

# Twitter

*Caroline Li, Jocelyn Hu, Natalie Labossier*

*April 24, 2019*

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(mosaic)
```

```
## Loading required package: lattice

## Loading required package: ggformula

## Loading required package: ggplot2

## Loading required package: ggstance

##
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':
##
##   geom_errorbarh, GeomErrorbarh

##
## New to ggformula? Try the tutorials:
##   learnr::run_tutorial("introduction", package = "ggformula")
##   learnr::run_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData

## Loading required package: Matrix

##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features. The original behavior of these functions should not be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
```

```
##
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
##
##     mean

## The following object is masked from 'package:ggplot2':
##
##     stat

## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median,
##     prop.test, quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum

library(ggplot2)
library(readr)
library(tidyr)

##
## Attaching package: 'tidyr'

## The following object is masked from 'package:Matrix':
##
##     expand

library(tidyverse)

## -- Attaching packages ----- tidyverse 1.2.1 --

## v tibble 2.1.1      v stringr 1.4.0
## v purrr 0.3.2       v forcats 0.4.0

## -- Conflicts ----- tidyverse_conflicts() --
## x mosaic::count() masks dplyr::count()
## x purrr::cross() masks mosaic::cross()
## x mosaic::do() masks dplyr::do()
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag() masks stats::lag()
## x mosaic::stat() masks ggplot2::stat()
## x mosaic::tally() masks dplyr::tally()
```

```
df<-read.csv("data_twitter - Sheet2.CSV")
```

```
day <- c("Wed","Wed","Thurs","Thurs","Fri","Fri","Wed","Wed","Thurs","Thurs","Fri","Fri","Wed","Wed","Thurs")
week <- c(rep("1", 6), rep("2",6), rep("3",6))
df<-cbind(day,df)
df<-cbind(week,df)
```

```
df<-df%>%
  mutate(sum_engage=likes_48+retweets_48+comments_48)%>%
  unite(cell, time, type,remove=FALSE)
```

```
glimpse(df)
```

```
## Observations: 18
## Variables: 10
## $ week      <fct> 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3
## $ day       <fct> Wed, Wed, Thurs, Thurs, Fri, Fri, Wed, Wed, Thurs,...
## $ date      <fct> 4/3, 4/3, 4/4, 4/4, 4/5, 4/5, 4/10, 4/10, 4/11, 4/...
## $ cell      <chr> "morning_informative", "afternoon_informative", "m...
## $ type      <fct> informative, informative, promotional, promotional...
## $ time      <fct> morning, afternoon, morning, afternoon, morning, a...
## $ likes_48   <int> 1, 1, 0, 0, 2, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0
## $ retweets_48 <int> 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
## $ comments_48 <int> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
## $ sum_engage <int> 2, 1, 0, 0, 2, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0
```

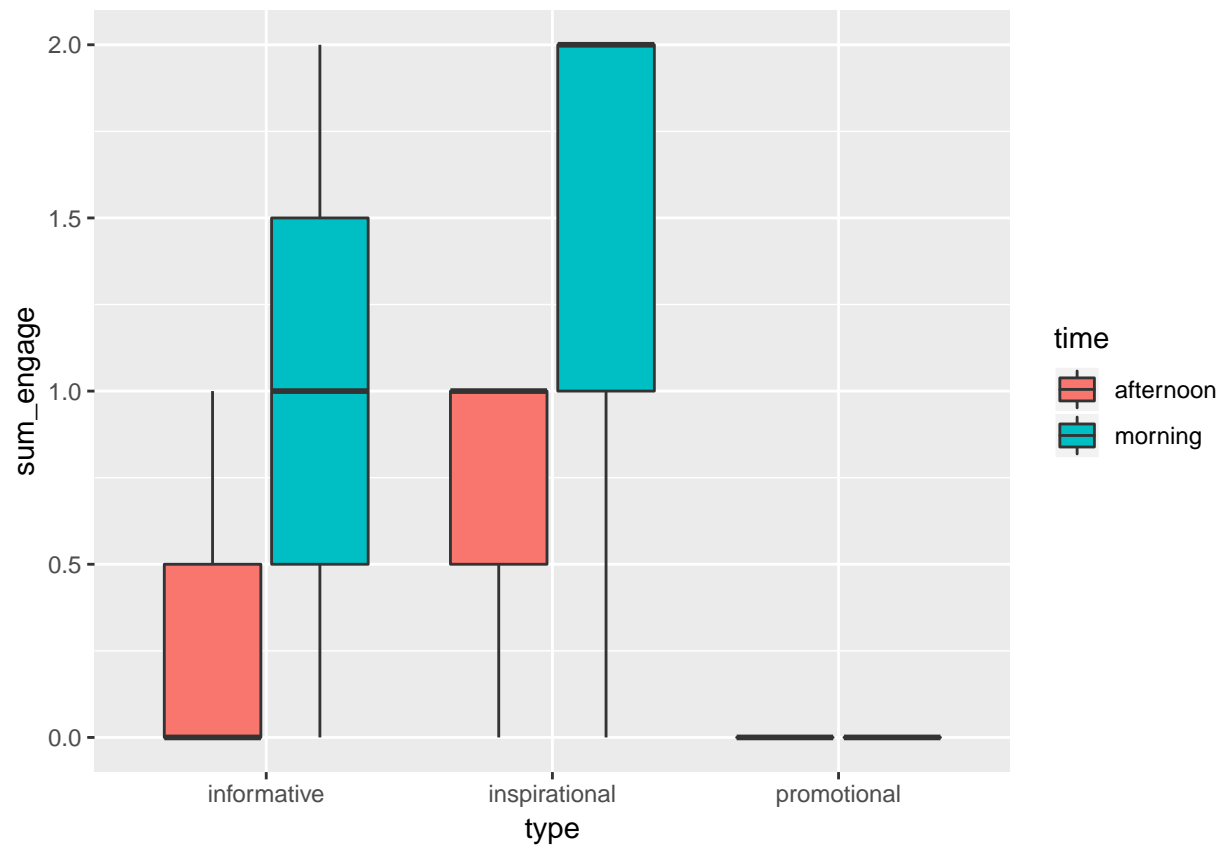
```
summary(df)
```

