Steps:

1. Read data (from simulation or file)
2. Add summary measures in wide form (lags, variance, mean, etc).
3. Wait / simulate the minimal number of measurements needed
4. Using all these measurements, we estimate the density for q_w given w-, (g given g-), and q_y given y-. (If a dist is binary, we don't have to estimate its distribution, but we still can using Oleg's package, for consistency. The same goes for if a variable is discrete.)
5. Now, using these densities, we generate a number of draws, using the scheme you sent earlier, in order to get the counterfactual distribution (j is time of intervention, tau is time of response) :

```
result <- 0
B <- seq(100000)
for b in B
  sample a  W(1) given C_w(1) from q_w
  sample an A(1) given C_a(1) from g_a
  sample a  Y(1) given C_y(1)
  sample a  W(2) given C_w(2) from q_w
  sample an A(2) given C_a(2) from g_a
  sample a  Y(2) given C_y(2)
  .
  .
  .
  sample a  W(j) given C_w(j) from q_w
  sample an A(j) = a (= specified, deterministic intervention)
  sample a  Y(j) given C_y(j)
  .
  .
  .
  sample a  W(tau) given C_w(tau) from q_w
  sample an A(tau) given C_a(tau) from g_a
  result += (sample a  Y(tau) given C_y(tau)) / B
endfor
result
```

   which will give you the response for an *intervention a on time j for an outcome at time tau.*
6. Questions:
   - How do we deal with the initialization? Up to now I just expected that we'd skip the first `Z` measurements, where `Z` is the maximum

number of observations needed for each of the summary measures we include.
- In this scheme of drawing observations we are not including the $g_a^*(a(t)|c(a))/g_a(a(t)|c(a))$

Basic roadmap

1. create structure of package
2. set up automatic compiling and testing
3. Use packages:

- 'R.oxygen'
- 'R.utils' ('Arguments$getNumeric' and the likes, 'throw', use of 'verbose')

```
## Example of verbose
## Argument 'verbose':
verbose <- Arguments$getVerbose(verbose)
verbose <- less(verbose, 10)
verbose && print(verbose, table(U))
verbose && enter(verbose, "Simulated  copy number and expression data
for class 1")
verbose && str(verbose, V)
verbose && exit(verbose)
```

4. Keep in mind that we're interested in oos (online one-step) and otmle (online tmle) estimators

4a. new object S3 structure

4b. high-level description – flow of the application

```
(i) learning step
(ii) targeting step
```

4c. encoding of a generic time series vocabulary:

```
- (W(t), A(t), Y(t)) is the t-th block
- W(t), A(t) and Y(t) are its nodes
- W(t) in WW and Y(t) in YY, WW and YY possibly multi-dim
- A(t) is discrete/binary?
```

4d. simulation (i) characterization of data-generating distribution (ii) simulation under the data-generating distribution (iii) characterization of the parameter of interest - => providing intervention nodes, and the corresponding intervention distributions (iv) evaluation of the parameter of interest

4e. encoding of summary measures of the past of each {W,A,Y}-node

4f. write functions implementing *online* (super-) learning - must rely on *online* "prediction" algorithms - think about H2O…

4g. Monte-Carlo procedures to derive - the estimates of the parameter from the estimated features - the efficient influence curve, used for targeting + CI (see draft)