

**Note: this error code is no longer emitted by the compiler.** Patterns used to bind names must be irrefutable. That is, they must guarantee that a name will be extracted in all cases. Instead of pattern matching the loop variable, consider using a `match` or `if let` inside the loop body. For instance:

```
let xs : Vec<Option<i32>> = vec![Some(1), None];
```

```
// This fails because `None` is not covered.
for Some(x) in xs {
    // ...
}
```

Match inside the loop instead:

```
let xs : Vec<Option<i32>> = vec![Some(1), None];
```

```
for item in xs {
    match item {
        Some(x) => {},
        None => {},
    }
}
```

Or use `if let`:

```
let xs : Vec<Option<i32>> = vec![Some(1), None];
```

```
for item in xs {
    if let Some(x) = item {
        // ...
    }
}
```