```
## week      day      date      cell      type
## 1:6  Fri :6  4/10 :2  Length:18      informative :6
## 2:6  Thurs:6  4/11 :2  Class :character inspirational:6
## 3:6  Wed  :6  4/12 :2  Mode  :character promotional  :6
##
##          4/17 :2
##          4/18 :2
##          4/19 :2
##          (Other):6
##      time      likes_48      retweets_48      comments_48
## afternoon:9  Min.    :0.0000  Min.    :0.00000  Min.    :0.00000
## morning :9   1st Qu.:0.0000  1st Qu.:0.00000  1st Qu.:0.00000
##           Median :0.0000  Median :0.00000  Median :0.00000
##           Mean   :0.4444  Mean    :0.05556  Mean    :0.05556
##           3rd Qu.:1.0000  3rd Qu.:0.00000  3rd Qu.:0.00000
##           Max.    :2.0000  Max.    :1.00000  Max.    :1.00000
##
##      sum_engage
## Min.    :0.0000
## 1st Qu.:0.0000
## Median :0.0000
## Mean    :0.5556
## 3rd Qu.:1.0000
## Max.    :2.0000
##
```

## Informal Analysis

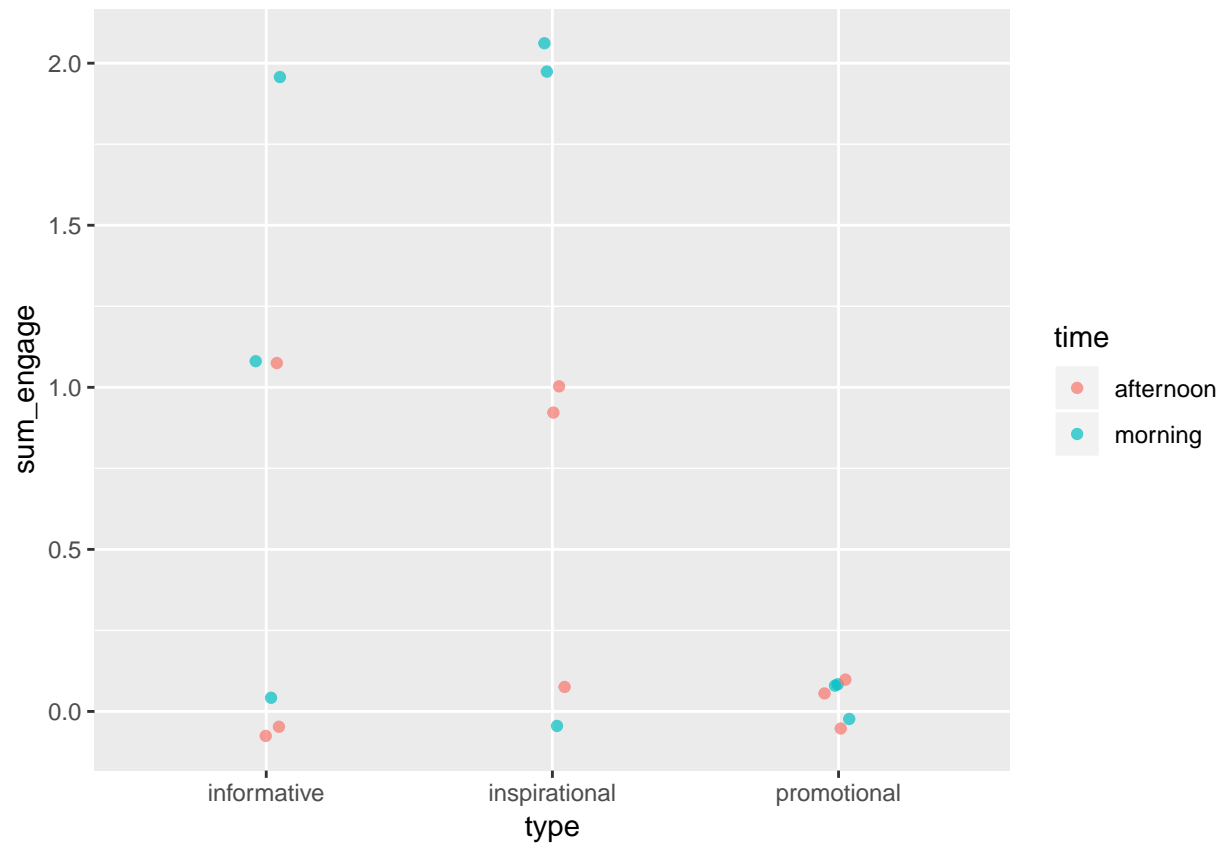
side by side boxplot

