CENSORING, SURVIVAL, & HAZARDS

Dr. Aric LaBarr

Institute for Advanced Analytics

SURVIVAL FUNCTION

Kaplan-Meier Method

- Estimating the survival function:
  - Want to estimate the proportion of individuals “still alive” at any given time *t*.

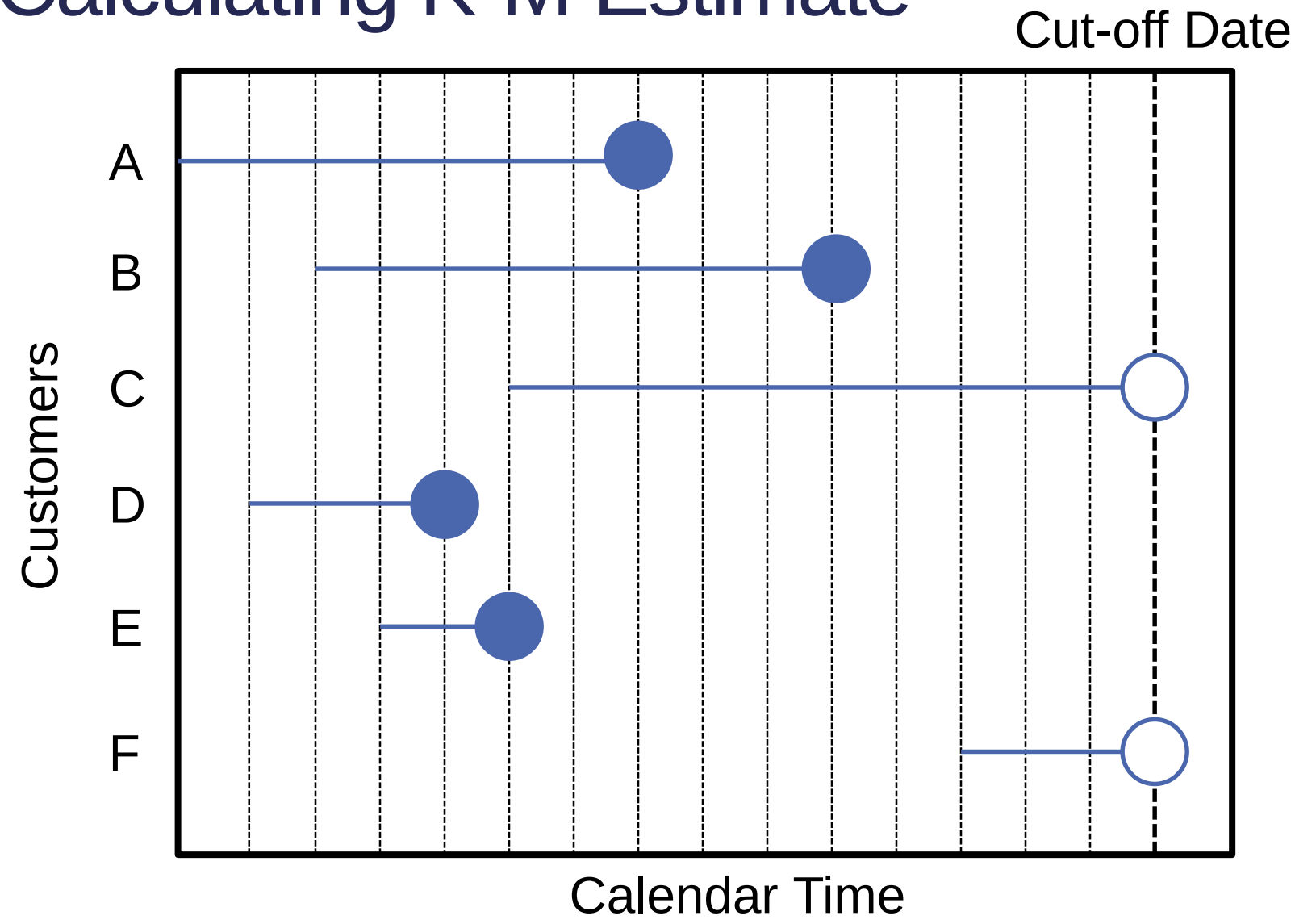
$$\hat{S}(t) = \prod_{k \le t} \left(1 - \frac{d_k}{r_k}\right)$$

d_k → # events occurring at time *t*

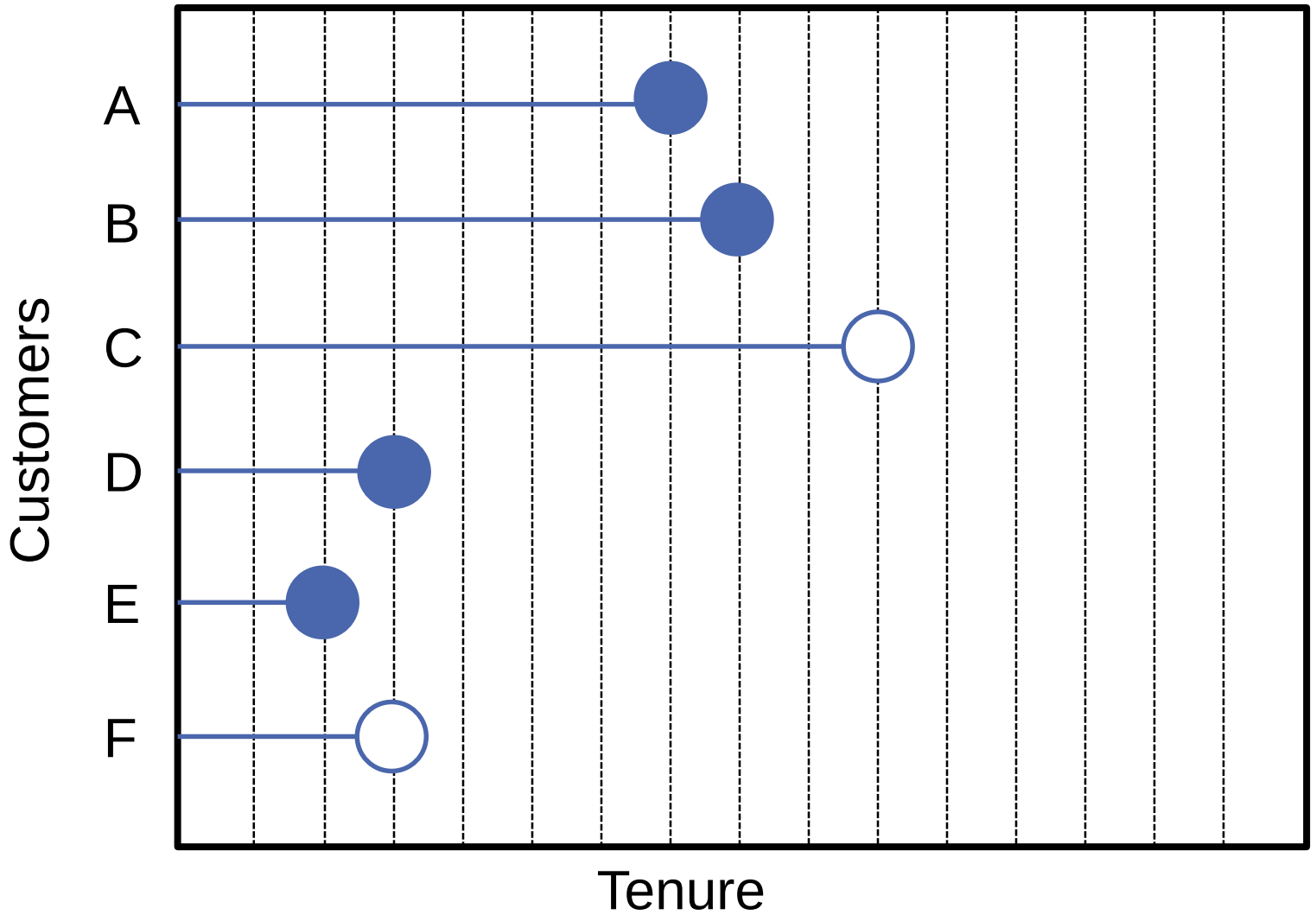
r_k → # observations available right before time *t* (**risk set**)

