ML Code Completeness Checklist Analysis

This notebook contains the ML Code Completeness analysis for NeurIPS 2019 repositories.

For a run & rendered version of this notebook please see: code_checklist-analysis.pdf.

Official repositories for NeurIPS 2019 papers fetched from: https://papers.nips.cc/book/advances-in-neural-information-processing-systems-32-2019

A random 25% sample has been selected and manually annotated according to the 5 critera of the ML Code Completness Checklist. The result has been saved into `code_checklist-neurips2019.csv`.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## -- Conflicts -----------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(MASS)
```

```
##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(RColorBrewer)

t = read_csv("code_checklist-neurips2019.csv")
```

```
## Parsed with column specification:
## cols(
##   url = col_character(),
##   stars = col_double(),
##   python = col_double(),
##   training = col_double(),
##   evaluation = col_double(),
##   pretrained_model = col_double(),
##   results = col_double(),
##   dependencies = col_double()
## )
```

```
cat("Number of rows:", nrow(t), "\n")
```

```
## Number of rows: 221
```

We'll focus only on Python repositories, since this is the dominant language in ML and repositories in other languages tend to have a smaller number of stars just because the community is smaller.

```
t = t[t$python==1,]
cat("Number of rows:", nrow(t), "\n")
```

```
## Number of rows: 200
```

Next, we calculate the score as a sum of of individual checklist items and calculate summary stats.

```
t$score = rowSums(t[,4:8])
```

We group repositories based on their score and calculate summary stats.

```
cat("Spread of values in each group:\n")
```

```
## Spread of values in each group:
```

```
summaries = tapply(t$stars, t$score, summary)
names(summaries) = paste(names(summaries), "ticks")
print(summaries)
```

```
## $`0 ticks`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     1.5    14.5    10.0    89.0
##
## $`1 ticks`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    5.00   11.94   11.00   59.00
##
## $`2 ticks`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    4.00   15.00   43.17   30.00  654.00
##
## $`3 ticks`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    6.00   19.00  171.15   75.75 6082.00
##
## $`4 ticks`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   22.25   62.50  457.88  148.50 5114.00
##
## $`5 ticks`
##     Min. 1st Qu.  Median     Mean 3rd Qu.     Max.
##    16.00   93.25  196.50  2664.89  517.00 36549.00
```

```
cat("Proportion of repos in each group:\n")
```

```
## Proportion of repos in each group:
```

```
props = tapply(t$stars, t$score, length)
props = props/sum(props)
names(props) = paste(names(props), "ticks")
print(props)
```

```
## 0 ticks 1 ticks 2 ticks 3 ticks 4 ticks 5 ticks
##   0.050   0.085   0.205   0.360   0.210   0.090
```

```
# Extract medians
medians = unlist(lapply(tapply(t$stars, t$score, summary), function(x) x["Median"]))
names(medians) = paste(sub(".Median", "", names(medians)), "ticks")
```

Generate summary graphs.

```
par(oma=c(0,1,0,1))
layout(matrix(c(1,2), 1, 2, byrow = TRUE), widths=c(3,2))
barplot(medians,
        xlab="",
        ylab="Median GitHub stars", ylim=c(0,200),
        col=brewer.pal(6, "Blues"), cex.axis=0.6, cex.names=0.6)
mtext("GitHub repos grouped by number of ticks on ML code checklist", side=1, line=3, cex=0.8)

