Replications of RQ2-4

Data preparation

We load the file with the scores from LIWC and rescale them in the range [1,5].

```
personality = read_delim(params$data, ";", escape_double = FALSE)
personality$openness <- resca(personality,openness,
                              new_min=1, new_max=5)$openness_res
personality$conscientiousness <- resca(personality,conscientiousness,
                                       new_min=1, new_max=5)$conscientiousness_res
personality$extraversion <- resca(personality,extraversion,
                                  new_min=1, new_max=5)$extraversion_res
personality$agreeableness <- resca(personality,agreeableness,
                                   new_min=1, new_max=5)$agreeableness_res
personality$neuroticism <- resca(personality,neuroticism,
                                 new_min=1, new_max=5)$neuroticism_res
```

Now we load the commit data and merge them with the personality data.

```
commit = read_delim(params$commits, ";", escape_double = FALSE)

# Find which developers appear in the intersection of the two data sets
both = intersect(unique(commit$uid), unique(personality$uid))

# Extract data only for the intersection developers
# (filter out people with 0 commits)
commit.both = subset(commit, uid %in% both &
                       num_authored_commits > 0)
# Count total number of projects and total number of commits across all projects, per person
commit.count = sqldf("select uid,
                     sum(num_authored_commits) as 'total_commits',
                     count(project) as 'total_projects',
                     max(last_authored_datetime) as last_authored_datetime,
                     min(first_integrated_datetime) as first_integrated_datetime
                     from `commit.both` group by uid")

# Filter out people who are still active (have at least one commit
# during the last 3 months before data collection). We can't know
# if they will remain one-time contributors or have more commits
commit.count = subset(commit.count, last_authored_datetime < as.POSIXct("2017-09-01 20:18:02"))

# Identify people with only one commit total, across all projects.
# These are the one-timers. The others are more active, even if they
# have projects with only one commit
one.timers = subset(commit.count, total_commits == 1)$uid
multi.timers = subset(commit.count, total_commits > 1)$uid
one.or.multi.timers = subset(commit.count, total_commits >= 1)$uid

# Assign a binary label "one_timer" to everyone in the personality
```

```
# data, based on the distinction above
p = subset(personality, uid %in% one.timers | uid %in% multi.timers)
p$one_timer = FALSE
for (i in 1:nrow(p)){
  p[i,]$one_timer = p[i,]$uid %in% one.timers
}

# Compute average personality scores per person, across time and across all projects
# We will use this simple data in the analysis below, because personality doesn't
# change much (see boxplots)
p.aggr = sqldf("select uid, avg(openness) as 'openness',
               avg(agreeableness) as 'agreeableness',
               avg(neuroticism) as 'neuroticism',
               avg(extraversion) as 'extraversion',
               avg(conscientiousness) as 'conscientiousness',
               sum(word_count) as word_count
               from p group by uid")

# Apply the "one_timer" binary label to this aggregate data set
p.aggr$one_timer = FALSE
for (i in 1:nrow(p.aggr)){
  p.aggr[i,]$one_timer = p.aggr[i,]$uid %in% one.timers
}
# Apply the "multi_timer" binary label to this aggregate data set
p.aggr$m_timer = FALSE
for (i in 1:nrow(p.aggr)){
  p.aggr[i,]$m_timer = p.aggr[i,]$uid %in% multi.timers
}
```

