Metric Design & Dashboarding with tidymetrics & shinybones

Kaelen L. Medeiros
DC R Conference 2019
@kaelen_medeiros

Kaelen Medeiros

@kaelen_medeiros
github.com/klmedeiros
Data Scientist

Overview

- North Star Metric
- Metric design principles
- Creating a metric in R with tidymetrics
  - cross_by functions
  - documentation
- Turning your metric into a dashboard with shinybones
  - _site.yml
  - app.R

North Star Metric (NSM)

- ONE metric that measures how your product delivers value to its users
- Examples:
  - Airbnb: Nights Booked
  - Facebook: # users adding 7 friends in the first 10 days
  - Duolingo: daily active users (DAU)

Gaming the NSM

- If the NSM is “# of questions asked”, this can easily be gamed by driving users to the site
    - You’ll see a spike in number of questions asked, but does it remain over time? Do you have to continue spending money to drive that traffic in?
- How do we account for long-term retention, while also keeping ourselves honest about spikes in the NSM?

“When a metric becomes a target, it ceases to become a good measure.”

- Marilyn Strathern, summarizing Goodhart’s law

“Safety” metrics

- Balance out the NSM by guarding against:
  - artificial inflations,
  - gaming the system,
  - the decisions you make that influence the metric,
  - etc.

Example: Stack Overflow questions

- Dataset of R Stack Overflow (S.O.) questions available on Kaggle
- How do users derive value from Stack Overflow?
  - Asking questions!
  - Getting answers!
  - Gaining knowledge!

$$\#\ questions\ asked\ =\ Average\ \#\ questions\ asked\ *\ \#\ active\ users$$

$$Average\ \#\ questions\ asked\ =\ \frac{\#\ questions\ asked}{\#\ active\ users}$$

$$\#\ active\ users\ =\ (\#\ users\ *\ \%\ users\ active)$$

$$\#\ questions\ asked\ =\ \frac{\#\ questions\ asked}{\#\ active\ users}\ *\ (\#\ users\ *\ \%\ users\ active)$$

Deconstructing the full S.O. NSM

To calculate:

- Number of S.O. R users
- Number of S.O. R questions
- Number of active S.O. R users
  - Active = asked at least 1 question in that time period
- Percent of active users
- Average number of questions asked by active users

tidymetrics

tidymetrics

- Developed by Ramnath Vaidyanathan (aut, maintainer) and Dave Robinson (aut)
- Available on github
- Create metrics in a tidy way
- cross_by_* functions
  - periods
  - dimensions

cross_by_periods()

- “Cross by any set of calendar periods (like day or week), rolling windows, or recent intervals (like "4 Weeks", or "8 Weeks").”
  - Accepts periods, date windows, intervals of time, etc. and creates your summarized output across all of those possible times
  - REQUIRES a variable to be named date, so you might have to mutate() an existing date variable to call it that (and should strip it to be a date alone, without timestamps)

cross_by_dimensions()

- “...acts as an extended group_by that allows complete summaries across each individual dimension and possible combinations.”
    - Accepts a categorical variable in some form, factor or not, and crosses every possible combination of your dimensions (and your periods, if applicable) before calculating your outcome metric

Stack Overflow R questions NSM example

- Variables:
  - owner_user_id
  - creation_date
  - Tag
- To calculate:
  - # of possible users (question-askers)
  - # of active users (asked at least 1 question in the time period)
  - # of R questions asked
  - Average number of questions asked by average users
  - Percent of active users

```
# A tibble: 76,616 x 3
   owner_user_id creation_date       tag
           <dbl> <dttm>              <chr>
 1         15842 2008-09-18 17:59:19 statistics
 2       1941213 2008-11-01 15:48:30 statistics
 3         67405 2009-02-18 09:08:38 statistics
 4         69117 2009-02-27 21:49:17 None
 5         69117 2009-02-27 22:45:59 None
 6         67405 2009-03-18 08:59:04 statistics
 7         20895 2009-04-10 02:18:38 statistics
 8         85950 2009-04-10 02:32:21 statistics
 9         14744 2009-04-13 12:52:51 None
10         85950 2009-04-15 07:55:21 statistics
# … with 76,606 more rows
```

Number of Users

```r
# nb_users
nb_users <- questions %>%
  filter(creation_date >= DATE_START) %>%
  mutate(date = as.Date(creation_date)) %>%
  cross_by_periods(PERIODS) %>%
  cross_by_dimensions(tag) %>%
  summarize(nb_users = n_distinct(owner_user_id))
```

