컴퓨터 통계 방법론 HW4

20180490 이재헌

1. Ch8-Prob2: It is mentioned in Section 8.2.3 that boosting using depth-one trees (or stumps) leads to an additive model: that is, a model of the form

$$f(X) = \sum_{j=1}^{p} f_j(X_j).$$

Explain why this is the case. You can begin with (8.12) in Algorithm 8.2.

A stump has only two terminal nodes(one split). Suppose a single split occurs on the variable X_j (at value s). Then we can simplify write this as follows:

$$f(X) = c_1 * 1(X \in R_1) + c_2 * 1(X \in R_2)$$

$$= \begin{cases} c_1 & if\ X_j < s \\ c_2 & if\ X_j \geq s \end{cases}$$

$$= c_1 * I(X_j < s) + c_2(X_j \geq s)$$

$$where\ c_n\ is\ the\ mean\ response\ value\ for\ the\ observation\ in\ n_{th}\ region.$$

Consider $b = 1,2, \dots, B$ stumps in the boosting model. Each stump has only one split.

Let's say the splits occur on variables $\{X_{j_1}, X_{j_2}, \dots, X_{j_B}\} \in \{X_1, \dots, X_p\}$ with the corresponding values $s_1, s_2, \dots, s_p$. Let $c_b\ and\ d_b$ be the mean response variables in b_{th} region.

Then, $\hat{f}^b(X) = c_b * I(X_{j_b} < s_b) + d_b * I(X_{j_b} \geq s_b)$

Proceeding the Algorithm 8.2,

$$b = 1:$$

$$\hat{f}^1(X) = c_1 * I(X_{j_1} < s_1) + d_1 * I(X_{j_1} \geq s_1)$$

$$\hat{f}(X) \leftarrow \hat{f}(X) + \lambda\hat{f}^1(X) = \lambda\hat{f}^1(X)$$

$$r_i \leftarrow r_i - \lambda\hat{f}^1(X) = y_i - \lambda\hat{f}^1(X)$$

$$b = 2:$$

$$\hat{f}^2(X) = c_2 * I(X_{j_2} < s_2) + d_2 * I(X_{j_2} \geq s_2)$$

$$\hat{f}(X) \leftarrow \hat{f}(X) + \lambda\hat{f}^2(X) = \lambda\hat{f}^1(X) + \lambda\hat{f}^2(X)$$

$$r_i \leftarrow r_i - \lambda\hat{f}^2(X) = y_i - \lambda\hat{f}^1(X) - \lambda\hat{f}^2(X)$$

$$\dots$$

$$b = B:$$

$$\hat{f}^B(X) = c_B * I(X_{j_B} < s_B) + d_B * I(X_{j_B} \geq s_B)$$

$$\hat{f}(X) \leftarrow \hat{f}(X) + \lambda \hat{f}^B(X) = \sum_{b=1}^{B} \lambda \hat{f}^b(X)$$

$$r_i \leftarrow r_i - \lambda \hat{f}^B(X) = y_i - \sum_{b=1}^{B} \lambda \hat{f}^b(X)$$

$$\hat{f}(X) = \sum_{b=1}^{B} \lambda \hat{f}^b(X) = \sum_{b=1}^{B} \lambda c_b * I(X_{j_b} < s_b) + \lambda d_b * I(X_{j_b} \geq s_b) = \sum_{\boldsymbol{j=1}}^{\boldsymbol{p}} \boldsymbol{f_j(X_j)}$$

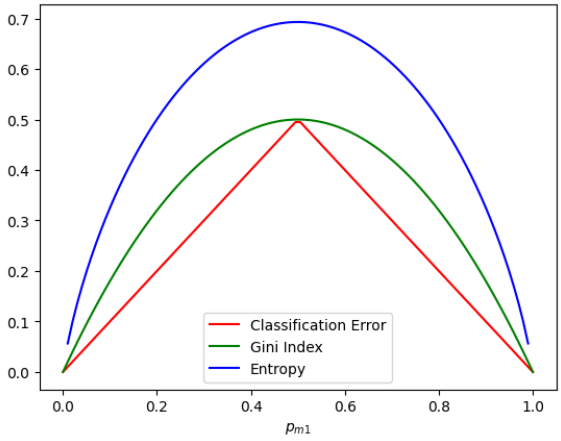
2. Ch8-Prob3: Consider the Gini index, classification error, and entropy in a simple classification setting with two classes. Create a single plot that displays each of these quantities as a function of $\hat{p}_{m1}$. The x-axis should display $\hat{p}_{m1}$, ranging from 0 to 1, and the y-axis should display the value of the Gini index, classification error, and entropy.

