is not necessarily **1**. When the prediction is perfect, the maximum possible R-square is given in the output.

here:

- RSquare=0.154
- Max possible=0.971.

Interpretation of $Exp(\beta)$

	exp(coef)	exp(-coef)	lower .95	upper .95
genderFemale	0.9478	1.0550	0.7473	1.2022
age	0.9396	1.0643	0.9287	0.9506
edHigh school degree	1.2760	0.7837	0.8329	1.9547
edSome college	1.5715	0.6363	1.0190	2.4235
edCollege degree	2.1949	0.4556	1.4641	3.2905
edPost-undergraduate degree	2.3550	0.4246	1.4160	3.9169

- **genderFemale:** $exp(\beta) = 0.9478$. The baseline is *Male*, so Hazard of Female / Hazard of Male $= 0.9478$. Or Females have (1-0.9478) **5.2%** less chance to churn compared to males
- **Age:** $exp(\beta) = 0.9396$, one unit increase in age decreases the hazard by **6%**

Proportional or Not?

Checking proportional hazard assumption of a Cox Regression:

H_0 : Hazard rates are proportional
H_1 : Hazard rates are not proportional

```
test_z = cox.zph(mod_cox, global=T, transform="rank")
test_z
        chisq df    p
gender 0.0465  1 0.83
age    2.3668  1 0.12
ed     1.4518  4 0.84
GLOBAL 4.4176  6 0.62
```

Proportional Hazard Model

Let's build a simple model with two variables:

- address (how many years does the person live in the current address)
- retired (Yes/No)

```
mod_cox1 = coxph(surv_obj~retire+address, data=telco)
summary(mod_cox1)
Call:
coxph(formula = surv_obj ~ retire + address, data = telco)

  n= 1000, number of events= 274

               coef exp(coef)  se(coef)      z           Pr(>|z|)
retireYes -0.994543  0.369892  0.584628 -1.701             0.0889 .
address   -0.084569  0.918909  0.008621 -9.809 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

          exp(coef) exp(-coef) lower .95 upper .95
retireYes    0.3699      2.703    0.1176    1.1634
address      0.9189      1.088    0.9035    0.9346

Concordance= 0.698  (se = 0.014 )
Likelihood ratio test= 146  on 2 df,   p=<0.0000000000000002
Wald test            = 104  on 2 df,   p=<0.0000000000000002
Score (logrank) test = 113.6  on 2 df,   p=<0.0000000000000002
```

Proportional Hazard Assumption

The *Proportional Hazard* assumption holds for all the variables (address has borderline p-value)

```
cox.zph(mod_cox1)
        chisq df     p
retire  0.816  1 0.366
address 3.235  1 0.072
GLOBAL  4.513  2 0.105
```

Cox proportional Hazard Model

Now, let's predict the probability of churn given that a person is:

- retired
- living at current address for 9 years

Cox proportional Hazard Model

Now, let's predict the probability of churn given that a person is:

- retired
- living at current address for 9 years

1. Create a dataframe with the case
2. Use *survfit* with Cox model output as an argument for **the formula**
3. Look at what is inside the object

Cox proportional Hazard Model

Now, let's predict the probability of churn given that a person is:

- retired
- living at current address for 9 years

1. Create a dataframe with the case
2. Use *survfit* with Cox model output as an argument for **the formula**
3. Look at what is inside the object

```
case1 = data.frame(retire="Yes", address=9)
survivals = survfit(mod_cox1, newdata=case1)
names(survivals)
 [1] "n"          "time"       "n.risk"     "n.event"    "n.censor"   "su
 [7] "cumhaz"     "std.err"    "logse"      "lower"      "upper"      "co
[13] "conf.int"   "call"
```

Survival Output Data

Make a dataframe with Survival probabilities and Time variable.

```
df = data.frame(Probability=survivals$surv, Time=survivals$time)
head(df);tail(df)
```

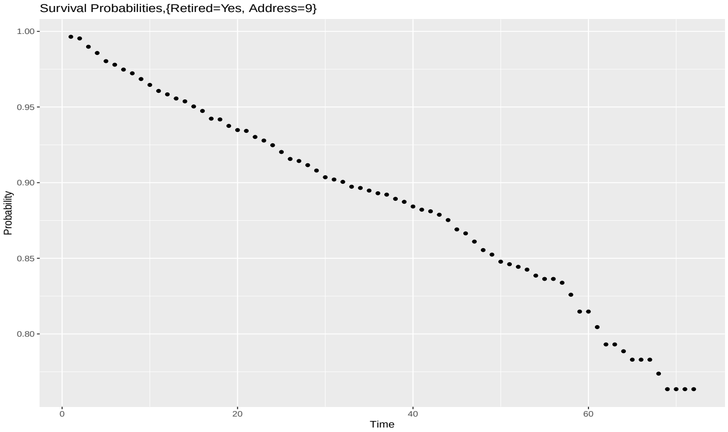
Probability	Time
0.9964267	1
0.9953411	2
0.9898337	3
0.9857048	4
0.9803075	5
0.9779540	6

	Probability	Time
67	0.7829748	67
68	0.7737420	68
69	0.7634804	69
70	0.7634804	70
71	0.7634804	71
72	0.7634804	72

Survival Curve

Survival plot

```
ggplot(df, aes(x=Time, y=Probability))+geom_point()+
  ggtitle("Survival Probabilities,{Retired=Yes, Address=9}")
```



Hazard and Cumulative Hazard | Throwback

Recall how we have defined hazard previously:

$$\lambda(t) = \frac{P(\text{Person dies by time } t + \Delta t | \text{Patient alive at time } t)}{\Delta t}$$

- We can think of hazard as a death rate for a very short time period
- As a rule, hazard is not probability, but still can be used as an indicator of risk

Recall how we have defined cumulative hazard (accumulated risk):

$$H(T) = -In(S(T))$$

Cumulative Hazard Calculation

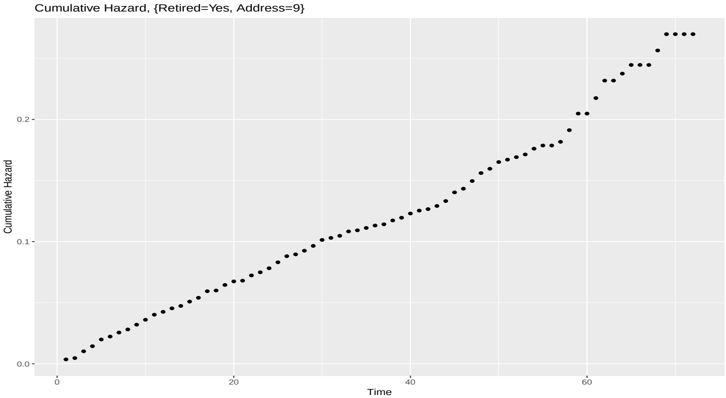
```
-log(survivals$surv)[1:10]
 [1] 0.003579654 0.004669782 0.010218280 0.014398390 0.019888940 0.0
 [7] 0.025578059 0.028110678 0.032019852 0.036026725
```

$$\Updownarrow$$

```
survivals$cumhaz[1:10]
 [1] 0.003579654 0.004669782 0.010218280 0.014398390 0.019888940 0.0
 [7] 0.025578059 0.028110678 0.032019852 0.036026725
```

Cumulative Hazard Plot

```
df1 = data.frame(Probability=survivals$cumhaz, Time=survivals$time)
ggplot(df1, aes(x=Time, y=Probability))+geom_point()+
  ggtitle("Cumulative Hazard, {Retired=Yes, Address=9}")+
  xlab("Time")+ylab("Cumulative Hazard")
```



Parametric Models: Accelerated Failure Time Model

Intuition

Here, we specify the *shape of the baseline hazard function* and *covariates' effects on the hazard function* **in advance**.
We are going to tryto fit the model by assuming some probabilistic distributions, which we are going to discuss throughout this section.

AFT

Regression formula (assuming exponential distribution for time to failure)

$$In(T) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \ldots$$

- Time-independent variables – gender, plan
- Time-dependent variables – age, inflation
- Need to specify the **distribution** assumed for the time/tenure variable

Options: Exponential

- Weibull
- Cauchy
- Normal distribution
- Others

This is called **Accelerated Failure Time Model**, in this case we are predicting the **survival time**.