```
#looking at variability by condition  
ggplot(df, aes(x = type, fill = time, y = sum_engage)) +  
  geom_boxplot()
```



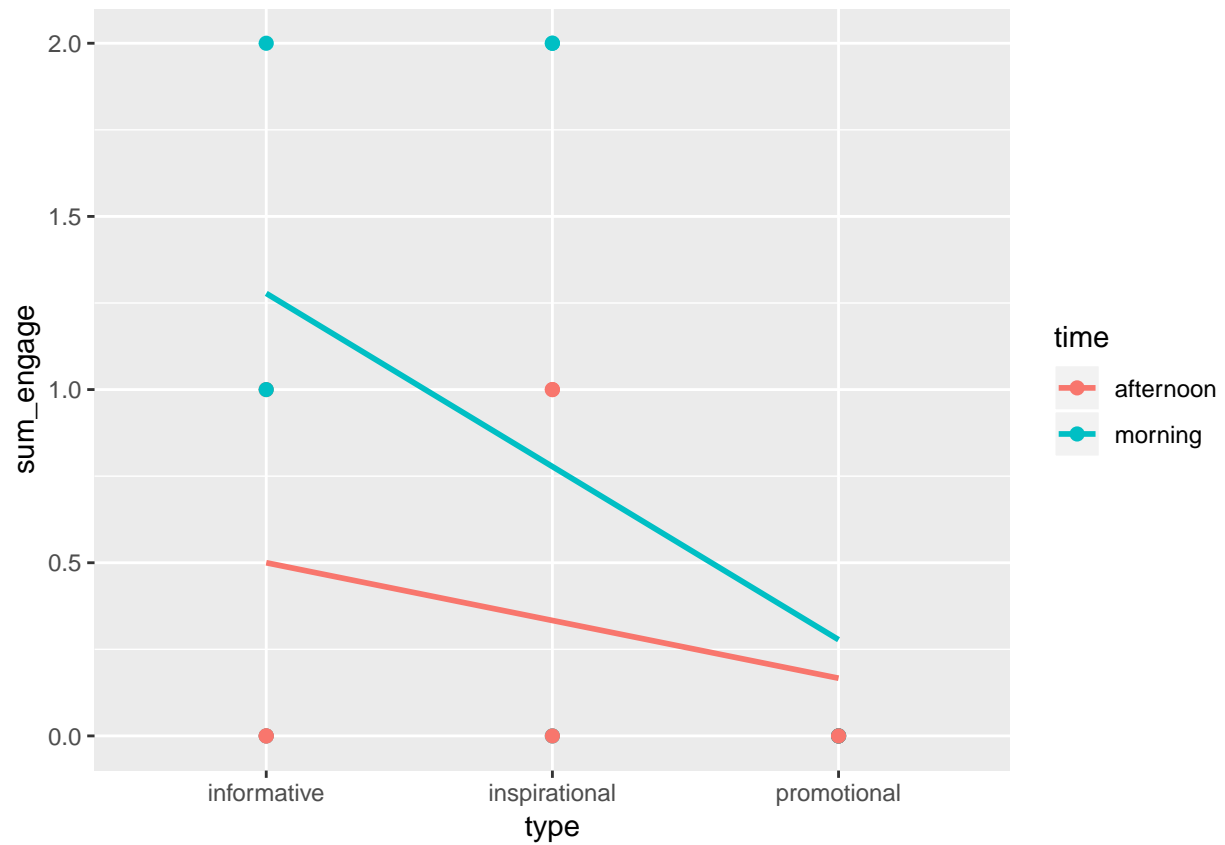
Parallel dot graph

