



IMPROVING CLINICAL TRIAL ENROLLMENT WITH R AND AN ENROLLMENT MODELING R PACKAGE

```
> library(EnrollmentModeling)
```

STEPHEN GORMLEY: CfDA > DATA SCIENCES

DATA SCIENCES, SENIOR MANAGER

PHUSE CONFERENCE, **12 NOVEMBER 2019**



WHAT AM I GOING TO TALK ABOUT?



WHO ARE WE?:

AMGEN: BIOTECHNOLOGY



RESEARCH AND DEVELOPMENT



CENTRE FOR DESIGN AND ANALYSIS (CfDA)



DATA SCIENCES



WHY SHOULD YOU LISTEN TO ME?



**BRIEFLY THE BUSINESS USE CASE(S)
AND
REQUIREMENTS BASED ON THE
BUSINESS USE CASE(S)**

FOUR REQUIREMENTS



The Data Sciences team shall allow key stakeholders during clinical study planning to:

1. Determine predicted stopping times and the probability of successfully achieving enrolment for a study, given a target enrolment duration and the number of planned subjects, given other site and country level constraints.
2. Determine the predicted stopping times in a specific country, given a target enrolment duration and number of subjects planned in the country.
3. Determine the optimal allocation of sites in each country (across the whole study) to minimize cost and/or speed of recruitment (given a number of study, site and country level constraints) for a number of probabilities to enrol by the target enrollment time.
4. Allow enrolment to be re-projected using historical and/or actual data mid-way through the clinical trial.

THE THREE MORE

In order to improve the reliability of the data and the inference drawn from any statistical analysis the Data Sciences team shall:

1. Identify the more appropriate external data repositories and create an automated pipeline to consume this data.
2. Convert the internal and external data repositories, where necessary, into a standard format and store in an internal Clinical Data Lake ("**CDL**") to easily be consumed by all stakeholders.
3. Visualise all findings in an RShiny application that can be deployed centrally and accessed by all stakeholders to answer the business use cases.



INPUT DATA

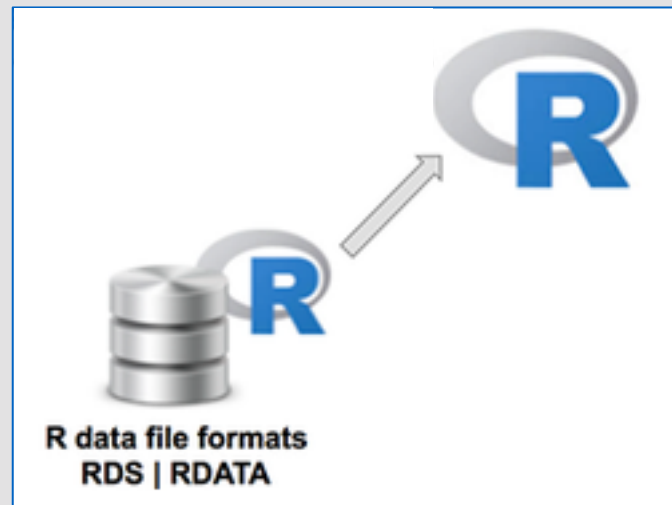
WHAT DATA IS NEEDED?

Country Level (from Historical and Amgen Internal Data)

- An Enrollment Rate (also via ML Algos)
- Study Start Up Quartiles
- Coefficient of Variation
- Minimum Number of Sites and Subjects
- Maximum Number of Sites and Subjects

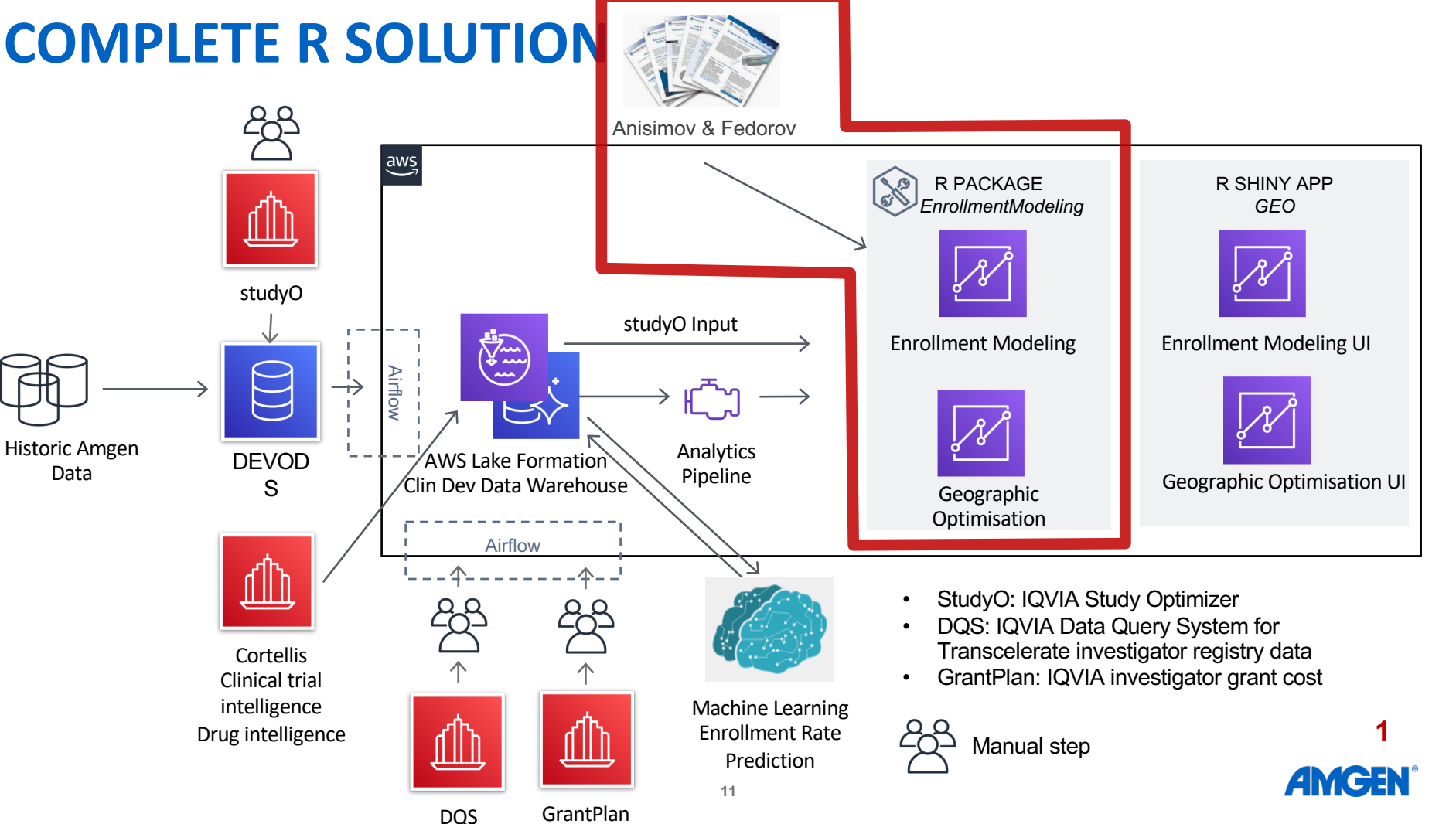
Study Level (from Amgen Internal Data)

