R Notebook

Running on Discovery

I recommend viewing this with a web-based Rstudio server on Discovery:

https://ood.discovery.neu.edu/pun/sys/dashboard/batch_connect/sys/RStudio/session_contexts/new

Press *Ctrl+Enter* to run a chunk.

Initialization

You may want to change the directories below.

```
suppressMessages(library(tidyverse))
library(stringr)
library(xtable)
suppressMessages(library(extrafont))
library(fontcm)
```

```
data_root <- "/home/donald/data-from-discovery-rsync/apr-10-more-combos"

perf_root <- "/home/donald/apr-10-more-combos-2"
oldness_root <- str_c(data_root, "-results")


results_root <- str_c(data_root, "-number-results")
dir.create(results_root, showWarnings = FALSE)

tables_dir <- str_c(data_root, "-tables")
dir.create(tables_dir, showWarnings = FALSE)

plots_dir <- str_c(data_root, "-plots")
dir.create(plots_dir, showWarnings = FALSE)


results_tex <- str_c(results_root, "/results.tex")

write("% These are results from the R Notebook.", results_tex, append=FALSE)
write("% Run the notebook from top to bottom", results_tex, append=TRUE)
```

Theme for output

```
mytheme <- function() {
  return(theme_bw() +
           theme(
             text = element_text(family = "CM Roman", size=10),
              panel.grid.major = element_blank(),
             # panel.grid.minor = element_blank(),
             # panel.grid.major = element_line(colour="gray", size=0.1),
             # panel.grid.minor =
             #  element_line(colour="gray", size=0.1, linetype='dotted'),
             axis.ticks = element_line(size=0.05),
             axis.ticks.length=unit("-0.05", "in"),
             axis.text.y = element_text(margin = margin(r = 5)),
             axis.text.x = element_text(hjust=1),
             legend.key = element_rect(colour=NA),
             legend.spacing = unit(0.001, "in"),
             legend.key.size = unit(0.2, "in"),
             legend.title = element_blank(),
             legend.position = c(0.75, .7),
             legend.background = element_blank()))
}

mysave <- function(filename) {
  path <- str_c(plots_dir, "/", filename)
  ggsave(path, width=3, height=3, units=c("in"))
  # embed_font(path)
}
```

Load the data

These are the results from running all experiments in parallel on Discovery. The timing information is *not* reliable.

```
raw_data <- read_csv(paste(data_root, "/results.csv", sep=""),
  col_types = cols(Status=col_factor(),
                   Project=col_factor(),
                   Rosette=col_logical(),
                   Consistency=col_factor(),
                   DisallowCycles=col_factor(),
                   Minimize=col_factor(),
                   Time=col_double(),
                   NDeps=col_integer()),
  show_col_types = FALSE)
```

We load more data later.

Manual Verification Step

Check that these are the factors that appear below:

1. *success*: everything worked!
2. *ERESOLVE*: depends on something that isn't in the repository
3. *ETARGET*: requires some other target architecture **verify**. Can also mean depending on something that doesn't exist.
4. *EBADPLATFORM*: requires some other platform (e.g., macOS)
5. *EUNSUPPORTEDPROTOCOL:* a dependency is in a format that NPM does not support
6. *unexpected*: something went wrong on Discovery. See experiment.out
7. *unavailable*: something went wrong and we didn't even capture the result. See experiment.out
8. *unsat*: Z3 failed on us

```
levels(raw_data$Status)
```

```
## [1] "success"              "ERESOLVE"             "ETARGET"
## [4] "EBADPLATFORM"         "EUNSUPPORTEDPROTOCOL" "timeout"
## [7] "unsat"                "unexpected"
```

```
levels(raw_data$Consistency)
```

```
## [1] ""      "npm"   "cargo" "pip"
```

```
levels(raw_data$Minimize)
```

```
## [1] ""                           "min_oldness,min_num_deps"
## [3] "min_num_deps,min_oldness"   "min_duplicates,min_oldness"
## [5] "min_oldness,min_duplicates" "min_oldness"
```

```
levels(raw_data$DisallowCycles)
```

```
## [1] ""                "allow_cycles"    "disallow_cycles"
```

Sanity check: there should be 1,000 of each kind of experiment.

```
num_experiments <- raw_data %>%
  group_by(Rosette,Minimize,Consistency,DisallowCycles) %>%
  summarize(Count = n()) %>%
  ungroup() %>%
  select(Count) %>%
  unique()
```

```
## `summarise()` has grouped output by 'Rosette', 'Minimize', 'Consistency'. You
## can override using the `.groups` argument.
```

```
stopifnot(nrow(num_experiments) == 1)
stopifnot(num_experiments[1] == 1000)
```

Failures

How many failures occur for each configuration? *See failures.tex.*

```
# failure_analysis <- raw_data %>%
#   filter(Status != "success") %>%
#   group_by(Rosette,Minimize,Consistency)

failure_analysis <- raw_data %>%
  filter(Status != "success") %>%
  group_by(Rosette,Minimize,Consistency,DisallowCycles) %>%
  summarise(Unsat = sum(Status == "unsat"),
            Timeout = sum(Status == "unavailable" | Status == "timeout"),
            Other = sum(Status != "unsat" & Status != "unavailable" & Status != "timeout")) %>%
  ungroup() %>%
  mutate(Solver = if_else(Rosette, "MinNPM", "NPM")) %>%
  rename(Minimization = Minimize) %>%
  select(-Rosette) %>%
  relocate(Solver,Consistency,DisallowCycles,Minimization,Unsat,Timeout,Other)
```

```
## `summarise()` has grouped output by 'Rosette', 'Minimize', 'Consistency'. You
## can override using the `.groups` argument.
```

```
print(xtable(as.data.frame(failure_analysis), type="latex"), include.rownames=FALSE, file=str_c(tables_
knitr::kable(failure_analysis)
```

Solver	Consistency	DisallowCycles	Minimization	Unsat	Timeout	Other
NPM				0	0	47
MinNPM	npm	allow_cycles	min_oldness,min_num_deps	0	27	1
MinNPM	npm	disallow_cycles	min_oldness,min_num_deps	0	27	1
MinNPM	cargo	allow_cycles	min_oldness,min_num_deps	3	54	1
MinNPM	cargo	disallow_cycles	min_oldness,min_num_deps	3	52	1
MinNPM	pip	allow_cycles	min_oldness,min_num_deps	19	54	1
MinNPM	pip	disallow_cycles	min_oldness,min_num_deps	19	54	1
MinNPM	npm	allow_cycles	min_num_deps,min_oldness	0	27	1
MinNPM	cargo	allow_cycles	min_num_deps,min_oldness	3	53	1
MinNPM	pip	allow_cycles	min_num_deps,min_oldness	19	54	1
MinNPM	npm	allow_cycles	min_duplicates,min_oldness	0	27	1
MinNPM	cargo	allow_cycles	min_duplicates,min_oldness	3	54	1
MinNPM	npm	allow_cycles	min_oldness,min_duplicates	0	26	1
MinNPM	cargo	allow_cycles	min_oldness,min_duplicates	3	53	1
MinNPM	npm	allow_cycles	min_oldness	0	25	1
MinNPM	cargo	allow_cycles	min_oldness	3	46	2
MinNPM	pip	allow_cycles	min_oldness	19	52	3

