EDUCE descriptive publication data figures

Kim Dill-McFarland

version July 11, 2019

Setup

Load packages

```
#Data manipulation and figures
library(tidyverse)
#Multi-panel figures
library(cowplot)
#Exact and Monte Carlo symmetry tests for paired contigency tables
library(rcompanion)
```

R session

```
sessionInfo()
```

```
## R version 3.5.3 (2019-03-11)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.5
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_CA.UTF-8/en_CA.UTF-8/en_CA.UTF-8/C/en_CA.UTF-8/en_CA.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] rcompanion_2.2.1 cowplot_0.9.4    forcats_0.4.0    stringr_1.4.0
##  [5] dplyr_0.8.3      purrr_0.3.2      readr_1.3.1      tidyr_0.8.3
##  [9] tibble_2.1.3     ggplot2_3.2.0    tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.1         lubridate_1.7.4    mvtnorm_1.0-11
##  [4] lattice_0.20-38    multcompView_0.1-7 zoo_1.8-6
##  [7] lmtest_0.9-37      assertthat_0.2.1   zeallot_0.1.0
## [10] digest_0.6.20      plyr_1.8.4         R6_2.4.0
## [13] cellranger_1.1.0   backports_1.1.4    EMT_1.1
## [16] stats4_3.5.3       evaluate_0.14      httr_1.4.0
## [19] pillar_1.4.2       rlang_0.4.0        lazyeval_0.2.2
## [22] multcomp_1.4-10    readxl_1.3.1       rstudioapi_0.10
## [25] Matrix_1.2-17      rmarkdown_1.13     splines_3.5.3
## [28] foreign_0.8-71     munsell_0.5.0      broom_0.5.2
## [31] compiler_3.5.3     modelr_0.1.4       xfun_0.8
## [34] pkgconfig_2.0.2    manipulate_1.0.1   libcoin_1.0-4
## [37] DescTools_0.99.28  htmltools_0.3.6    tidyselect_0.2.5
## [40] expm_0.999-4       coin_1.3-0         codetools_0.2-16
## [43] matrixStats_0.54.0 crayon_1.3.4       withr_2.1.2
## [46] MASS_7.3-51.4      grid_3.5.3         nlme_3.1-140
## [49] jsonlite_1.6       gtable_0.3.0       magrittr_1.5
## [52] scales_1.0.0       cli_1.1.0          stringi_1.4.3
## [55] xml2_1.2.0         vctrs_0.2.0        generics_0.0.2
## [58] nortest_1.0-4      sandwich_2.5-1     boot_1.3-23
## [61] TH.data_1.0-10     tools_3.5.3        glue_1.3.1
## [64] hms_0.5.0          parallel_3.5.3     survival_2.44-1.1
## [67] yaml_2.2.0         colorspace_1.4-1   rvest_0.3.4
## [70] knitr_1.23         haven_2.1.1        modeltools_0.2-22
```

Community of practice

Load data

```
cop <- read_tsv("data_clean/2017.18.19_EDUCEteam.txt")
```

```
## Parsed with column specification:
## cols(
##   dept = col_character(),
##   fac = col_character(),
##   Undergraduate = col_double(),
##   Graduate = col_double(),
##   Postdoc = col_double(),
##   Research = col_double(),
##   Instructor = col_double(),
##   Staff = col_double()
## )
```

```
cop
```

```
## # A tibble: 10 x 8
##    dept    fac     Undergraduate Graduate Postdoc Research Instructor Staff
##    <chr>   <chr>           <dbl>    <dbl>   <dbl>    <dbl>      <dbl> <dbl>
##  1 MICB    SCIE               NA        2       1        5          3     2
##  2 MGEN    MED                NA        2      NA       NA         NA    NA
##  3 IAM     ASCI               NA        1      NA       NA         NA    NA
##  4 BOTA    SCIE               NA        1       1       NA         NA    NA
##  5 STAT    SCIE                2       NA      NA        1          1     1
##  6 CPSC    SCIE                2       NA      NA       NA         NA    NA
##  7 ECE     ASCI               NA        1      NA       NA         NA    NA
##  8 MATH    SCIE               NA       NA      NA       NA          1    NA
##  9 Central Central            NA       NA      NA       NA         NA     3
## 10 LFS     LFS                NA        1      NA       NA         NA    NA
```

Calculate totals by career level (*e.g.* student, faculty, etc.) in each Department

