https://www.kaggle.com/code/pliptor/divide-and-conquer-0-82296

• combine train and test data for joint pre-processing

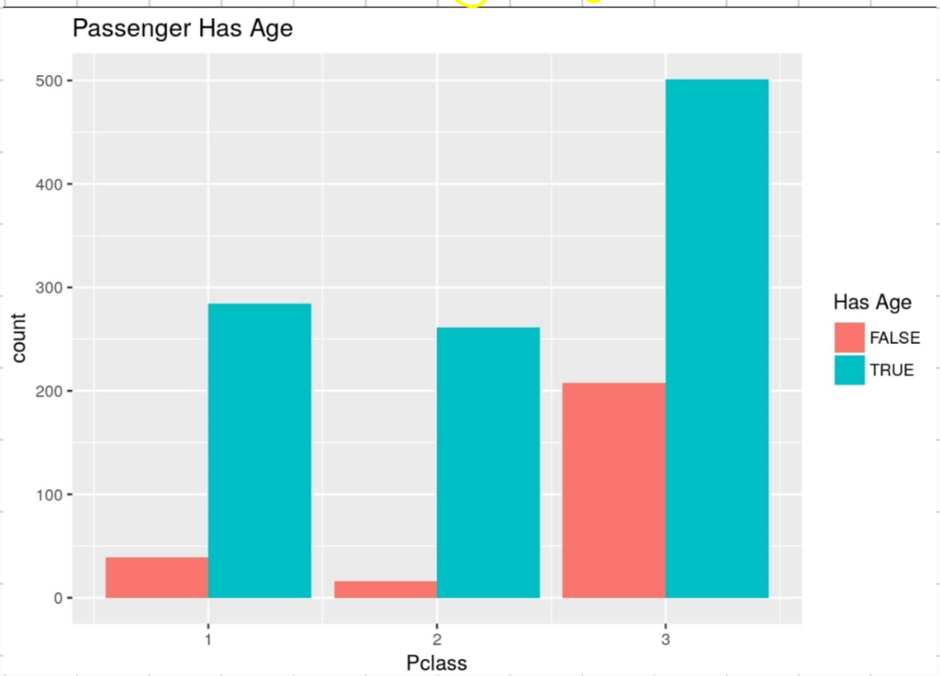
① Handle missing data

The first step will be completing missing data. There are four features with missing data.

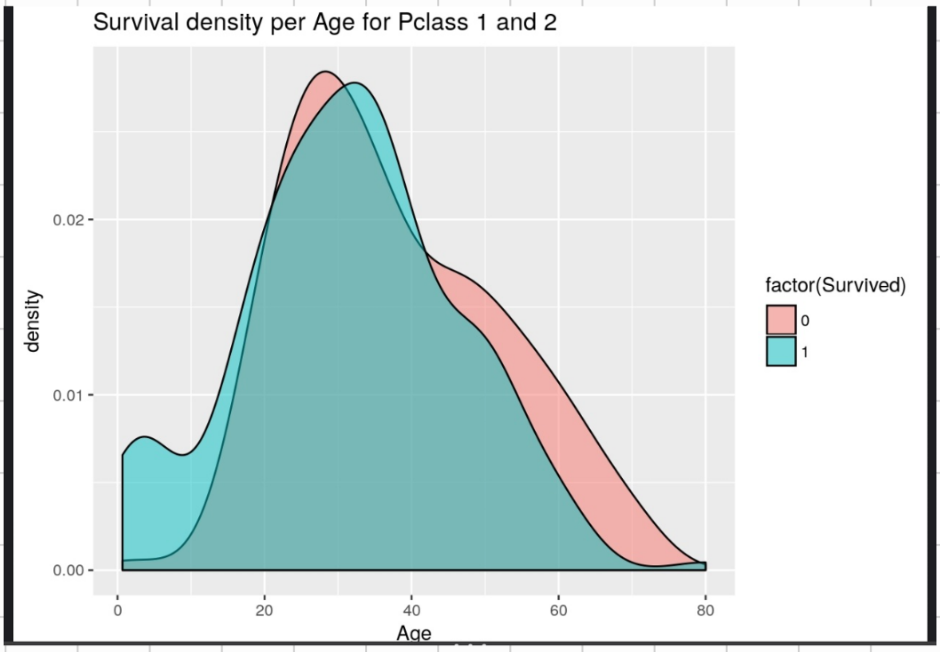
1. Fare values: 1 passenger.
2. Embarked values: 2 passengers.
3. Cabin values: 77% missing.
4. Age values: 20% missing.

→ too many missing vals, attempting to complete it might introduce noise
↳ better to add feature indicating if val is missing

→ most missing age values in PClass = 3



→ dismiss Age for PClass = 3 to avoid noise



add feature Minor for age < 14

The previous graph supports the case that children under around 14 for Pclass 1 and 2 have high likelihood of survival and other age bands are likely to have little impact for predictions. We create the feature $Minor$ that indicates children below 14 in Pclass 1 and Pclass 2.

3.1 Compute frequencies

First we will compute the size of groups for Ticket and FareFac. Here's a fancy way of how we may compute frequencies of features by group in R. These will be used to determine group sizes.

Code

3.2 Family

Here a Surname feature is engineered from the Name feature. It will be used later as one of the group indicators.

Code

3.3 Fares

An interesting characteristic of the fare prices in this data set is that they are very granular. They are so finely granulated some obscure potential groups would be left unnoticed without using it. For example, only two passengers paid the exact amount of 6.75 for their fare, embarked at the same port, had ticked numbers very close, etc. Perhaps identifying these tiny groups gives us an edge for an extra point or two.