- Target Enrollment In Months
- Target Number of Subjects
- Cost (Amgen specific)



OVERALL R (PLUS OTHER) SOLUTION

COMPLETE R SOLUTION



- StudyO: IQVIA Study Optimizer
- DQS: IQVIA Data Query System for Transcelerate investigator registry data
- GrantPlan: IQVIA investigator grant cost

THE R PACKAGE DESIGN:

```
library(EnrollmentModeling)
```

ENROLLMENT MODELING

Source Papers



Anisimov & Fedorov



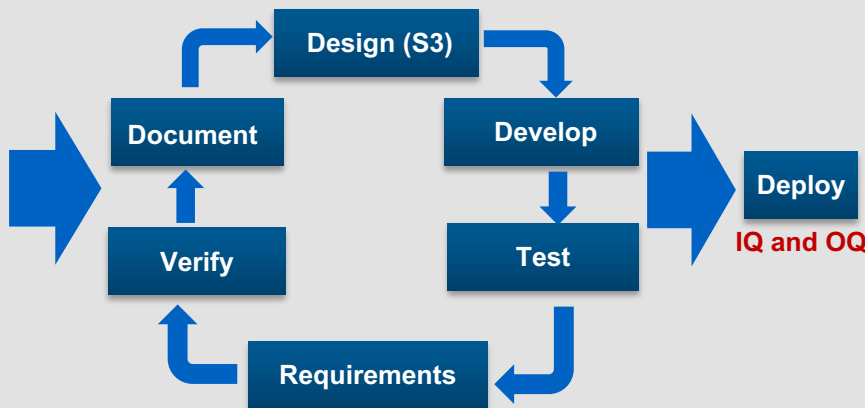
Anisimov

Numerous R Scripts

```
# Estimator for the remaining enrollment in active sites
estimator <- EstimatorforDesign(vePts, Time)
# setup
F[3] <- 5.38934763 200.58222089 R #1018821
## Global input parameters in one pool of sites
a1f <- estpar[1]
b1 <- estpar[2]
n <- estpar[3]

# PREDICTION OF THE REMAINING ENROLLMENT IN ACTIVE SITES
## Bayesian re-estimated parameters (a1f, l, n, 1) for All sites
vecreport <- (vePts ~ a1f)(Time=40)
vecreport <- (vePts ~ a1f)(Time=40)
## mean and variance of global rate in active sites
MeanRate <- sum(vecreport[1:3], SitesAct)
VarRate <- sum(vecreport[4:6], SitesAct)
```

Anisimov & Others



SDLC: AGILE SPRINT DEVELOPMENT

Four Main Outputs:

1. The probability of successfully achieving enrollment **globally** given a target enrollment duration and number of subjects.
2. The probability of achieving enrollment targets in a **specific country** given a target enrollment duration and number of subjects in the country.
3. Optimal site allocation for each country to minimize cost and/or speed of recruitment globally.
4. Re-project enrollment numbers based on real data mid way through the clinical trial.

Key Design Choice

With a large number of complex R scripts all revolving around Enrolment the solution leant itself to an OO solution and the benefits that an OO approach brings (particularly inheritance and polymorphism).