Engagement metrics

- by_day_questions gives us every question asked per tag per day

```
> by_day_questions
# A tibble: 48,600 x 4
   date       owner_user_id tag        nb_questions_asked
   <date>             <dbl> <chr>                   <int>
 1 2015-01-01       1610626 None                        1
 2 2015-01-01       2255656 data.table                  1
 3 2015-01-01       2497790 ggplot2                     1
 4 2015-01-01       3223138 data.table                  1
 5 2015-01-01       3368667 None                        1
 6 2015-01-01       3580859 ggplot2                     1
 7 2015-01-01       3580859 shiny                       1
 8 2015-01-01       3943160 None                        1
 9 2015-01-01       4278119 None                        1
10 2015-01-02        974465 data.table                  1
# … with 48,590 more rows
```

Engagement metrics

```
question_engagement_raw <- by_day_questions %>%
  cross_by_periods(PERIODS) %>%
  cross_by_dimensions(tag) %>%
  summarize(nb_active_users = n_distinct(owner_user_id),
            nb_questions_asked = sum(nb_questions_asked, na.rm = TRUE)) %>%
  ungroup()
```

Engagement metrics

```
# A tibble: 10,385 x 5
   period date       tag        nb_active_users nb_questions_asked
   <chr>  <date>     <chr>                <int>              <int>
 1 day    2015-01-01 All                      8                  9
 2 day    2015-01-01 data.table               2                  2
 3 day    2015-01-01 ggplot2                  2                  2
 4 day    2015-01-01 None                     4                  4
 5 day    2015-01-01 shiny                    1                  1
 6 day    2015-01-02 All                     17                 17
 7 day    2015-01-02 data.table               1                  1
 8 day    2015-01-02 ggplot2                  4                  4
 9 day    2015-01-02 None                     6                  6
10 day    2015-01-02 package                  1                  1
# … with 10,375 more rows
```

“Number of users”

```r
nb_users_fake <- question_engagement_raw %>%
  group_by(period, date, tag) %>%
  mutate(nb_users = as.integer(runif(1, nb_active_users, nb_active_users * 4))) %>%
  select(period, date, tag, nb_users)
```

Active Users Any

```
active_users_any <- question_engagement_raw %>%
  filter(tag == "All") %>%
  select(period, date,
         nb_active_users_any_tag = nb_active_users)
```

Engagement metrics

```
question_engagement <- question_engagement_raw %>%
  inner_join(active_users_any) %>%
  mutate(avg_questions_asked = nb_questions_asked / nb_active_users_any_tag) %>%
  inner_join(nb_users_fake) %>%
  mutate(pct_users_active = nb_active_users / nb_users) %>%
  select(-nb_active_users_any_tag) %>%
  group_by(period) %>%
  mutate(current_period = floor_date(today(), period[1])) %>%
  ungroup() %>%
  filter(date < current_period) %>%
  select(-current_period)
```

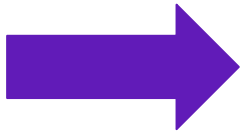
A metric!

```
# A tibble: 10,385 x 8
   period date       tag      nb_active_users nb_questions_asked avg_questions_as… nb_users pct_users_active
   <chr>  <date>     <chr>              <int>              <int>              <dbl>    <int>             <dbl>
 1 day    2015-01-01 All                    8                  9               1.12       27             0.296
 2 day    2015-01-01 data.ta…               2                  2               0.25        7             0.286
 3 day    2015-01-01 ggplot2                2                  2               0.25        6             0.333
 4 day    2015-01-01 None                   4                  4               0.5         4             1
 5 day    2015-01-01 shiny                  1                  1               0.125       2             0.5
 6 day    2015-01-02 All                   17                 17               1          48             0.354
 7 day    2015-01-02 data.ta…               1                  1               0.0588      3             0.333
 8 day    2015-01-02 ggplot2                4                  4               0.235      11             0.364
 9 day    2015-01-02 None                   6                  6               0.353      12             0.5
10 day    2015-01-02 package                1                  1               0.0588      1             1
# … with 10,375 more rows
```

The importance of documentation

- If you document your metric, your documentation will become the title/tooltip as applicable with shinybones
- Luckily, this is made easier for you with use_metrics_scaffold()

```
> use_metrics_scaffold(question_engagement)
metrics:
  nb_active_users:
    title: <TODO>
    description: <TODO>
  nb_questions_asked:
    title: <TODO>
    description: <TODO>
  avg_questions_asked:
    title: <TODO>
    description: <TODO>
  nb_users:
    title: <TODO>
    description: <TODO>
  pct_users_active:
    title: <TODO>
    description: <TODO>
dimensions:
  tag:
    title: <TODO>
    description: <TODO>
```



```
---
name: metrics_questions_northstar
owner: kaelen
metrics:
  nb_active_users:
    title: Number of Active Users
    description: Number of users who have asked at least one question.
  nb_questions_asked:
    title: Number of Questions Asked
    description: Total number of questions asked by active users.
  avg_questions_asked:
    title: Average Number of Questions Asked
    description: Average number of questions asked by active users, calculated over all possible
question tags.
  nb_users:
    title: Number of Users
    description: Total number of users.
  pct_users_active:
    title: Percent Users Active
    description: Percent of users active (asked at least one question) out of all users.
dimensions:
  tag:
    title: Tag
    description: None/dplyr/shiny/statistics/etc.
---
```