RQ2 Personality of core vs. peripheral developers

We perform a series o Wilcoxon Mann Whitney for unpaired groups, one for each trait. We adjust the significance with Bonferroni due to multiple comparisons. The differences are not significant (adjuste p-values all > 0.05).

```
integrations = subset(commit, uid %in% both & num_integrated_commits > 0)
integration.count = sqldf("select uid,
                          min(first_integrated_datetime) as first_integrated_datetime,
                          max(last_integrated_datetime) as last_integrated_datetime,
                          num_integrated_commits
                          from `integrations` group by uid")
integration.count = subset(integration.count, last_integrated_datetime < as.POSIXct("2017-09-01 20:18:0
integrators = subset(integration.count, num_integrated_commits >= 1)$uid

# peripheral
authors_non_integrators = unique(commit.both[!commit.both$uid %in% integrators, ]$uid)
c_b_n_i = Reduce(intersect, list(commit.both$uid, authors_non_integrators))
peripheral = personality[personality$uid %in% c_b_n_i, ]
peripheral = sqldf::sqldf("select uid,
                          avg(openness) as openness,
                          avg(conscientiousness) as conscientiousness,
                          avg(extraversion) as extraversion,
                          avg(agreeableness) as agreeableness,
                          avg(neuroticism) as neuroticism
```

```r
                          from peripheral
                          group by uid")
# core
c_b_i = Reduce(intersect, list(commit.both$uid, integrators))
core = personality[personality$uid %in% c_b_i, ]
core = sqldf::sqldf("select uid,
                          avg(openness) as openness,
                          avg(conscientiousness) as conscientiousness,
                          avg(extraversion) as extraversion,
                          avg(agreeableness) as agreeableness,
                          avg(neuroticism) as neuroticism
                          from core
                          group by uid")

print("**************************")
```

```
## [1] "**************************"
```

```r
print("          openness       ")
```

```
## [1] "          openness       "
```

```r
print("**************************")
```

```
## [1] "**************************"
```

```r
wto = wilcox.test(peripheral$openness, core$openness, conf.int=TRUE, paired=FALSE)
wto
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  peripheral$openness and core$openness
## W = 2233, p-value = 0.1
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.00919  0.08674
## sample estimates:
## difference in location
##                 0.0366
```

```r
#cliff.delta(peripheral$openness, core$openness, return.dm=TRUE)

print("**************************")
```

```
## [1] "**************************"
```

```r
print("    conscientiousness    ")
```

```
## [1] "    conscientiousness    "
```

```r
print("**************************")
```

```
## [1] "**************************"
```

```r
wtc = wilcox.test(peripheral$conscientiousness, core$conscientiousness, conf.int=TRUE, paired=FALSE)
wtc
```

```
##
```

```
##  Wilcoxon rank sum test with continuity correction
##
## data:  peripheral$conscientiousness and core$conscientiousness
## W = 1989, p-value = 0.7
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.0224  0.0339
## sample estimates:
## difference in location
##                0.00577
```

```
#cliff.delta(peripheral$conscientiousness, core$conscientiousness, return.dm=TRUE)

print("**************************")
```

```
## [1] "**************************"
```

```
print("       extraversion      ")
```

```
## [1] "       extraversion      "
```

```
print("**************************")
```

```
## [1] "**************************"
```

```
wte = wilcox.test(peripheral$extraversion, core$extraversion, conf.int=TRUE, paired=FALSE)
wte
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  peripheral$extraversion and core$extraversion
## W = 1902, p-value = 1
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.0407  0.0411
## sample estimates:
## difference in location
##              -0.000137
```

```
#cliff.delta(peripheral$extraversion, core$extraversion, return.dm=TRUE)

print("**************************")
```

```
## [1] "**************************"
```

```
print("       agreeableness     ")
```

```
## [1] "       agreeableness     "
```

```
print("**************************")
```

```
## [1] "**************************"
```

```
wta = wilcox.test(peripheral$agreeableness, core$agreeableness, conf.int=TRUE, paired=FALSE)
wta
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
```

```
## data:  peripheral$agreeableness and core$agreeableness
## W = 1685, p-value = 0.3
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.0639  0.0183
## sample estimates:
## difference in location
##                -0.024
```

```
#cliff.delta(peripheral$agreeableness, core$agreeableness, return.dm=TRUE)

print("**************************")
```

```
## [1] "**************************"
```

```
print("        neuroticism      ")
```

```
## [1] "        neuroticism      "
```

```
print("**************************")
```

```
## [1] "**************************"
```

```
wtn = wilcox.test(peripheral$neuroticism, core$neuroticism, conf.int=TRUE, paired=FALSE)
wtn
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  peripheral$neuroticism and core$neuroticism
## W = 1751, p-value = 0.4
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.0492  0.0212
## sample estimates:
## difference in location
##               -0.0139
```

```
#cliff.delta(peripheral$neuroticism, core$neuroticism, return.dm=TRUE)

print("Bonferroni")
```

```
## [1] "Bonferroni"
```

```
p.adjust(c(wto$p.value, wtc$p.value, wte$p.value, wta$p.value, wtn$p.value), method = "bonferroni")
```

```
## [1] 0.495 1.000 1.000 1.000 1.000
```

RQ3 Individual personality differences before and after becoming a project contributor