AFT | Exponential (1/5)

Let's start to build an AFT model

- Start with the intercept-only model
- Define the distribution of the time (*exponential*)

```
reg_m = survreg(surv_obj~1, dist="exponential")
```

Complete list of the distributions you can find, by running this:
`names(survreg.distributions)`

```
names(survreg.distributions)
 [1] "extreme"     "logistic"    "gaussian"    "weibull"     "exponential"
 [6] "rayleigh"    "loggaussian" "lognormal"   "loglogistic" "t"
```

AFT | Exponential (2/5)

Predicting for 1

- Get predicted λ
- Vector of probabilities
- Question: why is newdata = data.frame(1)?

```
pred = predict(reg_m, type="response", newdata = data.frame(1))
pred
       1
129.6569
```

$\Downarrow$

```
probs =  seq(.1,.9,length=9)
probs
[1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
```

AFT | Exponential (3/5)

Recall for exponential distribution:

- $P(T = t) = \lambda e^{-\lambda t}$
- $P(T \leq t) = 1 - e^{-\lambda t}$

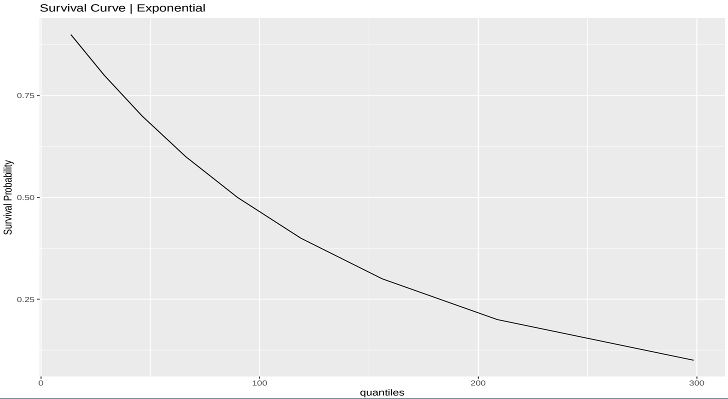
where:

- λ: rate of the event to happen,
- $E(x)$: equal to $\frac{1}{\lambda}$

AFT | Exponential (4/5)

Generate quantiles for the given probabilities using `qexp()` and plot the survival curve. *Look how rate is used.*

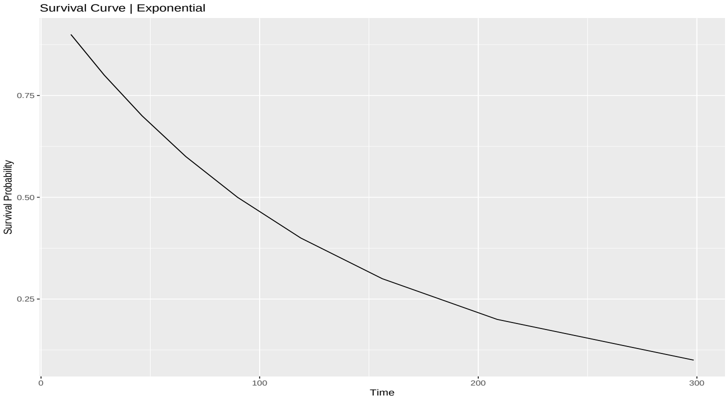
```
quantiles  = qexp(p = probs, rate = 1/pred, lower.tail = F)
df = data.frame(Probabilities=probs, quant=quantiles)
ggplot(df, aes(x = quantiles, y = probs)) + geom_line()+
  labs(title="Survival Curve | Exponential", y="Survival Probability")
```



AFT | Exponential (5/5)

Use `predict()` with `type=quantiles` to get the same result.

```
pred  =  predict(reg_m, type="quantile", p=1-probs, newdata=data.frame(1))
df  =  data.frame(Time=pred, Probabilities=probs)
ggplot(df, aes(x=Time, y=Probabilities))+geom_line()+
  labs(title="Survival Curve | Exponential", y="Survival Probability")
```



AFT with Covariates

Regression based on Gender:

```
reg_m  =  survreg(surv_obj~gender, data=telco, dist="exponential")
```

Interpreting the Coefficients

The interpretations are pretty close to *Logistic Regression!*

- The independent variable is gender
- The baseline category is Male
- Regression coefficient is "genderFemale 0.0087"
- Positive sign shows longer survival time and smaller hazard
- Negative sign shows shorter survival time and higher hazard

So when the gender is female the survival time is increased by 1-exp(0.0087) percent or 1.0087 times.

Model Evaluation

Is the predictor significant?

```
summary(reg_m)

Call:
survreg(formula = surv_obj ~ gender, data = telco, dist = "exponential")
               Value Std. Error     z                  p
(Intercept)  4.86034    0.08737 55.63 <0.0000000000000002
genderFemale 0.00869    0.12094  0.07               0.94

Scale fixed at 1

Exponential distribution
Loglik(model)= -1607   Loglik(intercept only)= -1607
    Chisq= 0.01 on 1 degrees of freedom, p= 0.94
Number of Newton-Raphson Iterations: 5
n= 1000
exp(summary(reg_m)$coefficients[2])
genderFemale
    1.008733
```

AFT with covariates | Exponential (1/9)

Regression on education level.
Again, is the predictor significant?

```
levels(telco$ed)
[1] "Did not complete high school" "High school degree"
[3] "Some college"                 "College degree"
[5] "Post-undergraduate degree"
```

```
reg_ed  =  survreg(surv_obj~ed, data=telco, dist="exponential")
summary(reg_ed)
```

	Value	Std..Error	z	p
(Intercept)	5.5242	0.1768	31.2496	0.0000
edHigh school degree	-0.3988	0.2171	-1.8372	0.0662
edSome college	-0.7329	0.2195	-3.3383	0.0008
edCollege degree	-1.1227	0.2052	-5.4703	0.0000
edPost-undergraduate degree	-1.1188	0.2588	-4.3234	0.0000

AFT with Covariates | Exponential (2/9)

Let's look at the exponents of the coefficients. Everything is compared to "Did not complete high school!"

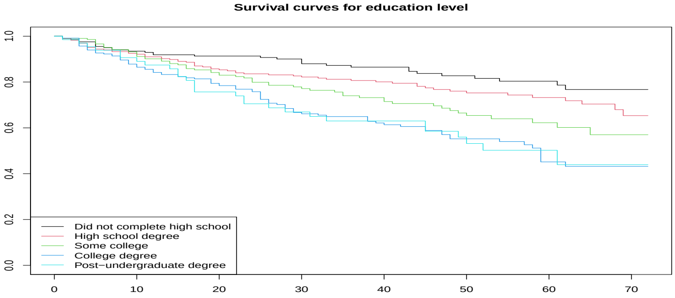
```
exp(coef(reg_ed))
```

	(Intercept)	edHigh school degree	edSome college	edCollege degree	edPost-undergraduate degree
expb	250.6875	0.6711068	0.4805091	0.3254095	0.3266731

AFT with Covariates | Exponential (3/9)

The same logic as with regression coefficients:

```
plot(survfit(surv_obj~ed, data=telco), conf.int = F, col=1:5,
     main="Survival curves for education level")
legend("bottomleft", legend=levels(telco$ed), col=1:5, lty=1)
```



AFT with Covariates | Exponential (4/9)

Let's build a model with **multiple** features.

What do you think??

```
reg_m = survreg(surv_obj~gender+ed+income, data=telco, dist="exponential")
summary(reg_m)

Call:
survreg(formula = surv_obj ~ gender + ed + income, data = telco,
    dist = "exponential")
                              Value Std. Error     z                 p
(Intercept)                 5.15708    0.19624 26.28 < 0.0000000000000002
genderFemale                0.00631    0.12128  0.05              0.95854
edHigh school degree       -0.42528    0.21737 -1.96              0.05041
edSome college             -0.83527    0.22026 -3.79              0.00015
edCollege degree           -1.21002    0.20558 -5.89          0.000000004
edPost-undergraduate degree -1.25597    0.25966 -4.84          0.000001318
income                      0.00585    0.00119  4.92          0.000000856

Scale fixed at 1
```

AFT with Covariates | Exponential (5/9)

- What about exponents of the coefficients?
- Are the values of education level and gender changed?

```
exp(reg_m$coefficients)
```

	Intercept	Female	High School Degree	Some College	College Degree	Post-Undergrad	Income
expb	173.657	1.006	0.654	0.434	0.298	0.285	1.006

AFT with Covariates | Exponential (6/9)

Predictions:

```
pred  =  predict(reg_m, type="response")
pred[1:10]
 [1]  75.31123 109.62523 344.56687 138.55209 206.98755 180.29849 127
 [8] 177.08032 136.81032 264.66589
```

