fn check_candidates

Michael Shoemate

May 9, 2025

This proof resides in **"contrib"** because it has not completed the vetting process.

Proves soundness of `check_candidates` in `mod.rs` at commit f5bb719 (outdated[1]). `check_candidates` raises an error if the discrete quantile candidate set is invalid.

1 Hoare Triple

Precondition

None

Function

```
def validate_candidates(candidates: list[T]):
    if not candidates: #
        raise ValueError("candidates must not be empty")

    i1 = iter(candidates)
    i2 = iter(candidates)
    next(i1)

    for c1, c2 in zip(i1, i2): #
        cmp = c1.partial_cmp(c2)
        if cmp is None or cmp != Ordering.Less:
            raise ValueError("candidates must be non-null and strictly increasing")
```

Postcondition

Theorem 1.1. Candidates must be:

1. non-empty
2. strictly increasing
3. totally ordered

Otherwise the function errors.

Proof. The postconditions can be directly checked:

1. `candidates` is non-empty, by the check on line 2
2. `candidates` is strictly increasing, because there is no window where the left candidate is not less than the right candidate

[1] See new changes with `git diff f5bb719..b9b5a3c rust/src/transformations/quantile_score_candidates/mod.rs`

3. `candidates` is totally ordered, because no comparisons may fail

Therefore the postcondition holds. □