$$E = 1 - \max\{\hat{p}_{m1}, \hat{p}_{m2}\}$$

$$G = \hat{p}_{m1}(1 - \hat{p}_{m1}) + \hat{p}_{m2}(1 - \hat{p}_{m2})$$

$$D = -\hat{p}_{m1} \log(\hat{p}_{m1}) - \hat{p}_{m2} \log(\hat{p}_{m2})$$

$$\hat{p}_{m2} = 1 - \hat{p}_{m1}$$



3. Ch8-Prob4: This question relates to the plots in Figure 8.14.

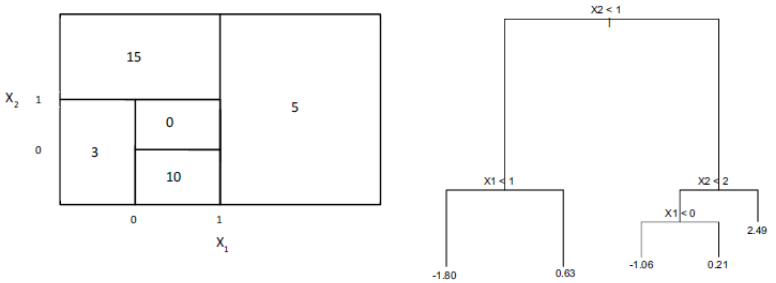
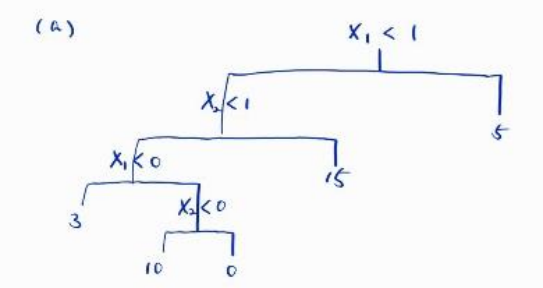
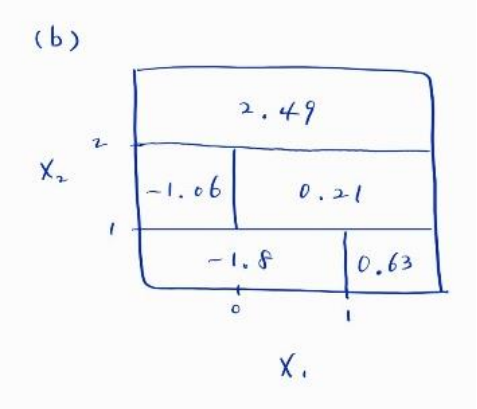


FIGURE 8.14. Left: *A partition of the predictor space corresponding to Exercise 4a.* Right: *A tree corresponding to Exercise 4b.*

(a) Sketch the tree corresponding to the partition of the predictor space illustrated in the left-hand panel of Figure 8.14. The numbers inside the boxes indicate the mean of Y within each region.



(b) Create a diagram similar to the left-hand panel of Figure 8.14, using the tree illustrated in the right-hand panel of the same figure. You should divide up the predictor space into the correct regions, and indicate the mean for each region.



4. Ch8-Prob5: Suppose we produce ten bootstrapped samples from a data set containing red and green classes. We then apply a classification tree to each bootstrapped sample and, for a specific value of X, produce 10 estimates of $P(Class\ is\ Red|X)$:

$$0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, and\ 0.75.$$

There are two common ways to combine these results together into a single class prediction. One is the majority vote approach discussed in this chapter. The second approach is to classify based on the average probability. In this example, what is the final classification under each of these two approaches?

1) Majority Vote Approach

If $P(Class\ is\ Red|X) \geq 0.5$, we classify as Red.

Red = 6, Green = 4. Therefore, the final classification is 'Red'.