Predicted average survival time:

$$In(T) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \dots$$

$$T = e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 \dots}$$

AFT with covariates | Exponential (7/9)

Predicting for the quantiles:

Case 1 (Row1): There is 10% chance that survival time is less than 7.9, 20% chance that survival time is less than 16.8, and so on...

```
probs  =  seq(0.1, .9, length=9)
pred  =  predict(reg_m, type="quantile", p=probs)
colnames(pred)  =  probs; head(pred, n=5)
          0.1      0.2       0.3       0.4       0.5       0.6       0.7
[1,]  7.93483 16.80522  26.86163  38.47091  52.20177  69.00699  90.67268
[2,] 11.55017 24.46216  39.10057  55.99938  75.98642 100.44858 131.98580
[3,] 36.30374 76.88787 122.89837 176.01359 238.83555 315.72343 414.84914
[4,] 14.59792 30.91701  49.41806  70.77596  96.03699 126.95399 166.81295
[5,] 21.80831 46.18794  73.82727 105.73454 143.47284 189.66077 249.20738
          0.8      0.9
[1,] 121.2088 173.4105
[2,] 176.4350 252.4214
[3,] 554.5590 793.3945
[4,] 222.9910 319.0280
[5,] 333.1336 476.6064
```

AFT with Covariates | Exponential (8/9)

Find probabilities of survival for the given values of t

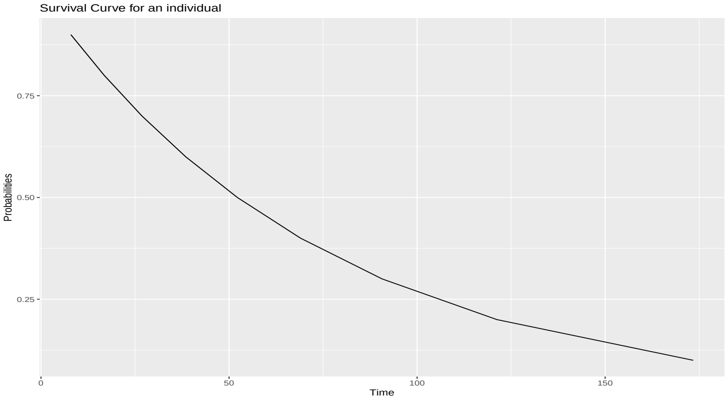
```
time = 1:72
pred_p  =  pexp(time, rate=1/pred[1])
pred_p[1:10]
 [1] 0.1184086 0.2227967 0.3148243 0.3959550 0.4674791 0.5305342 0.5
 [8] 0.6351296 0.6783334 0.7164215
```

Note that, $P(T \leq 1) = 0.118$

AFT with Covariates | Exponential (9/9)

Individual survival curve $P(T > t)$:

```
df  =  data.frame(Time=pred[1,], Probabilities=1-probs)
ggplot(df, aes(x=Time, y=Probabilities))+geom_line()+
  labs(title = "Survival Curve for an individual")
```



AFT | Weibull

So far we assumed that failure time has an *exponential* distribution, however, in practice, it can be something else.

As you have seen in previous slides `survreg()` allows other distributions as well and default one is **Weibull**.

- PDF:

$$f_X(x;\lambda,\gamma) = \begin{cases} \frac{\gamma}{\lambda}(\frac{x}{\gamma})^{\gamma-1}e^{-(x/\lambda)^{\gamma}} & ;\ x \geq 0 \\ 0 & ;\ x < 0 \end{cases}$$

where $\gamma > 0$ is called shape parameter and $\lambda > 0$ is called scale parameter

- CDF:

$$F(x;\lambda,k) = 1 - e^{(-(\frac{x}{\lambda})^{\gamma})}$$

- Expected Value:

$$E(x) = \lambda\Gamma(1 + \frac{1}{k})$$

AFT | Weibull | Parameters

- $\gamma < 1$ indicates that the failure rate decreases over time.
- $\gamma = 1$ indicates that the failure rate is constant over time. The Weibull distribution reduces to an exponential distribution.
- $\gamma > 1$ indicates that the failure rate increases with time.

In terms of churn:

- $\gamma < 1$ This means that churn decreases over time, implying higher loyalty.
- $\gamma = 1$ This means that the churn rate stays constant, and we fall back to our earlier exponential model.
- $\gamma > 1$ This means that churn increases over time, meaning you're more likely to unsubscribe or "fail" every new period.

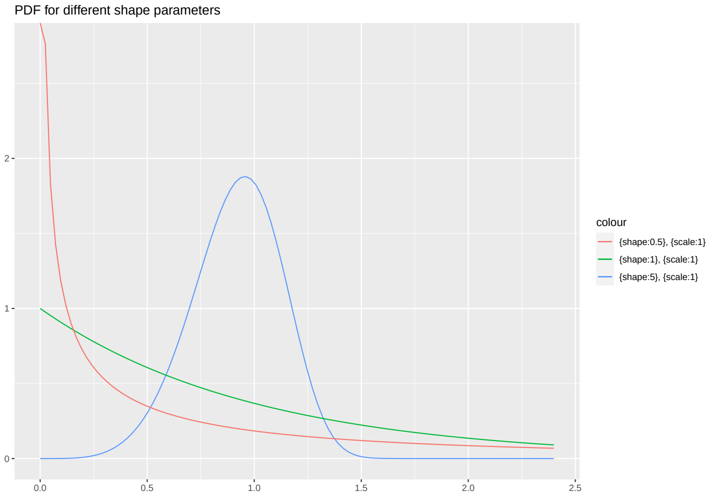
AFT | Weibull | PDF

Let's plot Weibull distribution with different *shape* values:

```
ggplot(data.frame(x = seq(0,2.5,0.2)), aes(x = x)) +
  geom_function(aes(colour="{shape:5}, {scale:1}"), fun = 'dweibull',
                args = list(shape = 5, scale = 1)) +
  geom_function(aes(colour="{shape:1}, {scale:1}"), fun = 'dweibull',
                args = list(shape = 1, scale = 1)) +
  geom_function(aes(colour="{shape:0.5}, {scale:1}"),fun = 'dweibull',
                args = list(shape = 0.5, scale = 1)) +
  labs(x = '', y = '', title = 'PDF for different parameters')
```

AFT | Weibull | PDF

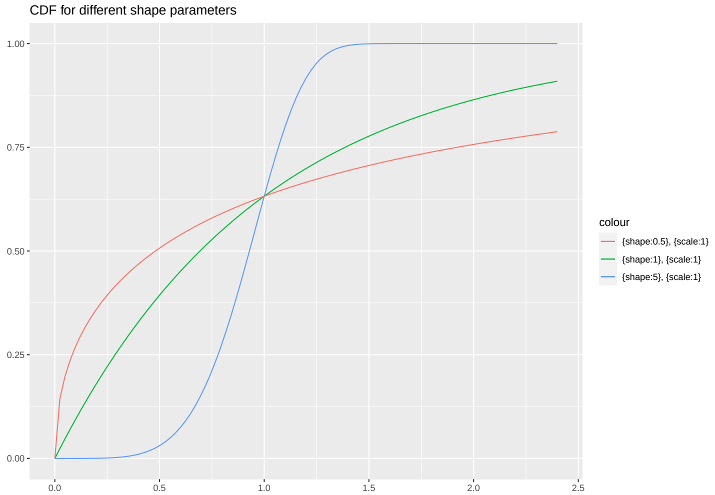
Let's see Weibull distribution with different *shape* values:



AFT | Weibull | CDF

```r
ggplot(data.frame(x = seq(0,2.5,0.2)), aes(x = x)) +
  geom_function(aes(colour="{shape:5}, {scale:1}"), fun = 'pweibull',
                args = list(shape = 5, scale = 1)) +
  geom_function(aes(colour="{shape:1}, {scale:1}"), fun = 'pweibull',
                args = list(shape = 1, scale = 1)) +
  geom_function(aes(colour="{shape:0.5}, {scale:1}"),fun = 'pweibull',
                args = list(shape = 0.5, scale = 1)) +
  labs(x = '', y = '', title = 'PDF for different shape parameters')
```

