

# fn check\_candidates

Michael Shoemate

May 25, 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Proves soundness of `check_candidates` in `mod.rs` at commit `f5bb719` (outdated<sup>1</sup>). `check_candidates` raises an error if the discrete quantile candidate set is invalid.

## 1 Hoare Triple

### Precondition

None

### Function

```
1 def validate_candidates(candidates: list[T]):
2     if not candidates: #
3         raise ValueError("candidates must not be empty")
4
5     i1 = iter(candidates)
6     i2 = iter(candidates)
7     next(i1)
8
9     for c1, c2 in zip(i1, i2): #
10        cmp = c1.partial_cmp(c2)
11        if cmp is None or cmp != Ordering.Less:
12            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

---

<sup>1</sup>See new changes with `git diff f5bb719..088b177 rust/src/transformations/quantile_score_candidates/mod.rs`

3. `candidates` is totally ordered, because no comparisons may fail  
Therefore the postcondition holds. □