2) Average Probability

$$np.mean([0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, 0.75]) = 0.45 < 0.5$$

Therefore, the final classification is 'Green'.

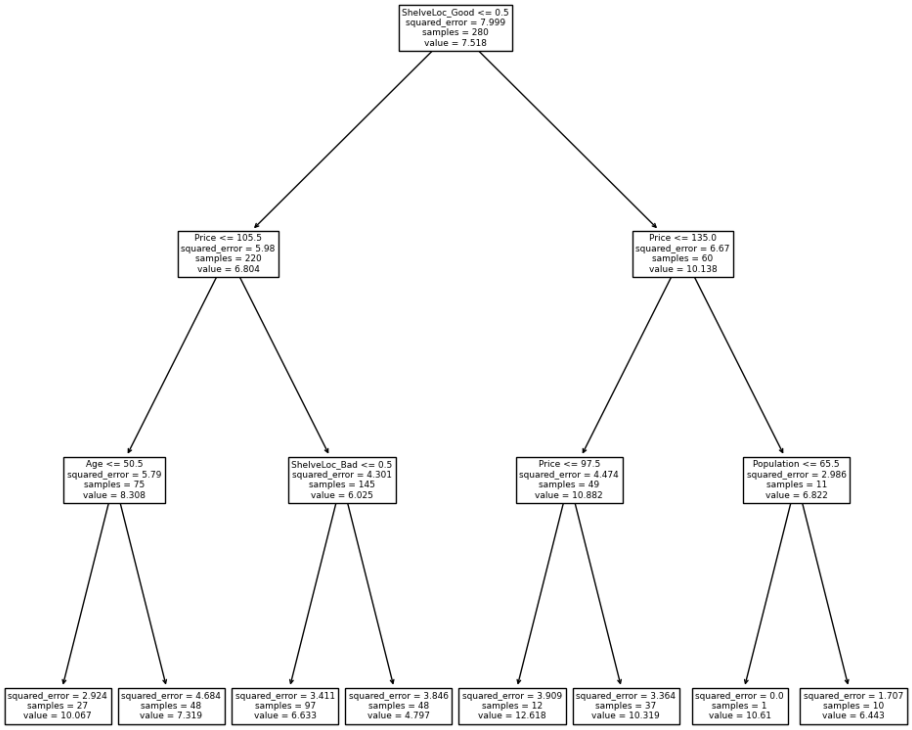
5. Ch8-Prob8: In the lab, a classification tree was applied to the Carseats data set after converting Sales into a qualitative response variable. Now we will seek to predict Sales using regression trees and related approaches, treating the response as a quantitative variable.

   (a) Split the data set into a training set and a test set.

```
[93]  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.3, random_state=1)
```

   (b) Fit a regression tree to the training set. Plot the tree, and interpret the results. What test MSE do you obtain?

```
reg = DTR(max_depth=3)
reg.fit(X_train , y_train)
feature_names = list(X.columns)
ax = subplots(figsize=(12,12))[1]
plot_tree(reg ,
feature_names=feature_names ,
ax=ax);
```



```
# Test MSE
print('Test MSE: ', mean_squared_error(y_test, reg.predict(X_test)))
```

```
Test MSE:  4.940009584939633
```

(c) Use cross-validation in order to determine the optimal level of tree complexity. Does pruning the tree improve the test MSE?