Results for the paper. These exclude PIP

```
failure_summary <- failure_analysis %>%
    mutate(Total = Unsat + Timeout + Other) %>%
  filter(Solver == "NPM" | Consistency == "npm") %>%
  select(Solver, Total, Consistency) %>%
  group_by(Solver) %>%
  summarize(Min = min(Total), Max = max(Total))

write(
  str_c("\\newcommand{\\dataNumNPMFailures}{",
        failure_summary %>% filter(Solver == "NPM") %>% select(Max),
        "}\n"),
  results_tex, append = TRUE)

write(
  str_c("\\newcommand{\\dataMinMinNPMFailures}{",
        failure_summary %>% filter(Solver == "MinNPM") %>% select(Min),
        "}\n"),
  results_tex, append = TRUE)

write(
  str_c("\\newcommand{\\dataMaxMinNPMFailures}{",
        failure_summary %>% filter(Solver == "MinNPM") %>% select(Max),
        "}\n"),
  results_tex, append = TRUE)
```

Projects that produced a Z3 unsat with Pip-consistency, but succeeded with Npm-consistency:

```
requires_multiple_versions <- raw_data %>%
  filter(Rosette == TRUE &
           Consistency == "pip" &
           Minimize == "min_oldness,min_num_deps" &
           DisallowCycles == "allow_cycles" &
           Status != "success") %>%
  select(Project) %>%
  inner_join(raw_data %>%
               filter(Rosette == TRUE &
                        Consistency == "npm" &
                        Minimize == "min_oldness,min_num_deps" &
                        DisallowCycles == "allow_cycles" &
                        Status == "success") %>%
               select(Project))
```

```
## Joining, by = "Project"
```

```
requires_multiple_versions
```

```
## # A tibble: 46 x 1
##    Project
##    <fct>
##  1 browserify-sign
##  2 jest-matcher-utils
##  3 jest-validate
##  4 @babel_plugin-transform-runtime
##  5 jest-haste-map
##  6 jest-changed-files
##  7 jest-diff
##  8 jest-serializer
##  9 pretty-format
## 10 nanomatch
## # ... with 36 more rows
```

```
fraction_require_npm_consistency <- nrow(requires_multiple_versions) /
  nrow(raw_data %>%  filter(Rosette == FALSE))
write(
  str_c("\\newcommand{\\dataFractionPIPUnsupported}{",
        round(fraction_require_npm_consistency * 100),
        "\\%}\n"),
  results_tex, append=TRUE)
```

Projects that failed in with MinNPM in NPM mode, but succeeded with NPM. The Status column shows the status with MinNPM. The status *unavailable* means a timeout, whereas *unexpected* likely means some kind of Z3 / Rosette crash.

```
minnpm_succeeds_npm_fails <- raw_data %>%
  filter(Rosette == TRUE &
           Consistency == "npm" &
           Minimize == "min_oldness,min_num_deps" &
           DisallowCycles == "allow_cycles" &
           Status != "success") %>%
  select(Project, Status) %>%
  inner_join(raw_data %>%
               filter(Rosette == FALSE &
                        Status == "success") %>%
               select(Project))
```

```
## Joining, by = "Project"
```

```
minnpm_succeeds_npm_fails
```

```
## # A tibble: 18 x 2
##    Project                      Status
##    <fct>                        <fct>
##  1 istanbul-lib-instrument      timeout
##  2 @eslint_eslintrc             timeout
##  3 jest-watcher                 timeout
##  4 @jest_test-result            timeout
##  5 @istanbuljs_load-nyc-config timeout
##  6 node-libs-browser            timeout
##  7 @jest_fake-timers            timeout
##  8 @babel_preset-env            timeout
##  9 jest-config                  timeout
## 10 crypto-browserify            timeout
## 11 jest                         timeout
## 12 jest-runner                  timeout
## 13 copy-concurrently            timeout
## 14 babel-plugin-istanbul        timeout
## 15 move-concurrently            timeout
## 16 @jest_test-sequencer         timeout
## 17 eslint                       timeout
## 18 jest-jasmine2                timeout
```

```
nrow(minnpm_succeeds_npm_fails)
```

```
## [1] 18
```

Projects that succeeded with MinNPM in NPM mode, but failed with NPM. The Status column shows the status with NPM. I've more carefully parsed the error codes from NPM. It is surprising, and nice, that there are nearly as many failures in this direction.

```
raw_data %>%
  filter(Rosette == TRUE &
           Consistency == "npm" &
           Minimize == "min_oldness,min_num_deps" &
           DisallowCycles == "allow_cycles" &
           Status == "success") %>%
  select(Project, NDeps) %>%
  inner_join(raw_data %>%
               filter(Rosette == FALSE &
                        Status != "success") %>%
               select(Project, Status))
```

```
## Joining, by = "Project"
```

```
## # A tibble: 37 x 3
##    Project              NDeps Status
##    <fct>                <int> <fct>
##  1 https-proxy-agent        3 ERESOLVE
##  2 exit                     0 ETARGET
##  3 cacache                 38 ETARGET
##  4 jest-matcher-utils      13 ETARGET
##  5 regexpp                  0 ERESOLVE
##  6 jest-haste-map          36 ETARGET
##  7 file-uri-to-path       0 ERESOLVE
##  8 request               42 ERESOLVE
##  9 jest-diff             12 ETARGET
## 10 date-fns               0 ERESOLVE
## # ... with 27 more rows
```

Can MinNPM produce fewer dependencies than NPM?