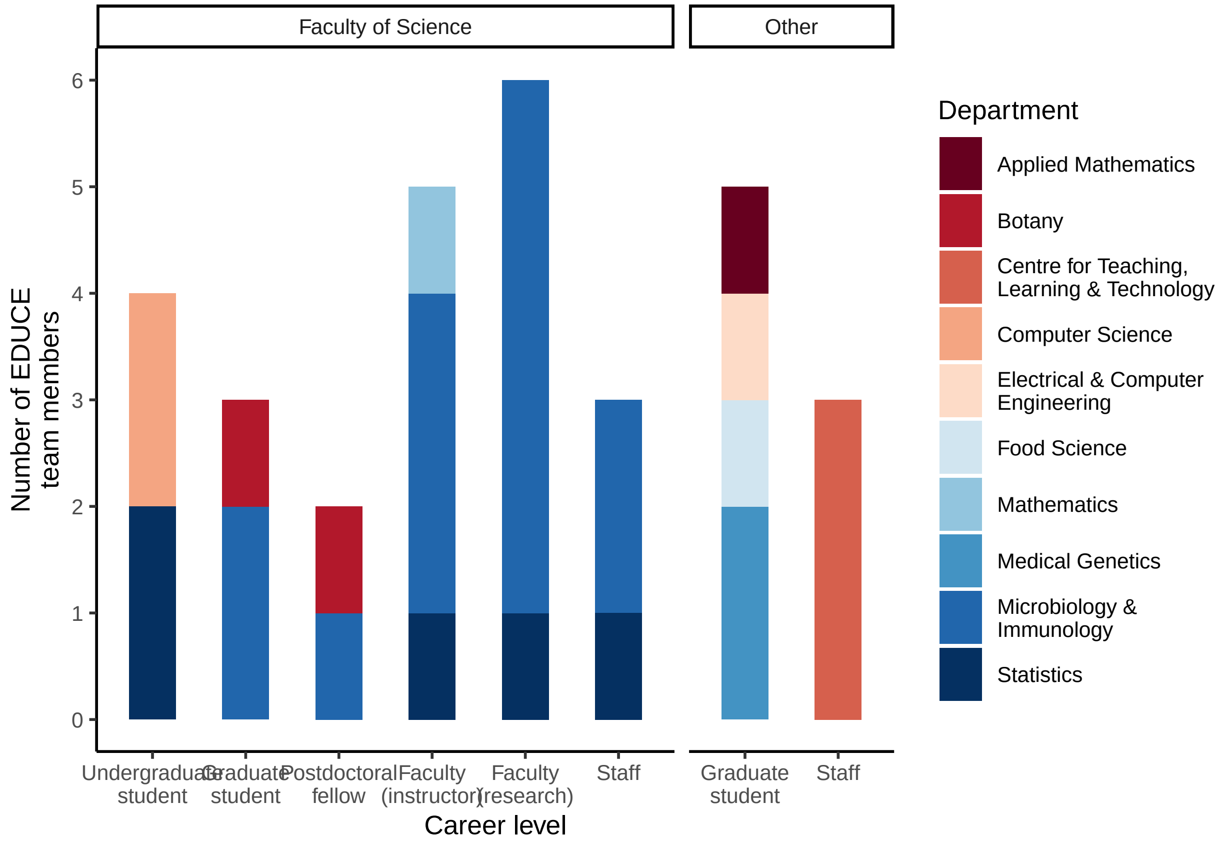
```
cop_sum <- cop %>%
  #Gather career levels into 1 column
  gather(key="key", value="total", -dept, -fac) %>%
  #Remove NAs
  filter(!is.na(total)) %>%
  #Sum totals of each career level in each department
  group_by(key, dept, fac) %>%
  summarize(n=sum(total)) %>%
  #Reorder variables for plot
  mutate(key_ord = factor(key,
                         levels=c("Undergraduate","Graduate",
                                  "Postdoc","Instructor","Research",
                                  "Staff"))) %>%
  mutate(dept_ord = factor(dept,
                   levels=c("IAM","BOTA","Central","CPSC","ECE","LFS","MATH","MGEN","MICB","STAT"))) %
  mutate(fac_ord = factor(fac, levels=c("ASCI","LFS","MED","SCIE","Central")))
```

Figure 3. EDUCE team members at UBC

```
cop_plot <- cop_sum %>%
  #Create variable for Science vs Other facets
  mutate(fac_group = ifelse(fac == "SCIE", fac, "Other")) %>%
  #Reorder variable for facets
  mutate(fac_group_ord = factor(fac_group, levels=c("SCIE","Other"))) %>%

#Plot
  ggplot() +
  geom_bar(aes(x=key_ord, y=n, fill=dept_ord), stat="identity", width = 0.5) +
  facet_grid(~fac_group_ord, scales = "free_x", space="free",
             labeller = as_labeller(c("SCIE"="Faculty of Science",
                                     "Other"="Other"))) +
  #Beautify
  theme_classic() +
  labs(x="Career level", y="Number of EDUCE\nteam members") +
  theme(legend.key.height = unit(0.75, "cm"),
        text=element_text(colour="black", size=10)) +
  scale_x_discrete(labels=c("Staff" = "Staff",
                            "Research"="Faculty\n(research)",
                            "Instructor" = "Faculty\n(instructor)",
                            "Postdoc" = "Postdoctoral\nfellow",
                            "Graduate" = "Graduate\nstudent",
                            "Undergraduate" = "Undergraduate\nstudent")) +
  scale_fill_brewer(name = "Department", labels = c("Applied Mathematics",
                                                    "Botany",
                                                    "Centre for Teaching,\nLearning & Technology",
                                                    "Computer Science",
                                                    "Electrical & Computer\nEngineering",
                                                    "Food Science",
                                                    "Mathematics",
                                                    "Medical Genetics",
                                                    "Microbiology &\nImmunology",
                                                    "Statistics"),
                    palette = "RdBu") +
  scale_y_continuous(breaks=c(0:6))

cop_plot
```



