Quarterly Meeting – 18 May 2021

24 Attendees

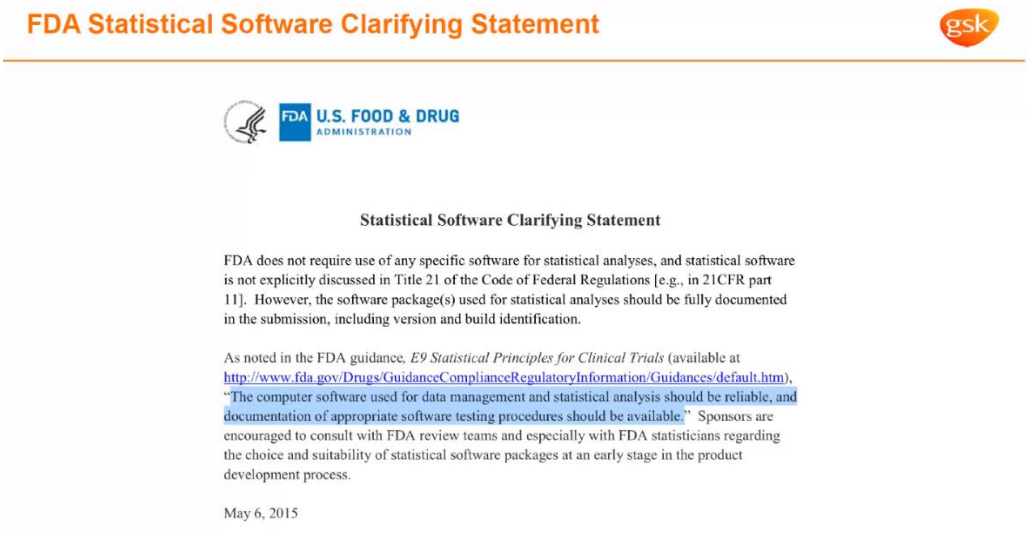
Juliane Manitz
Mark Penniston
Nicholas Masel
Matthew Montero
Satish Murthy
Jan Stiers Pieter
Soren Klim
Per Arne Stahl
Lyn Taylor
Andy Nicholls
Paulo Bargo
Bella Fang
Doug Kelkhoff
Eli Miller
Emma Martin
Stephen Glavin
Steven Haesendonckx
Jennifer Bradford
Joseph Rickert
Susanna Marquez Gargallo
Tilo Blenk
Matthias Trampisch
Jenny Wissmar
Yilong Zhang