pie(rev(props), col=rev(brewer.pal(6, "Blues")), cex=0.6)
mtext("Proportion of repositories in each group", side=1, line=3, cex=0.8)
```
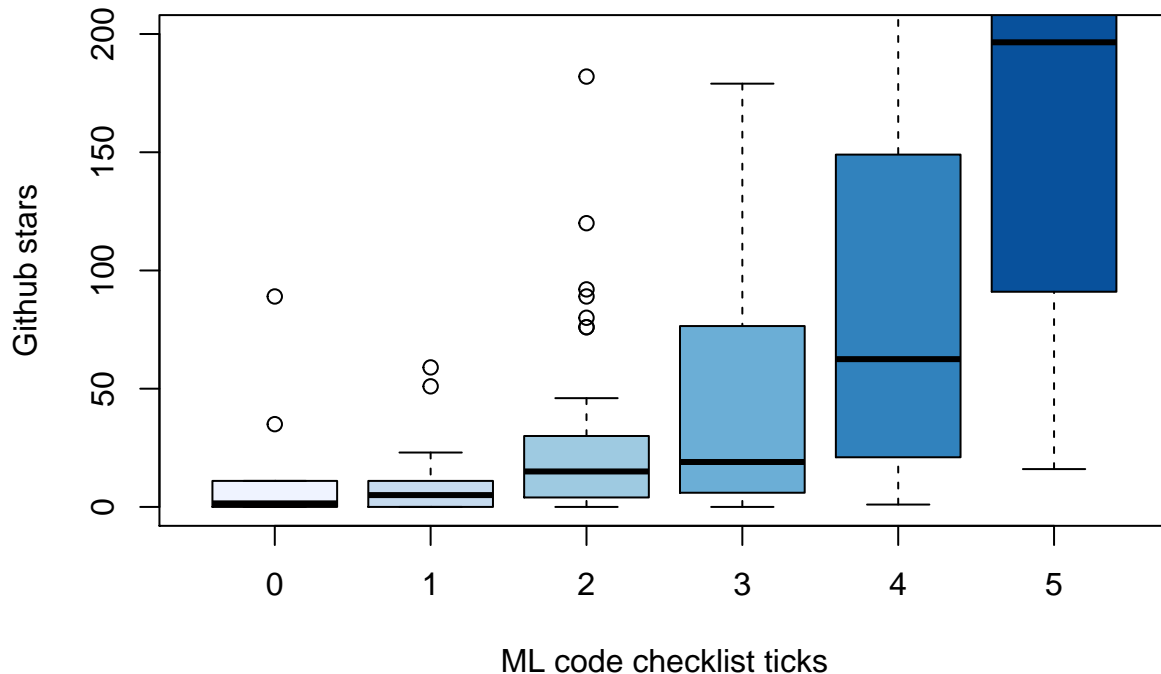

GitHub repos grouped by number of ticks on ML code checklist   Proportion of repositories in each group
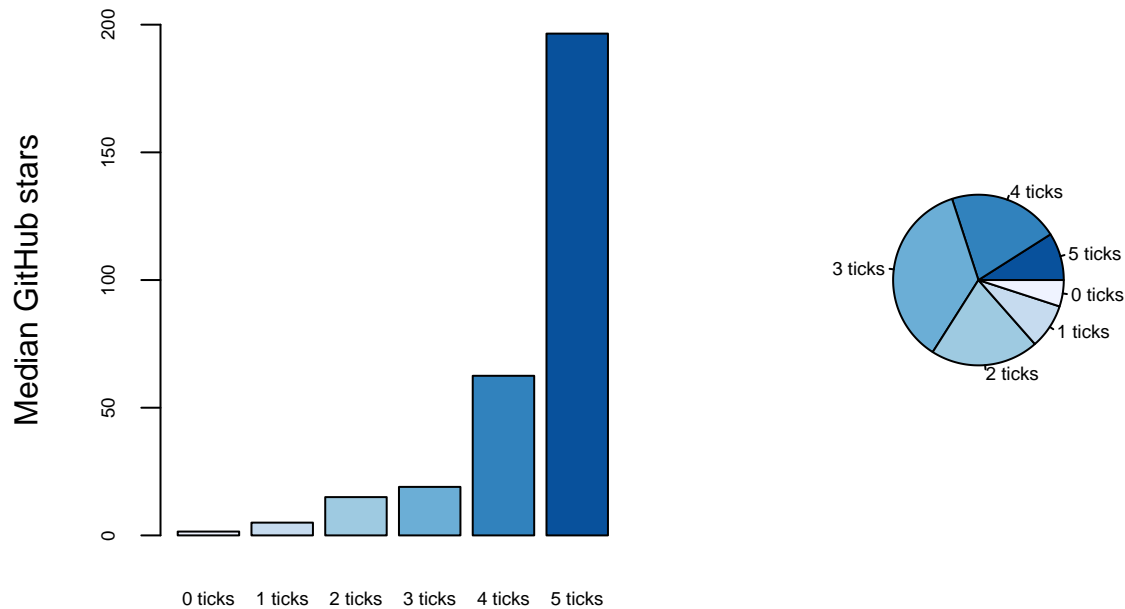
Compare using box plots.