Save Figure 3

```
ggsave(filename="Fig3.pdf", plot=cop_plot, width=19.05, height=9, units = "cm")
```

Student interest and experience

Load data

```
survey <- read_csv("data_clean/2017.18.19_survey_clean.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   Prev_unix = col_double(),
##   Prev_R = col_double(),
##   Prev_mothur = col_double(),
##   Prev_QIIME = col_double(),
##   Prev_metaG = col_double(),
##   Prev_EDUCE_MICB301 = col_double(),
##   Prev_EDUCE_MICB405 = col_double(),
##   Prev_EDUCE_MICB421 = col_double(),
##   Prev_EDUCE_MICB425 = col_double(),
##   Post_Interest_introR = col_double(),
##   Post_Interest_data = col_double(),
##   Post_Interest_ggplot = col_double(),
##   Post_Interest_statmodel = col_double(),
##   Post_Interest_repro = col_double(),
##   Post_Interest_microbiomeR = col_double(),
##   Post_Interest_mothur_QIIME = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

Information on survey data clean-up can be found in `EDUCE_survey_cleanup`

Survey question: How would you rate your interest in…

Monte Carlo symmetry tests for paired contingency tables

Pre responses in columns, post responses in rows

Data cleaning

```
interest <- survey %>%
  # Select variables of interest
  select(Course, year,
         Pre_Interest_BI, Post_Interest_BI,
         Pre_Interest_CPSC, Post_Interest_CPSC,
         Pre_Interest_STAT, Post_Interest_STAT) %>%
  # Filter to just MICB301 course data
  filter(Course == "MICB301") %>%
  # Convert numeric survey respones to groups
  ## None=0, low=1-3, med=4-7, high=8-10
  ### Create row ID to keep matched responses together
  rowid_to_column() %>%
  gather(key="key", value="value",
         -Course, -year, -rowid) %>%
  mutate(value = ifelse(value %in% c("0",0), "None",
                   ifelse(value %in% c("1","2","3"), "Low",
                   ifelse(value %in% c("4","5","6","7"), "Medium",
                   ifelse(value %in% c("8","9","10"), "High",
                     value))))) %>%
  group_by(rowid) %>%
  spread(key=key, value=value) %>%
  ungroup()
```

Bioinformatics

Test for differences in pre- vs. post- matched surveys.

```
BI_interest <- interest %>%
  # Select variables of interest
  select(Pre_Interest_BI, Post_Interest_BI) %>%
  drop_na() %>%
  # Count matched pre-post response
  group_by(Pre_Interest_BI, Post_Interest_BI) %>%
  summarize(n=n()) %>%
  # Format into contingency table
  spread(Pre_Interest_BI, n) %>%
  replace(is.na(.), 0) %>%
  # add Pre=none
  mutate(None=c(0,0,0,0)) %>%
  #Order variables
  select(Post_Interest_BI, None, Low, Medium, High) %>%
  arrange(factor(Post_Interest_BI, levels = c("None", "Low", "Medium", "High"))) %>%
  column_to_rownames(var="Post_Interest_BI")

BI_interest
```

```
##        None Low Medium High
## None      0   1      0    0
## Low       0  12      5    0
## Medium    0  12     54    8
## High      0   2     25   24
```

```
nominalSymmetryTest(BI_interest, digits=5, method="fdr", MonteCarlo=TRUE, ntrial=10000)
```

```
##
##  WARNING: Number of simulated withdrawels is lower than the number of possible outcomes.
##                This might yield unreliable results!
##
##
##  Monte Carlo Multinomial Test, distance measure: f
##
##        Events    fObs    p.value
##  743595781824       0      1e-04
```

```
## $Global.test.for.symmetry
##   Dimensions p.value
## 1      4 x 4   1e-04
##
## $Pairwise.symmetry.tests
##                  Comparison   p.value p.adjust
## 1       None/None : Low/Low         1 1.000000
## 2 None/None : Medium/Medium      <NA>       NA
## 3     None/None : High/High      <NA>       NA
## 4   Low/Low : Medium/Medium   0.14346 0.286920
## 5       Low/Low : High/High       0.5 0.666670
## 6 Medium/Medium : High/High 0.0045514 0.018206
##
## $p.adjustment
##   Method
## 1    fdr
```

Computer science

Test for differences in pre- vs. post- matched surveys.