- The Kaplan-Meier method existed long before Kaplan and Meier.
- Kaplan and Meier showed it was the maximum likelihood estimate for the nonparametric estimation of the survival curve.

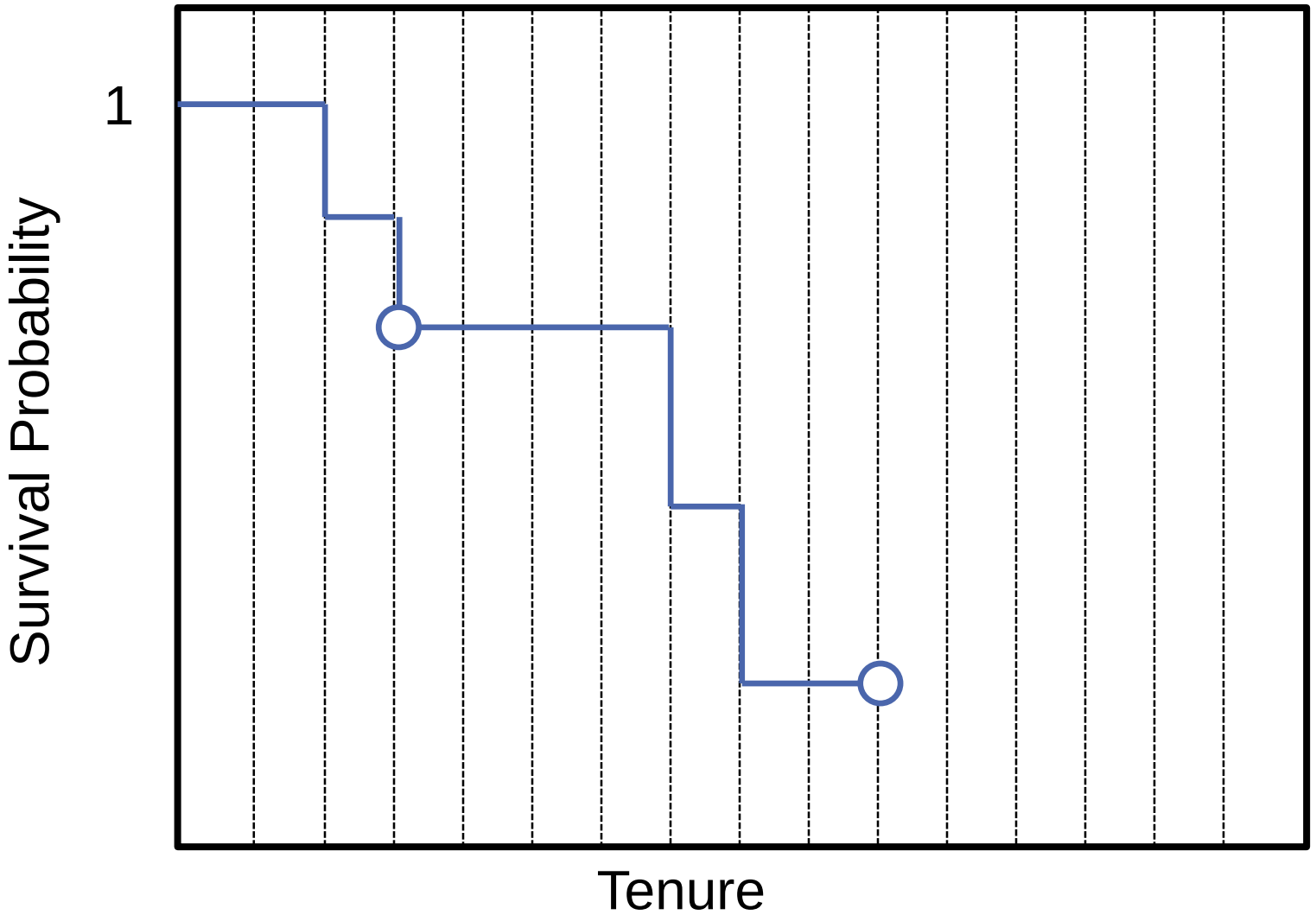
Calculating K-M Estimate



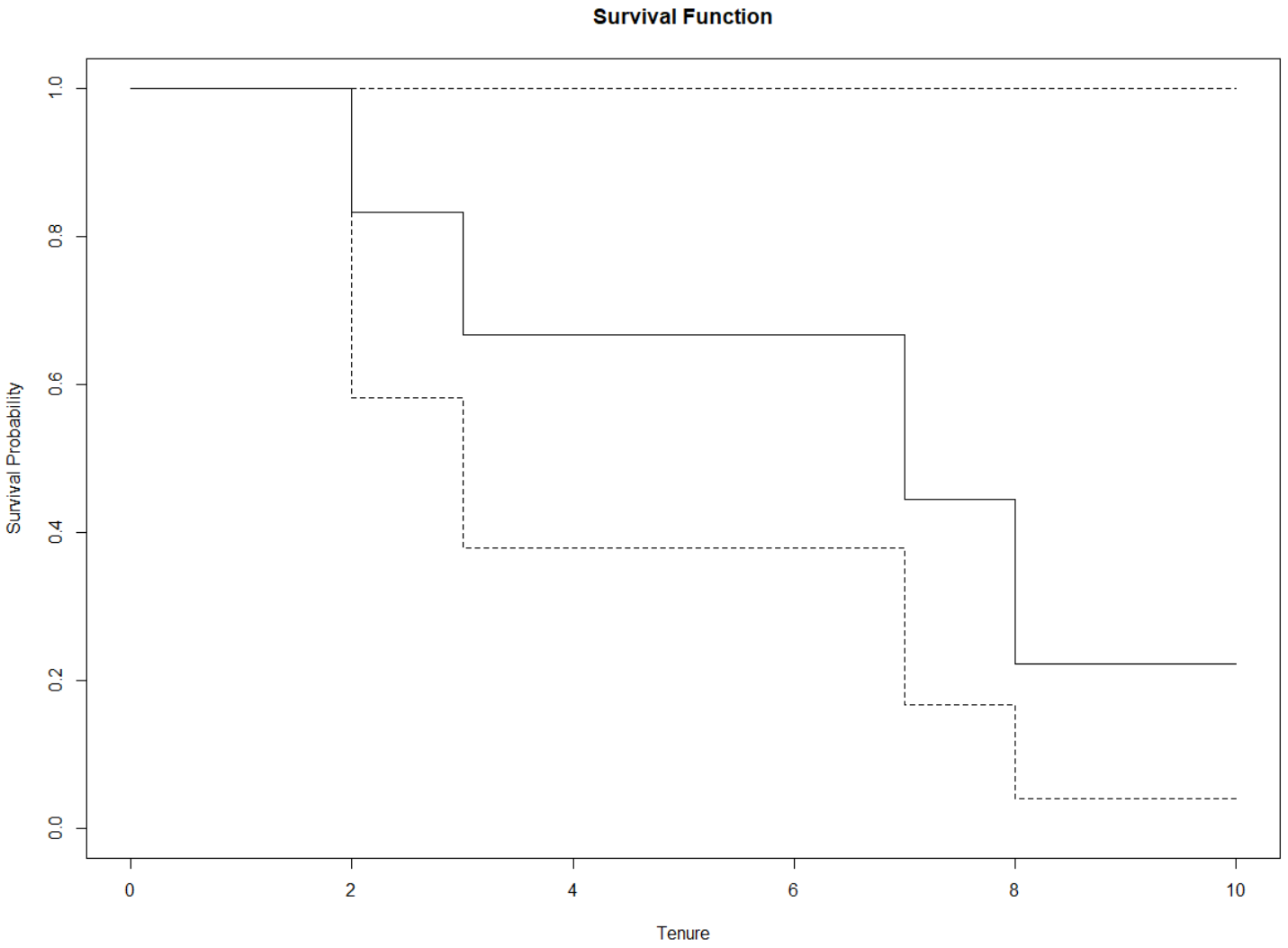
Calculating K-M Estimate



Visualizing K-M Estimate



Visualizing K-M Estimate



Survival Function – R

```
Surv(time = simple$tenure, event = simple$censored == 0)
```

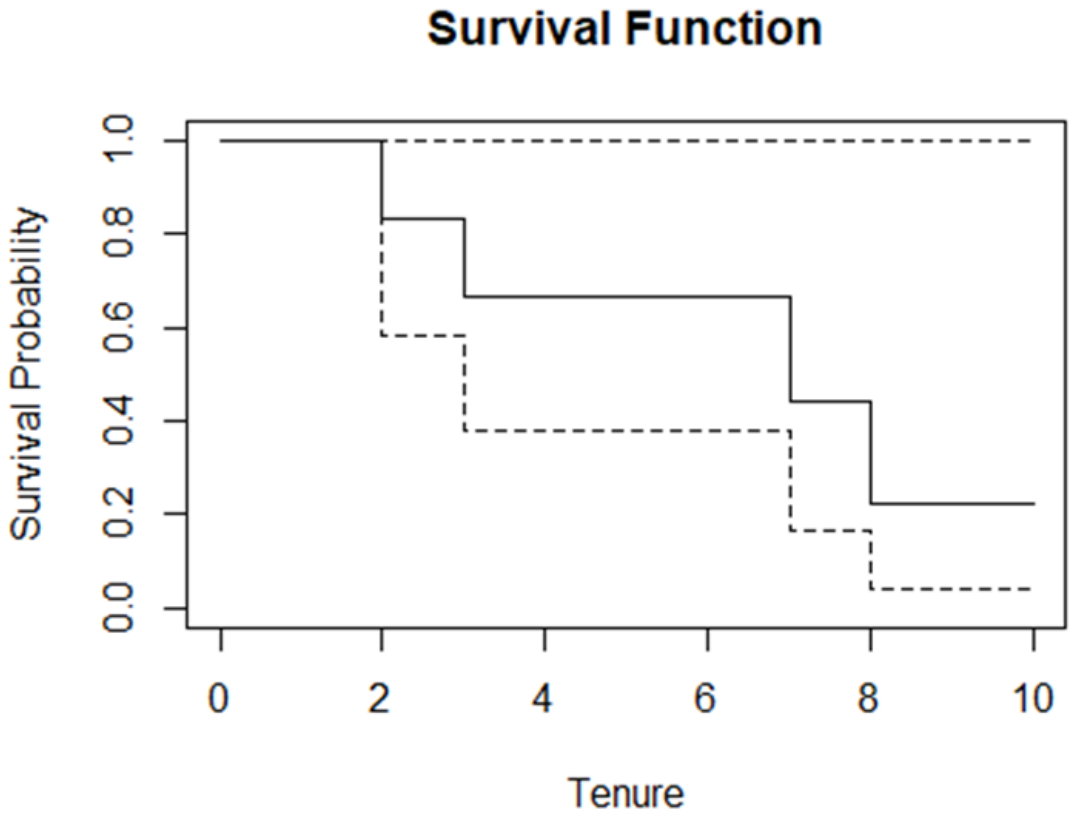
```
## [1]  7   8  10+  3   2   3+
```

```
simple_km <- survfit(Surv(time = tenure, event = (censored == 0)) ~ 1,
                     data = simple)
summary(simple_km)
```

```
## Call: survfit(formula = Surv(time = tenure, event = (censored == 0)) ~
##     1, data = simple)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     2      6       1    0.833   0.152       0.5827            1
##     3      5       1    0.667   0.192       0.3786            1
##     7      3       1    0.444   0.222       0.1668            1
##     8      2       1    0.222   0.192       0.0407            1
```