AFT | Weibull | CDF



AFT | Weibull | Survival

Survival function of Weibull distribution:

$$S(t) = e^{(-(\lambda t)^{\gamma})}$$

Hazard function:

$$h(t) = \gamma \lambda^{t} t^{(\gamma - 1)} e^{-(\lambda t)^{\gamma}}$$

Cumulative hazard:

$$H(t) = (\lambda t)^{\gamma}$$

AFT | Weibull | Data Generation

Generate dummy data using `simsurv` library

```
set.seed(1)
covariates = data.frame(id = 1:1000, gender = rbinom(1000, 1L, 0.5),
                        device = sample(1:3, 1000, replace=T))
s1 = simsurv(lambdas = 0.001, gammas =0.5,
             betas = c(gender = -0.5, device = 0.3),
             x = covariates, maxt = 15)
df = data.frame(covariates, s1[,-1])
df
       id gender device  eventtime status
1       1      0      3 15.0000000      0
2       2      0      3 15.0000000      0
3       3      1      1 15.0000000      0
4       4      1      3 15.0000000      0
5       5      0      1 15.0000000      0
6       6      1      2 15.0000000      0
7       7      1      1 15.0000000      0
8       8      1      2 15.0000000      0
9       9      1      2 15.0000000      0
10     10      0      2 15.0000000      0
```

AFT | Weibull | Model Building

Note, default value of `dist` is Weibull

```
reg_wb = survreg(Surv(df$eventtime, df$status)~1)
summary(reg_wb)

Call:
survreg(formula = Surv(df$eventtime, df$status) ~ 1)
             Value Std. Error    z        p
(Intercept) 10.975      2.815 3.90 0.000097
Log(scale)   0.563      0.333 1.69    0.091

Scale= 1.76

Weibull distribution
Loglik(model)= -74   Loglik(intercept only)= -74
Number of Newton-Raphson Iterations: 15
n= 1000
```

AFT | Weibull | Parameters

The distribution parametrization is different with `survreg`.
scale parameter:

```
sc = 1/reg_wb$scale
```

location parameter: $exp(-Intercept * k)$

```
exp(-reg_wb$coefficients*sc)
(Intercept)
0.001935446
```

Alternatively a `ConvertWeibull()` function from library `SurvRegCensCov`

```
ConvertWeibull(reg_wb)
$vars
          Estimate          SE
lambda 0.001935446 0.001183205
gamma  0.569241643 0.189518214
```

AFT with flexsurvreg | Weibull

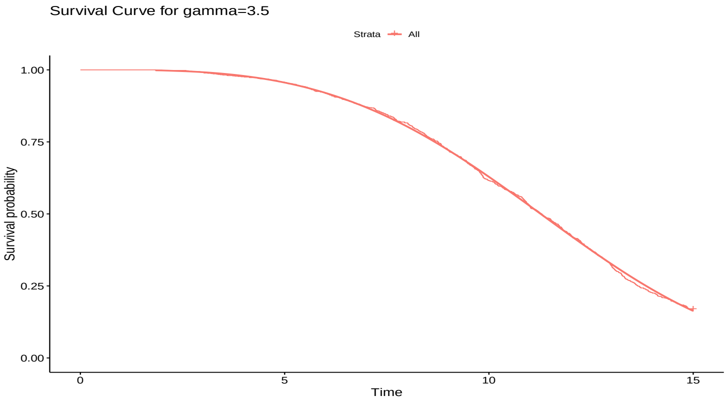
Regression with flexsurvreg

```
covariates = data.frame(id = 1:1000, gender = rbinom(1000, 1, 0.5),
                        device = sample(1:3, 1000, replace = T))
s1 = simsurv(lambdas = 0.0001, gammas =3.5,
             betas = c(gender = -0.5, device = 0.3),
             x = covariates, maxt = 15)
df = data.frame(covariates, s1[,-1])
wb_reg = flexsurvreg(Surv(df$eventtime, df$status)~1, data = df,
                     dist="weibull")
```

AFT with `flexsurvreg` | Weibull

Survival Curve

```
ggsurvplot(wb_reg, title = 'Survival Curve for gamma=3.5')
```



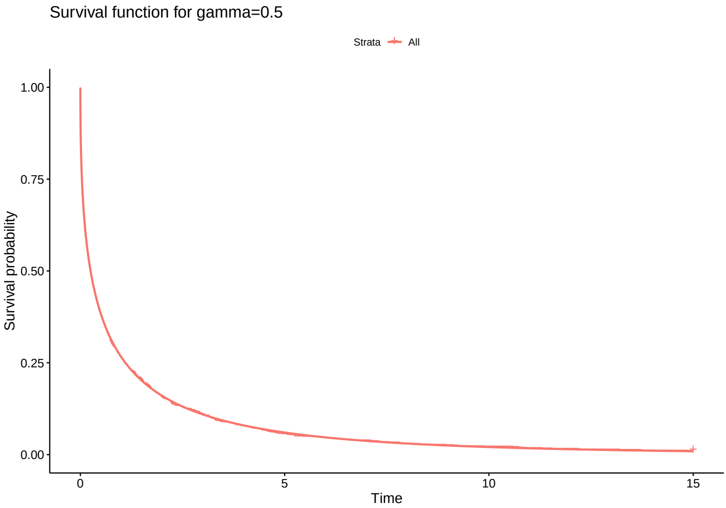
AFT with `flexsurvreg` | Weibull

Decrease the value of scale parameter ↓

```
s1 = simsurv(lambdas = 1, gammas =0.5,
             betas = c(gender = -0.5, device = 0.3),
             x = covariates, maxt = 15)
df = data.frame(covariates, s1[,-1])
wb_reg = flexsurvreg(Surv(df$eventtime, df$status)~1, data = df,
                     dist="weibull")
```

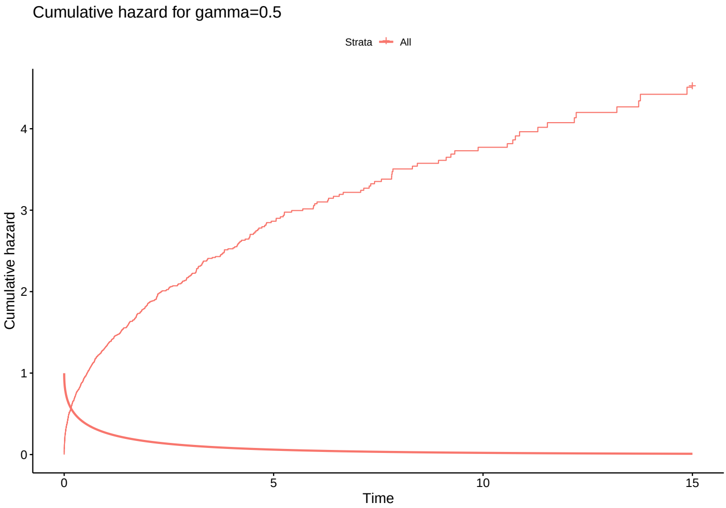
AFT with `flexsurvreg` | Weibull

```
ggsurvplot(wb_reg, title = 'Survival function for gamma=0.5')
```



AFT with `flexsurvreg` | Weibull

```
ggsurvplot(wb_reg, fun='cumhaz',title = 'Cumulative hazard for gamma=0.5')
```

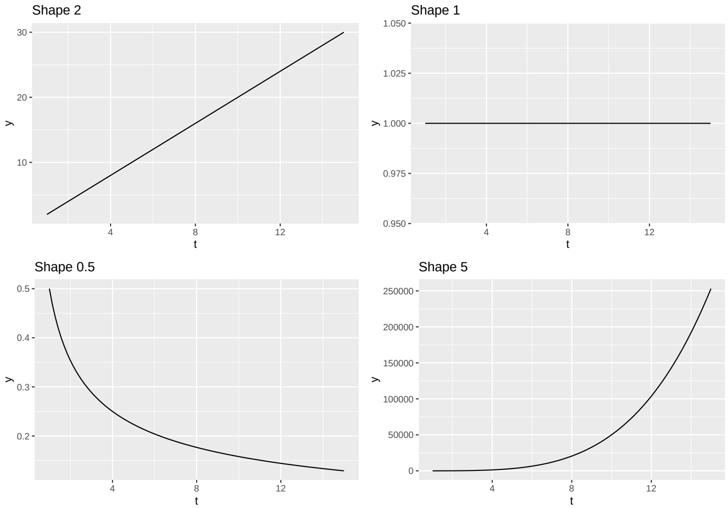


AFT | Weibull | Shapes

```
haz = function(t, gamma, lambda){gamma*(lambda^gamma)*t^(gamma-1)}

p = ggplot(data.frame(t = 1:15), aes(x = t))
p1 = p + geom_function(fun = 'haz', args = list(gamma = 2, lambda = 1)) +
  labs(title = "Shape 2")
p2 = p + geom_function(fun = 'haz', args = list(gamma = 1, lambda = 1)) +
  labs(title = "Shape 1")
p3 = p + geom_function(fun = 'haz', args = list(gamma = 0.5, lambda = 1)) +
  labs(title = "Shape 0.5")
p4 = p + geom_function(fun = 'haz', args = list(gamma = 5, lambda = 1)) +
  labs(title = "Shape 5")
plist = list(p1,p2,p3,p4)
```