```
CPSC_interest <- interest %>%
  # Select variables of interest
  select(Pre_Interest_CPSC, Post_Interest_CPSC) %>%
  drop_na() %>%
  # Count matched pre-post response
  group_by(Pre_Interest_CPSC, Post_Interest_CPSC) %>%
  summarize(n=n()) %>%
  # Format into contingency table
  spread(Pre_Interest_CPSC, n) %>%
  replace(is.na(.), 0) %>%
  #Order variables
  select(Post_Interest_CPSC, None, Low, Medium, High) %>%
  arrange(factor(Post_Interest_CPSC, levels = c("None", "Low", "Medium", "High"))) %>%
  column_to_rownames(var="Post_Interest_CPSC")

CPSC_interest
```

```
##        None Low Medium High
## None      3   0      0    0
## Low       2  16      9    0
## Medium    1  14     43   10
## High      0   5     18   25
```

```
nominalSymmetryTest(CPSC_interest, digits=5, method="fdr", MonteCarlo=TRUE, ntrial=10000)
```

```
##
##  WARNING: Number of simulated withdrawels is lower than the number of possible outcomes.
##                  This might yield unreliable results!
##
##
##  Monte Carlo Multinomial Test, distance measure: f
##
##        Events    fObs    p.value
##  2.163843e+12       0      1e-04

## $Global.test.for.symmetry
##   Dimensions p.value
## 1      4 x 4   1e-04
##
## $Pairwise.symmetry.tests
##                   Comparison p.value p.adjust
## 1        None/None : Low/Low     0.5  0.62500
## 2 None/None : Medium/Medium       1  1.00000
## 3      None/None : High/High    <NA>       NA
## 4   Low/Low : Medium/Medium 0.40487  0.62500
## 5       Low/Low : High/High  0.0625  0.31250
## 6 Medium/Medium : High/High 0.18493  0.46233
##
## $p.adjustment
##   Method
## 1    fdr
```

Statistics

Test for differences in pre- vs. post- matched surveys.

```
STAT_interest <- interest %>%
  # Select variables of interest
  select(Pre_Interest_STAT, Post_Interest_STAT) %>%
  drop_na() %>%
  # Count matched pre-post response
  group_by(Pre_Interest_STAT, Post_Interest_STAT) %>%
  summarize(n=n()) %>%
  # Format into contingency table
  spread(Pre_Interest_STAT, n) %>%
  replace(is.na(.), 0) %>%
  # Add Pre=None
  mutate(None=c(0,0,0,0)) %>%
  #Order variables
  select(Post_Interest_STAT, None, Low, Medium, High) %>%
  arrange(factor(Post_Interest_STAT, levels = c("None", "Low", "Medium", "High"))) %>%
  column_to_rownames(var="Post_Interest_STAT")

STAT_interest
```

```
##        None Low Medium High
## None      0   1      1    0
## Low       0  31     13    0
## Medium    0  17     55    6
## High      0   0     11    8
```

```
nominalSymmetryTest(STAT_interest, digits=5, method="fdr", MonteCarlo=TRUE, ntrial=10000)
```

```
##
##  WARNING: Number of simulated withdrawels is lower than the number of possible outcomes.
##                This might yield unreliable results!
##
##
##  Monte Carlo Multinomial Test, distance measure: f
##
##        Events    fObs    p.value
##  342700125300       0      1e-04

## $Global.test.for.symmetry
##   Dimensions p.value
## 1      4 x 4   1e-04
##
## $Pairwise.symmetry.tests
##                   Comparison p.value p.adjust
## 1       None/None : Low/Low       1        1
## 2 None/None : Medium/Medium       1        1
## 3     None/None : High/High    <NA>       NA
## 4   Low/Low : Medium/Medium 0.58466        1
## 5       Low/Low : High/High    <NA>       NA
## 6 Medium/Medium : High/High 0.33231        1
##
## $p.adjustment
##   Method
## 1    fdr
```

Interest plot

Data cleaning