Survival Function – R

```
plot(simple_km, main = "Survival Function", xlab = "Tenure",
     ylab = "Survival Probability")
```



Survival Function – R

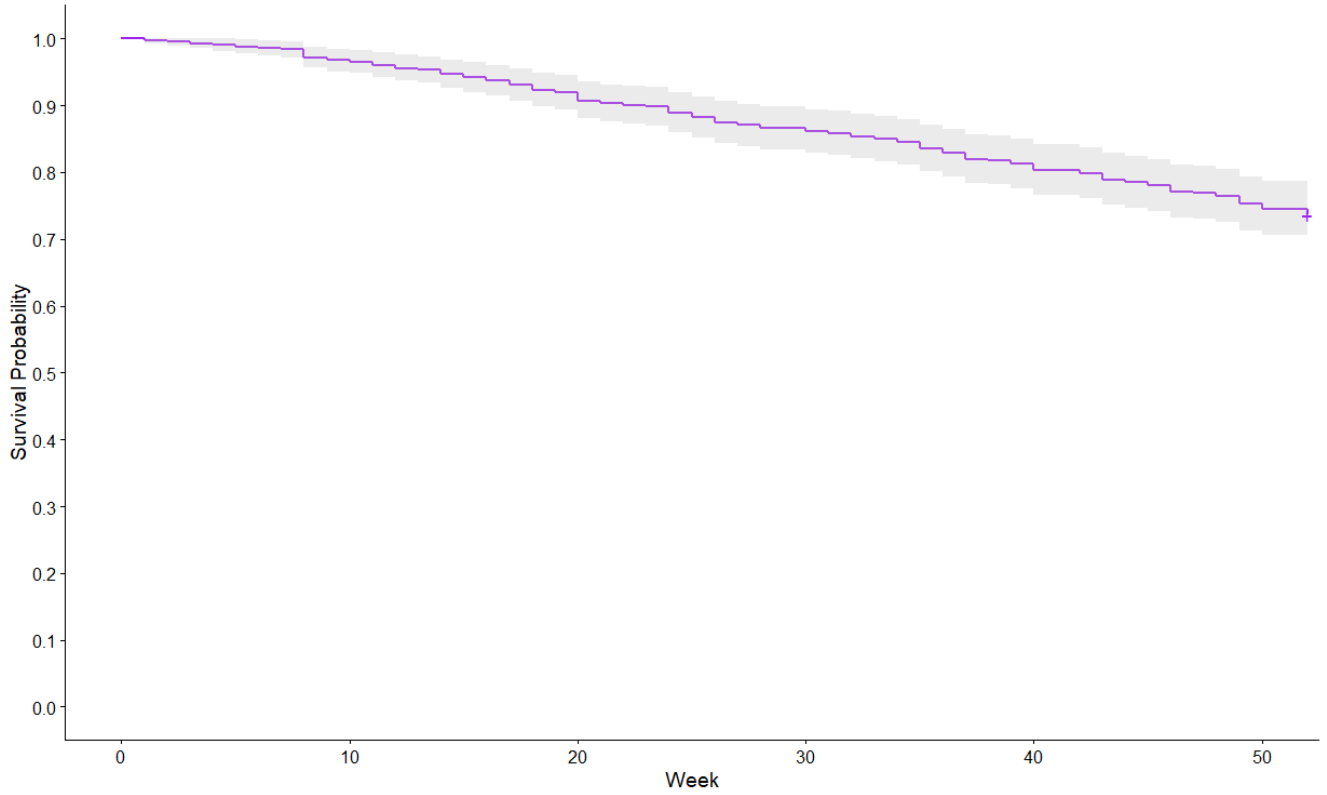
```
recid_surv <- Surv(time = recid$week, event = recid$arrest == 1)

recid_km <- survfit(recid_surv ~ 1, data = recid)
summary(recid_km)
```

```
## Call: survfit(formula = recid_surv ~ 1, data = recid)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1    432       1    0.998 0.00231        0.993        1.000
##     2    431       1    0.995 0.00327        0.989        1.000
##     3    430       1    0.993 0.00400        0.985        1.000
##     4    429       1    0.991 0.00461        0.982        1.000
##     5    428       1    0.988 0.00515        0.978        0.999
##     6    427       1    0.986 0.00563        0.975        0.997
##     7    426       1    0.984 0.00607        0.972        0.996
##     8    425       5    0.972 0.00791        0.957        0.988
```

Survival Function – R

```
ggsurvplot(recid_km, data = recid, conf.int = TRUE, palette = "purple",
           xlab = "Week", ylab = "Survival Probability", legend = "none",
           break.y.by = 0.1)
```



STRATIFIED ANALYSIS

Comparing Survival Function

- Log-Rank test:

$$\text{LogRank} = \frac{1}{\hat{\sigma}^2}\left\{\sum_{j=1}^{r}\left(d_{1,j} - e_{1,j}\right)\right\}^2$$

- Wilcoxon test (places larger emphasis on earlier event times):

$$\text{Wilcoxon} = \frac{1}{\hat{\sigma}^2}\left\{\sum_{j=1}^{r}\left(d_{1,j} - e_{1,j}\right)n_j\right\}^2$$

Stratified Analysis – R

```
survdiff(recid_surv ~ wexp, rho = 0, data = recid)
```

```
## Call:
## survdiff(formula = recid_surv ~ wexp, data = recid, rho = 0)
##
##          N Observed Expected (O-E)^2/E (O-E)^2/V
## wexp=0 185       62     45.6      5.91      9.91
## wexp=1 247       52     68.4      3.94      9.91
##
##  Chisq= 9.9  on 1 degrees of freedom, p= 0.002
```

Stratified Analysis – R

```
recid_strat <- survfit(recid_surv ~ wexp, data = recid)
summary(recid_strat)
```

```
## Call: survfit(formula = recid_surv ~ wexp, data = recid)
##
##                 wexp=0
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1    185       1    0.995 0.00539        0.984        1.000
##     3    184       1    0.989 0.00760        0.974        1.000
##     5    183       1    0.984 0.00929        0.966        1.000
```