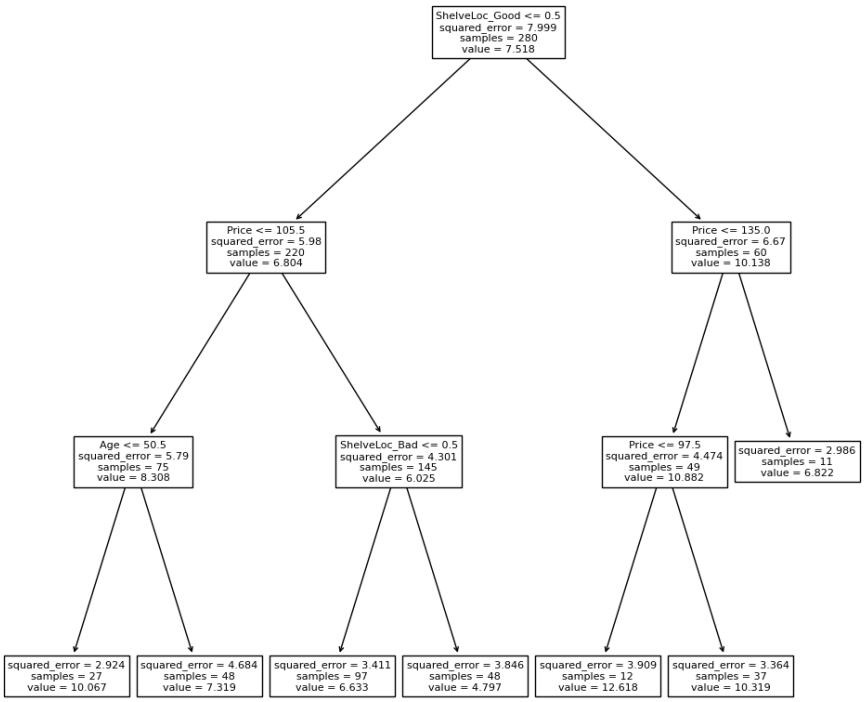
```
ccp_path = reg.cost_complexity_pruning_path(X_train , y_train)
kfold = skm.KFold(5,
shuffle=True ,
random_state =10)
grid = skm.GridSearchCV(reg ,
{'ccp_alpha': ccp_path.ccp_alphas},
refit=True ,
cv=kfold ,
scoring='neg_mean_squared_error')
G = grid.fit(X_train, y_train)
```

```
best_ = grid.best_estimator_
np.mean((y_test - best_.predict(X_test))**2)
```

```
4.8520887547743445
```

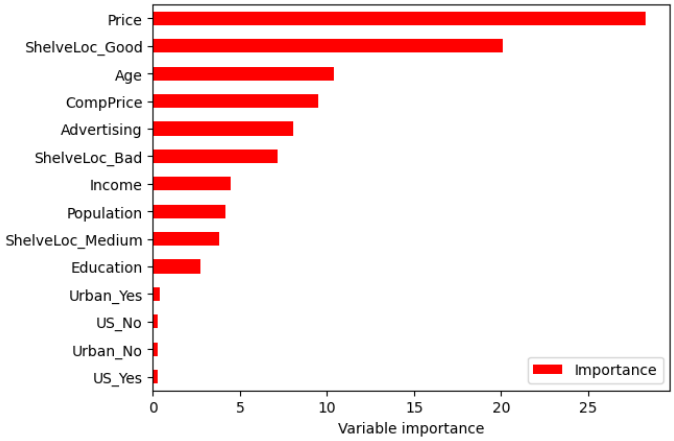
Test MSE = 4.85 < 4.94 (in (a)) is reduced. Pruning improved the test MSE.

Pruned tree is as follows.



(d) Use random forests to analyze this data. What test MSE do you obtain? Use the feature_importance_ values to determine which variables are most important. Describe the effect of m, the number of variables considered at each split, on the error rate obtained.

Variable importance is as follows.



More than half of the variables seem important.

Test MSE with different m is in below.

Number of Features	Test MSE
All	2.813
sqrt	3.160
Log2	3.214

The random forest using all features achieved the best test MSE. This is means almost all features are important. There are 14 features and nearly 10 features are important according to the variable importance plot. $\sqrt{14} \cong 3.74, \log_2 14 = 3.807$. A random forest using only 4 features does not generalize well.

(e) Now analyze the data using BART, and report your results.

```
bart_carseats = BART(random_state=0, burnin=5, ndraw=15)
bart_carseats.fit(X_train , y_train)
```

```
BART
BART(burnin=5, ndraw=15, random_state=0)
```

```
yhat_test = bart_carseats.predict(X_test.astype(np.float32).values)
np.mean((y_test - yhat_test)**2)
```

```
1.6152553699613175
```

Test MSE = 1.615

BART accomplishes much lower test MSE than all the others.