For each project, the number of dependencies with vanilla NPM, and with MinNPM configured to minimize #deps and oldness, in that order.

```
min_dep_analysis_tmp <-
  bind_rows(raw_data %>%
            filter(Rosette == FALSE & Status == "success") %>%
            select(Project,NDeps) %>%
            mutate(Solver="NPM"),
          raw_data %>%
            filter(Rosette == TRUE & Status == "success" & Consistency == "npm" & DisallowCycles == "al
                   Minimize == "min_num_deps,min_oldness") %>%
            select(Project, NDeps) %>%
            mutate(Solver="NPM_MinDepsOldness"),
          raw_data %>%
            filter(Rosette == TRUE & Status == "success" & Consistency == "npm" & DisallowCycles == "al
                   Minimize == "min_oldness") %>%
            select(Project, NDeps) %>%
            mutate(Solver="NPM_MinOldness"),
          raw_data %>%
            filter(Rosette == TRUE & Status == "success" & Consistency == "npm" & DisallowCycles == "al
                   Minimize == "min_duplicates,min_oldness") %>%
            select(Project, NDeps) %>%
            mutate(Solver="NPM_MinDuplicatesOldness"),
          raw_data %>%
            filter(Rosette == TRUE & Status == "success" & Consistency == "pip" & DisallowCycles == "al
                   Minimize == "min_oldness") %>%
            select(Project, NDeps) %>%
            mutate(Solver="PIP_MinOldness"),
          raw_data %>%
            filter(Rosette == TRUE & Status == "success" & Consistency == "cargo" & DisallowCycles == "
                   Minimize == "min_oldness") %>%
            select(Project, NDeps) %>%
            mutate(Solver="Cargo_MinOldness")) %>%
  pivot_wider(values_from=NDeps, names_from=Solver) %>%
  filter(NPM>0) %>%

  mutate(NPM_NPM_MinDepsOldness_Delta = NPM - NPM_MinDepsOldness) %>%
  mutate(NPM_NPM_MinDepsOldness_Shrinkage = NPM_MinDepsOldness / NPM) %>%

  mutate(NPM_NPM_MinOldness_Delta = NPM - NPM_MinOldness) %>%
  mutate(NPM_NPM_MinOldness_Shrinkage = NPM_MinOldness / NPM) %>%

  mutate(NPM_NPM_MinDuplicatesOldness_Delta = NPM - NPM_MinDuplicatesOldness) %>%
  mutate(NPM_NPM_MinDuplicatesOldness_Shrinkage = NPM_MinDuplicatesOldness / NPM) %>%

  mutate(NPM_PIP_MinOldness_Delta = NPM - PIP_MinOldness) %>%
  mutate(NPM_PIP_MinOldness_Shrinkage = PIP_MinOldness / NPM) %>%

  mutate(NPM_Cargo_MinOldness_Delta = NPM - Cargo_MinOldness) %>%
  mutate(NPM_Cargo_MinOldness_Shrinkage = Cargo_MinOldness / NPM) %>%

  na.omit()

min_dep_analysis_shrinkage <-
  min_dep_analysis_tmp %>%
  pivot_longer(cols = ends_with("Shrinkage"), names_to="shrinkage_comparison", values_to="Shrinkage") %
  mutate(Comparison=shrinkage_comparison) %>%
  select(Project,Comparison, Shrinkage)

min_dep_analysis_delta <-
  min_dep_analysis_tmp %>%
  pivot_longer(cols = ends_with("Delta"), names_to="delta_comparison", values_to="Delta") %>%
  mutate(Comparison=delta_comparison) %>%
  select(Project,Comparison, Delta)

min_dep_analysis_shrinkage
```

```
## # A tibble: 2,385 x 3
##    Project              Comparison                              Shrinkage
##    <fct>                <chr>                                       <dbl>
##  1 @babel_preset-react NPM_NPM_MinDepsOldness_Shrinkage            0.231
##  2 @babel_preset-react NPM_NPM_MinOldness_Shrinkage                0.231
##  3 @babel_preset-react NPM_NPM_MinDuplicatesOldness_Shrinkage      0.231
##  4 @babel_preset-react NPM_PIP_MinOldness_Shrinkage                0.231
##  5 @babel_preset-react NPM_Cargo_MinOldness_Shrinkage              0.231
##  6 nopt                 NPM_NPM_MinDepsOldness_Shrinkage            1
##  7 nopt                 NPM_NPM_MinOldness_Shrinkage                1
##  8 nopt                 NPM_NPM_MinDuplicatesOldness_Shrinkage      1
##  9 nopt                 NPM_PIP_MinOldness_Shrinkage                1
## 10 nopt                 NPM_Cargo_MinOldness_Shrinkage              1
## # ... with 2,375 more rows
```

These are cases where MinNPM produces significantly fewer dependences than NPM. We may want to dig into them further to explain why:

```
min_dep_analysis_delta %>%
  filter(Comparison=='NPM_NPM_MinDepsOldness_Delta') %>%
  arrange(desc(Delta)) %>%
  filter(Delta > 25)
```

```
## # A tibble: 17 x 3
##    Project                                          Comparison         Delta
##    <fct>                                            <chr>              <int>
##  1 @babel_preset-modules                            NPM_NPM_MinDepsOl~    47
##  2 @babel_plugin-proposal-export-namespace-from     NPM_NPM_MinDepsOl~    45
##  3 @babel_plugin-proposal-dynamic-import            NPM_NPM_MinDepsOl~    45
##  4 @babel_plugin-proposal-json-strings              NPM_NPM_MinDepsOl~    45
##  5 @babel_plugin-proposal-optional-catch-binding    NPM_NPM_MinDepsOl~    45
##  6 @babel_plugin-proposal-numeric-separator         NPM_NPM_MinDepsOl~    45
##  7 @babel_plugin-proposal-nullish-coalescing-operator  NPM_NPM_MinDepsOl~    45
##  8 @babel_plugin-proposal-logical-assignment-operators NPM_NPM_MinDepsOl~    45
##  9 @babel_plugin-transform-react-jsx                    NPM_NPM_MinDepsOl~    43
## 10 @babel_plugin-transform-named-capturing-groups-regex NPM_NPM_MinDepsOl~    42
## 11 @babel_plugin-transform-dotall-regex                 NPM_NPM_MinDepsOl~    42
## 12 @babel_plugin-proposal-optional-chaining             NPM_NPM_MinDepsOl~    42
## 13 @babel_plugin-proposal-unicode-property-regex        NPM_NPM_MinDepsOl~    42
## 14 @babel_plugin-transform-unicode-regex                NPM_NPM_MinDepsOl~    42
## 15 @babel_preset-react                                  NPM_NPM_MinDepsOl~    40
## 16 @babel_plugin-proposal-object-rest-spread            NPM_NPM_MinDepsOl~    37
## 17 assert                                               NPM_NPM_MinDepsOl~    33
```

