Error Logging in R

Using Purr

- Found some useful tools to test functions in R.
- Came across some good links as well linked to this.
- This '{R-bloggers}' link has some good information on this.
- I think I am also liking the chronicler package by '{Bruno Rodrigues}'.

```
#Let's use my package normfluodbf
library(normfluodbf)
fpath <- system.file("extdata", "dat_2.dat", package = "normfluodbf", mustWork = TRUE)

#Using safely
safe_log = purrr::safely(normfluodat)
a = safe_log(dat=fpath, tnp = 3, cycles = 40)
```

```
## The user is advised to input a character vector of rows used
```

```
#Using quietly
quiet_log = purrr::quietly(normfluodat)
b = quiet_log(dat=fpath, tnp = 3, cycles = 40)

# Based on the results I clearly see that quietly captures the message while safely fails to capture th
```

Using TryCatch

- Check out the function I cooked up based on some information from '{Datatechnotes}'. That is a good way to log error files locally and have them in hand for debugging.

Using Chronicler

- Seems to have some benchmarking abilities as well. Did not dive deep into it but like the way it works so far.

```
# Some snippets from Bruno's chronicler found on github
# I am rewriting because that page is found on my iPad

rsqrt = chronicler::record(sqrt)
a = rsqrt(1:5)
a
```

```
## OK! Value computed successfully:
## ---------------
## Just
## [1] 1.000000 1.414214 1.732051 2.000000 2.236068
##
## ---------------
## This is an object of type `chronicle`.
## Retrieve the value of this object with pick(.c, "value").
## To read the log of this object, call read_log(.c).
```

```
#get just the values
chronicler::pick(a,'value')
```

```
## [1] 1.000000 1.414214 1.732051 2.000000 2.236068
```

```
chronicler::pick(a,'log_df')
```

```
## # A tibble: 1 x 11
##   ops_number outcome     `function` arguments message start_time
##        <int> <chr>       <chr>      <chr>     <chr>   <dttm>
## 1          1 OK! Success sqrt       ""        NA      2023-12-25 15:05:35
## # i 5 more variables: end_time <dttm>, run_time <drtn>, g <list>,
## #   diff_obj <list>, lag_outcome <chr>
```

```
#get the log and this gives run time
chronicler::read_log(a)
```

```
## [1] "Complete log:"
## [2] "OK! sqrt() ran successfully at 2023-12-25 15:05:35.699391"
## [3] "Total running time: 0.000649213790893555 secs"
```