We can check how many times each variable appeared in the collection of trees. This gives a summary similar to the variable importance plot for boosting and random forests.

```
var_inclusion = pd.Series(bart_carseats.variable_inclusion_.mean(0),
index=X.columns)
var_inclusion
```

```
CompPrice          22.066667
Income             17.333333
Advertising        19.800000
Population         19.266667
Price              28.933333
Age                21.066667
Education          18.466667
ShelveLoc_Bad      24.800000
ShelveLoc_Good     23.733333
ShelveLoc_Medium   24.266667
Urban_No           20.066667
Urban_Yes          19.933333
US_No              20.733333
US_Yes             19.400000
dtype: float64
```

6. Ch8-Prob11: This question uses the Caravan data set.

(a) Create a training set consisting of the first 1,000 observations, and a test set consisting of the remaining observations.

```
df = pd.read_csv('/content/drive/MyDrive/ISLP/Caravan.csv')
df = pd.get_dummies(df)
df.head()
```

```
X_train = df.iloc[0:1000,:]
X_test = df.iloc[1000:,:]
X_train.head()
```

```
X_test.head()
```

	MOSTYPE	MAANTHUI	MGEMOMV	MGEMLEEF	MOSHOOFD	MGO
1000	40	1	5	3	10	
1001	26	1	1	5	6	
1002	10	1	4	2	3	
1003	38	1	2	2	9	
1004	39	2	3	3	9	

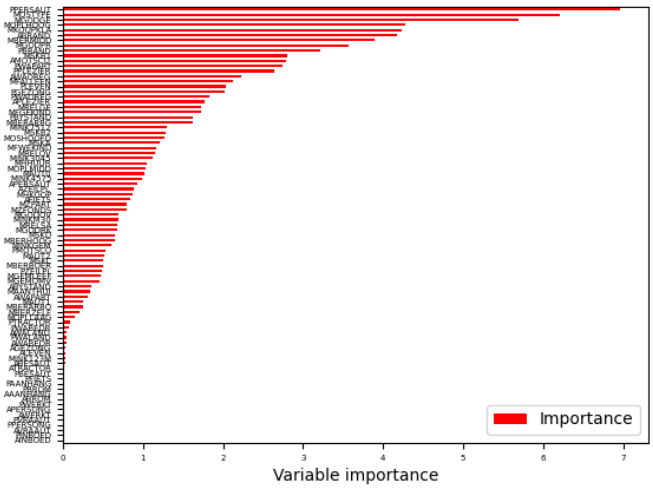
(b) Fit a boosting model to the training set with Purchase as the response and the other variables as predictors. Use 1,000 trees, and a shrinkage value of 0.01. Which predictors appear to be the most important?

```
reg = ensemble.GradientBoostingRegressor(**params)
reg.fit(X_train_new, y_train)

mse = mean_squared_error(y_test, reg.predict(X_test_new))
print("The mean squared error (MSE) on test set: {:.4f}".format(mse))

y_pred = reg.predict(X_test_new)
```

```
The mean squared error (MSE) on test set: 0.0615
```

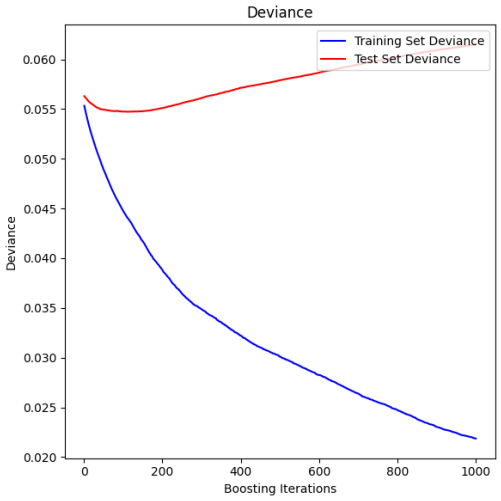
- Test MSE = 0.0615

Variable importance is as follows.



The three most important variables are 'PPERSAUT', 'MOSTYPE' and 'MGODGE'.

The training and test error with respect to the boosting iterations is as follows.



(c) Use the boosting model to predict the response on the test data. Predict that a person will make a purchase if the estimated probability of purchase is greater than 20%. Form a confusion matrix. What fraction of the people predicted to make a purchase do in fact make one? How does this compare with the results obtained from applying KNN or logistic regression to this data set?