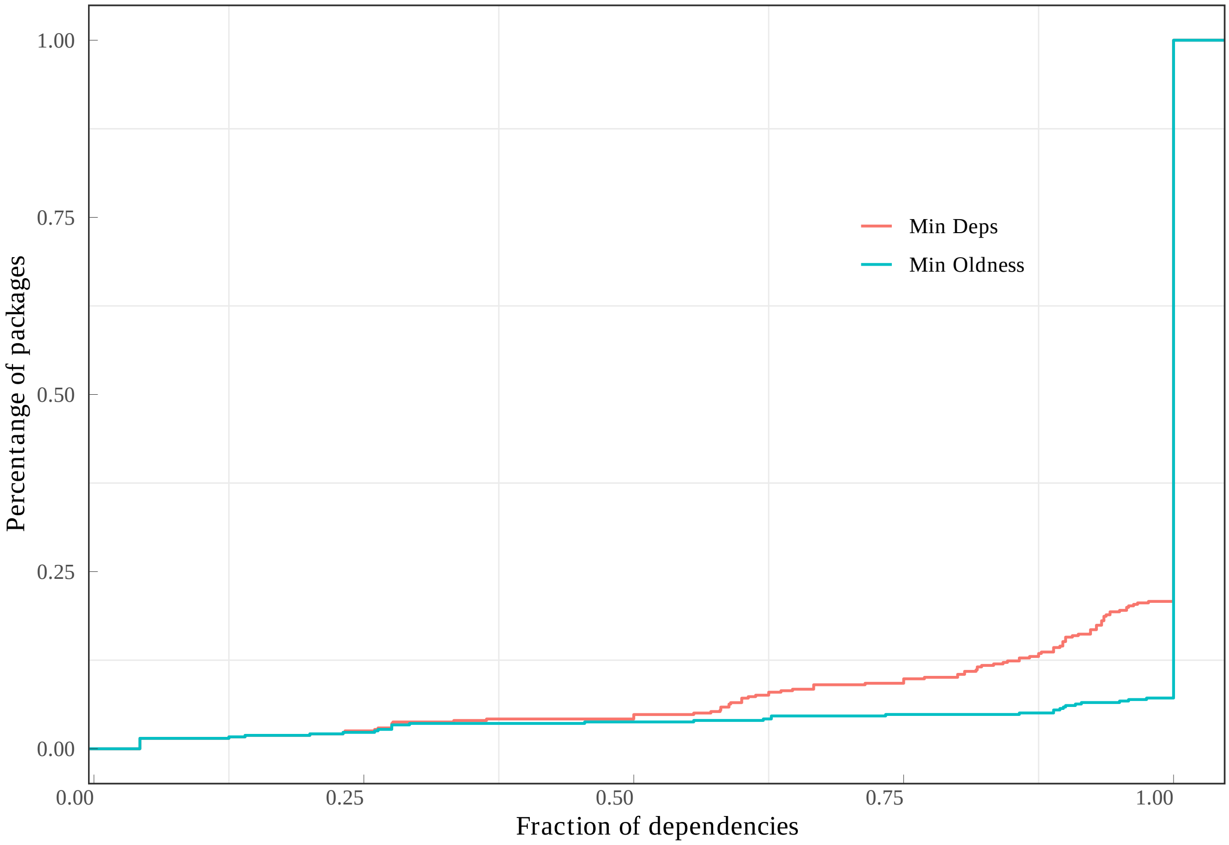
These are potentially bad cases, where MinNPM produces more dependencies than NPM:

```
min_dep_analysis_delta %>% arrange(Delta) %>% filter(Delta < 0)
```

```
## # A tibble: 6 x 3
##   Project       Comparison                          Delta
##   <fct>         <chr>                               <int>
## 1 babel-runtime NPM_NPM_MinDepsOldness_Delta          -23
## 2 babel-runtime NPM_NPM_MinOldness_Delta              -23
## 3 babel-runtime NPM_NPM_MinDuplicatesOldness_Delta    -23
## 4 babel-runtime NPM_PIP_MinOldness_Delta              -23
## 5 babel-runtime NPM_Cargo_MinOldness_Delta            -23
## 6 jsprim        NPM_NPM_MinOldness_Delta               -1
```

WARNING: This filters out the bogus result above.

```
min_dep_analysis_shrinkage %>%
  filter(Shrinkage <= 1.0) %>%
  filter(Comparison == "NPM_NPM_MinDepsOldness_Shrinkage" | Comparison == "NPM_NPM_MinOldness_Shrinkage
  mutate(Comparison = recode(Comparison,
                             NPM_Cargo_MinOldness_Shrinkage="Cargo",
                             NPM_NPM_MinDepsOldness_Shrinkage="Min Deps",
                             NPM_NPM_MinDuplicatesOldness_Shrinkage="MinDuplicates",
                             NPM_NPM_MinOldness_Shrinkage="Min Oldness",
                             NPM_PIP_MinOldness_Shrinkage="PIP vs. NPM")) %>%
  ggplot(aes(Shrinkage, colour=Comparison)) +
  stat_ecdf() +
  ylab("Percentange of packages") +
  xlab("Fraction of dependencies") +
  mytheme()
```



```
mysave("shrinkage.pdf")
```

and a histogram version...

```
# min_dep_analysis_shrinkage %>%
#   filter(Shrinkage <= 1.0) %>%
#   filter(Comparison == 'NPM_NPM_MinDepsOldness_Shrinkage') %>%
#   ggplot(aes(Shrinkage)) +
#   geom_histogram(aes(y=..ndensity..),binwidth=0.1) +
#   ylab("Count of packages") +
#   xlab("Fraction of dependencies") +
#   mytheme()
# mysave("shrinkage_hist.pdf")
```

What fraction of packages can we shrink? This goes in the paper.

```
group_counts <- min_dep_analysis_shrinkage %>% group_by(Comparison) %>% summarize(n = n())

shrink_group_counts <- min_dep_analysis_shrinkage %>% filter(Shrinkage < 1) %>% group_by(Comparison) %>
largen_group_counts <- min_dep_analysis_shrinkage %>% filter(Shrinkage > 1) %>% group_by(Comparison) %>

shrinkage_table <- group_counts %>%
  inner_join(shrink_group_counts) %>%
  inner_join(largen_group_counts) %>%
  mutate(percent_shrunk=100 * n_shrunk / n) %>%
  mutate(percent_larger=100 * n_largen / n) %>%
  mutate(Comparison = recode(Comparison,
                             NPM_Cargo_MinOldness_Shrinkage="Cargo; min_oldness",
                             NPM_PIP_MinOldness_Shrinkage="PIP; min_oldness",
                             NPM_NPM_MinOldness_Shrinkage="NPM; min_oldness,min_num_deps",
                             NPM_NPM_MinDepsOldness_Shrinkage="NPM; min_num_deps,min_oldness",
                             NPM_NPM_MinDuplicatesOldness_Shrinkage="NPM; min_duplicates,min_oldness"))
  arrange(desc(percent_shrunk)) %>%
  rename('# Shrunk (of 477)' = n_shrunk, '# Enlarged (of 477)' = n_largen, Configuration = Comparison) %
  select(Configuration, '# Shrunk (of 477)', '# Enlarged (of 477)')
```

```
## Joining, by = "Comparison"
## Joining, by = "Comparison"
```

```
shrinkage_table
```

```
## # A tibble: 5 x 3
##   Configuration                    `# Shrunk (of 477)` `# Enlarged (of 477)`
##   <chr>                                          <int>                 <int>
## 1 NPM; min_num_deps,min_oldness                     99                     1
## 2 NPM; min_duplicates,min_oldness                   36                     1
## 3 PIP; min_oldness                                  36                     1
## 4 Cargo; min_oldness                                34                     1
## 5 NPM; min_oldness,min_num_deps                     34                     2
```

```
print(xtable(as.data.frame(shrinkage_table), type="latex"), include.rownames=FALSE, file=str_c(tables_di
knitr::kable(shrinkage_table)
```