```
plot_I_dat <- interest %>%
  #Gather pre/post data
  gather("subject", "interest", -Course, -year, -rowid) %>%
  drop_na(interest) %>%
  # Create separate pre/post ID column
  separate(subject, into=c("survey","trash","subject"), sep="_") %>%
  #Reorder groups
  mutate(survey =  factor(survey, levels = c("Pre", "Post")),
         interest = factor(interest, levels=c("High","Medium","Low","None"))) %>%
  #Remove trash column containing just "Interest" part of name
  select(-trash) %>%
  # Calculate percentages of responses
  group_by(Course, survey, subject, interest) %>%
  summarize(n=n()) %>%
  mutate(freq=100*n/sum(n))
```

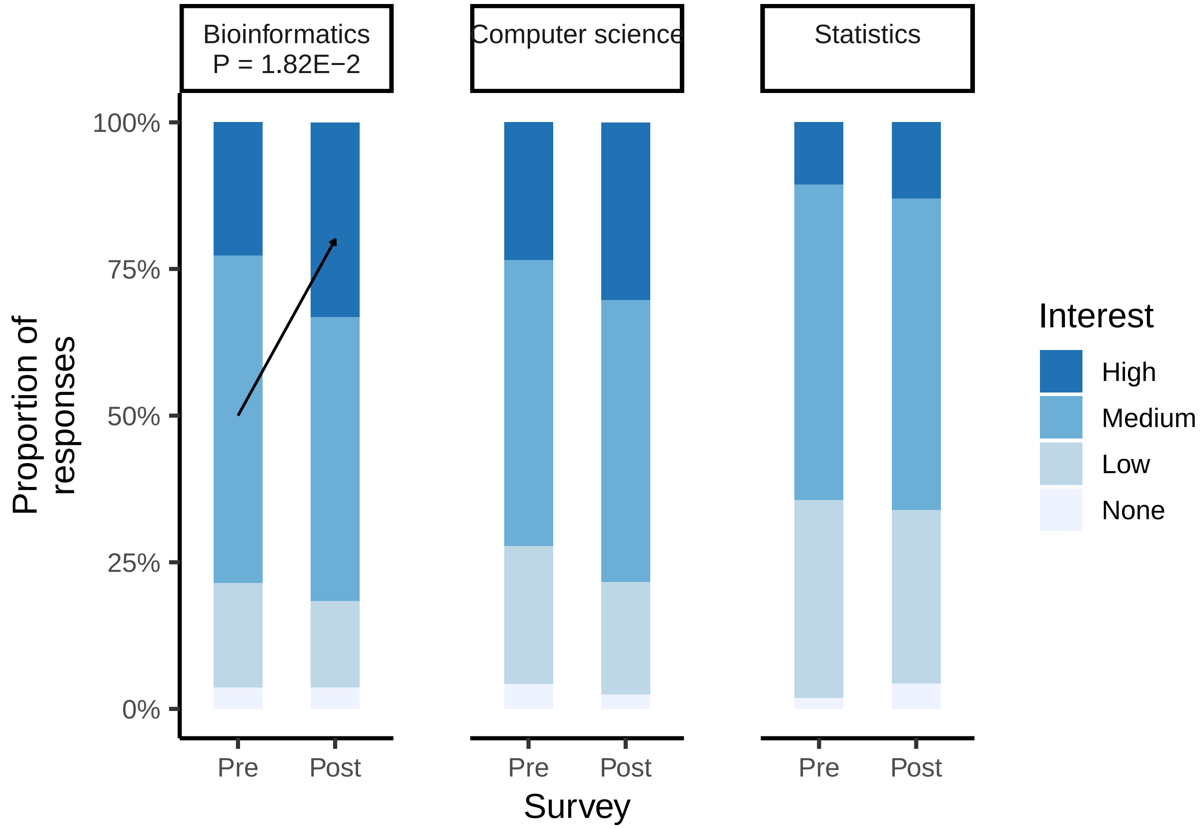
Bar plot

```
plot_I <- ggplot(plot_I_dat,
                 aes(x=survey, y=freq, fill=interest)) +
  geom_col(position = "fill", width=0.5) +
  #Beautify
  labs(x="Survey", y="Proportion of\nresponses", fill="") +
  facet_grid(~subject, labeller = as_labeller(c("BI"="Bioinformatics\nP = 1.82E-2",
                                                "CPSC"="Computer science\n",
                                                "STAT"="Statistics\n"))) +
  theme_classic(base_size = 16) +
  theme(text = element_text(size=13),
        panel.spacing = unit(2, "lines")) +
  scale_x_discrete(labels=c("Pre","Post")) +
  scale_fill_brewer(palette = "Blues", direction=-1,
                    name="Interest") +
  scale_y_continuous(labels=scales::percent)

## Add significant arrows
arrow_I_bi<-data.frame(
  x = 1, y = 0.5, xend = 2, yend = 0.8,
  subject=factor("BI", levels=c("BI","CPSC","STAT")))

plot_I <- plot_I + geom_segment(data=arrow_I_bi, aes(x=x, y=y, xend=xend, yend=yend),
               arrow = arrow(length = unit(0.03, "npc")),
               inherit.aes = FALSE)
```

```
plot_I
```



Survey question: What level of experience do you have in . . .

Monte Carlo symmetry tests for paired contingency tables

Pre responses in columns, post responses in rows

Data cleaning

```
exp <- survey %>%
  # Select variables of interest
  select(Course, year,
         Pre_Exp_BI, Post_Exp_BI,
         Pre_Exp_CPSC, Post_Exp_CPSC,
         Pre_Exp_STAT, Post_Exp_STAT) %>%
  # Filter to just MICB301 course data
  filter(Course == "MICB301") %>%
  # Convert numeric survey respones to groups
  ## None=0, low=1-3, med=4-7, high=8-10
  ### Create row ID to keep matched responses together
  rowid_to_column() %>%
  gather(key="key", value="value",
         -Course, -year, -rowid) %>%
  mutate(value = ifelse(value == "0", "None",
                        ifelse(value %in% c("1","2","3"), "Low",
                        ifelse(value %in% c("4","5","6","7"), "Medium",
                        ifelse(value %in% c("8","9","10"), "High",
                     value))))) %>%
  #  Convert 1 "very high" response to "high"
  mutate(value = ifelse(value=="veryHigh","High",value)) %>%
  #Spread back to wide format
  group_by(rowid) %>%
  spread(key=key, value=value) %>%
  ungroup()
```

Bioinformatics

Test for differences in pre- vs. post- matched surveys.