AFT | Weibull | Shapes

```
ggarrange(plotlist = plist)
```



Summary

Sum Up

Let's try to sum up what we have covered so far:

- Kaplan-Meier (K-M) Estimate
- Cox Proportional Hazard Model
- Accelerated Failure Time Model

Comparing Survival Curves (1/2)

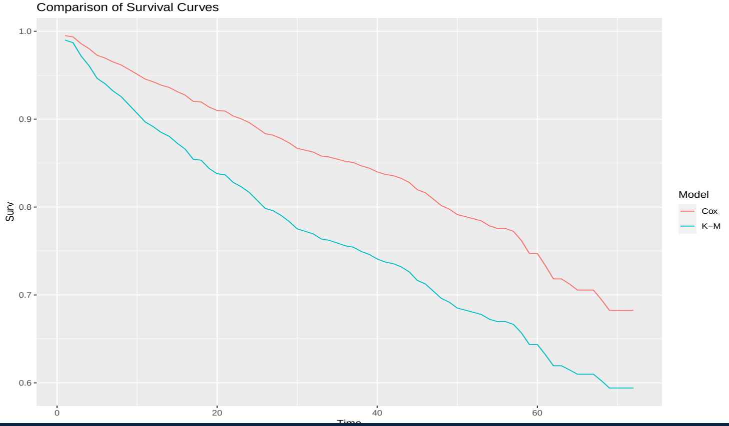
Let's create a dataframe by appending survival data from each model:

```
cox_model = coxph(surv_obj~gender+age+ed, data=telco)
cox_curve = survfit(cox_model)

km_data=data.frame(Time = km$time, Surv=km$surv, Model="K-M")
cox_data = data.frame(Time=cox_curve$time, Surv=cox_curve$surv,
                      Model="Cox")
survival_data = rbind(km_data, cox_data)
```

Comparing Survival Curves (2/2)

What about AFT model??

```
ggplot(data = survival_data, aes(x = Time, y = Surv, color=Model))+
  labs(title = "Comparison of Survival Curves")+
  geom_line()
```



Multi-State Survival Analysis

Two State Survival Analysis

So far, we have looked at the situations where the customer is either dead or alive.

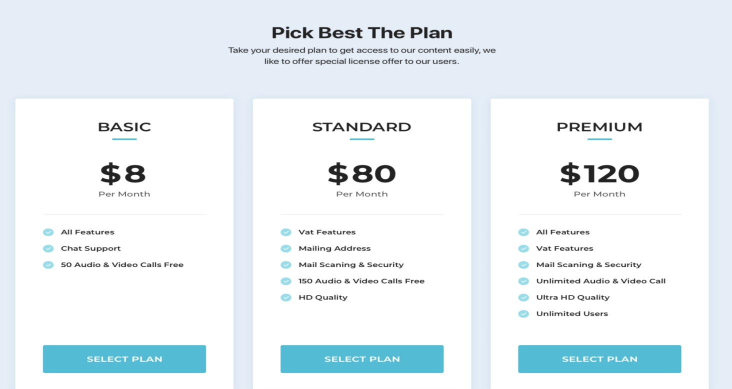
$$Subscribed(alive) \rightarrow Churned(dead)$$

Multi-state churn

There can be cases when there are more than two options. Thus the customer can be in more than two states.

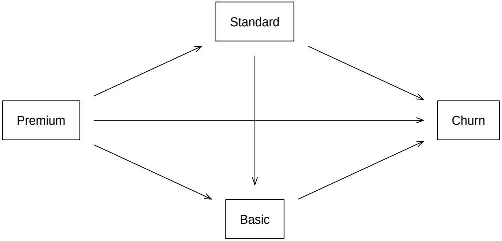
Here, customer could be at following states:

$$\text{Premium} \rightarrow \text{Standard} \rightarrow \text{Basic} \rightarrow \text{Churned}$$



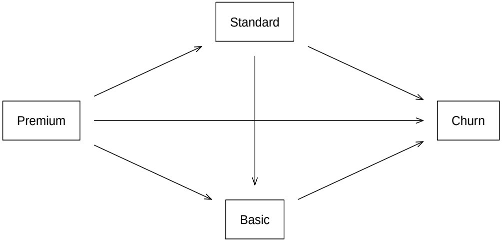
State Diagram

All the possible states that customer could be, during his/her entire relationship with the company.



State Diagram

All the possible states that customer could be, during his/her entire relationship with the company.



Data Requirments

- Customer ID
- Time Periods $t_1, t_2, ..., t_n$
- Features
- Transition state

Data Structure

Basically, we need to transform data in a way, which will be similar to bellow structure:

Id	Time1	Event1	Time2	Event2	Time3	Event3
1	10	0	16	1	16	1
2	20	1	20	0	20	1

Multi-State Data

Let's demonstrate it by using some subscription based business data.

```
msdata=read.csv("data/mstate.csv")
```

```
head(msdata)
```

id	st2	st2.s	st3	st3.s	st4	st4.s	st5	st5.s	year	age	discount	gender
1	22	1	995.0	0	995	0	995	0	2013-2017	20-40	no	male
3	1264	0	27.0	1	1264	0	1264	0	2013-2017	20-40	no	male
6	33	1	27.0	1	33	1	1427	0	2013-2017	20-40	no	male
7	29	1	28.5	1	29	1	775	1	2013-2017	>40	no	male
8	31	1	1618.0	0	1618	0	1618	0	2013-2017	20-40	no	male
9	87	1	29.0	1	87	1	1111	0	2013-2017	20-40	no	female

Multi-State Data

Let's demonstrate it by using some subscription based business data.

```
msdata=read.csv("data/mstate.csv")
```

```
head(msdata)
```

id	st2	st2.s	st3	st3.s	st4	st4.s	st5	st5.s	year	age	discount	gender
1	22	1	995.0	0	995	0	995	0	2013-2017	20-40	no	male
3	1264	0	27.0	1	1264	0	1264	0	2013-2017	20-40	no	male
6	33	1	27.0	1	33	1	1427	0	2013-2017	20-40	no	male
7	29	1	28.5	1	29	1	775	1	2013-2017	>40	no	male
8	31	1	1618.0	0	1618	0	1618	0	2013-2017	20-40	no	male
9	87	1	29.0	1	87	1	1111	0	2013-2017	20-40	no	female

Where is state 1?

Steps with `mstate` library

1. Transition Matrix: `transMat()`
2. Data Preparation:
    - `msprep()` for encoding
    - `expand.covs()` for expanding covariates
3. Model Building: `coxph()`

Transition Matrix

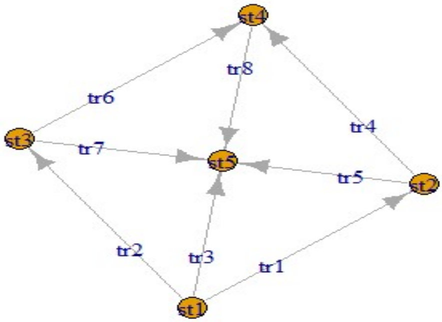
Creating an object for transitions by using `transMat()`

```
trans_mat <-
  transMat(names = c('st1', 'st2', 'st3', 'st4', 'st5'),
           x = list(c(2,3,5),
                    c(4,5),
                    c(4,5),
                    c(5),
                    c()))
kable(trans_mat, format = "latex")
```

	st1	st2	st3	st4	st5
st1	NA	1	2	NA	3
st2	NA	NA	NA	4	5
st3	NA	NA	NA	6	7
st4	NA	NA	NA	NA	8
st5	NA	NA	NA	NA	NA

Graph view

All the possible customer journeys initialized in `trans_mat` object.



Data sration | Encoding

Data preparation is the most challenging part in multistate models. Thus, working with multistate models, **pay attention to data structure requirements for each library/package**.

