

[Skip to main content](#)

Stack Overflow

1. [About](#)
2. Products
3. [For Teams](#)
4. [Stack Internal Implement a knowledge platform layer to power your enterprise and AI tools.](#)
5. [Stack Data Licensing Get access to top-class technical expertise with trusted & attributed content.](#)
6. [Stack Ads Connect your brand to the world's most trusted technologist communities.](#)
7. [Releases Keep up-to-date on features we add to Stack Overflow and Stack Internal.](#)
8. [About the company Visit the blog](#)



Loading...

[current community](#)

- [Stack Overflow](#)
- [help chat](#)
- [Meta Stack Overflow](#)

your communities

[Sign up](#) or [log in](#) to customize your list.

[more stack exchange communities](#)

[company blog](#)

3. [Log in](#)
4. [Sign up](#)

[AI Assist is now on Stack Overflow.](#) Start a chat to get instant answers from across the network. Sign up to save and share your chats.

- [Home](#)
- [Questions](#)
- [AI Assist](#)
- [Tags](#)
- [Challenges](#)
- 7. [Chat](#)
- [Articles](#)
- [Users](#)
- [Companies](#)
- [Collectives](#)

Communities for your favorite technologies. [Explore all Collectives](#)

Stack Internal

Stack Overflow for Teams is now called **Stack Internal**. Bring the best of human thought and AI automation together at your work.

[Try for free](#) [Learn more](#)

[Stack Internal](#)

Bring the best of human thought and AI automation together at your work. [Learn more](#)

Collectives™ on Stack Overflow

Find centralized, trusted content and collaborate around the technologies you use most.

[Learn more about Collectives](#)

[Stack Internal](#)

Knowledge at work

Bring the best of human thought and AI automation together at your work.

[Explore Stack Internal](#)

[Writing a function to take iterables of reference and value type](#)

[Ask Question](#)

Asked 8 years ago

Modified [8 years ago](#)

Viewed 423 times

1

I'd like to have a function that takes an iterable and returns its smallest and largest elements. This is part of an exercise in learning Rust, but I'm struggling in being able to handle reference types and value types at the same time.

This is what I have:

```
fn min_max<'a, I, T>(mut iter: I) -> Option<(&'a T, &'a T)>
where
    I: Iterator<Item = &'a T>,
    T: PartialOrd,
{
    let mut min = match iter.next() {
        Some(x) => x,
        // The collection is empty
        None => return None,
    };
    let mut max = min;

    for el in iter {
        if el < min {
            min = el;
        }
        if el >= max {
            max = el;
        }
    }
}

Some((min, max))
```

Then, I give this an iterator over some integers.

```
let nums: [u32; 6] = [4, 3, 9, 10, 4, 3];
if let Some((min, max)) = min_max(nums.iter()) {
    println!("{} {}", min, max);
}
```

This works, and prints 3 10. But then I want to do some operations on the numbers before I compute the minimum and maximum, like a `map` and/or a `filter`.

```
let doubled = nums.iter().map(|x| 2 * x);
if let Some((min, max)) = min_max(doubled) {
    println!("{} {}", min, max);
}
```

This gives a compiler error:

```
error[E0271]: type mismatch resolving `<[closure@src/main.rs:31:35: 31:44] as std::ops::FnOnce<(&u32,)>>::Output == &_
--> src/main.rs:32:31
|
32 |     if let Some((min, max)) = min_max(doubled) {
|           ^^^^^^ expected u32, found reference
|
|= note: expected type `u32`
      found type `&_`
= note: required because of the requirements on the impl of `std::iter::Iterator` for `std::iter::Map<std::slice::Iter<'_, u32>`
= note: required by `min_max`
```

This confused me, because if `nums.iter()` works as an argument, why shouldn't `nums.iter().map(...)`?

I understand the error message in principle: my array is of `u32`, not `&u32`, whereas my function requires `Iterator::Item` to be of type `&'a T`. But then I don't get why it errors only on the second sample (using `.iter().map()`) and not on the first (just `.iter()`).

I've made a [playground](#) with this example and a commented out example where I construct an iterable of integers from a string. This fails in exactly the same way as the second example above (and is closer to my actual use case).

```
let s = "4 3 9 10 4 3";
let parsed = s.split(" ").map(|x| x.parse::<u32>().unwrap());

if let Some((min, max)) = min_max(parsed) {
    println!("{} {}", min, max);
}
```

• [rust](#)

[Share](#)

[Improve this question](#)

Follow

[edited Dec 3, 2017 at 14:38](#)

[Shepmaster](#)

439k116116 gold badges1.3k1.3k silver badges1.5k1.5k bronze badges

asked Dec 3, 2017 at 14:27

[Alex](#)

9,50111 gold badge4141 silver badges4545 bronze badges

[Add a comment](#) | .