We use a series Wilcoxon Signed-Rank test for paired samples (before and after), repeated for each of the five traits. All the test returns differences taht are not statistically significant after Bonferroni corrections for multiple comparisons, as can also been observed from the box plots below.

```
df = data.frame(when=character(0), value=integer(0), stringsAsFactors = FALSE)

for (i in 1:length(integrators)){
```

```
  split_index = as.Date(integration.count[integration.count$uid == integrators[i], ]$first_integrated_da
  p_tot = personality[personality$uid == integrators[i],]
  p_prev = personality[personality$uid == integrators[i] & as.yearmon(personality$month) < as.yearmon(sp
  p_aft = personality[personality$uid == integrators[i] & as.yearmon(personality$month) >= as.yearmon(sp

  df[nrow(df)+1, ] = list("O before", mean(p_prev$openness))
  df[nrow(df)+1, ] = list("O after", mean(p_aft$openness))
  df[nrow(df)+1, ] = list("C before", mean(p_prev$conscientiousness))
  df[nrow(df)+1, ] = list("C after", mean(p_aft$conscientiousness))
  df[nrow(df)+1, ] = list("E before", mean(p_prev$extraversion))
  df[nrow(df)+1, ] = list("E after", mean(p_aft$extraversion))
  df[nrow(df)+1, ] = list("A before", mean(p_prev$agreeableness))
  df[nrow(df)+1, ] = list("A after", mean(p_aft$agreeableness))
  df[nrow(df)+1, ] = list("N before", mean(p_prev$neuroticism))
  df[nrow(df)+1, ] = list("N after", mean(p_aft$neuroticism))
}


df_o = df[df$when=="O before" | df$when=="O after", ]
df_o$when <- factor(df_o$when, levels = c("O before", "O after") )
ob = wilcox.test(df[df$when=="O before", ]$value, df[df$when=="O after", ]$value, correct = FALSE, pair
ob
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  df[df$when == "O before", ]$value and df[df$when == "O after", ]$value
## V = 62, p-value = 0.6
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.074  0.141
## sample estimates:
## (pseudo)median
##         0.0235
```

```
df_c = df[df$when=="C before" | df$when=="C after", ]
df_c$when <- factor(df_c$when, levels = c("C before", "C after") )
cb = wilcox.test(df[df$when=="C before", ]$value, df[df$when=="C after", ]$value, correct = FALSE, pair
cb
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  df[df$when == "C before", ]$value and df[df$when == "C after", ]$value
## V = 76, p-value = 0.2
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.0138  0.1124
## sample estimates:
## (pseudo)median
##         0.0302
```

```
df_e = df[df$when=="E before" | df$when=="E after", ]
df_e$when <- factor(df_e$when, levels = c("E before", "E after") )
eb = wilcox.test(df[df$when=="E before", ]$value, df[df$when=="E after", ]$value, correct = FALSE, pair
```

```
eb
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  df[df$when == "E before", ]$value and df[df$when == "E after", ]$value
## V = 46, p-value = 0.7
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.1264  0.0708
## sample estimates:
## (pseudo)median
##        -0.0217
```

```
df_a = df[df$when=="A before" | df$when=="A after", ]
df_a$when <- factor(df_a$when, levels = c("A before", "A after") )
ab=wilcox.test(df[df$when=="A before", ]$value, df[df$when=="A after", ]$value, correct = FALSE,  paire
ab
```