S3 is the primary design choice, a lightweight R OO approach (one of the three OO approaches in R)

A few of the benefits of R Package Development

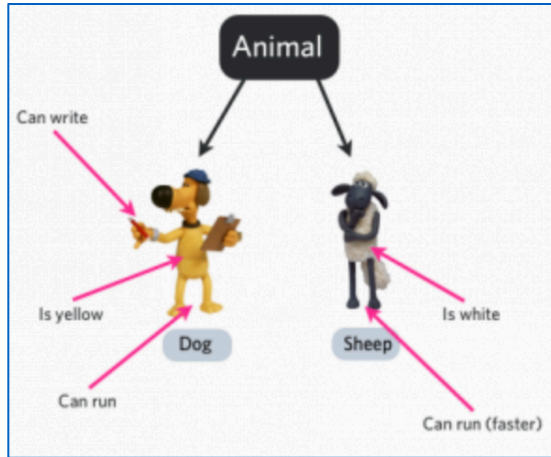
- Development, testing and deploying (read sharing) is straightforward.
- An easy to use test framework.
- Easy to create in-line documentation with auto code completion.
- A large number of CRAN packages with a wide R community.
- Seamless GitLab (and other source control) connectivity.
- R allows for an object oriented approach (i.e. **S3**, S4 and R6)

EnrollmentModeling::Key Packages Used



- [devtools](#): tools to make developing R packages easier.
- [roxygen2](#): In-line documentation for R.
- [testthat](#): Unit testing for R.
- [covr](#): Track and report code coverage for your package.
- [sinew](#): Automate roxygen2 comments.
- [S3](#): One of R's three OO approaches .
- [Rcpp](#): seamless integration of R and C++. Used in EM to reduce the time to optimize.

THE R PACKAGE DESIGN: OO Approaches and Code Layers



THE R PACKAGE OUTPUT:

```
library(EnrollmentModeling)
```

What is it?

- An **R Package** which, for a multicentre clinical trial, predicts the probability of achieving enrollment target times and calculates an optimal site allocation given a variety of country specific constraints. *Further, for event driven trials, predicts the time to the number of events.*

Source Papers and Code

- Code:** The package was built from complex R scripts developed by a CfDA Data Sciences key SME. With raw code developed locally in a large number of R scripts and underlying black box functions.
- Papers:** The raw R scripts are based on enrollment process models developed in Anisimov and Fedorov (2007) and Anisimov (2011).

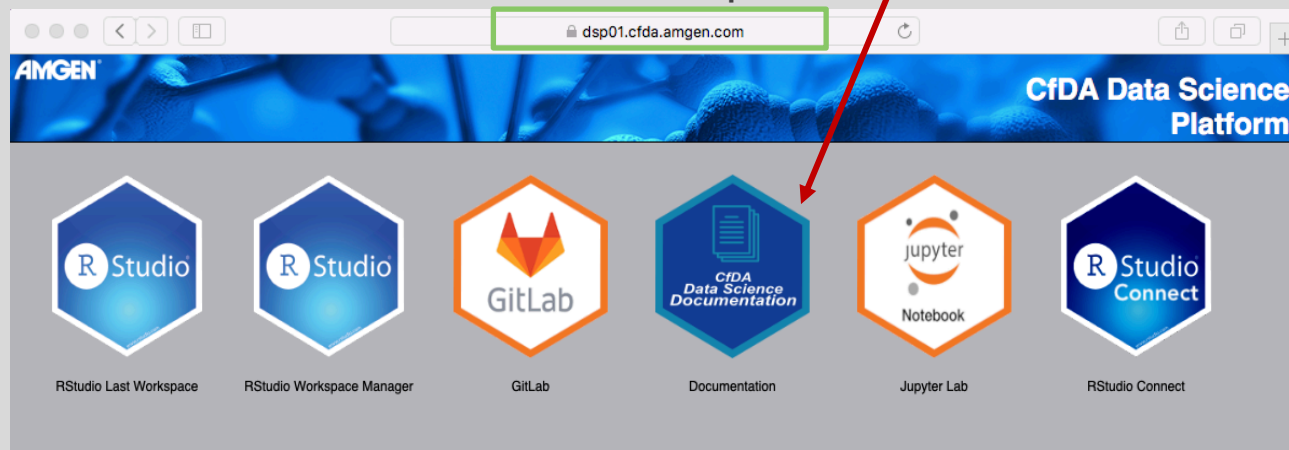
EnrollmentModeling::An R Package

- An R Package developed, tested and documented based on a large number of complex R scripts.
- Deployed on our data science platform:



<https://dsp01.cfda.amgen.com>

- Documentation also accessible on the platform
- Extensive in-line help



Four High Level Requirements

To allow key stakeholders to:

- **First**, determine the probability of successfully achieving enrollment given a target enrollment duration and number of subjects;
- **Secondly**, The probability of achieving enrollment targets in countries and a specific country.
- **Thirdly**, allocate sites optimally for each country to minimize cost and/or speed of recruitment;
- **Fourthly**, re-project enrollment numbers based on real data mid way through the clinical trial

Configuration

A variety of configuration settings can be configured, which run different modelling techniques. For example:

- Fewer or greater than 20 countries.
- Restrictions on min or max number of subjects.
- Model by site categories.
- Different enrollment rates

EnrollmentModeling::Exposed Functions

High Level Requirement 1

- `anEnrollmentFitmodel <- fitEnrollment(anEnrollFitobject, aConfigObject)`

```
> anEnrollmentFitModel
```

```
Planned Subjects: 500
Planned Sites: 80
Summary of predicted stopping time (days):
  Mean Median Lower Upper
   177    183   126   299
```

```
> anEnrollmentFitModel$enrollFitModel$probabilityOfSuccessfullyMeetingTargetEnrolment
[1] 0.998
```

High Level Requirement 2

- `summary(anEnrollmentFitmodel)` for country level summaries

```
> summary(anEnrollmentFitModel, "Italy")
Italy
Planned Subjects: 125
Planned Sites: 80
Summary of predicted stopping time (days):
  Mean Median Lower Upper
   177    201   100   688
```

```
> summary(anEnrollmentFitModel)
```