2 Answers

Sorted by: [Reset to default](#)

Highest score (default) Trending (recent votes count more) Date modified (newest first) Date created (oldest first)

3

I'd like to have a function that takes an iterable and returns its smallest and largest elements.

Use [Itertools::minmax](#).

handle reference types and value types at the same time.

You don't **need** to — references to numbers can also be compared:

```
fn foo(a: &i32, b: &i32) -> bool {
    a < b
}
```

In your case, remember that a value and a reference to that value are **different types**. That means you can accept an iterator of any type so long as the yielded values are comparable, and this includes both references and values, as requested:

```
fn min_max<I>(mut iter: I) -> Option<(I::Item, I::Item)>
where
    I: Iterator,
    I::Item: Clone + PartialOrd,
{
    let mut min = match iter.next() {
        Some(x) => x,
        // The collection is empty
        None => return None,
    };
    let mut max = min.clone();

    for el in iter {
        if el < min {
            min = el;
        } else if el >= max {
            max = el;
        }
    }
    Some((min, max))
}
```

I chose to add the `Clone` bound although to be more true to your original I could have used the `Copy` bound. Itertools returns an enum to avoid placing any restrictions on being able to duplicate the value.

This works with all three of your examples:

```

fn main() {
    let nums: [u32; 6] = [4, 3, 9, 10, 4, 3];
    if let Some((min, max)) = min_max(nums.iter()) {
        println!("{} {}", min, max);
    }

    let doubled = nums.iter().map(|x| 2 * x);
    if let Some((min, max)) = min_max(doubled) {
        println!("{} {}", min, max);
    }
}

let s = "4 3 9 10 4 3";
let parsed = s.split(" ").map(|x| x.parse::<u32>().unwrap());

if let Some((min, max)) = min_max(parsed) {
    println!("{} {}", min, max);
}
}

```

3 10
6 20
3 10

my array is of u32, not &u32, whereas my function requires `Iterator::Item` to be of type `&'a T`. But then I don't get why it errors only on the second sample (using `.iter().map()`) and not on the first (just `.iter()`).

Because [iterating over an array returns references](#). By using `map`, you are changing the type of the iterator's item from `&i32` to `i32`. You could have also [chosen to adapt the first call to return values](#).

[Share](#)

[Improve this answer](#)

Follow

[edited Dec 3, 2017 at 18:06](#)

answered Dec 3, 2017 at 14:46

[Shepmaster](#)

439k116116 gold badges1.3k1.3k silver badges1.5k1.5k bronze badges

Sign up to request clarification or add additional context in comments.

Comments

Add a comment

0

You have a type mismatch problem because the `.iter()` call produces a "slice" iterator (`Iterator with Item = &T`), but the `.map(|x| 2 * x)` is a iterator adaptor, the call of which produces a new "value" iterator (`Iterator with Item = T`). These values must necessarily be stored in memory before we can get them "slice", because we can only get a reference to the value that is already stored somewhere in the memory. Therefore, we need to collect the result of the `map` function before we can get an iterator with references to the values it returns:

```

let doubled: Vec<_> = nums.iter().map(|x| 2 * x).collect();

if let Some((min, max)) = min_max(doubled.iter()) {
    println!("{} {}", min, max);
}

```

For more details, see chapter [13.2 Iterators](#) of The Rust Programming Language book.

[Share](#)

[Improve this answer](#)

Follow

answered Dec 3, 2017 at 17:05

[freecoder](#)

14122 silver badges88 bronze badges

2 Comments

Add a comment

jilles

[jilles Over a year ago](#)

This will work but it unnecessarily allocates a vector with the doubled elements.

2017-12-03T22:26:08.56Z+00:00

1

Reply

- Copy link

freecoder

[freecoder Over a year ago](#)

@jilles Your function `min_max` returns references: `Option<(&'a T, &'a T)>`. Where in the case of doubled map adaptor you will store the min and max values?

2017-12-05T10:47:03.943Z+00:00

0

Reply

- Copy link

Your Answer

Thanks for contributing an answer to Stack Overflow!

- Please be sure to *answer the question*. Provide details and share your research!

But avoid ...

- Asking for help, clarification, or responding to other answers.
- Making statements based on opinion; back them up with references or personal experience.

To learn more, see our [tips on writing great answers](#).