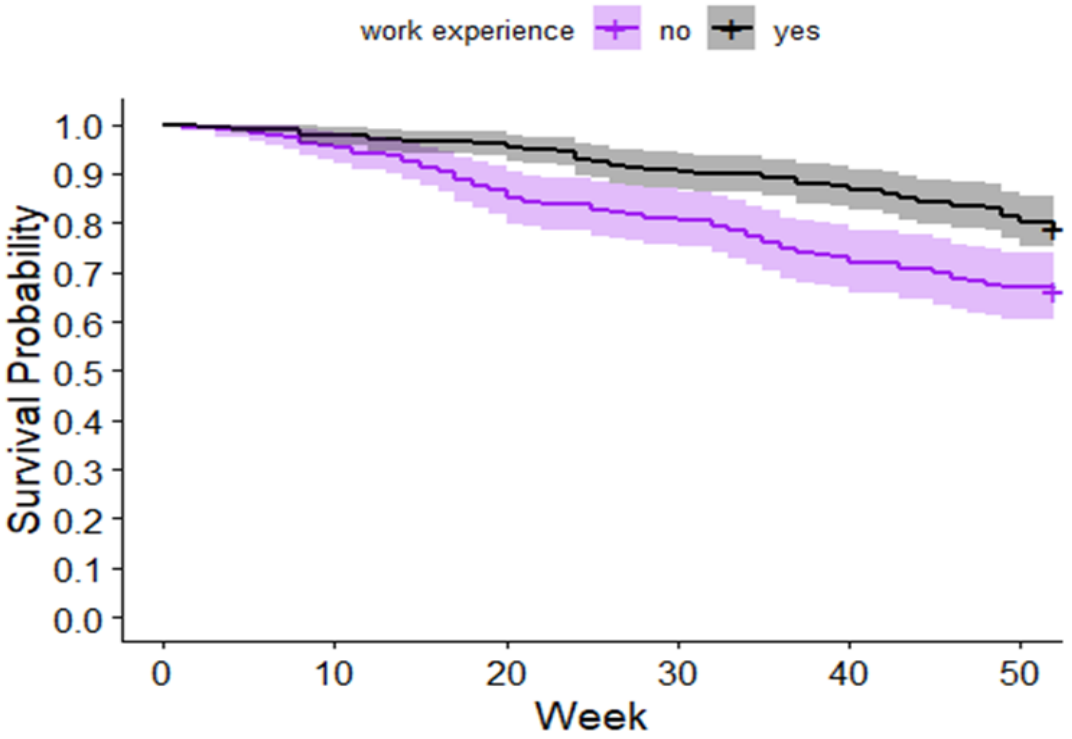
⋮

```
##                 wexp=1
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     2    247       1    0.996 0.00404        0.988        1.000
##     4    246       1    0.992 0.00570        0.981        1.000
##     8    245       3    0.980 0.00896        0.962        0.997
##     9    242       1    0.976 0.00980        0.957        0.995
```

⋮

Stratified Analysis – R

```
ggsurvplot(recid_strat, data = recid, conf.int = TRUE,
           palette = c("purple", "black"),
           xlab = "Week", ylab = "Survival Probability", break.y.by = 0.1,
           legend.title = "work experience", legend.labs = c("no", "yes"))
```



HAZARD FUNCTION

Hazard Function

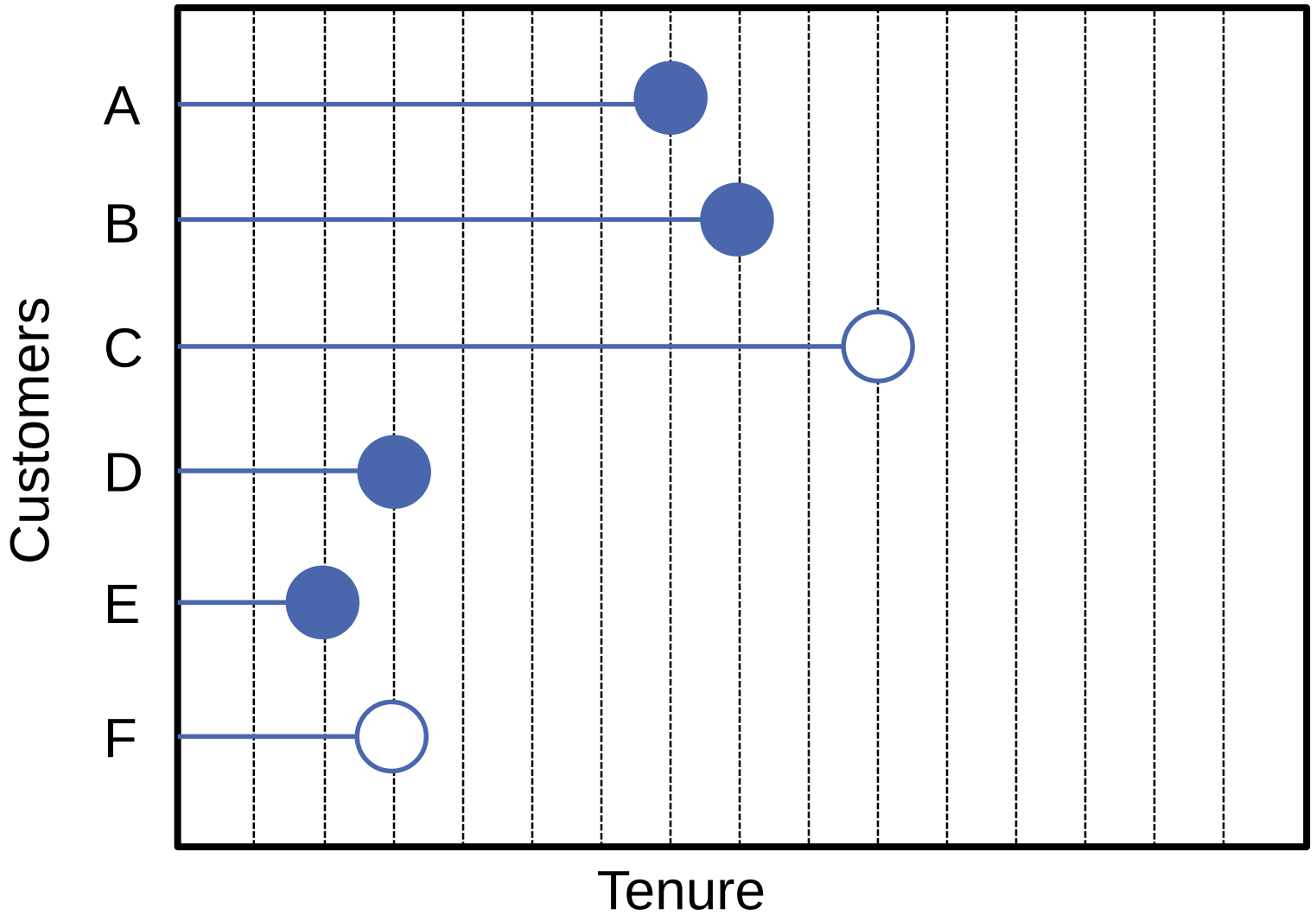
- In survival analysis we also use the **hazard function** to summarize the data.
- There are two common types of hazard functions:
  1. Hazard Probabilities:

$$h(t) = P(t < T < t + 1 \mid T > t)$$

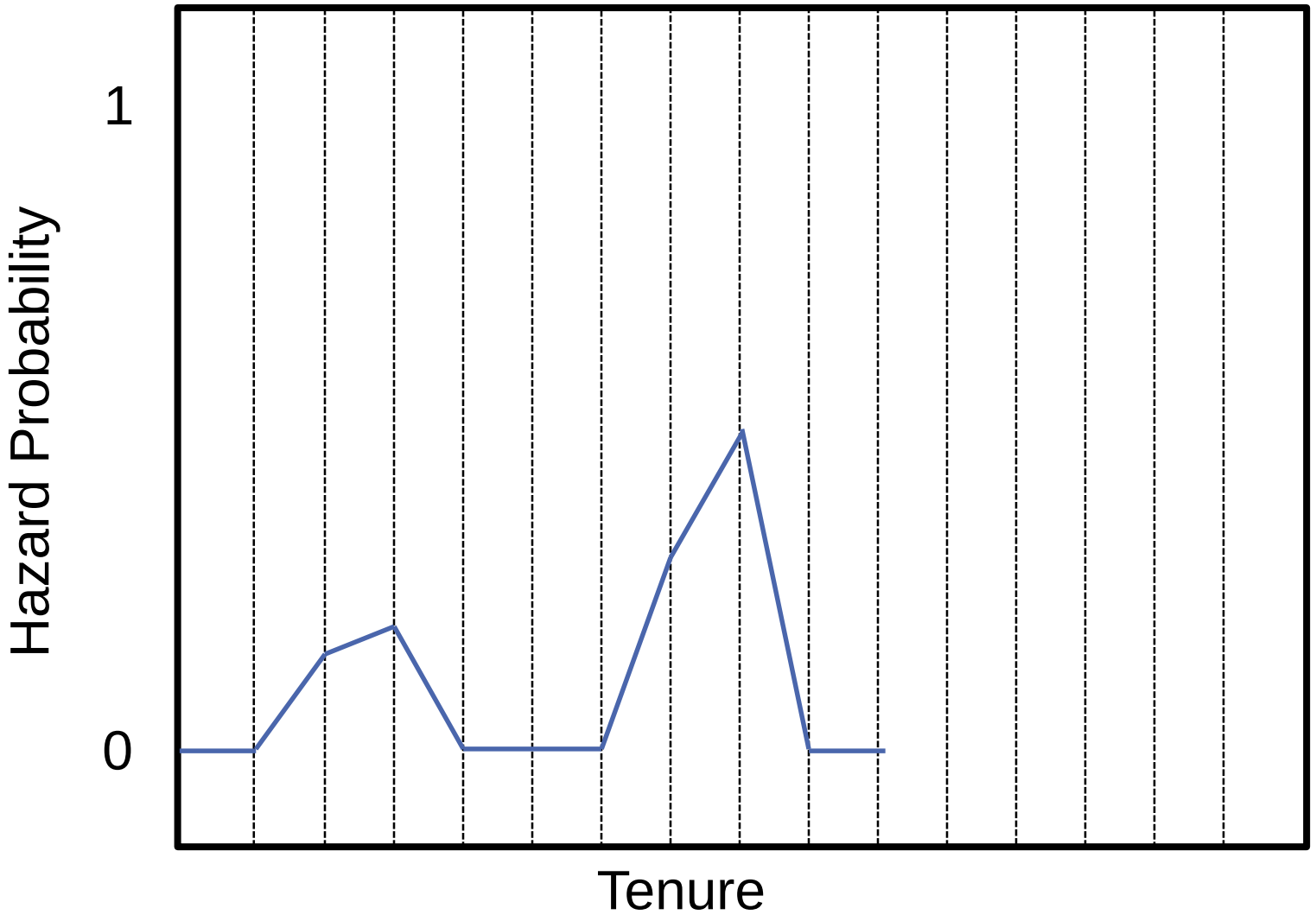
  2. Hazard Rates:

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

Calculating Hazard Probabilities



Visualizing Hazard Probabilities



Hazard Rates

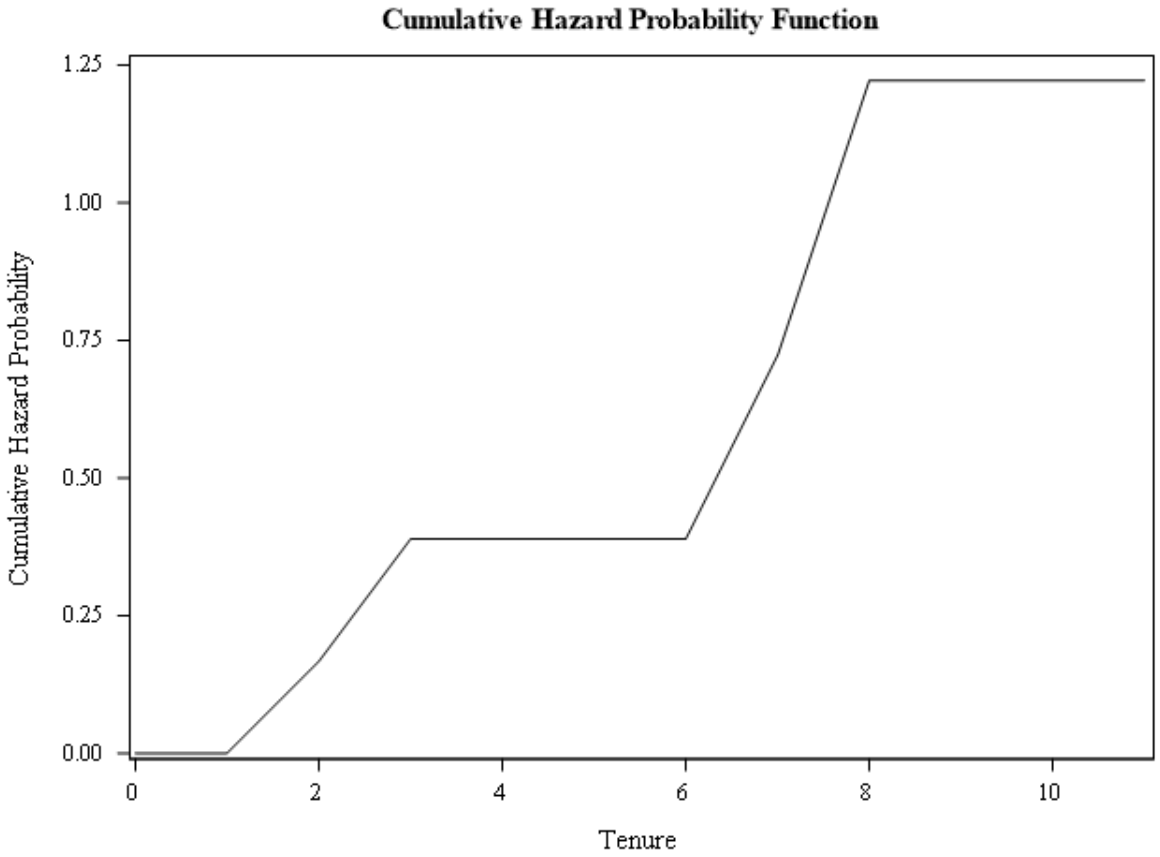
- Hazard rates have a slightly different interpretation than the hazard probabilities because they are limits of conditional probabilities.