Confusion matrix with Boosting is as follows.

```
confusion_matrix(y_test, y_pred_bst)
```

```
array([[4224,  309],
       [ 244,   45]])
```

Purchase probability with different regressions is as follows.

Regression Type	Probability to purchase
Boosting Tree	0.0734
Linear Regression	0.0336
KNN(K=5)	0.2648

Linear Regression predicts the lowest probability while KNN predicts the highest probability. Boosting predicts probability that is in between linear regression and KNN.

7. Ch11-Prob2: We conduct a study with n = 4 participants who have just purchased cell phones, in order to model the time until phone replacement. The first participant replaces her phone after 1.2 years. The second participant still has not replaced her phone at the end of the two-year study period. The third participant changes her phone number and is lost to follow up (but has not yet replaced her phone) 1.5 years into the study. The fourth participant replaces her phone after 0.2 years. For each of the four participants (i = 1, . . . , 4), answer the following questions using the notation introduced in Section 11.1:

   (a) Is the participant's cell phone replacement time censored?

   $$i = 1,4 : \ not\ censored./\ i = 2,3 : \ censored$$

   (b) Is the value of c_i known, and if so, then what is it?

   $$c_2 = 2, \qquad c_3 = 1.5$$

   (c) Is the value of t_i known, and if so, then what is it?

   $$t_1 = 1.2, \qquad t_4 = 0.2$$

   (d) Is the value of y_i known, and if so, then what is it?

   $$y_1 = 1.2, \qquad y_2 = 2, \qquad y_3 = 1.5, \qquad y_4 = 0.2$$

   (e) Is the value of δ_i known, and if so, then what is it?

   $$\delta_1 = \delta_4 = 1, \qquad \delta_2 = \delta_3 = 0$$

8. Ch11-Prob3: For the example in Exercise 2, report the values of $K, d_1, \ldots, d_K, r_1, \ldots, r_K, and\ q_1, \ldots, q_K$, where this notation was defined in Section 11.3.

   $$K = 2, \qquad d_1 = 0.2, \qquad r_1 = 4, \qquad q_1 = 1, \qquad d_2 = 1.2, \qquad r_2 = 3, \qquad q_2 = 1$$

9. Ch11-Prob4: This problem makes use of the Kaplan-Meier survival curve displayed in Figure 11.9. The raw data that went into plotting this survival curve is given in Table 11.4. The covariate column of that table is not needed for this problem.

Observation (Y)	Censoring Indicator (δ)	Covariate (X)
26.5	1	0.1
37.2	1	11
57.3	1	-0.3
90.8	0	2.8
20.2	0	1.8
89.8	0	0.4

TABLE 11.4. *Data used in Exercise 4.*

(a) What is the estimated probability of survival past 50 days?

$$S(26.5) = \frac{5-1}{5} = 0.8, \qquad S(37.2) = S(26.5) * \frac{4-1}{4} = 0.6$$

$$S(50) = S(37.2) = 0.6$$

(b) Write out an analytical expression for the estimated survival function. For instance, your answer might be something along the lines of

$$\widehat{S}(t) = \begin{cases} 0.8 & \text{if } t < 31 \\ 0.5 & \text{if } 31 \leq t < 77 \\ 0.22 & \text{if } 77 \leq t. \end{cases}$$

$$\hat{S}(t) = \begin{cases} 1 & if \ \ t < 26.5 \\ 0.8 & if \ \ 26.5 \leq t < 37.2 \\ 0.6 & if \ \ 37.2 \leq t < 57.3 \\ 0.4 & if \ \ t \geq 57.3 \end{cases}$$

10. Ch11-Prob5: Sketch the survival function given by the equation

$$\widehat{S}(t) = \begin{cases} 0.8 & \text{if } t < 31 \\ 0.5 & \text{if } 31 \leq t < 77 \\ 0.22 & \text{if } 77 \leq t. \end{cases}$$

Your answer should look something like Figure 11.9.