Code

```
##   PassengerId Survived Pclass                         Name    Sex Age
## 1         144        0      3          Burke, Mr. Jeremiah   male  19
## 2         655        0      3 Hegarty, Miss. Hanora "Nora" female  18
##   SibSp Parch Ticket Fare Cabin Embarked FareFac Minor TFreq FFreq Surname
## 1     0     0 365222 6.75              Q    6.75     0     1     2   Burke
## 2     0     0 365226 6.75              Q    6.75     0     1     2 Hegarty
```

Is this a young couple? Note that if it is, it would had been undetected by typical procedures to identify "families". This example is also a remarkable demonstration of why achieving a score of 100% is very unlikely. We have a young female in Pclass=3 with no relatives that didn't make it to the survived list. Let's list others with a similar profile:

3.4 Finding groups

We now assign group identifications (GID) to each passenger. The assignment follows the following rules:

1. The maximum group size is 11.
2. First we look for families by Surname and break potentially identical family names by appending a family size.
3. Single families by the above rule are labeled 'Single'.
4. Look at the 'Single' group and assign a GID to those that share a Ticket value.
5. Look at the 'Single' group and assign a GID to those that share a Fare value.

4 Engineer SLogL feature

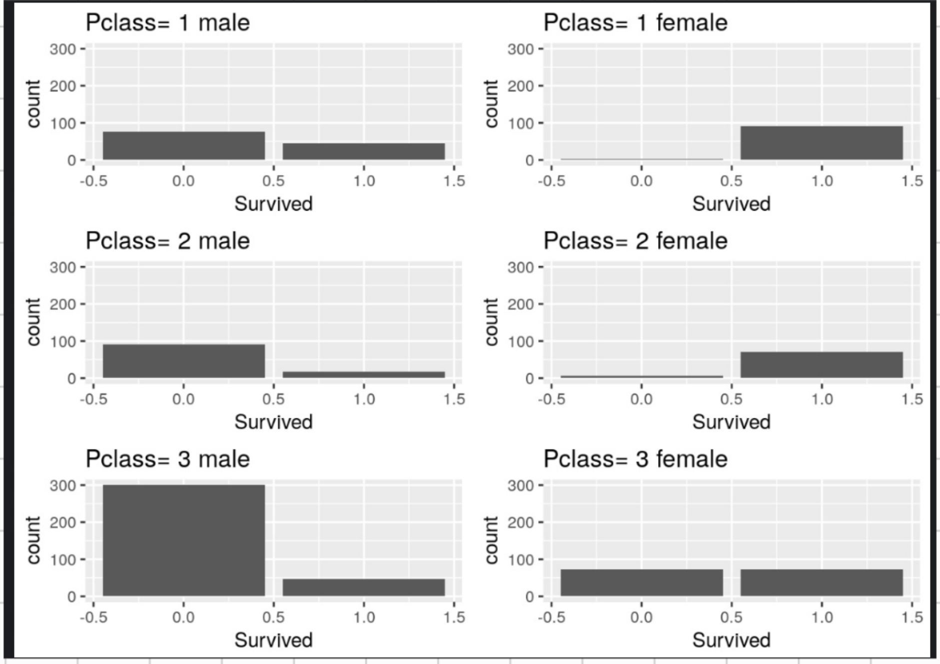
Secret sauce #2: Engineer a log likelihood ratio survival feature (SLogL). The idea is to consolidate all features into a single number indicative of Survival. Log likelihood ratio is a transformation from a binary random variable such as Survival to a point in the real line. SLogL gets bigger when survival is likely and gets smaller (negative) when survival is less likely. SLogL is zero when survival is fifty-fifty. The toss of a fair coin has a log likelihood ratio of zero.

Say you have this binary random variable (Survived) and there are multiple "features" that affect it. There is an underlying assumption that the features must be independent (one reason why I make an effort not to bring too many features unless needed because I know that it would violate the theory otherwise. Some of them are highly correlated. Sex and (Mr, Mrs) for those that process **Title** are examples. Otherwise you'll start double counting (overfitting) unless you take other preventive measures.

Say then you have three independent features A, B, and C that influence Survived. Feature A says that probability of survival is PA (and by exclusion death is 1-PA). Then the log likelihood contribution to SLogL by A is computed by SLogA = log(PA/(1-PA)). From the assumption of feature independence and the definition of log likelihood, SLogL becomes SLogA + SLogB + SlogC. In other words, we add the log likelihood ratio contributions of each of the independent features. In a real world, the features may be correlated. In this dataset, if A is Sex and B is Title you'll have twice the proper contribution to SLogL with respect to Sex.

More information about log likelihood ratio can be found here.

Previously I wrote the method is related to logistic regression (mainly because of the logit function). However, Chris Deotte pointed out in the comments section that this process is more akin to naive Bayes including detailed formulas. Thanks Chris.



1. The fate of passengers in Pclass 2 is almost certain: females almost all survive, males almost all perish.
2. Pclass 1 males are not as lucky as in Pclass 2 but females almost all survive.
3. The luck for females in Pclass 3 is mostly a flip of a coin, close to 50%. **I believe this is the one of the strongest indicators for the limits of achivable score in this dataset.**
4. Males in Pclass 1 has survival rate that is higher than others. **I believe this brings difficulties similarly as in the case above because the probability of survival gets closer to 50%**

From items 3 and 4, I believe that concentrating efforts on them might produce some extra points. This kernel doesn't make direct effort towards that but it is an idea to have in mind.

Now we compute the log likelihood ratio of survival for each of the 6 areas in the grid. We use `dplyr`'s grouping by multiple columns using `.dots`.