Using `msprep()` function to encode data for model fitting.

```
msdata_enc =
  msprep(
    data  = msdata,
    trans = trans_mat,
    time  = c(NA,   "st2",   "st3",   "st4",   "st5"),
    status= c(NA, "st2.s", "st3.s", "st4.s", "st5.s"),
    keep  = c('year', 'age', 'discount', 'gender')
  )
```

Data Preperation | Encoding

Let's see:

```
msdata_enc[msdata_enc$id %in% c(1, 3, 6, 1909), c(1:12)]
```

	id	from	to	trans	Tstart	Tstop	time	status	year	age	discount	gender
1	1	1	2	1	0	22	22	1	2013-2017	20-40	no	male
2	1	1	3	2	0	22	22	0	2013-2017	20-40	no	male
3	1	1	5	3	0	22	22	0	2013-2017	20-40	no	male
4	1	2	4	4	22	995	973	0	2013-2017	20-40	no	male
5	1	2	5	5	22	995	973	0	2013-2017	20-40	no	male
11	3	1	2	1	0	27	27	0	2013-2017	20-40	no	male
12	3	1	3	2	0	27	27	1	2013-2017	20-40	no	male
13	3	1	5	3	0	27	27	0	2013-2017	20-40	no	male
14	3	3	4	6	27	33	6	1	2013-2017	20-40	no	male
15	3	3	5	7	27	33	6	0	2013-2017	20-40	no	male
16	3	4	5	8	33	1427	1394	0	2013-2017	20-40	no	male
28	6	1	2	1	0	29	29	0	2013-2017	20-40	no	female
29	6	1	3	2	0	29	29	1	2013-2017	20-40	no	female
30	6	1	5	3	0	29	29	0	2013-2017	20-40	no	female
31	6	3	4	6	29	87	58	1	2013-2017	20-40	no	female
32	6	3	5	7	29	87	58	0	2013-2017	20-40	no	female
33	6	4	5	8	87	1111	1024	0	2013-2017	20-40	no	female
9109	1909	1	2	1	0	18	18	1	2013-2017	<=20	no	female
9110	1909	1	3	2	0	18	18	0	2013-2017	<=20	no	female
9111	1909	1	5	3	0	18	18	0	2013-2017	<=20	no	female
9112	1909	2	4	4	18	30	12	1	2013-2017	<=20	no	female
9113	1909	2	5	5	18	30	12	0	2013-2017	<=20	no	female
9114	1909	4	5	8	30	85	55	0	2013-2017	<=20	no	female

Data Preperation | Expanding

Expanding covariates: by adding type-specific covariates to the dataset.

Why is this important?

```
msdata_exp <-
  expand.covs(
    msdata_enc,
    covs = c('year', 'age', 'discount', 'gender'),
    longnames = TRUE
  )
```

Expanding covariates is library specific requirement

Expanding (1/2)

```
msdata_exp[msdata_enc$id == 1, c(1:8, 10, 29:44)]
An object of class 'msdata'

Data:
  id from to trans Tstart Tstop time status   age age.40.1 age.40.2 age.40.
1  1    1  2     1      0    22   22      1 20-40        0        0
2  1    1  3     2      0    22   22      0 20-40        0        0
3  1    1  5     3      0    22   22      0 20-40        0        0
4  1    2  4     4     22   995  973      0 20-40        0        0
5  1    2  5     5     22   995  973      0 20-40        0        0
  age.40.4 age.40.5 age.40.6 age.40.7 age.40.8 age20.40.1 age20.40.2 age20.
1        0        0        0        0        0          1          0
2        0        0        0        0        0          0          1
3        0        0        0        0        0          0          0
4        0        0        0        0        0          0          0
5        0        0        0        0        0          0          0
  age20.40.4 age20.40.5 age20.40.6 age20.40.7 age20.40.8
1          0          0          0          0          0
2          0          0          0          0          0
3          0          0          0          0          0
```

Expanding (2/2)

```
msdata_exp[msdata_enc$id == 1, c(1:8, 11, 45:52)]
An object of class 'msdata'

Data:
  id from to trans Tstart Tstop time status discount discountyes.1
1  1    1  2     1      0    22   22      1       no             0
2  1    1  3     2      0    22   22      0       no             0
3  1    1  5     3      0    22   22      0       no             0
4  1    2  4     4     22   995  973      0       no             0
5  1    2  5     5     22   995  973      0       no             0
  discountyes.2 discountyes.3 discountyes.4 discountyes.5 discountyes.6
1             0             0             0             0             0
2             0             0             0             0             0
3             0             0             0             0             0
4             0             0             0             0             0
5             0             0             0             0             0
  discountyes.7 discountyes.8
1             0             0
2             0             0
3             0             0
```

Events (1/2)

Let's see transactions by using `events()` function:

```
events(msdata_enc) %>%
{
  .$Proportions <- round(.$Proportions,3)
  .
}
```

Events (2/2)

What can we say?

```
$Frequencies
     to
from   st1  st2  st3  st4  st5 no event total entering
  st1    0  640  777    0  160      332           1909
  st2    0    0    0  194   39      407            640
  st3    0    0    0  359  197      221            777
  st4    0    0    0    0  137      416            553
  st5    0    0    0    0    0      533            533

$Proportions
     to
from    st1   st2   st3   st4   st5 no event
  st1 0.000 0.335 0.407 0.000 0.084    0.174
  st2 0.000 0.000 0.000 0.303 0.061    0.636
  st3 0.000 0.000 0.000 0.462 0.254    0.284
  st4 0.000 0.000 0.000 0.000 0.248    0.752
  st5 0.000 0.000 0.000 0.000 0.000    1.000
```

Cox PH Model | Intercept Only

Stratified Cox model, where

- `method` parameter specifies how to handle ties
- id: customer id, used only when we have mixed effect and multiple event type
- strata

```
c0 <- coxph(Surv(Tstart, Tstop, status) ~ strata(trans),
            data = msdata_exp,id = id)
```

Cox PH Model | Probabilities

In order to find the probability of *being in a state*, we can use `msfit()`:

```
msf0 <- msfit(object = c0, vartype = "greenwood",
              trans = trans_mat)
```

Reading Material

Cumulative Hazards

```
head(msf0$Haz)
```

time	Haz	trans
1	0.0000000	1
3	0.0000000	1
4	0.0000000	1
5	0.0000000	1
6	0.0000000	1
7	0.0010641	1

```
tail(msf0$Haz)
```

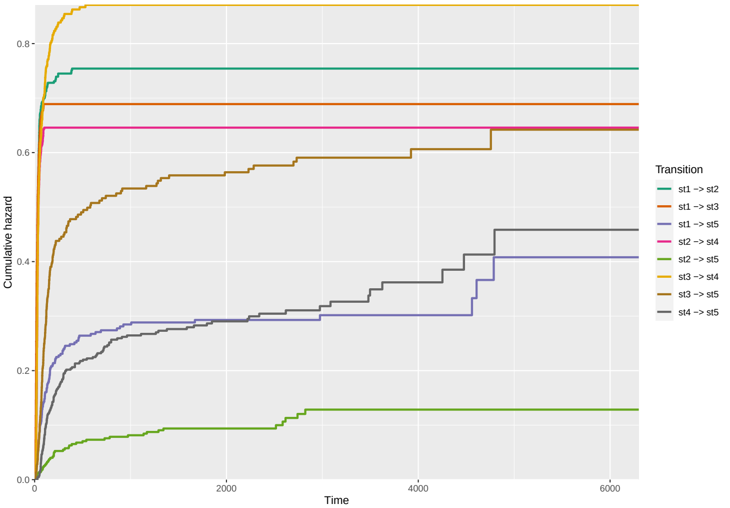
	time	Haz	trans
2835	4560	0.4129644	8
2836	4608	0.4129644	8
2837	4757	0.4129644	8
2838	4787	0.4129644	8
2839	4795	0.4584190	8
2840	6299	0.4584190	8

Cumulative Hazard Plots (1/2)

Probability that a customer would move from one state to another, with respect to the time since the initial subscription began.