	Patients	Sites	MeanEnrtime	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95
Global	500	80	177	126	136	150	161	172	183	195	210	231	265	299
Germany	125	20	169	96	110	130	148	168	191	221	261	326	463	649
Belgium	125	20	162	93	106	125	142	161	183	211	249	310	439	614
Italy	125	20	177	100	114	135	155	176	201	232	275	344	490	688
France	125	20	209	115	131	157	181	207	238	277	330	416	600	847

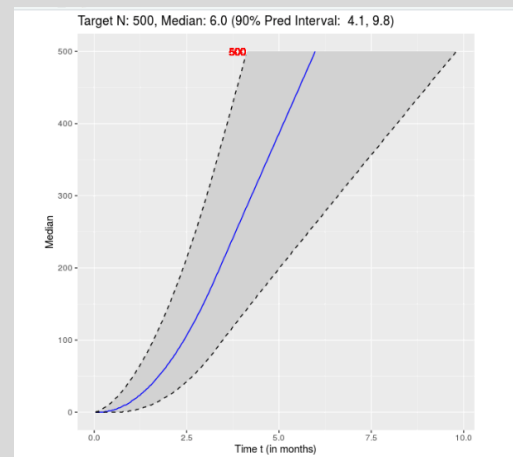
Four High Level Requirements

To allow key stakeholders to:

- **First**, determine the probability of successfully achieving enrollment given a target enrollment duration and number of subjects;
- **Secondly**, The probability of achieving enrollment targets in countries and a specific country.
- **Thirdly**, allocate sites optimally for each country to minimize cost and/or speed of recruitment;
- **Fourthly**, re-project enrollment numbers based on real data mid way through the clinical trial

EnrollmentModeling::Exposed Functions

- `plot(anEnrollmentFitmodel)`



High Level Requirement 3

- `anOptimizedModel <- optimizeEnrollment (anOptimizeEnrollObject, aConfigObject)`
- `print(anOptimizedModel)`

```
> anOptimizationModel
Countries OrigSites MinSites MaxSites P100 P50 P55 P60 P65 P70 P75 P80 P85 P90 P95 P98 Rank
1 Germany 20 0 30 30 22 24 26 28 30 30 30 30 30 30 1
3 Italy 20 0 30 30 0 0 0 0 0 4 7 12 19 30 30 2
2 Belgium 20 0 30 30 0 0 0 0 0 0 0 0 0 1 12 3
4 France 20 0 30 27 0 0 0 0 0 0 0 0 0 0 0 4
Cost at optimum: P100: $1645445, P50: $299217, P55: $326418, P60: $353620, P65: $380821, P70: $408023
, P75: $462756, P80: $503805, P85: $572221, P90: $668004, P95: $835341, P98: $1020377
```

S3: Print, Plot and Summary

Taking advantage of S3 polymorphism the package produces a variety of plot and prints without a function name change. For example:

- `plot(anEnrollFitModel)` and `plot(anEventFitModel)` produce different plots but have the same function call. S3 analyses the object being passed and dispatches appropriately.

Four High Level Requirements

To allow key stakeholders to:

- **First**, determine the probability of successfully achieving enrollment given a target enrollment duration and number of subjects;
- **Secondly**, The probability of achieving enrollment targets in countries and a specific country.
- **Thirdly**, allocate sites optimally for each country to minimize cost and/or speed of recruitment;
- **Fourthly**, re-project enrollment numbers based on real data mid way through the clinical trial

EnrollmentModeling::Exposed Functions

High Level Requirement 4

- `aReprojectModel <- reprojectEnrollment (aReprojectObject, aConfigObject)`

```
> aReprojectionModel
```

```
Planned Subjects: 840  
Number of Subjects Enrolled So Far: 284  
Summary of predicted stopping time (days):  
  Mean Median Lower Upper  
  456   457   407   514
```

```
> aReprojectionModel$enrollReprojectModel$probabilityOfSuccessfullyMeetingTargetEnrolment  
[1] 0.005
```

- `summary(aReprojectModel)` for country level summaries

```
> summary(aReprojectionModel)  
[ 2019-04-11 04:55:41.807 | WARN | ecs) | enrollReprojectModFunctions.R#070 ]: 0.0685
```

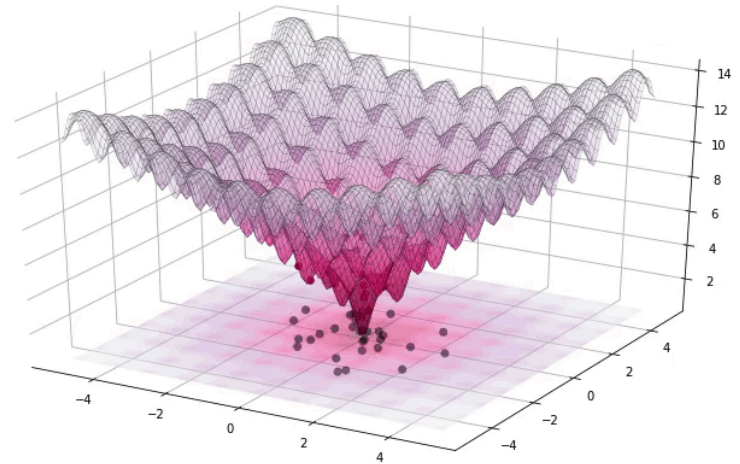
	Patients	Sites	MeanEnrtime	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95	
Global	840	72		456	407	418	431	441	449	457	466	475	486	501	514
United States	76	42		1	1	1	1	1	1	1	1	1	1	1	1
Greece	76	2		1641	976	1099	1276	1426	1572	1726	1899	2110	2396	2880	3375
Poland	76	5		205	147	159	175	187	199	210	222	236	253	279	302
Bulgaria	76	5		559	386	420	467	504	539	573	611	654	709	795	875
Spain	76	3		2140	1170	1343	1600	1825	2052	2297	2581	2937	3438	4329	5297
Czech Republic	76	5		529	370	402	444	479	510	542	576	615	664	740	811
Canada	76	4		2164	1444	1585	1778	1934	2081	2231	2393	2582	2828	3216	3587
Latvia	76	3		1475	924	1029	1175	1297	1415	1536	1671	1832	2046	2397	2745
France	76	1		6114	2768	3325	4217	5065	5977	7034	8348	10001	10001	10001	10001
South Africa	76	1		3328	1507	1810	2296	2757	3254	3829	4545	5515	7020	10001	10001
Portugal	76	1		3527	1597	1918	2433	2922	3448	4058	4816	5845	7440	10001	10001