[Sign up](#) or [log in](#)

Sign up using Google

Sign up using Email and Password

Submit

Post as a guest

Name

Email

Required, but never shown

Post as a guest

Name

Email

Required, but never shown

[Post Your Answer](#) [Discard](#)

Start asking to get answers

Find the answer to your question by asking.

[Ask question](#)

Explore related questions

- [rust](#)

See similar questions with these tags.

- The Overflow Blog
 - [Introducing Stack Overflow AI Assist—a tool for the modern developer](#)
 - [Treating your agents like microservices](#)
- Featured on Meta
 - [Chat room owners can now establish room guidelines](#)
 - [AI Assist is now available on Stack Overflow](#)
 - [Policy: Generative AI \(e.g., ChatGPT\) is banned](#)

Linked

[59](#)

[How to iterate over and filter an array?](#)

[52](#)

[Iterating over a slice's values instead of references in Rust?](#)

Related

[0](#)

[Generalize over iterators over references and iterators over values](#)

[124](#)

[How to write a Rust function that takes an iterator?](#)

[3](#)

[Pass an iterator of structs to a function accepting references in Rust?](#)

[5](#)

[How to properly pass Iterators to a function in Rust](#)

[2](#)

[Implement Iterator in Rust](#)

[5](#)

[How to create an `Iterable` trait for references in Rust?](#)

[4](#)

[How can I pass my mutable references to a function inside iterator loops in Rust?](#)

[3](#)

[Pass an iterable to a function and iterate twice in rust](#)

[4](#)

[How to flatten iterator that uses references](#)

[1](#)

[how to iterate a vector containing references?](#)

Hot Network Questions

- [What is the difference between an ultraviolet limit and ultraviolet stability?](#)
- [What's the point of swapping polarity between mod slots?](#)
- [Is the phrase "murder will out" from Chaucer?](#)
- [Displaying features in a layer of geometrytype = GeometryCollection25D](#)
- [Looking for a proper formal substitute for "quick fix" for formal letter](#)
- [Is Codepage 1252 guaranteed to be available on all Windows systems?](#)
- [variable name change causes segfault](#)
- [Is there a term for forming emotional or relational bonds with AI chatbots? If not, is "cyberpomorphic" a valid neologism?](#)
- [What international legal mechanisms are in place to stop a country from outright expropriating foreign assets?](#)

- [Is zero-point energy a mathematical artifact?](#)
- [Is there a website where we can find the average stipend/salary for the PhD and Postdoc for each country?](#)
- [How to check if two tables are identical?](#)
- [Diagonal Arrows Under an Equation](#)
- [EEPROM Decoupling Capacitor](#)
- [How do I add a text editor to my \(debian-based\) initramfs?](#)
- [Does short range teleporting end floating disk?](#)
- [How do Buddhists interpret the Buddha's explanation of earthquakes in AN 8.70?](#)
- [Curve has weird bumps when triangulated](#)
- [Proportional odds logistic regression for ordered category outcome - how to convert odds ratios to probabilities in this case?](#)
- [The first Pontryagin class of Milnor's example of exotic 7-sphere](#)
- [Is it possible to use LaTeX to create RPG-related PDFs, such as character sheets or adventure modules?](#)
- [Trying to understand UK tax brackets for salary above 150k](#)
- [TV program where real actors interact with 2-Dimensional characters](#)
- ["Save as webpage, HTML Only" Fail while offline, Firefox](#)

[more hot questions](#)

[Question feed](#)

Subscribe to RSS

[Question feed](#)

To subscribe to this RSS feed, copy and paste this URL into your RSS reader.

[Stack Overflow](#)

- [Questions](#)
- [Help](#)
- [Chat](#)

[Business](#)

- [Stack Internal](#)
- [Stack Data Licensing](#)
- [Stack Ads](#)

[Company](#)

- [About](#)
- [Press](#)
- [Work Here](#)
- [Legal](#)
- [Privacy Policy](#)
- [Terms of Service](#)
- [Contact Us](#)
- [Cookie Settings](#)
- [Cookie Policy](#)

[Stack Exchange Network](#)

- [Technology](#)
- [Culture & recreation](#)
- [Life & arts](#)
- [Science](#)
- [Professional](#)
- [Business](#)
- [API](#)
- [Data](#)
- [Blog](#)
- [Facebook](#)
- [Twitter](#)

- [LinkedIn](#)
- [Instagram](#)

Site design / logo © 2025 Stack Exchange Inc; user contributions licensed under [CC BY-SA](#). rev 2025.12.4.37651