A New Hope Case Study

David T. Noland

1/21/2019

A New Hope Case Study

Welcome to A New Hope - an introductory case study in beginning theoretical applications of using R and RStudio. For more information on R and RStudio, please review the https://rstudio.com. The following is a hypothetical, fictionalized accounting of a real-world collaboration within one technology company. The full repository for this case study (including utilized initial dataset and final output) is downloadable at **Github**: https://github.com/dnoland-wpe/anewhope

Context

In December 2018, PHP versions 5.6 and 7.0 both reached their respective EOL timeframe. As a result, these versions would no longer be supported by the PHP Community, thus no more patches to these Apache versions, leaving servers potentially vulnerable to new security threats that arise after the support cycle completed.

Platform Hosting Outer Rim Commerce Enterprises, Inc. (PHORCE) (formerly branded as the Trade Federation) is now tasked with upgrading their entire server farm that supports 10 planets and mobile stations within the Galactic Empire within a six month timeline. This necessitates collaboration from the **Operations Data Team (ODT)**, the **Imperial Consortium for Engineering (ICE)**, and the **Coruscant Communications Network (CCN)** to personally communicate with each of their customers to plan and execute the upgrade plan.

Purpose of Case Study

- Demonstrate how to efficiently use a single dataset to collaborate between multiple departments with different data requirements
- Demonstrate how to reduce amount of code needed by using replicable functions and simple for loops
- Provide self practice opportunity for creating a fictional case study and build an RMarkdown report highlighting results

Sample initial dataset: anh_environs.csv

Script: anh_case_study.R

Analysis: anh_case_study.Rmd

Sample of initial dataset

```
## 'data.frame':    48 obs. of  10 variables:
##  $ Phase          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ upgrade_date   : chr  "January 24, 2019" "January 24, 2019" "January 24, 2019" "January 24, 2019"
##  $ first_name     : chr  "Obi-wan" "Obi-wan" "Obi-wan" "Yoda" ...
##  $ last_name      : chr  "Kenobi" "Kenobi" "Kenobi" "" ...
##  $ comlink_address: chr  "old.ben@dunesea.net" "old.ben@dunesea.net" "old.ben@dunesea.net" "shawty@da
##  $ account        : chr  "Jedi" "Jedi" "Jedi" "Jedi" ...
##  $ environs       : chr  "master" "council" "lightsaber" "master" ...
##  $ plan           : chr  "Light" "Light" "Light" "Light" ...
##  $ planet         : chr  "Tatooine" "Tatooine" "Tatooine" "Dagobah" ...
##  $ php_version    : num  5.6 5.6 7 7 7 5.6 5.6 7 5.6 7.2 ...
```

```
##   Phase     upgrade_date first_name last_name     comlink_address account
## 1     1 January 24, 2019    Obi-wan    Kenobi old.ben@dunesea.net    Jedi
## 2     1 January 24, 2019    Obi-wan    Kenobi old.ben@dunesea.net    Jedi
## 3     1 January 24, 2019    Obi-wan    Kenobi old.ben@dunesea.net    Jedi
## 4     1 January 24, 2019       Yoda            shawty@dagobah.edu    Jedi
## 5     1 January 24, 2019       Yoda            shawty@dagobah.edu    Jedi
## 6     1 January 24, 2019       Luke Skywalker farmboy@dunesea.net   Rebel
##     environs  plan   planet php_version
## 1     master Light Tatooine         5.6
## 2    council Light Tatooine         5.6
## 3 lightsaber Light Tatooine         7.0
## 4     master Light  Dagobah         7.0
## 5   swamprat Light  Dagobah         7.0
## 6      pilot Light Tatooine         5.6
```