Validation and Verification

- Extensive beta testing by key end user (i.e. GEO Tool)
- Extensive SME review of model outputs in comparison to output from raw source code.
- Extensive unit testing with `testthat`
- Continuous integration with GitLab CI.

FOUR OPTIMISATION TECHNIQUES:

1. BRUTE FORCE (EXACT) IN R
2. BRUTE FORCE (EXACT) IN C++
3. DEOPTIM
4. GAISL



WHAT DID I SAY?



QUESTIONS?

4	=	21
5	=	27
6	=	36
7	=	49
8	=	57
9	=	76
10	=	?



*"Yes, I think I have good people skills.
What kind of idiot question is that?"*

BACK UP SLIDES AND FOR REFERENCE

CONTENTS

- **Who are we?**
- **Key Takeaways**
- **My High Level Requirements**
- **High Level Overview of Complete R Solution**
- **Detailed view of the R Package Solution**
 - The R Package Design:
 - Overview, dependent packages, S3, functional layers, S3 organisation and Other (subjective) R Choices.
 - The R Package Output:
 - Data Science Platform, Package Calls, Package Outputs
- **Questions?**

CONTENTS

- Who are we?
- Key Takeaways
- My High Level Requirements
- High Level Overview of Complete R Solution
- Detailed view of the R Package Solution
 - The R Package Design:
 - Overview, dependent packages, S3, functional layers, S3 organisation and Other (subjective) R Choices.
 - The R Package Output:
 - Data Science Platform, Package Calls, Package Outputs
- Questions?



R S3: One of R's OO approaches.

Amazon S3: Simple Storage Service

WHY SHOULD YOU LISTEN TO ME?

FOUR MAIN AIMS

1. Give you an understanding of this **Amgen** business use case for the use of R and the high level requirements.
2. Give you an understanding, at a high level, of the full R solution.
3. Give you an understanding, with a more detailed look, of the R package which forms part of the solution.
- ~~4. Brief overview of three/four optimisation techniques.~~

Note: The R solution is based on the enrollment process models and code developed in Anisimov & Fedorov (2007), Anisimov (2011). This presentation will not detail the underlying mathematical methodology that forms the basis of the package.

CfDA Data Science Goals



AUTOMATE
CONVERSION
OF RAW, MESSY
DATA INTO
USABLE DATA



EXTRACT VALUE
THROUGH
ANALYTICS AND
BUSINESS
UNDERSTANDING



CREATE EASILY
COMPREHENSIBLE SUMMARIES
TO INCREASE DATA-INFORMED
DECISION MAKING

Data Science Roles



Visualization & application developers

Operationalize analytical solutions to support broad, data-informed decision making in clinical development



Data engineers

Curate complex data sources to support efficient deployment of analytics

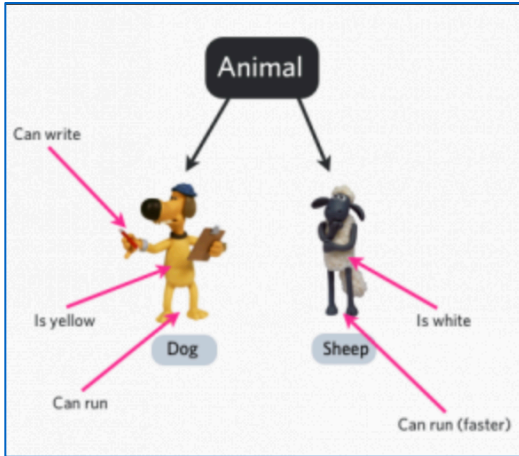


Data Scientists

Develop meaningful analytics to support important clinical development business decisions



THE R PACKAGE DESIGN: OO Approaches and Code Layers



KEY DESIGN ELEMENT: S3 OVER R6

S3 for Four Main Reasons:

- 1) Still get the OO benefit of Polymorphism (i.e. method dispatch).
- 2) Still get the OO benefit of Inheritance via (...) *and alternative ways (though it is a bit tricky)*.
- 3) Ubiquitously used by R contributors, easy to use and for others to comment (*and I just like it*)
- 4) In accordance with the **functional programming paradigm**: when I pass an object into an S3 function it is **not** going to change (i.e. it is immutable).

Note: The Data Sciences team do use R6 quite extensively, as the Java programmers in the team like it. Personally, for the above reasons and also the fact that R6 seems to be a little verbose and is hard to navigate (a disadvantage from a maintenance perspective)

I just prefer S3.