Save & preview your metric

- Create your metric
- Save it (locally if you're not going to be working with S3!)
- Preview what it will look like in your dashboard

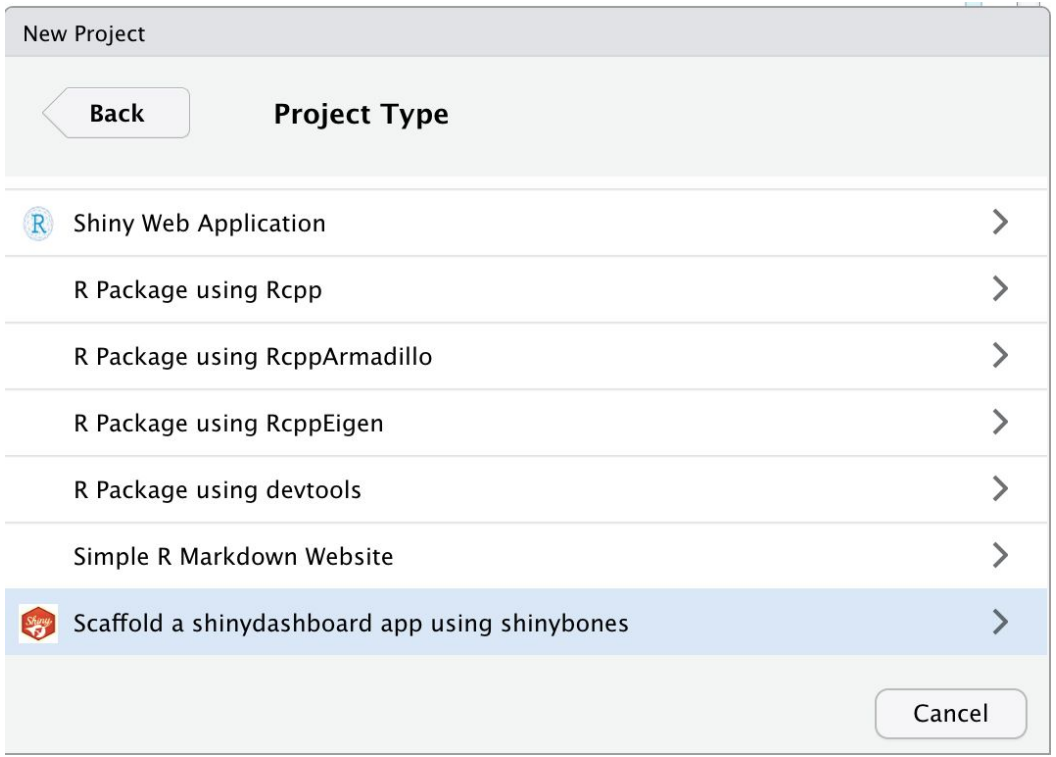
```{r create_metrics}
question_metrics <- create_metrics(question_engagement)
save_metrics(question_metrics, type="local")
```

```{r preview_metric, eval=FALSE}
shinymetrics::preview_metric(question_metrics$questions_northstar_avg_questions_asked)
```

shinybones & shinymetrics

shinybones

- Developed by Ramnath Vaidyanathan
- Available on github
- “Opinionated” framework for creating and organizing Shiny dashboards
  - Where you get visualizations for “free” (line and bar graphs)
  - Can create your dashboard with a yaml file & page modules



shinymetrics

- Developed by Ramnath Vaidyanathan
- Available on github
- Provides Shiny modules needed to display tidy metrics
    - You'll need it for preview_metrics() and for parts of shinybones to work!

_site.yml

- To create a dashboard, write a yaml file, where the following rules apply:
  - Each menu item becomes a menu item in the sidebar.
  - If a menu item has more than 1 child, the children become subitems.
  - If the child of a menu item has a tabs item, they are rendered as a tabSetPanel
  - Each item (page) is connected to a module by its name.

```yaml
name: Stack Overflow
sidebar:
  - text: North Star
    icon: star
    tabs:
      - text: Questions Asked
        module:
          page_metrics:
            metrics:
              - id: questions_northstar_nb_questions_asked
                selected_period: week
      - text: Average Questions Asked
        module:
          page_metrics:
            metrics:
              - id: questions_northstar_avg_questions_asked
                selected_period: week
                zero: TRUE
      - text: Percent of Users Active
        module:
          page_metrics:
            metrics:
              - id: questions_northstar_pct_users_active
                selected_period: week
      - text: Number of Users
        module:
          page_metrics:
            metrics:
              - id: questions_northstar_nb_active_users
                selected_period: week
              - id: questions_northstar_nb_users
                selected_period: week
```