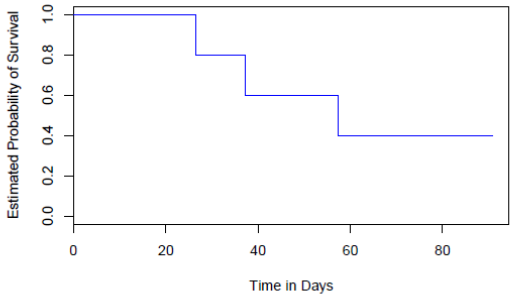
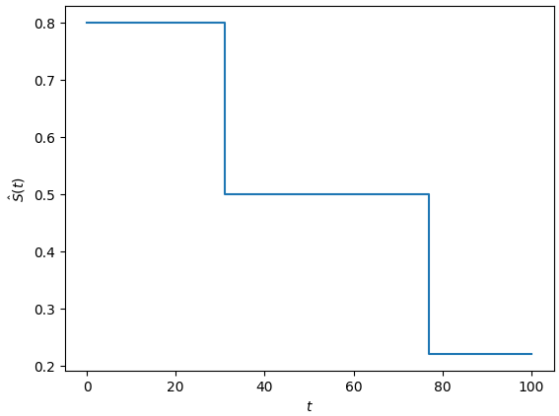


FIGURE 11.9. *A Kaplan-Meier survival curve used in Exercise 4.*

Answer:



11. Ch11-Prob7: In this problem, we will derive (11.5) and (11.6), which are needed for the construction of the log-rank test statistic (11.8). Recall the notation in Table 11.1.

    (a) Assume that there is no difference between the survival functions of the two groups. Then we can think of q_{1k} as the number of failures if we draw r_{1k} observations, without replacement, from a risk set of r_k observations that contains a total of q_k failures. Argue that q_{1k} follows a hypergeometric distribution. Write the hypergeometric distribution parameters of this distribution in terms of r_{1k}, r_k, and q_k.

    A random variable X follows the hypergeometric distribution if its probability mass function (pmf) is given by

    $$p_X(k) = \Pr(X = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}},$$

    $P(q_{1k} = i) = \frac{\binom{r_{1k}}{i}\binom{r_k - r_{1k}}{q_k - i}}{\binom{r_k}{q_k}}$. Therefore, q_{1k} follows a hypergeometric distribution.

    (b) Given your previous answer, and the properties of the hypergeometric distribution, what are the mean and variance of q_{1k}? Compare your answer to (11.5) and (11.6).

    $$E[q_{1k}] = q_k * \frac{r_{1k}}{r_{1k} + r_{2k}}, \qquad Var[q_{1k}] = \frac{q_k r_{1k} r_{2k}}{r_k^2}\left(1 - \frac{q_k - 1}{r_{1k} + r_{2k} - 1}\right)$$

    $$E[q_{1k}] = \frac{r_{1k} q_k}{r_k}, \qquad Var[q_{1k}] = \frac{q_k \left(\frac{r_{1k}}{r_k}\right)\left(1 - \frac{r_{1k}}{r_k}\right)(r_k - q_k)}{r_k - 1} \quad (11.5 \text{ \& } 11.6)$$

    $$Since\ r_{1k} + r_{2k} = r_k\ , they\ are\ equivalent.$$

12. Ch11-Prob8: Recall that the survival function $S(t)$, the hazard function $h(t)$, and the density function $f(t)$ are defined in (11.2), (11.9), and (11.11), respectively. Furthermore, define $F(t) = 1 - S(t)$. Show that the following relationships hold:

    $$f(t) = \frac{dF(t)}{dt}, \qquad S(t) = exp(-\int_0^t h(u)du\ )$$

    $$\frac{dF(t)}{dt} = \lim_{\Delta t \to 0} \frac{F(t + \Delta t) - F(t)}{\Delta t} = \frac{S(t) - S(t + \Delta t)}{\Delta t} = \frac{P(T > t) - P(T > t + \Delta t)}{\Delta t}$$
    $$= \frac{P(t < T \le t + \Delta t)}{\Delta t} = f(t)$$

    $$-\int_0^t h(u)du = -\int_0^t \frac{f(u)}{S(u)}du = \int_0^t \frac{S'(u)}{S(u)}du = \log(S(t)) - \log(S(0)) = \log(S(t))$$