KEY DESIGN ELEMENT: FUNCTION LAYERS

Highest Level: Main Exposed APIS

- Instantiate the S3 Objects, for example:
 - `anEnrollFitObject <- EnrollFit(...)`
 - `anOptimiseObject <- EnrollOptimise(...)`
- Getter functions, for example:
 - `getProbOfSuccess(...)`
 - `getProbMatrix(...)`
- Call the main enrollment modeling functions for example:
 - `fitEnrollmentModel <- fitEnrollment(anEnrollFitObject, config)`
 - `optimiseEnrollmentModel <- optimizeEnrollment(anOptimiseObject, config)`

Second Level Functions

- Dispatch to third level functions based on the configuration, not using S3, using IF ELSE, for example:
 - `fitEnrollment_withMinimumNoMaximum(anEnrollFitObject, config)`
 - `optimizeEnrollment_SmallStudyNoCategories(anOptimiseObject, config)`

KEY DESIGN ELEMENT: FUNCTION LAYERS

Third Level Functions

- Main controller functions, for example:
 - `fitEnrollment_withMinimumNoMaximum(...)`
- Well documented, commented and structured code.

Lowest Level Functions

- Black Box functions from the key SME.
- No refactoring of actual code, no commenting, some roxygen documentation.
- Accepting they work as SME with >40 years experience, Professor of Statistics and 200+ papers.
- Organisation:
 - Prefixed with **function_**, for example, `function_FunEventSensitivity(...)`
 - Suffixed with **_group** if more than one similar function, for example, `function_PoissonGamma_group(...)`

KEY DESIGN ELEMENTS: A FEW OTHER

Other Decisions

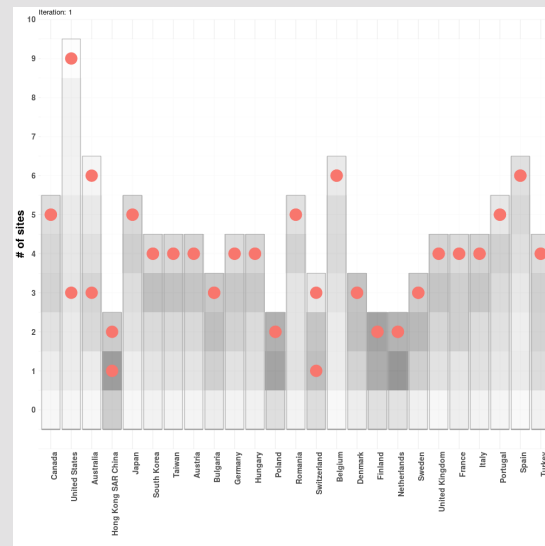
- Conventional use of `aaa.R` (for NAMESPACE creation) and `zzz.R` (`.onLoad`).
- Chained S3 naming conventions, for example:
 - `enrollfit`, `enrollfitmod`, `enrollfitmodsum`
 - `optimizeenroll`, `optimizeenrollmod`, `optimizeenrollmodsum`, `optimizeenrollmodsumspotfire`
- All tests are named `test<functionName>.R`
- All returns were coupled with the objects being passed.
- All returns are complex list objects, of all types.
- R CMD check is run very frequently.
- `camelCase` throughout, but we are now moving to consistently use `snake_case`
- All function calls (except base R) are prefixed with `::` **<- a little verbose when using dplyr**

OPTIMISATION

OPTIMISING ALLOCATION OF STUDY SITES

- Find an allocation of sites $\{N_1, \dots, N_J\}$ in J countries that minimizes:
 - $\text{TrialCost}(N_1 \dots N_J) = \sum_j (\text{PtCost}_j * \text{MeanPts}_j + \text{SiteCost}_j * N_j + \text{CountryCost}_j) * (N_j > 0)$
- Given the probability to enroll by the planned date meets a minimum PoS threshold:
 - $\text{Pr}(\text{EnrollmentTime}(N_1, \dots, N_J, \text{Cap}_j) \leq T_{\text{plan}}) \geq P_{\text{plan}}$where the number of sites in a country are restricted by country level minimums (e.g. Japan, China) and maximums.

Note: The computational formulae for $Pr(\dots)$ are derived using a Poisson-gamma enrollment model: **Anisimov & Fedorov (2007)**, **Anisimov (2011)**.



OPTIMISATION: INPUT AND OUTPUT

Input

targetNumberOfSubjects	targetEnrollmentInMonths	countries	minNumberOfSitesInEachCountry	maxNumberOfSitesInEachCountry	accrualRateInMonthsInEachCountry	coefficientOfVarianceInEachCountry	costPerSiteInCountry
625	16	Germany	0	30	1.5	3	13600.76
		Belgium	0	30	1.7	3	16821.52
		Italy	0	30	1.6	3	13683.21
		France	0	30	1.2	3	11936.29
		USA	0	20	0.8	3	9409.84

Output (inc Cost)

	Countries	OrigSites	MinSites	MaxSites	P100	P50	P55	P60	P65	P70	P75	P80	P85	P90	P95	P98	Rank
1	Germany	20	0	30	30	29	30	30	30	30	30	30	30	30	30	30	1
3	Italy	20	0	30	29	0	0	2	6	7	12	14	18	23	27	26	2
2	Belgium	20	0	30	0	0	2	0	0	0	0	1	0	0	0	0	3
4	France	20	0	30	29	1	0	3	1	2	0	0	1	0	4	13	4
5	USA	20	0	20	15	1	0	0	0	1	0	0	0	1	1	1	5