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

- The hazard rate is the **instantaneous event rate** for the risk set at time *t*.
  - Given survival up until time *t*, it is the rate of events in the interval $[t, t + \Delta t)$.

Cumulative Hazard Probability

- The **cumulative hazard probability** is just the total hazard rate up until time t – denoted $\Lambda(t)$.



Hazard Functions – R

```
summary(simple_km)
```

```
## Call: survfit(formula = Surv(time = tenure, event = (censored == 0))
~
##     1, data = simple)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     2      6       1    0.833   0.152       0.5827             1
##     3      5       1    0.667   0.192       0.3786             1
##     7      3       1    0.444   0.222       0.1668             1
##     8      2       1    0.222   0.192       0.0407             1
```

```
simple_km$hp <- simple_km$n.event/simple_km$n.risk
print(simple_km$hp)
```

```
## [1] 0.1666667 0.2000000 0.3333333 0.5000000 0.0000000
```

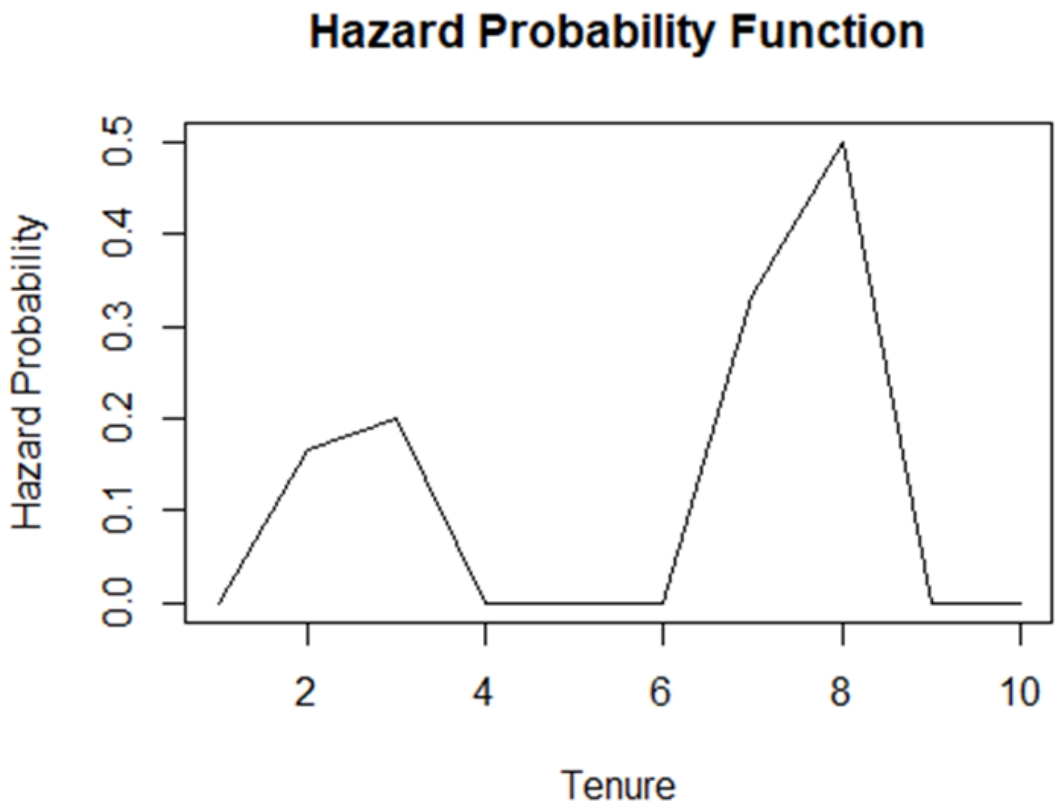
Hazard Functions – R

```
simple_haz <- merge(data.frame(time = seq(1,10,1)),
                    data.frame(time = simple_km$time, hp = simple_km$hp),
                    by = "time", all = TRUE)
simple_haz[is.na(simple_haz) == TRUE] <- 0
print(simple_haz)
```

```
##    time        hp
## 1     1 0.0000000
## 2     2 0.1666667
## 3     3 0.2000000
## 4     4 0.0000000
## 5     5 0.0000000
## 6     6 0.0000000
## 7     7 0.3333333
## 8     8 0.5000000
## 9     9 0.0000000
## 10   10 0.0000000
```