```
BI_exp <- exp %>%
  # Select variables of exp
  select(Pre_Exp_BI, Post_Exp_BI) %>%
  drop_na() %>%
  # Count matched pre-post response
  group_by(Pre_Exp_BI, Post_Exp_BI) %>%
  summarize(n=n()) %>%
  # Format into contingency table
  spread(Pre_Exp_BI, n) %>%
  replace(is.na(.), 0) %>%
  # Add data for Pre = High since none exist
  mutate(High = c(0,0,0,0)) %>%
  #Order variables
  select(Post_Exp_BI, None, Low, Medium, High) %>%
  arrange(factor(Post_Exp_BI, levels = c("None", "Low", "Medium", "High"))) %>%
  column_to_rownames(var="Post_Exp_BI")

BI_exp
```

```
##        None Low Medium High
## None      4   2      1    0
## Low      33  32      8    0
## Medium   21  22     11    0
## High      0   1      2    0
```

```
nominalSymmetryTest(BI_exp, digits=5, method="fdr", MonteCarlo=TRUE, ntrial=10000)
```

```
##
##  WARNING: Number of simulated withdrawels is lower than the number of possible outcomes.
##                 This might yield unreliable results!
##
##
##  Monte Carlo Multinomial Test, distance measure: f
##
##        Events    fObs    p.value
##  1.589401e+14       0      1e-04
```

```
## $Global.test.for.symmetry
##   Dimensions p.value
## 1      4 x 4   1e-04
##
## $Pairwise.symmetry.tests
##                   Comparison    p.value   p.adjust
## 1       None/None : Low/Low 3.6729e-08 1.8364e-07
## 2 None/None : Medium/Medium 1.0967e-05 2.7418e-05
## 3     None/None : High/High       <NA>         NA
## 4   Low/Low : Medium/Medium   0.016125 2.6875e-02
## 5       Low/Low : High/High          1 1.0000e+00
## 6 Medium/Medium : High/High        0.5 6.2500e-01
##
## $p.adjustment
##   Method
## 1    fdr
```

Computer science

Test for differences in pre- vs. post- matched surveys.

```
CPSC_exp <- exp %>%
  # Select variables of interest
  select(Pre_Exp_CPSC, Post_Exp_CPSC) %>%
  drop_na() %>%
  # Count matched pre-post response
  group_by(Pre_Exp_CPSC, Post_Exp_CPSC) %>%
  summarize(n=n()) %>%
  # Format into contingency table
  spread(Pre_Exp_CPSC, n) %>%
  replace(is.na(.), 0) %>%
  #Order variables
  select(Post_Exp_CPSC, None, Low, Medium, High) %>%
  arrange(factor(Post_Exp_CPSC, levels = c("None", "Low", "Medium", "High"))) %>%
  column_to_rownames(var="Post_Exp_CPSC")

CPSC_exp
```

```
##        None Low Medium High
## None     21   3      0    0
## Low      20  26      3    0
## Medium    5  13     32    3
## High      0   1      3    7
```

```
nominalSymmetryTest(CPSC_exp, digits=5, method="fdr", MonteCarlo=TRUE, ntrial=10000)
```

```
##
##  WARNING: Number of simulated withdrawels is lower than the number of possible outcomes.
##                 This might yield unreliable results!
##
##
##  Monte Carlo Multinomial Test, distance measure: f
##
##        Events    fObs    p.value
##  508271323092       0      1e-04
```

```
## $Global.test.for.symmetry
##   Dimensions p.value
## 1      4 x 4   1e-04
##
## $Pairwise.symmetry.tests
##                 Comparison    p.value  p.adjust
## 1       None/None : Low/Low 0.00048828 0.0024414
## 2 None/None : Medium/Medium     0.0625 0.1041700
## 3     None/None : High/High       <NA>        NA
## 4   Low/Low : Medium/Medium   0.021271 0.0531780
## 5       Low/Low : High/High          1 1.0000000
## 6 Medium/Medium : High/High          1 1.0000000
##
## $p.adjustment
##   Method
## 1    fdr
```

Statistics

Test for differences in pre- vs. post- matched surveys.

```
STAT_exp <- exp %>%
  # Select variables of interest
  select(Pre_Exp_STAT, Post_Exp_STAT) %>%
  drop_na() %>%
  # Count matched pre-post response
  group_by(Pre_Exp_STAT, Post_Exp_STAT) %>%
  summarize(n=n()) %>%
  # Format into contingency table
  spread(Pre_Exp_STAT, n) %>%
  replace(is.na(.), 0) %>%
  # Add Pre = High data
  mutate(High = c(0,0,0,0)) %>%
  #Order variables
  select(Post_Exp_STAT, None, Low, Medium, High) %>%
  arrange(factor(Post_Exp_STAT, levels = c("None", "Low", "Medium", "High"))) %>%
  column_to_rownames(var="Post_Exp_STAT")

STAT_exp
```

```
##        None Low Medium High
## None      2   4      1    0
## Low       2  12      8    0
## Medium    2  10     99    0
## High      0   1      2    0
```

```
nominalSymmetryTest(STAT_exp, digits=5, method="fdr", MonteCarlo=TRUE, ntrial=10000)
```

```
##
##  WARNING: Number of simulated withdrawels is lower than the number of possible outcomes.
##                 This might yield unreliable results!
##
##
##  Monte Carlo Multinomial Test, distance measure: f
##
##      Events    fObs    p.value
##  3159461968       0      1e-04
```