Cost at optimum: P100: \$1292136, P50: \$415768, P55: \$441666, P60: \$471198, P65: \$502058, P70: \$537088, P75: \$572221, P80: \$616409, P85: \$666257, P90: \$732146, P95: \$834624, P98: \$928368

Output (w/o Cost)

	Countries	OrigSites	MinSites	MaxSites	P100	P50	P55	P60	P65	P70	P75	P80	P85	P90	P95	P98	Rank
2	Belgium	20	0	30	30	25	26	28	30	27	30	30	30	30	30	30	1
1	Germany	20	0	30	30	0	0	0	2	0	7	11	7	5	16	12	2
3	Italy	20	0	30	30	4	5	5	3	11	3	2	10	17	13	24	3
4	France	20	0	30	2	0	0	0	0	0	0	0	0	0	0	0	4
5	USA	20	0	20	0	0	0	0	0	0	0	0	0	0	0	0	5

Note: 5-country scenario where each country could contribute up to 30 sites, an exhaustive search requires 31^5 possible combinations, here equal to **28,629,151** (noting that we also calculate the time for a number of probabilities).

OPTIMISATION FOUR APPROACHES

APPROACH 1: BRUTE FORCE in R

- Raw source code from SME.
- For loops round each combination of country and number of sites (from min to max).
- Also loop round Pr (Probability matrix).
- Advantage, always get the right solution.
- Disadvantage, very slow.

Time to run with 5 countries: **22mins**

No microbenchmarking for obvious reasons.

```
FoptProbPG6 <- function(va,vb,vs2,vL,vU,nn,Pr){
  opt.res <- c(sum(va*vU),vU)
  for(i1 in vL[1]:vU[1]){
    for(i2 in vL[2]:vU[2]){
      for(i3 in vL[3]:vU[3]){
        for(i4 in vL[4]:vU[4]){
          for(i5 in vL[5]:vU[5]){
            for(i6 in vL[6]:vU[6]){
              it1 <- c(i1,i2,i3,i4,i5,i6)
              if( PrTime(nn,vb,vs2,it1) < Pr )
                opt.res <- opt.res
            else{
              if(sum(va*it1) >= opt.res[1])
                opt.res <- opt.res
            else
              opt.res <- c(sum(va*it1),it1)
            }}}}}}
  return(opt.res)
}

PrTime <- function(nn,vb,vs2,x){
  i1 <- sum(x*vb)
  i2 <- sum(x*vs2)
  return(ifelse(i2<=0,
    1-ppois(nn-1,i1),
    1-pnbinom(nn-1,
      size=i1^2/i2,
      prob=i1/(i1+i2))))
}
```

OPTIMISATION FOUR APPROACHES

APPROACH 2: BRUTE FORCE in Rcpp

- One level of abstraction down, but still for loops
- Always get the right solution and quicker than brute force in R.
- Disadvantage, more coding (less understanding).
- A number of R Package updates required, including:
 - Code now in ./src/
 - useDynLib(EnrollmentModeling) in NAMESPACE
 - #include <Rcpp.h>
 - using namespace Rcpp;
 - // [[Rcpp::export]]
 - Learn to Install and Restart rather than CMD+SHIFT+L

Time to run with 5 countries: ~90 secs

Microbenchmarking: microbenchmark(optimizeEnrollment(...))

```
NumericVector FoptProbPG6(NumericVector va,
                          NumericVector vb,
                          NumericVector vs2,
                          NumericVector vL,
                          NumericVector vU,
                          double nn,
                          double Pr) {
:
:
if (_sumIteratorAndVs2 <= 0) {
    _rpois=R::ppois(nn - 1, _sumIteratorAndVb,TRUE,FALSE);
    _1minusrpois= 1 - _rpois;
    returnVar=_1minusrpois;
} else {
    _multiplySumIteratorAndVb=_sumIteratorAndVb * _sumIteratorAndVb;
    _rpnbinom=R::pnbinom(nn - 1, _multiplySumIteratorAndVb, _sumIter
    _1minusrpnbinom= 1 - _rpnbinom;
    returnVar=_1minusrpnbinom;
}
```

Unit: seconds

	expr	min	lq	mean	median	uq	max	neval
optimizeEnrollment(modelObj, additionalConfigObj, additionalCostObj)		86.06418	86.08524	86.13695	86.10583	86.11723	86.3662	100



OPTIMISATION FOUR APPROACHES

APPROACH 3: DEOPTIM

- Differential Evolution Optimisation.
- Performs evolutionary global optimisation via the Differential Evolution algorithm.
- Key arguments:
 - lower
 - upper
 - Other args into function
 - initialpop
 - Itermax
 - fnMap = round
- A lot lot quicker.

```
Deoptim::DEoptim(deOptimFunc,  
# specify lower and upper bounds on each parameter to be optimized  
    lower=vL,  
    upper=vU,  
# Other arguments needed for function  
    nn=nn,  
    vb=vb,  
    va=va,  
    vs2=vs2,  
    vec_cap=vecCap,  
    prob_criteria = i,  
    vA1 = vA1,  
    vprob = vprob,  
# a list of control parameters  
    control = list(initialpop=tmp_population,  
                    itermax = 400,  
                    trace=FALSE)))
```

OPTIMISATION FOUR APPROACHES

APPROACH 3: DEOPTIM

- Not an integer solution, need to set fnMap=round.
- Took a little deconstructing of the original function (Now: `deOptimFunc`)
- For our testing we matched the brute force with up to 10 countries.

```
deOptimFunc <- function(x,va,vb,vs2,nn,Pr) {  
  if(PrTime(nn,vb,vs2,x) < Pr) {  
    return(Inf)  
  }  
  return(sum(va*x))  
}
```