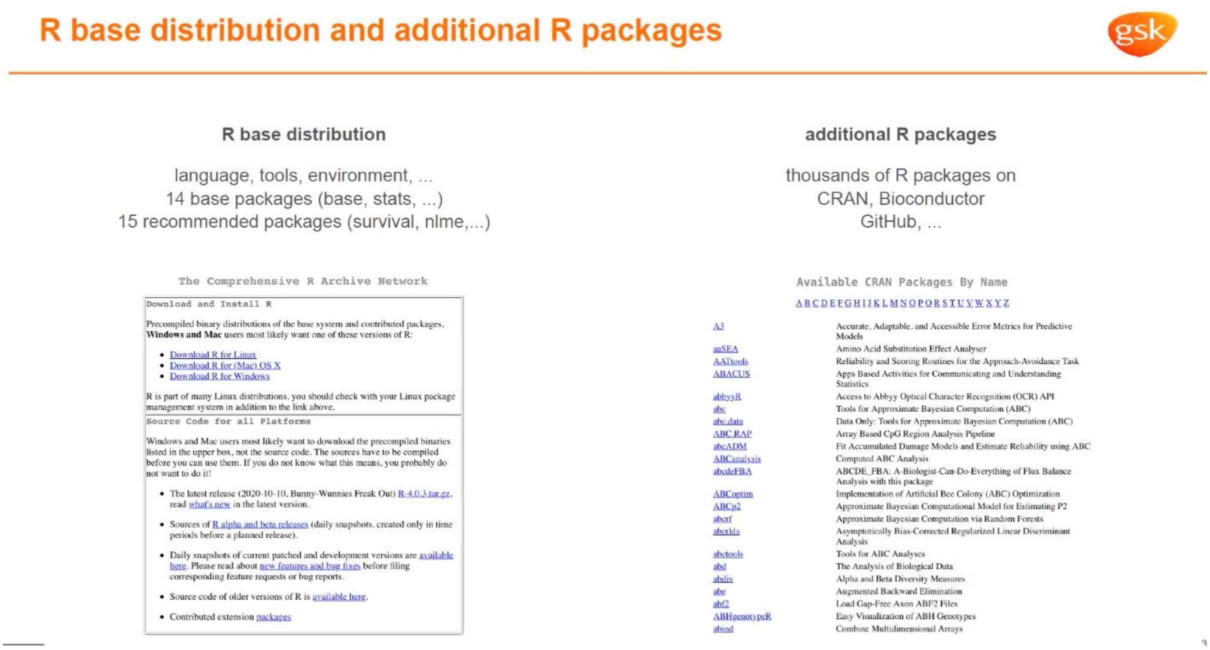
Agenda

- **Testing steam update (GSK progress): Tilo Blenk.**
- **Infrastructure team kick off: Doug Kelhoff**

Discussion

Testing steam update (GSK progress): Tilo Blenk provided the following slides summarizing the work GSK have done to date on R package testing, which will act as a basis to kick start activities of a testing stream.





GSK have built a system designed to satisfy the questions that the FDA would ask with respect to adequate testing.

Analysing data with R: functions/packages used

```r
library(readr)
library(dplyr)

hers <- read_csv("data/hersdata.csv")

hers %>%
  group_by(diabetes, exercise) %>%
  summarise(n = n(), mean_glc = mean(glucose))
#   diabetes exercise     n mean_glc
# 1 no       no        1191     97.4
# 2 no       yes        841     95.7
# 3 yes      no         504    155.
# 4 yes      yes        227    155.

fit <- lm(glucose ~ exercise, data = hers, subset = (diabetes == "no"))

summary(fit)
# Coefficients:
#              Estimate Std. Error t value Pr(>|t|)
# (Intercept)  97.3610     0.2815 345.848  < 2e-16
# exerciseyes  -1.6928     0.4376  -3.868 0.000113
#
# Residual standard error: 9.715 on 2030 degrees of freedom
# Multiple R-squared:  0.007318,  Adjusted R-squared:  0.006829
# F-statistic: 14.97 on 1 and 2030 DF,  p-value: 0.000113
```

functions used in the R script

function	package
`library()`	base
`<-`	base
`read_csv()`	readr
`%>%`	magrittr
`group_by()`	dplyr
`summarise()`	dplyr
`n()`	dplyr
`mean()`	base
`print()`	tibble
`lm()`	stats
`summary()`	stats
`print()`	stats

`print()` is called implicitly with `tibble` and `summary.lm` objects

Components of R testing/verifying

- **package assessment**
  Assessing packages to decide if they can be considered as sufficiently tested/verified as they are.
- **resource assessment**
  Assessing resources like the R Foundation and RStudio to decide if the products they provide, ie R base distribution from R Foundation or tidyverse, r-lib, etc from RStudio, can be considered as sufficiently tested/verified.
- **testing**
  Testing packages: (1) qualification tests for packages considered in package assessment as sufficiently tested and (2) verification tests (reliability/correctness) for packages considered as insufficiently tested.
- **frozen R installations**
  R installations with R base distribution and selected R packages which users cannot change, ie no package installation or update is possible for users.
- **controlled execution**
  When executing a R script for a GxP process (1) frozen installation is used, (2) checking that only tested/verified functions/packages are used, (3) executing as background process, and (4) capturing/saving R script, context information, and standard out/error.