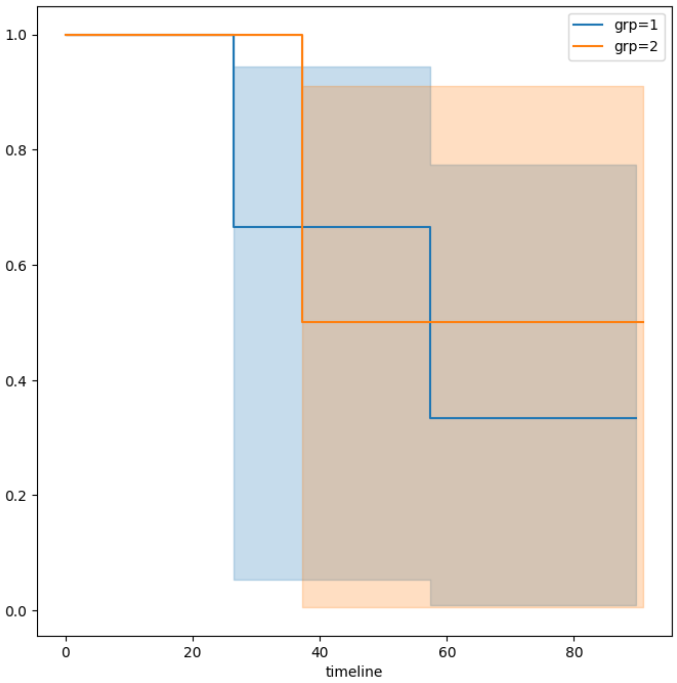
    $$Therefore, \qquad exp\ (-\int_0^t h(u)du\ ) = S(t)$$

13. Ch11-Prob11: This exercise makes use of the data in Table 11.4.

(a) Create two groups of observations. In Group 1, $X < 2$, whereas in Group 2, $X \geq 2$. Plot the Kaplan-Meier survival curves corresponding to the two groups. Be sure to label the curves so that it is clear which curve corresponds to which group. By eye, does there appear to be a difference between the two groups' survival curves?

```
df
```

	time	status	X	grp_ind
0	26.5	1	0.1	1
1	37.2	1	11.0	2
2	57.3	1	-0.3	1
3	90.8	0	2.8	2
4	20.2	0	1.8	1
5	89.8	0	0.4	1



By eye, there appears to be a difference between the two groups' survival curves.

(b) Fit Cox's proportional hazards model, using the group indicator as a covariate. What is the estimated coefficient? Write a sentence providing the interpretation of this coefficient, in terms of the hazard or the instantaneous probability of the event. Is there evidence that the true coefficient value is non-zero?

```
[47] coxph = CoxPHFitter # shorthand
     model_df = MS(['time', 'status', 'grp_ind'],intercept=False).fit_transform(df)
     cox_fit = coxph().fit(model_df,'time','status')
     cox_fit.summary[['coef', 'se(coef)', 'p']]
```

covariate	coef	se(coef)	p
grp_ind	-0.340143	1.235876	0.783144

Estimated coefficient = -0.34

It means, the lower group indicator is, the longer a person in the group survives.

However, since p value is high, (0.78 >> 0.05) there is no evidence that the true coefficient value is non-zero.

(c) Recall from Section 11.5.2 that in the case of a single binary covariate, the log-rank test statistic should be identical to the score statistic for the Cox model. Conduct a log-rank test to determine whether there is a difference between the survival curves for the two groups. How does the p-value for the log-rank test statistic compare to the p-value for the score statistic for the Cox model from (b)?

As we can see from the results below, the score test from the Cox model is exactly equal to the log rank test statistics. Since the p value is 0.78, there is no evidence of a difference in survival between the two groups.

```
cox_fit.log_likelihood_ratio_test()
```

null_distribution	chi squared
degrees_freedom	1
test_name	log-likelihood ratio test

	test_statistic	p	-log2(p)
0	0.08	0.78	0.36

(Cox model score test)

```
logrank_test(by_grp[1]['time'],by_grp[2]['time'],by_grp[1]['status'],by_grp[2]['status'])
```

t_0	-1
null_distribution	chi squared
degrees_of_freedom	1
test_name	logrank_test

	test_statistic	p	-log2(p)
0	0.08	0.78	0.35

(Log-Rank test)