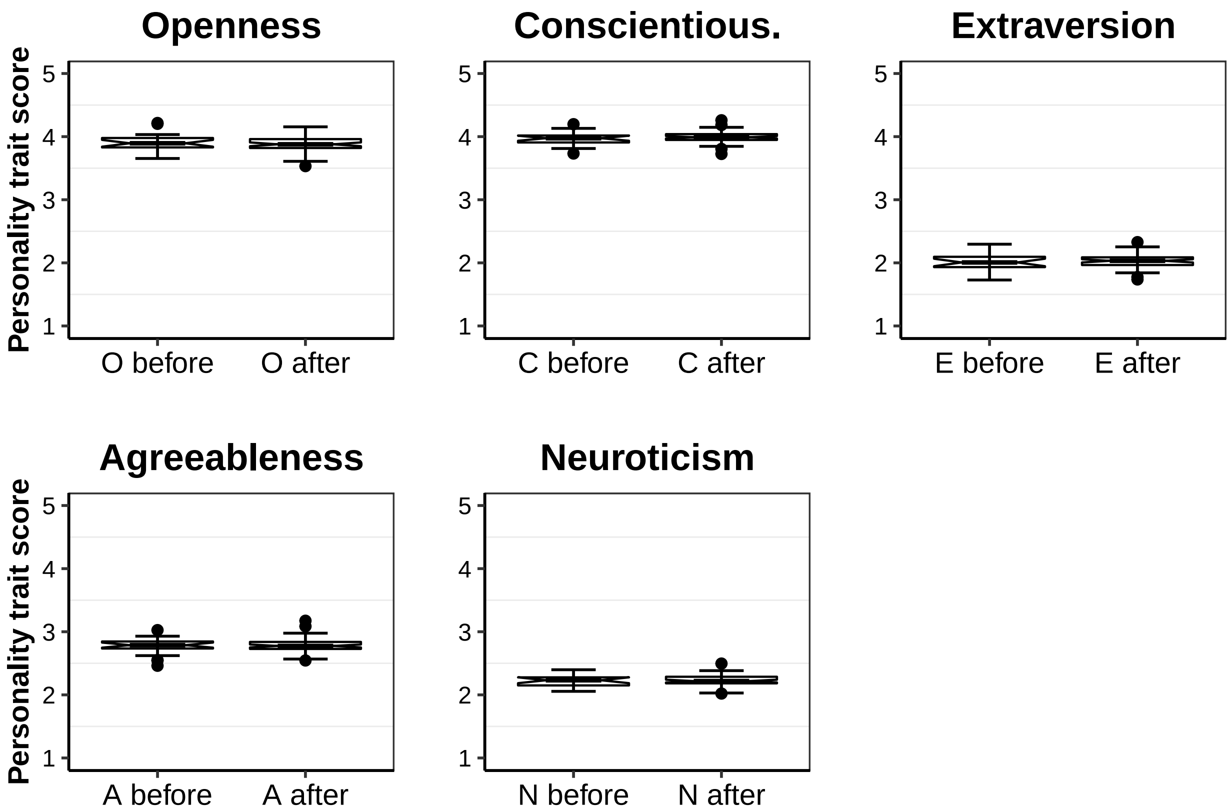
```
##
##  Wilcoxon signed rank exact test
##
## data:  df[df$when == "A before", ]$value and df[df$when == "A after", ]$value
## V = 49, p-value = 0.9
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.1212  0.0808
## sample estimates:
## (pseudo)median
##        -0.0126
```

```
df_n = df[df$when=="N before" | df$when=="N after", ]
df_n$when <- factor(df_n$when, levels = c("N before", "N after") )
nb = wilcox.test(df[df$when=="N before", ]$value, df[df$when=="N after", ]$value, correct = FALSE,  pai
nb
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  df[df$when == "N before", ]$value and df[df$when == "N after", ]$value
## V = 35, p-value = 0.3
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.1182  0.0178
## sample estimates:
## (pseudo)median
##        -0.0338
```

```
p.adjust(c(ob$p.value, cb$p.value, eb$p.value, ab$p.value, nb$p.value), method = "bonferroni")
```

```
## [1] 1.000 0.765 1.000 1.000 1.000
```