Automated testing with testthat

expression to be checked

expected result

```
expect_equal(1 + 2, 3)
```

```
> library(testthat)
>
> expect_equal(1 + 2, 3)
>
> expect_equal(1 + 1, 3)
Error: 1 + 1 not equal to 3.
1/1 mismatches
[1] 2 - 3 == -1
```

Automated testing with testthat

```
> x <- sample(1:10)                    # create numeric vector x with numbers 1 to 10 in random order
> x
[1]  4  7  8  3  1  2 10  5  9  6

> min(x)                               # get minimum of vector
[1] 1

> mean(x)                              # calculate arithmetic mean of vector
[1] 5.5

> expect_equal(min(x), 1)              # test min(x) against expected value of 1

> expect_equal(mean(x), 5.5)           # test mean(x) against expected value of 5.5

> expect_equal(mean(x), 111)           # failing test of mean(x)
Error: mean(x) not equal to 111.
1/1 mismatches
[1] 5.5 - 111 == -106
```

Testing basic analytic functions (obviously correct results)

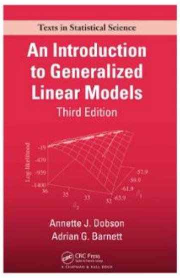
```
test_that("basic analytic functions", {
  x <- sample(1:10)
  expect_equal(min(x), 1)
  expect_equal(max(x), 10)
  expect_equal(range(x), c(1, 10))
  expect_equal(sort(x), 1:10)
  expect_equal(sum(x), 55)
  expect_equal(cumsum(sort(x)), c(1, 3, 6, 10, 15, 21, 28, 36, 45, 55))
  expect_equal(mean(x), 5.5)
  expect_equal(median(x), 5.5)
})
```

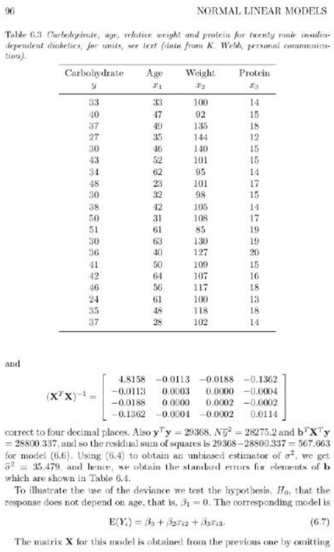
The above works good for simple tests, but not for statistical modelling testing.

Testing statistical models (external reference)

```r
test_that("linear regression models", {
  # data/results from: Dobson AJ, Barnett AG. An Introduction to Generalized Linear Models. 3rd ed. CRC Press. 2008.

  # table 6.3 page 96
  d <- data.frame(
    carbohydrate = c(33, 40, 37, 27, 30, 43, 34, 48, 30, 38, 50, 51, 30, 36, 41, 42, 46, 24, 35, 37),
    age = c(33, 47, 49, 35, 46, 52, 62, 23, 32, 42, 31, 61, 63, 40, 50, 64, 56, 61, 48, 28),
    weight = c(100, 92, 135, 144, 140, 101, 95, 101, 98, 105, 108, 85, 130, 127, 109, 107, 117, 100, 118, 102),
    protein = c(14, 15, 18, 12, 15, 15, 14, 17, 15, 14, 17, 19, 19, 20, 15, 16, 18, 13, 18, 14)
  )

  fit <- lm(carbohydrate ~ age + weight + protein, data = d)

  # table 6.4 page 97
  expect_equivalent(
    round(coefficients(fit), 3),
    c(36.960, -0.114, -0.228, 1.958)
  )
  expect_equivalent(
    round(summary(fit)$coefficients[,"Std. Error"], 3),
    c(13.071, 0.109, 0.083, 0.635)
  )
})
```


Textbooks as external references





96 NORMAL LINEAR MODELS

Table 6.3 *Carbohydrate, age, relative weight and protein for twenty male insulin-dependent diabetics; for units, see text (data from K. Webb, personal communication).*