Configuration	# Shrunk (of 477)	# Enlarged (of 477)
NPM; min_num_deps,min_oldness	99	1
NPM; min_duplicates,min_oldness	36	1
PIP; min_oldness	36	1
Cargo; min_oldness	34	1
NPM; min_oldness,min_num_deps	34	2

```
one_comparison <- min_dep_analysis_shrinkage %>% filter(Comparison == 'NPM_NPM_MinDepsOldness_Shrinkage

fraction_shrinking <- nrow(one_comparison %>% filter(Shrinkage < 1)) / nrow(one_comparison)
write(
  str_c("\\newcommand{\\dataFractionShrinking}{",
        round(fraction_shrinking * 100),
        "\\%}\n"),
  results_tex, append=TRUE)
fraction_shrinking
```

```
## [1] 0.2075472
```

How Old Are Dependencies?

I ran:

$ python3 all_oldness.py /scratch/a.guha/minnpm-exp/vanilla > oldness_vanilla.csv $ python3 all_oldness.py /scratch/a.guha/minnpm-exp/rosette/npm/min_oldness,min_num_deps > oldness_npm_oldness_deps.csv

Raw data

```r
oldness_data <- bind_rows(
  read_csv(paste(oldness_root, "/oldness/vanilla.csv", sep=""),
    col_types = cols(Package=col_factor(),
                     Oldness=col_double()),
    show_col_types = FALSE) %>%
    mutate(Solver = "NPM"),
  read_csv(paste(oldness_root, "/oldness/rosette-npm-allow_cycles-min_oldness-min_num_deps.csv", sep=""
    col_types = cols(Package=col_factor(),
                     Oldness=col_double()),
    show_col_types = FALSE) %>%
    mutate(Solver = "MinOldness"),
  read_csv(paste(oldness_root, "/oldness/rosette-npm-allow_cycles-min_num_deps-min_oldness.csv", sep=""
    col_types = cols(Package=col_factor(),
                     Oldness=col_double()),
    show_col_types = FALSE) %>%
    mutate(Solver = "MinNumDeps")) %>%
  mutate(Project=Package) %>%
  select(Project,Oldness,Solver)
```

```r
oldness_by_pkg <- oldness_data %>%
  pivot_wider(values_from = Oldness, names_from=Solver)

npm_success_non_trivial <- raw_data %>%
  filter(Rosette == FALSE & Status == "success") %>%
  filter(NDeps > 0) %>%
  select(Project)

min_oldenss_success_non_trivial <- raw_data %>%
  filter(Rosette == TRUE &
           Status == "success" &
           Consistency == "npm" &
           DisallowCycles == "allow_cycles" &
           Minimize == "min_oldness,min_num_deps") %>%
  filter(NDeps > 0) %>%
  select(Project)

min_num_deps_success_non_trivial <- raw_data %>%
  filter(Rosette == TRUE &
           Status == "success" &
           Consistency == "npm" &
           DisallowCycles == "allow_cycles" &
           Minimize == "min_num_deps,min_oldness") %>%
  filter(NDeps > 0) %>%
  select(Project)

all_success_non_trivial <- npm_success_non_trivial %>% inner_join(min_oldenss_success_non_trivial) %>%
```

```
## Joining, by = "Project"
## Joining, by = "Project"
```

```r
oldness_by_pkg_success_non_trivial <- oldness_by_pkg %>% inner_join(all_success_non_trivial)
```

```
## Joining, by = "Project"
```

```r
better_oldness <- nrow(oldness_by_pkg_success_non_trivial  %>% filter(MinOldness < NPM)) /
  nrow(oldness_by_pkg_success_non_trivial)
worse_oldness <- nrow(oldness_by_pkg_success_non_trivial  %>% filter(MinOldness > NPM)) /
  nrow(oldness_by_pkg)
write(
  str_c("\\newcommand{\\dataFractionNewer}{",
        round(better_oldness * 100),
        "\\%}\n"),
  results_tex, append=TRUE)
better_oldness
```

```
## [1] 0.1417476
```

```r
write(
  str_c("\\newcommand{\\dataFractionOlder}{",
        round(worse_oldness * 100),
        "\\%}\n"),
  results_tex, append=TRUE)
worse_oldness
```