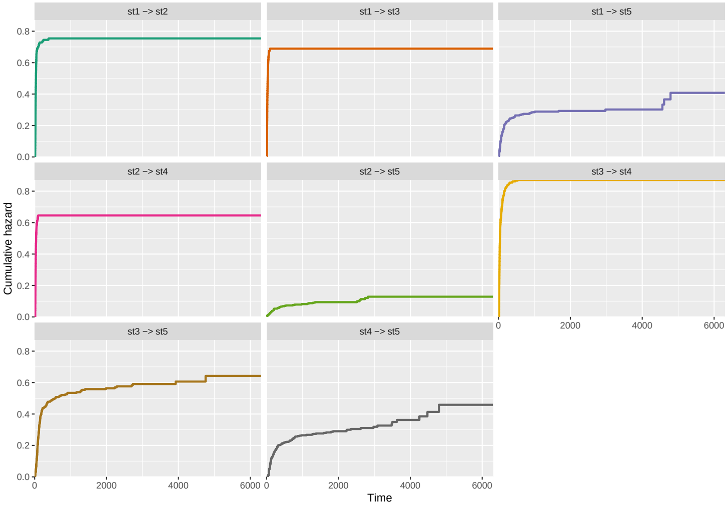
```
plot(msf0, use.ggplot = TRUE)
```

Cumulative Hazard Plots (2/3)



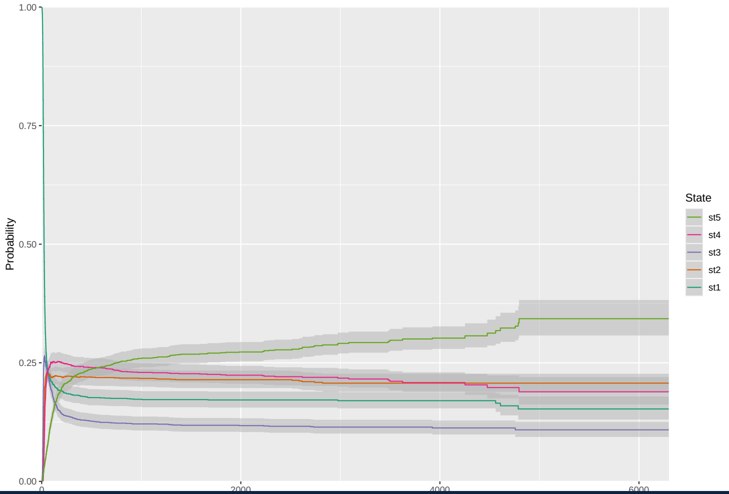
Comulative Hazard Plots (3/3)

```
plot(msf0, type = "separate", use.ggplot = TRUE, scale_type = "fixed")
```



Survival Curves | Transitions

```
pt0 <- probtrans(msf0 , predt = 0, method = "greenwood")
plot(pt0,use.ggplot = TRUE,type = "single")
```



Cox PH Model | Covariates (1/4)

Note, here you can use all the features from `msdata_exp` dataframe:

```
cox_mmodel<- coxph(Surv(Tstart, Tstop, status) ~ year + age + discount + gender+
     year2008.2012.1 + year2008.2012.2 + year2008.2012.3 + year2008.2012.4 + year2008.2012.5 +
     year2008.2012.6 + year2008.2012.7 + year2013.2017.1 + year2013.2017.2 + year2013.2017.3 +
     year2013.2017.4 + year2013.2017.5 + year2013.2017.6 + year2013.2017.7+ age20.40.1 + age20.4
     age20.40.3 + age20.40.4 + age20.40.5 + age20.40.6 + age20.40.7 + age.40.1 + age.40.2 + age
     age.40.4 + age.40.5  + age.40.6 + age.40.7 + discountyes.1 + discountyes.2 + discountyes.3
     discountyes.4 + discountyes.5 + discountyes.6 + discountyes.7 + gendermale.1 + gendermale.2
     gendermale.3 + gendermale.4 + gendermale.5 + gendermale.6 + gendermale.7 +
          strata(trans),
          data = msdata_exp)
```

Cox PH Model | Covariates (2/4)

Coefficients

```
coef(cox_mmodel)
  year2008-2012   year2013-2017          age>40        age20-40
    -0.32694473     -0.38050037      1.31952001      0.79673590
     gendermale year2008.2012.1 year2008.2012.2 year2008.2012.3 year
    -0.52279656      0.80153982      0.39138299      0.01784092
year2008.2012.5 year2008.2012.6 year2008.2012.7 year2013.2017.1 year
    -0.53750406      0.88056834     -0.34889098      0.94685317
year2013.2017.3 year2013.2017.4 year2013.2017.5 year2013.2017.6 year
    -0.10612388      0.22579554     -0.63305989      1.33899820
     age20.40.1      age20.40.2      age20.40.3      age20.40.4
    -0.76818847     -0.71782175     -0.03670966     -0.66128608
     age20.40.6      age20.40.7        age.40.1        age.40.2
    -1.23666362     -0.56759407     -1.14418226     -1.33804057
       age.40.4        age.40.5        age.40.6        age.40.7    di
    -0.86053748      0.14155423     -1.65687790     -0.82189505
  discountyes.2   discountyes.3   discountyes.4   discountyes.5    di
    -0.15019627      0.07554174     -0.09104062      0.14142296
```

Cox PH Model | Covariates (3/4)

Now, let's make it more intuitive by displaying the coefficients per each state.

```
data.frame(
  names = names(coef(cox_mmodel)),
  values = coef(cox_mmodel),
  stringsAsFactors = FALSE
) %>%
  dplyr::filter(grepl(".", names, fixed = TRUE)) %>%
  separate(names, sep = "\\.[0-9]{1}$", into = c('variable', 'transition'))
  mutate(transition = rep(1:7, length(unique(variable)))) %>%
  spread(variable, values)
```

Cox PH Model | Covariates (4/4)

transition	age.40	age20.40	discountyes	gendermale	year2008.2012	year2013.2017
1	-1.1441823	-0.7681885	-0.2604790	0.7190754	0.8015398	0.9468532
2	-1.3380406	-0.7178218	-0.1501963	0.5839045	0.3913830	0.3128180
3	-0.4310313	-0.0367097	0.0755417	0.5305225	0.0178409	-0.1061239
4	-0.8605375	-0.6612861	-0.0910406	0.2215672	0.1254660	0.2257955
5	0.1415542	-0.6420192	0.1414230	0.2449388	-0.5375041	-0.6330599
6	-1.6568779	-1.2366636	0.2391369	0.3499550	0.8805683	1.3389982
7	-0.8218950	-0.5675941	0.3917949	0.5699945	-0.3488910	0.0939134

User Personas | Creating

Now let's see two cases:

Client A

- discount: Yes
- gender: female
- year: 2013-2017
- age: $<=$20

Client B

- discount: No
- gender: Male
- year: 2002-2007
- age: 20-40

User Personas | Data manipulation

Client A

```
clientA <-
  msdata_exp %>%
  filter(
    discount == "yes",
    gender   == "female",
    year     == "2013-2017",
    age      == "<=20"
  ) %>%
  magrittr::extract(1:8, ) %>%
  {
    attr(., "trans") <- trans_mat
    .
  } %>%

  select(contains('year'), contains('age'),
         contains('discount'), contains('gender
  mutate(trans = 1:8) %>%
  mutate(strata = trans)
```

Client B

```
clientB <-
  msdata_exp %>%
  filter(
    discount == "no",
    gender   == "male",
    year     == "2002-2007",
    age      == "20-40"
  ) %>%
  magrittr::extract(1:8, ) %>%
  {
    attr(., "trans") <- trans_mat
    .
  } %>%

  select(contains('year'), contains('age'),
         contains('discount'), contains('gender'
  mutate(trans = 1:8) %>%
  mutate(strata = trans)
```

User Personas | Predictions

Client A

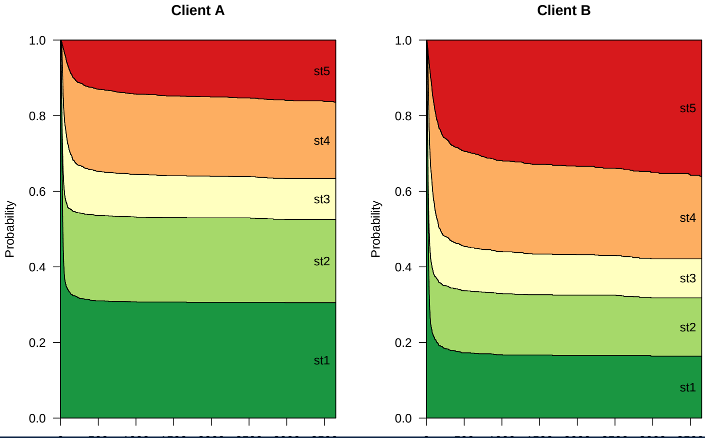
```
fitA <- msfit(cox_mmodel, newdata = clientA,
              trans = trans_mat) %>%
  probtrans(pred = 0)
```

Client B

```
fitB<-msfit(cox_mmodel, newdata = clientB,
                    trans = trans_mat) %>%
  probtrans(pred = 0)
```

User Personas | Visualization

```
par(mfrow = c(1,2))
plot(fitA, main = "Client A", las = 1, xlab = "Days since registration", type = "filled",
     xlim = c(0, 3650))
plot(fitB, main = "Client B", las = 1, xlab = "Days since registration", type = "filled",
     xlim = c(0, 3650))
```



Summary

Multistate models provide an opportunity for business owner (decision maker) to observe *customer journey* within different segments.