Carbohydrate y	Age x_1	Weight x_2	Protein x_3
33	33	100	14
40	47	92	15
37	49	135	18
27	35	144	12
30	46	140	15
43	52	101	15
34	62	95	14
48	23	101	17
30	32	98	15
38	42	105	14
50	31	108	17
51	61	85	19
30	63	130	19
36	40	127	20
41	50	109	15
42	64	107	16
46	56	117	18
24	61	100	13
35	48	118	18
37	28	102	14

and

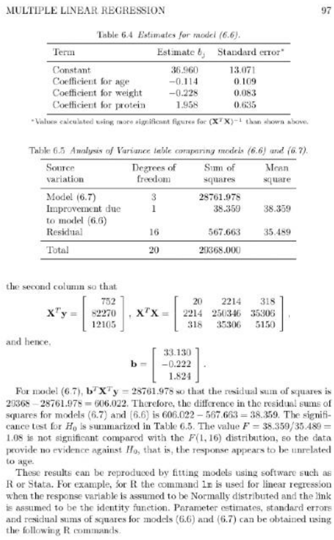
$$(\mathbf{X}^T\mathbf{X})^{-1} = \begin{bmatrix} 4.8158 & -0.0113 & -0.0188 & -0.1362 \\ -0.0113 & 0.0003 & 0.0000 & -0.0004 \\ -0.0188 & 0.0000 & 0.0002 & -0.0002 \\ -0.1362 & -0.0004 & -0.0002 & 0.0114 \end{bmatrix}$$

correct to four decimal places. Also $\mathbf{y}^T\mathbf{y} = 29368$, $N\overline{y}^2 = 28275.2$ and $\mathbf{b}^T\mathbf{X}^T\mathbf{y} = 28800.337$, and so the residual sum of squares is $29368 - 28800.337 = 567.663$ for model (6.6). Using (6.4) to obtain an unbiased estimator of σ^2, we get $\widehat{\sigma}^2 = 35.479$, and hence, we obtain the standard errors for elements of $\mathbf{b}$ which are shown in Table 6.4.

To illustrate the use of the deviance we test the hypothesis, H_0, that the response does not depend on age, that is, $\beta_1 = 0$. The corresponding model is

$$\mathrm{E}(Y_i) = \beta_0 + \beta_2 x_{i2} + \beta_3 x_{i3}. \qquad (6.7)$$

The matrix $\mathbf{X}$ for this model is obtained from the previous one by omitting



MULTIPLE LINEAR REGRESSION 97

Table 6.4 *Estimates for model (6.6).*

Term	Estimate b_j	Standard error*
Constant	36.960	13.071
Coefficient for age	−0.114	0.109
Coefficient for weight	−0.228	0.083
Coefficient for protein	1.958	0.635

*Values calculated using more significant figures for $(\mathbf{X}^T\mathbf{X})^{-1}$ than shown above.

Table 6.5 *Analysis of Variance table comparing models (6.6) and (6.7).*

Source variation	Degrees of freedom	Sum of squares	Mean square
Model (6.7)	3	28761.978	
Improvement due to model (6.6)	1	38.359	38.359
Residual	16	567.663	35.489
Total	20	29368.000	

the second column so that

$$\mathbf{X}^T\mathbf{y} = \begin{bmatrix} 752 \\ 82270 \\ 12105 \end{bmatrix}, \ \mathbf{X}^T\mathbf{X} = \begin{bmatrix} 20 & 2214 & 318 \\ 2214 & 250346 & 35306 \\ 318 & 35306 & 5150 \end{bmatrix},$$

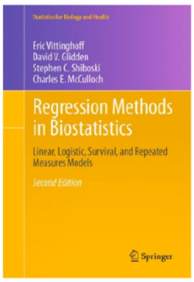
and hence,

$$\mathbf{b} = \begin{bmatrix} 33.130 \\ -0.222 \\ 1.824 \end{bmatrix}.$$

For model (6.7), $\mathbf{b}^T\mathbf{X}^T\mathbf{y} = 28761.978$ so that the residual sum of squares is $29368 - 28761.978 = 606.022$. Therefore, the difference in the residual sums of squares for models (6.7) and (6.6) is $606.022 - 567.663 = 38.359$. The significance test for H_0 is summarized in Table 6.5. The value $F = 38.359/35.489 = 1.08$ is not significant compared with the $F(1, 16)$ distribution, so the data provide no evidence against H_0, that is, the response appears to be unrelated to age.

These results can be reproduced by fitting models using software such as R or Stata. For example, for R the command `lm` is used for linear regression when the response variable is assumed to be Normally distributed and the link is assumed to be the identity function. Parameter estimates, standard errors and residual sums of squares for models (6.6) and (6.7) can be obtained using the following R commands.