Time to run with 5 countries: ~7 secs

Microbenchmarking: `microbenchmark(optimizeEnrollment(...))`

Unit: seconds

	expr	min	lq	mean	median	uq	max	neval
	<code>optimizeEnrollment(modelObj, additionalConfigObj, additionalCostObj)</code>	6.70907	6.873101	7.101971	7.001533	7.258835	8.329486	100

OPTIMISATION FOUR APPROACHES

APPROACH 4: GENETIC ALGORITHM

- This has not been implemented as of **September 2019**
- [GA::GAISL](#)
- Full integer solution
- Maximization of a fitness function using islands genetic algorithms (ISLGAs).
- Defer to the head of CfDA Data Sciences and the presentation given at:
 - **Applied Stochastic Modeling Conference 2019**
 - <http://www.asmda.es/asmda2019.html>
- Also the paper:
 - An analysis of Gray versus binary encoding in genetic search [Uday K. Chakraborty *, Cezary Z. Janikow, 2003].

KEY DESIGN ELEMENTS: S3 ORGANISATION

S3 Organisation

- **S3ClassMetadataFunctions.R:** Functions that wrap any object passed with S3 metadata, for example:

```
## EnrollFit Class
enrollfit <- function(x) {
  structure(x, class = c('enrollfit', 'enrollment', 'list'))
}
```

- **S3MethodDispatchFunctions.R:** All `xxxx.default` and `UseMethod()` for all self defined dispatch functions, for example:

```
optimizeEnrollment <- function(x, ...) {
  UseMethod("optimizeEnrollment")
}
optimizeEnrollment.default <- function(modelObj,
                                         additionalConfigObj,
                                         ...) {
  stop("modelObj is not recognised, function will not work without this object - see ?EnrollOptimise()")
}
```

Question: Organising storage with S3: either save all by the Class or all by the function name?

EXTENSIVE IN LINE HELP USING ROXYGEN2

R: Fit and display Poisson-Gamma enrollment models ▾

Find in Topic

Fit and display Poisson-Gamma enrollment models



Documentation for package 'EnrollmentModeling' version 1.1.0.20190815.0609

- [DESCRIPTION file.](#)
- [User guides, package vignettes and other documentation.](#)

Help Pages

[a](#)

[Cost](#)

[Cost.data.frame](#)

[Cost.default](#)

[EnrollFit](#)

[EnrollFit.data.frame](#)

[EnrollFit.default](#)

[EnrollFitAdditionalModelConfig](#)

[Enrollment](#)

[Enrollment.data.frame](#)

[Enrollment.default](#)

[EnrollPlan](#)

[EnrollPlan.data.frame](#)

[EnrollPlan.default](#)

[EnrollPlanAdditionalModelConfig](#)

[EnrollReproject](#)

[EnrollReproject.data.frame](#)

[EnrollReproject.default](#)

this file is to accumulate all of the required import statements in
place

Cost S3 Class

Cost S3 Class

Cost S3 Class

Enrollfit S3 Class

Enrollfit S3 Class

Enrollfit S3 Class

EnrollFitAdditionalModelConfig S3 Class

Enrollment S3 Class

Enrollment S3 Class

Enrollment S3 Class

EnrollPlan S3 Class

EnrollPlan S3 Class

EnrollPlan S3 Class

EnrollPlanAdditionalModelConfig S3 Class

Enrollreproject S3 Class

Enrollreproject S3 Class

Enrollreproject S3 Class

R: Fits an Enrollment Model ▾

Find in Topic

fitEnrollment.enrollfit (EnrollmentModeling)

R Documentation

Fits an Enrollment Model

Description

A function that fits an enrollment model based on the two objects passed.

This function performs a prediction of the main characteristics of patient enrollment at study start-up for different types of restrictions on enrollment in countries. There are several major blocks oriented to different restrictions on the enrollment and different techniques (which are selected using [EnrollFitAdditionalModelConfig](#), see below).

Usage

```
## S3 method for class 'enrollfit'  
fitEnrollment(anEnrollFitObj,  
              anEnrollFitAdditionalConfigObj, ...)
```

Arguments

`anEnrollFitObj` an enrollfit object (See [EnrollFit](#)).
`anEnrollFitAdditionalConfigObj` an additional EnrollFit configuration object for the model (See [EnrollFitAdditionalModelConfig](#)).

Returns

Returns an enrollment fit model object as a three level List:

1. `anEnrollFitObj` (Originally Passed, See `\link{EnrollFit}`)
2. `anEnrollFitAdditionalConfigObj` (Originally Passed, See `\link{EnrollFitAdditionalModelConfig}`)
3. `enrollFitModel`, returned from the enrolment fit modeling:\cr
 - `upperLimitOfPredictionRange` \cr
 - `siteStartUpIntervalForPredictEvent` \cr
 - `alphaUsedForPredictEvent` \cr
 - `minimumNumberOfSubjects` \cr
 - `predictedEnrollmentDataFrame` \cr
 - `delta` \cr
 - `probabilityOfSuccessfullyMeetingTargetEnrollment` \cr
 - `predictedStopTime` \cr
 - `predictedQuantilesDataFrame` \cr
 - `matrixOfEnrollFitQuantiles` \cr

See Also

Other EnrollFit Functions: [EnrollFitAdditionalModelConfig](#), [EnrollFit](#), [fitEnrollment](#), [print.enrollfit](#)

Other fitEnrollment functions: [fitEnrollment.default](#), [fitEnrollment](#)