```
tp = t
tp$score = as.factor(tp$score)
par(mfrow=c(1,1))
boxplot(stars~score, data=t, ylim=c(0,200), col=brewer.pal(6, "Blues"),
        xlab="ML code checklist ticks", ylab="Github stars")
```

Github stars — ML code checklist ticks

Fit robust regression and test significance of results

```r
print(summary(rlm(stars~training+evaluation+pretrained_model+results+dependencies, data=t)))
```

```
##
## Call: rlm(formula = stars ~ training + evaluation + pretrained_model +
##     results + dependencies, data = t)
## Residuals:
##       Min       1Q    Median       3Q       Max
##  -118.293   -25.391    -7.406   36.218 36414.707
##
## Coefficients:
##                   Value    Std. Error t value
## (Intercept)       -1.0246  11.5557    -0.0887
## training          24.3908  11.8245     2.0627
## evaluation       -12.0504   8.8434    -1.3626
## pretrained_model  70.3466   9.1685     7.6726
## results           36.7966   8.8318     4.1664
## dependencies      15.8344   9.2208     1.7172
##
## Residual standard error: 40.03 on 194 degrees of freedom
```

```r
for(i in 0:4){
  cat("\nScore5 vs Score", i, "\n")
  print(wilcox.test(t$stars[t$score==5], t$stars[t$score==i]))
}
```

```
##
## Score5 vs Score 0

## Warning in wilcox.test.default(t$stars[t$score == 5], t$stars[t$score == :
## cannot compute exact p-value with ties

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

```
## 
## data:  t$stars[t$score == 5] and t$stars[t$score == i]
## W = 174, p-value = 5.943e-05
## alternative hypothesis: true location shift is not equal to 0
## 
## 
## Score5 vs Score 1

## Warning in wilcox.test.default(t$stars[t$score == 5], t$stars[t$score == :
## cannot compute exact p-value with ties

## 
##  Wilcoxon rank sum test with continuity correction
## 
## data:  t$stars[t$score == 5] and t$stars[t$score == i]
## W = 300, p-value = 1.279e-06
## alternative hypothesis: true location shift is not equal to 0
## 
## 
## Score5 vs Score 2

## Warning in wilcox.test.default(t$stars[t$score == 5], t$stars[t$score == :
## cannot compute exact p-value with ties

## 
##  Wilcoxon rank sum test with continuity correction
## 
## data:  t$stars[t$score == 5] and t$stars[t$score == i]
## W = 677, p-value = 4.1e-07
## alternative hypothesis: true location shift is not equal to 0
## 
## 
## Score5 vs Score 3
## 
##  Wilcoxon rank sum test with continuity correction
## 
## data:  t$stars[t$score == 5] and t$stars[t$score == i]
## W = 1082, p-value = 1.22e-05
## alternative hypothesis: true location shift is not equal to 0
## 
## 
## Score5 vs Score 4

## Warning in wilcox.test.default(t$stars[t$score == 5], t$stars[t$score == :
## cannot compute exact p-value with ties

## 
##  Wilcoxon rank sum test with continuity correction
## 
## data:  t$stars[t$score == 5] and t$stars[t$score == i]
## W = 528.5, p-value = 0.01551
## alternative hypothesis: true location shift is not equal to 0
```

Session information

```
sessionInfo()
```

```
## R version 3.6.2 (2019-12-12)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Catalina 10.15.3
##
## Matrix products: default
## BLAS:    /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] RColorBrewer_1.1-2 MASS_7.3-51.4     forcats_0.4.0     stringr_1.4.0
##  [5] dplyr_0.8.4        purrr_0.3.3       readr_1.3.1       tidyr_1.0.2
##  [9] tibble_2.1.3       ggplot2_3.2.1     tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.0.0 xfun_0.12        haven_2.2.0      lattice_0.20-38
##  [5] colorspace_1.4-1 vctrs_0.2.3      generics_0.0.2   htmltools_0.4.0
##  [9] yaml_2.2.1       rlang_0.4.4      pillar_1.4.3     withr_2.1.2
## [13] glue_1.3.1       DBI_1.1.0        dbplyr_1.4.2     modelr_0.1.6
## [17] readxl_1.3.1     lifecycle_0.1.0  munsell_0.5.0    gtable_0.3.0
## [21] cellranger_1.1.0 rvest_0.3.5      evaluate_0.14    knitr_1.28
## [25] fansi_0.4.1      broom_0.5.4      Rcpp_1.0.3       scales_1.1.0
## [29] backports_1.1.5  jsonlite_1.6.1   fs_1.3.1         hms_0.5.3
## [33] digest_0.6.25    stringi_1.4.6    grid_3.6.2       cli_2.0.1
## [37] tools_3.6.2      magrittr_1.5     lazyeval_0.2.2   crayon_1.3.4
## [41] pkgconfig_2.0.3  xml2_1.2.2       reprex_0.3.0     lubridate_1.7.4
## [45] assertthat_0.2.1 rmarkdown_2.1    httr_1.4.1       rstudioapi_0.11
## [49] R6_2.4.1         nlme_3.1-142     compiler_3.6.2
```