```
#looking at variability by condition  
ggplot(df, aes(x = type, color = time, y = sum_engage)) +  
  geom_jitter(height = 0.10, width = 0.05, alpha = .7)
```

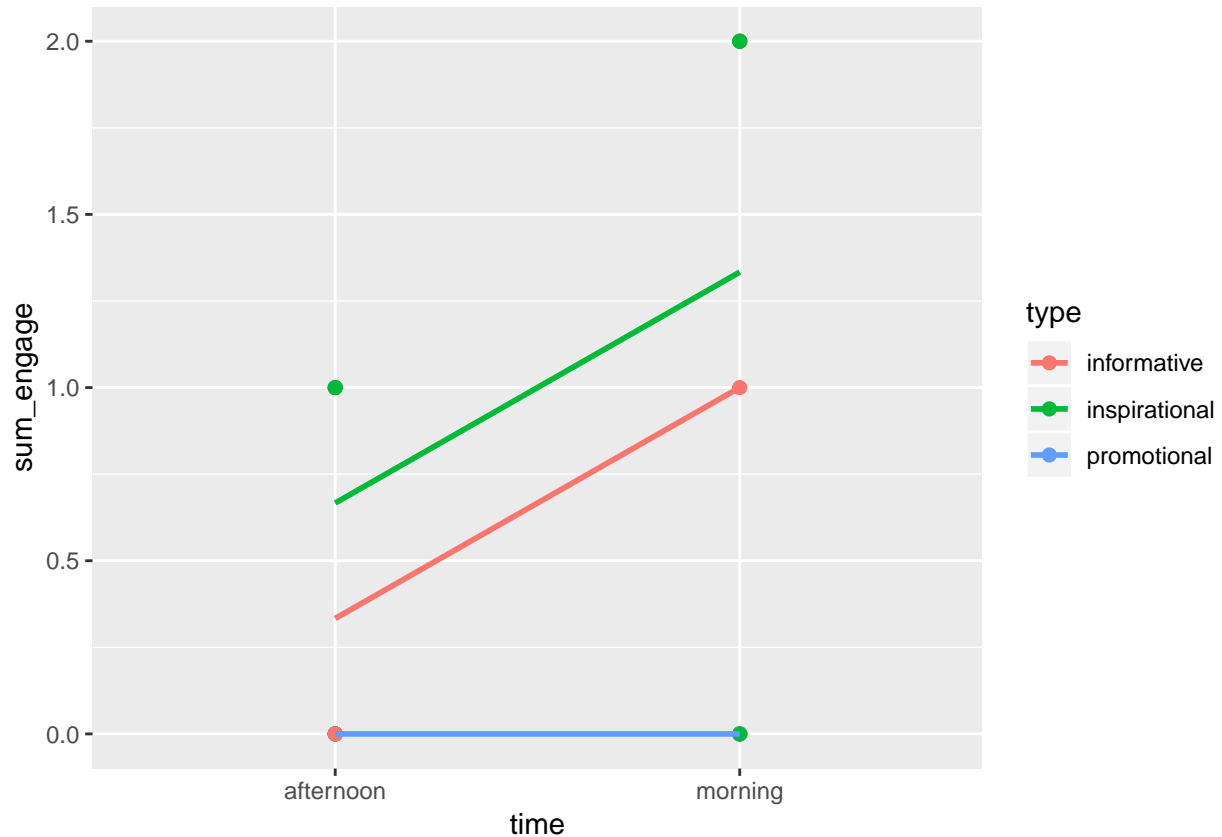


## Interaction effects

```
ggplot(df, aes(x = type,  
               y = sum_engage,  
               color = time,  
               group = time)) +  
  geom_point(size = 2) +  
  geom_smooth(method = "lm", se = 0)
```



```
ggplot(df, aes(x = time,  
               y = sum_engage,  
               color = type,  
               group = type)) +  
  geom_point(size = 2) +  
  geom_smooth(method = "lm", se = 0)
```



## Assembly Line Metaphor

### Assembly Line Instructions

To make the set of assembly instructions, first we calculate the effects of all of our structural factors under the Fisher assumptions. We use the data to calculate all of these effects.