Textbooks as external references



with publicly available real world clinical trial data

results in the book were calculated with Stata

10 — 2 Exploratory and Descriptive Methods

Table 2.1 Numerical description of systolic blood pressure

```
. summarize sbp, detail

                       systolic BP
-------------------------------------------------------------
      Percentiles      Smallest
 1%         104            98
 5%         110           100
10%         112           100       Obs                3154
25%         120           100       Sum of Wgt.        3154

50%         126                     Mean           128.6328
                        Largest     Std. Dev.      15.11773
75%         136           210
90%         148           210       Variance       228.5458
95%         156           212       Skewness       1.204397
99%         176           230       Kurtosis       5.792465
```

2.3.1.3 Graphical Description

Graphs are often the quickest and most effective way to get a sense of the data. For numerical data, three basic graphs are most useful: the histogram, boxplot, and normal quantile–quantile (or Q–Q) plot. Each is useful for different purposes.

4.1 Example: Exercise and Glucose — 71

Table 4.1 Unadjusted regression of glucose on exercise

```
. regress glucose exercise if diabetes == 0

      Source |       SS       df       MS              Number of obs =    2032
-------------+------------------------------           F(  1,  2030) =   14.97
       Model |  1412.50418     1  1412.50418           Prob > F      =  0.0001
    Residual |  191605.195  2030  94.3867954           R-squared     =  0.0073
-------------+------------------------------           Adj R-squared =  0.0068
       Total |  193017.699  2031  95.0357946           Root MSE      =  9.7153

------------------------------------------------------------------------------
     glucose |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    exercise |  -1.692789   .4375862    -3.87   0.000    -2.550954   -.8346243
       _cons |   97.36104   .2815138   345.85   0.000     96.80896    97.91313
------------------------------------------------------------------------------
```

clinical trial of hormone therapy (HT) (Hulley et al. 1998). Women with diabetes are excluded because the research question is whether exercise might help to prevent progression to diabetes among women at risk, and because the causal determinants of glucose may be different in that group. Furthermore, glucose levels are far more variable among diabetics, a violation of the assumption of homoscedasticity, as we

hers

	ht	exercise	physact	diabetes	age	bmi	tchol	hdl	ldl	glucose	raceth
1	placebo	no	much more active	no	70	23.69	189	52	122.4	84	African
2	placebo	no	much less active	no	62	28.62	307	44	241.6	111	African
3	hormone therapy	no	about as active	yes	69	42.51	254	57	166.2	114	White
4	placebo	no	much less active	no	64	24.39	204	56	116.2	94	White

Showing 1 to 4 of 2,763 entries, 37 total columns

Real world data, example above has 2763 observations, so realistic testing.

Correct results have to be known

```
> set.seed(123)

> x <- rnorm(100)                 # create numeric vector x with 100 random numbers
> x
 [1] -0.560475647 -0.230177489
 [3]  1.558708314  0.070508391
 [5]  0.129287735  1.715064987
...
[97]  2.187332993  1.532610626
[99] -0.235700359 -1.026420900

> mean(x)                         # calculate arithmetic mean of vector
[1] 0.09040591                    # DO WE REALLY KNOW THAT 0.09040591 IS CORRECT ?

> expect_equal(mean(x), 0.09040591)
```

To get the known results, you need to use external references or obvious results that are known.

How detailed need the tests to be?

```
> x <- sample(1:10)                        # create numeric vector x with numbers 1 to 10 in random order
> expect_equal(mean(x), 5.5)               # test mean(x) against expected value of 5.5

> x <- sample(1:1e6)                       # bigger input, vector with numbers 1 to 1 million
> expect_equal(mean(x), 500000.5)

> x <- c(1e6, 1e6, 0.01, 0.01)             # very big and small elements
> expect_equal(mean(x), 500000.005)

> x <- c(1, 2, 3, NA)                      # handling of NA values
> expect_true(is.na(mean(x)))
> expect_equal(mean(x, na.rm = TRUE), 2)

> x <- numeric(0)                          # numeric vector without elements
> expect_true(is.nan(mean(x)))

> x <- c(-10, 2:9, 500)
> expect_equal(mean(x, trim = 0.1), 5.5)   # more function arguments
```