Decision maker can develop strategies for:

- Acquisition activities: to target segments with higher survival probabilities
- Retention activities: to launch retention programs towards at risk segments

Individual Project Opportunity

You can find related papers to solve multistate problems!
It might be useful to research on Markov Chains process

Modeling CLV with Survival

Objective

We will try to:

- Redefine the concept of CLV for the service providing companies
- Demonstrate how Survival models can help to estimate CLV

We will estimate CLV with customer average monthly margin and customer survival curve.

CLV | Throwback

Recall the definition of CLV:
the Net Present Value of customers calculated profit over a certain number of months.

$$CLV = MM \sum_{i=1}^{t} \frac{p_i}{(1 + r/12)^{i-1}}$$

where:

- MM : average monthly margin (constant)
- p_i : survival probability in period i
- r : is the discount rate

Question: What will be equal p_1 ?

CLV estimation with Survival models

If we estimate p_i by the help of survival models, the problem will be solved !

$$CLV = MM \sum_{i=1}^{t} \frac{p_i}{(1 + r/12)^{i-1}}$$

Survival Data Preperation

Recreating survival object for telco data

```
load("Data/telco.Rda");
telco$churn=ifelse(telco$churn=='Yes',1,0)
surv_obj=Surv(time=telco$tenure, event=telco$churn)
surv_obj[1:10]
 [1] 13  11  68+ 33  23+ 41+ 45  38+ 45+ 68+
```

Survival Model Building

Let's build Cox proportional hazard model and predict on the same data.

Take a look at pred object.

```
cox_model = coxph(surv_obj~gender+age+ed, data=telco)
pred=survfit(cox_model, newdata = telco)
list.tree(pred)
 pred = list 14 (3078544 bytes)( survfitcox survfit )
.  n = integer 1= 1000
.  time = double 72= 1 2 3 4 5 6 7 8 ...
.  n.risk = double 72= 1000 987 980 960 ...
.  n.event = double 72= 10 3 15 11 14 6 ...
.  n.censor = double 72= 3 4 5 8 5 9 10 8 ...
.  surv = double 72000= named array 72 X 1000= 0.99063 0.98779 ...
.  cumhaz = double 72000= array 72 X 1000= 0.0094108 0.01228 ...
.  std.err = double 72000= array 72 X 1000= 0.0031902 0.0037206 ...
.  logse = logical 1= TRUE
.  lower = double 72000= named array 72 X 1000= 0.98446 0.98062 ...
.  upper = double 72000= named array 72 X 1000= 0.99685 0.99502 ...
.  conf.type = character 1= log
.  ...   and 2 more
```

Data Preperation for CLV

Let's keep only 24 months!

```
pred_data=data.frame(t(pred$surv))[,0:24]
head(pred_data)[,0:5]
         X1        X2        X3        X4        X5
1 0.9906334 0.9877948 0.9734323 0.9626963 0.9488228
2 0.9801670 0.9741981 0.9442934 0.9222621 0.8942005
3 0.9975337 0.9967829 0.9929595 0.9900742 0.9863104
4 0.9897654 0.9866656 0.9709940 0.9592934 0.9441915
5 0.9897988 0.9867090 0.9710877 0.9594242 0.9443694
6 0.9929452 0.9908040 0.9799465 0.9718045 0.9612497
```

CLV Calculation (1/2)

Now, having p_i we can calculate CLV by assuming that discount rate (r) is 10% and average monthly margin is 1300 AMD.

$$CLV = MM \sum_{i=1}^{t} \frac{p_i}{(1 + r/12)^{i-1}}$$

```
sequence = seq(1,length(colnames(pred_data)),1)
MM = 1300
r = 0.1
for (num in sequence) {
  pred_data[,num]=pred_data[,num]/(1+r/12)^(sequence[num]-1)
}
```

CLV Calculation (1/2)

```
head(pred_data)
         X1        X2        X3        X4        X5        X6        X7
1 0.9906334 0.9796312 0.9574090 0.9390245 0.9178435 0.9044990 0.8892681
2 0.9801670 0.9661469 0.9287497 0.8995846 0.8650046 0.8463438 0.8239767
3 0.9975337 0.9885451 0.9766148 0.9657292 0.9541070 0.9446474 0.9347082
4 0.9897654 0.9785113 0.9550108 0.9357053 0.9133634 0.8995521 0.8836894
5 0.9897988 0.9785544 0.9551030 0.9358329 0.9135354 0.8997420 0.8839035
6 0.9929452 0.9826155 0.9638159 0.9479088 0.9298646 0.9177871 0.9042753
         X8        X9       X10       X11       X12       X13       X14
1 0.8760247 0.8598646 0.8437661 0.8276197 0.8156632 0.8027129 0.7918651
2 0.8055856 0.7815676 0.7578137 0.7341217 0.7184275 0.7008987 0.6872974
3 0.9253538 0.9152244 0.9051394 0.8950664 0.8862138 0.8771145 0.8686551
4 0.8699856 0.8531164 0.8363181 0.8194724 0.8071634 0.7937781 0.7826640
5 0.8702173 0.8533752 0.8366036 0.8197846 0.8074889 0.7941202 0.7830163
6 0.8922893 0.8780709 0.8638970 0.8496820 0.8387049 0.8269656 0.8168632
        X15       X16       X17       X18       X19       X20       X21
1 0.7778878 0.7650431 0.7477860 0.7405073 0.7255060 0.7138741 0.7068243
2 0.6679577 0.6507652 0.6257491 0.6186223 0.5978237 0.5830409 0.5762249
3 0.8593292 0.8503618 0.8401279 0.8328606 0.8233447 0.8148571 0.8077781
4 0.7681684 0.7548968 0.7368716 0.7295984 0.7140084 0.7020464 0.6950082
```

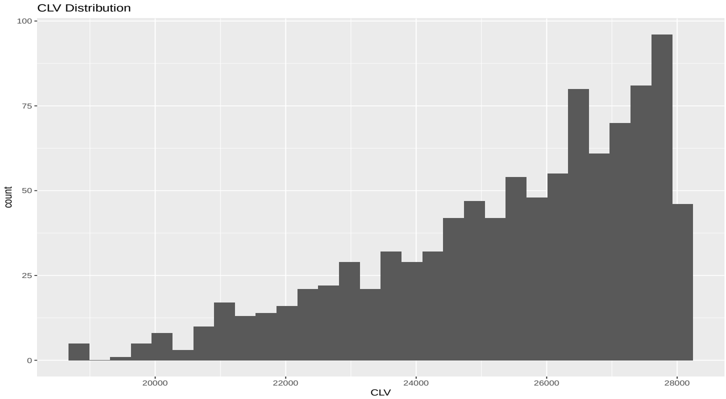
CLV Calculation (2/2)

Multiplying Monthly Margin (MM) with the sum of discounted probabilities. Here, we are using `rowSums()` function for row wise calculations:

```
pred_data$CLV=MM*rowSums(pred_data)
summary(pred_data$CLV)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  18980   24171   25909   25433   27189   28235
```

CLV Distribution

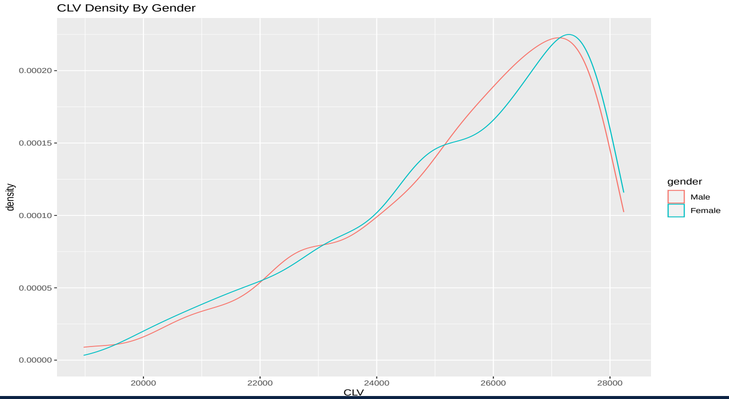
```
ggplot(pred_data,aes(x=CLV))+labs(title = "CLV Distribution")+
  geom_histogram()
```



CLV Distribution | Gender

Is there any difference?

```
telco$CLV = pred_data$CLV
ggplot(telco,aes(x=CLV, color=gender))+
  labs(title = "CLV Density By Gender")+
  geom_density()
```



CLV Distribution | Features

try with other features