app.R

- Every time:
  - Add your metric as an argument to DATA
- Good to do:
  - Set the dc_lazy_data option
- If you're working locally:
  - Set your DC_S3_ENVIRONMENT to local

```r
# Load Libraries -----
library(shiny)
library(shinydashboard)
library(shinybones)

# Set to local environment -----
Sys.setenv(DC_S3_ENVIRONMENT = "local")

# Load Utilities -----
source_dirs('utils')
source_dirs('components')
source_dirs('pages')

# Global Data ----
# This is passed to all page modules as an argument named data_global
DATA <- list(
  question_metrics
)

options(dc_lazy_data = TRUE)

# Configuration
options("yaml.eval.expr" = TRUE)
CONFIG <- yaml::read_yaml('_site.yml')

# UI ----
ui <- function(request){
  dashboardPage(
    # Header ----
    dashboardHeader(title = CONFIG$name),

    # Sidebar ----
    dashboardSidebar(
      sb_create_sidebar(CONFIG, DATA)
    ),

    # Body -----
    dashboardBody(
```

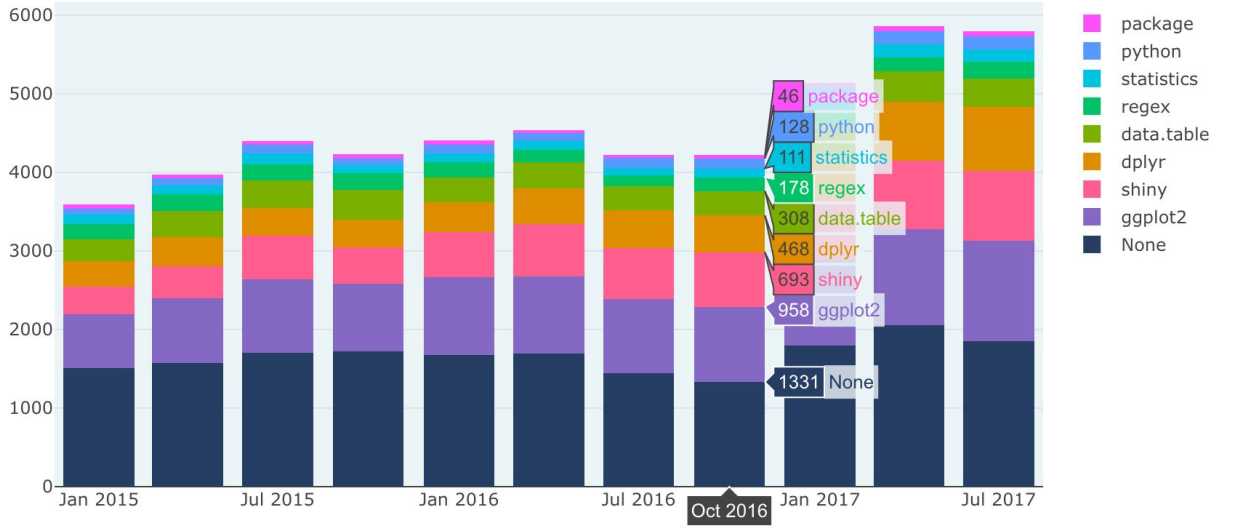
Stack Overflow

North Star

Questions Asked | Average Questions Asked | Percent of Users Active | Number of Users

Number of Questions Asked

All | Tag



Last updated: 23 hours ago

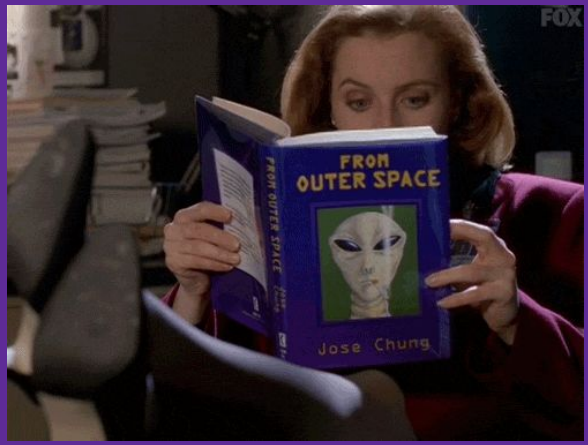
as ✔ | %

aggregated by Quarter

over All Time

CSV

Thanks to DC R & Jared Lander for having me!



Thanks to Dave Robinson & Ramnath Vaidyanathan for their work on these packages + teaching me how to use them!

Come talk to me later!!!

kaelenmedeiros@gmail.com
@kaelen_medeiros
github.com/klmedeiros

P.S. Hire me!!!!