Eliminating repetitious code with functions and simple loops

Intially the project script included repetitious code in the amount of nearly 100 lines that looked similar to the following sample:

```
  as.data.frame(df_concat) %>%
    filter(Phase == 1) %>%
    separate_rows(environments_concat, sep = ",") %>%
    rename(environs = environments_concat) %>%
    select(environs) %>%
    unique()
```

```
##      environs
## 1  lightsaber
## 2       pilot
## 3       rebel
## 4       handy
## 5      hermit
## 6    diplomat
## 7     general
## 9  resistance
## 10     master
## 11    council
## 14   swamprat
```

```
  as.data.frame(df_concat) %>%
    filter(Phase == 2) %>%
    separate_rows(environments_concat, sep = ",") %>%
    rename(environs = environments_concat) %>%
    select(environs) %>%
    unique()
```

```
##         environs
## 1       smuggler
## 2        gambler
## 4    nerf-herder
## 6 walking-carpet
```

After recognizing that the pattern of this same block of code extending for the entirety of the dataset to break out engineering groups for the ICE team, we decided to estimate how much code could be spared. As each phase required 6 lines of code to create an independent block, and there were 5 phases to consider, this amounted to 30 lines of repeated code. Reducing this with a single function and a for loop, reduced this block of code by 50%:

```
# set Phase to datatype factor and order by factor level
# this makes it possible to also order the viz in Phase order from top to bottom
df$Phase <- factor(df$Phase, levels = seq(max(df$Phase), min(df$Phase), by = -1))
# create a vector of range of unique Phase numbers, ordered by factor level
phases <- c(unique(df$Phase))

# Function to create engineering phase groups
create_engineering_phase_group <- function(p){
  as.data.frame(df_concat) %>%
    filter(Phase == p) %>%
    separate_rows(environments_concat, sep = ",") %>%
    rename(environs = environments_concat) %>%
    select(environs) %>%
    unique()
}
# For loop to create individual phase group dataframes
for (i in phases){
  i_group <- create_engineering_phase_group(i)
  assign(paste("Enginnering_phase_", i, sep = ""), i_group)
}
```

For the CCN team, 20 lines were reduced to the following (again, another 50% reduction in code):

```
# Function to create communications phase sets
create_comms_phase_group <- function(p){
  as.data.frame(df_concat) %>%
    filter(Phase == p)
}

# Loop to create phased comms groups
for (c in phases){
  c_group <- create_comms_phase_group(c)
  assign(paste("Comms_phase_", c, sep = ""), c_group)
}
```

Table count of environs by planet

```
df %>%
    group_by(planet) %>%
    count() %>%
    arrange(desc(n))
```

```
## # A tibble: 15 x 2
## # Groups:   planet [15]
##    planet              n
##    <chr>           <int>
##  1 Tatooine           13
##  2 Starkiller Base     6
##  3 Yavin               4
##  4 Coruscant           3
##  5 Jakku               3
##  6 Knights of Ren      3
##  7 Mustafar            3
##  8 Bespin              2
##  9 Black Squadron      2
## 10 Corellia            2
## 11 Dagobah             2
## 12 Kashyyyk            2
## 13 Chimera             1
## 14 Death Star          1
## 15 Dreadnaught         1
```

Chart of environs by Phase

```
as.data.frame(df) %>%
    select(Phase, environs) %>%
    arrange(Phase) %>%
    unique() %>%
    ggplot(aes(Phase)) +
    geom_bar(fill = "purple") +
    ggtitle("Environs by Phase") +
    theme(legend.position = "none") +
    coord_flip()
```

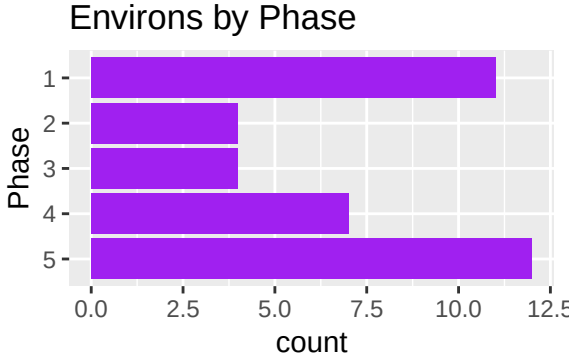

Chart of comlink messages by Phase

```
as.data.frame(df) %>%
    select(Phase, comlink_address) %>%
    arrange(Phase) %>%
    unique() %>%
    ggplot(aes(Phase)) +
        geom_bar(fill = "red") +
        ggtitle("Contacts by Phase") +
        theme(legend.position = "none") +
        coord_flip()
```