```
df <- df %>%
  filter(type == "informative" | type == "inspirational")
```

```
df <- df %>%
  mutate(benchmark = mean(sum_engage)) #universal factor
```

```
df <- df %>%
  group_by(type) %>%
  mutate(type_mean = mean(sum_engage),
         type_effect = type_mean - benchmark)
#for each level of type factor
```

```
df <- df %>%
  group_by(week) %>%
  mutate(week_mean = mean(sum_engage),
```

```

        week_effect = week_mean - benchmark)
#for each level of type factor

df <- df %>%
  group_by(day) %>%
  mutate(day_mean = mean(sum_engage),
         day_effect = day_mean - benchmark)
#for each level of type factor

df <- df %>%
  group_by(time) %>%
  mutate(time_mean = mean(sum_engage),
         time_effect = time_mean - benchmark)
#for each level of type factor

df <- df %>%
  group_by(cell) %>%
  mutate(cell_mean = mean(sum_engage),
         interaction_effect = cell_mean - (type_mean + time_mean - benchmark)) #for each cell of interaction

df<- df %>%
  ungroup() %>%
  mutate(residuals = sum_engage - (benchmark
                                + type_effect
                                + time_effect
                                + interaction_effect
                                + day_effect
                                + week_effect))%>% #universal factor
  select(-type_mean, -time_mean,
         -cell_mean, -day_mean, -week_mean) #removing the stuff we don't need

df2<-df%>%
  filter(type == "informative" | type == "inspirational")

```

## ANOVA

```

twitter <- aov(sum_engage ~ type+time+day+week+type*time, data = df2)

anova1<-anova(twitter)
summary(twitter)

```

```

##           Df Sum Sq Mean Sq F value Pr(>F)
## type       1  0.333  0.3333      2 0.2302
## time       1  1.333  1.3333      8 0.0474 *
## day        2  4.667  2.3333     14 0.0156 *
## week       2  0.667  0.3333      2 0.2500
## type:time   1  0.000  0.0000      0 1.0000
## Residuals   4  0.667  0.1667
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```



```
#favstats(sum_engage~type/time, data = df2)

# library(mosaic)
# tally(~time/as.character(type), data = df2)
#
# df2$type
```

```
is.num <- sapply(anova1, is.numeric)
anova1[is.num] <- lapply(anova1[is.num], round, 3)
anova1[4,5]<-"0.250"
anova1[1,5]<-"0.230"
anova1[1,6]<-" "
anova1[2,6]<-"*"
anova1[3,6]<-"*"
anova1[4,6]<-" "
anova1[5,6]<-" "
anova1[6,6]<-" "
anova1[6,5]<-" "
anova1[6,4]<-" "
anova2<-anova1[1:6,]
```

```
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
## group_rows
```

```
kable(anova2, digits = 3, format = "latex", booktabs = T,col.names = c("Df","Sum Sq","Mean Sq","F value",
  kable_styling(full_width = F, font_size = 12)%>%add_footnote("Signif. codes: 0 '***' 0.001 '**' 0.01
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
type	1	0.333	0.333	2	0.230	
time	1	1.333	1.333	8	0.047	*
day	2	4.667	2.333	14	0.016	*
week	2	0.667	0.333	2	0.250	
type:time	1	0.000	0.000	0	1	
Residuals	4	0.667	0.167			

<sup>a</sup> Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## summary statistics

```
library(tidyverse)
df2<-as.data.frame(df)%>%
  na.omit()%>%
  select(sum_engage,likes_48,retweets_48,comments_48,type)
```

```
library(stargazer)
```

```
##
```

```
## Please cite as:
```

```
## Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
## R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
stargazer(data = df2, type = "html", title="Summary statistics of DV", digits=2, out="tablesum.html",co
```

```
##
```

```
## <table style="text-align:center"><caption><strong>Summary statistics of DV</strong></caption>
```

```
## <tr><td colspan="8" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left".
```

```
## <tr><td colspan="8" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left".
```

```
## <tr><td style="text-align:left">#Likes</td><td>12</td><td>0.67</td><td>0.78</td><td>0</td><td>0</td></tr>
```

```
## <tr><td style="text-align:left">#Retweets</td><td>12</td><td>0.08</td><td>0.29</td><td>0</td><td>0</td></tr>
```

```
## <tr><td style="text-align:left">#Comments</td><td>12</td><td>0.08</td><td>0.29</td><td>0</td><td>0</td></tr>
```

```
## <tr><td colspan="8" style="border-bottom: 1px solid black"></td></tr></table>
```