```
## $Global.test.for.symmetry
##   Dimensions p.value
## 1      4 x 4   1e-04
##
```

```
## $Pairwise.symmetry.tests
##                  Comparison p.value p.adjust
## 1        None/None : Low/Low  0.6875        1
## 2 None/None : Medium/Medium       1        1
## 3      None/None : High/High    <NA>       NA
## 4   Low/Low : Medium/Medium 0.81453        1
## 5        Low/Low : High/High       1        1
## 6 Medium/Medium : High/High     0.5        1
##
## $p.adjustment
##   Method
## 1    fdr
```

Experience plot

Data cleaning

```
plot_E_dat <- exp %>%
  #Gather pre/post data
  gather("subject", "exp", -Course, -year, -rowid) %>%
  drop_na() %>%
  # Create separate pre/post ID column
  separate(subject, into=c("survey","trash","subject"), sep="_") %>%
  #Reorder groups
  mutate(survey =  factor(survey, levels = c("Pre", "Post")),
         exp = factor(exp, levels=c("veryHigh","High","Medium","Low","None"))) %>%
  #Remove trash column containing just "Interest" part of name
  select(-trash) %>%
  # Calculate percentages of responses
  group_by(Course, survey, subject, exp) %>%
  summarize(n=n()) %>%
  mutate(freq=100*n/sum(n))
```

Bar plot

```
plot_E <- ggplot(plot_E_dat, aes(x=survey, y=freq)) +
  geom_col(aes(fill=exp), position = "fill", width=0.5) +
  #Beautify
  labs(x="Survey", y="Proportion of\nresponses", fill="") +
  facet_grid(~subject, labeller = as_labeller(
    c("BI"="Bioinformatics\nP < 0.03",
      "CPSC"="Computer science\nP = 2.44E-3",
      "STAT"="Statistics\n"))) +
  theme_classic(base_size = 16) +
  theme(text = element_text(size=13),
        panel.spacing = unit(2, "lines")) +
  scale_x_discrete(labels=c("Pre","Post")) +
  scale_fill_brewer(palette = "Blues", direction=-1,
                    name="Experience") +
  scale_y_continuous(labels=scales::percent)

#Add arrows
arrow_bi<-data.frame(
  x=1,xend=2, y1=0.6,yend1=0.85, y2=0.2,yend2=0.35, y3=0.2,yend3=0.8,
  subject=factor("BI", levels=c("BI","CPSC","MICB")))
```

```
arrow_cpsc<-data.frame(
  x=1, xend=2, y=0.2,yend=0.4,
  subject=factor("CPSC", levels=c("BI","CPSC","MICB")))

plot_E <- plot_E +
  geom_segment(data=arrow_bi, aes(x=x, y=y1, xend=xend, yend=yend1),
               arrow = arrow(length = unit(0.03, "npc"))) +
  geom_segment(data=arrow_bi, aes(x=x, y=y2, xend=xend, yend=yend2),
               arrow = arrow(length = unit(0.03, "npc"))) +
  geom_segment(data=arrow_bi, aes(x=x, y=y3, xend=xend, yend=yend3),
               arrow = arrow(length = unit(0.03, "npc"))) +

  geom_segment(data=arrow_cpsc, aes(x=x, y=y, xend=xend, yend=yend),
               arrow = arrow(length = unit(0.03, "npc")))