Running tests interactively in RStudio IDE

```
> test_dir("rtests")
✓ |  OK F W S | Context
✓ | 345       | base [0.6 s]
✓ |  53       | biostats-reg [0.5 s]
✓ | 140       | dplyr [0.3 s]
✓ |  30       | forcats
✓ |  10       | haven
✓ |  15       | jsonlite
✓ |  16       | lubridate
✓ |  27       | purrr
✓ |  20       | readr
✓ | 123       | rsqlite [0.2 s]
✓ |  44       | stats
✓ |  26       | stringr
✓ |  13       | tibble
✓ |  19       | tidyr
✓ |  20       | xml2
✓ |   6       | yaml

══ Results ═══════════════════════════════
Duration: 2.2 s

[ FAIL 0 | WARN 0 | SKIP 0 | PASS 907 ]
```

Documenting tests with R Markdown and testthat

rtesting-report.Rmd

````
---
title: "R Testing Report"
output: html_document
---

Whatever additional text.

```{r, echo=FALSE, comment=""}
library(testthat)
source("rtesting-reporter.R")

r <- RTestingReporter$new()
test_dir("rtests/", reporter = r)

ntotal <- r$n_pass_reporter +
  r$n_fail_reporter + r$n_warn_reporter
ptotal <- r$n_pass_reporter / ntotal * 100
```

**Conclusion:** `r if (ptotal > 99) "Testing
PASSED " else "Testing FAILED "` with `r
round(ptotal, 2)`% of all tests passed
successfully.
````

23:1 (Top Level) R Markdown

R Markdown rendered to report

test are executed and results displayed

R Testing Report

Whatever additional text.

```
running tests start: 2021-03-26 08:59:21

   pass   fail   warn

     43      0      0    base : mathematical functions
     17      0      0    base : string functions
     19      0      0    base : date functions
     30      0      0    base : NA handling
     11      0      0    base : basic analytic functions
     33      0      0    base : vectors and data types
     39      0      0    base : some vector functions
     10      0      0    base : factors
     14      0      0    base : matrices
     16      0      0    base : basic matrix calculations
     18      0      0    base : data frames
     39      0      0    base : data processing with data frames
     12      0      0    base : reading and writing data
      4      0      0    base : serializing r objects
     13      0      0    base : set functions
     16      0      0    base : functionals
    334      0      0    base : total

      6      0      0    stats : basic analytic functions
     16      0      0    stats : sampling functions
     14      0      0    stats : linear models, matrix algebra
      3      0      0    stats : glm: logistic regression
      5      0      0    stats : glm: Poisson regression
     44      0      0    stats : total

    378      0      0    total
  100.0    0.0    0.0    total %

running tests stop: 2021-03-26 08:59:22
```

Conclusion: Testing PASSED with 100% of all tests passed successfully.

testing examples

Test examples: data frame data structure

```r
test_that("data frames", {
  i <- 1:10
  d <- data.frame(i = i, f = i + 0.12345, s = letters[i], stringsAsFactors = FALSE)
  expect_true(is.data.frame(d))
  expect_equal(dim(d), c(10, 3))
  expect_equal(nrow(d), 10)
  expect_equal(ncol(d), 3)
  expect_equal(colnames(d), c("i", "f", "s"))
  expect_equal(d$i, i)
  expect_equal(d[,"i"], i)
  expect_equal(d[,1], i)
  expect_equal(round(d[,2], 4), round(i + 0.12345, 4))
  expect_equal(d$s, letters[i])
  expect_equal(d[1,], data.frame(i = 1, f = 1.12345, s = "a", stringsAsFactors = FALSE))
  expect_equal(d[1,1], 1)
  expect_equal(d[3,3], "c")
})
```

Questions for the testing stream

Per Arne Stahl (AZ): AZ are at the same position as GSK and having same discussion about automated testing. One question coming back from QA is how to test the test scripts. Did you have this question at GSK? Tilo's response: No, they didn't get that, but for testthat, you can test it such that if the TRUE comes out when you know it to be true, then you can show testthat seems to work.