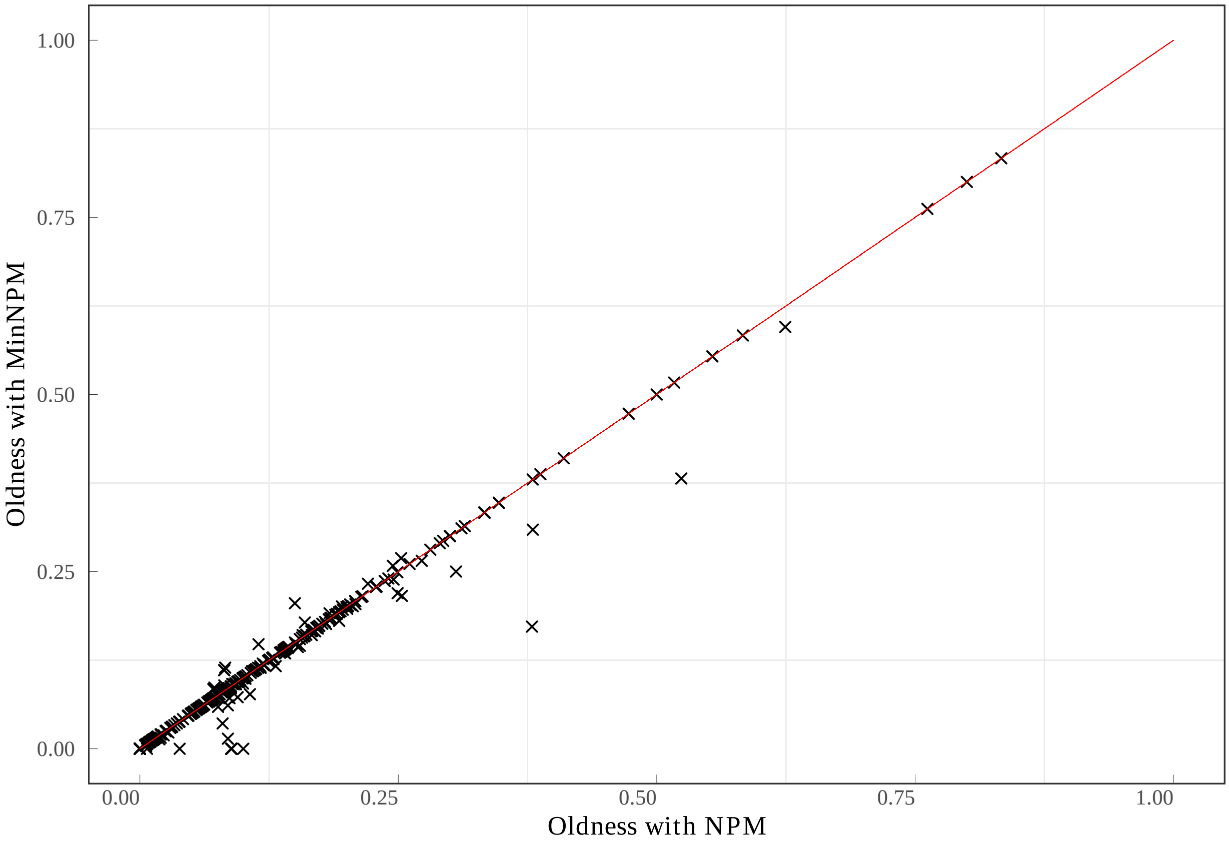
```
## [1] 0.02525253
```

```r
oldness_data %>%
  filter(!is.nan(Oldness)) %>%
  pivot_wider(names_from=Solver, values_from=Oldness) %>%
  select(!MinNumDeps) %>%
  mutate(Delta = NPM - MinOldness) %>%
  mutate(Ratio = MinOldness / NPM) %>%
  filter(Delta > 0)
```

```
## # A tibble: 73 x 5
##    Project                             NPM MinOldness   Delta Ratio
##    <fct>                             <dbl>      <dbl>   <dbl> <dbl>
##  1 @babel_preset-react             0.0800      0.0357 0.0443  0.446
##  2 browserify-sign                 0.0201      0.0156 0.00452 0.775
##  3 @babel_plugin-transform-classes 0.117       0.115  0.00213 0.982
##  4 read-pkg                        0.120       0.118  0.00207 0.983
##  5 eslint-module-utils             0.624       0.595  0.0291  0.953
##  6 postcss-modules-values          0.00685     0      0.00685 0
##  7 @babel_highlight                0.273       0.265  0.00735 0.973
##  8 memory-fs                       0.141       0.135  0.00595 0.958
##  9 class-utils                     0.306       0.250  0.0557  0.818
## 10 nanomatch                       0.253       0.216  0.0376  0.851
## # ... with 63 more rows
```

```r
oldness_data %>%
  filter(!is.nan(Oldness)) %>%
  pivot_wider(names_from=Solver, values_from=Oldness) %>%
  ggplot(aes(x=NPM,y=MinOldness)) +
  geom_point(shape=4, size=1.5) +
  geom_segment(aes(x = 0, y = 0, xend = 1, yend = 1), size=0.02, color="red") +
  xlab("Oldness with NPM") +
  ylab("Oldness with MinNPM") +
  mytheme()
```

```
## Warning: Removed 55 rows containing missing values (geom_point).
```



```
mysave("oldness_scatterplot.pdf")
```

```
## Warning: Removed 55 rows containing missing values (geom_point).
```

Do packages get smaller?

```
vanilla_sizes <- read_tsv("/home/donald/vanilla_sizes.tsv", col_names = c("Size", "Project"), show_col_
min_deps_sizes <- read_tsv("/home/donald/npm_min_num_deps.tsv", col_names = c("Size", "Project"), show_c
min_oldness_sizes <- read_tsv("/home/donald/npm_min_oldness.tsv", col_names = c("Size", "Project"), sho
min_duplicates_sizes <- read_tsv("/home/donald/npm_min_duplicates.tsv", col_names = c("Size", "Project"

ok_projects <- raw_data %>%
  filter(Rosette == FALSE & Status == "success") %>%
  select(Project) %>%
  inner_join(raw_data %>%
            filter(Rosette == TRUE & Status == "success" & Consistency == "npm" &
                   Minimize == "min_num_deps,min_oldness") %>%
            select(Project)) %>%
  inner_join(raw_data %>%
            filter(Rosette == TRUE & Status == "success" & Consistency == "npm" &
                   Minimize == "min_oldness,min_num_deps") %>%
            select(Project)) %>%
  inner_join(raw_data %>%
            filter(Rosette == TRUE & Status == "success" & Consistency == "npm" &
                   Minimize == "min_duplicates,min_oldness") %>%
            select(Project))
```

```
## Joining, by = "Project"
## Joining, by = "Project"
## Joining, by = "Project"
```

```
size_per_project_solver <- ok_projects %>%
  inner_join(vanilla_sizes) %>% rename(NPM = Size) %>%
  inner_join(min_deps_sizes) %>% rename(MinDeps = Size) %>%
  inner_join(min_oldness_sizes) %>% rename(MinOldness = Size) %>%
  inner_join(min_duplicates_sizes) %>% rename(MinDuplicates = Size)
```

```
## Joining, by = "Project"
## Joining, by = "Project"
## Joining, by = "Project"
## Joining, by = "Project"
```

```
size_shrinkage <- size_per_project_solver %>%
  mutate(ShrinkageMinDeps = MinDeps / NPM,
         ShrinkageMinOldness = MinOldness / NPM,
         ShrinkageMinDuplicates = MinDuplicates / NPM)

mean(size_shrinkage$ShrinkageMinDeps)
```

```
## [1] 0.5222152
```