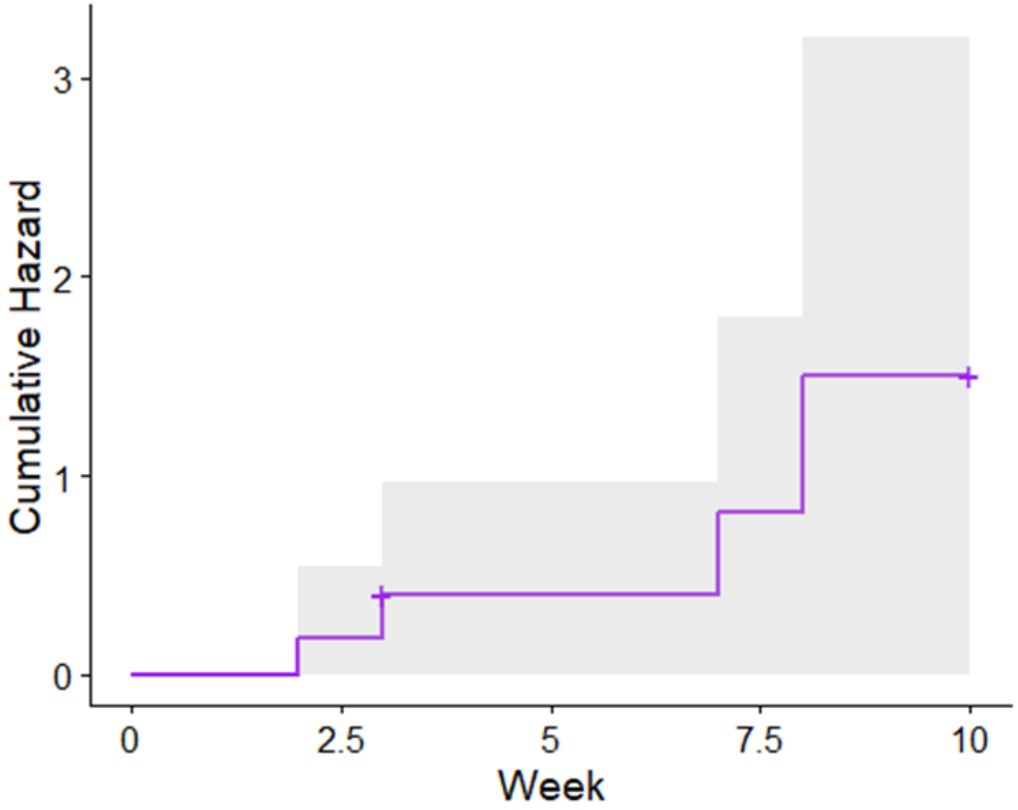
Hazard Functions – R

```
plot(y = simple_haz$hp, x = simple_haz$time,
     main = "Hazard Probability Function", xlab = "Tenure",
     ylab = "Hazard Probability", type = 'l')
```



Hazard Functions – R

```
ggsurvplot(simple_km, data = simple, fun = "cumhaz", conf.int = TRUE,
           palette = "purple", xlab = "Week",
           ylab = "Cumulative Hazard", legend = "none")
```

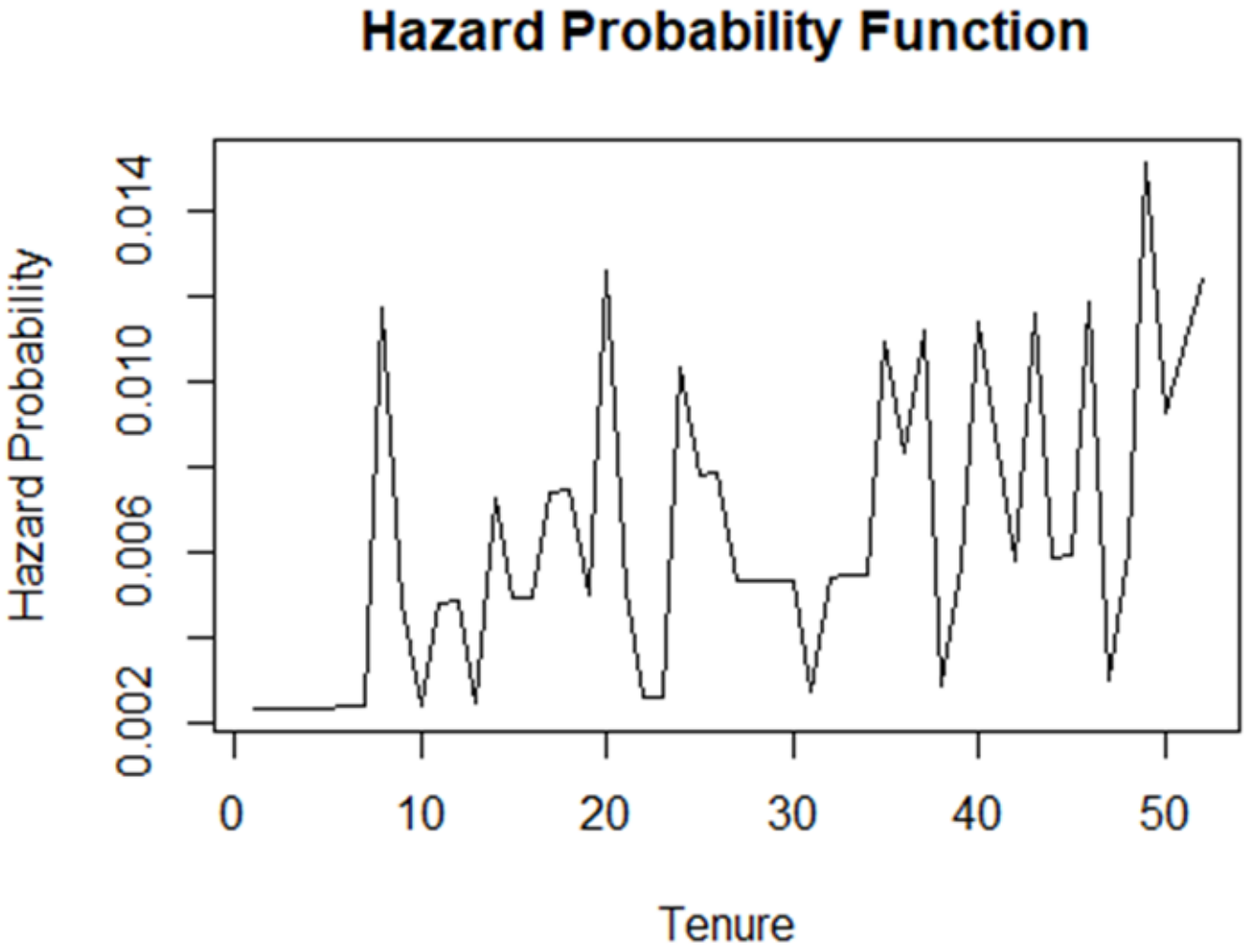


Hazard Functions – R

```
recid_km$hp <- recid_km$n.event/recid_km$n.risk
recid_haz <- merge(data.frame(time = seq(1,10,1)),
                   data.frame(time = recid_km$time, hp = recid_km$hp),
                   by = "time", all = TRUE)
recid_haz[is.na(recid_haz) == TRUE] <- 0

plot(y = recid_haz$hp, x = recid_haz$time,
     main = "Hazard Probability Function", xlab = "Tenure",
     ylab = "Hazard Probability", type = 'l')
```

Hazard Functions – R



Hazard Functions – R

```
ggsurvplot(recid_km, data = simple, fun = "cumhaz", conf.int = TRUE,
           palette = "purple", xlab = "Week",
           ylab = "Cumulative Hazard", legend = "none")
```