RQ4 — variation with the degree of development activity

Starting from the core and peripheral groups of developers of RQ2, we further split the groups into high vs. low contributors. We then perform the unpaired comparisons of the median personality scores between these high vs. low-activity developers within the core and peripheal groups. Results are in shown in the table below. The Wilcoxon Rank Sum tests reveal no cases of statistically significant differences between the pairs of trait distributions (i.e., adjusted p-values > 0.05 after Bonferroni correction).

```
csbo = sqldf::sqldf("select uid,
                         sum(num_authored_commits) as num_authored_commits,
                         sum(num_integrated_commits) as num_integrated_commits
                         from `commit.both`
                         group by uid")
# split peripheral in low vs higher commit authors around the mean
Mp = floor(mean(csbo$num_authored_commits))
peripheral_low = fn$sqldf("select *
                              from peripheral left join csbo on peripheral.uid = csbo.uid
                              where num_authored_commits <= $Mp")
peripheral_high = fn$sqldf("select *
                              from peripheral left join csbo on peripheral.uid = csbo.uid
                              where num_authored_commits > $Mp")


wtc_hlpo = wilcox.test(peripheral_high$openness, peripheral_low$openness, conf.int=TRUE, paired=FALSE)
#cliff.delta(peripheral_high$openness, peripheral_low$openness, return.dm=TRUE)

wtc_hlpc = wilcox.test(peripheral_high$conscientiousness, peripheral_low$conscientiousness, conf.int=TRU
#cliff.delta(peripheral_high$conscientiousness, peripheral_low$conscientiousness, return.dm=TRUE)
```

```
wtc_hlpe = wilcox.test(peripheral_high$extraversion, peripheral_low$extraversion, conf.int=TRUE, paired=
#cliff.delta(peripheral_high$extraversion, peripheral_low$extraversion, return.dm=TRUE)

wtc_hlpa = wilcox.test(peripheral_high$agreeableness, peripheral_low$agreeableness, conf.int=TRUE, paire
#cliff.delta(peripheral_high$agreeableness, peripheral_low$agreeableness, return.dm=TRUE)

wtc_hlpn = wilcox.test(peripheral_high$neuroticism, peripheral_low$neuroticism, conf.int=TRUE, paired=F
#cliff.delta(peripheral_high$neuroticism, peripheral_low$neuroticism, return.dm=TRUE)

adj_p <- p.adjust(c(wtc_hlpo$p.value, wtc_hlpc$p.value, wtc_hlpe$p.value, wtc_hlpa$p.value, wtc_hlpn$p.


df <- data.frame(matrix(ncol = 5, nrow = 12))
x <- c("Trait", "W", "p-value ", "CI 95% low", "CI 95% high")
colnames(df) <- x
df[1,] <- c("High vs. low commit authors (peripheral)", "", "", "", "")
df[2,] <- c("openness", as.numeric(wtc_hlpo$statistic), adj_p[1], round(as.numeric(wtc_hlpo$conf.int), 
df[3,] <- c("conscientiousness", as.numeric(wtc_hlpc$statistic), adj_p[2], round(as.numeric(wtc_hlpc$co
df[4,] <- c("extraversion", as.numeric(wtc_hlpo$statistic), adj_p[3], round(as.numeric(wtc_hlpe$conf.in
df[5,] <- c("agreeableness", as.numeric(wtc_hlpa$statistic), adj_p[4], round(as.numeric(wtc_hlpa$conf.i
df[6,] <- c("neuroticism", as.numeric(wtc_hlpn$statistic), adj_p[5], round(as.numeric(wtc_hlpn$conf.int


# split core in low vs higher commit integrators around the mean
Mc = ceiling(mean(csbo$num_integrated_commits))
core_low = fn$sqldf("select *
                        from core left join csbo on core.uid = csbo.uid
                        where num_authored_commits <= $Mc")
core_high = fn$sqldf("select *
                        from core left join csbo on core.uid = csbo.uid
                        where num_authored_commits > $Mc")

wtc_hlco = wilcox.test(core_high$openness, core_low$openness, conf.int=TRUE, paired=FALSE)
#cliff.delta(core_high$openness, core_low$openness, return.dm=TRUE)

wtc_hlcc = wilcox.test(core_high$conscientiousness, core_low$conscientiousness, conf.int=TRUE, paired=F
#cliff.delta(core_high$conscientiousness, core_low$conscientiousness, return.dm=TRUE)

wtc_hlce = wilcox.test(core_high$extraversion, core_low$extraversion, conf.int=TRUE, paired=FALSE)
#cliff.delta(core_high$extraversion, core_low$extraversion, return.dm=TRUE)

wtc_hlca = wilcox.test(core_high$agreeableness, core_low$agreeableness, conf.int=TRUE, paired=FALSE)
#cliff.delta(core_high$agreeableness, core_low$agreeableness, return.dm=TRUE)

wtc_hlcn = wilcox.test(core_high$neuroticism, core_low$neuroticism, conf.int=TRUE, paired=FALSE)
#cliff.delta(core_high$neuroticism, core_low$neuroticism, return.dm=TRUE)

adj_p<-p.adjust(c(wtc_hlco$p.value, wtc_hlcc$p.value, wtc_hlce$p.value, wtc_hlca$p.value, wtc_hlcn$p.va

df[7,] <- c("High vs. low commit authors (core)", "", "", "", "")
df[8,] <- c("openness", as.numeric(wtc_hlco$statistic), adj_p[1], round(as.numeric(wtc_hlco$conf.int), 
df[9,] <- c("conscientiousness", as.numeric(wtc_hlcc$statistic), adj_p[2], round(as.numeric(wtc_hlcc$co
df[10,] <- c("extraversion", as.numeric(wtc_hlce$statistic), adj_p[3], round(as.numeric(wtc_hlce$conf.i
```

```r
df[11,] <- c("agreeableness", as.numeric(wtc_hlca$statistic), adj_p[4], round(as.numeric(wtc_hlca$conf.i
df[12,] <- c("neuroticism", as.numeric(wtc_hlcn$statistic), adj_p[5], round(as.numeric(wtc_hlcn$conf.in

df
```

```
##                                       Trait   W p-value  CI 95% low CI 95% high
## 1  High vs. low commit authors (peripheral)
## 2                                  openness 520       1     -0.039       0.122
## 3                         conscientiousness 557       1     -0.021       0.078
## 4                              extraversion 520       1     -0.115       0.041
## 5                             agreeableness 489       1     -0.078         0.1
## 6                               neuroticism 389       1     -0.084       0.035
## 7        High vs. low commit authors (core)
## 8                                  openness 165       1     -0.139       0.046
## 9                         conscientiousness 253       1     -0.035       0.121
## 10                             extraversion 242       1      -0.05       0.084
## 11                            agreeableness 242       1     -0.052       0.102
## 12                              neuroticism 183       1     -0.105       0.038
```