Andy Nicholls added, that we also have to have faith/trust in the BASE language, such that TRUE is TRUE, FALSE is FALSE and so on.

Per Arne Stahl: stressed that given R has been used for 20 years or so, we all do believe it works as it's used by academics and they have written new statistical methods using it, however we just need to provide the documentation of this for industry regulators.

Per Arne Stahl – have you asked the regulators if they are happy with the approach you are using. Tilo's response: No. Andy Nicholls: the idea would be that we take something like this approach and the tests to the R validation Hub testing stream, and release it to the wider R Validation Hub community. This way we can come together and release this to the regulators to request that they do accept this approach and this method of testing for R use in industry.

Doug Kelkhoff: How do you see the collaboration happening in this space. How would companies collaborate to incorporate the tests into the packages. Could we ask the authors to include the tests in their packages? Tilo's response: as they in academia and not in industry they may not be happy to do this… and also you will need some qualification tests outside of the package tests. Hence, the aim is to write the tests, make them available through the R Validation Hub, to allow free use of the installed systems and packages along with the testing scripts. The intention is to share with the community, and discussions will continue through the testing stream of the R Validation Hub. The intention is to write a white paper on this topic.

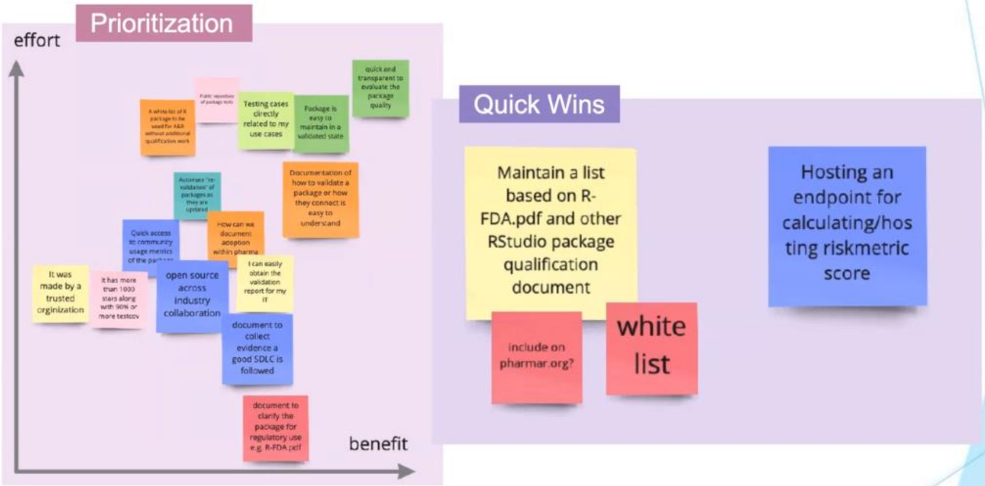
Infrastructure: Doug Kelhoff presented the following slides on the newly setup infrastructure team

Motivations

- **riskmetric:**
  Foundational tools for package assessment
- **Shiny app:**
  Interface for institutions to operationalize riskmetric as part of business process
- **Remainging Outreach Gap:**
  How do we make these tools available for communication, industry consistency, regulator reference?

Seed Ideas

Infrastructure Team Brainstorming Session



Special thanks to attendees: Edgar Manukyan (Roche), Eli Miller (Atorus), Eric Milliman (Biogen), Heidi Curinckx (J&J), Marly Cormar (Biogen), Mike Stackhouse (Atorus), Nan Xiao (Merck), Steven Haesendonckx (J&J), Yilong Zhang (Merck)

Infrastructure Team First Steps

Build the Team

- Contributors welcome!
- Lead(s) / organizer role

Infrastructure Team First Steps

Risk score API

- Build an API to run and return risk scores
- Build an endpoint for risk score badges
- Communicate new development needs to riskmetric/shiny app teams

? Estimate infrastructure need to host the api and/or cache risk scores in a database

? Help to write a R Consortium proposal for infrastructure budget

Infrastructure Team Formation

Interested? Join our Slack!

rvalidationhub.slack.com

infrastructure