plot_E
```

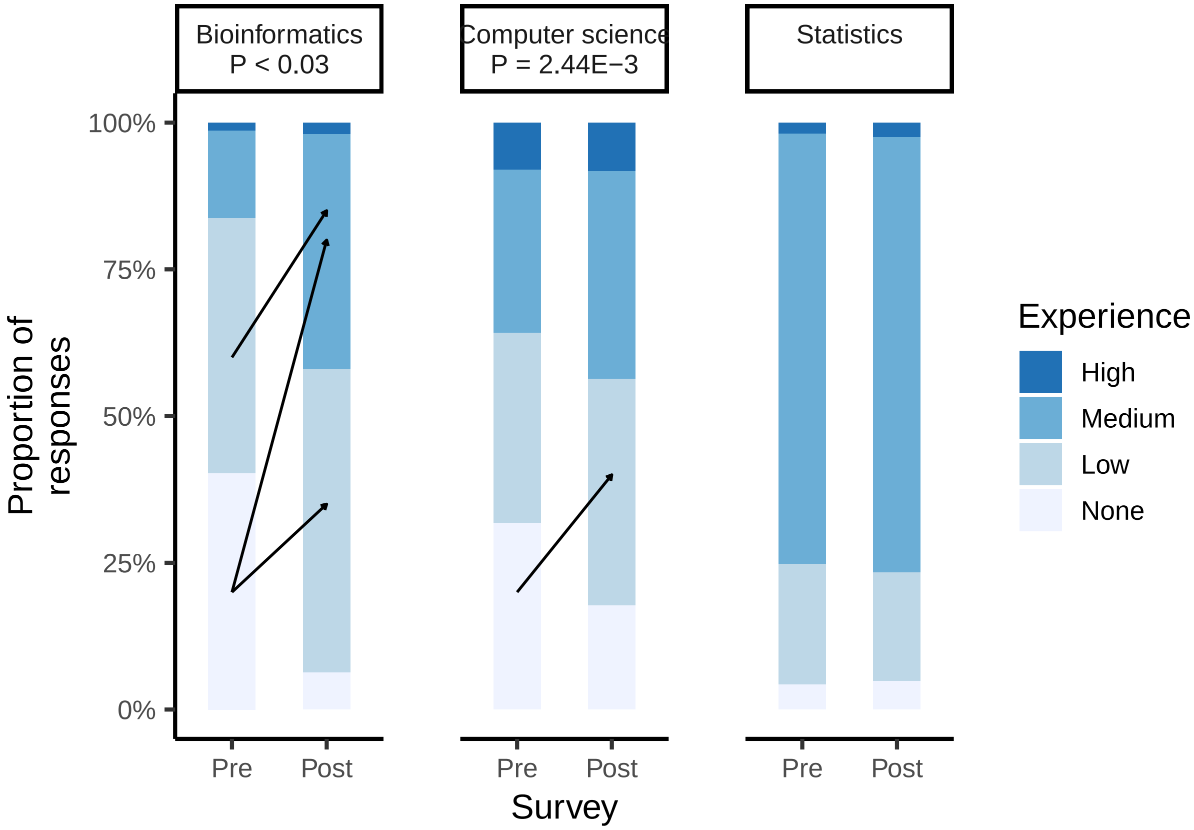
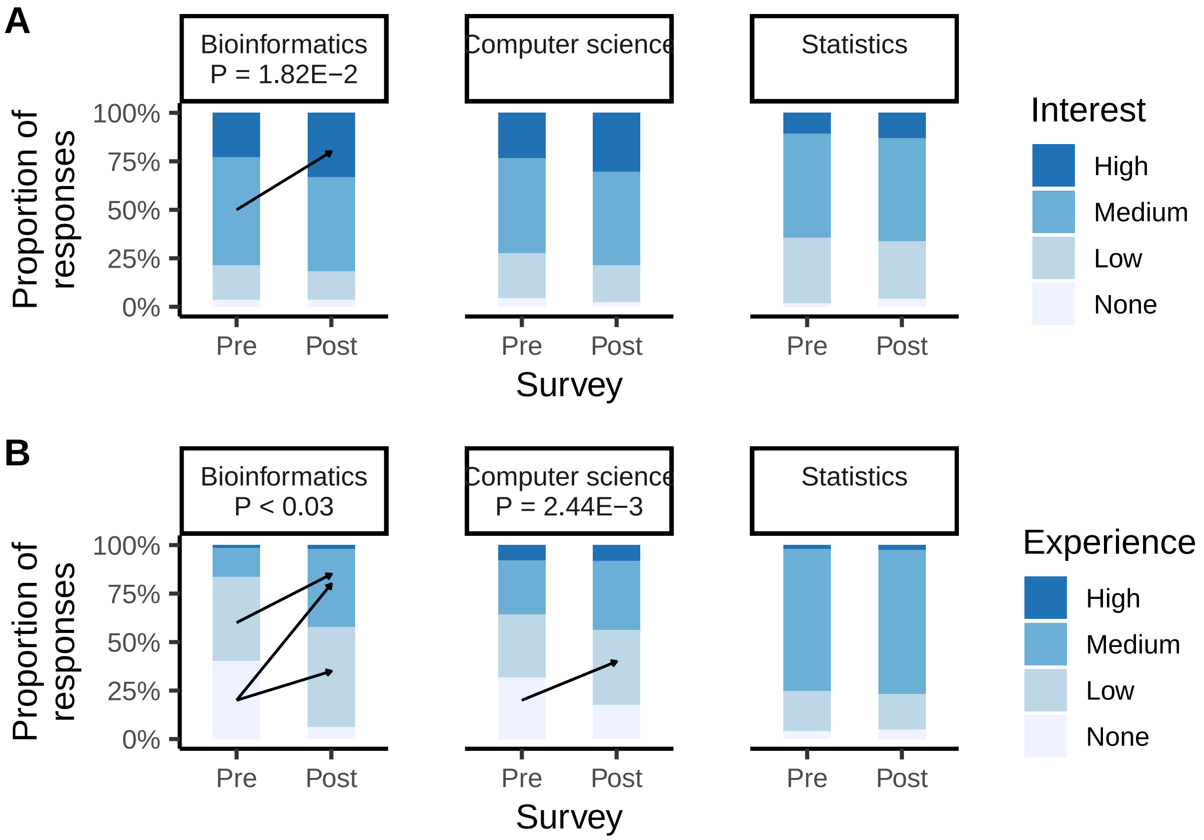


Figure 4. Student interest and experience in data science

Save composite figure

```
fig4 <- plot_grid(plot_I, plot_E, labels = c("A", "B"), nrow = 2, align = "v")

fig4
```



```r
ggsave(filename="Fig4.pdf", plot=fig4, width=19.05, height=14, units = "cm")
```