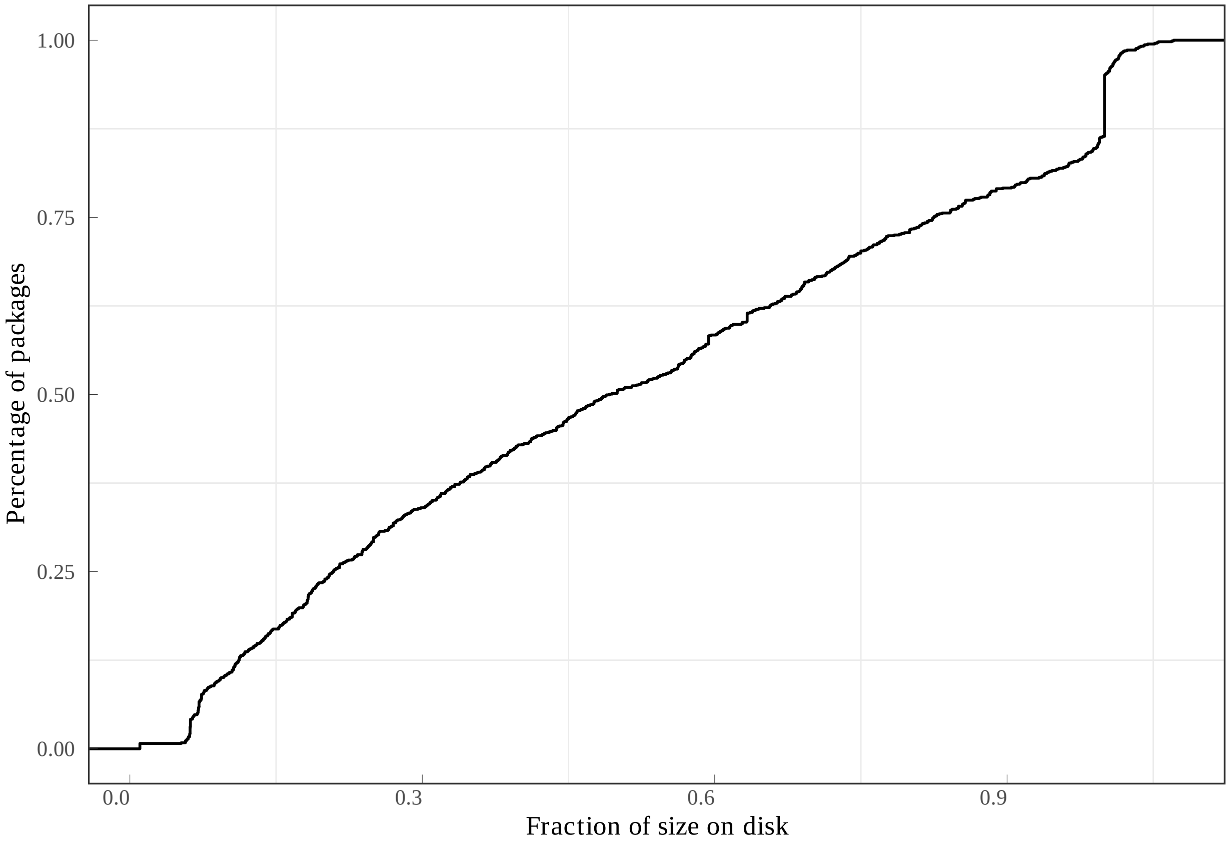
```
mean(size_shrinkage$ShrinkageMinOldness)
```

```
## [1] 0.532614
```

```
mean(size_shrinkage$ShrinkageMinDuplicates)
```

```
## [1] 0.5324975
```

```
size_shrinkage %>%
  select(Project,ShrinkageMinDeps,ShrinkageMinOldness,ShrinkageMinDuplicates) %>%
  pivot_longer(cols = starts_with("Shrinkage"), names_to="Config", values_to="Shrinkage") %>%
  filter(Config=="ShrinkageMinDeps") %>%
  ggplot(aes(x=Shrinkage)) + stat_ecdf() + mytheme() + xlab("Fraction of size on disk") + ylab("Percenta
```



```
mysave("disk_shrinkage_ecdf.pdf")
```

```
# size_shrinkage %>%
#   select(Project,ShrinkageMinDeps,ShrinkageMinOldness,ShrinkageMinDuplicates) %>%
#   pivot_longer(cols = starts_with("Shrinkage"), names_to="Config", values_to="Shrinkage") %>%
#   filter(Config=="ShrinkageMinDeps") %>%
#   ggplot(aes(x=Shrinkage)) + stat_ecdf() + mytheme() + xlim(0, 1.2)
#
# mysave("disk_shrinkage_no_outliers_ecdf.pdf")
```

Performance Analysis

```
slowdowns <- read_csv(paste(perf_root,"/vanilla-perf.csv",sep=""),
         col_names = c("Project", "Time"),
         col_types = cols(Project = col_factor(), Time = col_double()),
         show_col_types = FALSE) %>%
  group_by(Project) %>%
  summarise(NPM = mean(Time)) %>%
  ungroup() %>%
  inner_join(
    read_csv(paste(perf_root,"/rosette-perf.csv",sep=""),
             col_names = c("Project", "Time"),
             col_types = cols(Project = col_factor(), Time = col_double()),
             show_col_types = FALSE) %>%
        group_by(Project) %>%
      summarise(MinNPM = mean(Time)) %>%
      ungroup()) %>%
  mutate(Slowdown = MinNPM - NPM) %>%
  select(Project, Slowdown)
```

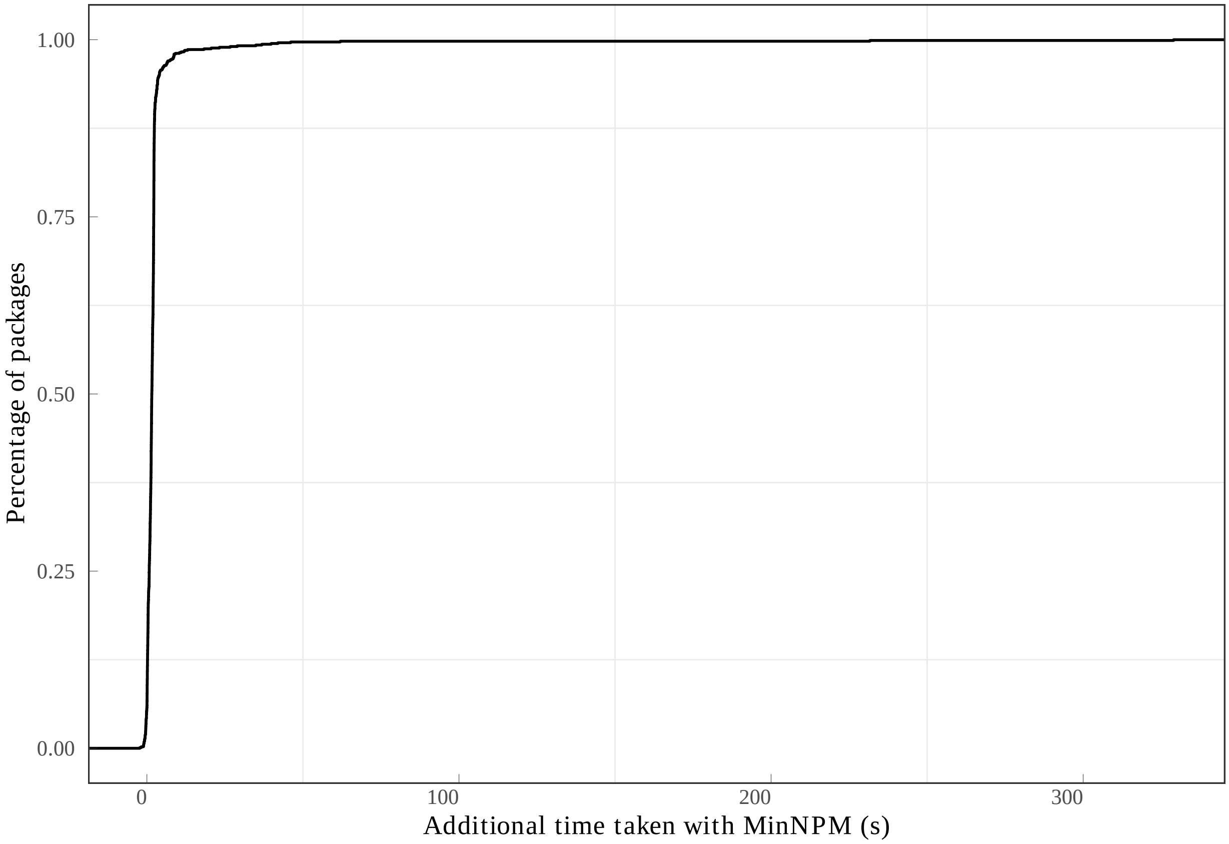
```
## Joining, by = "Project"
```

```
new_slows <- slowdowns %>% filter(Slowdown > 15)
new_slows
```

```
## # A tibble: 13 x 2
##    Project                                Slowdown
##    <fct>                                     <dbl>
##  1 jest-each                                  46.1
##  2 pretty-format                              20.6
##  3 babel-preset-jest                          36.8
##  4 babel-plugin-polyfill-corejs3              39.8
##  5 @babel_plugin-transform-runtime            42.0
##  6 @jest_environment                         329. 
##  7 eslint-plugin-import                       18.3
##  8 jest-changed-files                         23.2
##  9 @babel_helper-define-polyfill-provider     34.9
## 10 jest-resolve                              232. 
## 11 jest-message-util                          61.9
## 12 mississippi                                28.9
## 13 babel-plugin-jest-hoist                    26.7
```

```
slowdowns %>% ggplot(aes(x=Slowdown)) +
  stat_ecdf() +
  xlab("Additional time taken with MinNPM (s)") +
  ylab("Percentage of packages") +
  mytheme()
```

```
## Warning: Removed 1 rows containing non-finite values (stat_ecdf).
```

