## 1

Running speed is entirely dependent on dataset size. Method B should run faster for small datasets, but as the dataset increases in size, Method B's running time should scale roughly quadratically (see Q3 for an explanation), unlike Method A's, which should scale linearly. Depending on the size of our dataset, this may not matter, but I would guess that as we see tens or hundreds of thousands of observations, Method B will become so memory (and probably time)-intensive that it would be practically infeasible.

---

## 2

Yes.

Method A meets all criteria:

- Includes requisite columns.
- Takes race and sex columns from matching splits of the data, so meets that criterion.
- Filters out all potential matches with the same release polarity, so meets this criterion.

Method B meets all criteria:

- Includes requisite columns.
- Merges on race and sex so meets that criterion.
- Splits the data (before merging) into a dataset of those who have been released and a dataset of those who haven't. Since the outputted dataset is created by merging these two datasets together, this criterion is also met.

---

## 3

Honestly, I don't like either method, since both have serious flaws:

- Method A *iterates* over Pandas dataframes on multiple occasions (this is extremely time inefficient), does not have error handling where it should, and relies on manually written process rather than Pandas methods. Although it is valid, it feels like code that was written by a programmer who knows Python but not Pandas.

- On the other hand, while Method B is more concise and uses the more efficient pandas methods, it uses these methods inefficiently. Specifically, it conducts a many-to-many merge and then trims this down to the required sample size. This many-to-many merge has a memory (and possibly also time) complexity that is roughly quadratic (unlike Method A). For example, in Method B, all Black prisoners are merged to all other Black prisoners, and only then is the sample trimmed down; this operation uses memory equal to the number of Black prisoners squared, and this memory use is unnecessary.

- Note that both methods can also create a pair of a prisoner with themselves. E.g., if prisoner 1 is released the first time they're arrested but not released the second time, Method A and Method B can create a pairing of prisoner 1 with themselves.

If I absolutely had to choose a method, I would choose depending on the size of our dataset: given some trial and error, I'd probably end up using Method B for a dataset with less than ~5,000 observations, and Method A for anything larger.