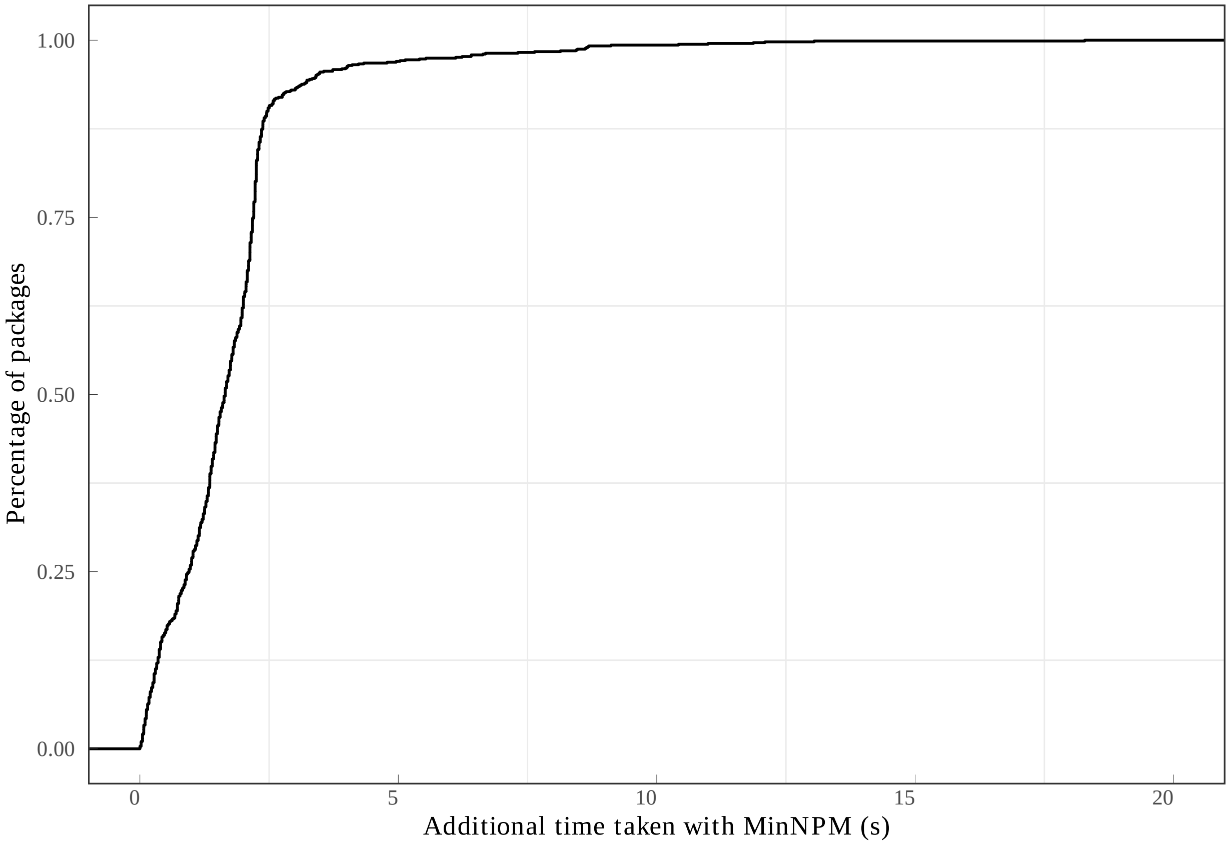


```
mysave("slowdown_ecdf.pdf")
```

```
## Warning: Removed 1 rows containing non-finite values (stat_ecdf).
```

```
slowdowns %>% ggplot(aes(x=Slowdown)) +
  stat_ecdf() +
  xlab("Additional time taken with MinNPM (s)") +
  ylab("Percentage of packages") +
  mytheme() + xlim(0, 20)
```

```
## Warning: Removed 67 rows containing non-finite values (stat_ecdf).
```



```
mysave("slowdown_ecdf_no_outliers.pdf")
```

```
## Warning: Removed 67 rows containing non-finite values (stat_ecdf).
```

Reported in paper:

```
mean_slowdown <- round(mean(na.omit(slowdowns$Slowdown)), digits = 1)
median_slowdown <- round(median(na.omit(slowdowns$Slowdown)), digits = 1)
max_slowdown <- round(max(na.omit(slowdowns$Slowdown)), digits = 1)

write(
  str_c("\\newcommand{\\dataMeanSlowdown}{",
        mean_slowdown,
        "s}\n"),
  results_tex, append=TRUE)
write(
  str_c("\\newcommand{\\dataMedianSlowdown}{",
        median_slowdown,
        "s}\n"),
  results_tex, append=TRUE)
write(
  str_c("\\newcommand{\\dataMaxSlowdown}{",
        max_slowdown,
        "s}\n"),
  results_tex, append=TRUE)

mean_slowdown
```

```
## [1] 2.6
```

```
median_slowdown
```

```
## [1] 1.6
```

```
max_slowdown
```

```
## [1] 329
```